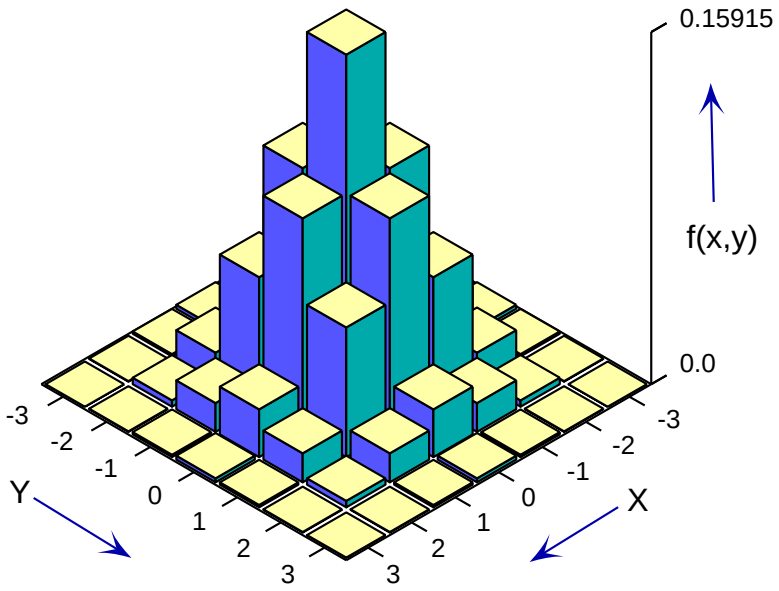
Simulation, fitting, statistics, and plotting.

W. G. Bardsley
https://simfit.uk
https://simfit.org.uk
https://simfit.silverfrost.com

Tutorials, Test Files, and Worked Examples.

Bivariate Normal Distribution

$\mu_x = \mu_y = 0$, $\sigma_x = \sigma_y = 1$, $\rho = 0$



Collected Tutorials: Version 8.0.0

Contents

1 Introduction

Tutorials and worked examples for simulation, curve fitting, statistical analysis, and plotting.
https://simfit.uk
https://simfit.org.uk
https://simfit.silverfrost.com

The SimFiT manual contains a description of all the procedures available to users including the necessary background theory together with worked examples. However, the manual was constructed for users with competence in mathematics and statistical understanding, so the worked examples were presented in a very succinct form which sometimes proves difficult to understand for users without these skills.

This set of tutorials and worked examples has been written to keep the mathematics and statistics to a minimum, and to present the worked examples in a simpler and more user-friendly form with more description and fewer equations. It is important that the the document `tutorials.pdf` must be consulted before attempting to follow instructions in the tutorials. However, it should be realized that the SimFiT reference can always be consulted for more details, and it must be pointed out that the tutorials are written with the understanding that users have installed the package and are aware of the following documents that are distributed with the package and are available from the SimFiT website.

- **tutorials.pdf**
  Must be read and understood before attempting to follow instructions in the tutorials
- **w_examples.pdf**
  The collected tutorials
- **w_manual.pdf**
  The comprehensive SimFiT manual with hyperlinks
- **mono_manual.pdf**
  The comprehensive SimFiT manual in monochrome PDF
- **install.pdf**
  Installation details
- **configure.pdf**
  Configuration details
- **speedup.pdf**
  Try these techniques when you are familiar with SimFiT and want to speed up execution by suppressing advisory messages
- **simfit_summary.pdf**
  Summary of the SimFiT package
- **source.pdf**
  Details of how to compile the simfit package from source code downloaded from the SimFiT website
- **pscodes.pdf**
  Tips for LaTeX and/or PostScript users who want to employ several advanced plotting procedures
- **ms_office.pdf**
  Description of how to interface SimFiT with suites such as MS Office, LibreOffice, and OpenOffice

2 Data preparation

Tutorials and worked examples for simulation, curve fitting, statistical analysis, and plotting.
https://simfit.uk
https://simfit.org.uk
https://simfit.silverfrost.com

2.1 Introduction to data preparation

The format required to input data into SIMFIT is extremely simple: all that is required is to read in a rectangular table of numbers with no missing values, either as a file or from the clipboard.

Data tables

However there are four alternative ways to indicate separation of data values into columns that will be accepted by SIMFIT as now discussed.

1. **Space separated variables**

   Here the columns are separated by spaces as in this example.

   ```
   11 12 13
   21 22 23
   31 32 33
   41 42 43
   ```

   This is how results files are formatted by SIMFIT, but this is not acceptable for table format using some word processors which generally require columns to be separated by tabs.

2. **Tab separated variables**

   Here the columns are separated by tabs (indicated by –>) as in this example.

   ```
   11->12->13
   21->22->23
   31->32->33
   41->42->43
   ```

   This is acceptable for tables using word processors and SIMFIT results tables can easily be transformed into this format, e.g. reading into Excel as a text file then exporting as a table for use by Word.

3. **Comma separated variables**

   Here the columns are separated by commas as in this example.

   ```
   11, 12, 13
   21, 22, 23
   31, 32, 33
   41, 42, 43
   ```

   This is the most common format used for data archiving, for instance *.csv files exported by Excel, and such tables can easily be transformed into tab format, e.g. reading into Excel as a *.csv file then exporting as a table for use by Word.

4. **Semicolon separated variables**

   This is when the continental practise of using commas for decimal points is used so that *.csv files are then exported from Excel in the following format

```
1,1; 1,2; 1,3
2,1; 2,2; 2,3
3,1; 3,2; 3,3
4,1; 4,2; 4,3
```

   instead of this format which is universally used in scientific work.

```
1.1 1.2 1.3
2.1 2.2 2.3
3.1 3.2 3.3
4.1 4.2 4.3
```

Input and output formats

SimFIT does not distinguish between integers and floating point numbers in data sets supplied for analysis, but does distinguish three data types of numbers in the files that are automatically created to archive results from analysis.

1. Integers such as 1, 1000, 1000000
2. Probability estimates such as 0.0004, 0.000039
3. Floating point numbers such as 1.2345 or, in exponential format, 1.2345E+00.

Of course all numbers used by SimFIT are represented internally either as integers to full significance or floating point numbers in 64-bit precision, but there are reasons for the way that SimFIT outputs floating point numbers.

Probabilities calculated for significance tests are at best only approximations and, as the numbers must lie between 0 and 1, four or sometimes six figures are output after the decimal point. More than this are for the blind leading the blind.

Calculations done with 64-bit precision have up to fifteen significant figures and it could be argued that results should be output with a similar number of significant figures. However experimental data rarely have more than four or five significant figures, and most calculations involve iterative procedures or approximations in any case so probably up to six significant figures should be sufficient.

Scientific notation

From version 8 users can configure SimFIT to use a selected number of significant figures in standard format or to use exponential notation with exactly six significant figures as follows.

Exponential	Standard
1.23456E+00	1.23456
1.23456E-01	0.123456
1.23456E-02	0.0123456
1.23456E-03	0.00123456
1.23456E-04	0.000123456
1.23456E+01	12.3456
1.23456E+02	123.456
1.23456E+03	1234.56
1.23456E+04	12345.6

However, to avoid monstrosities like 0.0000000012345 in output tables, SimFIT always outputs very small or very large numbers in exponential notation The advantage of exponential notation is that all output tables will be neatly formatted irrespective of the size of the numbers, and orders of magnitude can be seen at a glance. The disadvantage is that users not conversant with exponential notation may find confusing a notation that even programs like Excel have to resort to for very large or very small numbers. Actually it is very easy to replace exponential format by floating point format by importing a table formatted by SimFIT into Excel as a text file and then changing inside Excel from [scientific] format to [number] format using the [cells format] option. Such tables can then be imported into Word.

Row and Column labels

It has been explained that any rectangular data table with the same number of columns (i.e. variables) for each row (i.e. cases) can be imported into SimFIT by clipboard or file. However sometimes, by design in the case of multivariate data, or by accident otherwise, a full data matrix including row and column labels is submitted to SimFIT for analysis.

There are two ways when this is acceptable.

1. **All labels consist of a single word**

   In this example the interword spaces have been replaced by minus signs but any non-blank character such as an underscore can be used. Note the use of * to indicate that this is neither a row or a column label and so will be ignored.

*	column-1	column-2	column-3
row-1	11	12	13
row-2	21	22	23
row-3	31	32	33
row-4	41	42	43

2. **All labels are included within double quotes**

   This is done in the next example.

"*"	"column 1"	"column 2"	"column 3"
"row 1"	11	12	13
"row 2"	21	22	23
"row 3"	31	32	33
"row 4"	41	42	43

Conclusion

Any matrix with $n \geq 1$ rows and $m \geq 1$ columns can be submitted to SIMFIT for analysis from file or clipboard as long as two criteria are satisfied

1. All n rows have precisely m columns and there are no non-numerical or missing values.
2. If it is intended to use a data matrix with non-numerical data for row and column labels these must be as single words or be quoted.

The columns can be space, tab, comma, or semicolon separated and labels are not limited to one or two words as long as they are filled out with non-blank characters as in `Time_of_Day`, or surrounded by double quotes as in "Score on a scale of 1 to 10". For Excel users the macro `simfit6.xls` in the `C:\Program Files\Simfit\doc` folder can be used to transform arbitrary matrices into SIMFIT format.

2.2 Simfit data files

It has been explained that all SIMFIT requires to perform analysis is a rectangular table of numbers, with or without row and column labels. So why is there a need for data files formatted according to the SIMFIT convention? There are two answers to this question.

1. It is very easy to create data files according to the SIMFIT convention using any text editor, the programs supplied by SIMFIT, or the Excel macro **simfit6.xls**. Such files facilitate archiving and repeated analysis.
2. Many procedures used by SIMFIT require additional information such as starting values and ranges of parameter values for curve fitting, initial conditions from which to advance the solution of differential equations, flags to indicate which variables are to be included in multivariate analysis, etc.

Example 1

Consider a very simple example, namely the SIMFIT default ANOVA test file `anova1.tf1` shown below.

```
1-way ANOVA data from Zar: Biostatistics 3rd. edn. p-213
     6     5
  28.2  39.6  46.3  41.0  56.3
  33.2  40.8  42.1  44.1  54.1
  36.4  37.9  43.5  46.4  59.4
  34.6  37.1  48.8  40.2  62.7
  29.1  43.6  43.7  38.6  60.0
  31.0  42.4  40.1  36.3  57.3
     5
line 1: title for this data set
line 2: number of rows then number of columns
line 3: first row of data values
line 8: last row of data values
line 9: number of additional comment lines in the file
```

This data file has three sections as follows.

1. **The Header**
   The first line is the title and this is very useful as many SIMFIT procedures output results tables with the titles to identify the data set. It is also very convenient to scan the first line of a data file to quickly remind you about the contents. The second line is the size in the form of the number of rows (6 in this case) and the number of columns (5 in this case). There are some SIMFIT functions that must have these two dimensions in order to make decisions about the type of data. For instance, some graphics and curve fitting procedures.
   Note that, although a header section is not always required, it is very useful to supply one.

2. **The Data**
   This is just the rectangular table of data values with no missing values.

3. **The Trailer**
   In this example the first line of the trailer has the number of extra lines appended to the data. This value is not always necessary but is useful for some SIMFIT programs that edit data files. Also note that, in this case, the only material contained in the trailer section is advisory information. However this is not always the case. Although a trailer section is never vital and can always be omitted, nevertheless there are many circumstances when extremely important information required by SIMFIT can be conveniently added to the trailer which greatly simplifies analysis. This will be clear after analyzing another test file.

Example 2

Now consider another typical SIMFIT data file, namely `kmeans.tf1`, the default file to illustrate K-means clustering. Note that line numbers have been included in the first column of the following table for reference only and are not part of the actual data file.

Line 1	Data for 5 variables on 20 soils ...				
Line 2	20	5			
Line 3	77.3	13.0	9.7	1.5	6.4
Line 4	82.5	10.0	7.5	1.5	6.5
Line 5	66.9	20.6	12.5	2.3	7.0
Line 6	47.2	33.8	19.0	2.8	5.8
Line 7	65.3	20.5	14.2	1.9	6.9
Line 8	83.3	10.0	6.7	2.2	7.0
Line 9	81.6	12.7	5.7	2.9	6.7
Line 10	47.8	36.5	15.7	2.3	7.2
Line 11	48.6	37.1	14.3	2.1	7.2
Line 12	61.6	25.5	12.9	1.9	7.3
Line 13	58.6	26.5	14.9	2.4	6.7
Line 14	69.3	22.3	8.4	4.0	7.0
Line 15	61.8	30.8	7.4	2.7	6.4
Line 16	67.7	25.3	7.0	4.8	7.3
Line 17	57.2	31.2	11.6	2.4	6.5
Line 18	67.2	22.7	10.1	3.3	6.2
Line 19	59.2	31.2	9.6	2.4	6.0
Line 20	80.2	13.2	6.6	2.0	5.8
Line 21	82.2	11.1	6.7	2.2	7.2
Line 22	69.7	20.7	9.6	3.1	5.9

Lines numbered 1 to 2 are the optional header while lines 3 to 22 contain the data table as follows.

Line 1 This is the title of the data set

Line 2 This contains the size, i.e. the number of rows and columns

Line 3 to **Line 22** contain the 20 by 5 data table

Line 23	44 (i.e., Number of extra lines)
Line 24	Usage:
Line 25	Select statistics, then run program simstat, choose
Line 26	multivariate statistics, then go to K-means clustering
Line 27	
Line 28	The next line defines the starting clusters for k = 3
Line 29	begin{values} <– token to flag start of appended values
Line 30	82.5 10.0 7.5 1.5 6.5
Line 31	47.8 36.5 15.7 2.3 7.2
Line 32	67.2 22.7 10.1 3.3 6.2
Line 33	end{values}

Lines 23 to 28 are simply advisory but lines 29 to 33 illustrates the technique to set default starting estimates for the K-means clusters centroids. Note here how SIMFIT data files and user-supplied model files define various environments (in this case the environment is values) using flags as in

begin{values} ... end{values}

The final part of the trailer section contains lines 34 to 67 as follows.

Line 34	
Line 35	The next line defines the variables as 1=include, 0=suppress
Line 36	begin{indicators} <– token to flag start of indicators
Line 37	1 1 1 1 1
Line 38	end{indicators}
Line 39	
Line 40	The next line defines the row labels for plotting
Line 41	begin{labels} <– token to flag start of row and column labels
Line 42	A
line 43	B
Line 44	C
Line 45	D
Line 46	E
Line 47	F
Line 48	G
Line 49	H
Line 50	I
Line 51	J
Line 52	K
Line 53	L
Line 54	M
Line 55	N
Line 56	O
Line 57	P
Line 58	Q
Line 59	R
Line 60	S
Line 61	T
Line 62	V1
Line 63	V2
Line 64	V3
Line 65	V4
Line 66	V5
Line 67	end{labels}

Here we see that two further environments are defined.

1. **indicators**
   Lines 36 to 38 define the indicators, i.e. which variables are to be included in the analysis using the scheme that a 1 indicates a variable to be included while a 0 indicates a variable to be suppressed. So here the default position is to include all variables.

2. **labels**
   Lines 41 to 67 define the labels. First the row labels in lines 42 to 61 then the column labels in lines 62 to 66. These labels can then be used to identify rows and/or columns when graphs are plotted. It is recommended to use very short labels, as done here, to avoid confusion resulting from long labels.

Note that environments defining parameters such as values, indicators, and labels as illustrated in this test file can be placed anywhere in the trailer section. SIMFIT simply scans the trailer section of data files for appropriate environments and, if none are found, it uses default settings which can be edited retrospectively as required. However it should be pointed out that with many advanced SIMFIT techniques, such as constrained nonlinear regression or simulating and fitting differential equations, supplying starting estimates or initial conditions is very much easier if these are appended to the individual data sets.

2.3 Creating and editing Simfit data files

It has been explained that data for analysis by SIMFIT must be supplied in the form of a rectangular table of numbers with no missing values. Further, row and column labels can be present as long as they either have no spaces as in `Time_of_Day` or are double quoted as in `"Time of Day"`.

So, for many purposes, it is adequate merely to copy the table to the clipboard from a spreadsheet program such as Excel or Calc, and then use the [Paste] button on the SIMFIT file opening control, which simply makes a temporary file in SIMFIT format from the clipboard data. There is nothing wrong with this way of proceeding but two things must then be realized.

1. **Archiving**

   Doing it this way using copy and paste means it has to be done every time you want to repeat the process, for instance, to fit several models to the same data. Saving well-named data files with short meaningful titles makes retrospective use so much easier. In addition, it permits the gathering of files together to fit several data sets simultaneously, or plot multiple sets of coordinates, say using SIMFIT library files.

2. **Environments**

   Many analytical procedures require more than just the data table. For instance.

   (a) Setting parameter starting estimates and limits for nonlinear model fitting.

   (b) Providing initial conditions and range for numerical solution of differential equations, as well as the limits and number of points for plotting trajectories.

   (c) Defining starting clusters for K-means clustering.

   (d) Indicating variables to include or suppress in multivariate analysis.

   (e) Assigning variables to groups as in canonical correlation.

   (f) Adding row and column labels to data files to use in multivariate analysis plots.

   Evidently it would be extremely tedious to have to do this every time analysis is carried out using the clipboard to copy and paste spreadsheet data data into SIMFIT for analysis, as these additional parameters would then have to be edited interactively each time for use by the calling program.

Before proceeding any further an important point must be made about such data files.

SimFIT data files are simple ASCII text files which means that, given such a file, it is easy to edit it retrospectively in any text editor, such as notepad, in order to add, remove, or edit any of the information in it.

However, if rows of numbers are added to the data table or removed from it, then the first integer on the second line of the file which indicates the number of rows must be corrected.

There is also another matter which may cause concern if it is not understood.

Numbers in data files prepared by SimFIT are usually represented in scientific notation with a fixed number of significant figures. So if you input 1, 2, 3 from the clipboard it will be written to file as 1.0000000E+00, 2.0000000E+00, 3.0000000E+00 or similar. Of course calculations by SimFIT are carried out to 64-bit precision, so in the unlikely event that you do want to input data with more significant figures, just input in CSV format.

These are the ways to create data files in the SimFIT format.

1. Paste in from the clipboard but then save the temporary file created with a new name.
2. Use a macro with your spreadsheet program. For instance `simfit6.xls` with Excel.
3. Read a spreadsheet export file into program **maksim** or paste a table in from the clipboard.
4. Create a data file using a text editor such as **notepad**, or better **notepad++**.
5. Create a data file using one of the SimFIT programs such as **makfil** for curve fitting files, or **makmat** for arbitrary data tables.

Having created a data file then any environments that need to be added can be pasted in anywhere at the end of the data table using a text editor.

For small data sets it may be convenient to create data files using the SimFIT file creating programs **makmat** and **makfil** which guarantees correctly formatted data files. Then, for simple editing to correct, add, or deleet a few values it is probably easiest to use a text editor like **notepad**. However, when it comes to serious editing of data files the SimFIT data file editing programs **editmt** and **editfl** provide many procedures that are very difficult if not impossible to perform using a text editor or spreadsheet program. So the following features of the SimFIT data preparation and editing programs should be realized before the functionalities are discussed.

The SimFIT data preparation programs may only be useful when creating a file from relatively small data sets, but do have some advantages that will be outlined. The SimFIT file editing programs read in a source file and output a target file, but the source file will never be altered.

Standard data files

SimFIT program **makmat**

With this program you can simply type in numbers into an empty grid in the usual way. However, in order to facilitate the creation of special data sets, matrices can be zeroed with selected numbers, which can be very useful where diagonals have special significance. After filling in all the cells the matrix can be edited before exit. If the file creation process is closed before all the cells are filled in then uncompleted cells are set to a fixed number (1 or 1000000) which can only be changed by further editing.

SimFIT program **editmt**

Some of the functionality is summarized.

- Patches of the matrix can be written to file and new patches can be added from files. This is a very useful way to fuse multiple data sets that all have the same number of rows, or alternatively the same number of columns.
- Individual rows or columns can be deleted or restored which is a convenient way to swap rows or columns
- Individual rows or columns can be transformed by algebraic, probability, or trigonometric functions.
- Individual rows or columns can be set to fixed values.
- The total matrix can be edited to change selected values or for such processes as centering, scaling, or centering and scaling rows or columns. Such editing can be aborted at any stage without overwriting the current default matrix.
- On exit the title and trailer section of the data set can be edited.

Of course users must be aware of the need to proceed in an orderly and methodical fashion if these procedures are to applied sensibly with the desired mathematical results.

Curve fitting Files

There are also several special considerations with curve fitting files that must be considered briefly here, noting that there is much more detail on this subject in the SimFIT reference manual.

These have either two columns x and y, or three columns x, y, and s which have the following meanings.

X in column 1

The independent variable known with great accuracy, e.g. time, weight, concentration.

Usually x values are increasing order because of four reasons.

1. The first x_i, y_i pairs are used to obtain starting estimates for model parameters that have influence at low x.
2. The last x_i, y_i pairs are used to obtain starting estimates for model parameters that have influence at high x.
3. Numerical estimation of differential equations is best done sequentially onwards from the initial conditions to avoid unnecessary re-calculations.
4. SimFIT parses the data first time and assigns logical variables to identify groups of replicates so that the model error is only calculated for the first member of each group of replicates to avoid unnecessary re-calculations.

Y in column 2

The measured response assumed to result from random experimental error added to a deterministic effect.

S in column 3

The weights for fitting are calculated using $w_i = 1/s_i^2$.

There are five possibilities, all of them being controversial.

1. All $s_i = 1$. Constant variance is assumed.
   This is also the case when only two columns x the y are supplied. In other words, there is no such thing as unweighted regression.
2. The s_i are investigated independently and are know accurately.
   This is unquestionably the best method but is seldom used.
3. The s_i are estimated using the sample standard deviations based on replicates.
   This is only acceptable if the sample sizes are sufficiently large, definitely ≥ 5.
4. The s_i are assumed to be functions of the data i.e., y_i.
   This means that replicates will be weighted differently.
5. The s_i are assumed to be functions of the best-fit model.
   Whatever functional dependence is assumed the weights will be different for each iteration and only make sense if the fitted model is actually the correct one, the assumed functional dependence is correct, and in addition the weights only become asymptotically reliable as the regression converges to the solution point.

SIMFIT program **makfil**

The user can choose to make a x, y or a x, y, s file and can choose whether to input x in increasing order or, for special use where this is not necessary, in arbitrary order. Note that x can also be input for data such as those from doubling dilution experiments as described in the information available when the program is run, but this option is only to be used when it is properly understood. If the option to make a x, y file is chosen, the output file will have a third column with $s_i = 1$.

As this program is designed to prepare data files for curve fitting you will be forced to only input x in increasing order unless this option has been suppressed, and if you choose to make a x, y, s file, you will be forced to input meaningful s_i values with $s_i > 0$.

Note that you can plot the x, y values when the data input phase has been completed, and this is a very valuable way to check that sensible data have been input. So, if outliers are seen suggesting a typing error, this editing can be done before exit.

SIMFIT program **editfl**

Just as with **editmt** you specify a source file and a target file in case an undo functionality is required, and you can fuse multiple curve fittng data files together. A valuable feature is to rearrange data so that x is in nondecreasing order, and a check is provided to make sure that s_i, y_i pairs suggesting a sensible signal to noise ration have been input. If replicates have been provided these can be used to calculate weights and error bars, although some SIMFIT programs can do this at run-time.

Before final exit the ability to edit the title and trailer section is provided in case environments such as

```
begin{limits} ...  end{limits}
```

need to be added or updated.

The great advantage of using programs **makfil** and **editfl** is that the extensive checks for consistency, x order, sensible signal to noise ratios for y_i, s_i pairs, and visual checking for accidental outliers during the data input phase, greatly decreases the chance of a spurious result from trying to fit badly formatted data.

Missing values

Data tables used by SIMFIT must have no missing values. So, if you are in the unfortunate situation of requiring such dishonesty for the greater good, then you will have to use the Excel macro called `simfit6.xls`, or some other program dedicated to cheating in (one hopes) the least objectional way.

However there are sometimes cases where analysis of a matrix with unequal length columns can proceed and where missing values do not need to be replaced by estimates. For instance, 1-way ANOVA, analysis of multiple samples for equality of variance, creating box and whisker plots, etc. Such situations can be handled using individual column vectors, specifying data samples using a library file, or choosing individual samples from your project archive. Note that now these procedures can also use incomplete matrix files which will be described separately.

2.4 Incomplete matrices with missing values

Many procedures require sets of data with possibly unequal sample sizes, and the standard ways to deal with such situations in SIMFIT has been to use one of the following three techniques.

1. Opening a library file to reference sample files, e.g., `anova1.tfl`.
2. Opening individual files like `column1.tf1`, `column1.tf2` such as are specified in `anova1.tf1`.
3. Opening collections of files like `column1.tf1`, `column1.tf2` from your vector project archive.

There is now also the possibility of using an incomplete matrix file with missing values where the columns are padded out with non-numeric character strings like blanks, X, NA, etc. However, note that this technique is not intended for situations where a value is missing because an observation was not recorded, or because an outlier is suspected, and an estimate is to be used in order to allow statistical analysis to proceed. For this the Excel macro `simfit6.xls` can be used. Rather it is simply designed so that a single matrix can be used for convenience whenever the numbers of numeric data are not the same in every column of a matrix.

The format for incomplete matrix files

Here are the rules.

- In order for such data to be interpreted correctly the matrix columns must be separated in an unambiguous way, for example, using commas, semicolons, or tabs.
- The separators used cannot occur elsewhere in the file except as column separators.
- Down any given column the missing values can occur at the level of any row.
- Unambiguous non-numeric character strings must be used to pad out the cells with missing values.
- Headers and trailers can be added as long as they do not contain the column separators.
- If labels are required for plotting they cannot be added to the trailer but must be added from the configuration files, or interactively, e.g. from a separate labels file.

Example 1: comma separated variables

The SIMFIT test file `incomplete.tf1` uses commas as separators as follows.

```
23,   29,   38,   30,   31
27,   25,   31,   27,   33
26,   33,   28,   28,   31
19,   36,   35,   22,   28
```

```
30,   32,   33,   33,   30
  ,   28,   36,   34,   24
  ,   30,     ,   34,   29
  ,   31,     ,   32,   30
```

Such a file is just a normal comma separated file such as a `*.csv` file exported from any spread sheet program. Of course this format cannot be used if continental notation is used for non-integer values, such as using 1,234 for 1.234. Cells can use scientific notation for 1.234 such as 1.234E+00 or 0.1234E+01, 1.234e0, etc.

Example 2: semicolon separated variables

The SIMFIT test file `incomplete.tf2` uses semicolons as separators as follows.

```
23;   29;   38;   30;   31
27;   25;   31;   27;   33
26;   33;   28;   28;   31
19;   36;   35;   22;   28
30;   32;   33;   33;   30
  ;   28;   36;   34;   24
  ;   30;     ;   34;   29
  ;   31;     ;   32;   30
```

Such a file would be output as a `*.csv` file from a spreadsheet program set up to use commas instead of the decimal points used in standard scientific documents.

Example 3: tab separated variables

This how test file `incomplete.tf3` will look in **Notepad**

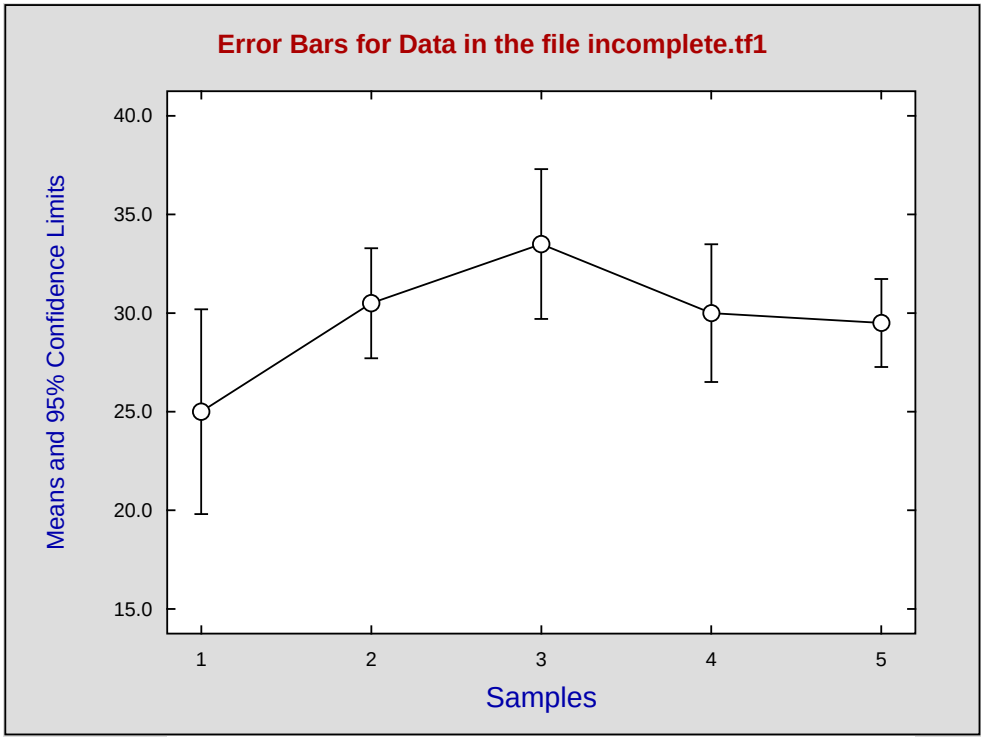
```
23    29    38    30    31
27    25    31    27    33
26    33    28    28    31
19    36    35    22    28
30    32    33    33    30
      28    36    34    24
      30          34    29
      31          32    30
```

while here is how it would look in **Notepad++** if the option to display tabs is switched on.

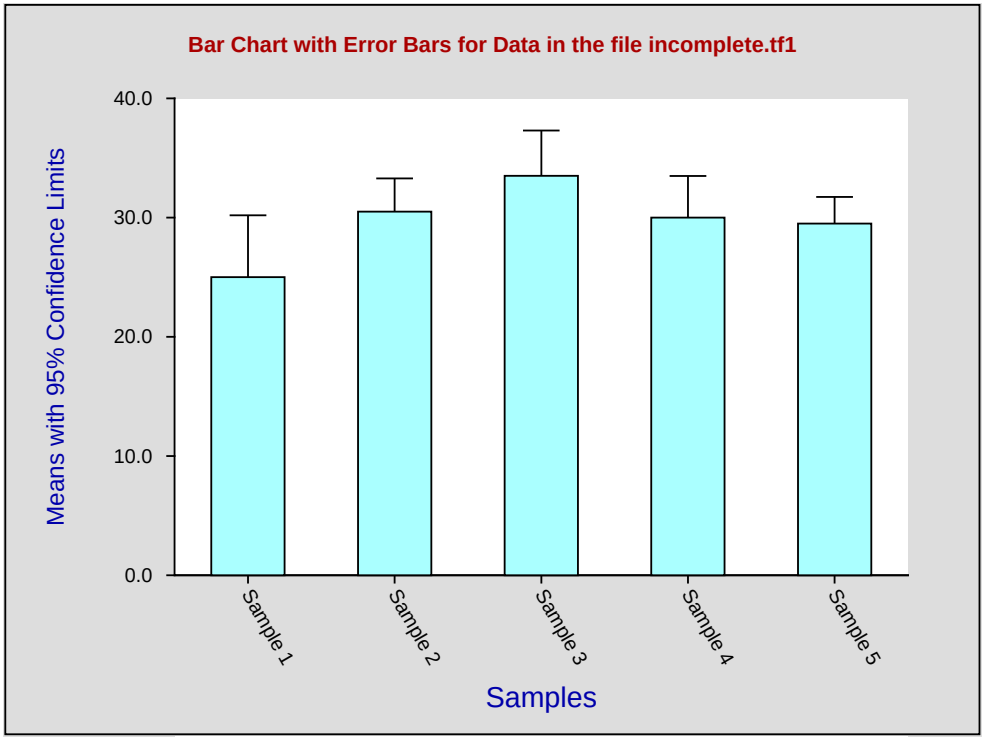
```
23->29->38->30->31
27->25->31->27->33
26->33->28->28->31
19->36->35->22->28
30->32->33->33->30
  ->28->36->34->24
  ->30->  ->34->29
  ->31->  ->32->30
```

This is usually the default format to copy to the clipboard or export to files with many spreadsheet programs, as it allows either continental or scientific notation to be used for non-integer values. However, considerable care must be taken when editing such files in a simple editor like **Notepad** which does not display tabs, and the more advanced editior **Notepad++** must be used to prevent accidental deletion of tabs, when it would no longer be recognized by SimFIT as an incomplete matrix file. Here are some of the graph plotting styles that can be used with both incomplete and complete SimFITdata files.

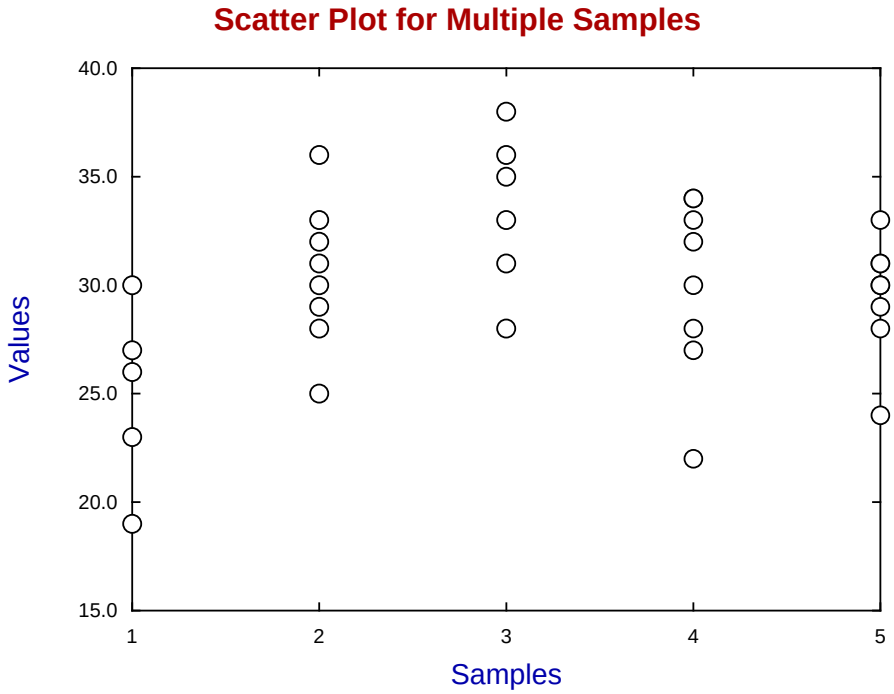
Plot 1: Means as symbols with error bars calculated from individual samples



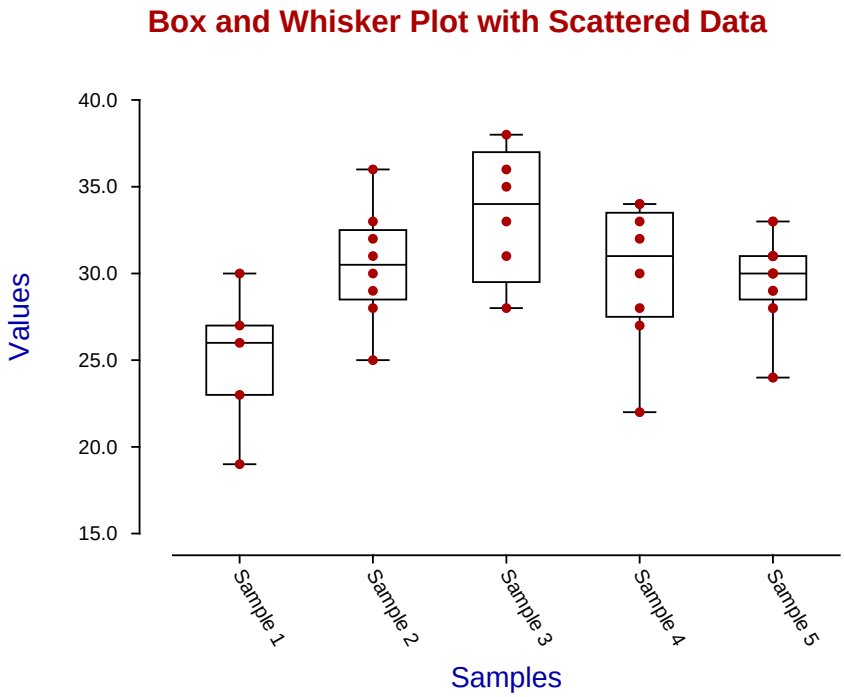
Plot 2: A bar chart plot with added error bars calculated as for Plot 1



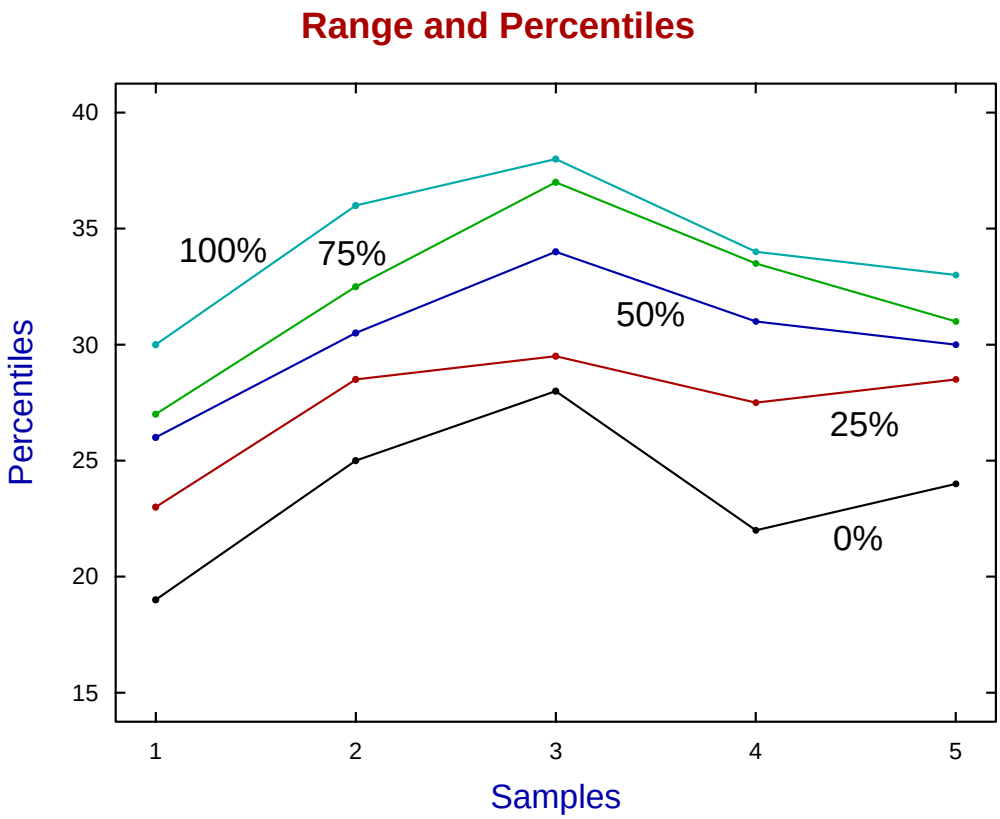
Plot 4: Showing all the sample values as clusters



Plot 5: A box and whisker plot with added data



Plot 6: Connecting the percentiles by straight line segments



3 Results files

Tutorials and worked examples for simulation, curve fitting, statistical analysis, and plotting.
https://simfit.uk
https://simfit.org.uk
https://simfit.silverfrost.com

3.1 Options for the number of significant digits in tables

Floating–point numbers (i.e., numbers with a decimal point) are stored by computers in binary form but data files for input into SimFIT and tables output with results are in text format. Given a number x, the range of values allowed by SimFIT for the absolute value of x to 3 significant digits (i.e., 3 digits not counting the decimal point) is

$$2.25 \times 10^{-308} \leq |x| \leq 1.79 \times 10^{308}$$

and the maximum number of significant digits allowed in this range is approximately 15.

SimFIT will accept data within such limits and performs calculations to double precision but, because all calculations are subject to rounding and truncation errors, only a lower number of significant digits, say 12, is required especially given that most calculations SimFIT performs involve iterative procedures like nonlinear optimisation. Further, scientific instruments are seldom accurate at this level of accuracy and in many cases it is not reasonable to accept results with more significant digits than used to represent the data. Because of the difficulty of reading and writing extremely large numbers there is a format called scientific notation to be explained next.

All floating–point numbers can be written concisely using powers of ten as multiplication factors as follows

$$1000000.0 = 1.0 \times 10^{6}$$
$$0.0000001 = 1.0 \times 10^{-6}$$

where the value and convenience of using powers of ten will be clear at a glance. To avoid superscripts and also to limit the number of characters required to represent numbers, scientific notation simply uses the idea of one digit in front of the decimal point and a fixed number of digits after the decimal point with the code E+ab for $\times 10^{ab}$ and E-ab for $\times 10^{-ab}$. Here for example is the number -1.23456 that unquestionably has 6 significant digits but multiplied by powers of ten as they would be displayed by SimFIT using Option 6 (scientific notation) and Option 7 (standard notation).

```
    Option 6         Option 7
-1.23456E+09     -1.23456E+09
-1.23456E+08     -1.23456E+08
-1.23456E+07      -12345600.0
-1.23456E+06       -1234560.0
-1.23456E+05        -123456.0
-1.23456E+04         -12345.6
-1.23456E+03         -1234.56
-1.23456E+02         -123.456
-1.23456E+01         -12.3456
-1.23456E+00         -1.23456
-1.23456E-01        -0.123456
-1.23456E-02       -0.0123456
-1.23456E-03      -0.00123456
-1.23456E-04     -0.000123456
```

```
-1.23456E-05     -0.0000123456
-1.23456E-06      -1.23456E-06
-1.23456E-07      -1.23456E-07
```

It will be clear that the scientific format has a fixed width with the numbers aligned at the decimal point whereas the numbers in standard notation are of variable width and, when the numbers become rather large or very small, SIMFIT Option 7 resorts to scientific notation.

The options for results

SIMFIT releases up to version 7 always displayed numbers in scientific notation with a field width and significant digits appropriate for the analysis being employed. To many analysts this is by far the most valuable way to display numbers as the field width is fixed, all numbers are aligned at the decimal points and orders of magnitude can be seen at a glance, but some SIMFIT users find this difficult to understand so from version 8 SIMFIT provides an interface that users can employ to change number format interactively. This is done using the following sequence of steps starting from the [Configure] option from the SIMFIT main page

```
[Configure] --> [Advanced] --> [Change number of significant digits in results tables]
```

which takes immediate effect after choosing the format required without requiring the [Apply] button to be pressed.

The ten options are

```
Option  1:  Up to  12 significant digits
Option  2:  Up to  11 significant digits
Option  3:  Up to  10 significant digits
Option  4:  Up to   9 significant digits
Option  5:  Exactly 7 significant digits
Option  6:  Exactly 6 significant digits (scientific notation)
Option  7:  Exactly 6 significant digits (standard notation: recommended)
Option  8:  Exactly 5 significant digits
Option  9:  Exactly 4 significant digits
Option 10:  Exactly 3 significant digits
```

which all use a field width of 13 characters. For this reason the number of significant digits in options 1 to 4 cannot be exact but are upper limits.

To illustrate the difference between Scientific notation and standard notation consider the results created by SIMFIT after fitting one then two exponentials using SIMFIT program **exfit** to analyse the test file `exfit.tf4` (which has six significant digits) in order to decide if fitting two exponentials after one exponential justifies the higher order model with extra parameters.

Option 6: Scientific notation

```
For best-fit 1-exponential function

Parameter     Value        Std.Error    Lower95%cl    Upper95%cl     p
   A(1)    1.69443E+00  2.67006E-02  1.63974E+00  1.74912E+00  0.0000
   k(1)    1.46094E+00  5.77654E-02  1.34261E+00  1.57926E+00  0.0000
   AUC     1.15982E+00  3.78135E-02  1.08237E+00  1.23728E+00  0.0000
AUC is the area under the curve from t = 0 to t = infinity
Initial time point (A)   =  3.59830E-02
```

```
 Final time point   (B)   =  1.61100E+00
 Area over range (A,B)    =  9.90210E-01
 Average over range (A,B) =  6.28698E-01

 For best-fit 2-exponential function

 Parameter    Value       Std.Error   Lower95%cl   Upper95%cl    p
    A(1)   8.52553E-01  6.77105E-02  7.13372E-01  9.91734E-01  0.0000
    A(2)   1.17644E+00  7.47538E-02  1.02278E+00  1.33010E+00  0.0000
    k(1)   6.79334E+00  8.54540E-01  5.03681E+00  8.54987E+00  0.0000
    k(2)   1.11206E+00  5.10959E-02  1.00703E+00  1.21709E+00  0.0000
    AUC    1.18339E+00  1.47096E-02  1.15316E+00  1.21363E+00  0.0000
 AUC is the area under the curve from t = 0 to t = infinity
 Initial time point (A)   =  3.59830E-02
 Final time point   (B)   =  1.61100E+00
 Area over range (A,B)    =  9.38322E-01
 Average over range (A,B) =  5.95754E-01

F test results

WSSQ-previous (WSSQ1)              =  2.24923E+02
WSSQ-current (WSSQ2)               =  2.43970E+01
Number of parameters-previous (M1) =      2
Number of parameters-current (M2)  =      4
Number of data points (NPTS)       =     30
Akaike AIC-previous                =  6.44368E+01
Akaike AIC-current                 =  1.79794E+00
Evidence ratio (ER)                =  3.99818E+13
Schwarz SC-previous                =  6.72392E+01
Schwarz SC-current                 =  7.40273E+00
Mallows Cp                         =  2.13701E+02
Mallows ratio (Cp/M1)              =  1.06851E+02
Numerator degrees of freedom       =      2
Denominator degrees of freedom     =     26
F test statistic (FS)              =  1.06851E+02
p = P(F >= FS)                     =  0.0000
1 - p = P(F =< FS)                 =  1.0000
5% upper tail point                =  3.36902E+00
1% upper tail point                =  5.52633E+00

Conclusion based on F test
Reject previous model at 1% significance level
There is strong support for the extra parameters
Tentatively accept the current best fit model
```

You will observe that all floating point numbers in these results tables have exactly six significant digits and all the numbers are lined up at the decimal point but use exponential notation for powers of ten.

Option7: Standard notation

Now performing the same analysis after selecting the default Option number 7 leads to the following analysis in which all the floating point numbers are still displaying six significant digits but now in standard format. Note however that, even in standard format, it is necessary to swap to scientific notation when the absolute value of the numbers become very large ($> 10^7$) or very small ($< 10^{-6}$). This is necessary to maintain a maximum width of 13 characters per number in multi–column tables.

```
 For best-fit 1-exponential function

 Parameter     Value       Std.Error    Lower95%cl    Upper95%cl    p
    A(1)       1.69445     0.0267064       1.63974       1.74915  0.0000
    k(1)       1.46101     0.0578035        1.3426       1.57941  0.0000
    AUC        1.15978     0.0378344       1.08228       1.23728  0.0000
 AUC is the area under the curve from t = 0 to t = infinity
 Initial time point (A)   = 0.035983
 Final time point   (B)   = 1.611
 Area over range (A,B)    = 0.990184
 Average over range (A,B) = 0.628681

 For best-fit 2-exponential function

 Parameter     Value       Std.Error    Lower95%cl    Upper95%cl    p
    A(1)      0.852548     0.0677272      0.713332      0.991763  0.0000
    A(2)       1.17645     0.0747742       1.02275       1.33015  0.0000
    k(1)       6.79344      0.854289       5.03743       8.54946  0.0000
    k(2)       1.11206     0.0511103         1.007       1.21712  0.0000
    AUC        1.18339     0.0147092       1.15316       1.21363  0.0000
 AUC is the area under the curve from t = 0 to t = infinity
 Initial time point (A)   = 0.035983
 Final time point   (B)   = 1.611
 Area over range (A,B)    = 0.938323
 Average over range (A,B) = 0.595754

F test results

WSSQ-previous (WSSQ1)              = 224.923
WSSQ-current (WSSQ2)               = 24.397
Number of parameters-previous (M1) = 2
Number of parameters-current (M2)  = 4
Number of data points (NPTS)       = 30
Akaike AIC-previous                = 64.4368
Akaike AIC-current                 = 1.79794
Evidence ratio (ER)                = 3.99818E+13
Schwarz SC-previous                = 67.2392
Schwarz SC-current                 = 7.40273
Mallows Cp                         = 213.701
Mallows ratio (Cp/M1)              = 106.851
Numerator degrees of freedom       = 2
Denominator degrees of freedom     = 26
F test statistic (FS)              = 106.851
p = P(F >= FS)                     = 0.0000
1 - p = P(F =< FS)                 = 1.0000
5% upper tail point                = 3.36902
1% upper tail point                = 5.52633

Conclusion based on F test
Reject previous model at 1% significance level
There is strong support for the extra parameters
Tentatively accept the current best fit model
```

3.2 Introduction to results file

Each time a procedure outputs a table of results to the display, a copy is written to a results file for archive purposes. So there is no need to copy the results at run-time, but it is important to appreciate the steps needed to retrieve this information for retrospective use. The way this SIMFIT scheme works will now be explained, noting that the files are saved in the folder `C:\...ProgramData\user name\Simfit\res`.

1. **File names of the results files**

   The 101 results files are named in sequence as follows.

   ```
   f$result.txt
   f$result.001
   f$result.002
   ...
   f$result.100
   ```

2. **Action taken each time a program starts**

   The last file `f$result.100` is deleted and the list of saved files is renamed so that `f$result.099` becomes `f$result.100` all the way down to `f$result.txt` which becomes `f$result.001`, then a new file `f$result.txt` is opened ready to receive the anticipated results.

3. **The format of results files**

   The results files are standard ASCII text files which, using a monospaced font like Courier, will be lined up as rectangular tables with associated header and trailer sections. They can be edited and printed using any text editor, such as program **notepad**.

4. **Notation used for numbers in the results files**

   These may be in scientific notation using decimal points as follows

   ```
   5.4321E-03   0.0054321
   5.4321E+00      5.4321
   5.4321E+03      5432.1
   ```

   where the number of significant digits is determined by the analytical technique being used. However, from SIMFIT Version 8 users can choose to output results in standard notation with a chosen number of significant digits. Of course all numbers are stored in the computer to full 64-bit precision, but the number output to tables is designed to reflect the number that would be meaningful. So, for instance, parameter estimates from curve fitting are usually only meaningful up to about six significant digits, while four decimal places is probably sufficient for probability estimates and significance tests.

5. **Extracting tables**

   The results files are available for viewing, printing, saving with new names, and for extracting tables as tabbed-text, html, xml, or LaTeX from the [Results] button on the main SIMFIT menu.

Advice

Finally, it should be obvious that any results that may be useful retrospectively must be saved before the file is rolled off the end of the list. Another issue is that when SIMFIT starts it checks that all results file that are effectively empty are deleted with subsequent rolling and renaming, which can sometimes cause a delay when SIMFIT starts. Such empty files are created when a program is started but not then used to create results.

Each time a data set is analyzed the results are written to a file called `f$result.txt`, and the current files are renamed so that the existing `f$result.txt` becomes `f$result.001` while `f$result.001` becomes `f$result.002` and so on. These can be viewed using the [Results] option from the main SimFIT menu.

These SimFIT results files are formatted so that the numbers displayed only contain the number of significant digits that are meaningful in context. For instance, probabilities will usually only have four digits after the decimal point, which indicates that it does not make any sense to consider any subsequent digits for purpose of statistical testing, and in any case probability estimates will not be accurate for more than about four digits. Furthermore, as experimental data are rarely more accurate than about three or four significant digits anyway, it may be wishful thinking to ever consider more than say six. In addition, the tables are formatted using a fixed font with scientific notation to line up column entities irrespective of absolute size, and many users do not want this in a thesis or published document. Naturally, these arguments do not apply to integers.

As an example consider the following case with the title Table 1.

```
Table 1
1-Way Analysis of Variance:  1 (Grand Mean  43.16)

Transformation:-  x (untransformed data)
Source               SSQ     NDOF        MSQ          F        p
Between Groups    2193.0        4      548.4      56.15      0.0000
Residual           244.1       25      9.765        ...         ...
Total             2438.0       29        ...        ...         ...
```

Actually, most users would want to import such a table formatted as tabbed-text, html, xml, or LaTeX into documents such as a report, thesis, or publication looking something like Table 2.

Table 2
1-Way Analysis of Variance: 1 (Grand Mean 43.16)
Transformation:- x (untransformed data)

Source	SSQ	NDOF	MSQ	F	p
Between Groups	2193.0	4	548.4	56.15	0.0000
Residual	244.10	25	9.765		
Total	2438.0	29			

Or even, for those with artistic leanings, possibly something like Table 3.

Table 3
1-Way Analysis of Variance: Grand Mean 43.16
Transformation: *x* (untransformed data)

Source	SSQ	NDOF	MSQ	*F*	*p*
Between Groups	2193.0	4	548.4	56.15	0.0000
Residual	244.10	25	9.765		
Total	2438.0	29			

This article explains the procedures required to export tables from from SIMFIT results files into forms suitable for inclusion into word processors, spreadsheet programs, website scripts, or even professional document preparation systems such as LaTeX. Also, decimal points can be replaced by commas as in continental notation if required.

The procedure

It is important to realize that the [Results] option from the SIMFIT main menu gives access to all the currently saved results files.

- **Choosing a results file.**

  Early versions of SIMFIT allowed users to name results files individually to avoid anything being lost. However, now that up to 100 results files are saved and users have the option [Results] from the main SIMFIT menus from which to view, print, save, edit, or make tables, this is no longer usually necessary. Clearly, if results are always required for retrospective use, regular back-up or saving will be necessary.

- **Extracting a table.**

  From the [Make tables] option view the file to make sure it is the one required then copy to the clipboard only the table required along with any associated header and trailer sections ... but nothing else.

- **Preliminary editing.**

  Sometimes editing of the file is required to make sure that every row of the table has exactly the same number of columns. So note that, for extracting a table there can be no empty cells, and each cell must contain precisely one word. Any column titles must be edited so that they consist of one word, for instance changing `Time of Day` to `Time_of_Day` for instance, or filling empty cells by three dots. Added underscores and sets of three contiguous dots can be removed when the final table is written to file. A pre-processing option is provided for editing before attempting to create a table.

  Note that often tables have cells with added comments relating to goodness of fit or results of statistical testing, and these no not need to be underscored. There are also special tables with only two columns containing several words in some cells, and Example 1 later will make this clear.

- **Viewing the hashtag table.**

  The algorithm attempts to identify cells in a table by inserting a hashtag between every column. If the algorithm succeeds there would be no need to view this hashtag table. However this option should be switched on until the process of the algorithm is understood, or if it fails and you need to see why.

  The hashtag table is clarified in Example 1.

- **Headers and trailers.**

  Frequently tables have header and trailer sections that are descriptive and not part of the table itself. As these can have strings of word and numbers that would confuse the algorithm checking that every row must have the same number of columns, they must be identified. This is done by using buttons on a window that allows the header and trailer lines to be highlighted. If this is not done the table creating algorithm will fail.

  Selecting headers and trailers is clarified in Example 2.

- **Saving the table.**

  For programs that produce Windows quality hardcopy the table should be saved as html or xml as these will import directly into word processors or spreadsheets. Tabbed-text is also available but is much less versatile than html or xml, and LaTeX is available for those up to it.

- **Fine tuning.**

  Inevitably there will often be the need for dealing with details. For instance, users will sometimes want to replace `alpha` by α or `Chisd` by χ^2 and this can be done for html, xml, and LaTeX output but not for

tabbed-text. On the other hand html, xml, and LaTeX have reserved letters and these must be dealt with retrospectively.

For instance, consider the transformation of the following expressions which can be done in html, xml, and LaTeX but not in tabbed text.

Original	Transformed
>=	$\geq$
=<	$\leq$
alpha	α
beta	β
P(Chisqd >= TS) = 0.2037	$P(\chi^2 \geq TS) = 0.2037$
p_0 + p_1x + p_2x$\hat{2}$	$p_0 + p_1 x + p_2 x^2$

In cases where ambiguity could arise in LaTeX using underscores or similar special characters they will be replaced by question marks. So LaTeX users should search for ? characters that will need replacing in the final table.

- **Padding with zeros.**

  As the [Make tables] option will never remove significant digits a problem arises if users wish to replace numbers in scientific notation by floating point representation. In order to extend the range over which this can be done, padding zeros can be introduced as illustrated in this next table.

Scientific notation	Padding	Floating point representation
1.234E-01	0	.1234
1.234E+00	0	1.234
1.234E+01	0	12.34
1.234E+02	0	123.4
1.234E-02	2	0.01234
1.234E-01	2	0.12340
1.234E+00	2	1.23400
1.234E+01	2	12.3400
1.234E+02	2	123.400
1.234E+03	2	1234.00
1.234E+04	2	12340.0
1.234E-04	4	0.0001234
...	...	...
1.234E+06	4	1234000.0

Evidently increasing the number of padding zeros increases the range over which transformation from scientific to floating point representation can be achieved, and the default is two which allows a wide range, but four ensures that a mixture of transformed and untransformed numbers will occupy the same width in the columns of a table.

However, there is here a problem because adding padding zeros could suggest all trailing zeros are meaningful. For instance, the number 1.2341213179 stored in the computer could be written as 1.234E+00 in SimFIT output because the analysis in question only justifies accuracy or meaning for up to four significant digits. However, 1.23400 could be mistaken for indicating the internal representation with six significant digits. So users may wish to suppress trailing zeros in such cases, noting that this could result in numbers with different widths in a column. In any case, setting the number of padding zeros to -1 switches off the transformation from scientific notation to floating point.

Sometimes, for instance with numerical analysis where more significant digits are justified than with data analysis, eight significant digits are output, and some procedures can optionally allow more. In addition special DLLs dedicated for particular routines can be supplied for this purpose.

Example 1

This example shows the transformation of a special type of SIMFIT table containing just two columns separated by equals signs (i.e., =) and containing cells with multiple words. As long as the equals signs are perfectly lined up and there is no header or trailer section, this type of table can always be transformed. Here is the table with no header or trailer sections as extracted using the [Results] then [Make tables] options from the main SIMFIT menu using the results file `f$result.txt` following the exhaustive analysis of a vector process used to analyze data contained in the default test file `normal.tf1`.

```
Sample size                    =      50
Minimum value                  = -2.20820E+00
Maximum value                  =  1.61750E+00
Coefficient of skew            = -1.66905E-02
Coefficient of kurtosis        = -7.68395E-01
Lower Hinge  (25th percentile) = -8.55015E-01
Median value (50th percentile) = -9.73615E-02
Upper Hinge  (75th percentile) =  7.85965E-01
Sample mean                    = -2.57897E-02
Sample standard deviation      =  1.00553E+00
Coefficient of variation (CV%) =  > 100%
Standard error of the mean     =  1.42203E-01
Upper 2.5% t-value             =  2.00958E+00
Lower 95% con lim for mean     = -3.11558E-01
Upper 95% con lim for mean     =  2.59978E-01
Sample variance                =  1.01109E+00
Lower 95% con lim for variance =  7.05519E-01
Upper 95% con lim for variance =  1.57006E+00
Shapiro-Wilks W statistic      =  9.62693E-01
Significance level for W       =  0.1153
Conclusion                     = Tentatively accept normality
```

This is the corresponding intermediate hashtag table.

```
Sample size                    #      50
Minimum value                  # -2.20820E+00
Maximum value                  #  1.61750E+00
Coefficient of skew            # -1.66905E-02
Coefficient of kurtosis        # -7.68395E-01
Lower Hinge  (25th percentile) # -8.55015E-01
Median value (50th percentile) # -9.73615E-02
Upper Hinge  (75th percentile) #  7.85965E-01
Sample mean                    # -2.57897E-02
Sample standard deviation      #  1.00553E+00
Coefficient of variation (CV%) #  > 100%
Standard error of the mean     #  1.42203E-01
Upper 2.5% t-value             #  2.00958E+00
Lower 95% con lim for mean     # -3.11558E-01
Upper 95% con lim for mean     #  2.59978E-01
Sample variance                #  1.01109E+00
Lower 95% con lim for variance #  7.05519E-01
Upper 95% con lim for variance #  1.57006E+00
Shapiro-Wilks W statistic      #  9.62693E-01
Significance level for W       #  0.1153
Conclusion                     # Tentatively accept normality
```

The hashtag table is very useful for detecting the source of errors. The table making algorithm attempts to locate the position separating columns and writes a hashtag there. If every row has the same number of columns then every row will have the same number of hashtags and the algorithm has succeeded. Observing this hashtag table when the algorithm has failed will allow you identify then correct the error.

Here is the selected table as it would be written to the output file.

Sample size	50
Minimum value	-2.2082000
Maximum value	1.6175000
Coefficient of skew	-0.0166905
Coefficient of kurtosis	-0.7683950
Lower Hinge (25th percentile)	-0.8550150
Median value (50th percentile)	-0.0973615
Upper Hinge (75th percentile)	0.7859650
Sample mean	-0.0257897
Sample standard deviation	1.0055300
Coefficient of variation (CV%)	> 100%
Standard error of the mean	0.1422030
Upper 2.5% t-value	2.0095800
Lower 95% con lim for mean	-0.3115580
Upper 95% con lim for mean	0.2599780
Sample variance	1.0110900
Lower 95% con lim for variance	0.7055190
Upper 95% con lim for variance	1.5700600
Shapiro-Wilks W statistic	0.9626930
Significance level for W	0.1153
Conclusion	Tentatively accept normality

Here it is with a few minor cosmetic changes.

Exhaustive analysis of a vector

Sample size	50
Minimum value	-2.2082000
Maximum value	1.6175000
Coefficient of skew	-0.0166905
Coefficient of kurtosis	-0.7683950
Lower Hinge (25th percentile)	-0.8550150
Median value (50th percentile)	-0.0973615
Upper Hinge (75th percentile)	0.7859650
Sample mean	-0.0257897
Sample standard deviation	1.0055300
Coefficient of variation (CV%)	> 100%
Standard error of the mean	0.1422030
Upper 2.5% t-value	2.0095800
Lower 95% confidence limit for mean	-0.3115580
Upper 95% confidence limit for mean	0.2599780
Sample variance	1.0110900
Lower 95% confidence limit for variance	0.7055190
Upper 95% confidence limit for variance	1.5700600
Shapiro-Wilks W statistic	0.9626930
Significance level for W	0.1153
Conclusion: *Tentatively accept normality*	

Example 2

From fitting a two-exponential model to data in the test file `exfit.tf4` using program **exfit** with scientific notation the following results can be extracted from the results file.

```
For best-fit 2-exponential function

Parameter    Value       Std.Error   Lower95%cl   Upper95%cl     p
   A(1)    8.5255E-01   6.7731E-02   7.1332E-01   9.9177E-01   0.0000
   A(2)    1.1765E+00   7.4779E-02   1.0227E+00   1.3302E+00   0.0000
   k(1)    6.7935E+00   8.5386E-01   5.0383E+00   8.5486E+00   0.0000
   k(2)    1.1121E+00   5.1128E-02   1.0070E+00   1.2172E+00   0.0000
   AUC     1.1834E+00   1.4714E-02   1.1531E+00   1.2136E+00   0.0000
AUC is the area under the curve from t = 0 to t = infinity
Initial time point (A)    =  3.5983E-02
Final time point   (B)    =  1.6110E+00
Area from t = A to t = B =  9.3832E-01
Average over range (A,B) =  5.9575E-01
```

Now the file has an additional head and trailer section so, if the full table is selected, it will have to be highlighted as follows in the header and trailer selection control as shown next, where the header is colored magenta and the trailer colored cyan.

```
For best-fit 2-exponential function

Parameter        Value     Std.Error   Lower95%cl   Upper95%cl     p
   A(1)      8.5255E-01   6.7731E-02   7.1332E-01   9.9177E-01   0.0000
   A(2)      1.1765E+00   7.4779E-02   1.0227E+00   1.3302E+00   0.0000
   k(1)      6.7935E+00   8.5386E-01   5.0383E+00   8.5486E+00   0.0000
   k(2)      1.1121E+00   5.1128E-02   1.0070E+00   1.2172E+00   0.0000
    AUC      1.1834E+00   1.4714E-02   1.1531E+00   1.2136E+00   0.0000
AUC is the area under the curve from t = 0 to t = infinity
Initial time point (A)    = 3.5983E-02
Final time point   (B)    = 1.6110E+00
Area from t = A to t = B = 9.3832E-01
Average over range (A,B) = 5.9575E-01
```

Using two padding zeros this leads to the following table.

For best-fit 2-exponential function

Parameter	Value	Std.Error	Lower95%cl	Upper95%cl	p
A(1)	0.852550	0.067731	0.713320	0.991770	0.0000
A(2)	1.176500	0.074779	1.022700	1.330200	0.0000
k(1)	6.793500	0.853860	5.038300	8.548600	0.0000
k(2)	1.112100	0.051128	1.007000	1.217200	0.0000
AUC	1.183400	0.014714	1.153100	1.213600	0.0000

AUC is the area under the curve from t = 0 to t = infinity
Initial time point (A) = 3.5983E-02
Final time point (B) = 1.6110E+00
Area from t = A to t = B = 9.3832E-01
Average over range (A,B) = 5.9575E-01

However, note that, with this example, three points emerge.

1. Numbers outside the main table may not be transformed into floating point numbers.
2. Equals signs lined up the trailer may not lead directly to secondary tabulation.
3. Some special words, like infinity, may not be recognized.

So, because a certain amount of fine tuning will be required, the possibilities for handcrafting are endless. Here, for example, the header is enlarged by adding a formula, while the trailer is added in the form of a footnote to the main table.

For the best-fit 2-exponential function

$f(t) = A_1 \exp(-k_1 t) + A_2 \exp(-k_2 t)$

Parameter	Value	Std.Error	Lower95%cl	Upper95%cl	p
A_1	0.852550	0.067731	0.713320	0.991770	0.0000
A_2	1.176500	0.074779	1.022700	1.330200	0.0000
k_1	6.793500	0.853860	5.038300	8.548600	0.0000
k_2	1.112100	0.051128	1.007000	1.217200	0.0000
AUC	1.183400	0.014714	1.153100	1.213600	0.0000

Area under the curve $AUC = \int_0^\infty \hat{f}(t)\, dt$

Initial time point (A) = 3.05983

Final time point (B) = 1.61100

Area from t = A to t = B = 0.93832

Average over range (A,B) = 0.59575

Example 3

A special situation exists with symmetric matrices where just a lower or upper triangle is displayed, and also some other related situations. For instance, following on from the previous example we have the parameter correlation matrix expressed in the following form.

```
Parameter correlation matrix
 1.0000
-0.8758  1.0000
-0.5964  0.8996  1.0000
-0.8480  0.9485  0.8200  1.0000
```

Clearly, the algorithm to count the number of columns per row in order to insert hashtags will fail because all the rows have different numbers of columns, unless editing is performed like this.

```
Parameter correlation matrix
 1.0000  ...     ...     ...
-0.8758  1.0000  ...     ...
-0.5964  0.8996  1.0000  ...
-0.8480  0.9485  0.8200  1.0000
```

Now transformation would be possible leading to a table such as the following.

Parameter correlation matrix			
1			
-0.8758	1		
-0.5964	0.8996	1	
-0.8480	0.9485	0.8200	1

Another example to consider is from correlation analysis which leads to r values in the strict upper triangle and significance levels in the strict lower triangle as shown next followed by the extracted table.

```
Pearson correlation results
Upper triangle = r, Lower = corresponding two-tail p values
 .....  0.5295  0.2874  0.0662  0.1941  0.6255 -0.5876  0.3010
0.0766   .....  0.3285 -0.0219  0.7930  0.5338 -0.4230  0.3006
0.3650  0.2971  .....  -0.2833  0.2165  0.0264  0.2314 -0.0304
0.8381  1.0000  1.0000   .....  0.2787  0.2837 -0.5238 -0.1166
0.5455  0.0021  0.4992  0.3804   .....  0.2029 -0.1949  0.2144
0.0296  0.0738  0.9351  1.0000  0.5271   ..... -0.4532  0.1360
1.0000  1.0000  0.4694  1.0000  1.0000  1.0000   ..... -0.1696
0.3418  0.3424  1.0000  1.0000  0.5035  0.6735  1.0000   .....
Test for absence of any significant correlations
H0: correlation matrix is the identity matrix
Determinant         =  2.476E-03
Test statistic (TS) =  4.501E+01
Degrees of freedom  =      28
P(chi-sq >= TS)     =  0.0220 Reject H0 at 5% sig.level
```

Pearson correlation results

Upper triangle = r, Lower = corresponding two-tail p values

.....	0.5295	0.2874	0.0662	0.1941	0.6255	-0.5876	0.3010
0.0766	.....	0.3285	-0.0219	0.7930	0.5338	-0.4230	0.3006
0.3650	0.2971	.....	-0.2833	0.2165	0.0264	0.2314	-0.0304
0.8381	1.0000	1.0000	.....	0.2787	-0.2837	-0.5238	-0.1166
0.5455	0.0021	0.4992	0.3804	.....	0.2029	-0.1949	0.2144
0.0296	0.0738	0.9351	1.0000	0.5271	.....	-0.4532	0.1360
1.0000	1.0000	0.4694	1.0000	1.0000	1.0000	.....	-0.1696
0.3418	0.3424	1.0000	1.0000	0.5035	0.6735	1.0000	.....

Test : for absence of any significant correlations

H_0 : correlation matrix is the identity matrix

Determinant = 0.002476

Test statistic(TS) = 45.01

Degrees of freedom = 28

$P(\chi^2 \geq TS) = 0.0220$ Reject H_0 at 5% significance level

Here the five dots (.....) denote that the diagonal elements have no meaning and this is just a convenient way to conserve space by having one matrix instead of two. Note that the option to blank out three dots (...) used as temporary column separators does not blank out groups with less three or more than three contiguous dots.

A summary of the options available and procedure to be used comes next.

Summary

- Up to Version 8 the default notation for numbers in results files was scientific notation, but from Version 8 users can decide whether to use standard notation or scientific notation and can also select the number of significant digits required. The default is six digits in standard notation.
- The first step is to select just one table from the results file.
- This can be followed by an optional pre-processing step to edit the table so that every row has exactly the same number of columns.
- Empty cells must be denoted by three dots (...) and cells containing words must have them joined by underscores or similar.
- There is an option to remove all three dot symbols from the output file.
- The input table can have optional header and trailer sections if required, but these must be highlighted by the control to select headers and trailers.
- There is an option to transform scientific numbers into floating point format by specifying the number of padding zeros required. Setting this parameter to -1 switches off this transformation.
- Numbers in the header and trailer may not be transformed in this way.
- If it is required, decimal points in floating point numbers can be replaced by commas.
- If transformation fails then the option for pre-processing should be switched on, and also the hashtag table should be requested. By viewing the hashtag table most errors can easily be diagnosed, then rectified by a re-run using pre-process editing.
- If tabbed-text output is selected the resulting file will have to be input into a spreadsheet program for formatting before importing into a word processing program.
- Both html and xml output can allow a certain number of further changes, like changing alpha into α, or adding cell borders.
- LaTeX output will have question marks (?) inserted to replace forbidden character such as underscores which must be edited retrospectively depending the intention, e.g., linking words, or denoting subscripts.

Three further things should be emphasized.

1. Some tables have specialized features such as lined up equals signs that allow multiple words in a column and, as long as every row in the table has an equals sign in exactly the same position, this feature will be recognized.
2. Some SimFIT results files output tables to the display without three dot separators (...) to create a more pleasing effect, but add them to the results files to assist the processing described in this document.
3. There are several widely used tables that can have empty cells and multi-word titles that the parsing routine will recognize and format automatically.

Finally, should you require further worked examples, you can browse the SimFIT tutorials, or the document `w_examples.pdf`, where a large number of alternative display styles are demonstrated.

3.3 Extracting tables to include in documents

Each time a data set is analyzed the results are written to a file called `f$result.txt`, and the current files are renamed so that the existing `f$result.txt` becomes `f$result.001` while `f$result.001` becomes `f$result.002` and so on. These can be viewed using the [Results] option from the main SimFIT menu.

These SimFIT results files are formatted so that the numbers displayed only contain the number of significant digits that are meaningful in context. For instance, probabilities will usually only have four digits after the decimal point, which indicates that it does not make any sense to consider any subsequent digits for purpose of statistical testing, and in any case probability estimates will not be accurate for more than about four digits. Furthermore, as experimental data are rarely more accurate than about three or four significant digits anyway, it may be wishful thinking to ever consider more than say six. In addition, the tables are formatted using a fixed font with scientific notation to line up column entities irrespective of absolute size, and many users do not want this in a thesis or published document. Naturally, these arguments do not apply to integers.

As an example consider the following case with the title Table 1.

```
Table 1
1-Way Analysis of Variance:  1 (Grand Mean  4.316E+01)

Transformation:-  x (untransformed data)
Source                SSQ       NDOF      MSQ          F          p
Between Groups    2.19344E+03      4  5.48316E+02  5.61546E+01  0.0000
Residual          2.44130E+02     25  9.76520E+00  ...         ...
Total             2.43757E+03     29  ...          ...         ...
```

Actually, most users would want to import such a table formatted as tabbed-text, html, xml, or LaTeX into documents such as a report, thesis, or publication looking something like Table 2.

Table 2
1-Way Analysis of Variance: 1 (Grand Mean 43.16)
Transformation:- x (untransformed data)

Source	SSQ	NDOF	MSQ	F	p
Between Groups	2193.44	4	548.316	56.1546	0.0000
Residual	244.130	25	9.76520		
Total	2437.57	29			

Or even, for those with artistic leanings, possibly something like Table 3.

Table 3
1-Way Analysis of Variance: Grand Mean 43.16
Transformation: x (untransformed data)

Source	SSQ	NDOF	MSQ	F	p
Between Groups	2193.44	4	548.316	56.1546	0.0000
Residual	244.130	25	9.76520		
Total	2437.57	29			

This article explains the procedures required to export tables from from SIMFIT results files into forms suitable for inclusion into word processors, spreadsheet programs, website scripts, or even professional document preparation systems such as LaTeX. Also, decimal points can be replaced by commas as in continental notation if required.

The procedure

It is important to realize that the [Results] option from the SIMFIT main menu gives access to all the currently saved results files.

- **Choosing a results file.**

  Early versions of SIMFIT allowed users to name results files individually to avoid anything being lost. However, now that up to 100 results files are saved and users have the option [Results] from the main SIMFIT menus from which to view, print, save, edit, or export tables, this is no longer usually necessary. Clearly, if results are always required for retrospective use, regular back-up or saving will be necessary.

- **Extracting a table.**

  From the [Extract tables] option view the file to make sure it is the one required then copy to the clipboard only the table required along with any associated header and trailer sections ... but nothing else.

- **Preliminary editing.**

  Sometimes editing of the file is required to make sure that every row of the table has exactly the same number of columns. So note that, for extracting a table there can be no empty cells, and each cell must contain precisely one word. Any column titles must be edited so that they consist of one word, for instance changing `Time of Day` to `Time_of_Day`, or filling empty cells by three dots. Added underscores and sets of three contiguous dots can be removed when the final table is written to file. A pre-processing option is provided for editing before attempting to create a table.

  Note that often tables have cells with added comments relating to goodness of fit or results of statistical testing, and these no not need to be underscored. There are also special tables with only two columns containing several words in some cells, and Example 1 later will make this clear.

- **Viewing the hashtag table.**

  The algorithm attempts to identify cells in a table by inserting a hashtag between every column. If the algorithm succeeds there would be no need to view this hashtag table. However this option should be switched on until the process of the algorithm is understood, or if it fails and you need to see why.

  The hashtag table is clarified in Example 1.

- **Headers and trailers.**

  Frequently tables have header and trailer sections that are descriptive and not part of the table itself. As these can have strings of word and numbers that would confuse the algorithm checking that every row must have the same number of columns, they must be identified. This is done by using buttons on a window that allows the header and trailer lines to be highlighted. If this is not done the table creating algorithm will fail.

  Selecting headers and trailers is clarified in Example 2.

- **Saving the table.**

  For programs that produce Windows quality hardcopy the table should be saved as html or xml as these will import directly into word processors or spreadsheets. Tabbed-text is also available but is much less versatile than html or xml, and LaTeX is available for those up to it.

- **Fine tuning.**

  Inevitably there will often be the need for dealing with details. For instance, users will sometimes want to replace `alpha` by α or `chi-sqd` by χ^2 and this can be done for html, xml, and LaTeX output but not

for tabbed-text. On the other hand html, xml, and LaTeX have reserved letters and these must be dealt with retrospectively.

For instance, consider the transformation of the following expressions which can be done in html, xml, and LaTeX but not in tabbed text.

Original	Transformed
>=	$\geq$
=<	$\leq$
alpha	α
beta	β
delta	δ
gamma	γ
lambda	λ
infinity	∞
P(chi-sqd >= TS) = 0.2037	$P(\chi^2 \geq TS) = 0.2037$

Where ambiguity could arise in LaTeX using underscores or similar special characters they will be replaced by question marks. So LaTeX users should search for ? characters to replace for the final table.

- **Padding with zeros.**

  As the [Extract tables] option will never remove significant digits a problem arises if users wish to replace numbers in scientific notation by floating point representation. In order to extend the range over which this can be done, padding zeros can be introduced as illustrated in this next table.

Scientific notation	Padding	Floating point representation
1.234E-01	0	.1234
1.234E+00	0	1.234
1.234E+01	0	12.34
1.234E+02	0	123.4
1.234E-02	2	0.01234
1.234E-01	2	0.12340
1.234E+00	2	1.23400
1.234E+01	2	12.3400
1.234E+02	2	123.400
1.234E+03	2	1234.00
1.234E+04	2	12340.0
1.234E-04	4	0.0001234
...	...	...
1.234E+06	4	1234000.0

Evidently increasing the number of padding zeros increases the range over which transformation from scientific to floating point representation can be achieved, and the default is two which allows a wide range, but four ensures that a mixture of transformed and untransformed numbers will occupy the same width in the columns of a table.

However, there is here a problem because adding padding zeros could suggest all trailing zeros are meaningful. For instance, the number 1.2341213179 stored in the computer could be written as 1.234E+00 in SimFIT output because the analysis in question only justifies accuracy or meaning for up to four significant digits. However, 1.23400 could be mistaken for indicating the internal representation with six significant digits. So users may wish to suppress trailing zeros in such cases, noting that this could result in numbers with different widths in a column. In any case, the transformation from scientific notation to floating point can be switched off.

Sometimes, for instance with numerical analysis where more significant digits are justified than with data analysis, eight significant digits are output, and some procedures can optionally allow more. In addition special DLLs dedicated for particular routines can be supplied for this purpose.

Example 1

This example shows the transformation of a special type of SIMFIT table containing just two columns separated by equals signs (i.e., =) and containing cells with multiple words. As long as the equals signs are perfectly lined up and there is no header or trailer section, this type of table can always be transformed. Here is the table with no header or trailer sections as extracted using the [Results] then [Extract tables] options from the main SIMFIT menu using the results file `f$result.txt` following the exhaustive analysis of a vector process used to analyze data contained in the default test file `normal.tf1`.

```
Sample size                    =      50
Minimum value                  = -2.20820E+00
Maximum value                  =  1.61750E+00
Coefficient of skew            = -1.66905E-02
Coefficient of kurtosis        = -7.68395E-01
Lower Hinge  (25th percentile) = -8.55015E-01
Median value (50th percentile) = -9.73615E-02
Upper Hinge  (75th percentile) =  7.85965E-01
Sample mean                    = -2.57897E-02
Sample standard deviation      =  1.00553E+00
Coefficient of variation (CV%) =  > 100%
Standard error of the mean     =  1.42203E-01
Upper 2.5% t-value             =  2.00958E+00
Lower 95% con lim for mean     = -3.11558E-01
Upper 95% con lim for mean     =  2.59978E-01
Sample variance                =  1.01109E+00
Lower 95% con lim for variance =  7.05519E-01
Upper 95% con lim for variance =  1.57006E+00
Shapiro-Wilks W statistic      =  9.62693E-01
Significance level for W       =  0.1153
Conclusion                     = Tentatively accept normality
```

This is the corresponding intermediate hashtag table.

```
Sample size                    #      50
Minimum value                  # -2.20820E+00
Maximum value                  #  1.61750E+00
Coefficient of skew            # -1.66905E-02
Coefficient of kurtosis        # -7.68395E-01
Lower Hinge  (25th percentile) # -8.55015E-01
Median value (50th percentile) # -9.73615E-02
Upper Hinge  (75th percentile) #  7.85965E-01
Sample mean                    # -2.57897E-02
Sample standard deviation      #  1.00553E+00
Coefficient of variation (CV%) #  > 100%
Standard error of the mean     #  1.42203E-01
Upper 2.5% t-value             #  2.00958E+00
Lower 95% con lim for mean     # -3.11558E-01
Upper 95% con lim for mean     #  2.59978E-01
Sample variance                #  1.01109E+00
Lower 95% con lim for variance #  7.05519E-01
Upper 95% con lim for variance #  1.57006E+00
Shapiro-Wilks W statistic      #  9.62693E-01
Significance level for W       #  0.1153
Conclusion                     # Tentatively accept normality
```

The hashtag table is very useful for detecting the source of errors. The table making algorithm attempts to locate the position separating columns and writes a hashtag there. If every row has the same number of columns then every row will have the same number of hashtags and the algorithm has succeeded. Observing this hashtag table when the algorithm has failed will allow you identify then correct the error.

Here is the selected table as it would be written to the output file.

Sample size	50
Minimum value	-2.2082000
Maximum value	1.6175000
Coefficient of skew	-0.0166905
Coefficient of kurtosis	-0.7683950
Lower Hinge (25th percentile)	-0.8550150
Median value (50th percentile)	-0.0973615
Upper Hinge (75th percentile)	0.7859650
Sample mean	-0.0257897
Sample standard deviation	1.0055300
Coefficient of variation (CV%)	> 100%
Standard error of the mean	0.1422030
Upper 2.5% t-value	2.0095800
Lower 95% con lim for mean	-0.3115580
Upper 95% con lim for mean	0.2599780
Sample variance	1.0110900
Lower 95% con lim for variance	0.7055190
Upper 95% con lim for variance	1.5700600
Shapiro-Wilks W statistic	0.9626930
Significance level for W	0.1153
Conclusion	Tentatively accept normality

Here it is with a few minor cosmetic changes.

Exhaustive analysis of a vector

Sample size	50
Minimum value	-2.2082000
Maximum value	1.6175000
Coefficient of skew	-0.0166905
Coefficient of kurtosis	-0.7683950
Lower Hinge (25th percentile)	-0.8550150
Median value (50th percentile)	-0.0973615
Upper Hinge (75th percentile)	0.7859650
Sample mean	-0.0257897
Sample standard deviation	1.0055300
Coefficient of variation (CV%)	> 100%
Standard error of the mean	0.1422030
Upper 2.5% t-value	2.0095800
Lower 95% confidence limit for mean	-0.3115580
Upper 95% confidence limit for mean	0.2599780
Sample variance	1.0110900
Lower 95% confidence limit for variance	0.7055190
Upper 95% confidence limit for variance	1.5700600
Shapiro-Wilks W statistic	0.9626930
Significance level for W	0.1153
Conclusion: *Tentatively accept normality*	

Example 2

From fitting a two-exponential model to data in the test file `exfit.tf4` using program **exfit** the following results can be extracted from the results file.

```
For best-fit 2-exponential function

Parameter    Value       Std.Error   Lower95%cl   Upper95%cl     p
   A(1)    8.5255E-01  6.7731E-02  7.1332E-01  9.9177E-01  0.0000
   A(2)    1.1765E+00  7.4779E-02  1.0227E+00  1.3302E+00  0.0000
   k(1)    6.7935E+00  8.5386E-01  5.0383E+00  8.5486E+00  0.0000
   k(2)    1.1121E+00  5.1128E-02  1.0070E+00  1.2172E+00  0.0000
   AUC     1.1834E+00  1.4714E-02  1.1531E+00  1.2136E+00  0.0000
AUC is the area under the curve from t = 0 to t = infinity
Initial time point (A)    =  3.5983E-02
Final time point   (B)    =  1.6110E+00
Area over range (A,B)     =  9.3832E-01
Average over range (A,B) =  5.9575E-01
```

Now the file has an additional head and trailer section so, if the full table is selected, it will have to be highlighted as follows in the header and trailer selection control as shown next, where the header is colored magenta and the trailer colored cyan.

```
For best-fit 2-exponential function

Parameter        Value     Std.Error   Lower95%cl   Upper95%cl     p
   A(1)      8.5255E-01  6.7731E-02  7.1332E-01  9.9177E-01  0.0000
   A(2)      1.1765E+00  7.4779E-02  1.0227E+00  1.3302E+00  0.0000
   k(1)      6.7935E+00  8.5386E-01  5.0383E+00  8.5486E+00  0.0000
   k(2)      1.1121E+00  5.1128E-02  1.0070E+00  1.2172E+00  0.0000
    AUC      1.1834E+00  1.4714E-02  1.1531E+00  1.2136E+00  0.0000
AUC is the area under the curve from t = 0 to t = infinity
Initial time point (A)    = 3.5983E-02
Final time point (B)       = 1.6110E+00
Area over range (A,B)     = 9.3832E-01
Average over range (A,B) = 5.9575E-01
```

Using two padding zeros this leads to the following table.

For best-fit 2-exponential function

Parameter	Value	Std.Error	Lower95%cl	Upper95%cl	p
A(1)	0.852550	0.067731	0.713320	0.991770	0.0000
A(2)	1.176500	0.074779	1.022700	1.330200	0.0000
k(1)	6.793500	0.853860	5.038300	8.548600	0.0000
k(2)	1.112100	0.051128	1.007000	1.217200	0.0000
AUC	1.183400	0.014714	1.153100	1.213600	0.0000

AUC is the area under the curve from $t = 0$ to $t = \infty$
Initial time point (A) = 0.035983
Final time point (B) = 1.611
Area over range (A,B) = 0.93832
Average over range (A,B) = 0.59575

However, note that, with this example, three points emerge.

1. Numbers outside the main table will also be transformed into floating point numbers.
2. Equals signs lined up the trailer will not lead directly to secondary tabulation.
3. Some special words, like infinity, will be recognized.

So, because a certain amount of fine tuning will be required, the possibilities for handcrafting are endless. Here, for example, the header is enlarged by adding a formula, while the trailer is added in the form of a footnote to the main table.

For the best-fit 2-exponential function

$f(t) = A_1 \exp(-k_1 t) + A_2 \exp(-k_2 t)$

Parameter	Value	Std.Error	Lower95%cl	Upper95%cl	p
A_1	0.852550	0.067731	0.713320	0.991770	0.0000
A_2	1.176500	0.074779	1.022700	1.330200	0.0000
k_1	6.793500	0.853860	5.038300	8.548600	0.0000
k_2	1.112100	0.051128	1.007000	1.217200	0.0000
AUC	1.183400	0.014714	1.153100	1.213600	0.0000

Area under the curve $AUC = \int_0^\infty \hat{f}(t)\, dt$

Initial time point (A) = 0.035983

Final time point (B) = 1.611

Area over range (A,B) = 0.93832

Average over range (A,B) = 0.59575

Example 3

A special situation exists with symmetric matrices where just a lower or upper triangle is displayed, and also some other related situations. For instance, following on from the previous example we have the parameter correlation matrix expressed in the following form.

```
Parameter correlation matrix
 1.0000
-0.8758  1.0000
-0.5964  0.8996  1.0000
-0.8480  0.9485  0.8200  1.0000
```

Clearly, the algorithm to count the number of columns per row in order to insert hashtags will fail because all the rows have different numbers of columns, unless editing is performed like this.

```
Parameter correlation matrix
 1.0000  ...     ...     ...
-0.8758  1.0000  ...     ...
-0.5964  0.8996  1.0000  ...
-0.8480  0.9485  0.8200  1.0000
```

Now transformation would be possible leading to a table such as the following.

Parameter correlation matrix			
1			
-0.8758	1		
-0.5964	0.8996	1	
-0.8480	0.9485	0.8200	1

Another example to consider is from correlation analysis which leads to r values in the strict upper triangle and significance levels in the strict lower triangle as shown next followed by the extracted table.

```
Pearson correlation results
Upper triangle = r, Lower = corresponding two-tail p values
 .....  0.5295  0.2874  0.0662  0.1941  0.6255 -0.5876  0.3010
0.0766   .....  0.3285 -0.0219  0.7930  0.5338 -0.4230  0.3006
0.3650  0.2971  .....  -0.2833  0.2165  0.0264  0.2314 -0.0304
0.8381  1.0000  1.0000   .....  0.2787  0.2837 -0.5238 -0.1166
0.5455  0.0021  0.4992  0.3804   .....  0.2029 -0.1949  0.2144
0.0296  0.0738  0.9351  1.0000  0.5271   ..... -0.4532  0.1360
1.0000  1.0000  0.4694  1.0000  1.0000  1.0000   ..... -0.1696
0.3418  0.3424  1.0000  1.0000  0.5035  0.6735  1.0000   .....
Test for absence of any significant correlations
H0: correlation matrix is the identity matrix
Determinant         =  2.476E-03
Test statistic (TS) =  4.501E+01
Degrees of freedom  =      28
P(chi-sq >= TS)     =  0.0220 Reject H0 at 5% sig.level
```

Pearson correlation results

Upper triangle = r, Lower = corresponding two-tail p values

.....	0.5295	0.2874	0.0662	0.1941	0.6255	-0.5876	0.3010
0.0766	.....	0.3285	-0.0219	0.7930	0.5338	-0.4230	0.3006
0.3650	0.2971	.....	-0.2833	0.2165	0.0264	0.2314	-0.0304
0.8381	1.0000	1.0000	.....	0.2787	-0.2837	-0.5238	-0.1166
0.5455	0.0021	0.4992	0.3804	.....	0.2029	-0.1949	0.2144
0.0296	0.0738	0.9351	1.0000	0.5271	.....	-0.4532	0.1360
1.0000	1.0000	0.4694	1.0000	1.0000	1.0000	.....	-0.1696
0.3418	0.3424	1.0000	1.0000	0.5035	0.6735	1.0000	.....

Test : for absence of any significant correlations

H_0 : correlation matrix is the identity matrix

Determinant = 0.002476

Test statistic(TS) = 45.01

Degrees of freedom = 28

$P(\chi^2 \geq TS) = 0.0220$ Reject H_0 at 5% significance level

Here the five dots (.....) denote that the diagonal elements have no meaning and this is just a convenient way to conserve space by having one matrix instead of two. Note that the option to blank out three dots (...) used as temporary column separators does not blank out groups with less three or more than three contiguous dots.

A summary of the options available and procedure to be used comes next.

Summary

- Up to Version 8 the default notation for numbers in results files was scientific notation, but from Version 8 users can decide whether to use standard notation or scientific notation and can also select the number of significant digits required. The default is six digits in standard notation.
- The first step is to select just one table from the results file.
- This can be followed by an optional pre-processing step to edit the table so that every row has exactly the same number of columns.
- Empty cells must be denoted by a three dot ellipsis (...) and cells containing multiple words must have them joined by underscores or similar.
- Three dot symbols to denote empty cells are deleted from the output file.
- The input table can have optional header and trailer sections if required, but these must be highlighted by the control to select headers and trailers.
- There is an option to transform scientific numbers into floating point format by specifying the number of padding zeros required. This option can be switched off.
- Numbers in the header and trailer will also be transformed in this way.
- If it is required, decimal points in floating point numbers can be replaced by commas.
- If transformation fails then the option for pre-processing should be switched on, and also the hashtag table should be requested. By viewing the hashtag table most errors can easily be diagnosed, then rectified by a re-run using pre-process editing.
- If tabbed-text output is selected the resulting file will have to be input into a spreadsheet program for formatting before importing into a word processing program.
- Both html and xml output can allow a certain number of further changes, like changing alpha into α, or adding cell borders.
- LaTeX output will have question marks (?) inserted to replace forbidden character such as underscores which must be edited retrospectively depending the intention, e.g., linking words, or denoting subscripts.

Three further things should be emphasized.

1. Some tables have specialized features such as lined up equals signs that allow multiple words in a column and, as long as every row in the table has an equals sign in exactly the same position, this feature will be recognized.
2. Some SimFIT results files output tables to the display without three dot separators (...) to create a more pleasing effect, but add them to the results files to assist the processing described in this document.
3. There are several widely used tables that can have empty cells and multi-word titles that the parsing routine will recognize and format automatically.

Finally, should you require further worked examples, you can browse the SimFIT tutorials, or the document `w_examples.pdf`, where a large number of alternative display styles are demonstrated.

4 Statistical analysis

4.1 Statistical distributions

Tutorials and worked examples for simulation, curve fitting, statistical analysis, and plotting.
https://simfit.uk
https://simfit.org.uk
https://simfit.silverfrost.com

4.1.1 Introduction

Data analysis will usually consist of assuming a statistical distribution and comparing a sample, or a test statistic derived from it, to possible values from the assumed distribution. If the test statistic proves to have a rather extreme value when referred to the assumed distribution it may be taken to suggest that the assumed distribution may not be correct. So statistical testing will often consist of a null hypothesis, denoted as H_0, and there may be an alternative hypothesis or several alternative hypotheses, say H_A.

The situation can be summarized by the following sequence.

1. Collect data.
   An example could be a sample of sizes, times, weights, distances, etc.

2. Calculate a test statistic.
   An example could be calculating the sample mean or standard distribution.

3. Assume a theoretical null distribution, denoted by H_0.
   An example H_0 could be assuming a normal distribution with mean of 6 and standard deviation of 4.

4. Assume a possible alternative distribution, denoted by H_A.
   For instance H_A might be a normal distribution with a mean of 7 and a standard deviation of 4.

5. Check if the test statistics would be extreme if coming from the assumed distribution.
   For instance, to do this we could see if the sample estimates for mean and standard deviation are more consistent with H_0 rather than H_A. This would lead to one of two possible courses of action.

   - Consider the possibility that H_0 is likely to be correct.
   - If no satisfactory conclusion can be reached then accumulate more data or assume a new distribution, or the same distribution with different parameters.

Obviously, if the assumed distribution is incorrect, any conclusions drawn from this procedure will be of questionable value. Now almost no scientific experiment ever leads to data that follows a known distribution exactly, so what happens in practice is that a number of standard distributions are chosen in the hope that one of these will be sufficiently close to the distribution of the test statistic, or that the data can be transformed into an alternative form that is closer to an assumed distribution.

Actuality, only a limited number of distributions, such as the following, are encountered in data analysis.

a) Normal

b) t

c) chi-square

d) F

e) Binomial

f) Poisson

g) Uniform

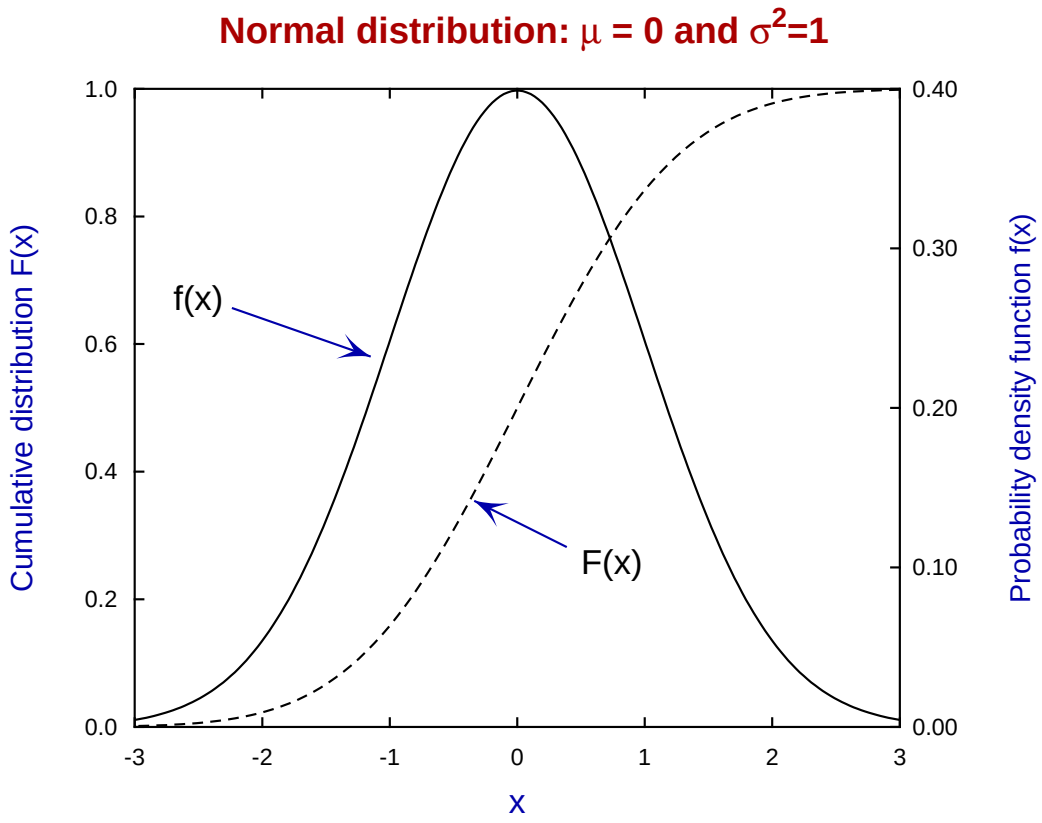
Even so-called nonparametric tests often finish up by relying on some standard distribution, and frequent use is made in statistical theory of the Gauss central limit theorem. This shows that sums of suitably normalized values will tend, in the limit of large sample size n, to a normal distribution. However n may often be very large before such convergence is achieved. Because of all this uncertainty it is often stated that statistical analysis can prove nothing, or alternatively anything. Nevertheless this is all we have so it is useful to sum up some unifying concepts that will be assumed in subsequent SIMFIT tutorials.

Continuous variables

A continuous random variable X is a number that can take all values in a range, say $-\infty \leq X \leq \infty$ but is subject to certain constraints. Typical continuous variables would be time, size, blood pressure, etc., which like so many measured variables happen to be necessarily non-negative. In particular, there will be a non-negative probability distribution function $f(x) \geq 0$ and a cumulative distribution function $F(x)$ such that that the probability that X has a particular value x in the range $A \leq X \leq B$ will be

$$P(A \leq X \leq B) = \int_A^B f(t)dt$$
$$= F(B) - F(A).$$

Here, for example are $f(x)$ and $F(x)$ for a normal distribution with mean zero and variance one.



Evidently values of X less than -3 or greater than 3 would be very unlikely for this distribution and could indicate a mean differing from zero and/or a variance differing from 1. A statistical test using the sample mean and sample variance could be constructed by such reasoning.

Note also that, because the variable is continuous, it makes no sense to assign a probability of the random variable having a definite value, but only the probability of it taking a value in an interval $A \leq X \leq B$. However, the integration of $f(x)$ over the possible range, say $-\infty \leq X \leq \infty$ would be one, i.e.

$$\int_{-\infty}^{\infty} f(t)dt = 1.$$

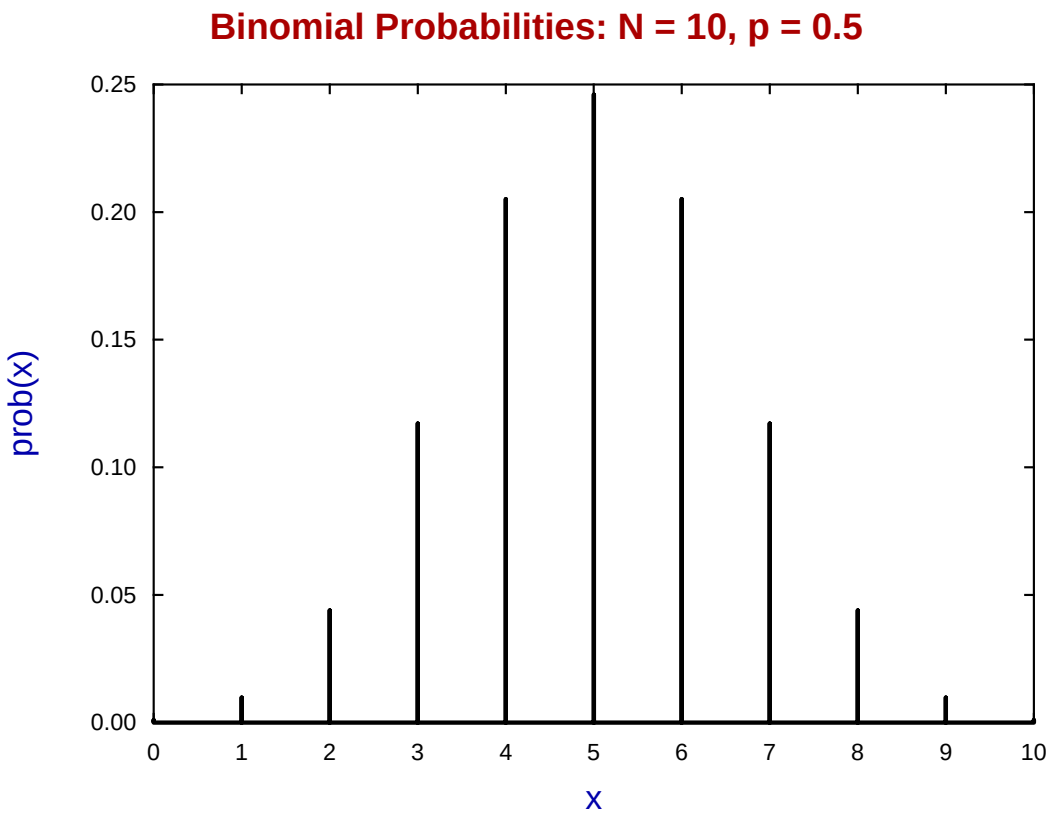
Discrete variables

A discrete random variable X is an integer that can only take a limited number of values. Examples would be the number of heads resulting from a fixed number of coin tossings, or the number of eggs hatching as males from a clutch of eggs.

In particular, there will be a non-negative probability mass function $p(x) \geq 0$ which would describe the probability of X having a particular integer value, that is $P(X = k) = p(k)$. Obviously, if there are n possible values that X can have, say $k_1, k_2, \ldots k_n$ then

$$\sum_{i=1}^{n} p(k_i) = 1.$$

Here, for example is the plot of probabilities for a binomial distribution with $N = 10$ and $p = 0.5$ such as would result, for instance, by adding up the number of times a head would occur in ten throws of a coin.



Evidently numbers of heads of 0, 1, 9, or 10 would be very unlikely for this distribution and could be taken to indicate a biased coin. A statistical test using the sample mean and sample variance could be constructed by such reasoning.

4.1.2 Uniform distribution

Given two numbers A and B with $B > A$, then a random variable that can take all values in the interval between A and B with equal probability is referred to as having a uniform distribution, $U(A, B)$, or alternatively a rectangular distribution. This distribution is of immense value in simulation studies as will be explained subsequently. Two frequently encountered special cases are when only integer values are allowed, and also when $A = 0$ and $B = 1$.

Definitions

A random variable Y distributed as $U(A, B)$ has probability density function $g(y)$, cumulative distribution function $G(y)$, expectation $E(Y)$ and variance $V(Y)$ given by

$$\begin{aligned} g(y) &= \frac{1}{B-A} \\ G(y) &= \frac{y-A}{B-A} \\ E(Y) &= \frac{A+B}{2} \\ V(Y) &= \frac{(A+B)^2}{12}. \end{aligned}$$

It is interesting to note two important facts used by SimFIT concerning any arbitrary continuous random variable X with distribution function $F(x)$, and a random variable Y which follows a continuous uniform distribution on (0,1), say with distribution $G(y)$, i.e. with $A = 0$ and $B = 1$, so that $G(y) = y$.

1. If $U(0, 1)$ random numbers $y_1, y_2, \ldots, y_n$ are available, then random numbers $x_1, x_2, \ldots, x_n$ with distribution function $F(x)$ can be generated from them using

   $$x_i = F^{-1}(y_i).$$

   This can be appreciated from a graph of the cumulative distribution $F(x)$ as a function of x but taking as vertical axis $Y = F(x)$ so that

   $$P(X \le x) = P(Y \le F(x)) = G(F(x)) = F(x).$$

2. Conversely, given random numbers $x_1, x_2, \ldots, x_n$, then uniformly distributed random numbers $y_1, y_2, \ldots, y_n$ can be generated from them using

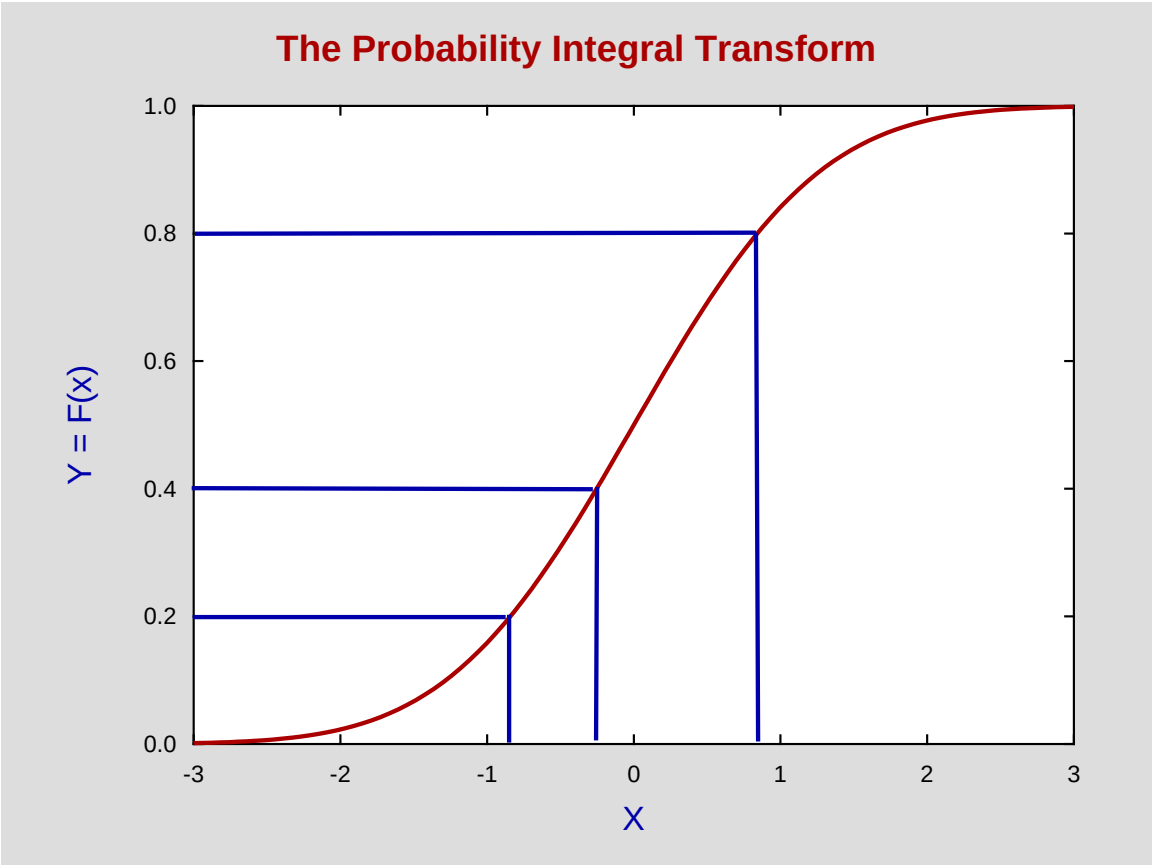
   $$y_i = F(x_i).$$

   This follows since

   $$P(Y \le y) = P(X \le F^{-1}(y)) = F(F^{-1}(y)) = y = G(y).$$

The Probability Integral transform

Not surprisingly, there are technical details to consider before accepting the previous results, known as the probability integral transform. However, the following diagram and table illustrating the uniform distribution on 0,1 with distribution function $G(y)$ and the standard normal distribution with distribution function $F(x)$ should make it clear.



y	$G(y)$	x	$F(x)$
0.2	0.2	-0.8416	0.2
0.4	0.4	-0.2533	0.4
0.8	0.8	0.8416	0.8

The point is that equally spaced divisions on the Y axis correspond to unequal divisions on the X axis, but the probabilities in the intervals are identical.

In other words, in terms of the inverse standard normal distribution function,

$$F^{-1}(0.2) = -0.8416$$
$$F^{-1}(0.4) = -0.2533$$
$$F^{-1}(0.8) = 0.8416,$$

so that

$$P(-0.8416 \leq X \leq -0.2533) = P(0.2 \leq Y \leq 0.4) = 0.2.$$

Pseudo random numbers

Computers cannot generate true random numbers, but they can generate extremely long deterministic sequences of numbers that do have properties closely similar to random numbers. As all such schemes are cyclic, the starting point in the sequence can be determined arbitrarily, usually by using the system clock, or from a fixed starting point using a seed or array of seeds. Generation of such pseudo random numbers begins by obtaining a sample of n such $U(0,1)$ numbers that are then transformed into numbers from a selected distribution. As evaluation of $F(.)$ and $F^{-1}(.)$ for standard distributions requires numerical methods, SimFIT does not use the scheme $x = F^{-1}(y)$ outlined in 1. above, as there are more convenient techniques. However SimFIT does use the scheme $y = F(x)$ outlined in 2. above to transform numbers into $U(0,1)$ numbers, because this is valuable when testing if numbers do arise from an assumed distribution, and it is particularly useful when visually inspecting values generated in experiments if these can be transformed so as to be collected into bins with equal probability, as will be illustrated.

Unfortunately, pseudo random numbers do have appreciable autocorrelation and other deficiencies, particularly if long sequences are required for simulation, and much ingenuity has been expended to surmount such obstacles. Accordingly, methods to test the performance of particular random number generators have been developed, and SimFIT provides the option to test the random number generator provided. This is a Marsaglia-Zaman type using subtract-with-borrow and has a cycle length of 2^{1376}, compared to the value of 2^{57} available with some standard linear congruential generators.

Testing the Simfit U(0,1) generator

Choose [A/Z] from the main SimFIT menu and open program **rannum** when the following options will be available.

```
Generate sequences of random numbers
Generate and plot random walks
Generate random matrices
Generate random permutations
Test the current U(0,1) generator
Set the seed type
```

After selecting the option to test the current $U(0,1)$ generator these further options become available.

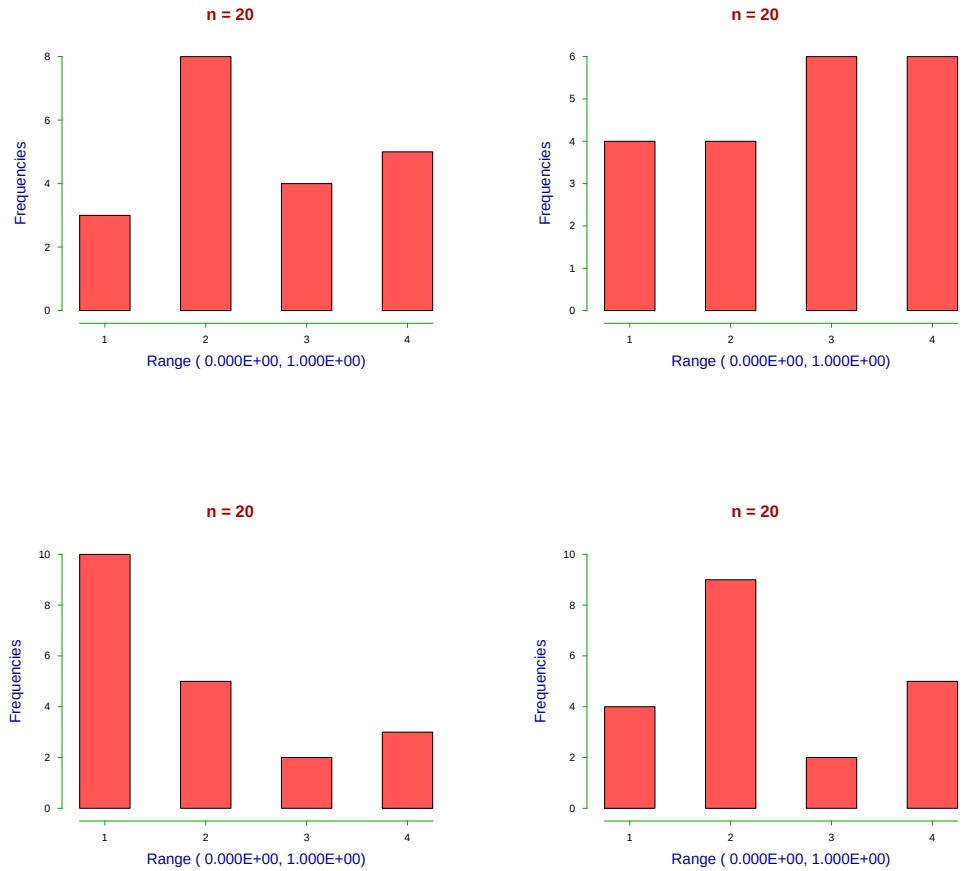
```
Runs up (or down) test
Bar chart plot
Chi-square test
Kolmogorov-Smirnov test
```

The runs up (or down) test requires a very large sample and tests for significant autocorrelations, the bar chart plot simply displays a histogram, the chi-square test measures departure of the histogram from a $U(0,1)$ distribution, while the Kolmogorov-Smirnov test examines the maximum deviation of the sample cumulative distribution from the expected straight line.

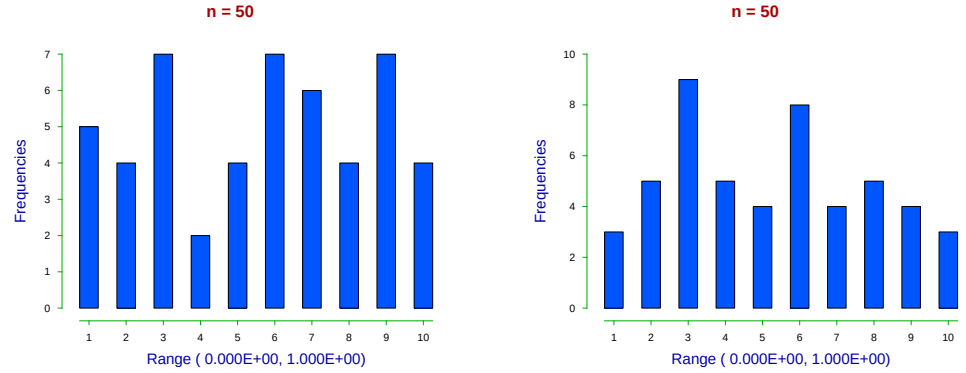
It is often advised that a minimum sample size of $n = 20$ is required to test if a sample is consistent with an assumed distribution and, although statistical tests like the above can be employed, decisions are more often made by visual inspection of a histogram. Now in the limit of very large samples with many bins histograms do converge in shape to the population distribution. However the next examples are intended to demonstrate that, in reality, sample sizes much greater than $n = 20$ are required to carry conviction. The $U(0,1)$ distribution is particularly suited for this purpose as the histogram should have every bin frequency of approximately the same size, since the probability density function is a horizontal line.

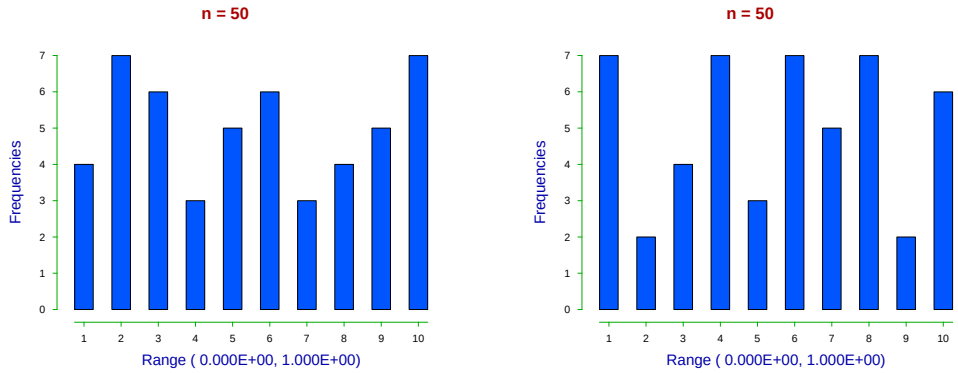
The following histograms display four consecutive simulations using program **rannum**, and note that the usual advice is to have an expected value of at least 5, and preferably an observed value of the same order, for each

bin. Of course, a major failing of analysis based on histograms is that the visual appearance and results from statistical analysis depend on the number of bins chosen.

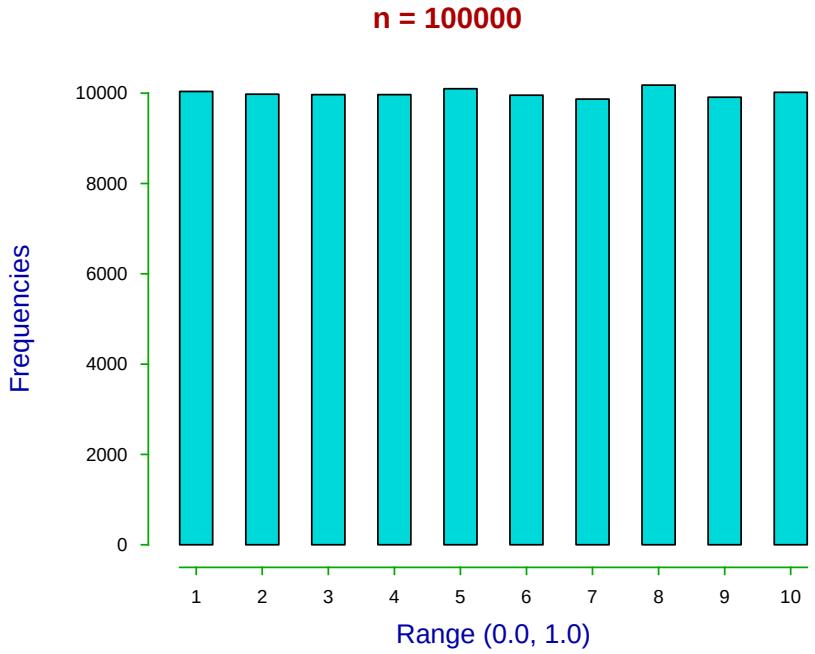


It will be clear from these results that a sample size of $n = 20$ is insufficient and could easily lead to false conclusions, as the histogram can suggest almost any shape for the population distribution. Increasing the sample size to $n = 50$ can still appear to be rather low as will be clear from the next four successive simulations.





Actually numerical results concerning the sample size required can be obtained from the SIMFIT section on power as a function of sample size. Meanwhile here is the sort of convincing result obtained with large samples, in the next case $n = 10000$.



4.1.3 Normal distribution

The normal distribution has great importance in data analysis because, although experimental measurements never follow normal distributions exactly, many observations are approximately normally distributed, or become so after transformations such as replacing observations by the logarithms. For instance experimental error is often approximately normally distributed

Definitions

A random variable X is said to normally distributed if the probability density function (pdf) $f(x)$ and cumulative distribution function (cdf) $F(x)$ are

$$f(x) = \frac{1}{\sqrt{2\pi}\,\sigma} \exp -\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2$$
$$F(x) = \int_{-\infty}^{x} f(t)\,dt$$

so that the probability of a value occurring in the range A, B with $B > A$ is

$$P(A \leq X \leq B) = F(B) - F(A).$$

Here μ is the mean, the standard deviation is σ, and the variance is σ^2. Because μ can have any value at all and σ can have any positive value it is useful to consider the standardized variable Z defined as

$$Z = \frac{X-\mu}{\sigma}$$

which is normally distributed with mean zero and variance one.

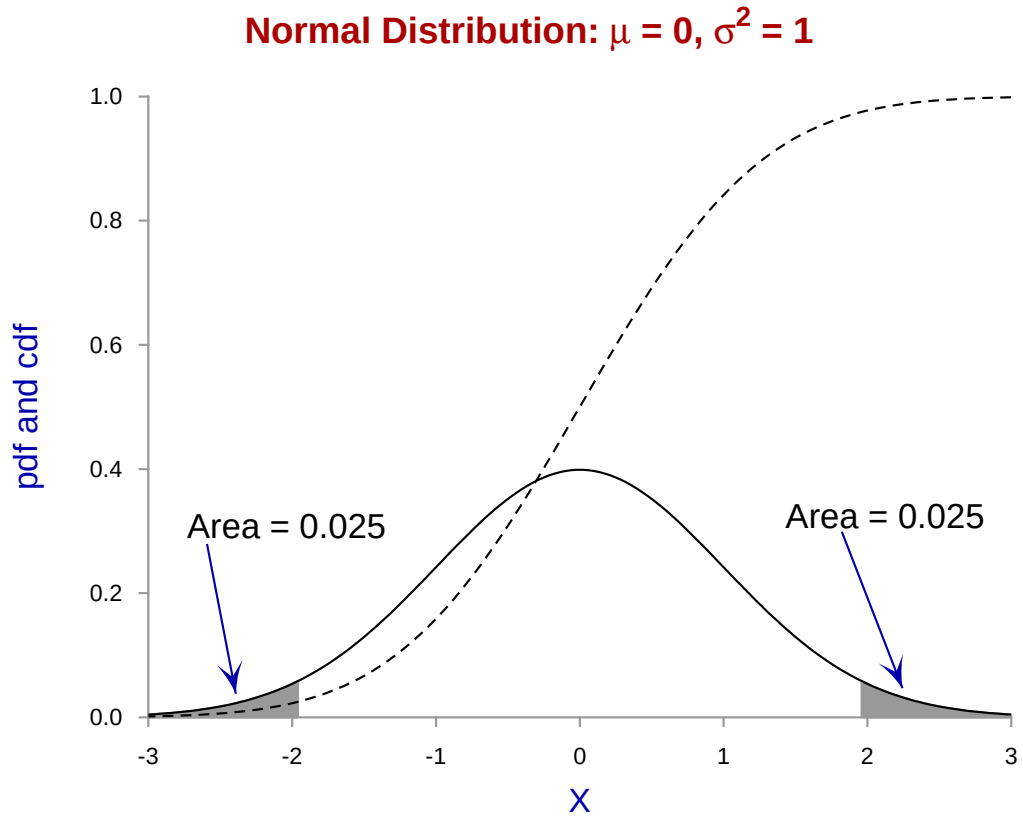
Simfit program normal

In order to understand this distribution by plotting profiles and calculating deviates we shall consider some of the procedures available using the SimFIT program **normal**. To do this, select menu item `[A/Z]` from the main SimFIT menu, and open program **normal** when the following options will be available

```
Input: mu and sigma
Input: x, calculate pdf(x)
Input: x, calculate cdf(x)
Input: alpha, calculate x
```

as well as options to test if data are normally distributed, to perform power and sample size calculations, or to investigate the multivariate normal distribution.

If the default values for μ and σ are accepted then the following plot can be obtained.



This illustrates that the normal distribution pdf is a symmetrical bell-shaped curve with tails that rapidly decrease after some two standard deviations. The cdf on the other hand is a monotonic sigmoidal curve rising from a minimum value of zero to a maximum value of one. It is in fact the integral of the pdf, that is, the value of $F(x)$ at the value x is simply the area under the pdf curve from $-\infty$ to x.

Particular interest attaches to the area in the lower and upper tails of this distribution. In fact, the tails illustrated in this figure are the lower and upper 2.5% points. In other words, the probability of a value occurring in the lower tail is 0.025, the probability of a value occurring in the upper tail is 0.025, so that the probability of a value occurring in either the lower or upper tail is obviously 0.05. Perhaps the best known 2-tail critical points are the 68% and 95% ones, i.e.

$$P(\mu - \sigma \le X \le \mu + \sigma) = 0.68, \quad \text{and} \quad P(\mu - 1.96\sigma \le X \le \mu + 1.96\sigma) = 0.95.$$

Out of interest, it should be pointed out that the tails were shaded in this figure by using the advanced option to transfer the data into the SimFIT program **simplot** followed by assigning the first two lines to be closed polygons which were then colored grey.

Before the widespread availability of computers, values of such critical points were read off from tables, and also the inverses were obtained in this way, that is the values of X calculated from specific values of $F(x)$. We now explain how to do this using SimFIT program **normal**

Obtaining critical values

SimFIT program **normal** was used to obtain three values of $-1, 0, 1$ for the pdf, the same three for the cdf, and three critical values for 2.5%, 5.0% and 50.0% as in this table from the results log file, which was archived by SimFIT when program **normal** was closed.

```
pdf values
Current parameters: mu = 0.0E+00, sigma = 1.0E+00, sigma^2 = 1.0E+00
pdf(-1.000E+00) = 2.420E-01
pdf( 0.000E+00) = 3.989E-01
pdf( 1.000E+00) = 2.420E-01

cdf values
Current parameters: mu = 0.0E+00, sigma = 1.0E+00, sigma^2 = 1.0E+00
P(X =< -1.000E+00) = 0.1587 ... P(X >= -1.000E+00) = 0.8413
P(X =<  0.000E+00) = 0.5000 ... P(X >=  0.000E+00) = 0.5000
P(X =<  1.000E+00) = 0.8413 ... P(X >=  1.000E+00) = 0.1587

critical points
Current parameters: mu = 0.0E+00, sigma = 1.0E+00, sigma^2 = 1.0E+00
P(X =<  1.960E+00) = 0.9750 *** P(X >=  1.960E+00) = 0.0250
P(X =<  1.645E+00) = 0.9500 *** P(X >=  1.645E+00) = 0.0500
P(X =<  0.000E+00) = 0.5000 *** P(X >=  0.000E+00) = 0.5000
```

The pdf values illustrate in numbers what is displayed in the graph, that $f(-1) = f(1)$ because of the fact that values equally spaced below and above the mean give the same pdf values due to the symmetry.

The cdf values also illustrate that the areas in the lower and upper tails at values equally spaced below and above the means are equal, and clearly the areas below and above the mean are 0.5.

The critical points illustrated show the same symmetry, but it should be emphasized that using lower and upper critical points for statistical testing would normally require critical points based on the sum of lower and upper tail probabilities, as in a two-tail test. For instance, if a statistical test is conducted to see if an observation is consistent with a certain mean, then the two-tail test would allow for the observation being either extremely low or extremely large. If the analyst was just not prepared to consider such an outcome but would only countenance the possibility of an observation being too large for the null hypothesis, or too small as the case may be, would a one-tail test based on only one of the tails be used.

A trick often resorted to when submitting grant proposals is to do power and sample size calculations using one-tail tests when two-tailed would be more honest. Like claiming a lower variance than is justified experimentally to reduce the sample size required, statistics is always open to such abuse.

4.1.4 t distribution

The great importance of the t distribution in data analysis lies in the existence of numerous tests based upon it, such as the 1-sample t, unpaired t, and paired t, as well as the use in calculating confidence intervals.

Definitions

Consider two independent random variables, Z which has a normal distribution with $\mu = 0$, $\sigma^2 = 1$, and C which has a chi-square distribution with k degrees of freedom. Then the ratio

$$t_k = \frac{Z}{\sqrt{C/k}}$$

is described as a t variable with k degrees of freedom. It should be noted incidentally that t_k^2 is distributed as $F(1, k)$.

A special case arises when analyzing a sample of size n from a normal distribution with population mean μ and population variance σ^2, because the sample mean

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

is normally distributed with mean μ and variance σ^2/n, while nS^2/σ^2 using

$$S^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

has a χ^2 distribution with $n - 1$ degrees of freedom. Hence the statistic

$$t_{n-1} = \frac{\bar{x} - \mu}{S/\sqrt{n-1}}$$

has a t distribution with $n - 1$ degrees of freedom. Note that this t variable only has one unknown parameter, the population mean μ.

Simfit program ttest

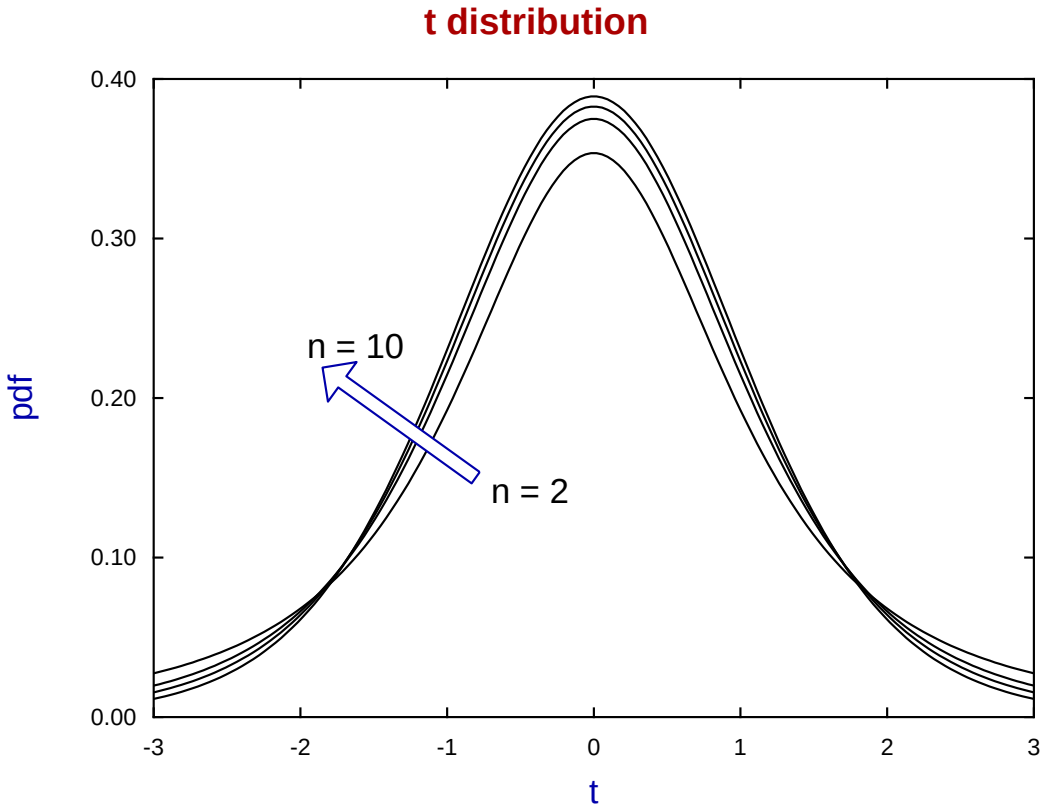
Choose [A/Z] from the main SimFIT menu and open program **ttest** when the following options will be available.

```
Input: N, number of degrees of freedom
Input: t, calculate pdf(t)
Input: t, calculate cdf(t)
Input: alpha, calculate t inverse
Input: data, 1-sample t test
Input: data, 2-sample unpaired t test
Input: data, 2-sample paired t test
Input: matrix, groups across rows t test
Power and sample size
Non-central t distribution.
```

Degrees of freedom

An important use of the t distribution is when calculating confidence limits, for instance with a sample mean, or parameter estimate. The main thing to realize in such circumstances is that, although the mean value for t_n

is zero irrespective of n, the variance is heavily dependent on n. This is why the confidence limits shrink as the sample size increases. Actually the t_n distribution is asymptotic to a standardized normal distribution as n increases, as shown by the next graph created from **ttest**.



Note how the area under the tails decreases rapidly as n increases from 2 to 6 but less slowly thereafter. A more detailed inspection of this will be clear from this table copied from the **ttest** results log file for a 95% confidence interval.

```
P(t =< 4.303E+00) = 0.975 *** P(t >= 4.303E+00) = 0.025, N = 2
P(t =< 2.776E+00) = 0.975 *** P(t >= 2.776E+00) = 0.025, N = 4
P(t =< 2.447E+00) = 0.975 *** P(t >= 2.447E+00) = 0.025, N = 6
P(t =< 2.306E+00) = 0.975 *** P(t >= 2.306E+00) = 0.025, N = 8
P(t =< 2.228E+00) = 0.975 *** P(t >= 2.228E+00) = 0.025, N = 10
```

Confidence range for the sample mean

Given $\bar{x}$ and S^2 from a sample of size n, then a symmetrical $100(1-\alpha)\%$ confidence range for the population mean μ can be constructed using the upper tail critical value $t_{\alpha/2,n-1}$. We have that

$$P\left(\frac{\bar{x}-\mu}{S/\sqrt{n-1}} \geq t_{\alpha/2,n-1}\right) = \alpha/2$$

and

$$P\left(\frac{\bar{x}-\mu}{S/\sqrt{n-1}} \leq -t_{\alpha/2,n-1}\right) = \alpha/2,$$

so that

$$P\left(\bar{x} - t_{\alpha/2,n-1}S/\sqrt{n-1} \leq \mu \leq \bar{x} + t_{\alpha/2,n-1}S/\sqrt{n-1}\right) = 1-\alpha.$$

Alternatively, note that it often causes confusion because an unbiased estimate of the population variance is not S^2 but the sample variance

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x}\ )^2,$$

so that an equivalent expression for t_{n-1} would then be

$$t_{n-1} = \frac{\bar{x} - \mu}{s/\sqrt{n}},$$

whereupon

$$P\left(\bar{x} - t_{\alpha/2,n-1}s/\sqrt{n} \le \mu \le \bar{x} + t_{\alpha/2,n-1}s/\sqrt{n}\right) = 1 - \alpha.$$

using s^2 instead of S^2.

We see from the above table that the multipliers of the sample standard error required for a 95% confidence interval with sample sizes of n = 3, 5, 7, 9, and 11 would be 4.303, 2.776, 2.447, 2.306, and 2.228. Clearly, using the sample mean plus or minus twice the standard error as an approximate 95% confidence range will always underestimate the actual 95% confidence range unless the sample size exceeds 10, say.

4.1.5 chi-square distribution

There is an ever present need in data analysis to estimate goodness of fit. That is, an experimentalist makes n observations

$$O_1, O_2, \ldots, O_n$$

and wishes to test how well a theory that predicts expected values

$$E_1, E_2, \ldots, E_n$$

fits the data. This leads naturally to the chi-square variable and chi-square tests.

Definitions

Given a normally distributed random variable x_i with mean μ and variance σ^2 it is possible to derive from it a standard normal variable z_i using

$$z_i = \frac{x_i - \mu}{\sigma}$$

which is normally distributed with mean 0 and variance 1. A sum of squares of n such independent variables defines a chi-square variable with n degrees of freedom. That is,

$$\chi^2 = z_1^2 + z_2^2 + \ldots + z_n^2$$

is chi-square distributed with n degrees of freedom, and has expectation n and variance $2n$. For $n = 1$ the density is infinite at $\chi^2 = 0$, for $n = 2$ it is that of the exponential distribution, while the distribution becomes asymptotically normal for large n.

In applications the actual distribution and its parameters are unknown and must be estimated, say from the sample. Tests based on chi-square usually require the estimation of $k \geq 0$ such parameters in order to asses the size of test statistics like C^2 defined by

$$C^2 = \frac{(O_1 - E_1)^2}{E_1^2} + \frac{(O_2 - E_2)^2}{E_2^2} + \cdots + \frac{(O_n - E_n)^2}{E_n^2}$$

which becomes asymptotically χ^2 distributed with $n - 1 - k$ degrees of freedom as $n \to \infty$. Instead of frequencies, the objective function from weighted nonlinear regression, namely

$$WSSQ = \sum_{i=1}^{n} \left\{ \frac{y_i - f(x_i, \hat{\theta})}{s_i} \right\}^2$$

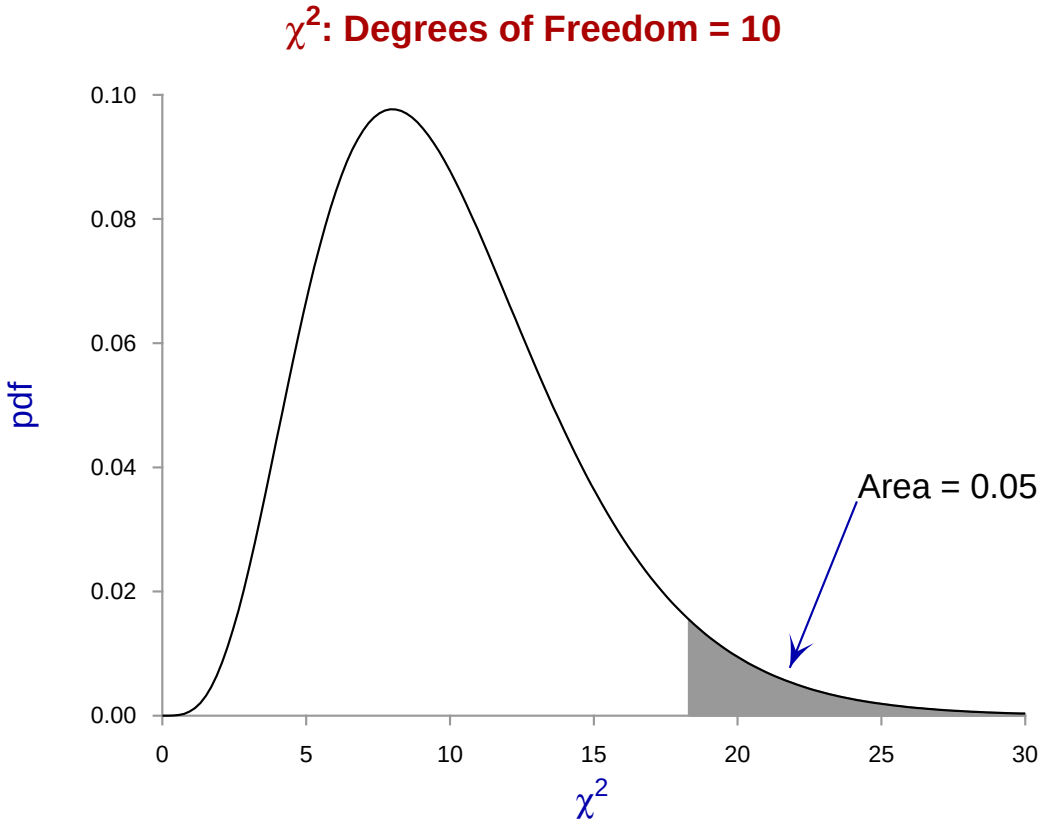
where parameters $\hat{\theta}$ have been estimated, converges to a χ^2 distribution as long as the model is correct and not over-determined, and the weights s_i are accurate.

Using the chi-square distribution

Choose `[A/Z]` from the main SimFIT menu and open program **chisqd** when the following options will be available.

```
Input: number of degrees of freedom
Input: x-values then output pdf(x)
Input: x-values then output cdf(x)
Input: alpha then output x-critical
Input: sample then test for chi-square distribution
Input: O and E values for a chi-square test
Input: contingency table for chi-square test
Input: parameters for non-central chi-square distribution
```

After input of the number of degrees of freedom a graph like the following can be viewed.



The essence of chi-square testing is to see if test statistics such as C^2 or $WSSQ$ fall in the upper tail of the appropriate χ^2 distribution. For instance, in the above graph, the shaded region contains 5% of the probability, and a test statistic falling in this region would be considered as sufficiently extreme to support rejecting a null hypothesis, such as consistency of the data with the assumed model, at the 5% significance level. Of course it is always assumed that the sample size is sufficiently large to justify treating the test statistic as a χ^2 variable instead of an approximate χ^2 variable.

4.1.6 F distribution

It is frequently necessary to compare sample variances for equality, and there are also numerous other applications for examining variance estimates, such as analysis of variance or excess variance, where a test statistic is required. The F distribution arises naturally in such contexts.

Definitions

If there are two independent random chi-square variables: U with m degrees of freedom, and V with n degrees of freedom, then the ratio of these divided by their respective degrees of freedom as in

$$F = \frac{U/m}{V/n}$$

defines the F distribution with m and n degrees of freedom. The expectation is given by

$$E(F) = \frac{n}{n-2}$$

when $n > 2$ and, rather than performing upper, lower, or two tail tests, the ratio is often inverted in some applications so that the numerator is always greater than the denominator, resulting in values of test statistics $F \geq 1$.

There is a good reason for this. Because of a special property of this distribution, lower tail percentiles are readily available from upper tail percentiles and vice versa. To see this, note that

$$\begin{aligned} P(F < F_{.05}) &= 0.05 \\ &= 1 - P(F > F_{.05}) \\ &= 1 - P\left(\frac{1}{F} < \frac{1}{F_{.05}}\right) \end{aligned}$$

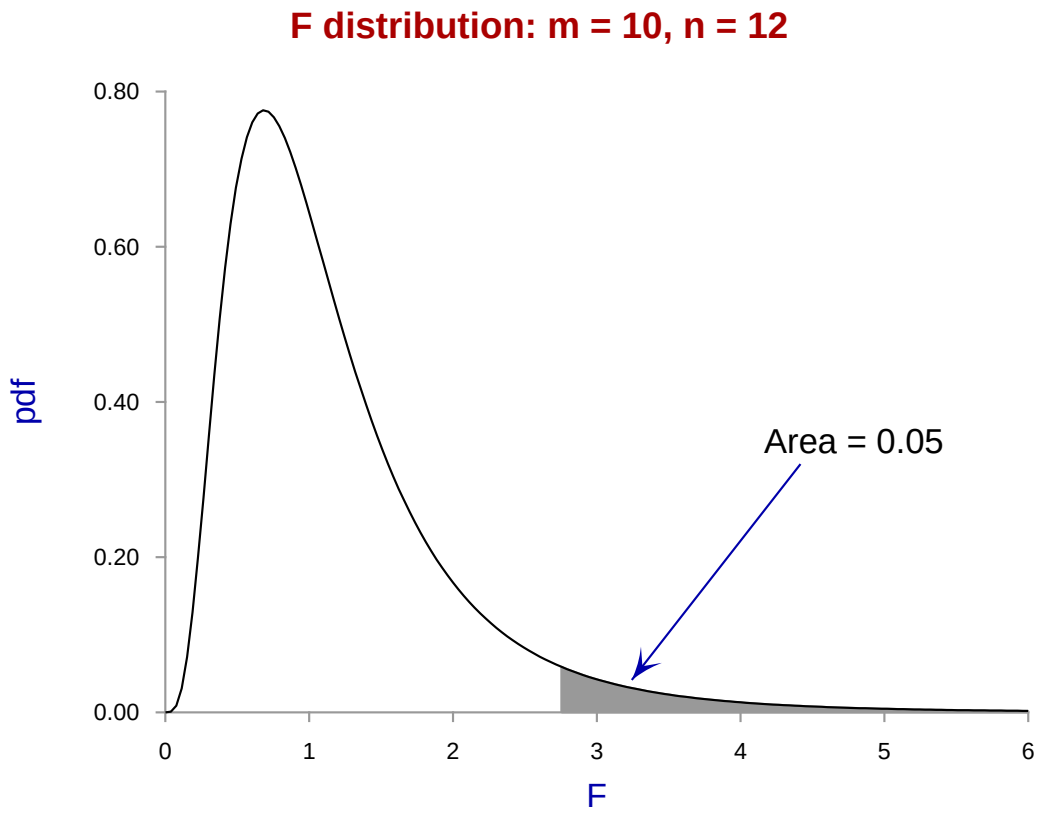
where $F_{.05}$ is the 5% critical point for the distribution of F, so that $1/F_{.05}$ is the 95% critical point for the distribution of $1/F$. Obviously, as F is distributed with m and n degrees of freedom, then $1/F$ is F distributed with n and m degrees of freedom.

Using the F distribution

Choose `[A/Z]` from the main SimFIT menu and open program **ftest** when the following options will be available.

```
Input: current parameters m and n
Input: x-value then output pdf(x)
Input: x-value then output cdf(x)
Input: alpha then calculate x-critical
Input: sample then test distributed F(m,n)
Input: sums of squares then perform F test
Do 1,2,3-way Analysis of Variance
Calculate non-central F distribution values
```

Selecting the first option and choosing $m = 10$ and $n = 12$ displays the following plot.



When using the F distribution, SimFIT will usually return the probability of a test statistic, say TS, being acceptable or not in the following way

$$P(TS \geq F) = 0.6742$$

or similar. If TS is sufficiently large to fall in the upper tail for the corresponding F distribution, such as in the region shaded in the above figure, there would then be a further message indicating the possibility for rejecting the null hypothesis at the 5% significance level, i.e. $p \leq 0.05$ or, in even more extreme cases, $p \leq 0.01$ In other words, $P(TS \geq F)$ is simply the area in the upper tail beyond the test statistic TS, i.e. the significance level for rejection of the null hypothesis.

4.1.7 Binomial distribution

Just as the normal distribution plays an important part in the analysis of measurements, the multinomial distribution is central to the analysis of counts, e.g. frequencies, contingency tables,etc. For instance, suppose an experiment leads to just one of m possible and exclusive events $E_1, E_2, \ldots, E_m$ each with fixed probability $p_i \geq 0$, then the probability that in n successive experiments there will be x_i events E_i, for $i = 1, 2, \ldots, m$ is

$$f(x_1, x_2, \ldots, x_m) = \frac{n!}{x_1!x_2! \cdots x_m!} p_1^{x_1} p_2^{x_2} \cdots p_k^{x_m}$$

where

$$p_1 + p_2 + \cdots + p_k = 1.$$

Definitions

A special case is the binomial distribution where each experiment has only one of two possible outcomes, e.g. heads or tails, success or failure, positive or negative, black or white, improvement or deterioration, and where each experiment is independent of all previous experiments. A succession of N such Bernoulli trials where the probability of success is p, so that the possibility of failure is $1 - p$, can be described by a random variable X, which is just the number of successes in the N trials. Then the probability of k successes being recorded $P(X = k)$ is

$$f(k) = \binom{N}{k} p^k (1 - p)^{N-k}$$

where the binomial coefficient, i.e. the number of ways of selecting k items from N, is

$$\binom{N}{k} = \frac{N!}{k!(N - k)!},$$

and the expected value and variance of such a binomial variable are

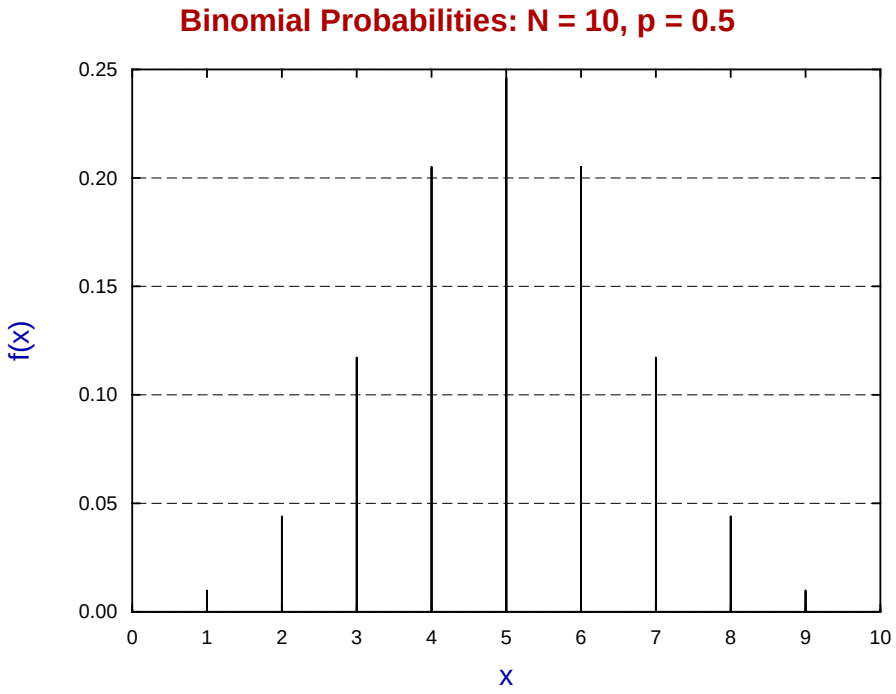
$$E(X) = Np$$
$$V(X) = Np(1 - p).$$

Simfit program binomial

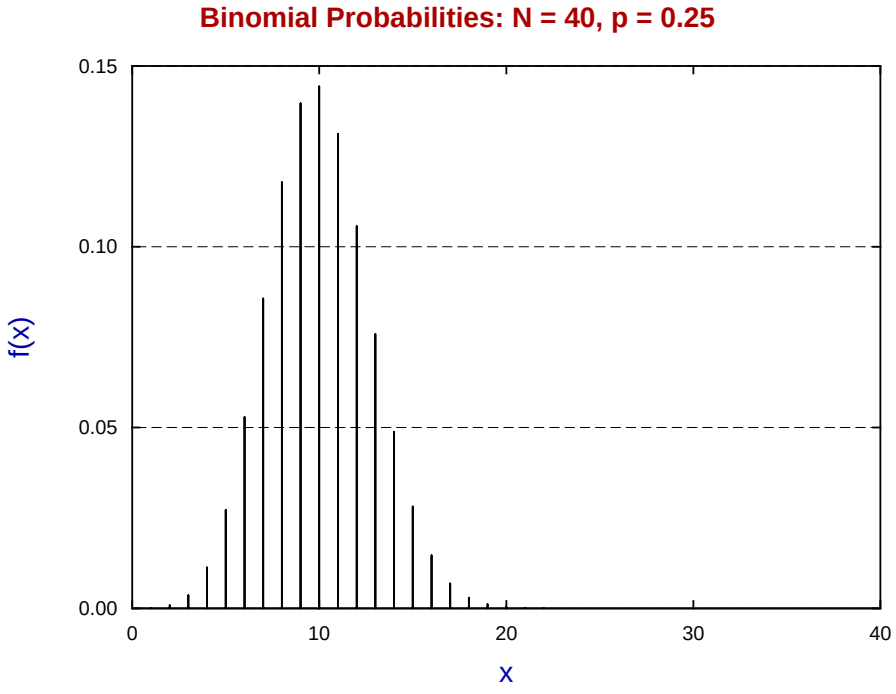
Choose [A/Z] from the main SimFIT menu and open program **binomial** when the following options will be available.

```
Input: b(N,p), and P(lambda) parameters
Input: binomial x ... calculate pmf(x)
Input: binomial x ... calculate cdf(x)
Input: binomial % ... calculate x-critical
Input: binomial N,x, calculate NCX(x)
Input: binomial N,x, estimate p, con. lim.
Input: a sample, test if distributed b(N,p)
Input: binomial x,N,t, analysis of proportions
Input: trinomial x,y,N, plot conf. reg.
Input: Poisson x ... calculate pmf(x)
Input: Poisson x ... calculate cdf(x)
Input: Poisson % ... calculate x-critical
Input: Poisson x, estimate lambda and con.lim.
Input: a sample, test if distributed P(lambda)
Calculate: power and sample size
Calculate: change confidence limits (now ,i3,%)
Calculate: using the non-central beta distribution
```

Choosing the first option allows you to change the binomial parameters as required for the subsequent binomial options. For instance, the plot with default parameters $N = 10$ and $p = 0.5$ is as follows



which demonstrates the symmetry when $p = 0.5$.
Changing the p value away from 0.5 results in skewing the probabilities and removing the symmetry as will be seen from the next plot.



Critical values

It should be noted that calculating critical points for a discrete distribution, like the binomial distribution, is problematical as the cumulative distribution is a step function, and so only ranges can be given, as in the following example for the 99%, 95%, 90%, 75%, and 50% points.

```
Current binomial parameters: N = 100, p = 0.75
P(X =< 64) = 0.00941     P(X =< 65) = 0.01643
P(X  > 64) = 0.99059 *** P(X  > 65) = 0.98357 (99%, 64 or 65)
P(X =< 67) = 0.04460     P(X =< 68) = 0.06935
P(X  > 67) = 0.95540 *** P(X  > 68) = 0.93065 (95%, 67 or 68)
P(X =< 68) = 0.06935     P(X =< 69) = 0.10379
P(X  > 68) = 0.93065 *** P(X  > 69) = 0.89621 (90%, 68 or 69)
P(X =< 71) = 0.20754     P(X =< 72) = 0.27762
P(X  > 71) = 0.79246 *** P(X  > 72) = 0.72238 (75%, 71 Or 72)
P(X =< 74) = 0.44653     P(X =< 75) = 0.53833
P(X  > 74) = 0.55347 *** P(X  > 75) = 0.46167 (50%, 74 or 75)
```

Binomial parameter confidence limits

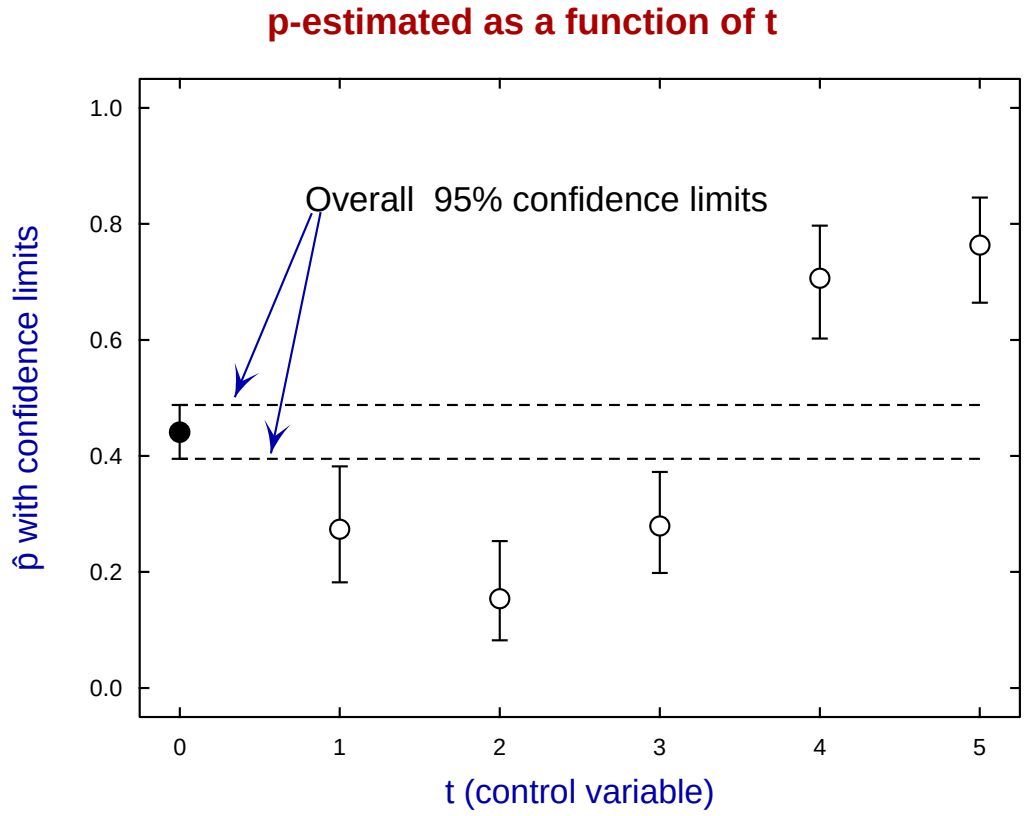
If there are X successes in N trials then the best estimate for the population parameter is the sample estimate $\hat{p}$ given by

$$\hat{p} = X/N.$$

However, to get some idea of the reliability of such estimates it is necessary to calculate confidence limits, and it often surprises experimentalists that these limits are not symmetrical, as illustrated by the next set of results.

```
Binomial p and 95% limits given X successes in N trials.
N = 10   X = 5   :Lower 95% = 0.18709, p = 0.5, Upper 95% = 0.81291
N = 100  X = 50  :Lower 95% = 0.39832, p = 0.5, Upper 95% = 0.60168
N = 1000 X = 500 :Lower 95% = 0.46855, p = 0.5, Upper 95% = 0.53145
N = 10   X = 3   :Lower 95% = 0.06674, p = 0.3, Upper 95% = 0.65245
N = 100  X = 30  :Lower 95% = 0.21241, p = 0.3, Upper 95% = 0.39981
N = 1000 X = 300 :Lower 95% = 0.27172, p = 0.3, Upper 95% = 0.32946
```

For large N and p close to 0.5 limits are approximately central, but the asymmetry increases and the confidence range becomes increasingly skewed as p moves away from 0.5 since, of course, confidence limits must remain between 0 and 1. When comparing sequential estimates for a binomial parameter, say as a function of some variable t, it is useful to plot $\hat{p}$ values with individual confidence limits along with the overall estimate from the combined sample to detect trends, as shown next.


p-estimated as a function of t
Overall 95% confidence limits
p̂ with confidence limits
t (control variable)
1.0
0.8
0.6
0.4
0.2
0.0
0
1
2
3
4
5

4.1.8 Poisson distribution

The Poisson distribution is widely used to model the occurrence of events in time or space. It can be presented as a limiting form of the binomial distribution, or more formally by way of the Poisson postulates defined using $P_n(h)$ for the probability of n events occurring in an interval of width h in time (or space) as follows.

1. The number of events in nonoverlapping intervals of time are independent.

2. Probability does not change during the occurrence of the events.

3. Probability of 1 event in a small interval of time is approximately proportional to the size of the interval.

4. Probability of 1 event in a small interval of time is much larger than that for occurrence of multiple events.

Definitions

A random integer variable X that can take all values ≥ 0 is said to obey the Poisson distribution with parameter $\lambda > 0$ if the discrete probability mass function $f(x)$ is

$$f(x) = \frac{\lambda^x}{x!} \exp(-\lambda)$$

with mean and variance both equal to λ.

Example 1: *Counting arbitrary independent events.*

In the case of the binomial distribution with parameters N and p where N is very large and p very small, then the following approximation becomes valid

$$\binom{N}{k} p^k (1-p)^{N-k} \approx \frac{(Np)^k}{k!} \exp(-Np).$$

In other words, the Poisson distribution with one parameter $\lambda = Np$ becomes a good approximation to the binomial distribution with two parameters N and p when $N \to \infty$ and $p \to 0$ but Np remains finite.

Example 2: *Counting independent events as a function of time.*

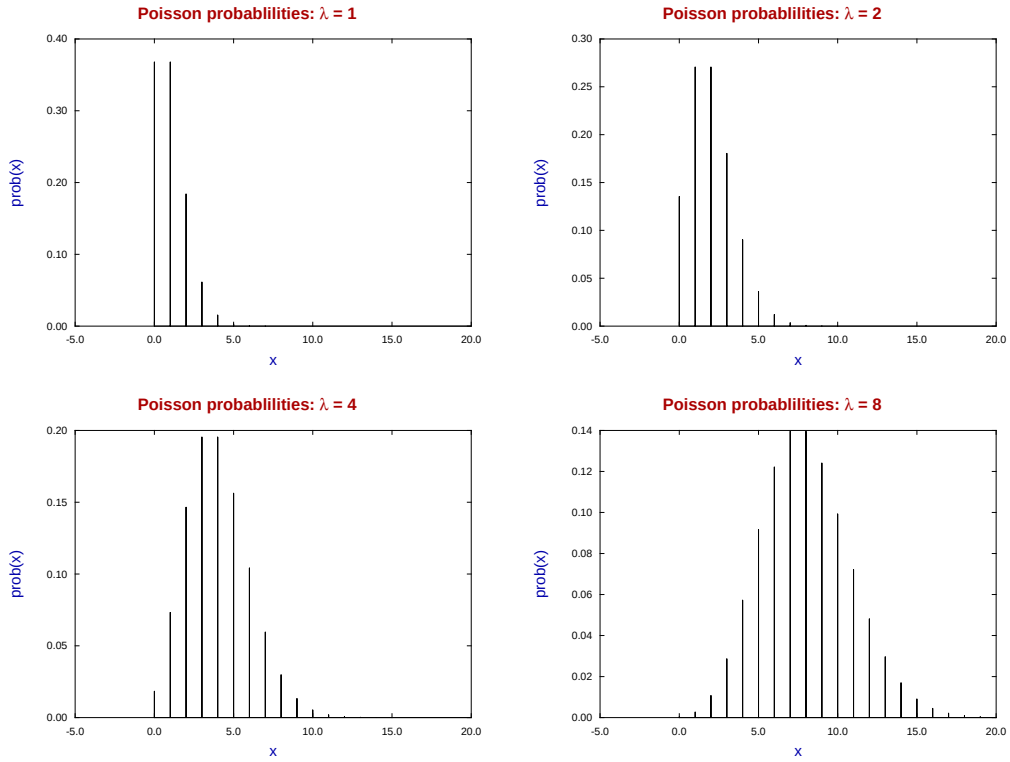
Again, for a process with an average rate of μ events per unit of time, then the probability of k events in time interval t is

$$P_k(t) = \frac{(\mu t)^k}{k!} \exp(-\mu t),$$

which defines a Poisson process with parameter $\lambda = \mu t$.

Plotting Poisson probabilities

The next plots illustrate how the distribution moves to the right as λ increses.



Simfit program binomial

Choose [A/Z] from the main SIMFIT menu and open program **binomial** when the following Poisson options will be available.

```
Input: Poisson x ... calculate pmf(x)
Input: Poisson x ... calculate cdf(x)
Input: Poisson % ... calculate x-critical
Input: Poisson x, estimate lambda and con.lim.
Input: a sample, test if distributed P(lambda)
Calculate: power and sample size
Calculate: change confidence limits (now 95%)
Calculate: using the non-central beta distribution
```

Choosing to analyze test file `poisson.tf1` for consistency with a Poisson distribution using a dispersion test, and also a Fisher exact test first warns that Bonferroni $n = 2$ then outputs these results.

Sample size	40
Sample total	44
Sample sum of squares	80
Sample mean	1.1
Lower 95% confidence limit	0.7993
Upper 95% confidence limit	1.477
Sample variance	0.8103
Dispersion (D)	28.73
$P(\chi^2 \geq D)$	0.88632
Degrees of freedom	39
Fisher exact Probability	0.91999

Note that the Bonferroni $n = 2$ declaration is a warning not to use both test statistics uncritically. Actually SimFIT often lists the results of several tests at the same time, but this is only for convenience, and users should always take note if a Bonferroni correction is required.

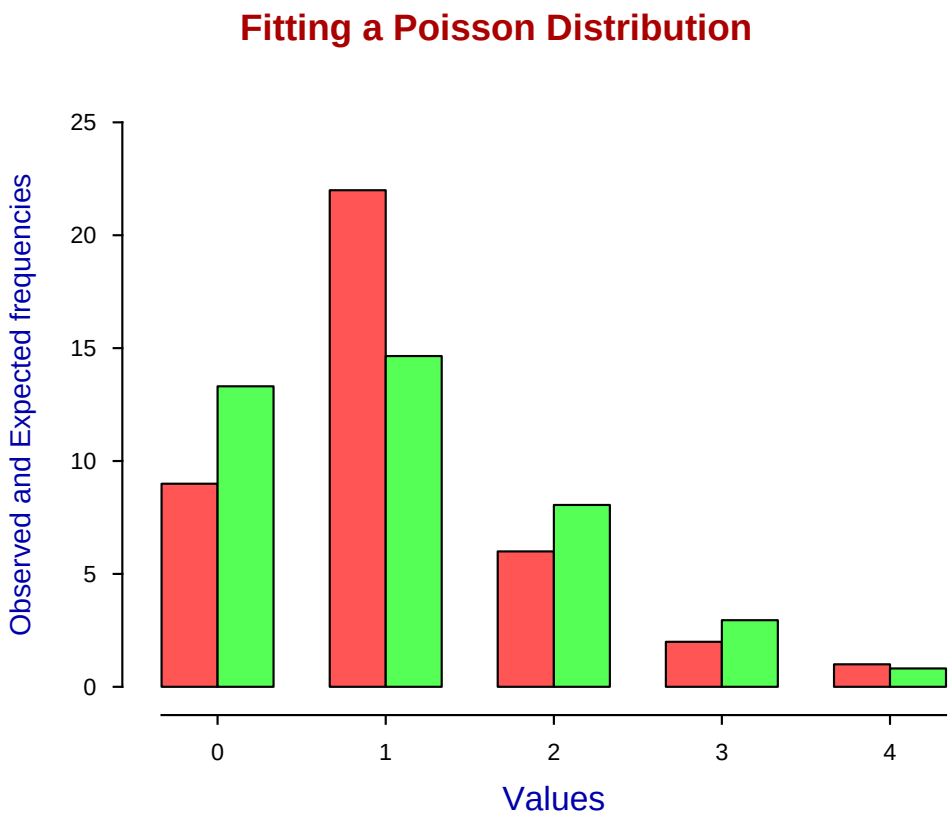
It is frequently required to confirm that it is sensible to use the Poisson distribution, with all the associated assumptions that are involved, as a model when analyzing a given data set. The dispersion test examines if there is any evidence that the dispersion D

$$D = \sum_{i=1}^{n} (x_i - \bar{x}\,)^2 / \bar{x}$$

is significantly greater than 1 (indicating over-dispersion, i.e. clumping or clustering) or significantly less 1 (indicating under-dispersion, i.e. too evenly scattered), while the Fisher exact test, which can only be done with small samples, estimates the probability of the sample based on all partitions consistent with the sample size, mean, and total. In this case there seems no evidence to reject the null hypothesis

H_0: the sample is consistent with a Poisson distribution.

The following plot compares the observed and expected frequencies in order to visualize the goodness of fit.



Note the use of a Poisson distribution to assess the significance of k, a small number of counts for one outcome, out of total number number $n > k$, by the rule of thumb of taking a 95% confidence range for the population parameter K as $k - 2\sqrt{k} \le K \le k + 2\sqrt{k}$.

Simfit program chisqd

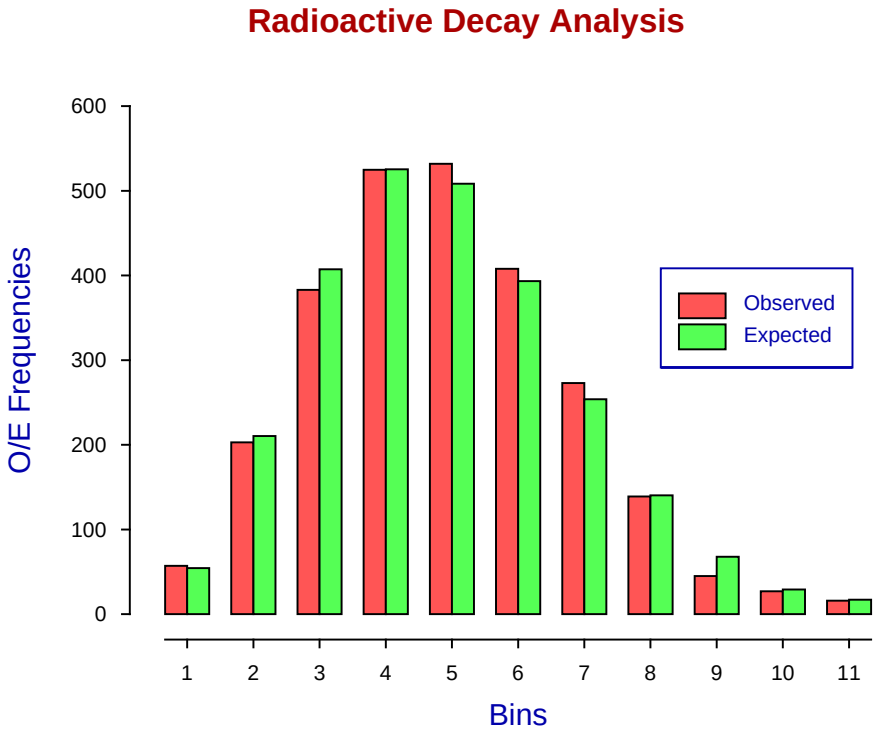
Radioactive decay is an exponential process but, during a sufficiently small time interval where the decay rate can be regarded as approximately constant, particle emission follows a Poissson distribution. In a famous experiment Rutherford counted k, the number of particles emitted in 2608 intervals of 7.5 seconds to obtain the following results, where the expected values were calculated using $\lambda = 10094/2608 = 3.87$.

k	Observed	Expected
0	57	54.399
1	203	210.523
2	383	407.361
3	525	525.496
4	532	508.418
5	408	393.515
6	273	253.817
7	139	140.325
8	45	67.882
9	27	29.189
≥ 10	16	17.075

The SIMFIT program **chisqd** was used to analyze the observed and expected frequencies to obtain these results.

Number of partitions (bins)	11	
Number of degrees of freedom	9	
Chi-square test statistic C	12.88	
$P(\chi^2 \geq C)$	0.1679	*Consider accepting* H_0
Upper tail 5% critical point	16.92	
Upper tail 1% critical point	21.67	

and the following bar chart.



4.1.9 Bivariate normal distribution

The bivariate normal distribution is an extension of the univariate normal distribution to the case of two variables. It is the basis of many procedures including Pearson product-moment correlation analysis.

Definitions

A pair of continuous random variables X and Y will constitute a bivariate distribution if there is a joint density function $f(x, y)$ which defines probabilities P as follows

$$\begin{aligned} f(x,y) &\geq 0 \\ \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\,dx\,dy &= 1 \\ \iint_{R} f(x,y)\,dx\,dy &= P(X,Y \in R) \\ \int_{a}^{b}\int_{c}^{d} f(x,y)\,dx\,dy &= P(a \leq X \leq b \text{ and } c \leq Y \leq d) \end{aligned}$$

where R is an arbitrary region in the X, Y plane, while $a \leq X \leq b$ together with $c \leq Y \leq d$ defines a rectangular region in the X, Y plane.

The marginal distributions are defined as

$$\begin{aligned} f_X(x) &= \int_{-\infty}^{\infty} f(x,y)\,dy \\ f_Y(y) &= \int_{-\infty}^{\infty} f(x,y)\,dx \end{aligned}$$

while the conditional distributions are

$$\begin{aligned} f_{X|Y}(x) &= f(x,y)/f_Y(y) \\ f_{Y|X}(y) &= f(x,y)/f_X(x). \end{aligned}$$

Of particular interest are the definitions of variances σ_x^2 and σ_y^2, and the covariance $Cov(X,Y)$ between X and Y in terms of the expectations $\mu_x = E[X]$ and $\mu_y = E[Y]$

$$\begin{aligned} \sigma_x^2 &= E[X - E(X)]^2 \\ &= E[X^2] - E[X]^2 \\ \sigma_y^2 &= E[Y - E(Y)]^2 \\ &= E[Y^2] - E[Y]^2 \\ Cov(X,Y) &= E\{[X - E(X)][Y - E(Y)]\} \\ &= E[XY] - E[X]E[Y] \end{aligned}$$

as this leads to the correlation coefficient ρ

$$\rho = \frac{Cov(X,Y)}{\sigma_x \sigma_y}$$

where $-1 < \rho < 1$, and the fact that when the two variables are independent $\rho = 0$. However, the condition $\rho = 0$ is not a sufficient condition that two variables are independent as this requires the stronger condition

$$f(x,y) = g(x)h(y).$$

So, if variables X and Y are jointly distributed as a bivariate normal distribution, the density function is

$$f(x,y) = \frac{1}{2\pi\sigma_x\sigma_y\sqrt{1-\rho^2}} \exp\left(-\frac{1}{2}Q\right)$$
$$\text{where } Q = \frac{1}{1-\rho^2}\left(\frac{(x-\mu_x)^2}{\sigma_x^2} - 2\rho\frac{(x-\mu_x)(y-\mu_y)}{\sigma_x\sigma_y} + \frac{(y-\mu_y)^2}{\sigma_y^2}\right)$$

with $\sigma_x > 0, \sigma_y > 0$, and $-1 < \rho < 1$.

Here the marginal and conditional densities for X and Y are normal, as will be clear from the following

$$f_X(x) = \frac{1}{\sqrt{2\pi}\,\sigma_x} \exp -\frac{1}{2}\left(\frac{x-\mu_x}{\sigma_x}\right)^2$$
$$f_{X|Y}(x) = \frac{1}{\sqrt{2\pi(1-\rho^2)}\,\sigma_x} \exp -\frac{1}{2(1-\rho^2)\sigma_x^2}\left(x - [\mu_x + \rho\frac{\sigma_x}{\sigma_y}(y-\mu_y)]\right)^2$$

with corresponding expressions for $f_Y(y)$ and $f_{Y|X}(y)$ by symmetry. At fixed probability levels, the quadratic form Q defines an ellipse in the X,Y plane which will have axes parallel to the X,Y axes if $\rho = 0$, but with rotated axes otherwise. Note that the marginal distribution for X has mean μ_x and variance σ_x^2, and the marginal distribution of Y has mean μ_y and variance σ_y^2, but also consider the expressions for the means in the conditional distributions, namely

$$m_{X|Y} = \mu_x + \rho\frac{\sigma_x}{\sigma_y}(y-\mu_y)$$
$$m_{Y|X} = \mu_y + \rho\frac{\sigma_y}{\sigma_x}(y-\mu_x)$$

which will be mentioned later.

Using program MAKDAT to simulate f(x,y)

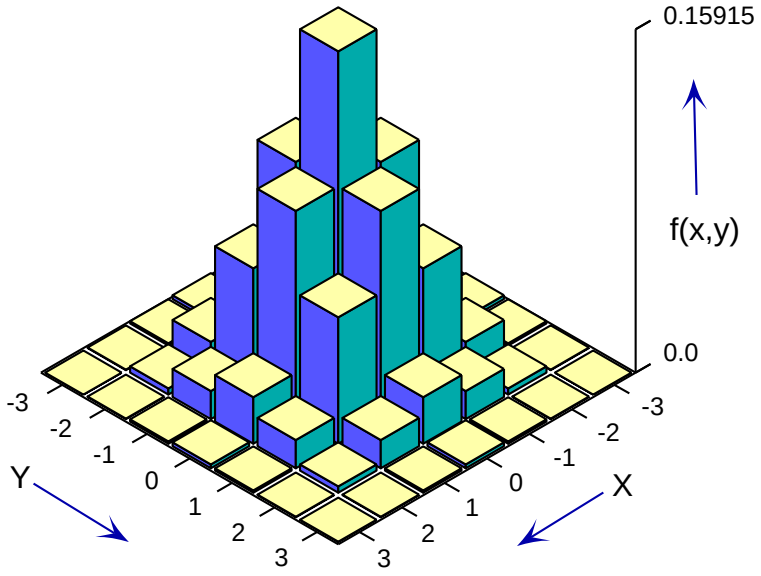
From the SIMFIT main menu select [A/Z], then run program **makdat** choosing the bivariate normal distribution. Using the parameters

$$\mu_x = \mu_y = 0,\ \sigma_x = \sigma_y = 1,\ \rho = 0,\ -3 \le x \le 3,\ -3 \le y \le 3,\ n_x = n_y = 7$$

allows the creation of the following three dimensional skyscraper plot for a bivariate normal distribution, where the volume of the individual bars indicates the probability of a x, y pair of random variables occurring in the area of the X,Y plane at the base of the bars. Increasing the number of divisions to $n_x = n_y = 50$ as in the subsequent plot indicates how the three dimensional bar chart converges to a smooth surface as the number of divisions increases.

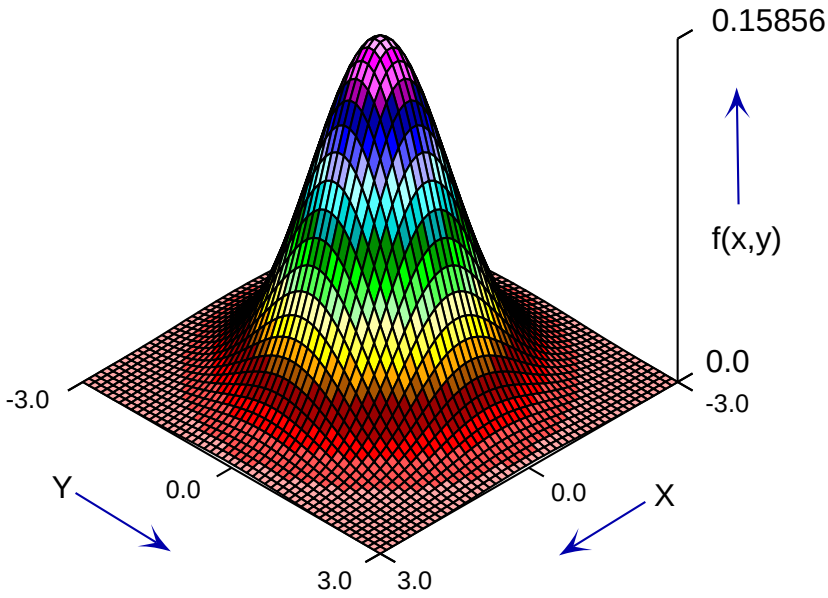
Bivariate Normal Distribution

$\mu_x = \mu_y = 0, \sigma_x = \sigma_y = 1, \rho = 0$



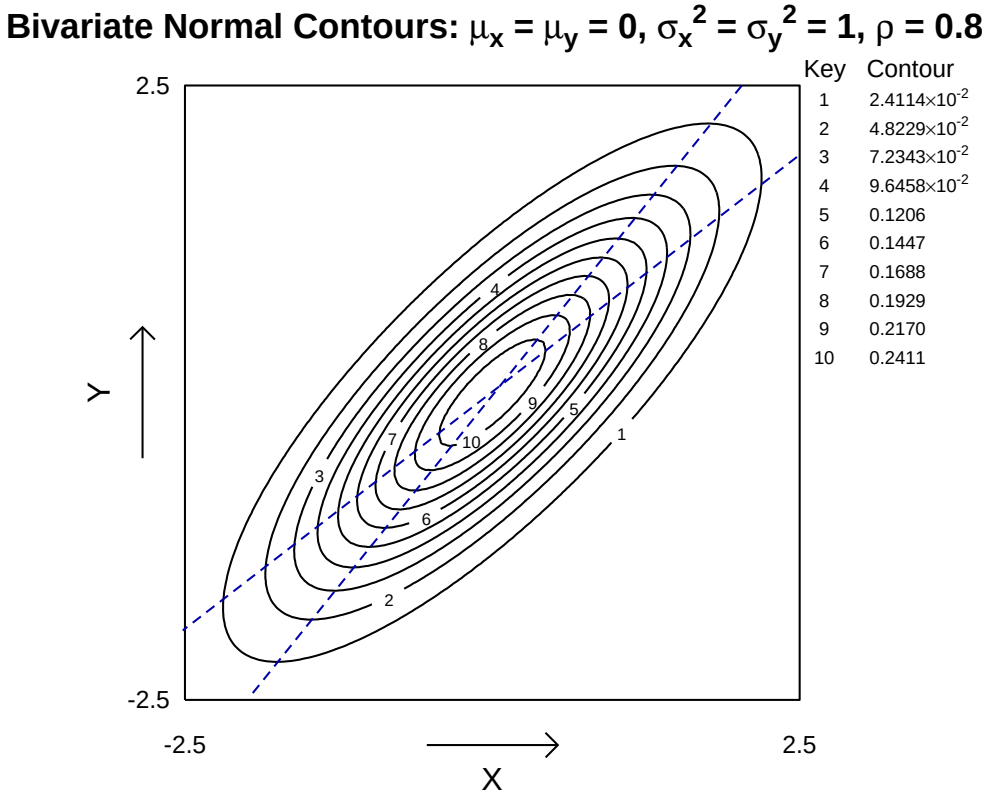
Bivariate Normal Distribution

$\mu_x = \mu_y = 0, \sigma_x = \sigma_y = 1, \rho = 0$



Plotting bivariate normal contour diagrams

Contour diagrams are at the basis of checking scatter diagrams when using Pearson product-moment analysis, and especially the practise of plotting lines on such plots to indicate linear correlation. From SIMFIT program **makdat** simply change the the current value of $\rho = 0$ which models the absence of correlation to a value such $\rho = 0.8$ for positive correlation then create the contour diagram below.



This diagram also includes the two mean regression lines indicating the mean values of the conditional distributions as functions of their arguments. These lines show the limiting positions that would be obtained with a swarm of random points following a bivariate normal distribution when plotting best-fit lines for Y as a function of X and X as a function of Y. The lines are

$$\text{For } m_{Y|X}: \quad y = \rho\frac{\sigma_x}{\sigma_y}x + \left\{\mu_y - \rho\frac{\sigma_y}{\sigma_x}\mu_x\right\}$$

$$\text{For } m_{X|Y}: \quad y = \frac{1}{\rho}\frac{\sigma_y}{\sigma_x}x + \left\{\mu_y - \frac{1}{\rho}\frac{\sigma_x}{\sigma_y}\mu_x\right\}$$

where it is understood that $\rho^2 < 1$ and $\rho \neq 0$.

Given two lines with slopes α and β then the angle between them (subject to the usual sign conventions) is given by $\tan\theta = (\alpha - \beta)/(1 + \alpha\beta)$. So, for the above conditional mean regression lines, we have

$$\tan\theta = \frac{1}{2}\left(\rho\frac{\sigma_x}{\sigma_y} - \frac{1}{\rho}\frac{\sigma_y}{\sigma_x}\right)$$

which, apart from the singular case with θ as a right angle when $\rho = 0$, defines $|\theta|$ as a decreasing function of ρ^2 thereafter.

4.2 Statistical tests

Tutorials and worked examples for simulation, curve fitting, statistical analysis, and plotting.
https://simfit.uk
https://simfit.org.uk
https://simfit.silverfrost.com

4.2.1 Introduction

Once a data set has been collected it is useful to see what statistical tests could be done to characterize a possible theoretical statistical distribution underlying the observations. Clearly success in this endeavor depends on having a sufficiently large sample with maximum possible signal to noise ratio, as well as a sensible presumed distribution. First we consider some standard data types.

- **A single one-dimensional data set.**
  Such a sample would ordinarily consist of n observations x_i, such as estimates of blood pressure in n subjects, and it might be sensible in this case to consider an underlying normal distribution.
  Such a sample will be referred to in SimFIT as a n dimensional vector X that is

$$X = x_1, x_2, \ldots, x_n.$$

  However, in SimFIT this data set would have to be submitted for analysis as a single vertical column of numbers, either from the clipboard or else from a file.

- **Two independent one-dimensional data sets, e.g. unpaired data.**
  For instance a set of n blood pressure measurements on one group, say X, and another set of m blood pressure measurements on an independent group, say Y. That is

$$X = x_1, x_2, \ldots, x_n$$
$$Y = y_1, y_2, \ldots, y_m.$$

  Here the question could be if the two sample estimates for the means and variances suggest a common distribution, and the data would have to be presented to SimFIT as two separate vertical columns of measurements, unless $n = m$ when a two-dimensional matrix would be a possible data format.

- **Two dependent one-dimensional data sets, e.g. paired data.**
  For instance a set of n blood pressure measurements on a group before medication, say X_b, and another set of n blood pressure measurements on the same group, say X_a, after medication. That is

$$X_a = x_{a1}, x_{a2}, \ldots, x_{an}$$
$$X_b = x_{b1}, x_{b2}, \ldots, x_{bn}.$$

  Here a more searching test for equality of means, that is for the presence or absence of a treatment effect, could be conducted because in such cases the obvious correlation between the groups can be exploited.
  In this instance the two samples could also be submitted to SimFIT as a matrix with n rows and 2 columns, or as two columns selected from a n by m matrix if $m > 2$.

Types of data

Before summarizing the simple statistical tests options in SimFIT it must be clear that there are essentially two types of data.

1. **Continuous**
   These would be observations that can take values in a range and are collected using apparatus. Examples could be where X is temperature measured using a thermometer, or time measured by a clock. Of course continuous variables by definition can have an infinite number of values between any two limits but, given the limits of measurement accuracy, they could well be collected as integers. For example temperature to the nearest degree, or time to the nearest second, would still be analyzed using continuous statistical distributions even though the values are expressed as integers.

2. **Discrete**
   These would be observation that are definitely integers and are often presented as counts. Examples could be where X is the number of observations in a limited number of defined categories, such as the number responding to treatment or not responding to treatment in a group of subjects. A special case is dichotomous data where each experiment has only one of two possible outcomes, for example improvement or deterioration, say 0 or 1. Such categorical data are analyzed using discrete statistical distributions, but often the test statistics are continuous random variables, such as a chi-square statistic resulting from a contingency table.

Types of test

Another distinction that can be made separates statistical tests into one of two categories.

A) Parametric tests.
These depend upon choosing a defined statistical distribution to model the data. If the model is correct these are the most powerful tests. However, if the model is wrong, or the parameters assumed for the model are incorrect or are estimated from the sample, then performance is degraded. In extreme cases the test may not just be compromised but could lead to completely erroneous conclusions.

B) Nonparametric tests.
These do not depend on an assumed model and frequently use ranks instead of just measured values. They are much weaker than parameteric tests but have the advantage that they seldom lead to false conclusions. That is, for instance, why the Mann-Whitney U test for equal medians is frequently preferred to the Student t test for equal means. It should be noted however that nonparametric tests often lead to test statistics that are only asymptotic to known continuous distributions, so that they can require large samples for reliability.

Statistics

Any function evaluated using a data set can be called a statistic, and some listed below are used as test statistics, that is, numbers that can tested for extreme values given a data set and a theoretical distribution.

a) The sample mean
Given a sample $X = x_1, x_2, \ldots, x_n$ of size n, the sample mean $\bar{x}$ defined as $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ estimates the center of the distribution, as opposed to the median which is the point where half of the sample is below the median and half above. The sample mean is frequently used in parametric tests, and the sample median in nonparametric tests.

b) The sample variance
The sample variance s^2 is defined as

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

and its square root s is the sample standard deviation. It is also known as the dispersion and sums up the spread of the data.

c) The sample cumulative distribution

Suppose the sample is rearranged into a non-decreasing order, then the sample cumulative distribution function $C(x_i)$ is a step function which is zero for values below x_1, one for values greater than or equal to x_n, and increases in steps of $1/n$ at each consecutive value of x_i.

d) Rank

The rank of an observation is the position it would take if the sample was to be arranged into non-decreasing order, and ranks are used in many nonparametric tests.

Summary of available tests

- **1-sample t test.**
  This is a parametric test used to check if a single sample of observations can be considered as arising from a defined normal distribution with a known mean which users can input interactively.

- **1-sample Kolmogorov-Smirnov test.**
  This is a nonparametric test to see if a single sample is consistent with one of a known set of standard distributions. The test statistic calculated is the maximum vertical distance between the sample cumulative distribution and the theoretical distribution. It is only really useful when the assumed distribution is a known continuous distribution with specified parameters, and is much weaker with discrete distributions, or where the parameters are estimated from the sample.

- **1-sample Shapiro-Wilks test for normality.**
  This is a recommended test to see if a sample is consistent with a normal distribution. It is based on the correlation between the sample scores and the expected normal scores.

- **1-sample Dispersion and Fisher exact Poisson test.**
  This tests if a sample of non-negative integers is consistent with a Poisson distribution. It is based on examining the sample variance to see if the data suggest over-dispersion due to clumping or an over-uniform dispersion, both of which can suggest departure from the behavior expected for a Poisson distribution. The Fisher exact test is only performed for small samples.

- **2-sample unpaired t and variance ratio tests.**
  This is the most frequently used test to see if two samples have the same mean. It relies on the samples being normally distributed with the same variance and, with large samples, these two assumptions can be tested interactively. It is essentially analysis of variance (ANOVA) but with only two columns.

- **2-sample paired t test.**
  This depends on the same assumptions as the 2-sample unpaired t test for equality of means, but in the additional circumstance that pairs of corresponding values are necessarily correlated. For instance, when the column vectors are body temperatures measured with the same subjects but before and after treatment. The correlation allows the use of a test statistic that is more searching than with the 2-sample unpaired t test.

- **2-sample Kolmogorov-Smirnov test.**
  This is a nonparameteric test for equality of distribution. It is based on the maximum vertical distance between the two sample cumulative distributions but is rather weak and requires large samples.

- **2-sample Wilcoxon-Mann-Whitney U test.**
  This is the most widely used nonparametric test to see if two samples can be regarded as having the same but unspecified distribution. It is based on the extent to which one of the samples dominates the other, that is, has a larger median.

- **2-sample Wilcoxon signed-ranks test.**
  Just as the Wilcoxon-Mann-Whitney U test is the nonparametric equivalent of the unpaired t test, this is the nonparametric equivalent of the paired t test and tests for equality of medians in two paired samples.

- **Chi-square test on observed and expected frequencies.**
  Given a set of n actual observed frequencies this tests if the observed frequencies are consistent with those generated as expected frequencies by some assumed distribution. It requires users to supply the expected frequencies along with the observed frequencies.

- **Chi-square and Fisher-exact contingency table tests.**
  The chi-square test can always be done on an arbitrary n by m contingency table, but the Fisher-exact test is only useful with small contingency tables. Yates's continuity correction is used for two by two tables.

- **McNemar test.**
  This requires paired samples of dichotomous data (i.e. values 0 or 1) as a two by two contingency table.

- **Cochran Q repeated measures test.**
  This also requires dichotomous data, but in a repeated measures design.

- **Binomial test.**
  This test checks to see if a set of dichotomous observations are consistent with a binomial distribution. The binomial p value and number of trials N are input along with the number of successes or failures.

- **Sign test.**
  Checks if a set of outcomes such as success or failure are consistent with a binomial distribution with $p = 0.5$, that is, equally likely outcomes.

- **Run test.**
  This is used to test if a set of residuals have a succession of signs that is consistent with equally likely positive and negative values occurring randomly along the sequence and not clustering to suggest a biased fit. Whereas the sign test just examines the overall number of positive and negative signs, the run test also depends on the order of occurrence. For instance, if a model fitted to 20 data points resulted in 10 positive and 10 negative residuals the sign test would report this as perfectly normal. If there were 10 positive residuals followed by 10 negative residuals, that is the sign pattern ++++++++++----------, the run test would draw attention to a badly fitting model.

- **F test for excess variance.**
  This test is performed automatically when SimFIT fits a nested set of models to some data but can also be done interactively using this option. It is based on the fact that adding extra parameters to a model, for example higher order terms in a polynomial, will generally improve the fit, i.e. decrease $WSSQ$, the weighted sum of squared residuals resulting from fitting. This test examines if the increased number of parameters required to decrease the $WSSQ$ is justified on statistical grounds.

- **Runs up and down for randomness.**
  This is used to test for correlation in a set of numbers, for example pseudo random numbers from a random number generator. The runs up test is based on counting the lengths of runs of numbers increasing in magnitude within in a sequence, and the runs down test is done by simply changing the signs of all the numbers and doing the runs up test.

- **Mood and David tests for equal dispersion.**
  This is a nonparametric test for variance equality in two samples. It examines the ranks of observations in a pooled sample.

- **Kendall coefficient of concordance.**
  This measures the degree of agreement between k comparisons of n objects.

- **Bartlett and Levene tests for homogeneity of variance**
  Analysis of variance is based on the presumption that the samples under investigation are all normally distributed with the same variance. This test examines if all the variances are consistent with this assumption.

4.2.2 1-sample t test

The one sample t test is one of many tests designed to see if a sample can be regarded as consistent with a normal distribution but, for this test, it is also a strict requirement that the theoretical mean value is μ_0, a parameter that is known in advance from previous investigations.

To be precise, the user has a sample (i.e. vector X) of n observations

$$X = (x_1, x_2, \ldots, x_n)$$

and wishes to test if these numbers are consistent with a normal distribution where the mean μ_0 has been previously estimated with great precision from an independent very large sample. Preferably the data should cover a wide range and n should not be too small, say $n \geq 20$?

Choose [A/Z] from the main SIMFIT menu, open program **ttest**, select the 1-sample-t-test option, input a theoretical mean $\mu_0 = 0$, then analyze the test data set to obtain the following results.

Normal distribution test

Data: Test file `normal.tf1` with 50 pseudo-random numbers	
Shapiro-Wilks statistic W	0.9627
Significance level for W	0.1153
Conclusion: *Tentatively accept normality*	

One sample t test

Number of x-values	50
Number of degrees of freedom	49
Theoretical mean (μ_0)	0
Sample mean ($\bar{x}$)	−0.02579
Standard error of mean (SE)	0.1422
$TS = (\bar{x} - \mu_0)/SE$	−0.1814
$P(t \geq TS)$ (upper tail p)	0.5716
$P(t \leq TS)$ (lower tail p)	0.4284
p for two tailed t test	0.8568
Difference $D = \bar{x} - \mu_0$	−0.02579
Lower 95% confidence limit for D	−0.3116
Upper 95% confidence limit for D	0.2600
Conclusion: *Consider accepting equality of means*	

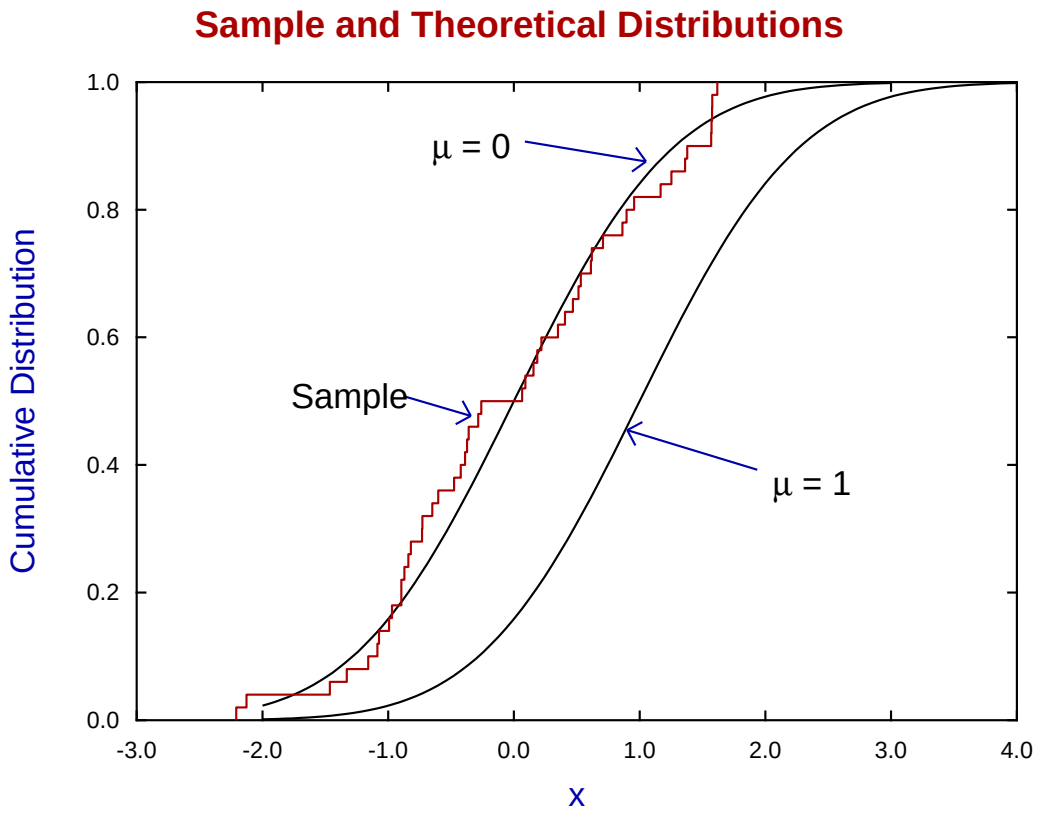
The analysis begins by performing a Shapiro-Wilks test to see if the sample can be regarded as normally distributed using the sample estimates for both the mean and the standard deviation. This test is less powerful than the one sample t test and, if it rejects the null hypothesis of an arbitrary normal distribution, the subsequent results can be ignored. Clearly, there is no evidence to support rejection of the hypothesis of an arbitrary normal distribution. The further analysis goes on to examine the size of the difference between the sample mean $\bar{x}$ and the theoretical mean μ_0, given the sample variance estimate, using the t distribution, and concludes that the data do appear to be normally distributed with mean close to $\mu_0 = 0$, as the two tail p value is 0.8568.

In order to appreciate the sensitivity of this test to the assumed value for μ_0, you should repeat the test using a different value for the theoretical mean, say $\mu_0 = 1$ for instance, which leads to the next results.

One sample t test	
Number of of x-values	50
Number of degrees of freedom	49
Theoretical mean (μ_0)	1
Sample mean ($\bar{x}$)	−0.02579
Standard error of mean (SE)	0.1422
$TS = (\bar{x} - \mu_0)/SE$	−7.214
$P(t \geq TS)$ (upper tail p)	1.0000
$P(t \leq TS)$ (lower tail p)	0.0000
p for two tailed t test	0.0000
Difference $D = \bar{x} - \mu_0$	−1.026
Lower 95% confidence limit for D	−1.312
Upper 95% confidence limit for D	−0.740
Conclusion: *Reject equality of means at 1% significance level*	

Obviously the Shapiro-Wilks test result is unchanged, but now the lower tail p value strongly indicates that the sample is shifted to the left of a normal distribution with $\mu = 1$, and the two tail p value, which is twice the lesser of the upper and lower tail probabilities, clearly rejects the null hypothesis $H_0 : \mu_0 = 1$.

The following graph makes it clear why $H_0 : \mu_0 = 1$ was rejected.



4.2.3 1-sample Kolmogorov-Smirnov test

The Kolmogorov-Smirnov nonparametric test is used to check if a sample is consistent with a known continuous distribution. It is most powerful with large samples from such a distribution when the true parameters are known, and is much weaker if parameters have to be estimated from the sample, or if a discontinuous distribution is to be considered.

To be precise, the user has a sample (i.e. vector X) of n observations

$$X = (x_1, x_2, \ldots, x_n)$$

and wishes to test if these numbers are consistent with a known distribution, where the parameters have been previously estimated with great precision from an independent very large sample, or are known due to further information. Preferably the data should cover a wide range and n should not be too small, say $n > 20$?

The test is based upon the largest vertical distance where the sample cumulative exceeds the theoretical distribution ($D+$), the largest vertical distance where the theoretical distribution exceeds the sample cumulative distribution ($D-$), or the the maximum of these ($D = \max(D+, D-)$). The standardized Z values are defined as $Z = D\sqrt{n}$, and SimFiT calculates exact p for small samples, but uses a series expansion for Z with large samples.

Choose [A/Z] from the main SimFiT menu, open program **simstat**, select statistical tests, then choose the 1-sample Kolmogorov-Smirnov test. First of all select to test for a uniform distribution $U(A, B)$ with $A = 0$ and $B = 2$ to get the following results.

Kolmogorov-Smirnov one sample test 1: Uniform(A,B)

Data: test file g08cbf.tf1 (Kolmogorov-Smirnov 1-sample test)
Parameters fixed by user: A = 0, B = 2
Sample size = 30, i.e. number of x-values

H_0: $F(x)$ equals $G(y)$ (x and theory are comparable) against
H_1: $F(x)$ not equal to $G(y)$ (x and theory not comparable)

D	0.2800	
Z	1.534	
p	0.0143	Reject H_0 at 5% significance level

H_2: $F(x) > G(y)$ (x tend to be smaller than theoretical)

D	0.2800	
Z	1.534	
p	0.0071	Reject H_0 at 1% significance level

H_3: $F(x) < G(y)$ (x tend to be larger than theoretical)

D	0.02333
Z	0.1278
p	0.5000

Here $F(x)$ is the sample distribution while $G(y)$ is the theoretical distribution, and these figures are interpreted as follows. The three D values were

$$D+ = 0.28$$
$$D- = 0.02333$$
$$D = 0.28$$

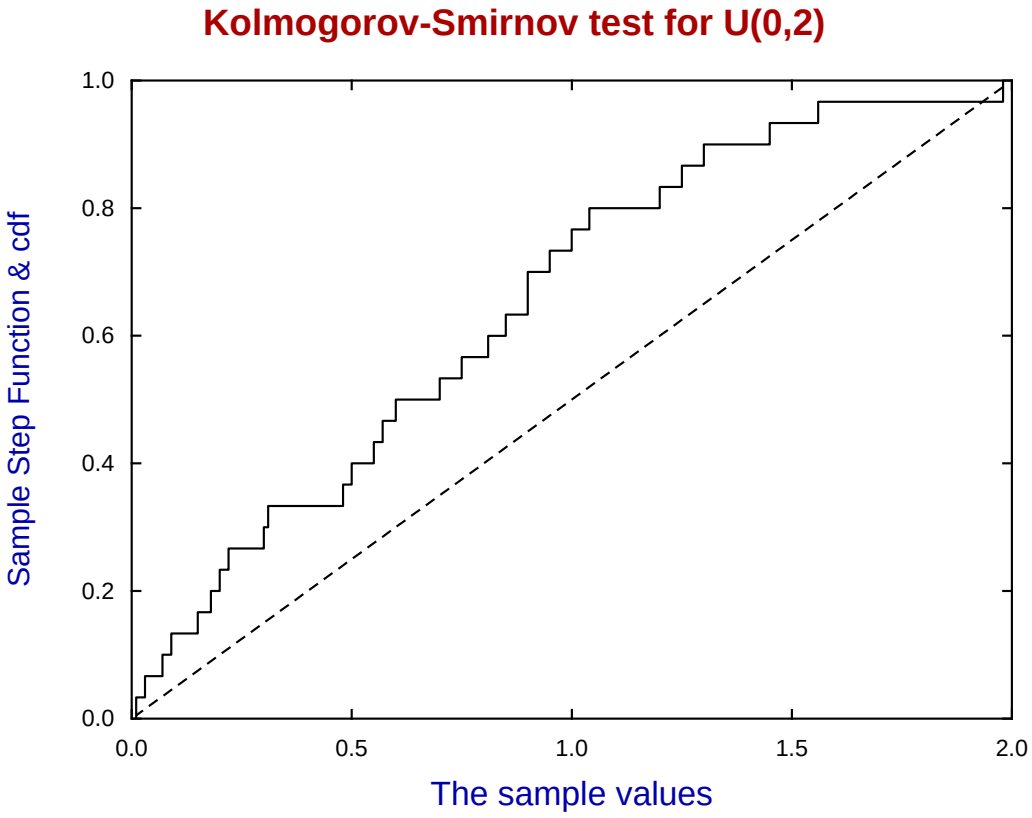
and the three cases are therefore as follows.

1. Reject H_0 against H_1 as a two-tail test indicates $F(x)$ and $G(y)$ are unlikely to be equal ($p = 0.0143$).

2. Reject H_0 against H_2 as a one-tail test indicates that $F(x)$ is more likely to be larger than $G(y)$ than to be equal to it ($p = 0.0071$).

3. Do not reject H_0 against H_3 as a one tail test offers no support for the case that $F(x)$ is smaller than $G(y)$ ($p = 0.5$).

These results clearly reject the case $F(x) = G(y)$, in favor of H_1, i.e. that x values tend to be smaller than y values, indicating that the sample cumulative distribution was heavily displaced to the left of the theoretical distribution. To confirm this interpretation visually we can plot the sample cumulative distribution and theoretical distribution as in the next graph.



Having rejected the null hypothesis H_0: the sample is distributed as $U(0, 2)$, we can try another theoretical distribution.

So, in this next case, we proceed to test the null hypothesis that the theoretical distribution is normal with parameters $\mu = 0.6967$ and $\sigma^2 = 0.2564$ as estimated from the sample. We conclude that this cannot be rejected and that the sample distribution is close to the theoretical distribution, as displayed graphically.

Kolmogorov-Smirnov one sample test 2: Normal(μ, σ^2)

Parameters estimated from sample are:
μ = 6.967E-01, se = 9.244E-02, 95%confidence limits = (5.076E-01, 8.857E-01)
σ = 5.063E-01, σ^2 = 2.564E-01, 95%cl = (1.626E-01,4.633E-01)

H_0: $F(x)$ equals $G(y)$ (x and theory are comparable) against
H_1: $F(x)$ not equal to $G(y)$ (x and theory not comparable)

D	0.1108
Z	0.6068
p	0.8162

H_2: $F(x) > G(y)$ (x tend to be smaller than theoretical)

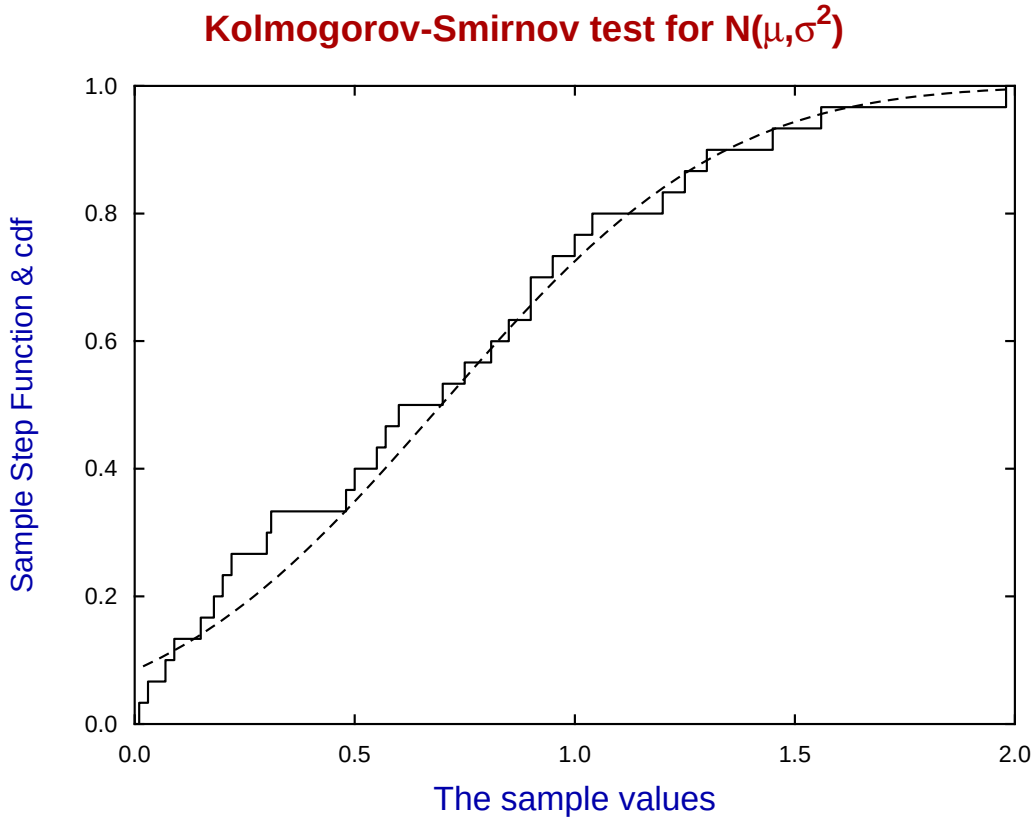
D	0.1108
Z	0.6068
p	0.4081

H_3: $F(x) < G(y)$ (x tend to be larger than theoretical)

D	0.08753
Z	0.4794
p	0.4801

Shapiro-Wilks normality test (Note: Bonferroni $n \geq 2$):

W	0.9529	
p	0.2019	Tentatively accept normality



Note that SIMFIT often presents the results from several tests at the same time, which is convenient for preliminary data investigation but not for precise analysis. Which is why, as in the last table, hints about the Bonferroni correction are frequently given.

4.2.4 1-sample Shapiro-Wilks test

Of the many tests for normality, the Shapiro-Wilks test is usually recommended to check if a sample is consistent with a normal distribution, and it leads naturally to the normal scores plot which is a convenient technique to examine normality graphically.

To be precise, the user has a sample (i.e. vector X) of n observations

$$X = (x_1, x_2, \ldots, x_n)$$

and wishes to test if these numbers are consistent with a normal distribution, where the parameters have been previously estimated with great precision from an independent very large sample, or are known due to further information. Preferably the data should cover a wide range and n should not be too small, say $n > 20$?

There are many statistical methods provided by SimFIT program **normal** for doing this as now summarized.

- **Kolmogorov-Smirnov**
  This only has advantages in the case when both parameters are known, and not estimated from the sample.

- **One sample t**
  This always uses the sample variance, and also is best when the true mean is known in advance.

- **Chi-square**
  This is also a rather poor test, especially if the expected values are estimated from the sample.

- **Shapiro-Wilks**
  This is now generally thought to be the best all purpose test where parameters are estimated from the sample, but it does require intensive computation which can limit the maximum value of n, e.g. $n \leq 5000$ in SimFIT.

In addition there are the following graphical methods.

- **Histogram**
  This can easily be done, but SimFIT can also be used to rationalize the situation by analyzing the data after transformation to $U(0, 1)$, as it is somewhat easier to detect deviations of a histogram from the case where all cells have equal expected frequency from one where a bell-shaped curve is anticipated.

- **Cumulative distribution**
  This is much better than a histogram, as histogram shape depends on the number of bins whereas the sample cumulative distribution is of fixed shape.

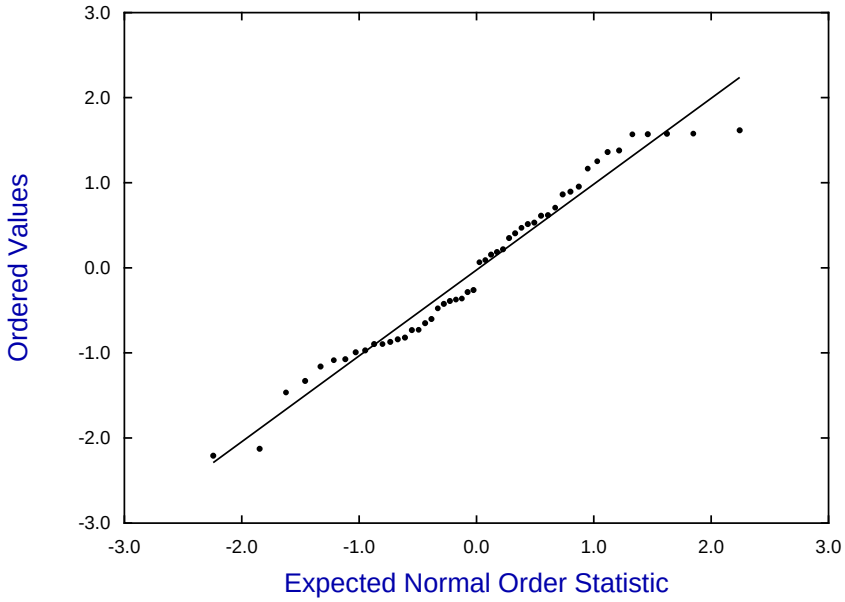
- **Normal scores**
  The n values that can be calculated to divide a standard normal cumulative distribution into $n + 1$ sections, each of area $1/(n+1)$, are referred to as normal scores. A plot of sample scores against normal scores should be close to a straight line, and is widely recognized as the best plot for detecting departure from normality. A variant is the half normal plot, where negative values are changed in sign, and this is usually preferred for testing that residuals from regression do not differ too widely from normality.

In the case of mixtures of normal distributions there are dedicated statistical tests, but SimFIT program **qnfit** also provides the ability to fit histograms or cumulative distributions if the sample size is very large, and the distributions well separated. The advantage here is that the SimFIT graphical deconvolution technique can be used to display how the distribution is made up from sums of normal distributions.
Here is the conclusion from program **normal** and the test data `normal.tf1` provided, which establishes that a normal distribution cannot be rejected, as the Shapiro-Wilks test statistic is $W = 0.9627$ with $p = 0.1153$, i.e. close to $W = 1$, which indicates strong correlation between the sample scores and normal scores.

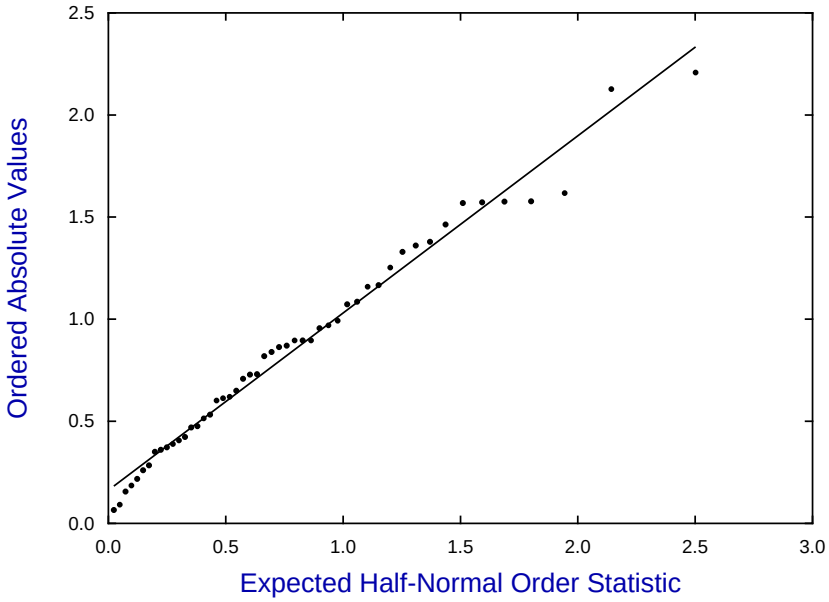
Normal distribution test

Data: Test file `normal.tf1` with 50 pseudo-random numbers	
Shapiro-Wilks statistic W	0.9627
Significance level for W	0.1153
Conclusion: *Tentatively accept normality*	

Normal Scores Plot: r = 0.9851



Half-Normal Scores Plot: r = 0.9922



4.2.5 1-sample Poisson distribution test

The situation envisaged is where a user has a set counts of nonnegative integers and wishes to see if the frequencies observed are consistent with a Poisson distribution with parameter λ. In SIMFIT this can be done using a chi-square test, a dispersion test, a Fisher exact test, or a Kolmogorov-Smirnov test.

From the SIMFIT main menu choose `[A/Z]`, then program **binomial**, and select the option to test for a Poisson distribution. The most widely used test is a chi-square test so this is done first, choosing to use the sample estimate of $\hat{\lambda} = 1.1$ instead of the current fixed value of 2, and opting for a minimum of 5 counts per bin, leading to the next results.

Chi-square test for $P(\lambda)$ with $\lambda = 1.1$

H_0: Poisson distribution for data with title:		
Test file `Poisson.tf1`: 40 random numbers		
Sample estimate used in chi-square test		
Sample estimate for λ	1.100	
Lower 95% confidence limit	0.7993	
Upper 95% confidence limit	1.477	
Mean of x-values	1.100	
Variance of x-values	0.8103	
Standard deviation of x	0.9001	
Mean using fixed λ	2.000	
Poisson dispersion value D	28.73	
$P(\chi^2 \geq D)$	0.8863	
Number of partitions (bins) used	3	
Number of degrees of freedom	1	
Chi-square test statistic C	5.857	
$P(\chi^2 \geq C)$	0.0155	*Reject H_0 at 5% level*
Upper tail 5% critical point	3.841	
Upper tail 1% critical point	6.635	

Now it should be emphasized that the chi-square test is an approximate test, as the test statistic only becomes asymptotic to a chi-square variable with large samples. Further, if any cells have a small frequency, say < 5, it is usually recommended to combine adjacent bins until this is the case. That is why choosing a minimum frequency of 5 resulted in only 3 bins. If the test is now repeated on the same data but choosing a minimum frequency of 3 the next results are obtained.

Number of partitions (bins) used	4	
Number of degrees of freedom	2	
Chi-square test statistic C	5.858	
$P(\chi^2 \geq C)$	0.0535	*Consider accepting H_0*
Upper tail 5% critical point	5.991	
Upper tail 1% critical point	9.210	

It should be noted that by simply changing the number of bins from 3 to 4 a rejection ($p = 0.0155$) becomes an acceptance ($p = 0.053$), which should serve to emphasize how the outcome of such chi-square tests depend on the number of bins.

A classical example is the famous data collected by von Bortkiewicz for 0, 1, 2, 3, or 4 deaths per year by horse kick in the Prussian cavalry during the period 1875 - 1894 for 10 groups, as this is a typical example of using the Poisson distribution to model rare events.

The 200 frequencies recorded are contained in the test file `poisson.tf2` as follows

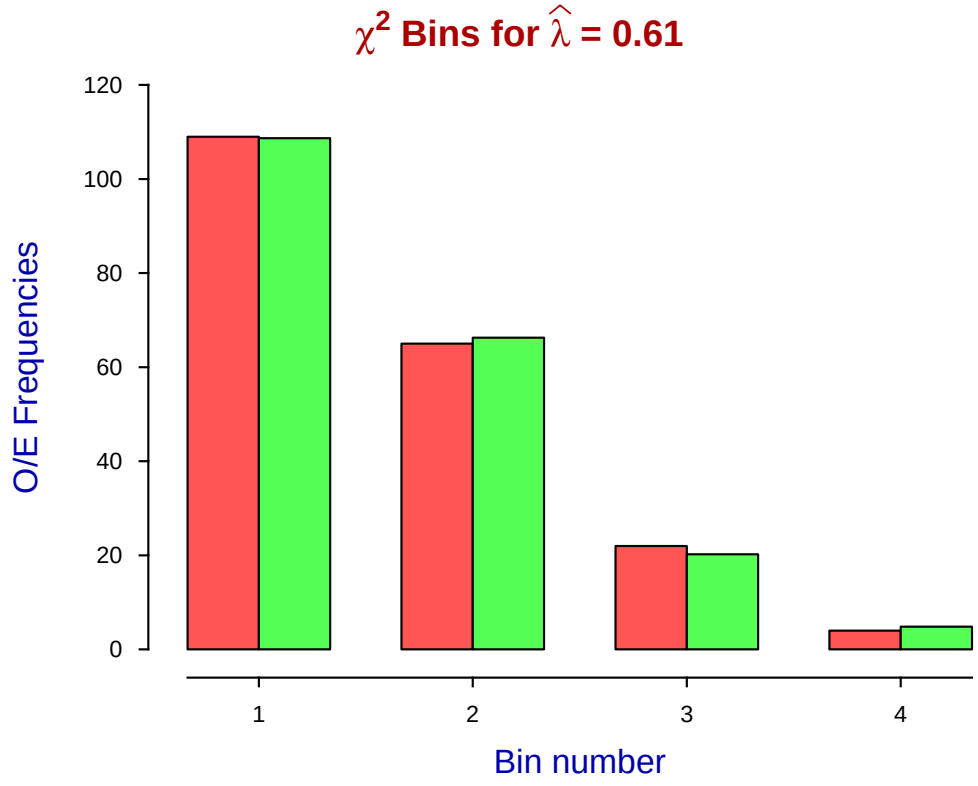
Deaths	0	1	2	3	4
Frequency	109	65	22	3	1

leading to the result, shown below, that $p = 0.8506$, so that the null hypothesis of a Poisson distribution with $\hat{\lambda} = 0.61$ cannot be rejected.

Chi-square test for $P(\lambda)$ with $\lambda = 0.61$

H_0: Poisson distribution for data with title:		
Death from horse kicks in Prussian cavalry 1875-1894		
Sample estimate used in chi-square test		
Sample estimate for λ	0.6100	
Lower 95% confidence limit	0.5066	
Upper 95% confidence limit	0.7283	
Mean of x-values	0.6100	
Variance of x-values	0.6110	
Standard deviation of x	0.7816	
Number of partitions (bins) used	4	
Number of degrees of freedom	2	
Chi-square test statistic C	0.3235	
$P(\chi^2 \geq C)$	0.8506	*Consider accepting H_0*
Upper tail 5% critical point	5.991	
Upper tail 1% critical point	9.210	

The observed and expected values used in this test are displayed below.



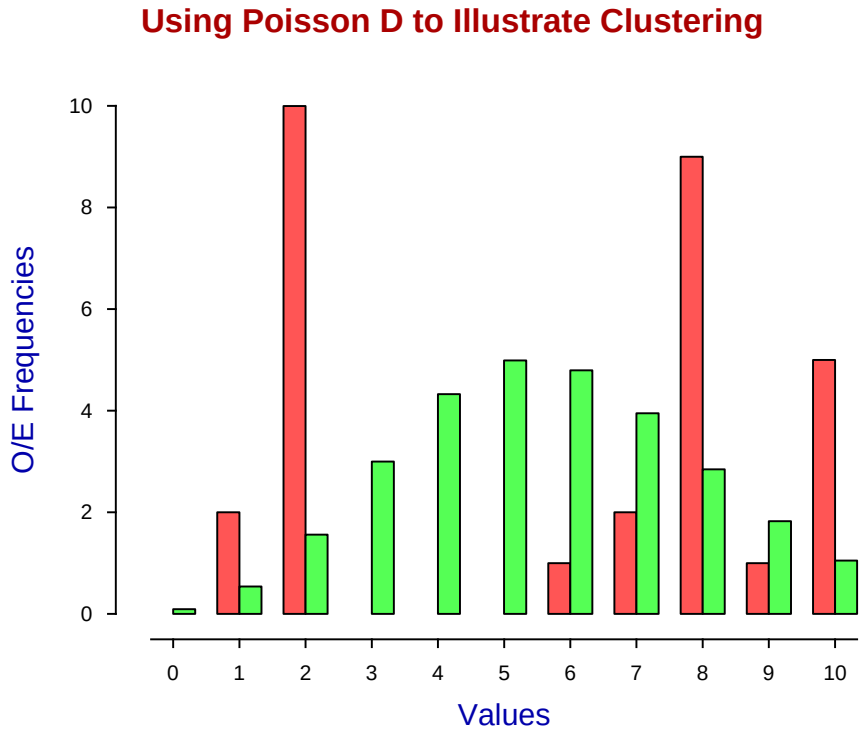
The Fisher Exact test will frequently fail with large or aberrant data sets and this will be indicated when it happens, but the dispersion test can always be used to test for data that are too uniform or too clustered to be from a Poisson distribution. This is used to study clumping of objects studied microscopically, and similar situations concerning spatial or temporal distributions of counts. Consider, for instance, the analysis of this data set contained in the test file `poisson.tf3`.

Counts	0	1	2	3	4	5	6	7	8	9	10
Frequency	0	2	10	0	0	0	1	2	9	1	5

Dispersion and Fisher-exact Poisson tests

Bonferroni $n = 2$		
Data: Poisson clumping data		
Sample size	30	
Sample total	173	
Sample sum of squares	1333	
Sample mean	5.767	
Lower 95% confidence limit	4.939	
Upper 95% confidence limit	6.693	
Sample variance	11.56	*Too large ?*
Dispersion (D)	58.16	
$P(\chi^2 \geq D)$	0.00104	*Reject H_0 at 1% sig.level*
Number of degrees of freedom	29	
Fisher exact probability	1.00000	
IFAIL = 1: Fisher p is only an upper bound		

In a Poisson distribution the mean is equal to the variance, and a variance much less than the mean suggests a distribution that is too uniform, while a variance exceeding the mean could indicate clustering, as will be clear from the results above and the following plot.



4.2.6 2-sample Unpaired t test

The unpaired t test is used to see if it is reasonable to conclude that two sets of independent observations have the same population means. It is based on the assumptions that

- Both samples are normally distributed
- Both samples have the same variance
- Both sample sizes are greater than 1 (and preferably very much greater)

and hence it is equivalent to analysis of variance (ANOVA) with just two columns.

To be precise, the user has two samples (i.e. vectors X and Y) with m and n observations

$$X = (x_1, x_2, \ldots, x_m)$$
$$Y = (y_1, y_2, \ldots, y_n)$$

and wishes to test the null hypothesis that the samples have the same population means, μ_x and μ_y, against the alternative hypothesis that they are not equal, or possibly the one-sided alternatives. That is

$$H_0 : \mu_x = \mu_y$$
$$H_1 : \mu_x \neq \mu_y$$
$$H_2 : \mu_x > \mu_y$$
$$H_3 : \mu_x < \mu_y$$

and SimFIT provides all the information that is required to perform such tests.

From the main SimFIT menu select [A/Z], choose to open program **ttest**, then analyze the test files provided.

The first choice offered is to test for a normal distribution and equal variances and, if these are chosen, we get the following analysis.

Normal distribution test 1		
Data: X-data for t test		
Shapiro-Wilks statistic W	0.9539	
Significance level for W	0.7146	*Tentatively accept normality*

Normal distribution test 2		
Data: Y-data for t test		
Shapiro-Wilks statistic W	0.9360	
Significance level for W	0.5089	*Tentatively accept normality*

This informs us that the Shapiro-Wilks test does not provide any evidence to reject the assumption that both samples are normally distributed. However, this test should only be used if both m and n are sufficiently large, say $m, n > 20$, so in this case we should really have chosen not to do Shapiro-Wilks tests.

Then a F test for equality of variances is carried out, yielding the next results.

F test for equality of variances	
X-data: Test file `ttest.tf2`: for paired t test with TTEST.TF3	
Number of x-values	10
Mean x	14.80
Sample variance of x	21.96
Sample standard deviation of x	4.686
Y-data: Test file `ttest.tf3`: for paired t test with TTEST.TF2	
Number of y-values	10
Mean y	16.10
Sample variance of y	24.54
Sample standard deviation of y	4.954
Variance ratio VR	1.118
Degrees of freedom (numerator)	9
Degrees of freedom (denominator)	9
$P(F \geq VR)$	0.4354
Two tail p value	0.8708
Conclusion: *Consider accepting equality of variances*	

Again, although this does not reject equality of variances, it should be pointed out that this test is only reliable with large samples, say $m, n > 50$ and should not have been performed with such small samples. It is anticipated that, for routine analysis with small samples, the Shapiro-Wilks and variance ratio tests would be switched off.

Finally, the unpaired t test yields the following results.

Unpaired t test ([ ] = corrected for unequal variances)		
Number of x-values	10	
Number of y-values	10	
Number of degrees of freedom	18	[18]
Unpaired t test statistic U	−0.6029	[-0.6029]
$P(t \geq U)$ (upper tail p)	0.7229	[0.7229]
$P(t \leq U)$ (lower tail p)	0.2771	[0.2771]
p for two tailed t test	0.5541	[0.5541]
Difference between means DM	−1.300	
Lower 95% confidence limit for DM	−5.830	[-5.830]
Upper 95% confidence limit for DM	3.230	[3.230]
Conclusion: *Consider accepting equality of means*		

Note that p values are given for either two-tailed or one-tailed testing, but this is just for convenience as users should have decided in advance which p value to accept, or in doubt would usually just rely on the two-tailed test.

For situations where there is doubt about variance equality, corrected values are given, but in this case where the sample size is small and the data are actually paired, correction is not required. Provided that deviations from normality and variance equality are fairly small, the unpaired t test has been claimed to be reassuringly robust. However this does not mean that using sample sizes much less than 10 is acceptable.

To explore these last points note that SIMFIT can calculate power as a function of sample size and, in addition, has extensive facilities to explore particular situations concerning sample size, deviations from normality, and variance inequality by simulation.

4.2.7 2-sample Paired t test

The paired t test should not be viewed as an alternative to the unpaired t test. It is used to see if it is reasonable to conclude that the mean of the differences between two sets of dependent observations is zero in the poulations, and the null hypothesis is based on the following assumptions.

- The pairwise differences are normally distributed
- The mean of the pairwise differences is zero
- The sample sizes are equal and greater than 1 (and preferably very much greater)
- The samples are pairwise dependent, i.e. highly correlated

For instance, the columns could be repeated observations of blood pressure on the same set of subjects, before, then after treatment.

The paired t test is equivalent to repeated analysis of variance (ANOVA) with just two columns if the samples are normally distributed with the same variance. Alteratively, as the test is only examining the hypothesis that a single sample of differences is normally distributed with zero mean, and variance estimated from the sample, it can be viewed as a one sample t test.

To be precise, the user has two samples (i.e. vectors X and Y) with n observations

$$X = (x_1, x_2, \ldots, x_n)$$
$$Y = (y_1, y_2, \ldots, y_n)$$

and wishes to test the null hypothesis that the samples have the same zero differences, against the alternative hypothesis that they are not equal, or possibly the one-sided alternatives. That is

$$H_0 : \; x_i = y_i$$
$$H_1 : \; x_i \neq y_i$$
$$H_2 : \; x_i > y_i$$
$$H_3 : \; x_i < y_i$$

and SimFIT provides all the information that is required to perform such tests.

It is important to emphasize the difference between a paired and an unpaired t test, so that it will not be mistakenly assumed that the choice between them is arbitrary. For instance, given two random variables X and Y, then the expectation and variance of the difference is as follows.

$$E(X - Y) = E(X) - E(Y)$$
$$V(X - Y) = V(X) + V(Y) - 2CV(X, Y)$$

When the samples are uncorrelated the covariance $CV(X, Y)$ would be zero, so that the variance of the difference only depends on the variance of X and Y. However, the paired hypothesis under consideration would be consistent with a strong positive correlation between X and Y, so that the covariance term would make the variance of the difference smaller.

In the case of the unpaired t test, the test statistic is

$$U = \frac{\bar{x} - \bar{y}}{\sqrt{s_P^2 \left(\frac{1}{m} + \frac{1}{n} \right)}},$$

which is t distributed with $m + n - 2$ degrees of freedom under the unpaired t test null hypothesis. Here $\bar{x}$ and $\bar{y}$ are sample means, and the pooled estimate of variance is expressed in terms of the independent sample variance estimates s_x^2 and s_y^2 as

$$s_p^2 = \frac{(m-1)s_x^2 + (n-1)s_y^2}{m+n-2}.$$

The paired t test is quite different. It uses the paired differences d_i, the mean difference $\hat{d}$, and the variance estimate for the differences s_d^2 to define the test statistic t_d defined as follows

$$\begin{aligned} d_i &= x_i - y_i \\ \bar{d} &= \sum_{i=1}^{n} d_i/n \\ s_d^2 &= \sum_{i=1}^{n} (d_i - \bar{d})^2/(n-1) \\ t_d &= \frac{\bar{d}}{\sqrt{s_d^2/n}} \end{aligned}$$

which is t distributed with $n - 1$ degrees of freedom under the paired t test null hypothesis. It therefore makes allowances for the strong correlation between the two samples.

From the main SIMFIT menu select [A/Z], then choose to open program **ttest**, and perform a paired t test to obtain the following results.

Paired *t* test	
Number of paired comparisons	10
Number of degrees of freedom	9
Paired t test statistic S	−0.9040
$P(t \geq S)$	0.8052
$P(t \leq S)$	0.1948
p for two tailed t test	0.3895
Mean of differences MD	−1.300
Lower 95% confidence limit for MD	−4.553
Upper 95% confidence limit for MD	1.953
Conclusion: *Consider accepting* $H_0 : MD = 0$	

Note that the paired t test only requires that the differences $x_i - y_i$ are normally distributed with zero mean, and the requirement for X and Y to be both normally distributed with the same variance, is not so strictly required. Nevertheless, as discussed for the unpaired t test, the Shapiro-Wilks and variance ratio tests are provided to explore the distribution of the observations if that is thought necessary, but they should only be used routinely for large samples, say $n > 50$.

4.2.8 2-sample Kolmogorov-Smirnov test

The Kolmogorov-Smirnov two sample nonparametric test can be used to examine if it is reasonable to conclude that two sets of independent observations have the same unknown distribution. It is based on a test statistic estimated from the largest differences between the two sample cumulative distributions.

To be precise, the user has two samples (i.e. vectors X and Y) with m and n observations

$$X = (x_1, x_2, \ldots, x_m)$$
$$Y = (y_1, y_2, \ldots, y_n)$$

and wishes to test the null hypothesis that the samples have the same distribution, against the alternative hypothesis that they are not equal, or possibly the one-sided alternatives. That is

$$H_0 : F(x) = G(y)$$
$$H_1 : F(x) \neq G(y)$$
$$H_2 : F(x) > G(y)$$
$$H_3 : F(x) < G(y)$$

and SIMFIT provides all the information that is required to perform such tests.

From the main SIMFIT menu select [A/Z], choose to open program **rstest**, then analyze the test files provided to obtain the following results.

Kolmogorov-Smirnov two sample test 1

X-data: Test file ttest.tf2: for paired t test with TTEST.TF3
Y-data: Test file ttest.tf3: for paired t test with TTEST.TF2
X sample size 10
Y sample size 10
H_0: $F(x)$ is equal to $G(y)$ (x and y are comparable) against
H_1: $F(x)$ not equal to $G(y)$ (x and y are not comparable)

D	0.2000
Z	0.08944
p	0.7869

H_2: $F(x) > G(y)$ (x tend to be smaller than y)

D	0.2000
Z	0.08944
p	0.3935

H_3: $F(x) < G(y)$ (x tend to be larger than y)

D	0.1000
Z	0.04472
p	0.4972

The test statistic uses D, which would be either the largest difference where $F(x) > G(y)$, D_m, the largest difference where $F(x) < G(y)$, D_n, or the maximum of these, $D_{mn} = \max(D_m, D_n)$, and is defined as

$$Z = \sqrt{\frac{mn}{m+n}} D.$$

This test is very weak unless the distributions are continuous and the sample sizes fairly large, so it is not surprising that the first example does not lead to a rejection of the null hypothesis. However, the next example compares data from two test files, `normal.tf1` with 50 pseudo random numbers from a normal distribution with $\mu = 0$ and $\sigma = 1$, and `normal.tf2` with 50 pseudo random numbers from a normal distribution with

$\mu = 1$ and $\sigma = 2$. The plot confirms very convincingly the table results, that identical distribution can be rejected in favor of the alternative hypothesis that $X < Y$, but not in favor of the alternative that $X > Y$.

Kolmogorov-Smirnov two sample test 2

X-data:Test file normal.tf1: mean = 0, standard deviation = 1
Y-data:Test file normal.tf2: mean - 1, standard deviation = 2
X sample size 50
Y sample size 50
H_0: $F(x)$ is equal to $G(y)$ (x and y are comparable) against
H_1: $F(x)$ not equal to $G(y)$ (x and y are not comparable)

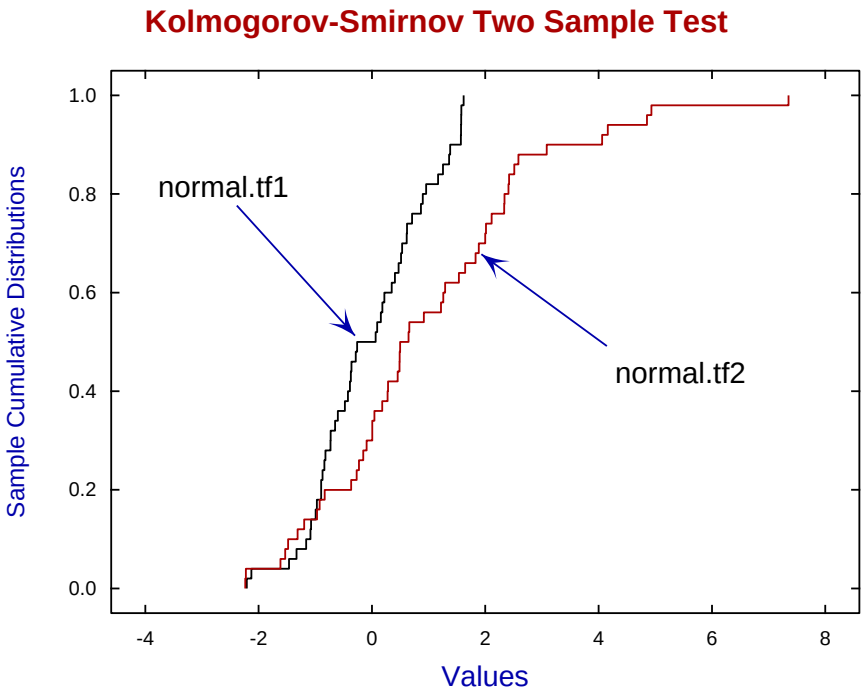
D	0.3600	
Z	0.07200	
p	0.0013	Reject H_0 at 1% significance level

H_2: $F(x) > G(y)$ (x tend to be smaller than y)

D	0.3600	
Z	0.07200	
p	0.0007	Reject H_0 at 1% significance level

H_3: $F(x) < G(y)$ (x tend to be larger than y)

D	0.06000
Z	0.01200
p	0.4989



4.2.9 2-sample Mann-Whitney U test

The Mann-Whitney U test is a sort of nonparametric equivalent of the unpaired t test that is used to examine the relative size of observations in two data sets, say X and Y without assumptions about the distributions.

To be precise, the user has two samples (i.e. vectors X and Y) with m and n observations

$$X = (x_1, x_2, \ldots, x_m)$$
$$Y = (y_1, y_2, \ldots, y_n)$$

where the ranks of the two sets of observations within a combined, i.e. pooled, data set can be consulted to see if is is reasonable to conclude that either

- data values in both samples are similar,
- data values in sample X tend to be smaller than those in sample Y, or
- data values in sample X tend to be larger than those in sample Y.

The test is weak unless large samples are used, and is further weakened by ties within the data, that is, multiple observations with the same value.

From the main SIMFIT menu select [A/Z], choose to open the SIMFIT nonparametric testing program **rstest**, then analyze the test files provided to obtain the following results.

Wilcoxon-Mann-Whitney U test

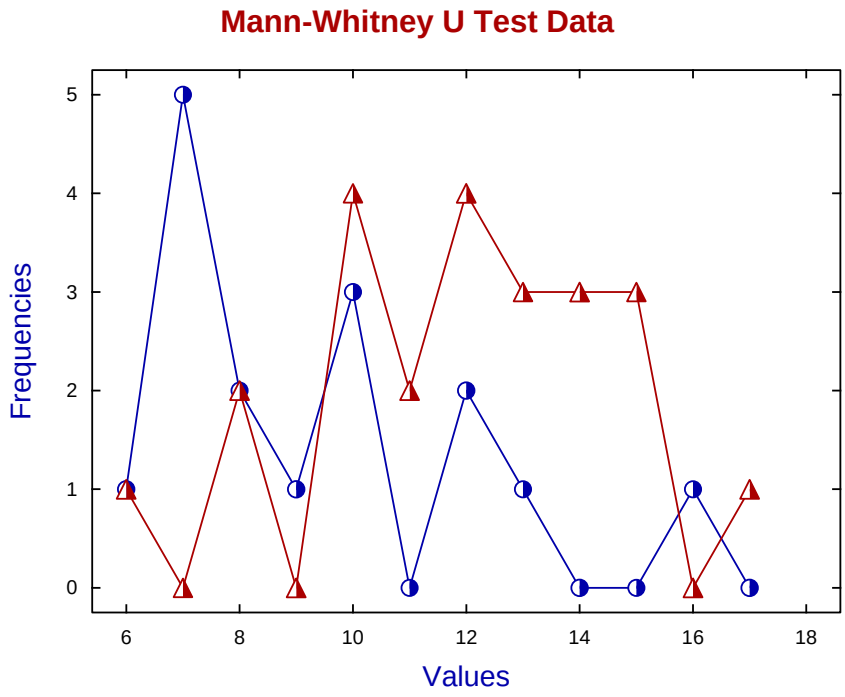
X-data: g08ahf.tf1 (Mann-Whitney U test)
Y-data: g08ahf.tf2 (Mann-Whitney U test)
X sample size 16
Y sample size 23
U 86.00
Z -2.804
H_0: $F(x)$ is equal to $G(y)$ (x and y are comparable)
as null hypothesis against the alternatives:-
H_1: $F(x)$ is not equal to $G(y)$ (x and y not comparable)
p 0.0050 Reject H_0 at 1% significance level
H_2: $F(x) > G(y)$ (x tend to be smaller than y)
p 0.0025 Reject H_0 at 1% significance level
H_3: $F(x) < G(y)$ (x tend to be larger than y)
p 0.9977

Note that U is the Mann-Whitney test statistic which is used to calculate an exact p value, while Z is an approximately normal test statistic and, using SIMFIT program **normal**, we find that $P(Z \leq -2.804) = 0.0025$.

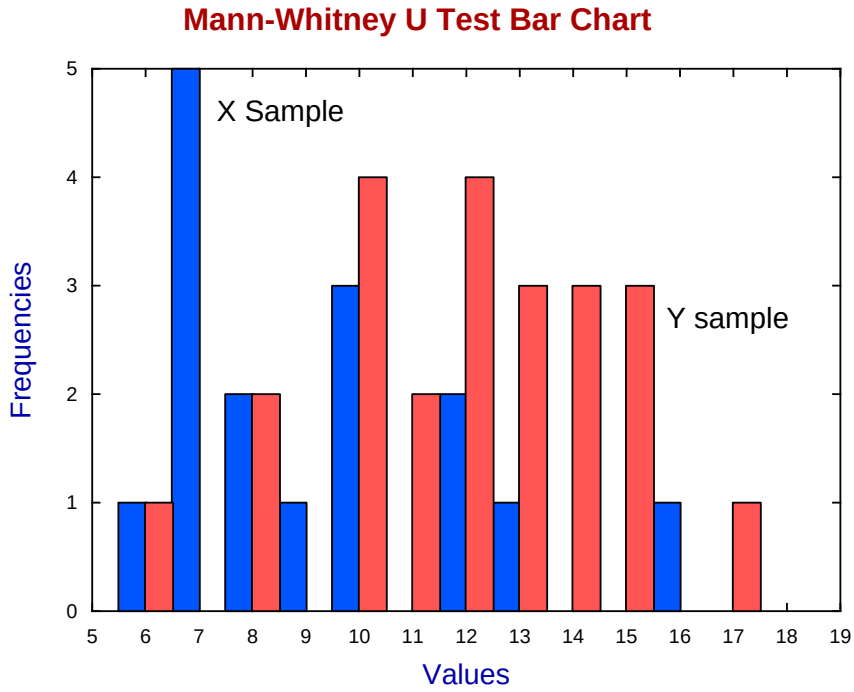
To understand how to interpret the meaning of the above two-tail and one-tail test statistics you can just look at a table of frequencies. This is easily constructed using SIMFIT program **editmt** to rearrange the samples into increasing order as follows, where bracketed values are frequencies.

X	6(1)	7(5)	8(2)	9(1)	10(3)	11(0)	12(2)	13(1)	14(0)	15(0)	16(1)	17(0)
Y	6(1)	7(0)	8(2)	9(0)	10(4)	11(2)	12(4)	13(3)	14(3)	15(3)	16(0)	17(1)

Alternatively, the frequencies can be plotted, as lines and symbols by first using SIMFIT program **makfil** to generate plotting files, followed by SIMFIT program **simplot** to create the following plot which emphasizes the test results, i.e. the most likely conclusion is that X-sample values tend to be smaller than the Y-sample values.



Using the built-in data editor in **simplot** to move *X* leftwards and *Y* rightwards to prevent overlapping, then replacing symbols by bars and suppressing the lines gives the next alternative way to plot the data.



4.2.10 2-sample Wilcoxon signed rank test

The Wilcoxon signed rank test is a sort of nonparametric equivalent of the paired t test that is used to examine the differences between matched observations in two data sets, say X and Y, just assuming a symmetrical, but unspecified, distribution for the paired differences.

To be precise, the user has two samples (i.e. vectors X with median X_{med}, and Y with median Y_{med}) with n observations, and specifies a hypothetical median for the paired differences, say D_{med}, that generates a vector of signed differences $d_i = x_i - y_i$ defined by

$$X = (x_1, x_2, \ldots, x_n)$$
$$Y = (y_1, y_2, \ldots, y_n)$$
$$D = (d_1, d_2, \ldots, d_n).$$

Users have to decide whether to include or discard zero differences, and whether to change the default median difference of $D_{med} = 0$ to $D_{med} = k$ for some hypothetical k, then analysis of the signed differences is performed to test if it is reasonable to conclude that either

- both samples have the same population median,
- the population median for sample X is smaller than that for sample Y, or
- the population median for sample X is larger than that for sample Y.

The test is weak unless large samples are used, and is further weakened by ties within the data, that is, multiple observations with the same value.

From the main SimFIT menu select `[A/Z]`, choose to open the SimFIT statistics program **simstat**, then the standard tests option to analyze the test files provided. Choosing a specified zero median and opting to suppress zero differences yields the following results.

Wilcoxon paired-sample signed-rank test

Zero differences suppressed, median test value = 0
X-data: test file `g08agf.tf1`
Y-data: test file `g08agf.tf2`
Size of data = 8, Number of values suppressed = 0
W 32.00
Z 1.890
H_0: X median = Y median
as null hypothesis against the alternatives:-
H_1: Medians differ
p 0.0547
H_2: X median < Y median
p 0.9805
H_3: X median > Y median
p 0.0273 Reject H_0 at 5% significance level

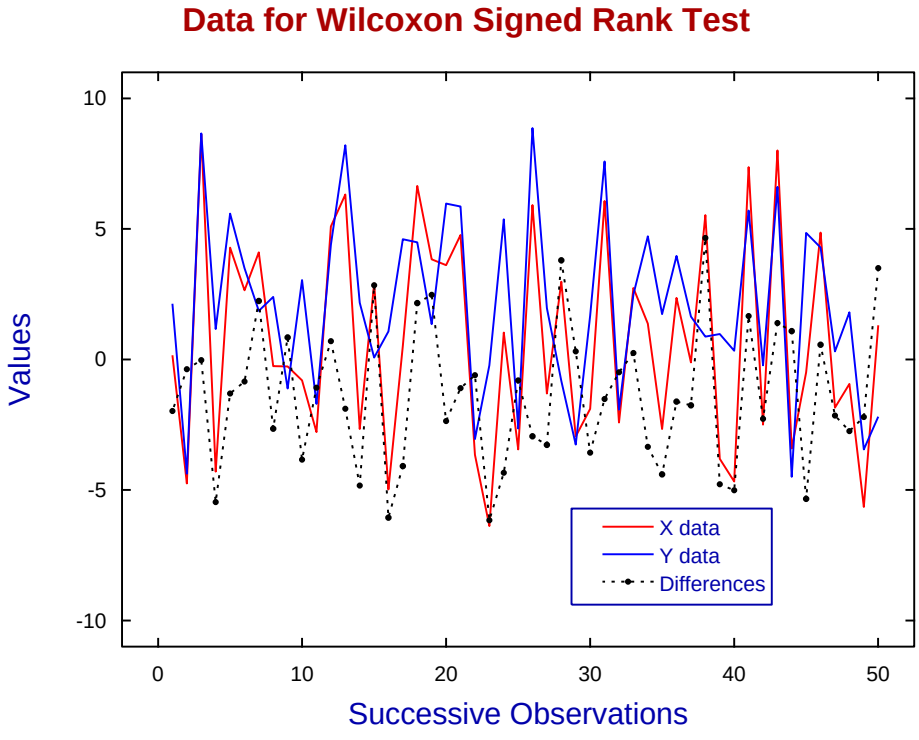
In this example there were no zero differences to reject, and here W is the signed ranks test statistic, while Z is an approximate normal test statistic. Using SimFIT program **normal**, we find that $P(Z \geq 1.89) = 0.0294$. In order to make the interpretation of this test as clear as possible, especially the effect of the value chosen for D_{med}, the results from sequential analysis of data in SimFIT test files `wilcoxon.tf1` and `wilcoxon.tf2` using two different values of $D_{med} = 0$ then $D_{med} = -2$ are displayed next.

Wilcoxon paired-sample signed-rank test 1 and test 2

Median test value = 0
X-data: test file `wilcoxon.tf1`
Y-data: test file `wilcoxon.tf2`
Size of data = 50, Number of values suppressed = 0
W 306.0
Z -3.195
H_0: X median = Y median
H_1: Medians differ
p 0.0011 Reject H_0 at 1% significance level
H_2: X median < Y median
p 0.0005 Reject H_0 at 1% significance level
H_3: X median > Y median
p 0.9995

Median test value = -2
W 783.0
Z 1.400
H_0: X median = Y median
H_1: Medians differ
p 0.1629
H_2: X median < Y median
p 0.9200
H_3: X median > Y median
p 0.0815

The following graph shows that, although X and Y do appear to be matched, the difference is mostly around -2, which explains why $D_{med} = 0$ is rejected, but $D_{med} = -2$ is not rejected, emphasizing the importance of choosing D_{med} sensibly on the outcome of this test.



4.2.11 Chi-square test on observed and expected frequencies

The chi-square test on observed and expected frequencies is based on forming a test statistic that, in the limit of a very large sample size, becomes asymptotic to a chi-square distribution, with the number of degrees of freedom dependent on the number of categories, and also on the number of parameters estimated from the sample.

To be precise, it is assumed that the user has counted the frequency of occurrence of k observations partitioned into n categories with O_i in category i, and also knows the frequencies E_i expected under the null hypothesis that the observations are consistent with the expected frequencies given by the assumed distribution. This allows the calculation of C defined as

$$C = \frac{(O_1 - E_1)^2}{E_1} + \frac{(O_2 - E_2)^2}{E_2} + \cdots + \frac{(O_n - E_n)^2}{E_n}$$

which has an approximate chi-square distribution with ν degrees of freedom given by

$$\nu = n - 1 - m$$

where $\nu \geq 2$, and m is the number of parameters estimated from the sample. The reason for subtracting $1 + m$ from n to get the degrees of freedom will be clear by considering the identity

$$\sum_{i=1}^{n} E_i = \sum_{i=1}^{n} O_i = k$$

which reduces the effective numbers of terms in the calculation of C to $n-1$. Similarly, every further equation of constraint can be considered to reduce by one the effective number of terms in the calculation of C.

It is usually recommended that the expected values are at least 5 and, if this cannot be realized, then categories could be combined until this condition is met. Alternatively, if the total number of observations is k as above, and the number of categories is not fixed by other considerations, then the number of bins n used to partition the data is sometimes suggested as

$$n \approx k^{0.4},$$

but obviously this all depends on the shape of the assumed distribution.

To illustrate this test consider the next table, which records the results from one hundred observations on the number of heads resulting from tossing five different coins. Clearly there are six categories, as the number of heads per toss of the five coins can only be $0, 1, 2, 3, 4$, or 5, but note that $100^{0.4} \approx 6$ anyway in this case.

Number of Heads	0	1	2	3	4	5
Observed	3	16	36	32	11	2
Expected	4.0	17.9	32.6	29.6	13.5	2.4

There were 238 heads in all from the total of 500 tosses, so the expected frequencies were calculated using a binomial distribution with binomial $N = 5$ and estimated parameter $\hat{p} = 238/500 = 0.476$, and therefore 4 degrees of freedom.

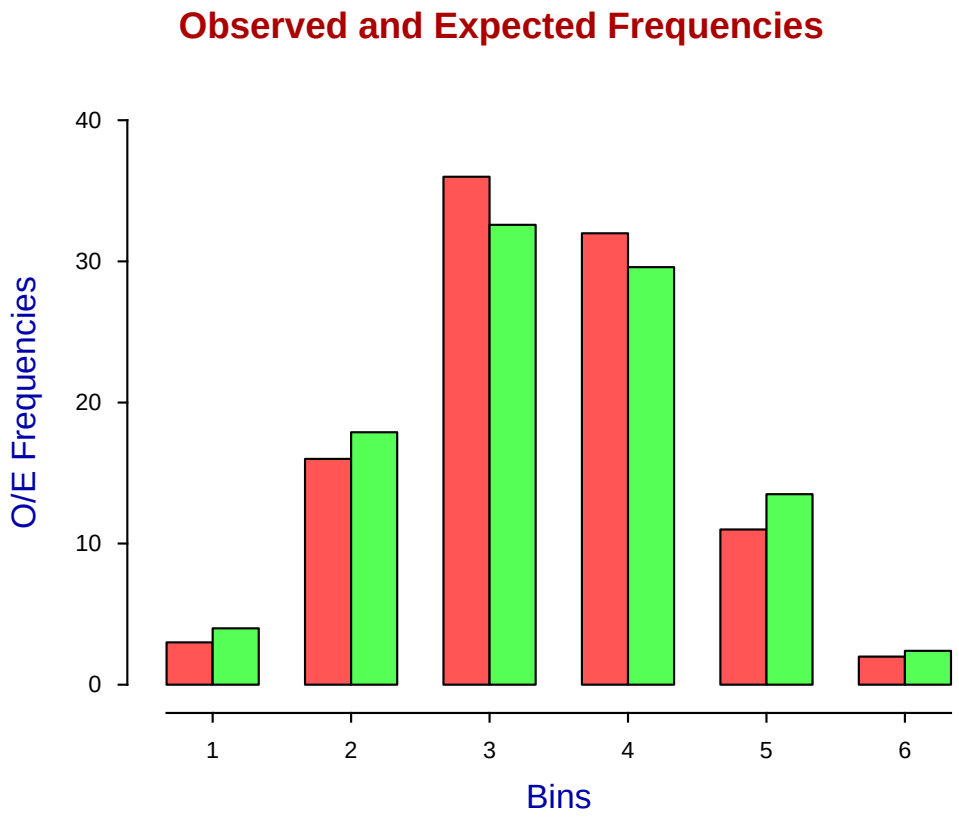
Choose `[A/Z]` from the main SimFIT menu, open program **chisqd**, select chi-square test on observed and expected frequencies, then analyze the above data contained in the test files `chisqd.tf2` and `chisqd.tf3`, with one parameter estimated from the sample to get these results.

Number of partitions (bins)	6	
Number of degrees of freedom	4	
Chi-square test statistic C	1.531	
$P(\chi^2 \geq C)$	0.8212	Consider accepting H0
Upper tail 5% critical point	9.488	
Upper tail 1% critical point	13.28	

SimFIT first displays a warning that the expected frequencies for 0 and 5 heads are below 5, and so these two categories could be combined if it was thought necessary. However, in this case the p value of 0.8212 is much larger than 0.05, so the conclusion is that the null hypothesis of a binomial distribution with parameters $N = 5$, and $p = 0.476$ cannot be rejected.

Note that SimFIT also lists the 1% and 5% upper tail critical points, as this is how the test results were analyzed in the past by looking up tables of critical points, before the availability of computers made this unnecessary.

The most widely used technique to display the agreement between the observed and expected frequencies is a bar chart, as in the next figure.



4.2.12 Contingency table analysis

A contingency table is an array of nonnegative frequencies with n rows and m columns, such as this table contained in SimFIT test file `chisqd.tf4`, for 15 observations carried out on two populations to test for equal probabilities of success.

	Success	Failure	
Sample 1	3	3	6
Sample 2	7	2	9
	10	5	15

Here, the cell frequencies are $(3, 3, 7, 2)$, the sum of row frequencies known as row marginals are $(6, 9)$, the sum of column frequencies known as column marginals are $(10, 5)$, and obviously the row and column marginals must separately both add up to the total number of frequencies (15).

To be precise, in the general case there will be frequencies f_{ij} where $i = 1, 2, \ldots, n$, and $j = 1, 2, \ldots, m$, and it is wished to test for homogeneity, i.e. independence, or no association between the variables, which can be stated as the null hypothesis

$$H_0 : \mu_{ij} = \mu_{i.}\mu_{.j}, \text{ for } i = 1, 2, \ldots, n, \text{ and } j = 1, 2, \ldots, m$$

where each cell probability μ_{ij} is completely determined by the corresponding row marginal $\mu_{i.}$, and the column marginal $\mu_{.j}$. To examine a given data set SimFIT provides the following three alternatives.

1. **The chi-square test.**
   This is the easiest to perform and and interpret, and is the test most generally used. However, it must be emphasized that the test statistic is only asymptotically distributed as chi-square with $(n-1)(m-1)$ degrees of freedom in the limit for large samples. Where there are small frequencies the option to combine cells should be considered, and note that the Yate's continuity correction may be used where appropriate.

2. **The Fisher exact test.**
   This is very powerful and widely used, but sometimes suffers from being difficult to interpret with large samples, which also may lead to computational problems.

3. **The loglinear contingency table analysis.**
   This uses general linear modeling assuming a Poisson error distribution and log link, but it does require some expertise on the part of users.

Choose `[A/Z]` from the main SimFIT menu, then open SimFIT program **chisqd**.

Chi-square contingency table test

For all tables, SimFIT calculates a chi-square test statistic C from the observed frequencies f_{ij}, and expected frequencies e_{ij}, and also a likelihood ratio test statistic L defined in terms of the expected values e_{ij} and marginals $f_{i.}$ and $f_{.j}$ as follows

$$\begin{aligned} e_{ij} &= f_{i.}f_{.j}/N \\ C &= \sum_{i=1}^{n}\sum_{j=1}^{m}\frac{(f_{ij}-e_{ij})^2}{e_{ij}} \\ L &= -2\log\lambda \\ &= 2\sum_{i=1}^{n}\sum_{j=1}^{m} f_{ij}\log(f_{ij}/e_{ij}) \end{aligned}$$

It is often recommended to combine cells where the expected values are small, say $e_{ij} < 0.5$, and this facility is provided.

Select chi-square contingency table analysis, then analyze the above data which leads to calculation of the approximate chi-square test statistic with the Yate's continuity correction

$$C = \frac{N(|f_{11}f_{22} - f_{12}f_{21}| - N/2)^2}{r_1 r_2 c_1 c_2}$$

for this 2 by 2 contingency table, where N is the sum of frequencies f_{ij}, r_i are the row marginals, and c_j are the column marginals, leading to the following results, which do not suggest rejecting H_0.

Number of rows	2
Number of columns	2
chi-square test statistic C	0.3125
Number of degrees of freedom	1
$P(\chi^2 \geq C)$	0.5762
Upper tail 5% point	3.841
Upper tail 1% point	6.635
$L = -2\log(\lambda)$	1.243
$P(\chi^2 \geq L)$	0.2649

Fisher exact test

For 2 by 2 contingency tables, and $N \leq 100$, tables like the following are also displayed.

Observed		Rearranged so r_1 = smallest marginal, $c_2 \geq c_1$	
3	3	3	2
7	2	3	7

$p(r)$	p for $f_{11} = r$ after rearranging and adjusting	
$p(0)$	0.041958	
$p(1)$	0.251748	
$p(2)$	0.419580	
$p(3)$	0.239760	$p(*)$, observed frequencies
$p(4)$	0.044955	
$p(5)$	0.001998	
P_sums, 1-tail and 2-tail test statistics		
P_sum1	0.041958	sum of $p(r) \leq p(*)$ for $r < 3$
P_sum2	0.953047	sum of all $p(r)$ for $r \leq 3$
P_sum3	0.286713	sum of all $p(r)$ for $r \geq 3$
P_sum4	0.046953	sum of $p(r) \leq p(*)$ for $r > 3$
P_sum5	1.000000	P_sum2 + P_sum4
P_sum6	0.328671	P_sum1 + P_sum3

For convenience, this test starts by rearranging the data table until r_1 is the smallest marginal and $c_2 \geq c_1$. Then all hypothetical tables that are possible with the same marginals are considered, but now for $r = f_{11}$ for $r = 0, 1, \ldots, r_1$ as follows, where the observed frequencies are indicated by stars (*).

0	5	1	4	2	3	*3	*2	4	1	5	0
6	4	5	5	4	6	*3	*7	2	8	1	9

Assuming the null hypothesis, the probabilities $p(r)$ for tables with $f_{11} = r$ are then calculated for a hypergeometric distribution using

$$p(r) = \frac{r_1! r_2! c_1! c_2!}{f_{11}! f_{21}! f_{12}! f_{22}! N!}$$

With the tables under consideration it is clear that, had the outcome been as for the hypothetical tables indicated by $p(0)$, $p(4)$, or $p(5)$ then the possibility of rejecting H_0 would have to be considered. However, the current data $p(3)$, indicated by $p(*)$ would be accepted, as for the chi-square test on the same data. With less obvious results, various one-tailed and two-tailed tests can be based on considering probabilities for more extreme contingency tables, or sums of such probabilities. As an example consider the following data

	Boys	Girls	
Left-handed	6 (18%)	12 (22%)	18
Right-handed	28 (82%)	24 (67%)	52
	34	36	70

and possible hypotheses for this sample
H_0: left-handedness is not less common in boys than girls
H_A: left-handedness is less common in boys than girls.

$p(r)$	p for $f_{11} = r$ after rearranging and adjusting	
$p(0)$	0.000000	
$p(1)$	0.000013	
$p(2)$	0.000177	
$p(3)$	0.001436	
$p(4)$	0.007590	
$p(5)$	0.027720	
$p(6)$	0.072572	$p(*)$, observed frequencies
$p(7)$	0.139338	
$p(8)$	0.198959	
$p(9)$	0.212877	
$p(10)$	0.171062	
$p(11)$	0.102959	
$p(12)$	0.046046	
$p(13)$	0.015082	
$p(14)$	0.003535	
$p(15)$	0.000571	
$p(16)$	0.000060	
$p(17)$	0.000004	
$p(18)$	0.000000	
P_Sums, for 1-tail and 2-tail test statistics		
P_sum1	0.036936	sum of $p(r) \leq p(*)$ for $r < 6$
P_sum2	0.109508	sum of all $p(r)$ for $r \leq 6$ (one-tailed p)
P_sum3	0.963064	sum of all $p(r)$ for $r \geq 6$
P_sum4	0.065297	sum of $p(r) \leq p(*)$ for $r > 6$
P_sum5	0.174805	P_sum2 + P_sum4
P_sum6	1.000000	P_sum1 + P_sum3

Adding up the probabilities for the observed table $p(6) = p(*)$ and all the possible tables more extreme than this that would favor H_A against H_0 we see that the appropriate one-tailed p value is

$$p(0) + p(1) + p(2) + p(3) + p(4) + p(5) + p(6) = 0.109508$$

and so, for this sample with $\alpha = 0.05$ we would not consider rejecting H_0.

loglinear contingency table test

The full details for this test will be found in the SIMFIT reference manual, but meaningful interpretation of the results is possible without detailed understanding. Essentially, a statistical model is constructed for the contingency table with the following characteristics.

- Best-fit theoretical cell frequencies are calculated using a loglinear model.
- The parameter estimates are displayed along with standard errors and p values.
- Predicted cell frequencies are then compared with the observed data to generate differences, residuals, and leverages.
- The deviance is calculated, and the chi-square significance reported.

Here are the results for the SIMFIT test data set.

Log-linear contingency table analysis
Data: Test file `chisqd.tf4`
number of rows = 2, number of columns = 2
Deviance (D) = 1.243, degrees of freedom = 1
$P(\chi^2 \geq D) = 0.2649$

Parameter	Estimate	Std.Err.	Lower 95%	Upper 95%	p
Constant	1.792	0.380	-3.04	6.62	0.1330 ***
Row 1	-0.4055	0.527	-7.10	6.29	0.5823 ***
Row 2	0.4055	0.527	-6.29	7.10	0.5823 ***
Col 1	-0.6931	0.547	-7.65	6.26	0.4254 ***
Col 2	0.6931	0.547	-6.26	7.65	0.4254 ***

Data	Model	Delta	Residual	Leverage
3	4	-1	-0.5234	0.7997
3	2	1	0.6579	0.6005
7	6	1	0.3976	0.8664
2	3	-1	-0.6149	0.7335

The model that is assumed expresses the theoretical cell probability μ_{ij} as a constant θ, plus row parameters α_i, column parameters β_j, and mixed row-column parameters γ_{ij} in the following way

$$\log \mu_{ij} = \theta + \alpha_i + \beta_j + \gamma_{ij}$$

where

$$\sum_{i=1}^{n} \alpha_i = \sum_{j=1}^{m} \beta_j = 0.$$

The null hypothesis of homogeneity, that is $\mu_{ij} = \mu_{i.}\mu_{.j}$, can then be stated as

$$H_0 : \ \gamma_{ij} = 0 \text{ for } i = 1, 2, \ldots, n, \text{ and } j = 1, 2, \ldots, m$$

and the deviance measures the extent to which the hypothesis of homogeneity can be supported. Note that the purpose of starred parameter estimates is simply to warn users about suspiciously large ratios of standard errors to parameter estimates, i.e. where $p \geq 0.05$. Also, with large contingency tables, the ability to plot the residuals in a variety of ways to visualize goodness of fit is provided.

As before, this test provides no support for rejecting the null hypothesis of homogeneity with these data.

4.2.13 McNemar test

The McNemar test is used to analyze paired observations of a dichotomous variable, i.e. where there can only be one of two possible values such as: success/failure, +/-, 0/1, etc. and it is of interest to examine if the paired values are associated or are independent.

To be precise, consider the possible outcome from testing fifty specimens of sputum cultured on two different culture media, A and B, with the intention of detecting a particular bacterium. The four possible outcomes were as follows.

Type	Medium A	Medium B	Number
Both	+	+	20
A only	+	-	12
B only	-	+	2
Neither	-	-	16

These data can be arranged as a 2 by 2 contingency table, such as this table contained in SimFIT test file `mcnemar.tf1`.

	B +	B -	Total
A +	20	12	32
A -	2	16	18
Total	22	28	50

Here, the cell frequencies are $(f_{11} = 20, f_{12} = 12, f_{21} = 2, f_{22} = 16)$, the sum of row frequencies known as row marginals are $(32, 18)$, the sum of column frequencies known as column marginals are $(22, 28)$, and obviously the row and column marginals must separately both add up to the total number of frequencies $(n = 50)$.

From the main SimFIT menu choose [Statistics] then [Standard tests] and analyze the above data using the McNemar option to get the following table.

McNemar test 1

H_0: Expected value of [(f(1,2) - (f(2,1))/n] = 0		
Title: Data for 2 by 2 McNemar test		
Number of rows/columns	2	
Chi-square test statistic C	5.786	
Number of degrees of freedom	1	
$P(\chi^2 \geq C)$	0.0162	Reject H_0 at 5% level
Upper tail 5% point	3.841	
Upper tail 1% point	6.635	

Continuity correction used in chi-square

The frequencies f_{ij} in this table are analyzed by calculating the χ^2 test statistic given by

$$\chi^2 = \frac{(|f_{12} - f_{21}| - 1)^2}{f_{12} + f_{21}}.$$

which has an approximate chi-square distribution with 1 degree of freedom. The outcome emphasizes the obvious fact that culture medium A is more effective than culture medium B.

Note that this test does not perform so well with small frequencies and, in particular, if $r = 2$ and

$$f_{12} + f_{21} \leq 20$$

a warning will be displayed. In such cases it may be preferable to do a binomial test using $N = f_{12} + f_{21}$ then $X = f_{12}$ or $X = f_{21}$ to check if $\hat{p} = X/N$ is consistent with a binomial distribution with parameters N and $p = 0.5$. Since in the 2 by 2 case the McNemar test is equivalent to testing if two sample estimates for a binomial probability parameter differ significantly, we can use SimFiT to calculate exact 95% confidence limits as follows

$$\begin{aligned} \text{For } \ 2/14: &\quad 0.0178 \le \hat{p} = 0.1429 \le 0.4281 \\ \text{For } 12/14: &\quad 0.5719 \le \hat{p} = 0.8571 \le 0.9822 \end{aligned}$$

which convincingly demonstrates the superiority of culture medium A over culture medium B.

To explain the logic behind this analysis, note that the overall proportion of successes with medium A is $(f_{11} + f_{12})/n$, while the overall proportion of successes with medium B is $(f_{11} + f_{21})/n$, so that the difference between these estimates for the probability of success depends only on $f_{12} - f_{21}$, and the null hypothesis for such a 2 by 2 table can be expressed as expectations in several equivalent ways without using the diagonal frequencies f_{ii} except in the sample size n, such as

$$\begin{aligned} H_0 &: E\left(\frac{f_{12} - f_{21}}{n}\right) = 0, \text{ or} \\ H_0 &: E\left(\frac{f_{12}}{f_{21}}\right) = 1. \end{aligned}$$

Note that it is important that tables for the McNemar test are set up correctly to reflect the pairwise nature of the data, so an additional example is given using data in test file `mcnemar.tf2` for the case where medication A was applied to one arm and medication B to the other arm with subjects suffering from a rash on both arms.

	B worked	B failed	Total
A worked	11	6	17
A failed	10	24	34
Total	21	30	50

Chi-square test statistic C	0.5625	
Number of degrees of freedom	1	
$P(\chi^2 \ge C)$	0.4533	Consider accepting H_0
Upper tail 5% point	3.841	
Upper tail 1% point	6.635	

The outcome is that there is no evidence to support a significant difference between medications A and B in this experiment.

More generally, for larger r by r tables with identical paired row and column categorical variables, the continuity correction is not used, and the appropriate test statistic is

$$\chi^2 = \sum_{i=1}^{r} \sum_{j>i} \frac{(f_{ij} - f_{ji})^2}{f_{ij} + f_{ji}}$$

with $r(r-1)/2$ degrees of freedom. Unlike the normal contingency table analysis where the null hypothesis is independence of rows and columns, with this test there is intentional association between rows and columns. The test statistic does not use the diagonal frequencies f_{ii} and is testing whether the upper right corner of the table is symmetrical with the lower left corner. The SimFiT test file `mcnemar.tf3` contains data for a such a 3 by 3 McNemar table.

4.2.14 Cochran Q test

The Cochran Q test is used to analyze randomized block or repeated-measures observations of a dichotomous variable, i.e. where there can only be one of two possible values such as: success/failure, +/-, 0/1, etc. and it is of interest to examine if successive measurements differ significantly.

To be precise, consider the possible outcome from testing eight people exposed to mosquito attacks with five different types of clothing as follows, where a 1 indicates attacked by mosquitos and a 0 indicates freedom from attack.

Blocks	Groups (Clothing Type)				
(Subjects)	Light-loose	Light-tight	Dark-long	Dark-short	None
1	0	0	0	1	0
2	1	1	1	1	1
3	0	0	0	1	1
4	1	1	0	1	0
5	0	1	1	1	1
6	0	1	0	0	1
7	0	0	1	1	1
8	0	0	1	1	0

The results for blocks (e.g., subjects) are in rows from 1 to n of a matrix while the attributes, which can be either 0 or 1, are in groups, that is, columns 1 to m. So, with n blocks, m groups, G_i as the number of attributes equal to 1 in group i, and B_j as the number of attributes equal to 1 in block j, then the statistic Q is calculated, where

$$Q = \frac{(m-1)\left[\sum_{i=1}^{m} G_i^2 - \frac{1}{m}\left(\sum_{i=1}^{m} G_i\right)^2\right]}{\sum_{j=1}^{n} B_j - \frac{1}{m}\sum_{j=1}^{n} B_j^2}$$

and Q is distributed as approximately chi-square with $m-1$ degrees of freedom. It is recommended that m should be at least 4 and mn should be at least 24 for the approximation to be satisfactory.

For example, open the main SIMFIT menu, select [Statistics], then [Standard tests], and perform the Cochran Q test on the test file `cochranq.tf1` to obtain the results shown below.

Number of blocks (rows)	7	Rows suppressed: 1 (all 0 or all 1)
Number of groups (columns)	5	Columns suppressed: 1 (not data)
Cochran Q value	6.947	
$P(\chi^2 \geq Q)$	0.1387	
95% chi-square point	9.488	
99% chi-square point	13.28	

Clearly, the test provides no reason to reject the null hypothesis that the proportion of humans in this study attacked by mosquitos is the same for all clothing types. Also, note the following facts about the SIMFIT file format for a Cochran Q test.

1. Rows containing only a 0 or only a 1, like row 2 above, can be included in the data file but they do not contribute to the analysis.

2. An extra column of successive integers, which must be in order from 1 to n, can be included as a first column if required to help you identify the subjects in the results file.

4.2.15 Binomial test

This procedure, based on the binomial distribution, is used with dichotomous data, i.e., where an experiment has only two possible outcomes and it is wished to test H_0: binomial $p = p_0$ for some $0 \le p_0 \le 1$. For instance, to test if consecutive outcomes are independent with the same probability, i.e. are Bernoulli trials.

To be precise, you input the number of successes, k, the number of Bernoulli trials, N, and the supposed probability of success, p_0, then SIMFIT calculates the probabilities associated with k, N, p_0, and $l = N - k$, including the estimated probability parameter $\hat{p}$ with 95% confidence limits, and the two-tail binomial test statistic. The probabilities for X equal to the number of successes, which can be used for upper-tail, lower-tail, or two-tail testing are

$$\hat{p} = k/N$$

$$P(X = k) = \binom{N}{k} p^k (1-p)^{N-k}$$

$$P(X > k) = \sum_{i=k+1}^{N} \binom{N}{i} p^i (1-p)^{N-i}$$

$$P(X < k) = \sum_{i=0}^{k-1} \binom{N}{i} p^i (1-p)^{N-i}$$

$$P(X = l) = \binom{N}{l} p^l (1-p)^{N-l}$$

$$P(X > l) = \sum_{i=l+1}^{N} \binom{N}{i} p^i (1-p)^{N-i}$$

$$P(X < l) = \sum_{i=0}^{l-1} \binom{N}{i} p^i (1-p)^{N-i}$$

$$P(\text{two tail}) = \min(P(X \ge k), P(X \le k)) + \min(P(X \ge l), P(X \le l)).$$

Open the SIMFIT main menu then choose [Statistics] followed by [Standard tests] and run the binomial test option using the default parameters to obtain this table of results.

Binomial test analysis 1

Successes k	5	
Trials N	10	
$l = N - k$	5	
p-theory	0.50000	
p-estimate	0.50000	95% confidence limits = 0.18709, 0.81291
$P(X > k)$	0.37695	
$P(X < k)$	0.37695	
$P(X = k)$	0.24609	
$P(X \ge k)$	0.62305	
$P(X \le k)$	0.62305	
$P(X > l)$	0.37695	
$P(X < l)$	0.37695	
$P(X = l)$	0.24609	
$P(X \ge l)$	0.62305	
$P(X \le l)$	0.62305	
Two tail binomial test statistic = 1.00000		

From this table it is obvious that 5 successes in 10 trials is perfectly consistent with a binomial distribution having $N = 10$ and $p = 0.5$. However, consider the results when the number of Bernoulli trials is reduced to 5, where intuition might suggest that five successive heads in coin tossing would suggest a two-headed coin.

Binomial test analysis 2		
Successes k	5	
Trials N	5	
$l = N - k$	0	
p-theory	0.50000	
p-estimate	1.00000	95% confidence limits = 0.47818, 1.0000
$P(X > k)$	0.00000	
$P(X < k)$	0.96875	
$P(X = k)$	0.03125	
$P(X \geq k)$	0.03125	
$P(X \leq k)$	1.00000	
$P(X > l)$	0.96875	
$P(X < l)$	0.00000	
$P(X = l)$	0.03125	
$P(X \geq l)$	1.00000	
$P(X \leq l)$	0.03125	
Two tail binomial test statistic = 0.06250		

This shows, for example, that the probability of obtaining five successes (or alternatively five failures) in an experiment with equiprobable outcome would not lead to rejection of $H_0 : p = 0.5$ in a two tail test. Note, for instance, that the exact confidence limits for the estimated probability include 0.5. Many life scientists when asked what is the minimal sample size to be used in an experiment, e.g. the number of experimental animals in a trial, would use a minimum of six, since the null hypothesis of no effect would never be rejected with a sample size of five.

The next table illustrates that an experiment with six consecutive successes (or failures) would indicate that the 95% confidence region for the parameter p does not include 0.5, and would provide grounds for rejecting the null hypothesis H_0: The trials are all independent with $p = 0.5$.

Binomial test analysis 3		
Successes k	6	
Trials N	6	
$l = N - k$	0	
p-theory	0.50000	
p-estimate	1.00000	95% confidence limits = 0.54074, 1.0000
$P(X > k)$	0.00000	
$P(X < k)$	0.98438	
$P(X = k)$	0.01563	
$P(X \geq k)$	0.01563	
$P(X \leq k)$	1.00000	
$P(X > l)$	0.98438	
$P(X < l)$	0.00000	
$P(X = l)$	0.01563	
$P(X \geq l)$	1.00000	
$P(X \leq l)$	0.01563	
Two tail binomial test statistic = 0.03125, Reject H_0 at 5% significance level		

4.2.16 Sign test

This procedure, which is based on the binomial distribution, but assuming the special case $p = 0.5$, is used with dichotomous data, i.e., where an experiment has only two possible outcomes, and it is wished to test if the outcomes, say success or failure, are equally likely. The test is usually described in terms of positive signs (+) or negative signs (-) but, as it is only concerned with a succession of observations that can only be one of two types and does not necessarily involve any sort of measurement scale, it has much wider application. Unfortunately the test does not take into account the order of positive and negative signs and would not differentiate between the patterns $+-+-+-+-+-$ and $+++++-----$, so the test is rather weak and large samples, say greater than 20, are usually recommended. The run test does take the order of occurrence into account, and should be used where order in the sequence of signs has significance.

Open the SIMFIT main menu, choose [Statistics] then [Standard tests] and run the sign test option. This can be used to input numbers of positive and negative signs and, using the default options for number of positive signs $m = 5$, and negative signs $n = 5$, the next results are obtained.

Sign test analysis 1, $m + n = 10$		
$P(+ve = m)$	0.24609	$m = 5$
$P(+ve > m)$	0.37695	
$P(+ve < m)$	0.37695	
$P(+ve \geq m)$	0.62305	
$P(+ve \leq m)$	0.62305	
$P(-ve = n)$	0.24609	$n = 5$
$P(-ve < n)$	0.37695	
$P(-ve > n)$	0.37695	
$P(-ve \leq n)$	0.62305	
$P(-ve \geq n)$	0.62305	
Two tail sign test statistic = 1.00000		

The test could be used, for instance, to find out how many consecutive successes you would have to observe before the likelihood of an equiprobable outcome would be questioned. From these five successes and five failures it is quite clear that such an outcome is perfectly consistent with the null hypothesis $H_0 : p = 0.5$,

On the other hand, the case with $m = 9$, and $n = 1$, summarized in the next table is obviously more extreme.

Sign test analysis 2, $m + n = 10$		
$P(+ve = m)$	0.00977	$m = 9$
$P(+ve > m)$	0.00098	
$P(+ve < m)$	0.98926	
$P(+ve \geq m)$	0.01074	
$P(+ve \leq m)$	0.99902	
$P(-ve = n)$	0.00977	$n = 1$
$P(-ve < n)$	0.00098	
$P(-ve > n)$	0.98926	
$P(-ve \leq n)$	0.01074	
$P(-ve \geq n)$	0.99902	
Two tail sign test statistic = 0.02148, Reject H_0 at 5% significance level		

Clearly, nine outcomes of one kind but only one of the opposite kind, suggests rejection of the null hypothesis that both outcomes are equally likely irrespective of the order of occurrence of the observations.

4.2.17 Run test

SimFIT provides two types of run test.

1. **Run test on successive signs.**
   This is used to analyze residuals from regression or, in fact, any succession of observations of a variable which can only have one of two values, say positive (+ve) or negative (-ve).

2. **Runs up or down test.**
   This is mainly used to test sequences of numbers for significant correlation, as in examining the performance of a random number generator, and is discussed elsewhere.

To be precise, the run test considered in this article is based on an application of the binomial distribution, and is used when the sequence of successes and failures (presumed in the null hypothesis to be equally likely) is of interest, not just the overall proportions. To understand the definition of runs as dealt with by this test, just consider the sequence

$$++\,+--++---+-$$

or alternatively

$$111001100010$$

which has twelve items with six runs, as will be clear by adding brackets like this

$$(aaa)(bb)(aa)(bbb)(a)(b).$$

Open the SimFIT main menu, select [A/Z], then choose to run SimFIT program **rstest**. This provides three quite separate ways to perform a run test as follows.

1. **Direct input of parameters.**
   You simply type in the number of negative and positive signs observed and the associated runs to get an analysis like the following results for the default values of 10 positives, 10 negatives, and 10 runs.

Run and sign test 1	
Number of -ve values	10
Number of +ve values	10
Number of runs	10
Probability(runs ≤ observed; given number of +ve, -ve values)	0.41407
Critical number for 1% significance level	5
Critical number for 5% significance level	6
Probability(runs ≤ observed; given number of non zero values)	0.50000
Probability(signs ≤ observed) (Two tail sign test statistic)	1.00000

   Note that when defining parameters in this way you will be warned if there is an inconsistency in the data supplied.

2. **Direct input of residuals.**
   Using this method a file containing the residuals is input to give results like the following for the default test file `rstest.tf1`.

Run and sign test 2	
Number of -ve values	29
Number of +ve values	21
Number of runs	21
Probability(runs ≤ observed; given number of +ve, -ve values)	0.12867
Critical number for 1% significance level	17
Critical number for 5% significance level	19
Probability(runs ≤ observed; given number of non zero values)	0.12643
Probability(signs ≤ observed) (Two tail sign test statistic)	0.32224

3. **Direct input of two sequences of values.**
   Another way is to input two files containing numerical values, then allow SimFIT to calculate the residuals as for the next results with the default test files `rstest.tf1` and `normal.tf1`.

Run and sign test 3	
Number of -ve values	24
Number of +ve values	26
Number of runs	26
Probability(runs ≤ observed; given number. of +ve, -ve values)	0.56142
Critical number for 1% significance level	17
Critical number for 5% significance level	20
Probability(runs ≤ observed; given number of non zero values)	0.61228
Probability(signs ≤ observed) (Two tail sign test statistic)	0.88772

To emphasize the advantages of the run test over the sign test, consider a situation that a sample of ten new born babies in a hospital ward consisted of five boys and five girls? That would appear reasonable. However, what if all the boys were born first in the morning, then all the girls in the afternoon, that is, two runs? Clearly the sign test alone does not help, but the next table would confirm what most would believe intuitively: the event may not represent random sampling but could suggest the operation of other factors. In this way the run test, particularly when conditional upon the number of successes and failures, is using information from the sequence of outcomes and is therefore more powerful than the sign test alone.

Run and sign test 4		
Number of -ve values	5	
Number of +ve values	5	
Number of runs	2	
Probability(runs ≤ observed; given number of +ve, -ve values)	0.00794	Reject H_0 at 1% significance level
Critical number for 1% sig. level	2	
Critical number for 5% sig. level	3	
Probability(runs ≤ observed; given number of non zero values)	0.01953	Reject H_0 at 5% significance level
Probability(signs ≤ observed) (Two tail sign test statistic)	1.00000	

Note that every time SimFIT performs curve fitting it gives an analysis of goodness of fit which includes the run test to draw attention to bias in the fit resulting in too few runs caused by sections where the best-fit curve lies appreciably to one side of the data. Actually residuals from regression are not exactly normally

distributed even if the experimental errors are and the model fitted is correct, due to dependence introduced by the estimation of parameters. However, if the total numbers of data points fitted is much larger than the number of parameters estimated, this complication will not be very important.

Note that the run test in the analysis of residuals depends on a natural ordering, for instance, when the residuals are arranged to correspond to the order of a single independent variable. This is not possible if there are replicates, or several independent variables so, to use the run test in such circumstances, the residuals must be arranged in some meaningful sequence, such as the order in time of the observation, otherwise arbitrary results can be obtained by rearranging the order of residuals. The formulas used by SIMFIT to calculate the statistics in the run test analysis are presented next.

Given the numbers of positive and negative residuals, the probability of any possible number of runs can be calculated by enumerating all possible arrangements. For instance, the random number of runs R given m positive and n negative residuals (redefining if necessary so that $m \le n$) depends on whether the number of runs is even or odd as follows

$$P(R = 2k) = \frac{2\binom{m-1}{k-1}\binom{n-1}{k-1}}{\binom{m+n}{m}},$$

$$\text{or } P(R = 2k+1) = \frac{\binom{m-1}{k-1}\binom{n-1}{k} + \binom{m-1}{k}\binom{n-1}{k-1}}{\binom{m+n}{m}}.$$

Here the maximum number of runs is $2m + 1$ if $m < n$, or $2m$ if $m = n$, and $k = 1, 2, \ldots, m \le n$. However, in the special case that $m > 20$ and $n > 20$, the probabilities of r runs can be estimated by using a normal distribution with

$$\mu = \frac{2mn}{m+n} + 1,$$

$$\sigma^2 = \frac{2mn(2mn - m - n)}{(m+n)^2(m+n-1)},$$

$$\text{and } z = \frac{r - \mu + 0.5}{\sigma},$$

where the usual continuity correction is employed.

The previous conditional probabilities depend on the values of m and n, but it is sometimes useful to know the absolute probability of R runs given $N = n + m$ nonzero residuals. There will always be at least one run, so the probability of r runs occurring depends on the number of ways of choosing break points where a sequence of residuals changes sign. This will be the same as the number of ways of choosing $r - 1$ items from $N - 1$ without respect to order, divided by the total number of possible configurations, i.e. the probability of $r - 1$ successes in $N - 1$ independent Bernoulli trials given by

$$P(R = r) = \binom{N-1}{r-1}\left(\frac{1}{2}\right)^{N-1}.$$

This is the value referred to as the probability of runs given the number of nonzero residuals in the previous tables of results.

4.2.18 F test for excess variance

Is is often required to fit two or more possible models in sequence to a data set in order to decide which model is best justified for the data supplied. SimFIT has several programs designed to do this automatically, e.g., **exfit** for sums of exponentials, **mmfit** for sums of Michaelis-Menten equations, **rffit** for positive rational functions, etc. At each stage of model fitting these programs output goodness of fit measures, and perform an F tests for excess variance. The SimFIT program **qnfit**, like these other programs, also allows users to store F test details for retrospective testing which can also be done interactively, as now described.

Open the SimFIT main menu, choose [Statistics], then [Standard tests], and select the F test option. This requires you to input the following values.

1. The objective function Q_1 and number of parameters m_1 in the simpler model.
2. The objective function Q_2 and number of parameters m_2 in the richer model.
3. The number of experimental data points n.

Obviously this test requires $Q_1 > Q_2$, $m_2 > m_1$, and $n > m_2$.

For instance, using the default parameters gives these results.

F test results	
$Q1$ ($(W)SSQ$ for model 1)	12.00
$Q2$ ($(W)SSQ$ for model 2)	10.00
$m1$ (number of parameters in model 1)	2
$m2$ (number of parameters in model 2)	3
n (number of experimental points)	12
Numerator degrees of freedom	1
Denominator degrees of freedom	9
F test statistic TS	1.800
$P(F \geq TS)$	0.2126
$P(F \leq TS)$	0.7874
5% upper tail critical point	5.117
1% upper tail critical point	10.56
Conclusion:	
Model 2 is not justified ... Tentatively accept model 1	

From such values SimFIT calculates the test statistic TS given by

$$TS = \frac{(Q_1 - Q_2)/(m_2 - m_1)}{Q_2/(n - m_2)}$$

which may be distributed, or more usually only approximately distributed, according to the F distribution with $m_2 - m_1$ an $n - m_2$ degrees of freedom. The table indicates that, as TS was less than the upper 5% critical point, there are no grounds for accepting the richer model in this particular case.

It must be emphasized that, while simulations suggest that this test is fairly robust in simple cases like distinguishing one exponential from two, the test is only really justified for fitting linear nested model families like polynomials, or multilinear regression, as now outlined.

Justification for the F test can be illustrated by successive fitting of polynomials to the same data

$$\begin{aligned}
H_0 &: f(x) = \alpha_0 \\
H_1 &: f(x) = \alpha_0 + \alpha_1 x \\
H_2 &: f(x) = \alpha_0 + \alpha_1 x + \alpha_2 x^2 \\
&\cdots \\
H_k &: f(x) = \alpha_0 + \alpha_1 x + \alpha_2 x^2 + \cdots + \alpha_k x^k
\end{aligned}$$

in a situation where experimental error is normal with zero mean and constant variance, and the true model is a polynomial, a situation that will never be encountered in real life.

The important distributional results, can be illustrated for the case of two models i and j, with $j > i \geq 0$, so that the number of points and parameters satisfy $n > m_j > m_i$ while the sums of squares are $Q_i > Q_j$, then

1. $(Q_i - Q_j)/\sigma^2$ is $\chi^2(m_j - m_i)$ under model i
2. Q_j/σ^2 is $\chi^2(n - m_j)$ under model j
3. Q_j and $Q_i - Q_j$ are independent under model j.

So the likelihood ratio test statistic

$$F = \frac{(Q_i - Q_j)/(m_j - m_i)}{Q_j/(n - m_j)}$$

is distributed as $F(m_j - m_i, n - m_j)$ if the true model is model i, which is a special case of model j in the nested hierarchy of the polynomial class of linear models.

In most experiments the models $f(x, \theta)$ fitted to n observations y_i are nonlinear in the parameters θ, and the variance of the experimental error usually increases as the measured responses increase. In an attempt to allow for this, weighting factors w_i are introduced so that the objective function at the minimum would be the sum of weighted squared residuals $WSSQ$ given by

$$WSSQ = \sum_{i=1}^{n} w_i \{y_i - f(x_i, \hat{\theta})\}^2.$$

However, the choice of weighting factors is highly controversial. For instance, if the variances of the experimental errors were known to be σ_i^2, then an obvious choice for the weighting factors would be

$$w_i = \frac{1}{\sigma_i^2}$$

effectively reducing the problem to the special case of constant variance, much beloved by theoreticians.

To study individual cases, these options are available using SimFIT, especially with program **qnfit**.

- Set all $w_i = 1$.
  Fitting will tend to be unduly dominated by large responses y_i.
- Set all $w_i = 1/s_i^2$, where s_i are standard errors determined from replicates.
  Standard error estimates will be unreliable unless large number of replicates are available.
- Set all $w_i = 1/g(y_i)$, for some specified function $g(y)$.
  Fitting may be dominated by small responses y_i if $g(y) = (\alpha y)^2$ for some α.
- Set all $w_i = 1/h(f)$, for some specified function $h(f(x, \theta))$.
  Weights will change for each cycle of optimization as the parameters change.

All these can lead to bias and using $s, g(y)$ or $h(f)$ should be justified by further investigation.

4.2.19 Tests for equal dispersion

It is frequently of interest to compare two samples without any assumptions about the population distribution, and SimFIT provides an interface to conduct such nonparametric tests for equality of the median and dispersion, i.e. the variance, with two such samples.

Open the main SimFIT menu, choose [A/Z], then select the SimFIT nonparametric test program **rstest**, and run the Median, Mood, and David tests using the following default data

X-values	6	9	12	4	10	11
Y-values	8	1	3	7	2	5

leading to these results.

Median, Mood and David tests number 1	
Current data sets X and Y are:	
G08BAF.TF1: Mood-David tests for equal dispersions	
Number of X-values	6
G08BAF.TF2: Mood-David tests for equal dispersions	
Number of Y-values	6
Results for the median test:	
H_0: medians are the same	
Number of X-scores < pooled median	2
Number of Y-scores < pooled median	4
Probability under H_0	0.2835
Results for the Mood test	
H_0: dispersions are equal	
H_1: X-dispersion > Y-dispersion	
H_2: X-dispersion < Y-dispersion	
The Mood test statistic	75.50
Probability under H_0	0.8339
Probability under H_1	0.4170
Probability under H_2	0.5830
Results for the David test	
H_0: dispersions are equal	
H_1: X-dispersion > Y-dispersion	
H_2: X-dispersion < Y-dispersion	
The David test statistic	9.467
Probability under H_0	0.3972
Probability under H_1	0.8014
Probability under H_2	0.1986

As usual with SimFIT, all three results are given for convenience, but with the understanding that either only one pre-decided test is to be used, or that the Bonferroni correction will be employed if more than one test result is to be considered.

These tests all start by forming a pooled sample, then calculating the overall median M of the pooled sample and considering various functions of the ranks r_i within this pooled sample. It is not surprising that with such small samples no significant differences were detected in this case.

However, to better understand what these tests do, you should now use test files `g08acf.tf1` and `g08acf.tf2`, which have larger and more distinct samples and lead to the following results.

Median, Mood and David tests number 2		
Current data sets X and Y are:		
G08ACF.TF1: the median test		
Number of X-values	16	
G08ACF.TF2: the median test		
Number of Y-values	23	
Results for the median test:		
H_0: medians are the same		
Number of X-scores < pooled median	13	
Number of Y-scores < pooled median	6	
Probability under H_0	0.0009	Reject H_0 at 1% significance level
Results for the Mood test		
H_0: dispersions are equal		
H_1: X-dispersion > Y-dispersion		
H_2: X-dispersion < Y-dispersion		
The Mood test statistic	1947	
Probability under H_0	0.8200	
Probability under H_1	0.5900	
Probability under H_2	0.4100	
Results for the David test		
H_0: dispersions are equal		
H_1: X-dispersion > Y-dispersion		
H_2: X-dispersion < Y-dispersion		
The David test statistic	69.77	
Probability under H_0	0.0130	Reject H_0 at 5% significance level
Probability under H_1	0.9935	
Probability under H_2	0.0065	Reject H_0 at 1% significance level

The calculations used to perform these tests will now be outlined.

The Median test

If there are n observations overall, with individual sample sizes n_x and n_y so that $n = n_x + n_y$, then the data can be expressed as a 2 by 2 contingency table with frequencies

$$
\begin{aligned}
f_{11} &= \text{ Number of } X \leq M \\
f_{21} &= n_x - f_{11} \\
f_{12} &= \text{ Number of } Y \leq M \\
f_{22} &= n_y - f_{12}
\end{aligned}
$$

then a chi-square test, or with small samples ($n \leq 100$) a Fisher exact test, is carried out. The analysis for these data leads to the following table of results when a contingency table analysis is performed using SIMFIT, but displaying only the most important results.

Fisher exact test

Observed	Rearranged so r_1 = smallest marginal, $c_2 \geq c_1$	
13 6	13 3	
3 17	6 17	
$p(13)$	0.000820	$p(*)$, observed frequencies
$p(14)$	0.000059	
$p(15)$	0.000002	
$p(16)$	0.000000	
P_sum3	0.000881	sum of all $p(r)$ for $r \geq 13$

Of course, it is obvious from the way the two data sets are partitioned by the overall median M in this contingency table that the Y values tend to be larger than the X values, and the Fisher exact probability confirms this. Note that, in order to calculate the significance level for this table, the Fisher exact test must not only consider the probability of the given table $p(*)$ but must add the sum of probabilities for the more extreme tables, i.e., with f_{11} equal to 14, 15, and 16.

Mood's test

This assumes that the two samples have the same mean so that

$$W = \sum_{i=1}^{n_x} \left(r_i - \frac{n+1}{2} \right)^2,$$

which is the sum of squares of deviations from the average rank in the pooled sample, is approximately normally distributed for large n. The test statistic is

$$z = \frac{W - n_x(n^2-1)/12}{\sqrt{n_x n_y (n+1)(n^2-4)/180}}.$$

This test suffers from the disadvantage that is assumes equal means for the two samples and, if this is not justified, it can lead to inflated values for W.

David's test

This test uses the mean rank

$$\bar{r} = \sum_{i=1}^{n_x} r_i / n_x$$

to reduce the effect of the assumption of equal means in Mood's test by calculating

$$V = \frac{1}{n_x - 1} \sum_{i=1}^{n_x} (r_i - \bar{r})^2,$$

and V is also approximately normally distributed for large n. The test statistic is

$$z = \frac{V - n(n+1)/12}{\sqrt{n n_y (n+1)(3(n+1)(n_x+1) - n n_x)/360 n_x (n_x - 1)}}.$$

Note that it is not the values of W or V alone that determine the significance level for these dispersion tests, but the z statistics calculated from them as defined above. It is often recommended that David's test is more discerning than Mood's test, which seems to be the case with these data.

4.2.20 Tests for equal variance

It is frequently necessary to check for homogeneity of variance given n samples, that is to perform a statistical test to see if it reasonable to assume they have a common variance in the poulation. In particular, analysis of variance (ANOVA) is based on the assumption that all samples are normally distributed with the same variance.

Now it is often stated that ANOVA procedures are relatively insensitive to small departures from normality, but are much more affected by differences in variances between groups so, for that reason, variance-stabilizing transformations are frequently resorted to. Variance homogeneity tests are best done interactively on the data set, so that the effect of transformations on variance stabilizations can be judged before proceeding to ANOVA. SimFiT provides the facility to read in samples as individual data sets with possibly differing sample sizes, or as a matrix if all samples have the same size, but then to test for homogeneity of variance under all conditions of variance stabilizing transformations in order to check that the transformation selected has succeeded.

The next table illustrates analysis of data in the test file `anova1.tf1` for homogeneity of variance, using the Bartlett test, and also the Levene test.

Homogeneity of variance test 1: Bartlett

Transformation	x (untransformed data)
B	0.69006
C	1.0800
B/C	0.63895
Number of degrees of freedom	4
$P(\chi^2 \geq B/C)$	0.9586
Upper tail 1% point	13.277
Upper tail 5% point	9.4877

Homogeneity of variance test 2: Levene (median)

Transformation	x (untransformed data)
W	0.18458
Degrees of freedom 1	4
Degrees of freedom 2	25
$P(F \geq W)$	0.9442
Upper tail 1% point	4.1774
Upper tail 5% point	2.7587

In order to interpret these results it is necessary to understand the assumptions involved and the statistics that are calculated, so such issues will now be discussed.

Bartlett's test

With just two normal samples the F test is recommended, and this can be performed routinely in SimFiT as part of the t test procedure, which is actually equivalent to 1-way ANOVA when there are only two samples. Where there are n normal samples the Bartlett test is recommended, and this is just the same as the F test when there are two samples of the same size. If there are k groups, with sample size n_i, $\nu_i = n_i - 1$, and sample

variances s_i^2, then the pooled variance estimate s_p^2, and parameters B and C can be calculated as follows,

$$s_p^2 = \frac{\sum_{i=1}^{k} \nu_i s_i^2}{\sum_{i=1}^{k} \nu_i}$$

$$B = \log(s_p^2) \sum_{i=1}^{k} \nu_i - \sum_{i=1}^{k} \nu_i \log(s_i^2)$$

$$C = 1 + \frac{1}{3(k-1)} \left(\sum_{i=1}^{k} \frac{1}{\nu_i} - \frac{1}{\sum_{i=1}^{k} \nu_i} \right).$$

To test homogeneity of variance, the Bartlett test statistic B/C is approximately chi-square distributed with $k-1$ degrees of freedom.

Levene's test

When normality cannot be assumed, the Levene test can be performed. If the total number of observations is $N = \sum_{i=1}^{k} n_i$, then the test statistic W is defined as

$$W = \frac{(N-k) \sum_{i=1}^{k} n_i (Z_{i.} - Z_{..})^2}{(k-1) \sum_{i=1}^{k} \sum_{j=1}^{n_i} (Z_{ij} - Z_{i.})^2},$$

where $Z_{..}$ is the mean of all Z_{ij}, and $Z_{i.}$ is the group mean of the Z_{ij}. If Y_{ij} is observation j in group i the definitions are

$$Z_{ij} = |Y_{ij} - Y_{i.}|$$

$$Z_{..} = \frac{1}{N} \sum_{i=1}^{k} \sum_{j=1}^{n_i} Z_{ij}$$

$$Z_{i.} = \frac{1}{n_i} \sum_{j=1}^{n_i} Z_{ij},$$

but note that there are several ways to define $Y_{i.}$. Usually this is taken to be the median of group i, but if there are long tails in the distribution as with the Cauchy distribution, the the trimmed mean can be used. The group mean can also be used if the data are similar to a normal distribution. To test variance homogeneity, the Levene test statistic W is approximately F distributed with $k-1$ and $N-k$ degrees of freedom.

The above table illustrates that the null hypothesis

$$H_0 : \sigma_1^2 = \sigma_2^2 = \cdots = \sigma_k^2$$

cannot be rejected for the data in test file `anova1.tf1`.

4.2.21 Kendall's coefficient of concordance

Kendall's coefficient of concordance estimates the extent of agreement between n objects ranked on k different variables in order to test the null hypothesis

$$H_0: \quad \text{There is no agreement between the comparisons.}$$

Open the SIMFIT main menu, choose [A/Z], select the SIMFIT nonparametric test program **rstest**, then run the Kendall coefficient of concordance option and examine the following default data set of rankings for 10 objects ranked on 3 variables.

1.0	4.5	2.0	4.5	3.0	7.5	6.0	9.0	7.5	10.0
2.5	1.0	2.5	4.5	4.5	8.0	9.0	6.5	10.0	6.5
2.0	1.0	4.4	4.5	4.5	4.5	8.0	8.0	8.0	10.0

The results are as follows, suggesting that H_0 should be rejected.

Kendall coefficient of concordance analysis 1

H_0: no agreement between comparisons		
Data: test file G08DAF.TF1		
Number of columns (objects)	10	
Number of rows (variables)	3	
Kendall coefficient W	0.8277	
$P(\chi^2 \geq W)$	0.0078	Reject H_0 at 1% significance level

The data matrix supplied for analysis can contain observations or ranks as follows.

- Data must have n columns for data/objects/ranks (across), and k rows for comparisons/variables (down).
- The matrix can have original values to be ranked automatically along rows, or else contain pre-calculated ranks instead of values.
- Tied ranks are averaged as usual, so a ranked matrix must have these two properties:
  1. $A(i, j) > 0$ for $i = 1, 2, ..., k$ and $j = 1, 2, ..., n$
  2. Sum of $A(i, j)$ for each i and for $j = 1, 2, ..., n$ must be $n(n + 1)/2$
- Note that, if data values are supplied instead of ranks, then SIMFIT will calculate the ranks automatically.

For instance, the data in test file `kendall.tf1` contains these measurements for wing length, tail length,and bill length for 12 birds

10.4	10.8	11.1	10.2	10.3	10.2	10.7	10.5	10.8	11.2	10.6	11.4
7.4	7.6	7.9	7.2	7.4	7.1	7.4	7.2	7.8	7.7	7.8	8.3
17.0	17.0	20.0	14.5	15.5	13.0	19.5	16.0	21.0	20.0	18.0	22.0

and then the following ranks, calculated internally by SIMFIT before performing the test,

4.0	8.5	10.0	1.5	3.0	1.5	7.0	5.0	8.5	11.0	6.0	12.0
5.0	7.0	11.0	2.5	5.0	1.0	5.0	2.5	9.5	8.0	9.5	12.0
5.5	5.5	9.5	2.0	3.0	1.0	8.0	4.0	11.0	9.5	7.0	12.0

lead to the next table of results.

Kendall coefficient of concordance analysis 2

H_0: no agreement between comparisons		
Data: test file KENDALL.TF1		
Number of columns (objects)	12	
Number of rows (variables)	3	
Kendall coefficient W	0.9241	
$P(\chi^2 \geq W)$	0.0013	Reject H_0 at 1% significance level

As before, the null hypothesis is rejected for this alternative data set.

Calculating the Kendall coefficient of concordance *W*

Ranks r_{ij} for the the rank of object j in comparison i (with tied values being given averages) are used to calculate the n column rank sums R_j, which would be approximately equal to the average rank sum $k(n+1)/2$ under

$$H_0: \text{ There is no association among the variables.}$$

For total agreement the R_j would have values from some permutation of $k, 2k, \ldots, nk$, and the total squared deviation of these is $k^2 n(n^2-1)/12$.

Then the coefficient W is calculated according to

$$W = \frac{\sum_{j=1}^{n} (R_j - k(n+1)/2)^2}{k^2 n(n^2-1)/12 - k\sum T}$$

which lies between 0 for complete disagreement and 1 for complete agreement.

Here the denominator correction for ties uses T defined as

$$T = \sum t(t^2-1)/12$$

where t is the number of occurrences of each tied rank within a comparison.

For large samples ($n > 7$), $k(n-1)W$ is approximately χ^2_{n-1} distributed, otherwise tables should be used for accurate significance levels.

4.3 Data exploration

Tutorials and worked examples for simulation, curve fitting, statistical analysis, and plotting.
https://simfit.uk
https://simfit.org.uk
https://simfit.silverfrost.com

4.3.1 Introduction

It is frequently useful to examine a given data set without necessarily having any particular statistical tests in mind, but rather as a preliminary to more searching investigations.

In this section the following two definitions are used.

1. A single sample supplied as a column of numbers will be referred to as a vector
2. A rectangular table of numbers will be referred to as a matrix

SIMFIT provides a selection of techniques that can be used in order to make a preliminary examination of such data sets by choosing [Statistics] from the main SIMFIT menu followed by [Data exploration].

- **Exhaustive analysis of a vector**

  Displays a table with a summary of estimated moments, ranges, coefficients of variation/skew/kurtosis. Creates plots as histograms, cumulative distributions, pie charts, bar charts, time series, centered rods, normal/half-normal scores. Performs tests on runs, and/or signs, or for a normal distribution.

- **Exhaustive analysis of a matrix**

  In addition to plotting barcharts and calculating covariance or correlation matrices it is possible to select individual rows or columns for the previous exhaustive analysis of a vector procedure.

- **Exhaustive analysis of a multivariate normal matrix**

  Numerous tests are provided to ascertain if it is reasonable to assume multivariate normality and sphericity before uncritically proceeding to employ MANOVA techniques.

- **Parametric t tests on groups across rows of a matrix**

  This creates a summary of the results from tests on rows where columns have been assigned to groups.

- **Nonparametric tests across rows of a matrix**

  Where the normality and constant variance required for t tests are not justified a similar procedure can be performed using nonparametric tests.

- **All possible pairwise tests on n vectors or a library file**

  Given sets of vectors it is possible to use a selection of parametric and nonparametric tests on all possible pairs of columns. Naturally, with such a procedure there are limits to the number of comparisons allowed, i.e. $n(n-1)/2$ for n vectors.

- **Robust analysis of 1 sample**

  Results can be obtained for parameters from winzorized samples.

- **Robust analysis of 2 samples**

  Two samples can be compared by robust methods.

Of course it is assumed that users will be aware of the limitations arising from multiple tests on the same data implied by some of these procedures, as described for the Bonferroni and related techniques in the SIMFIT reference manual. They should rather interpret the results as preliminary, as should be expected for data exploration.

4.3.2 Exhaustive analysis of a vector

Given any sample it is useful to generate a summary of the all the parameters that can be estimated together with the ability to plot the data in alternative ways.

Summary statistics

For example, from the main SimFIT menu choose [Statistics] then [Data exploration] and read the default vector test file `normal.tf1` into the procedure called exhaustive analysis of a vector. Here you can obtain the usual summary statistics as in this table, including the range, hinges (i.e. quartiles), mean $\bar{x}$, standard deviation s, coefficient of variation CV% ($100s/\bar{x}$, i.e. the reciprocal of the signal to noise ratio), and the normalized sample moments s_3 (coefficient of skewness), and s_4 (coefficient of kurtosis).

Exhaustive analysis of a vector
Data: Test file normal.tf1: 50 random numbers

Sample size	50
Minimum, Maximum values	-2.20820, 1.61750
Lower and Upper Hinges	-0.85502, 0.78597
Coefficient of skewness	-0.01669
Coefficient of kurtosis	-0.76840
Median value	-0.09736
Sample mean	-0.02579
Sample standard deviation	1.00553: CV% = 3899%
Standard error of the mean	0.14220
Upper 2.5% t-value	2.00958
Lower 95% confidence limit for mean	-0.31156
Upper 95% confidence limit for mean	0.25998
Variance of the sample	1.01109
Lower 95% confidence limit for variance	0.70552
Upper 95% con limit for variance	1.57006
Shapiro-Wilks W statistic	0.96270
Significance level for W	0.1153 *Tentatively accept normality*

Testing for a normal distribution

The normalized sample moments shown in this table are useful for seeing how far a sample departs from a normal distribution and are defined in a sample of size n by the following equations.

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

$$s_3 = \frac{n}{(n-1)(n-2)}\frac{\sum_{i=1}^{n}(x_i - \bar{x})^3}{s^3}$$

$$s_4 = \frac{(n+1)n}{(n-1)(n-2)(n-3)}\frac{\sum_{i=1}^{n}(x_i - \bar{x})^4}{s^4} - \frac{3(n-1)^2}{(n-2)(n-3)}.$$

The coefficient of skewness or symmetry indicates the extent to which the sample suggest deviation from a symmetrical distribution. Values less than zero indicate skew to the left with a mean less than the median, while values greater than zero indicate skew to the right with mean greater than the median. The coefficient of kurtosis indicates the amount of peakedness in the distribution. Values less than zero indicate a platykurtic

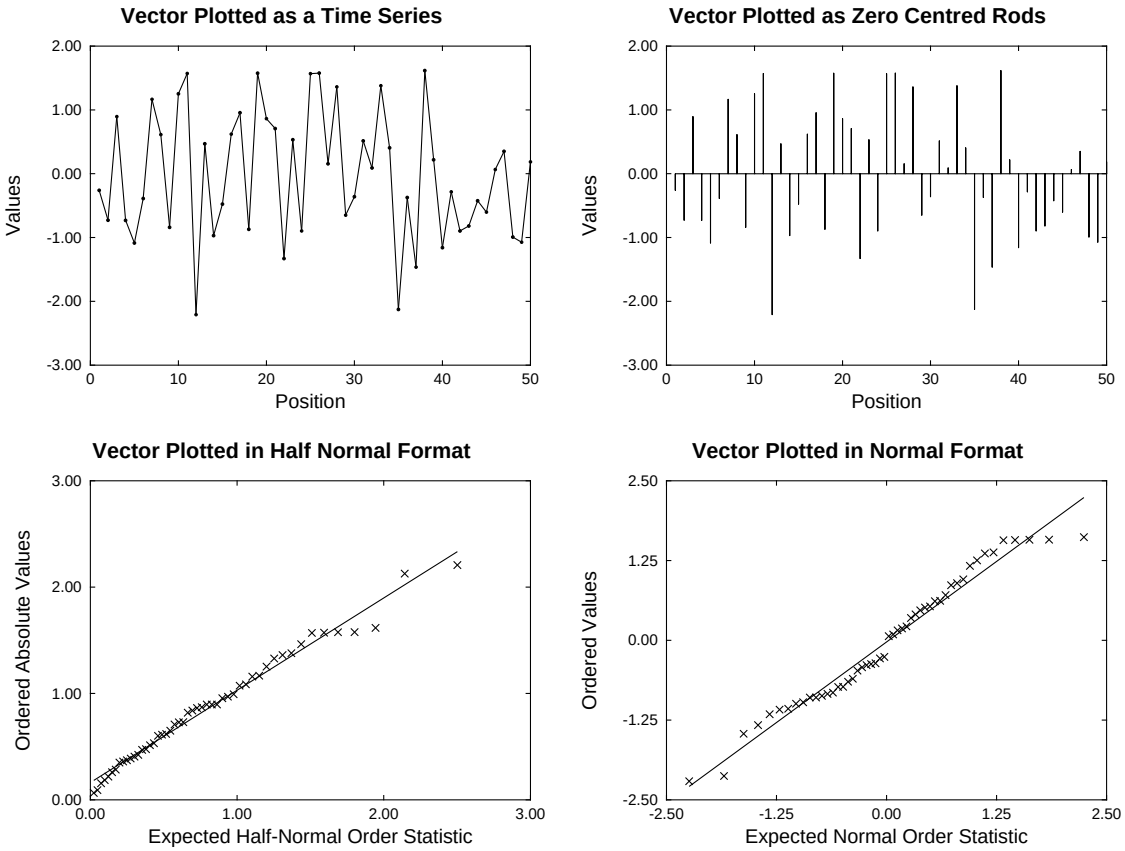
distribution which is more humped than a normal distribution, while values greater than zero indicate a leptokurtic distribution which is more peaked than a normal distribution. A normal distribution is said to be mesokurtic with both coefficients equal to zero.

As it is often wished to see how closely a sample resembles a normal distribution several options are provided for this purpose. You can perform a Shapiro-Wilks test for normality (only on demand since this will, of course, not always be appropriate) or create a histogram, pie chart, cumulative distribution plot or appropriate curve-fitting files. This option is a very valuable way to explore any single sample before considering other tests.

Graph plotting options

Since vectors have only one coordinate, graphical display requires a further coordinate. In the case of histograms the extra coordinate is provided by the choice of bins, which dictates the shape, but in the case of cumulative distributions it is automatically created as steps and therefore of unique shape. Pie chart segments are calculated in proportion to the sample values, which means that this is only appropriate for positive samples, e.g., counts.

The other techniques illustrated in this next figure may require further explanation.



If the sample values have been measured in some sequence of time or space, then the y values could be the sample values while the x values would be successive integers, as in the time series plot. Sometimes it is useful to see the variation in the sample with respect to some fixed reference value, as in the zero centered rods plot. The data can be centered automatically about zero by subtracting the sample mean if this is required.

The half normal and normal plots are particularly useful when testing for a normal distribution with residuals, which should be approximately normally distributed if the correct model is fitted.

In the half normal plot, the absolute values of a sample of size n are first ordered then plotted as $y_i, i = 1, \ldots, n$, while the half normal order statistics are approximated by

$$x_i = \Phi^{-1}\left(\frac{n + i + \frac{1}{2}}{2n + \frac{9}{8}}\right),\ i = 1, \ldots, n$$

which is valuable for detecting outliers in regression.

The normal scores plot simply uses the ordered sample as y and the normal order statistics are approximated by

$$x_i = \Phi^{-1}\left(\frac{i - \frac{3}{8}}{n + \frac{1}{4}}\right),\ i = 1, \ldots, n$$

which makes it easy to visualize departures from normality. Best fit lines, correlation coefficients, and significance values are also calculated for half normal and normal plots.

Note that elsewhere in SIMFIT a more accurate calculation for expected values of normal order statistics is employed for a normal scores plot and also the Shapiro-Wilks test is just one of several tests for normality available.

4.3.3 Exhaustive analysis of a matrix

Given any data sample in the form of a rectangular table of values with no missing values it is useful to generate a summary of the all the parameters that can be estimated together with the ability to plot the data in alternative ways.

For instance, choose [Statistics] from the main SIMFIT menu then [Data exploration]. Open the [Exhaustive analysis of an arbitrary matrix] option and examine the data set contained in test file `cluster.tf1` which is the following 12 by 8 matrix.

1.0	4.0	2.0	11.0	6.0	4.0	3.0	9.0
8.0	5.0	1.0	14.0	19.0	7.0	13.0	21.0
3.0	1.0	3.0	1.0	3.0	6.0	23.0	37.0
9.0	0.0	7.0	7.0	1.0	2.0	21.0	2.0
7.0	12.0	9.0	5.0	14.0	9.0	12.0	14.0
2.0	13.0	15.0	2.0	23.0	6.0	34.0	8.0
11.0	7.0	2.0	1.0	4.0	17.0	11.0	4.0
6.0	3.0	7.0	12.0	11.0	8.0	8.0	0.0
8.0	21.0	1.0	10.0	31.0	9.0	3.0	18.0
19.0	14.0	12.0	9.0	16.0	10.0	0.0	27.0
17.0	18.0	10.0	6.0	19.0	14.0	1.0	24.0
15.0	21.0	8.0	7.0	17.0	12.0	4.0	22.0

The possibilities for further analysis are now listed.

- Summarize all columns (or rows)
- Exhaustive analysis of any column (or row)
- Analyze/paired-test any two rows (or columns)
- Plot
  2D barchart or stack plot with rows as groups
  2D box and whisker plot or bars and error bars
  2D scattergrams with symbols (and lines if requested)
  3D barchart or cylinder plot
- Display/file Sum–of–Squares, covariance, or correlation matrix

Summarizing all rows or columns

For instance, the option to summarize all columns results in this analysis.

Column	Mean	Variance	St.Dev.	Coeff.Var.
1	8.83333	33.4242	5.78137	65.45%
2	9.91667	57.7197	7.59735	76.61%
3	6.41667	21.5379	4.64089	72.33%
4	7.08333	18.6288	4.31611	60.93%
5	13.6667	81.3333	9.01850	65.99%
6	8.66667	17.6970	42.0678	48.54%
7	11.0833	107.720	10.3788	93.64%
8	15.5000	127.364	11.2855	72.81%

Here, for each column, the summary statistics are calculated downwards for all rows, and a similar table can be generated for rows calculated across all columns.

Pairwise statistical tests between rows or columns

Next consider pairwise statistical tests between selected rows or columns. If you choose to apply more than one statistical test this would be to use the absolutely forbidden technique of multiple tests on the same data.

In such a situation you can either use the Bonferroni method or similar with a factor related to the actual number of tests applied as explained in the SIMFIT reference manual, or just use commonsense and regard this as a preliminary examination where the p values are simply being regarded as indicators of the differences between paired or columns and not being used for hypothesis tests.

Analysis and two-tail tests for: $N = 12$, $X =$ column 1, $Y =$ column 2	
Unpaired t test:	
t	-0.39309
p	0.69804
Paired t test:	
t	-0.56175
p	0.58555
Kolmogorov-Smirnov 2-sample test:	
d	0.25000
z	0.10206
p	0.53610
Mann-Whitney U test:	
u	68.5000
z	-0.17339
p	0.85377
Wilcoxon signed rank test:	
w	33.5000
z	-0.39299
p	0.70752
Run test:	
+	6 (no. $x > y$)
-	6 (no. $x < y$)
p	0.60823
Sign test: N for non-tied pairs	
N	12
-	6 (no. $x < y$)
p	1.00000

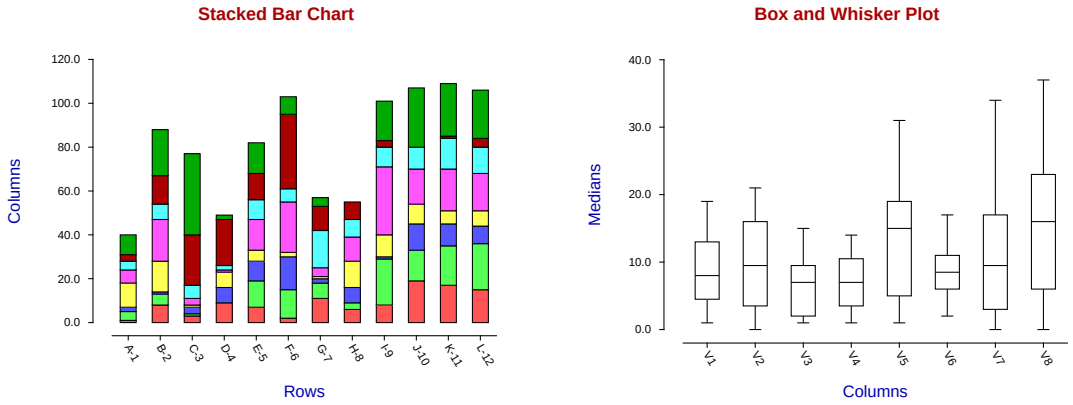
After an analysis like the above has been carried out, less controversial results are calculated, that is, the inner product of the two selected rows or columns regarded as vectors, leading to the angle between them and Euclidean distance between them, i.e. the square root of the sum of squared differences.

n	dot product	x size	y size	distance	$\cos(\theta)$	radians	degrees
12	1307.0	36.1109	42.6028	22.4722	0.849569	0.5556	31.835

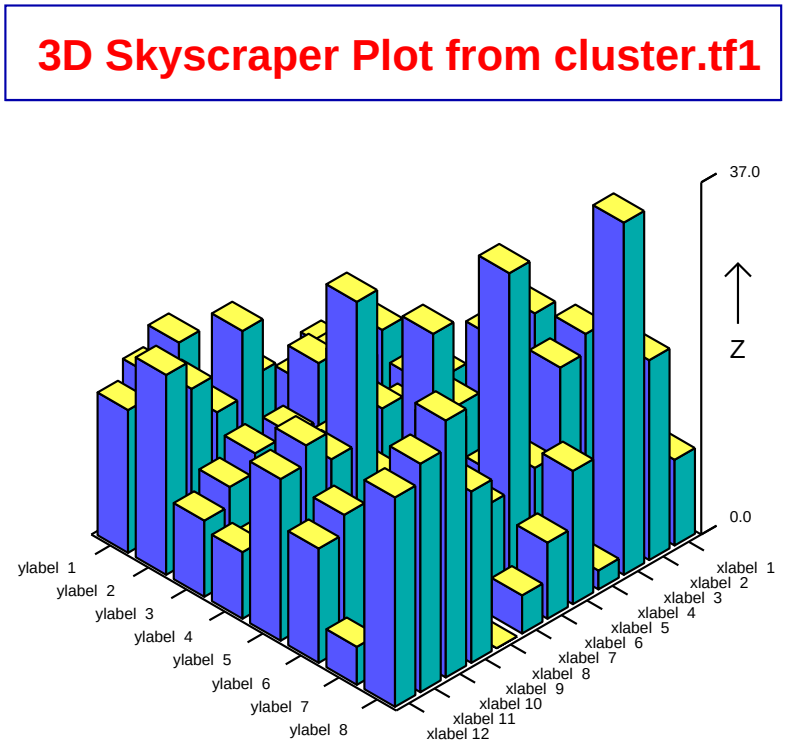
Plotting a matrix

As long as the number of rows and columns is fairly small, say < 20, and for some procedures the matrix contains only positive values, several graphs can be drawn to visualize the relative magnitude of column values across row.

For instance, the left-hand figure below plots a bar for each row with a stacked bar of segments each proportional to the column values for the corresponding rows. On the right is a box and whisker plot to illustrate the quartiles for each column calculated for all rows. Of course it is easy to interpret the row and column effects illustrated when it is realized that the data set has 12 rows and 8 columns.



The next plot illustrates a 3D skyscraper plot where, for each value in the data matrix, say x_{ij}, the vertical height of the bars is proportional to the x_{ij} values.



Numerous other graphs are available where the sign of x_{ij} is irrelevant, for instance clusters and 95% confidence ellipses for the data means or for the overall data ranges, and also linear regression according to all three conventions is possible for selected pairs of rows and/or columns.

Lower triangles of the covariance and correlation matrices

The exhaustive analysis of an arbitrary matrix can also calculate several symmetrical matrices from the data. For instance the sum of squares matrix or the covariance matrix as this will give some idea if the columns are independent.

Variance-Covariance matrix

33.4242							
23.2576	57.7197						
7.71212	11.5833	21.5379					
1.65152	-0.71970	-5.67424	18.6288				
10.1212	54.3333	9.06061	10.8485	81.3333			
15.2121	17.0606	0.51515	-5.15152	7.69697	17.6970		
-35.2576	-33.3561	11.1439	-23.4621	-18.2424	-19.7879	107.720	
19.6364	25.7727	-1.59091	-5.68182	21.8182	6.45455	-19.8636	127.364

An easier matrix to visualize for correlations in the data is the correlation matrix, which is sometimes given, as below, with unit diagonals to avoid confusion.

Pearson product-moment correlations

1							
0.529507	1						
0.287436	0.328526	1					
0.066185	-0.021948	-0.283279	1				
0.194119	0.792994	0.216482	0.278704	1			
0.625474	0.533805	0.026387	-0.283722	0.202879	1		
-0.587590	-0.423024	0.231361	-0.523754	-0.194895	-0.453213	1	
0.300959	0.300591	-0.030375	-0.116647	0.214369	0.135954	-0.169585	1

For a n by m matrix $\boldsymbol{X}$ with values x_{ij}, the sample column means $\bar{x}_j$, vector of column means $\bar{\boldsymbol{x}}$, variance of the j'th variable s_{jj}, covariance between the j and k'th variable s_{jk}, correlation between the j and k'th variable c_{jk}, and covariance matrix $\boldsymbol{S}$ are defined as follows.

$$\bar{x}_j = \frac{1}{n}\sum_{i=1}^{n} x_{ij}$$

$$\bar{\boldsymbol{x}}^T = (\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_m)$$

$$s_{jj} = \frac{1}{n-1}\sum_{i=1}^{n}(x_{ij} - \bar{x}_j)^2$$

$$s_{jk} = \frac{1}{n-1}\sum_{i=1}^{n}(x_{ij} - \bar{x}_j)(x_{ik} - \bar{x}_k)$$

$$c_{jk} = \frac{s_{jk}}{\sqrt{s_{jj}s_{kk}}}$$

$$\boldsymbol{S} = \frac{1}{n-1}\sum_{i=1}^{n}(\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^T$$

Alternatively, if $\hat{\boldsymbol{X}}$ is the matrix centered by subtracting the sample column means and $\tilde{\boldsymbol{X}}$ is the centered matrix scaled by dividing by the column standard deviations, then the sample covariance $\boldsymbol{S}$ and correlation matrices $\boldsymbol{C}$ are

$$\boldsymbol{S} = \frac{1}{n-1}\hat{\boldsymbol{X}}^T\hat{\boldsymbol{X}} \text{ and } \boldsymbol{C} = \frac{1}{n-1}\tilde{\boldsymbol{X}}^T\tilde{\boldsymbol{X}}.$$

4.3.4 Exhaustive analysis of a multivariate normal matrix

Most multivariate techniques make no assumptions about the data and therefore do not calculate significance levels. Actually, where multivariate hypothesis tests are provided, they are based on definite assumptions about the data, generally assuming a multivariate normal population, and also often making additional assumptions about the nature of the covariance matrix.

For these reasons the SimFiT procedure for exhaustive analysis of a normal multivariate matrix provides options that are useful before proceeding to more specific techniques that depend on multivariate normality, e.g., MANOVA and some types of ANOVA. These multivariate normal analysis procedures can be used by selecting [Statistics] from the main SimFiT menu, then [Data exploration] followed by [Exhaustive analysis of a multivariate normal matrix].

Example 1: The sample means and covariance matrix

Choosing the option to display the means and covariance matrix leads to the following results with test file `hotel.tf1`.

Variable	Mean	Std.err.	lower95%cl	upper95%cl
1	-0.53000	0.46261	-0.55983	-0.50017
2	-0.03000	0.38559	-0.05486	-0.00514
3	-0.59000	0.49091	-0.62165	-0.55835
4	3.10000	1.94622	2.97451	3.22549

Covariance matrix			
2.14011			
-0.11878	1.48678		
-0.89411	0.79144	2.40989	
3.59222	1.88111	-4.60111	37.8778

Correlation matrix			
1			
-0.066588	1		
-0.393709	0.418118	1	
0.398982	0.250668	-0.481584	1

CV matrix eigenvalues
0.683568
0.821943
0.775086
0.231793

Determinant	9.81414E+01

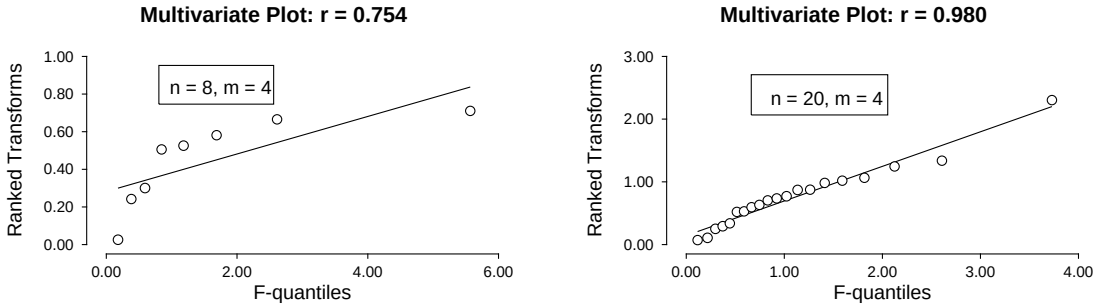
The column means $\bar{x}_j$, and m by m sample covariance matrix S displayed here are defined for a n by m data matrix x_{ij} with $n \geq 2, m \geq 2$ as

$$\bar{x}_j = \frac{1}{n}\sum_{i=1}^{n} x_{ij}$$

$$s_{jk} = \frac{1}{n-1}\sum_{i=1}^{n}(x_{ij} - \bar{x_j})(x_{ik} - \bar{x_k})$$

Example 2: Graphical test for multivariate normality

A graphical technique is provided for investigating if a data matrix with n rows and m columns, where $n >> m > 1$, is consistent with a multivariate normal distribution. For example, the next figure



which shows plots for two random samples from a multivariate normal distribution. The plot uses the fact that, for a multivariate normal distribution with sample mean $\bar{x}$ and sample covariance matrix S,

$$(x-\bar{x})^T S^{-1}(x-\bar{x}) \sim \frac{m(n^2-1)}{n(n-m)} F_{m,n-m},$$

where x is a further independent observation from this population, so that the transforms plotted against the quantiles of an F distribution with m and $n-m$ degrees of freedom, i.e. according to the cumulative probabilities for $(i-0.5)/n$ for $i = 1, 2, \ldots, n$ should be a straight line. It can be seen from the above figure that this plot is of little value for small values of n, say $n \approx 2m$ but becomes progressively more useful as the sample size increases, say $n > 5m$.

Example 3: Hotelling T^2 test H_0 : means = a supplied reference vector

It is possible to perform two variants of the Hotelling T^2 test, namely

- testing for equality of the mean vector with a specified reference vector of means, or
- testing for equality of all means without specifying a reference mean.

Dealing first with testing that a vector of sample means is consistent with a reference vector, consider the next table for a zero reference vector.

Hotelling one sample T^2 test

H_0: Column means = Expected values supplied (i.e. 0)

Number of rows = 10, Number of columns = 4
Hotelling T^2 = 7.43910
F Statistic (FTS) = 1.23985
Degrees of freedom ($d1, d2$) = 4, 6
$P(F(d1, d2) \geq FTS)$ = 0.386864

Column	Mean	Std.Err.	Expected	Delta	t	p
1	-0.53000	0.46261	0.0000	-0.53000	-1.14567	0.281481
2	-0.03000	0.38559	0.0000	-0.03000	-0.07780	0.939687
3	-0.59000	0.49091	0.0000	-0.59000	-1.20186	0.260086
4	3.10000	1.94622	0.0000	3.10000	1.59283	0.145662

This resulted when the test file `hotel.tf1` was analyzed using the Hotelling one sample test procedure. This tests the null hypothesis $H_0 : \mu = \mu_0$ against the alternative $H_1 : \mu \neq \mu_0$, where μ_0 is a known mean vector

and no assumptions are made about the covariance matrix Σ. Hotelling's T^2 is

$$T^2 = n(\bar{x} - \mu_0)^T S^{-1}(\bar{x} - \mu_0)$$

and, if H_0 is true, then an F test can be used since $(n-m)T^2/(m(n-1))$ is distributed asymptotically as $F_{m,n-m}$. Users can input any reference mean vector μ_0 to test for equality of means but, when the data columns are all differences between two observations for the same subjects and the aim is to test for no significant differences, so that μ_0 is the zero vector, as with `hotel.tf1`, the test is a sort of higher dimensional analogue of the paired t test. The table also shows the results when t tests are applied to the individual columns of differences between the sample means $\bar{x}$ and the reference means μ_0, which is suspect because of multiple testing but, in this case, the conclusion is the same as the Hotelling T^2 test: none of the column means are significantly different from zero.

Example 4: Hotelling T^2 test H_0 : all means are equal

Now, turning to a test that all means are equal as displayed in the next table.

Hotelling one sample T^2 test

H_0: Column means are all equal

Number of rows	5	
Number of columns	4	
Hotelling T^2	170.474	
F Statistic (FTS)	28.4123	
Degrees of freedom ($d1, d2$)	3, 2	
$P(F(d1, d2) \geq FTS)$	0.034191	*Reject H_0 at 5% significance level*

This shows the results when the data in `anova6.tf1` are analyzed, and the theoretical background to this test is explained in the reference manual in connection with repeated measures analysis.

Example 5: Compound symmetry

Options are provided for investigating the structure of the covariance matrix. The sample covariance matrix and its inverse can be displayed along with eigenvalues and determinants, and there are also options to check if the covariance matrix has a special form, namely

- testing for compound symmetry,
- testing for spherical symmetry, and
- testing for spherical symmetry of the covariance matrix of orthonormal contrasts.

For instance, using the test file `hotel.tf1` produces the results in the next table.

Compound symmetry test

H_0: Covariance matrix has compound symmetry

Number of groups	1	
Number of variables (m)	4	
Sample size (n)	10	
Determinant of CV	98.14	
Determinant of S_0	1452.0	
$LRTS$ ($-2\log(\lambda)$)	36.30	
Degrees of Freedom	8	
$P(\chi^2 \geq LRTS)$	0.0000	*Reject H_0 at 1% significance level*

This shows an application of a test for compound symmetry which is when a covariance matrix Σ has a special form with constant nonnegative diagonals and equal nonnegative off-diagonal elements as follows.

$$\Sigma = \sigma^2 \begin{pmatrix} 1 & \rho & \dots & \rho \\ \rho & 1 & \dots & \rho \\ \dots & \dots & \dots & \dots \\ \rho & \rho & \dots & 1 \end{pmatrix}$$

This can be tested using estimates for the diagonal and off-diagonal elements σ^2 and $\sigma^2\rho$ as follows

$$s^2 = \frac{1}{m}\sum_{i=1}^{m} s_{ii}$$

$$s^2 r = \frac{2}{m(m-1)}\sum_{i=2}^{m}\sum_{j=1}^{i-1} s_{ij}.$$

Example 6: Sphericity

The sphericity test, designed to test the null hypothesis $H_0 : \Sigma = kI$ against $H_1 : \Sigma \neq kI$. In other words, the population covariance matrix Σ is a simple multiple of the identity matrix, which is a central requirement for some analytical procedures.

Likelihood ratio sphericity test

H_0: Covariance matrix = k*Identity (for some $k > 0$)

Number of small eigenvalues	0 (i.e. $< 1.00E-07$)	
Number of variables (m)	4	
Sample size (n)	10	
Determinant of CV	98.1414	
Mauchly W statistic	0.00676	
$LRTS$ $(-2\log(\lambda))$	49.9740	
Degrees of Freedom	9	
$P(\chi^2 \geq LRTS)$	0.000000	*Reject H_0 at 1% significance level*

The Wilks generalized likelihood-ratio statistic is

$$L = \frac{|S|}{(s^2 - s^2 r)^{m-1}[s^2 + (m-1)s^2 r]},$$

where the numerator is the determinant of the covariance matrix estimated with ν degrees of freedom, while the denominator is the determinant of the matrix with average variance on the diagonals and average covariance as off-diagonal elements, and this is used to construct the test statistic $LRTS$

$$\chi^2 = -\left[\nu - \frac{m(m+1)^2(2m-3)}{6(m-1)(m^2+m-4)}\right]\log L$$

which, for large ν, has an approximate chi-squared distribution with $m(m+1)/2 - 2$ degrees of freedom.

If the sample covariance matrix S has eigenvalues α_i for $i = 1, 2, \dots, m$ then, defining the arithmetic mean A and geometric mean G of these eigenvalues as

$$A = (1/m)\sum_{i=1}^{m} \alpha_i$$

$$G = (\prod_{i=1}^{m} \alpha_i)^{1/m},$$

the likelihood ratio test statistic

$$-2\log\lambda = nm\log(A/G)$$

is distributed asymptotically as χ^2 with $(m-1)(m+2)/2$ degrees of freedom. Using the fact that the determinant of a covariance matrix is the product of the eigenvalues while the trace is the sum, the Mauchly test statistic W can also be calculated from A and G since

$$W = \frac{|S|}{\{Tr(S)/m\}^m} = \frac{\prod_{i=1}^m \alpha_i}{\{(\sum_{i=1}^m \alpha_i)/m\}^m}$$

$$\text{so that } -2\log\lambda = -n\log W.$$

Clearly, the test rejects the assumption that the covariance matrix is a multiple of the identity matrix in this case, a conclusion which is obvious from inspecting the sample covariance and correlation matrices. Since the calculation of small eigenvalues is very inaccurate when the condition number of the covariance matrix is appreciable, any eigenvalues less than the minimal threshold indicated are treated as equal to that threshold when calculating the test statistic.

Example 7: Helmert orthonormal contrasts

The next table results from analysis of `hotel.tf1` and full details will be found in the tutorial on repeat measures ANOVA or in the SIMFIT reference manual.

Sphericity test on CV of Helmert orthonormal contrasts

H_0: Covariance matrix = k*Identity (for some $k > 0$)

Number of small eigenvalues	0 (i.e. $< 1.00E-07$)	
Number of variables (m)	4	
Sample size (n)	10	
Determinant of CV	41.5788	
Trace of CV	32.6105	
Mauchly W statistic	0.03237	
$LRTS$ $(-2\log(\lambda))$	26.4909	
Degrees of Freedom	5	
$P(\chi^2 \geq LRTS)$	0.000072	*Reject H_0 at 1% significance level*
e (Geisser-Greenhouse)	0.402592	
e (Huynh-Feldt)	0.431104	
e (lower bound)	0.333333	

4.3.5 t tests across rows of a matrix

Sometimes data are stored in a matrix such that each row contains values observed within groups that are defined by columns, and where it is wished to test for equality of group means. If normality and identical variance are assumed then systematic t tests can be performed as long as membership of groups is easy to define and change interactively.

To clarify the situation choose [Statistics] from the main SimFIT menu followed by [Data exploration] and then [t tests across rows of a matrix], and browse the default test file `ttest.tf6` which has the following data.

```
 8.8   8.4   7.9   8.7   9.1   9.6   9.9   9.0  11.1   9.6   8.7  10.4   9.5
 8.0   7.4   6.9   8.2   9.7   9.1   9.9   7.0  11.1   8.6   9.7   8.4   9.1
20.2  16.8  15.4  17.1  18.1  20.0   7.7   8.9   9.7   9.1  10.6  10.2  10.1
 1.0   1.0   1.0   1.0   1.0   1.0   1.0   1.0   1.0   1.0   1.0   1.0   1.0
 7.7   8.9  10.2   8.5   9.1   8.6  18.9  19.2  15.1  19.5  18.9  19.9  19.4

begin{indicators}
   1     1     1     1     1     1    -1    -1    -1    -1    -1    -1    -1
end{indicators}
```

Note that, after the data, the group membership is defined by the sectioning commands

begin{indicators} . . . end{indicators}

with a 1 if a column belongs to group X, a -1 if the column belongs to group Y and a 0 if a column is not to be used. So in this data set the first 6 columns belong to group X while the next 7 columns belong to group Y and there are no suppressed columns.

Analysis leads to the following results for the sample means, standard deviations, standard error of the differences, t values, and p values, and where a -1 in the significance level column indicates zero sample variance, as with row 4 emphasized in red, so that a t test cannot be performed.

Variables: $NX = 6$, $NY = 7$

$\bar{X}$	std X	$\bar{Y}$	std Y	se diff	t	2-tail p
8.7500	0.5822	9.7429	0.8182	0.4009	-2.4765	0.030765
8.2167	1.0420	9.1143	1.3005	0.6621	-1.3558	0.202338
17.933	1.8886	9.4714	0.9878	0.8164	10.365	0.000001
-1.0000	-1.0000	-1.0000	-1.0000	-1.0000	-1.0000	-1.000000
8.8333	0.8238	18.700	1.6258	0.7360	-13.405	0.000000

The type of t test (i.e. lower, upper, or 2-tail) and partitioning of columns into groups can be done interactively.

4.3.6 Nonparametric tests across rows of a matrix

Sometimes data are stored in a matrix such that each row contains values observed within groups that are defined by columns, and where it is wished to test for equality of group means. If normality and identical variance cannot be assumed then systematic nonparametric tests can be performed as long as membership of groups is easy to define and change interactively.

To clarify the situation choose [Statistics] from the main SimFIT menu followed by [Data exploration] and then [Nonparametric tests across rows of a matrix], and browse the default test file `ttest.tf6` which has the following data.

```
 8.8   8.4   7.9   8.7   9.1   9.6   9.9   9.0  11.1   9.6   8.7  10.4   9.5
 8.0   7.4   6.9   8.2   9.7   9.1   9.9   7.0  11.1   8.6   9.7   8.4   9.1
20.2  16.8  15.4  17.1  18.1  20.0   7.7   8.9   9.7   9.1  10.6  10.2  10.1
 1.0   1.0   1.0   1.0   1.0   1.0   1.0   1.0   1.0   1.0   1.0   1.0   1.0
 7.7   8.9  10.2   8.5   9.1   8.6  18.9  19.2  15.1  19.5  18.9  19.9  19.4

begin{indicators}
   1     1     1     1     1     1    -1    -1    -1    -1    -1    -1    -1
end{indicators}
```

Note that, after the data, the group membership is defined by the sectioning commands

begin{indicators} . . . end{indicators}

with a 1 if a column belongs to group X, a -1 if the column belongs to group Y and a 0 if a column is not to be used. So in this data set the first 6 columns belong to group X while the next 7 columns belong to group Y and there are no suppressed columns.

Choosing to perform both the Mann-Whitney U and Kolmogorov-Smirnov D two sample tests leads to the following results.

Mann-Whitney U and Kolmogorov-Smirnov D tests

Variables: NX = 6, NY = 7

MW U	MW Z	MW 2-tail p	KS D	KS Z	KS 2-tail p
7.00000	-1.93389	0.981352	0.54762	0.30467	0.146853
11.0000	-1.36089	0.927739	0.52381	0.29142	0.277389
42.0000	2.92857	0.000583	1.00000	0.55635	0.000000
-1.00000	-1.00000	-1.000000	-1.00000	-1.00000	-1.000000
0.00000	-2.93260	1.000000	1.00000	0.55635	0.000000

The type of test (i.e. lower, upper, or 2-tail) and partitioning of columns into groups can be done interactively, but the Bonferroni or similar correction must be used if, as here, both tests are done. Note the setting of values to -1 with singular data as emphasized in red for row 4.

4.3.7 All pairwise comparisons on n samples

Given a set of samples it is useful to perform pairwise statistical tests to determine if any of the samples can be regarded as atypical. For instance, if the samples are normally distributed with the same variance then ANOVA followed by the Tukey post-ANOVA would be used, but SimFIT also provides a facility to perform all pairwise comparisons on sets of samples using nonparametric tests as well as t tests.

As the sample sizes may differ then individual samples could be input, however the library file approach should be used to facilitate this procedure. From the main SimFIT menu choose [Statistics] then [Data exploration] followed by [All pairwise tests] and input the library file `npcorr.tfl` which contains the following information.

```
Data for non-parametric correlation analysis
column2.tf1
column2.tf2
column2.tf3
```

This library file has a title followed by the data file names which, as they are SimFIT test files, are identified by filename only, otherwise the full path would have to be be supplied. The data contained in the three files are shown in the next table followed by the results from analysis quoting the Dunn-Sidak corrected significance levels instead of the Bonferroni ones for k procedures on n samples, i.e. considered as $kn(n-1)/2$ tests in all on the same data to test H_0: samples have the same distributions.

column2.tf1	column2.tf2	column2.tf3
1.70	1.00	0.50
4.00	2.80	3.00
0.60	6.00	2.50
9.00	1.80	6.00
0.99	4.00	2.50
2.00	1.40	5.50
1.80	9.00	7.50
7.00	2.50	0.00
0.99	5.00	3.00

Mann-Whitney-U/Kolmogorov-Smirnov-D/unpaired-t tests
Number of tests = 9, p(1%) = 0.001116, p(5%) = 0.005683 [Dunn-Sidak]

```
C:\Program Files (x86)\simfit\dem\column2.tf1
C:\Program Files (x86)\simfit\dem\column2.tf2
N1 = 9, N2 = 9    MW U = 8.00000    p = 0.002262 *
                  KS D = 0.77778    p = 0.000740 **
                   t = -3.71551     p = 0.001880 *
C:\Program Files (x86)\simfit\dem\column2.tf1
C:\Program Files (x86)\simfit\dem\column2.tf3
N1 = 9, N2 = 9    MW U = 21.0000    p = 0.088893
                  KS D = 0.55556    p = 0.033566
                   t = -2.04236     p = 0.057955
C:\Program Files (x86)\simfit\dem\column2.tf2
C:\Program Files (x86)\simfit\dem\column2.tf3
N1 = 9, N2 = 9    MW U = 55.5000    p = 0.195886
                  KS D = 0.44444    p = 0.125874
                   t = 1.46055      p = 0.163497
```

4.3.8 One sample robust analysis

It is obvious that outliers in a sample lead to biased parameters estimates. In some instances an experimenter is able to examine the data and make a decision to eliminate certain observations, usually extremely low or high values, that indicate a systematic source of variation beyond the usual spread of observational errors. Alternatively, to avoid subjective doctoring of data, a robust method can be used which generally involves discarding extreme values and using more appropriate numerical methods that do not assume that the sample is normally distributed.

As an example, choose statistics from the main SIMFIT menu, navigate to [Data exploration] and open the option for [Robust analysis of one sample]. The results from examining the test file `robust.tf1` after trimming 10% off the extreme values are shown below, followed by the results from handling the full data set without any trimming in the exhaustive analysis procedure.

Robust analysis	
Data: 50 N(0,1) random numbers with 5 outliers	
Total sample size	50
Median value	0.2019
Median absolute deviation	1.0311
Robust standard deviation	1.5288
Trimmed mean (TM)	0.2227
Variance estimate for TM	0.0192
Winsorized mean (WM)	0.2326
Variance estimate for WM	0.0192
Number of discarded values	10
Number of included values	40
Percentage of sample used	80% (for TM and WM)
Hodges-Lehmann estimate (HL)	0.2586

Exhaustive analysis	
Minimum, Maximum values	-2.208, 7.000
Lower and Upper Hinges	-0.829, 1.307
Coefficient of skewness	1.690
Coefficient of kurtosis	3.566
Median value	0.202
Sample mean	0.512
Sample standard deviation	1.853: CV% = 361.736%
Standard error of the mean	0.262
Upper 2.5% t-value	2.010
Lower 95% confidence limit for mean	-0.014
Upper 95% confidence limit for mean	1.039
Variance of the sample	3.435
Lower 95% confidence limit for variance	2.397
Upper 95% confidence limit for variance	5.335
Shapiro-Wilks W statistic	0.851
Significance level for W	0.000 Reject normality at 1% sig.level

Clearly the exhaustive analysis indicates that the presence of outliers has created a sample that is not normally distributed and the results from robust analysis yield better estimates for the population mean and variance which, before adding outliers, were $\mu = 0, \sigma^2 = 1$. An outline of the theory and definitions used in this robust analysis follows.

Theory

If the sample vector is $x_1, x_2, \ldots, x_n$ the following calculations are done.

1. Using the whole sample and the inverse normal function $\Phi^{-1}(.)$, the median M, median absolute deviation D and a robust estimate of the standard deviation S are calculated as

$$\begin{aligned} M &= \text{median}(x_i) \\ D &= \text{median}(|x_i - M|) \\ S &= D/\Phi^{-1}(0.75). \end{aligned}$$

2. The percentage of the sample chosen by users to be eliminated from each of the tails is $100\alpha\%$, then the trimmed mean TM, and Winsorized mean WM, together with variance estimates VT and VW, are calculated as follows, using $k = [\alpha n]$ as the integer part of αn.

$$\begin{aligned} TM &= \frac{1}{n-2k}\sum_{i=k+1}^{n-k} x_i \\ WM &= \frac{1}{n}\left\{\sum_{i=k+1}^{n-k} x_i + kx_{k+1} + kx_{n-k}\right\} \\ VT &= \frac{1}{n^2}\left\{\sum_{i=k+1}^{n-k} (x_i - TM)^2 + k(x_{k+1} - TM)^2 + k(x_{n-k} - TM)^2\right\} \\ VW &= \frac{1}{n^2}\left\{\sum_{i=k+1}^{n-k} (x_i - WM)^2 + k(x_{k+1} - WM)^2 + k(x_{n-k} - WM)^2\right\}. \end{aligned}$$

3. If the assumed sample density is symmetrical, the Hodges-Lehman location estimator HL can be used to estimate the center of symmetry. This is

$$HL = \text{median}\left\{\frac{x_i + x_j}{2}, 1 \le i \le j \le n\right\},$$

and it is calculated along with 95% confidence limit. This would be useful if the sample was a vector of differences between two samples X and Y for a Wilcoxon signed rank test that X is distributed $F(x)$ and Y is distributed $F(x - \theta)$.

4.3.9 Two samples robust analysis

Sometimes a robust estimate is required for the difference in location (with corresponding confidence limits) for two samples, not necessarily of the same size, but without assuming normality or any other distribution.

From the main SIMFIT menu choose [Statistics], navigate to [Data exploration] and open the option for [Robust analysis of two samples]. The two default test files are `ttest.tf4` and `ttest.tf5` with these values

ttest.tf4	ttest.tf5
134	70
146	118
104	101
119	85
124	107
161	132
107	94
83	
113	
129	
97	
123	

while analysis produces the following results.

Robust analysis of two samples	
X-sample size	12
Y-sample size	7
Difference in location	-18.501
Lower confidence limit	-40.009
Upper confidence limit	2.997
Percentage confidence limit	95.30%
Lower Mann-whitney U-value	19.000
Upper Mann-Whitney U-value	66.000

The procedure is based on the assumption that X of size n_x is distributed as $F(x)$ and Y of size n_y as $F(x-\theta)$, so an estimate $\hat{\theta}$ for the difference in location is calculated as

$$\hat{\theta} = \text{ median}(y_j - x_i, i = 1, 2, \ldots, n_x, j = 1, 2, \ldots, n_y).$$

$100\alpha\%$ confidence limits U_L and U_H are then estimated by inverting the Mann-Whitney U statistic so that

$$\begin{aligned} P(U \le U_L) &\le \alpha/2 \\ P(U \le U_L + 1) &> \alpha/2 \\ P(U \ge U_H) &\le \alpha/2 \\ P(U \ge U_H - 1) &> \alpha/2. \end{aligned}$$

4.4 Analysis of variance (ANOVA)

Tutorials and worked examples for simulation, curve fitting, statistical analysis, and plotting.
https://simfit.uk
https://simfit.org.uk
https://simfit.silverfrost.com

4.4.1 Introduction

Analysis of Variance (ANOVA) is one of the most widely used techniques in data analysis. For example, this next data set which is contained in the test file `anova.tf1` is for six replicate estimates for strontium concentrations (mg/ml) in five different locations, and it is wished to test if there are significant differences between population means based on the sample means as listed in the last row.

	28.2	39.6	46.3	41.0	56.3
	33.2	40.8	42.1	44.1	54.1
	36.4	37.9	43.5	46.4	59.4
	34.6	37.1	48.8	40.2	62.7
	29.1	43.6	43.7	38.6	60.0
	31.0	42.4	40.1	36.3	57.3
Means	32.1	40.2	44.1	41.1	58.3

In the subsequent discussion concerning ANOVA it will be assumed that the reader is familiar with the normal, chi-square, and F distributions, and statistical tests based on them as described in the appropriate SimFIT tutorial documents. In particular, the Shapiro-Wilks test for normality and the Bartlett or Levene tests for homogeneity of variance could be used by purists determined to check if ANOVA is justified, because it should be pointed out that ANOVA is often used uncritically where better techniques may be more appropriate.

Theory

In studying the distribution of the variance estimate from a sample of size n from a normal distribution with mean μ and variance σ^2, you will have encountered the following decomposition of a sum of squares

$$\sum_{i=1}^{n}\left(\frac{y_i-\mu}{\sigma}\right)^2=\sum_{i=1}^{n}\left(\frac{y_i-\bar{y}}{\sigma}\right)^2+\left(\frac{\bar{y}-\mu}{\sigma/\sqrt{n}}\right)^2$$

into independent chi-square variables with $n-1$ and 1 degrees of freedom respectively. Analysis of variance is an extension of this procedure based on linear models, assuming normality and constant variance, then partitioning of chi-square variables into two or more independent components, invoking Cochran's theorem and comparing the ratios to F variables with the appropriate degrees of freedom for variance ratio tests. It can be used, for instance, when you have a set of samples (column vectors) that come from normal distributions with the same variance and wish to test if all the samples have the same mean. Due to the widespread use of this technique, many people use it even though the original data are not normally distributed with the same variance, by first applying variance stabilizing transformations, like the square root with counts, which can sometimes transform non-normal data into transformed data that are approximately normally distributed. Note that you should never make the common mistake of supposing that ANOVA is model free: ANOVA is always based upon data collected as replicates and organized into groups, where it is assumed that all the data are normally distributed with the same variance but with mean values that differ from cell to cell according to an assumed linear model.

Variance stabilizing transformations

A number of transformations are in use that attempt to create new data that is more approximately normally distributed than the original data, or at least has more constant variance, as the two aims can not usually both

be achieved. If the distribution of a random variable X is known, then the variance of a function of X can in some cases be calculated explicitly. However, to a very crude first approximation, if a random variable X is transformed by $Y = f(X)$, then the variances are related by the differential equation

$$V(Y) \approx \left(\frac{dY}{dX}\right)^2 V(X)$$

which yields $f(.)$ on integration, e.g. for constant variance where $V(Y) = k$ for some constant k would be required, given $V(X)$.

Note that SIMFIT provides the ability to explore the commonly used transformations, to be discussed next, whenever ANOVA or tests for homogeneity of variance are used.

The angular transformation

This arcsine transformation is sometimes used for binomial data with parameters N and p, e.g., for X successes in N trials, when

$$\begin{aligned} X &\sim b(N, p) \\ Y &= \arcsin(\sqrt{X/N}) \\ E(Y) &\simeq \arcsin(\sqrt{p}) \\ V(Y) &\simeq 1/(4N) \text{ (using radial measure).} \end{aligned}$$

However, note that the variance of the transformed data is only constant in situations where there are constant binomial denominators.

The square root transformation

This is often used for counts, e.g., for Poisson variables with mean μ, when

$$\begin{aligned} X &\sim \text{Poisson}(\mu) \\ Y &= \sqrt{x} \\ E(Y) &\simeq \sqrt{\mu} \\ V(Y) &\simeq 1/4. \end{aligned}$$

The log transformation

When the variance of X is proportional to a known power α of $E(X)$, then the power transformation $Y = X^\beta$ will stabilize variance for $\beta = 1 - \alpha/2$. The angular and square root transformations are, of course, just special cases of this, but a singular case of interest is the constant coefficient of variation situation $V(X) \propto E(X)^2$ which justifies the log transform, as follows

$$\begin{aligned} E(X) &= \mu \\ V(X) &\propto \mu^2 \\ Y &= \log X \\ V(Y) &= k, \text{ a constant.} \end{aligned}$$

Introduction to 1-way ANOVA

As this is the most frequently encountered situation and is the model for subsequent variants it will be discussed in some detail.

This procedure is used when you have groups (i.e. samples) of normally distributed measurements with the same variance and wish to test if all the population means are equal. With two groups it is equivalent to the two-sample unpaired t test, so it can be regarded as an extension of this test to cases with more than two groups. Suppose a random variable Y is measured for groups $i = 1, 2, \ldots, k$ and subjects $j = 1, 2, \ldots n_i$, and it is assumed that the appropriate general linear model for the $n = \sum_{i=1}^{k} n_i$ observations is

$$y_{ij} = \mu + \alpha_i + e_{ij}$$
$$\text{where } \sum_{i=1}^{k} \alpha_i = 0$$

and the errors e_{ij} are independently normally distributed with zero mean and common variance σ^2.

Then the 1-way ANOVA null hypothesis is

$$H_0 : \alpha_i = 0, \text{ for } i = 1, 2, \ldots, k,$$

that is, the means for all k groups are equal, and the basic equations are as follows.

$$\bar{y}_i = \sum_{j=1}^{n_i} y_{ij}/n_i$$
$$\bar{y} = \sum_{i=1}^{k} \sum_{j=1}^{n_i} y_{ij}/n$$
$$\sum_{i=1}^{k} \sum_{j=1}^{n_i} (y_{ij} - \bar{y})^2 = \sum_{i=1}^{k} \sum_{j=1}^{n_i} (y_{ij} - \bar{y}_i)^2 + \sum_{i=1}^{k} n_i (\bar{y}_i - \bar{y})^2$$
$$\text{Total } SSQ = \sum_{i=1}^{k} \sum_{j=1}^{n_i} (y_{ij} - \bar{y})^2, \text{ with } DF = n - 1$$
$$\text{Residual } SSQ = \sum_{i=1}^{k} \sum_{j=1}^{n_i} (y_{ij} - \bar{y}_i)^2, \text{ with } DF = n - k$$
$$\text{Group } SSQ = \sum_{i=1}^{k} n_i (\bar{y}_i - \bar{y})^2, \text{ with } DF = k - 1.$$

Here Total SSQ is the overall sum of squares, Group SSQ is the between groups (i.e. among groups) sum of squares, and Residual SSQ is the residual (i.e. within groups, or error) sum of squares. The mean sums of squares and F value can be calculated from these using

$$\text{Total } SSQ = \text{ Residual } SSQ + \text{ Group } SSQ$$
$$\text{Total } DF = \text{ Residual } DF + \text{ Group } DF$$
$$\text{Group } MS = \frac{\text{Group } SSQ}{\text{Group } DF}$$
$$\text{Residual MS } = \frac{\text{Residual } SSQ}{\text{Residual } DF}$$
$$F = \frac{\text{Group } MS}{\text{Residual } MS},$$

so that the degrees of freedom for the F variance ratio to test if the between groups MS is significantly larger than the residual MS are $k - 1$ and $n - k$. The SimFIT 1-way ANOVA procedure allows you to include or exclude selected groups, i.e., data columns, and to employ variance stabilizing transformations if required, but it also provides a nonparametric test, and it allows you to explore which group or groups differ significantly in the event of the F value leading to a rejection of H_0.

4.4.2 1-way ANOVA

Example 1

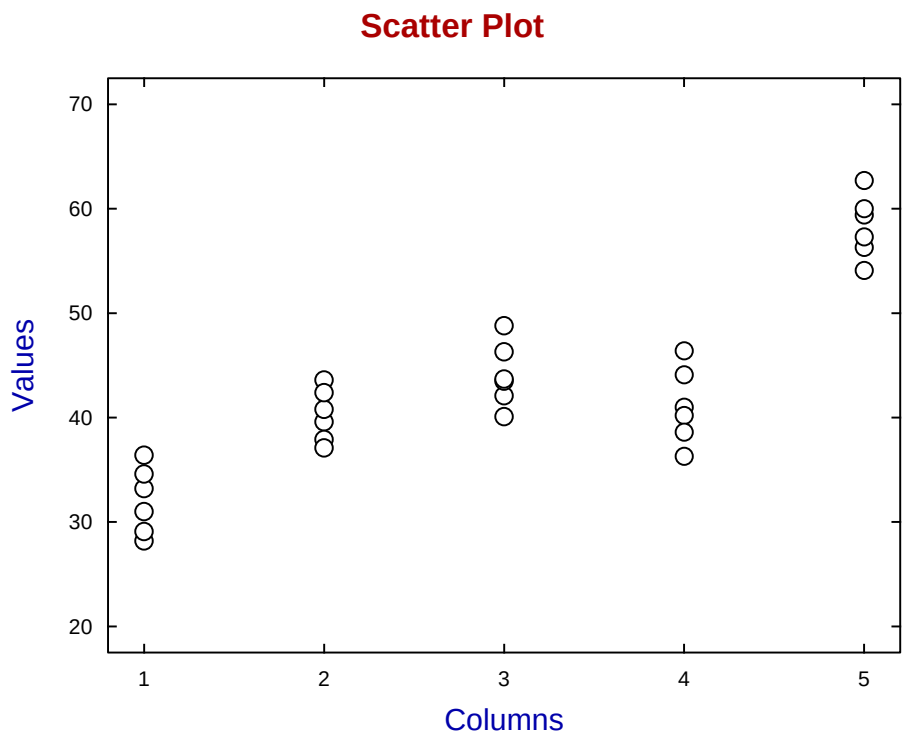
Open the SIMFIT main menu, select the [Statistics] option, choose 1-way-ANOVA, indicate that untransformed data are to be used, then analyze the test file provided which is a data matrix contained in `anova.tf1`. This particular data set is for six replicate estimates for strontium concentrations (mg/ml) in five different locations, and it is wished to test if there are significant differences between the mean levels as listed in the last row.

	28.2	39.6	46.3	41.0	56.3
	33.2	40.8	42.1	44.1	54.1
	36.4	37.9	43.5	46.4	59.4
	34.6	37.1	48.8	40.2	62.7
	29.1	43.6	43.7	38.6	60.0
	31.0	42.4	40.1	36.3	57.3
Means	32.1	40.2	44.1	41.1	58.3

The results, followed by a scatter plot, are as follows.

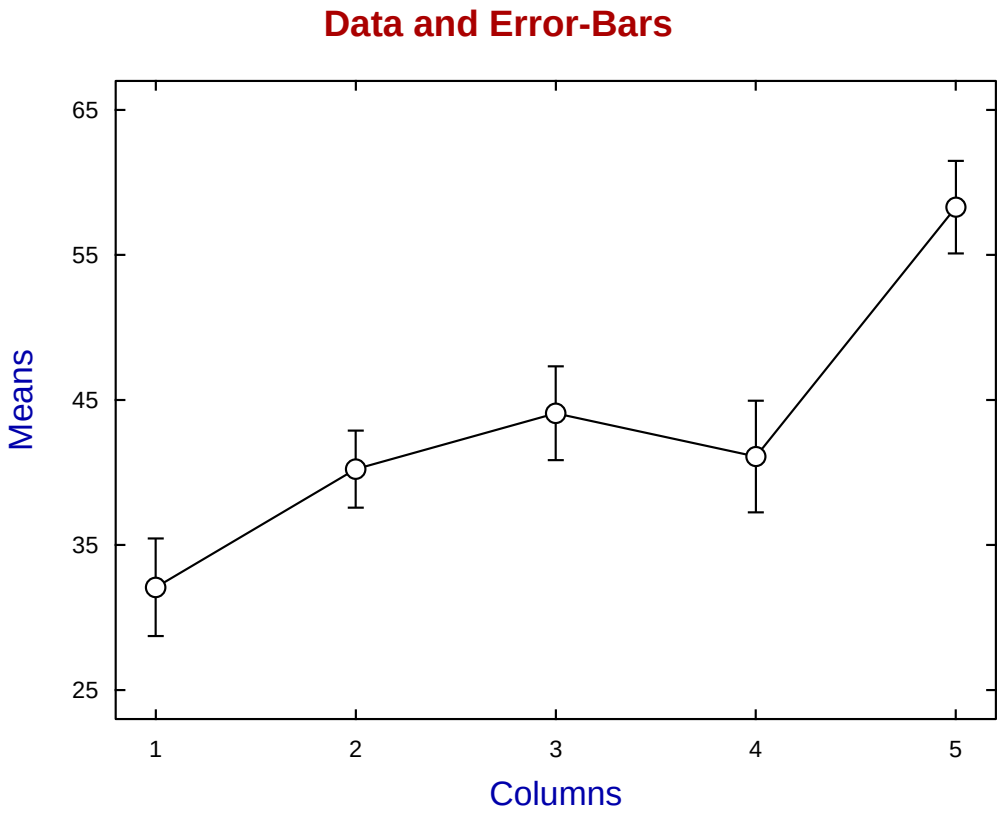
1-Way Analysis of Variance: Grand Mean 43.16
Transformation: x (untransformed data)

Source	SSQ	NDOF	MSQ	F	p
Between Groups	2193	4	548.4	56.15	0.0000
Residual	244.1	25	9.765		
Total	24383	29			



Clearly the null hypothesis of equal column means must be rejected at the 1% significance level as $p < 0.01$. However, this does not tell us which columns are significantly different from the rest, only that at least one column differs significantly. The previous scatter plot does however suggests that column 5 appears atypical, and possibly column 1 also.

Another way to explore this data set is to plot the means with error bars representing the 95% confidence limits as follows.



Example 2

The sample sizes need not be identical for 1-way ANOVA, and the next case to be considered is where there are 5 groups of sizes 5, 8, 6, 8, and 8 for weight gain in pounds of pigs from 5 different litters.

23	29	38	30	31
27	25	31	27	33
26	33	28	28	31
19	36	35	22	28
30	32	33	33	30
	28	36	34	24
	30		34	29
	31		32	30

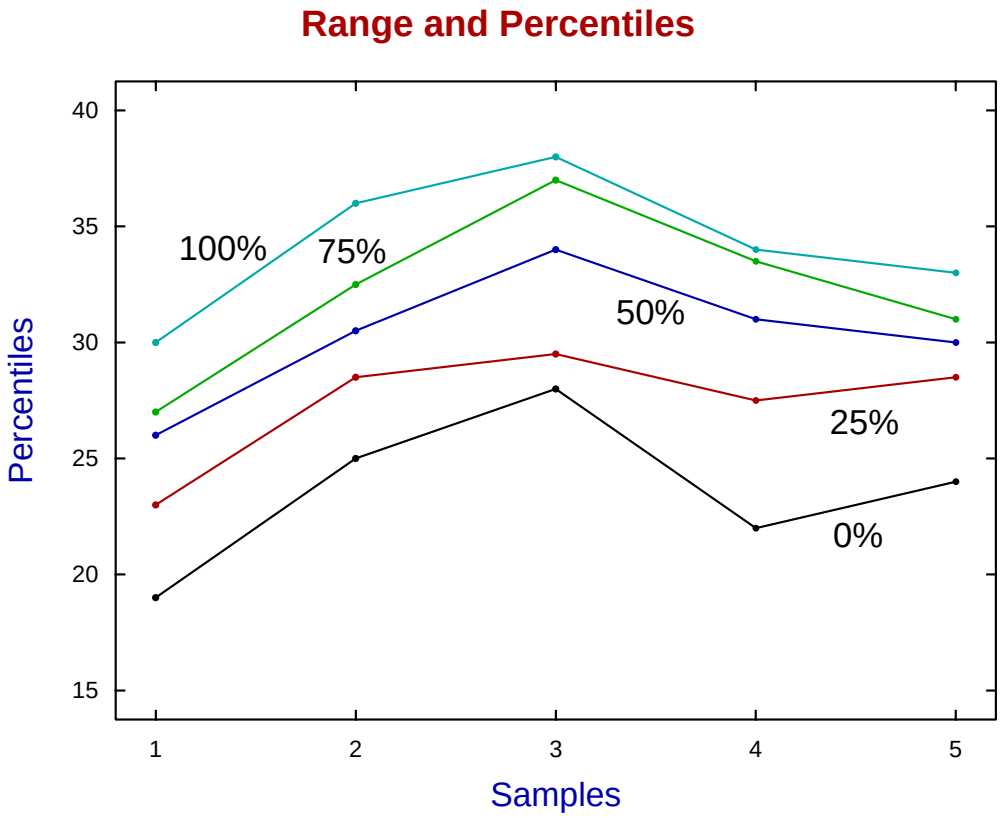
As the sample sizes differ the data cannot be entered as a matrix this time, and must be entered as individual column vectors, from a project archive, or as a library file which simply holds the locations of individual data files for each of the columns.

So now repeat the above procedure, but this time select to supply a library file and input the test file `anova.TFL` which then reads in data from the test files `column1.tf1`, `column1.tf2`, ..., `column1.tf5`.

1-Way Analysis of Variance: Grand Mean 29.89
Transformation: x (untransformed data)

Source	SSQ	NDOF	MSQ	F	p
Between Groups	202.0	4	50.51	3.931	0.0111
Residual	385.5	30	12.85		
Total	587.5	34			

There is yet another way to display 1-way ANOVA data as illustrated by the next plot.



Here the lowest line segments join the lowest sample value for the corresponding groups, the upper line segments join the largest sample values, while between them the line segments join the points corresponding to the 25%, 50%, and 75% levels.

This time the results suggest rejecting the null hypothesis of equal means at the 5% significance level as $p < 0.05$, but not the 1% significance level as $p > 0.01$.

The Tukey post-ANOVA Q test to further illuminate the results from this type of analysis will be described in another tutorial document.

4.4.3 1-way ANOVA (Kruskal-Wallis nonparametric)

If it is clear that the data are not normally distributed with the same variance then it is possible to perform the nonparametric Kruskal-Wallis test, either alone, or at the same time as 1-way ANOVA to compare the results. Of course, it would be usual to pre-determine which result to accept, otherwise the Bonferroni principle would have to be used.

Example 1

Open the SIMFIT main menu, select the [Statistics] option, choose 1-way-ANOVA, indicate that untransformed data are to be used, then analyze the test file provided which is a data matrix contained in `anova.tf1`. This particular data set is for six replicate estimates for strontium concentrations (mg/ml) in five different locations, and it is wished to test if there are significant differences between the mean levels as listed in the last row.

	28.2	39.6	46.3	41.0	56.3
	33.2	40.8	42.1	44.1	54.1
	36.4	37.9	43.5	46.4	59.4
	34.6	37.1	48.8	40.2	62.7
	29.1	43.6	43.7	38.6	60.0
	31.0	42.4	40.1	36.3	57.3
Means	32.1	40.2	44.1	41.1	58.3

The results are as follows.

1-Way Analysis of Variance: Grand Mean 43.16
Transformation: x (untransformed data)

Source	SSQ	NDOF	MSQ	F	p
Between Groups	2193	4	548.4	56.15	0.0000
Residual	244.1	25	9.765		
Total	24383	29			

Kruskal-Wallis Nonparametric One Way Analysis of Variance

Test statistic	NDOF	p
23.30	4	0.0001

Clearly the null hypothesis of equal column means and medians must be rejected at the 1% significance level as $p < 0.01$ for both parametric and nonparameteric 1-way ANOVA.

Example 2

The sample sizes need not be identical for 1-way ANOVA, and the next case to be considered is where there are 5 groups of sizes 5, 8, 6, 8, and 8 for weight gain in pounds of pigs from 5 different litters.

23	29	38	30	31
27	25	31	27	33
26	33	28	28	31
19	36	35	22	28
30	32	33	33	30
	28	36	34	24
	30		34	29
	31		32	30

As the sample sizes differ, the data cannot be entered as a matrix this time, and must be entered as individual column vectors, from a project archive, or as a library file which simply holds the locations of individual data files for each of the columns.

So now repeat the above procedure, but this time select to supply a library file and input the test file `anova.TFL` which then reads in data from the test files `column1.tf1`, `column1.tf2`, ..., `column1.tf5`.

1-Way Analysis of Variance: Grand Mean 29.89
Transformation: x (untransformed data)

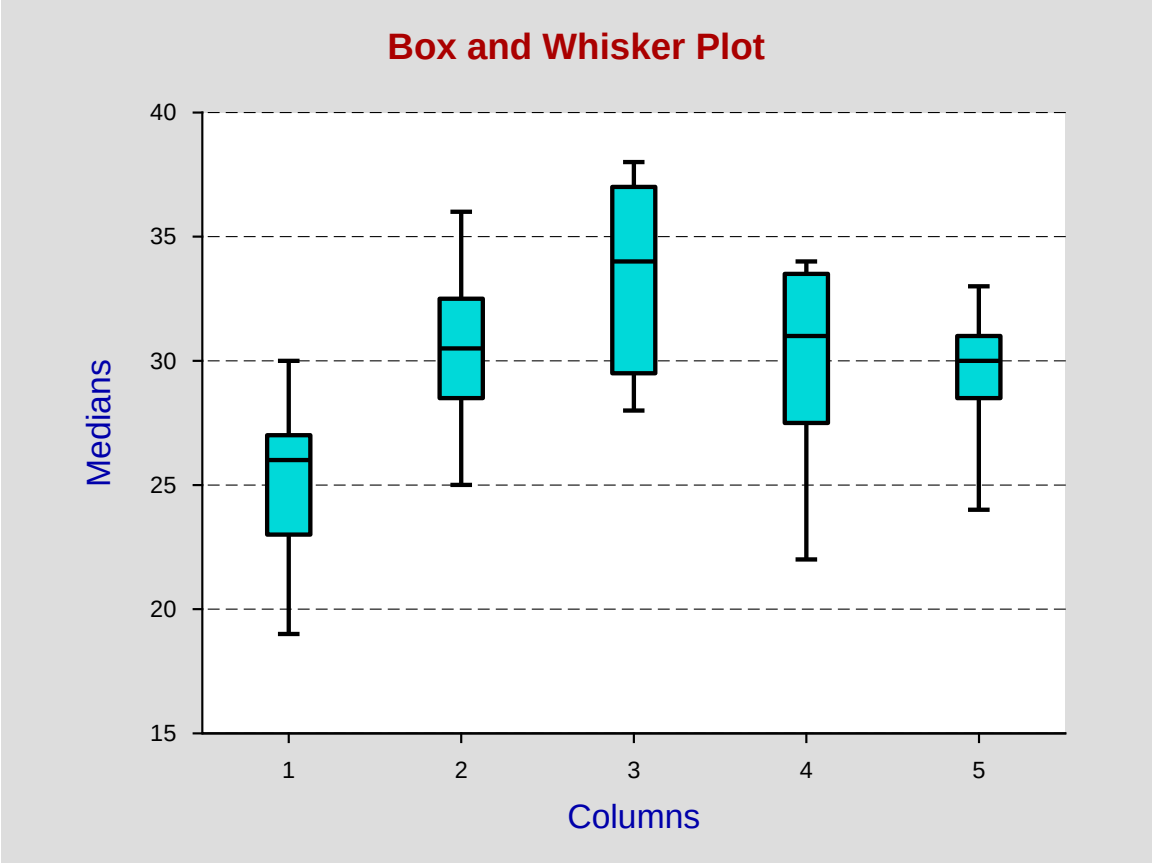
Source	SSQ	NDOF	MSQ	F	p
Between Groups	202.0	4	50.51	3.931	0.0111
Residual	385.5	30	12.85		
Total	587.5	34			

Kruskal-Wallis Nonparametric One Way Analysis of Variance

Test statistic	NDOF	p
10.54	4	0.0323

Clearly the null hypothesis of equal column means and medians must be rejected at the 5% significance level, but not the 1% level, as $p < 0.05$ for both the parametric and nonparameteric 1-way ANOVA.

There is yet another way to display 1-way ANOVA data as illustrated by the next plot.



This is more in keeping with a nonparameteric test and displays the ranges and medians as a box and whisker plot, that is: the lowest value, the 25% point, the 50% median point, the 75% point, and the largest value.

The Kruskal-Wallis test

The null hypothesis for standard 1-way ANOVA is

H_0: The groups (i.e., column vectors) are from the same normal distribution,

while for the Kruskal-Wallis analysis of variance by ranks it is the weaker condition

H_0: The groups (i.e., column vectors) are from the same distribution.

The Kruskal-Wallis test is in reality an extension of the Mann-Whitney U test to k independent samples, and it is actually designed to test H_0: the medians are all equal.

The pooled sample is ranked, with tied scores assigned average ranks, then a test statistic H for k groups, each with n_i observations, is calculated as

$$H = \frac{12}{n(n+1)} \sum_{i=1}^{k} \frac{R_i^2}{n_i} - 3(n+1)$$

$$\text{where } n = \sum_{i=1}^{k} n_i$$

and R_i is the sum of the ranks of the n_i observations in group i.

This test is actually a 1-way ANOVA carried out on the ranks of the data. The p value are calculated exactly for small samples, but the fact that H approximately follows a χ^2_{k-1} distribution is used for large samples.

If there are ties, then H is corrected by dividing by λ defined as

$$\lambda = 1 - \frac{\sum_{i=1}^{m} (t_i^3 - t_i)}{n^3 - n}$$

where t_i is the number of tied scores in the ith group of ties, and m is the number of groups of tied ranks.

The test is $3/\pi$ (i.e, 95%) as powerful as the 1-way ANOVA test when the parametric test is justified, but it is more powerful, and should always be used if the assumptions of the linear normal model are not appropriate.

As it is unusual for the sample sizes to be large enough to verify that all the samples are normally distributed and with the same variance, rejection of H_0 in the Kruskal-Wallis test (which is the higher order analogue of the Mann-Whitney U test, just as 1-way ANOVA is the higher analogue of the t test) should always be taken seriously. This is one reason why SimFIT provides the convenient option to perform both 1-way ANOVA and the Kruskal-Wallis test at the same time.

4.4.4 Tukey Q post-ANOVA test

The ANOVA 1-way analysis tests for equality between the means in a number of samples but, if the null hypothesis is rejected, it does not indicate for which samples the means are significantly different. In fact, for k samples there are $k(k-1)/2$ possible pairwise comparisons so alternative techniques have been developed. The SIMFIT ANOVA procedure provides the option to use the Tukey Q post-ANOVA test for this multiple comparison purpose.

Open the SIMFIT main menu, select the [Statistics] option, choose 1-way-ANOVA, indicate that untransformed data are to be used, then analyze the test file provided which is a data matrix contained in `anova.tf1`. This particular data set is for six replicate estimates for strontium concentrations (mg/ml) in five different locations, and it is wished to test if there are significant differences between the mean levels as listed in the last row.

	28.2	39.6	46.3	41.0	56.3
	33.2	40.8	42.1	44.1	54.1
	36.4	37.9	43.5	46.4	59.4
	34.6	37.1	48.8	40.2	62.7
	29.1	43.6	43.7	38.6	60.0
	31.0	42.4	40.1	36.3	57.3
Means	32.1	40.2	44.1	41.1	58.3

The results are as follows.

1-Way Analysis of Variance: Grand Mean 43.16
Transformation: x (untransformed data)

Source	SSQ	NDOF	MSQ	F	p
Between Groups	2193	4	548.4	56.15	0.0000
Residual	244.1	25	9.765		
Total	24383	29			

Clearly the null hypothesis of equal column means must be rejected at the 1% significance level as $p < 0.01$ so the following results were obtained for the Tukey test.

Tukey Q-test with 5 means and 10 comparisons
5% point = 4.189, 1% point = 5.125

Column	Column	Q	p	5%	1%	n_B	n_A
5	1	20.55	0.0001	*	*	6	6
5	2	14.16	0.0001	*	*	6	6
5	4	13.48	0.0001	*	*	6	6
5	3	11.14	0.0001	*	*	6	6
3	1	9.406	0.0001	*	*	6	6
3	2	3.018	0.2377	NS	NS	6	6
3	4	[[2.338	0.4792]]	No-Test	No-Test	6	6
4	1	7.068	0.0005	*	*	6	6
4	2	[[0.6793	0.9885]]	No-Test	No-Test	6	6
2	1	6.388	0.0013	*	*	6	6

[5%] and/or [[1%]] No-Test results given for reference only

In this table, columns where means differ significantly are indicated by *, columns where means are not significantly different are indicated by NS, and columns that were not tested are indicated by No-test with hypothetical p values in square brackets.
Note that, for the Tukey test, the means are ranked and columns with means between those of extreme columns that differ significantly are not tested, according to the protocol that is recommended for this test. This involves a systematic procedure where the largest mean is compared to the smallest, then the largest mean is compared with the second largest, and so on. If no difference is found between two means then it is concluded that no difference exists between any means enclosed by these two, and so no testing is done.

The test statistic Q for comparing columns A and B with means $\bar{y_A}$, $\bar{y_B}$ and sample sizes n_A, n_B is

$$Q = \frac{\bar{y_B} - \bar{y_A}}{SE}$$

$$\text{where } SE = \sqrt{\frac{s^2}{n}}, \text{ if } n = n_A = n_B$$

$$SE = \sqrt{\frac{s^2}{2}\left(\frac{1}{n_A} + \frac{1}{n_B}\right)}, \text{ if } n_A \neq n_B$$

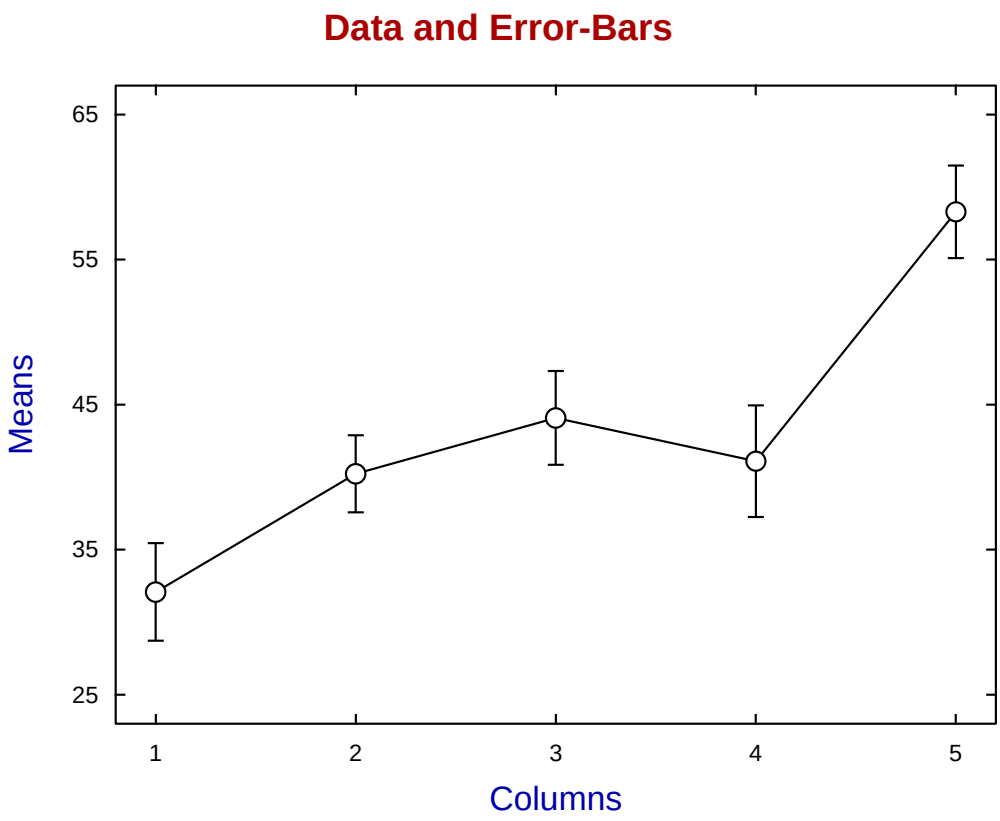
$$s^2 = \text{ error } MS$$

and the significance level for Q is calculated as a studentized range.

Evidently, for these data, we reach the conclusion that

- data in column 5 differs significantly from data in columns 1, 2, 3, and 4,
- data in column 3 differs significantly from data in column 1, and
- data in column 1 differs significantly from data in columns 2 and 4.

This conclusion is also fairly obvious from the plot of means with error bars representing the 95% confidence limits as follows.



4.4.5 2-way ANOVA

The difference between 1-way ANOVA and 2-way ANOVA is that, whereas 1-way ANOVA only tests for differences between column means, 2-way ANOVA also considers the possibility of effects dependent on the rows. Note that, because the number of observations is the same for each subject, the data can be input as a data matrix into the SimFIT file selection control either from a data file, by typing in from the keyboard, or by copying and pasting from a spreadsheet.

Worked example for treatments and clotting times

From the main SimFIT menus choose [Statistics] followed by [ANOVA] then select [2-way ANOVA] and, instead of using the default test file `anova2.tf1`, use the [Browse] feature on the file selection control to search for and then open the SimFIT test file `anova2.tf2` which has the following data set.

	Treatment			
Subject	1	2	3	4
1	8.40	9.40	9.80	12.2
2	12.8	15.2	12.9	14.4
3	9.60	9.10	11.2	9.80
4	9.80	8.80	9.90	12.0
5	8.40	8.20	8.50	8.50
6	8.60	9.90	9.80	10.9
7	8.90	9.00	9.20	10.4
8	7.90	8.10	8.20	10.0

These are clotting times in minutes from eight subjects treated by four methods and analysis leads to the following results, indicating both subject and treatment dependent effects.

2-Way Analysis of Variance: Grand mean 9.994

Source	SSQ	NDOF	MSSQ	F	p
Between rows (Subjects)	78.99	7	11.28	17.20	0.0000
Between columns (Treatments)	13.02	3	4.339	6.615	0.0025
Residual	13.77	21	0.6559		
Total	105.8	31			

Note that now there are variance ratio test statistics F and corresponding p values for both the rows and the columns and the calculations involved in constructing this ANOVA 2-way table follow.

The assumed linear model

The 2-way ANOVA procedure is used when you want to include row and column effects in a completely randomized design, i.e., assuming no interaction and one replicate per cell so that the appropriate linear model is

$$y_{ij} = \mu + \alpha_i + \beta_j + e_{ij}$$

$$\sum_{i=1}^{r} \alpha_i = 0$$

$$\sum_{j=1}^{c} \beta_j = 0$$

for a data matrix with r rows and c columns, i.e. $n = rc$.

Calculating the variance ratio statistics

The mean sums of squares and degrees of freedom for row and column effects are worked out, then the appropriate F and p values are calculated. Using R_i for the row sums, C_j for the column sums, and $T = \sum_{i=1}^{r} R_i = \sum_{j=1}^{c} C_j$ for the sum of observations, these are

$$\text{Row } SSQ = \sum_{i=1}^{r} R_i^2/c - T^2/n, \text{ with } DF = r - 1$$

$$\text{Column } SSQ = \sum_{j=1}^{c} C_j^2/r - T^2/n, \text{ with } DF = c - 1$$

$$\text{Total } SSQ = \sum_{i=1}^{r}\sum_{j=1}^{c} y_{ij}^2 - T^2/n, \text{ with } DF = n - 1$$

$$\text{Residual } SSQ = \text{Total } SSQ - \text{Row } SSQ - \text{Column } SSQ, \text{ with } DF = (r-1)(c-1)$$

where Row SSQ is the between rows sums of squares, Column SSQ is the between columns sum of squares, Total SSQ is the total sum of squares and Residual SSQ is the residual, or error sum of squares. Now two F statistics can be calculated from the mean sums of squares as

$$F_R = \frac{\text{Rows } MS}{\text{Residual } MS}$$
$$F_C = \frac{\text{Column } MS}{\text{Residual } MS}.$$

The statistic F_R is compared with $F(r-1,(r-1)(c-1))$ to test

$$H_R : \alpha_i = 0, i = 1, 2, \ldots, r$$

i.e., absence of row effects, while F_C is compared with $F(c-1,(r-1)(c-1))$ to test

$$H_C : \beta_j = 0, j = 1, 2, \ldots, c$$

i.e., absence of column effects.

As with 1-way ANOVA, normality is assumed, and the technique can be extended to the case with replicates if it is wished to study variation within cells.

4.4.6 2-way ANOVA (Friedman nonparametric)

When the 2-way ANOVA assumptions are not justified, the Friedman nonparametric 2-way analysis of variance by ranks is often used. This investigates the score differences between k matched sets of size l. If $k = 2$ then the sign test, or else the Wilcoxon signed rank test, should be used.

From the main SimFIT menu choose [Statistics], [ANOVA], then the Friedman test, and read in data from the default test file `anova2.tf1`, which has data for scores for matched samples of eighteen rats under three different patterns of enforcement as follows.

1.00	3.00	2.00
2.00	3.00	1.00
1.00	3.00	2.00
1.00	2.00	3.00
3.00	1.00	2.00
2.00	3.00	1.00
3.00	2.00	1.00
1.00	3.00	2.00
3.00	1.00	2.00
3.00	1.00	2.00
2.00	3.00	1.00
2.00	3.00	1.00
3.00	2.00	1.00
2.00	3.00	1.00
2.50	2.50	1.00
3.00	2.00	1.00
3.00	2.00	1.00
2.00	3.00	1.00

Analysis then leads to the results below.

Friedman Nonparametric 2-way ANOVA	
Test Statistic (FR)	8.583
Number of degrees of freedom	2
Significance (i.e., p-value)	0.0137

As the data matrix represents scores rather than normally distributed variables with identical variances, the matrix was analyzed as a two way table using the nonparametric Friedman 2-way ANOVA procedure to test

H_0 : all medians are equal, against the alternative,

H_1 : they come from different populations.

For this analysis SimFIT first rearranges these data into a $k = 3$ by $n = 18$ matrix, then ranks column scores for this transposed matrix as r_{ij} for row i and column j, assigning average ranks for ties, works out rank sums as $t_i = \sum_{j=1}^{l} r_{ij}$, then calculates FR given by

$$FR = \frac{12}{nk(k+1)} \sum_{i=1}^{k} (t_i - n(k+1)/2)^2 .$$

For small samples, exact significance levels are calculated, while for large samples it is assumed that FR follows a χ^2_{k-1} distribution.

4.4.7 Repeat measures ANOVA

Repeat measures ANOVA is a special type of 2-way ANOVA where the rows in the data matrix are subjects but the columns are now repeated observations of the same variable in some sequence, for instance at fixed intervals of time. It is usual to investigate the data for sphericity, which is when the covariance matrix of orthonormal contrasts is a multiple of the identity matrix, as this is required before the repeat measures ANOVA procedure is valid. Note that SIMFIT also provides the options for Hotelling T^2 and Friedman nonparametric ANOVA tests at the same time, in case the hypothesis of sphericity is not supported but the indication of a column effect is still of interest.

Open the main SIMFIT menu, choose [Statistics], [ANOVA], then repeat measures, and analyze the default data set contained in test file `anova6.tf1`, which has four measurements of the same variable for each of five subjects arranged as follows.

Subject	Measurement 1	Measurement 2	Measurement 3	Measurement 4
A	30	28	16	34
B	14	18	10	22
C	24	20	18	30
D	38	34	20	44
E	26	28	14	30

Now choose to analyze without a data transformation which leads to the following result where a likelihood ratio test statistic ($LRTS$) is calculated to test for sphericity.

Repeat-Measures Analysis of Variance

Data file: `anova6.tf1`

Sphericity test on CV of Helmert orthonormal contrasts
H_0: Covariance matrix = k*Identity (for some $k > 0$)

Number of small eigenvalues	0	i.e. $< 1.00E-07$
Number of variables (m)	4	
Sample size (n)	5	
Determinant of CV	154.9	
Trace of CV	28.20	
Mauchly W statistic	0.1865	
$LRTS(-2\log(\lambda))$	4.572	
Degrees of Freedom	5	
$P(\chi^2 \geq LRTS)$	0.4704	
e(Geisser-Greenhouse)	0.6049	
e (Huynh-Feldt)	1.0000	
e (lower bound)	0.3333	

Clearly the hypothesis of sphericity cannot be rejected for these data.

The next table displays the ANOVA results with, in this example, the optional Friedman nonparametric test, and Hotelling T^2 test also included.

Results for repeat-measures ANOVA: Grand mean 24.90

Source	SSQ	NDOF	MSSQ	F	p	
Subjects	6.808E+02	4				
Treatments	6.982E+02	3	232.7	24.76	0.0000	
					0.0006	Greenhouse-Geisser
					0.0000	Huyhn-Feldt
					0.0076	Lower-bound
Remainder	112.8	12	9.400			
Total	1492	19				

Results for Friedman Nonparametric Two-Way Analysis of Variance	
Test Statistic	13.56
Number of degrees of freedom	3
Significance	0.0036

Results for the Hotelling one sample T^2 test
H_0: Column means are all equal

Number of rows	5	
Number of columns	4	
Hotelling T^2	170.5	
F Statistic (FTS)	28.41	
Degrees of Freedom ($d1, d2$)	3, 2	
$P(F(d1, d2) \geq FTS)$	0.0342	*Reject H_0 at 5% significance level*

Note that, for these data, all three tests reject the null hypothesis of the absence of a column effect.

Theory

The repeat measures procedure is used when you have paired measurements, and wish to test for absence of treatment effects. With two samples it is equivalent to the two-sample paired t test, so it can be regarded as an extension of this test to cases with more than two columns. If the rows of a data matrix represent the effects of different column-wise treatments on the same subjects, so that the values are serially correlated, and it is wished to test for significant treatment effects irrespective of differences between subjects, then repeated-measurements design is appropriate. The simplest, model-free, approach is to treat this as a special case of 2-way ANOVA where only between-column effects are considered and between-row effects, i.e., between subject variances, are expected to be appreciable, but are not considered. Many further specialized techniques are also possible, when it is reasonable to attempt to model the treatment effects, e.g., when the columns represent observations in sequence of, say, time or drug concentration, but often such effects are best fitted by nonlinear rather than linear models. A useful way to visualize repeated-measurements ANOVA data with small samples ($\leq$ 12 subjects) is to input the matrix into the exhaustive analysis of a matrix procedure and plot the matrix with rows identified by different symbols.

The previous tables show the results from analyzing data in the test file `anova6.tf1` in three sections, a Mauchly sphericity test, an ANOVA table, and a Hotelling T^2 test, all of which will now be discussed.

In order for the normal two-way univariate ANOVA to be appropriate, sphericity of the covariance matrix of orthonormal contrasts is required. The test is based on a orthonormal contrast matrix, for example a Helmert matrix of the form

$$C = \begin{pmatrix} 1/\sqrt{2} & -1/\sqrt{2} & 0 & 0 & 0 & \dots \\ 1/\sqrt{6} & 1/\sqrt{6} & -2/\sqrt{6} & 0 & 0 & \dots \\ 1/\sqrt{12} & 1/\sqrt{12} & 1/\sqrt{12} & -3/\sqrt{12} & 0 & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \end{pmatrix}$$

which, for m columns, has dimensions $m-1$ by m, and where every row sum is zero, every row has length unity, and all the rows are orthogonal. Such Helmert conrasts compare each successive column mean with the average of the preceding (or following) column means but, in the subsequent discussion, any orthonormal contrast matrix leads to the same end result, namely, when the covariance matrix of orthonormal contrasts satisfies the sphericity condition, then the sums of squares used to construct the F test statistics will be independent chi-square variables and the two-way univariate ANOVA technique will be the most powerful technique to test for equality of column means.

The sphericity test uses the sample covariance matrix S to construct the Mauchly W statistic given by

$$W = \frac{|CSC^T|}{[Tr(CSC^T)/(m-1)]^{m-1}}.$$

If S is estimated with ν degrees of freedom then

$$\chi^2 = -\left[\nu - \frac{2m^2-3m+3}{6(m-1)}\right]\log W$$

is approximately distributed as chi-square with $m(m-1)/2-1$ degrees of freedom. Clearly, the results in the previous tables show that the hypothesis of sphericity cannot be rejected, and the results from two-way ANOVA can be tentatively accepted. However, in some instances, it may be necessary to alter the degrees of freedom for the F statistics as discussed next.

The model for univariate repeated measures with m treatments used once on each of n subjects is a mixed model of the form

$$y_{ij} = \mu + \tau_i + \beta_j + e_{ij},$$

where τ_i is the fixed effect of treatment i so that $\sum_{i=1}^{m}\tau_i = 0$, and β_j is the random effect of subject j with mean zero, and $\sum_{j=1}^{n}\beta_j = 0$. Hence the decomposition of the sum of squares is

$$\sum_{i=1}^{m}\sum_{j=1}^{n}(y_{ij}-\bar{y}_{.j})^2 = n\sum_{i=1}^{m}(\bar{y}_{i.}-\bar{y}_{..})^2 + \sum_{i=1}^{m}\sum_{j=1}^{n}(y_{ij}-\bar{y}_{i.}-\bar{y}_{.j}+\bar{y}_{..})^2,$$

that is

$$SSQ_{\text{Within subjects}} = SSQ_{\text{treatments}} + SSQ_{\text{Error}}$$

with degrees of freedom

$$n(m-1) = (m-1) + (m-1)(n-1).$$

To test the hypothesis of no treatment effect, that is

$$H_0 : \tau_i = 0 \text{ for } i = 1, 2, \ldots, m,$$

the appropriate test statistic would be

$$F = \frac{SSQ_{\text{treatment}}/(m-1)}{SSQ_{\text{Error}}/[(m-1)(n-1)]}$$

but, to make this test more robust, it may be necessary to adjust the degrees of freedom when calculating critical levels. In fact the degrees of freedom should be taken as

$$\begin{aligned}\text{Numerator degrees of freedom} &= \epsilon(m-1)\\ \text{Denominator degrees of freedom} &= \epsilon(m-1)(n-1)\end{aligned}$$

where there are four possibilities for the correction factor ϵ, all with $0 \le \epsilon \le 1$.

1. The default epsilon.
   This is $\epsilon = 1$, which is the correct choice if the sphericity criterion is met.

2. The Greenhouse-Geisser epsilon.
   This is
$$\epsilon = \frac{(\sum_{i=1}^{m-1} \lambda_i)^2}{(m-1)\sum_{i=1}^{m-1} \lambda_i^2}$$
   where λ_i are the eigenvalues of the covariance matrix of orthonormal contrasts, and it could be used if the sphericity criterion is not met, although some argue that it is an ultraconservative estimate.

3. The Huyhn-Feldt epsilon.
   This is can also be used when the sphericity criterion is not met, and it is constructed from the Greenhouse-Geisser estimate $\hat{\epsilon}$ as follows
$$\begin{aligned} a &= n(m-1)\hat{\epsilon} - 2 \\ b &= (m-1)(n - G - (m-1)\hat{\epsilon}) \\ \epsilon &= \min(1, a/b), \end{aligned}$$
   where G is the number of groups. It is generally recommended to use this estimate if the ANOVA probabilities given by the various adjustments differ appreciably.

4. The lower bound epsilon.
   This is defined as
$$\epsilon = 1/(m-1)$$
   which is the smallest value and results in using the F statistic with 1 and $n-1$ degrees of freedom.

If the sphericity criterion is not met, then it is possible to use multivariate techniques such as MANOVA as long as $n > m$, as these do not require sphericity, but these will always be less powerful than the univariate ANOVA just discussed.

One possibility is to use the Hotelling T^2 test to see if the column means differ significantly, and the results displayed in the previous tables were obtained in this way. Again a matrix C of orthonormal contrasts is used together with the vector of column means
$$\bar{y} = (\bar{y}_1, \bar{y}_2, \ldots, \bar{y}_m)^T$$
to construct the statistic
$$T^2 = n(C\bar{y})^T (CSC^T)^{-1}(C\bar{y})$$
since
$$\frac{(n-m+1)T^2}{(n-1)(m-1)} \sim F(m-1, n-m+1)$$
if all column means are equal.

4.4.8 3-way ANOVA (Latin square)

Experimental design seeks to optimize ANOVA in order to eliminate systematic effects, and one such design is the Latin square. This requires a m by m matrix of observations, but also a corresponding m by m Latin square matrix of treatments, so that

- each row of the Latin square contains all of the m treatments;
- the order of treatments is different within every row; and
- the observations are arranged according to the pattern dictated by the Latin square.

Note that SimFIT provides the option to generate random Latin squares to avoid systematic effects with repeated experiments.

From the SimFIT main menu choose [Statistics], [ANOVA], then [Latin squares] and open the test file `anova3.tf1` which contains a 10 by 5 data matrix as follows.

5	4	1	3	2
2	5	4	1	3
3	2	5	4	1
1	3	2	5	4
4	1	3	2	5
6.67	7.15	8.29	8.95	9.62
5.40	4.77	5.40	7.54	6.93
7.32	8.53	8.50	9.99	9.68
4.92	5.00	7.29	7.85	7.08
4.88	6.16	7.83	5.38	8.51

Here the upper 5 by 5 matrix colored red is the Latin square, which corresponds to the lower 5 by 5 matrix of observations. In other words, observation$(5+i, j)$ corresponds to treatment(i, j) for $i = 1, 2, \ldots, 5$ and $j = 1, 2, \ldots 5$. For example, observation$(1, 1)$ which is 6.67 resulted from treatment 5, observation$(1, 2)$ which is 7.15 resulted from treatment 4, observation$(5, 4)$ which is 5.38 resulted from treatment 2, etc.

Analysis leads to the following table.

Three Way Analysis of Variance: Grand mean 7.186
Data file: `anova3.tf1`
Data title: Latin square ANOVA Data ... see NAG routine G04ADF

Source	NDOF	SSQ	MSQ	F	p
Rows	4	29.42	7.356	9.027	0.0013
Columns	4	22.99	5.749	7.055	0.0037
Treatments	4	0.5423	0.1356	0.1664	0.9514
Error	12	9.779	0.8149		
Total	24	62.74	2.614		

Mean Values					
Row means	8.136	6.008	8.804	6.428	6.552
Column means	5.838	6.322	7.462	7.942	8.364
Treatment means	7.318	7.244	7.206	6.900	7.260

Theory for Latin square ANOVA

The linear model for a m by m Latin square ANOVA is

$$y_{ijk} = \mu + \alpha_i + \beta_j + \gamma_k + e_{ijk}$$
$$\sum_{i=1}^{m} \alpha_i = 0$$
$$\sum_{j=1}^{m} \beta_j = 0$$
$$\sum_{k=1}^{m} \gamma_k = 0$$

where α_i, β_j and γ_k represent the row, column and treatment effect, and e_{ijk} is assumed to be normally distributed with zero mean and variance σ^2. The sum of squares partition is now

$$\text{Total } SSQ = \text{ Row } SSQ + \text{ Column } SSQ + \text{ Treatment } SSQ + \text{ Residual } SSQ$$

where the m^2 observations are arranged in the form of a m by m matrix so that every treatment occurs once in each row and column. This design, which is used for economical reasons to account for row, column, and treatment effects, leads to the three variance ratios

$$F_R = \frac{\text{Row } MS}{\text{Residual } MS}$$
$$F_C = \frac{\text{Column } MS}{\text{Residual } MS}$$
$$F_T = \frac{\text{Treatment } MS}{\text{Residual } MS}$$

to use in F tests with $m - 1$, and $(m - 1)(m - 2)$ degrees of freedom. Note that SimFIT data files for Latin square designs with m treatment levels have $2m$ rows and m columns, where the first m by m block identifies the treatments, and the next m by m block of data are the observations. When designing such experiments, the particular Latin square used should be chosen randomly if possible as described later. For instance, study the test file `anova3.tf1`, which should be consulted for details, noting that integers (1, 2, 3, 4, 5) are used instead of the usual letters (A, B, C, D, E) in the data file header to indicate the position of the treatments.

Note that in the Latin square results table there are now three p values for significance testing between rows, columns, and treatments.

Generating random Latin squares

This section describes how to generate a data file so that SimFIT can perform Latin square analysis. This involves three procedures.

1. Generate a random Latin square
2. Organize the observations according to the design
3. Fuse the Latin square and Observations to make a data matrix

From the SimFIT main menu choose [A/Z], open **rannum**, choose random permutations, then generate a Latin square like this next one.

Random k by k Latin square: $k = 5$

5	2	4	1	3
3	5	2	4	1
4	1	3	5	2
2	4	1	3	5
1	3	5	2	4

Equivalent alphabetical representation

E	B	D	A	C
C	E	B	D	A
D	A	C	E	B
B	D	A	C	E
A	C	E	B	D

Now proceed as follows.

- Save the k by k data from **rannum** to a file in integer format, not alphabetical format.
- Create a k by k data-only file with the observations arranged according to the saved Latin square
- Fuse the two files to make a $2k$ by k data file where the upper matrix contains the Latin square integers.

In order to fuse the two files you can use **editmt** to combine the two files but it is very easy to proceed as follows.

- Open the Latin square matrix in a text editor such as Notepad
- Select the k by k Latin square integer matrix and copy to the clipboard
- Open the k by k data matrix in a text editor
- Paste the k by k Latin square integer matrix before the data section
- Edit the edited data matrix at line 1 to indicate the changes
- Change the matrix dimension on line 2 from $k \quad k$ into $2k \quad k$
- Save the edited data file.

Alternatively, as data files copied from the clipboard into SIMFIT do not require titles and array dimensions, you can easily paste the following data matrix directly into SIMFIT from the clipboard.

```
5       4       1       3       2
2       5       4       1       3
3       2       5       4       1
1       3       2       5       4
4       1       3       2       5
6.67    7.15    8.29    8.95    9.62
5.40    4.77    5.40    7.54    6.93
7.32    8.53    8.50    9.99    9.68
4.92    5.00    7.29    7.85    7.08
4.88    6.16    7.83    5.38    8.51
```

4.4.9 Groups and subgroups ANOVA

In order to appreciate what is involved in groups and subgroups ANOVA, consider the following data set contained in the SIMFIT test file `anova4.tf1` and made available after opening the main SIMFIT menu, followed by choosing [Statistics], [ANOVA], then the groups and subgroups option.

Groups	Subgroups	Observations
1	1	2.1
1	1	2.4
1	1	2.0
1	1	2.0
1	1	2.0
1	2	2.4
1	2	2.1
1	2	2.2
1	3	2.4
1	3	2.2
1	3	2.6
1	4	2.4
1	4	2.4
1	4	2.5
1	5	1.9
1	5	1.7
2	1	2.1
2	1	1.5
2	1	2.0
2	2	1.9
2	2	1.7
2	2	1.9
2	2	1.9
2	2	1.9
2	3	2.0
2	3	2.1
2	3	2.3

The first column is the group number (in nondecreasing order), the second column is the subgroup within the group (in nondecreasing order), and the third column holds the corresponding observations. Note that there are no limits to the number of groups, nor to the sizes of the subgroups, but any data file supplied for groups and subgroups ANOVA must be arranged exactly as above.

These data were obtained as follows.

- The two groups represent data obtained in two consecutive years.
- The subgroups of size five and three are consignments of materials in those two years.
- The observations are replicates for percentages of stretch of the material within the groups and subgroups.

It is wished to test if there are significant differences between the properties of material delivered in the two years, and also between the samples drawn, and analysis then leads to the following results table.

Results for Groups/Subgroups 2-Way ANOVA
Transformation = x (i.e. untransformed data)

Source	SSQ	NDOF	F	p
Between Groups	0.4748	1	16.15	0.0007
Subgroups	0.8162	6	4.626	0.0047
Residual	0.5587	19		
Total	1.850	26		

Group	Subgroup	Mean
1	1	2.100
1	2	2.233
1	3	2.400
1	4	2.433
1	5	1.800
2	1	1.867
2	2	1.860
2	3	2.133

Description	Mean	Sample size
Group 1	2.206	16
Group 2	1.936	11
Grand	2.096	27

These results could be interpreted to suggest a between groups effect ($p = 0.0007$) and also a subgroup effect ($p = 0.0047$) and the requisite theory will be presented next.

Theory

The linear models for ANOVA are easy to manipulate mathematically and trivial to implement in computer programs, and this has lead to a vast number of possible designs for ANOVA procedures. This situation is likely to bewilder users, and may easily mislead the unwary, as it stretches credulity to the limit to believe that experiments, which almost invariably reflect nonlinear non-normal phenomena, can be analyzed in a meaningful way by such elementary models. Nevertheless, ANOVA remains valuable for preliminary data exploration, or in situations like clinical or agricultural trials, where only gross effects are of interest and precise modelling is out of the question. So groups and subgroups ANOVA is a versatile and flexible technique provided by SIMFIT for two-way hierarchical classification with subgroups of possibly unequal size, assuming a fixed effects model.

Suppose, for instance, that there are $k \geq 2$ treatment groups, with group i subdivided into l_i treatment subgroups, where subgroup j contains n_{ij} observations. That is, observation y_{mij} is observation m in subgroup j of group i where

$$1 \leq i \leq k, 1 \leq j \leq l_i, 1 \leq m \leq n_{ij}.$$

The between groups, between subgroups within groups, and residual sums of squares are

$$\text{Group } SSQ = \sum_{i=1}^{k} n_{i.}(\bar{y}_{.i.} - \bar{y}_{...})^2$$

$$\text{Subgroup } SSQ = \sum_{i=1}^{k}\sum_{j=1}^{l_i} n_{ij}(\bar{y}_{.ij} - \bar{y}_{.i.})^2$$

$$\text{Residual } SSQ = \sum_{i=1}^{i}\sum_{j=1}^{l_i}\sum_{m=1}^{n_{ij}} (y_{mij} - \bar{y}_{.ij})^2$$

which, using $l = \sum_{i=1}^{k} l_i$ and $n = \sum_{i=1}^{k} n_{i.}$, and normalizing give the variance ratios

$$F_G = \frac{\text{Group } SSQ/(k-1)}{\text{Residual } SSQ/(n-l)}$$
$$F_S = \frac{\text{Subgroup } SSQ/(l-k)}{\text{Residual } SSQ/(n-l)}$$

to test for between groups and between subgroups effects.

Of course, there are now two p values for significance testing and, also note that, because this technique allows for many designs that cannot be represented by rectangular matrices. However, the data files must have three columns and n rows: column one containing the group numbers, column two containing the subgroup numbers, and column three containing the n observations as a sample vector in the order of groups, and subgroups within groups.

Note that, by defining groups and subgroups correctly in this way, a large number of ANOVA techniques can be performed using this groups and subgroups ANOVA procedure.

4.4.10 Factorial ANOVA

Factorial ANOVA is designed to analyze the effects of categorical or quantitative variables, called factors, on observed quantities in order to detect the influence of these factors both separately and jointly.

Example 1

Open the SimFIT main menu, select[Statistics], [ANOVA], then [Factorial ANOVA], and analyze the data set below for no blocks but two factors, *A* and *B*, contained in the default SimFIT test file `anova5.tf1`.

Block	Factor *A*	Factor *B*	Observation
1	1	1	16.5
1	1	1	18.4
1	1	1	12.7
1	1	1	14.0
1	1	1	12.8
1	1	2	14.5
1	1	2	11.0
1	1	2	10.8
1	1	2	14.3
1	1	2	10.0
1	2	1	39.1
1	2	1	26.2
1	2	1	21.3
1	2	1	35.8
1	2	1	40.2
1	2	2	32.0
1	2	2	23.8
1	2	2	28.8
1	2	2	25.0
1	2	2	29.3

In order to perform factorial ANOVA using SimFIT the data matrix must have a very precise structure which, in the case of the two factors above, is in standard order as follows.

1. Column 1 contains block numbers (in this case all 1 as there are no blocks)
2. Column 2 contains levels of factor *A* (in this case Level 1 = no hormone, Level 2 = hormone)
3. Column 3 contains levels of factor *B* (in this case Level 1 = female, level 2 = male)
4. Column 4 contains observed values (in this case of blood calcium in mg/100ml with 5 replicates)
5. Block numbers must be in nondecreasing order
6. Levels of *A* must be in nondecreasing order within each block
7. Levels of *B* must be in nondecreasing order within each level of *A*
8. There must be the same number of replicates in each group
9. Block numbers and factor levels must be consecutive integers ≥ 1

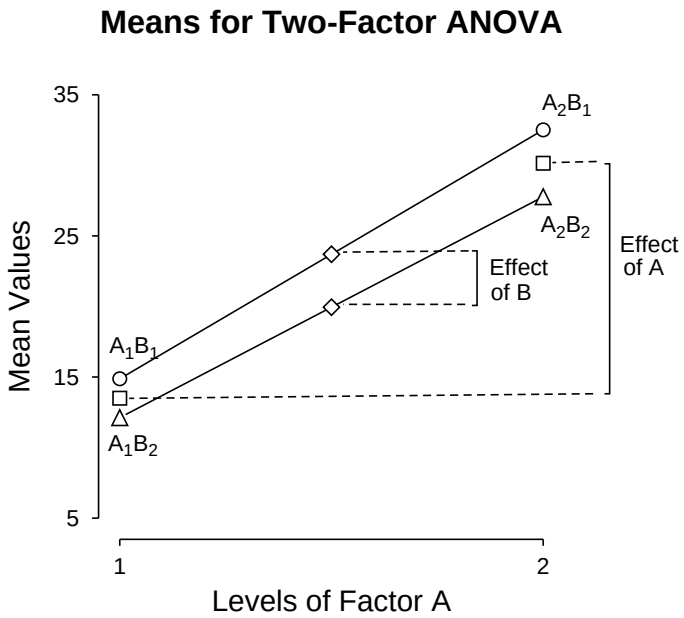
After analyzing the data in test file `anova5.tf1` SimFIT displays the following results table.

Table 1: Results for Factorial ANOVA with test file `anova5.tf1`
Transformation: x (untransformed data)

Source	SSQ	$NDOF$	MS	F	p
Blocks	0.000	0	0.000	0.000	0.0000
Effect 1 (A)	1386	1	1386	60.53	0.0000
Effect 2 (B)	70.31	1	70.31	3.071	0.0989
Effect 3 $(A * B)$	4.900	1	4.900	0.2140	0.6499
Residual	366.4	16	22.90		
Total	1828	19			

Treatment Means and Standard Errors				
Overall mean	21.82			
Treatment means				
Effect 1	13.50	30.15		
Standard Error of difference in means:	2.140			
Effect 2	23.70	19.95		
Standard Error of difference in means:	2.14			
Effect 3	14.88	12.12	32.52	27.78
Standard Error of difference in means:	3.026			

Note that in factorial ANOVA tables the treatment effects are always output in standard order as follows.
Effect 1: A_1, A_2
Effect 2: B_1, B_2
Effect 3: $A_1B_1, A_1B_2, A_2B_1, A_2B_2$
Also, after analyzing the data, a plot like the next one can be useful to illustrate the effects of the factors.



Example 2

To further emphasize the format required for factorial ANOVA, consider the following data set contained in SimFITtest file `anova5.tf2`.

Block	Factor A	Factor B	Observation
1	1	2	361
1	1	3	253
1	2	1	325
1	2	2	317
1	2	3	339
1	3	1	326
1	3	2	402
1	3	3	336
1	4	1	379
1	4	2	345
1	4	3	361
1	5	1	352
1	5	2	334
1	5	3	318
1	6	1	339
1	6	2	393
1	6	3	358
2	1	1	350
2	1	2	340
2	1	3	203
2	2	1	397
2	2	2	356
2	2	3	298
2	3	1	382
2	3	2	376
2	3	3	355
2	4	1	418
2	4	2	387
2	4	3	379
2	5	1	432
2	5	2	339
2	5	3	293
2	6	1	322
2	6	2	417
2	6	3	342
3	1	1	82
3	1	2	297
3	1	3	133
3	2	1	306
3	2	2	352
3	2	3	361
3	3	1	220
3	3	2	333
3	3	3	270
3	4	1	388
3	4	2	379
3	4	3	274
3	5	1	336
3	5	2	307
3	5	3	266
3	6	1	389
3	6	2	333
3	6	3	353

This data set is for the yields of turnips in an agricultural experiment with 3 blocks, 6 levels of A (phosphate), and 3 levels of B (lime).

Table 2: Results for Factorial ANOVA with test5 file `anova5.tf2`
Transformation: x (untransformed data)

Source	SSQ	$NDOF$	MS	F	p
Blocks	30120	2	15060	7.685	0.0018
Effect 1 (A)	73010	5	14600	7.451	0.0001
Effect 2 (B)	21600	2	10800	5.510	0.0085
Effect 3 ($A * B$)	31190	10	3119	1.592	0.1513
Residual	66630	34	1960		
Total	222500	53			

Treatment Means and Standard Errors					
Overall mean	331.1				
Block means	339.6	354.8	298.8		
Treatment means					
Effect 1	254.8	339.0	333.3	367.8	330.8
	360.7				
Standard Error of difference in means: 20.87					
Effect 2	334.3	353.8	305.1		
Standard Error of difference in means: 14.76					
Effect 3	235.3	332.7	196.3	342.7	341.7
	332.7	309.3	370.3	320.3	395.0
	370.3	338.0	373.3	326.7	292.3
	350.0	381.0	351.0		
Standard Error of difference in means: 36.14					

Theory

The appropriate linear model for factorial ANOVA is

$$y_{ijk} = \mu + \alpha_i + \beta_j + \gamma_{ij} + e_{ijk}$$

where there are a levels of factor A, b levels of factor B and n replicates per group, that is, n observations at each fixed pair of i and j values.

As usual, μ is the mean, α_i is the effect of A at level i, β_j is the effect of B at level j, γ_{ij} is the effect of the interaction between A and B at levels i and j, and e_{ijk} is the random error component at replicate k.

Also there are the necessary constraints on the parameters estimated, that is

$$\sum_{i=1}^{a} \hat{\alpha}_i = 0$$

$$\sum_{j=1}^{b} \hat{\beta}_j = 0$$

$$\sum_{i=1}^{a} \hat{\gamma}_{ij} = 0, \text{ for each } j, \text{ and}$$

$$\sum_{j=1}^{b} \hat{\gamma}_{ij} = 0 \text{ for each } i.$$

The null hypotheses would be

$$H_0 : \alpha_i = 0, \text{ for } i = 1, 2, \ldots, a$$

to test for the effects of factor A,

$$H_0 : \beta_j = 0, \text{ for } j = 1, 2, \ldots, b$$

to test for the effects of factor B, and

$$H_0 : \gamma_{ij} = 0, \text{ for all } i, j$$

to test for possible AB interactions.

The analysis of variance table is based upon calculating F statistics as ratios of sums of squares that arise from the partitioning of the total corrected sum of squares as follows

$$\begin{aligned}\sum_{i=1}^{a}\sum_{j=1}^{b}\sum_{k=1}^{n}(y_{ijk}-\bar{y}_{...})^2 &= \sum_{i=1}^{a}\sum_{j=1}^{b}\sum_{k=1}^{n}[(\bar{y}_{i..}-\bar{y}_{...})+(\bar{y}_{.j.}-\bar{y}_{...})\\ &\quad+(\bar{y}_{ij.}-\bar{y}_{i..}-\bar{y}_{.j.}+\bar{y}_{...})+(y_{ijk}-\bar{y}_{ij.})]^2\\ &= bn\sum_{i=1}^{a}(\bar{y}_{i..}-\bar{y}_{...})^2+an\sum_{j=1}^{b}(\bar{y}_{.j.}-\bar{y}_{...})^2\\ &\quad+n\sum_{i=1}^{a}\sum_{j=1}^{b}(\bar{y}_{ij.}-\bar{y}_{i..}-\bar{y}_{.j.}+\bar{y}_{...})^2+\sum_{i=1}^{a}\sum_{j=1}^{b}\sum_{k=1}^{n}(\bar{y}_{ijk}-\bar{y}_{ij.})^2\end{aligned}$$

It is clear from the F statistics and significance levels p in Table 1 table for `anova5.tf1` that, with these data, A has a large effect, B has a small effect, and there is no significant interaction. From table 2 for `anova5.tf2` the effects of blocking as well as A and B are significant, but the interaction between A and B is small.

Note that the factorial ANOVA table always outputs results in standard order, e.g. $A_1B_1, A_1B_2, A_2B_1, A_2B_2$ and so on, while the actual coefficients $\alpha_i, \beta_j, \gamma_{ij}$ in the model can be estimated by subtracting the grand mean from the corresponding treatment means. In the previous marginals plot, the line connecting the circles is for observations with B at level 1 and the line connecting the triangles is for observations with B at level 2. The squares are the overall means of observations with factor A at level 1 (13.5) and level 2 (30.15), while the diamonds are the overall means of observations with factor B (i.e. 23.7 and 19.95) from the results table. Parallel lines indicate the lack of interaction between factors A and B while the larger shift for variation in A as opposed to the much smaller effect of changes in levels of B merely reinforces the conclusions reached previously from the p values in the results table.

If the data set contains blocking, as with test files `anova5.tf2` and `anova5.tf4`, then there will be extra information in the ANOVA table corresponding to the blocks, e.g., to replace the values shown as zero in the results table for `anova5.tf1`, as there is no blocking with the data in `anova5.tf1`.

The SimFIT factorial ANOVA test files illustrate the following examples of data formatting, and these can be consulted and subsequently analyzed if further clarification is required.

`anova5.tf1`: 0 blocks and 2 factors
`anova5.tf2`: 3 blocks and 2 factors
`anova5.tf3`: 0 blocks and 3 factors
`anova5.tf4`: 3 blocks and 3 factors

4.5 Analysis of frequencies and proportions

Tutorials and worked examples for simulation, curve fitting, statistical analysis, and plotting.
https://simfit.uk
https://simfit.org.uk
https://simfit.silverfrost.com

4.5.1 Introduction

Many experiments record frequencies with which events occur, and then these frequencies are used to calculate proportions to be used as estimates for population probabilities.

As a simple example, consider tossing a coin N times resulting in h heads and $N - h$ tails. Then the frequency of heads would be the integer h, while the proportion of heads would be the floating point number h/N, which would converge to the true probability of a head occurring for large values of N. In this case of dichotomous data we could define a variable x_i to have a value of 1 for a success (e.g. heads) and 0 for failure (e.g. tails) at the i'th trial, leading to a random variable X as the sum of the x_i values as follows

$$X = x_1 + x_2 + \cdots + x_N .$$

Then the appropriate statistical model would be the binomial distribution with parameters N and p, so that the probability of observing h successes in N independent trials would be

$$P(X = h) = \binom{N}{h} p^h (1-p)^{N-h} \text{ where } \lim_{N\to\infty} \frac{h}{N} = p.$$

More generally, suppose that a total of N observations can be classified into k categories with frequencies consisting of y_i observations in category i, so that $0 \le y_i \le N$ and $\sum_{i=1}^{k} y_i = N$, then there are k proportions, that is ratios r_i of frequencies to sample size, defined as

$$r_i = y_i/N,$$

of which only $k - 1$ are independent since $r_1 + r_2 + \cdots + r_k = 1$. If these proportions are then interpreted as estimates of the multinomial probabilities and it is wished to make inferences about these probabilities, then we are in a situation that can be described as the analysis of proportions, or the analysis of categorical data.

Since the observations are integer frequencies and not measurements, they are not normally distributed, so techniques like ANOVA should not be used, instead specialized methods to analyze frequencies must be employed. In particular, exact estimates for variances and confidence limits are not always available, and approximate confidence range estimates often exceed the theoretically possible limits since, for an estimate, say $\hat{p}$, with lower 95% confidence limit C_L, and upper 95% confidence limit C_U we must have

$$0 \le C_L \le \hat{p} \le C_U \le 1.$$

Furthermore, although exact confidence limits will not be symmetrical, approximate confidence limits will be. For example, with 2 successes in 10 trials the estimate for the binomial parameter would be

$$\hat{p} = 0.2,$$

but the exact 95% confidence range calculated by SimFiT was found to be

$$0.0252 \le 0.2 \le 0.5561$$

so that $C_L = 0.2 - 0.1748$ while $C_U = 0.2 + 0.3561$. This illustrates a typical result that, for probability estimates less than 0.5 confidence ranges are skewed to the right, while for estimates greater than 0.5 confidence ranges are skewed to the left. So it is not accurate to report estimates as, e.g. $\hat{p} = (h/N) \pm \alpha$ for some α.

4.5.2 Binomial proportions (dichotomous data)

If there only two categories, such as success or failure, male or female, dead or alive, etc., the data are referred to as dichotomous, and there is only one parameter to consider. So the analysis of two–category data is based on the binomial distribution which is required when y events (e.g., successes) have been recorded in N independent trials with constant probability of success (i.e., Bernoulli trials) and it is wished to explore possible variations in the binomial parameter estimate

$$\hat{p} = y/N,$$

and its unsymmetrical confidence limits, possibly as ordered by an indexing parameter x.

Analyzing binomial proportions

From the main SimFIT menu choose [Statistics], [Analysis of proportions], then [Binomial proportions], and examine the default test file `binomial.tf3` which has the following format.

y	N	x
23	84	1
12	78	2
31	111	3
65	92	4
71	93	5

The columns in this data format must be as follows.

- Column 1: The number of successes $0 \leq y \leq N$
- Column 2: The number of Bernoulli trials $N > 0$
- Column 3: An optional indexing parameter x

Note that the indexing parameter x is not used for any calculations, it is only required in order to identify, label, and space the data for subsequent plotting. If this third column is missing, as in `binomial.tf2`, SimFIT simply appends a third column of successive integers $1, 2, \ldots, N$. Typically x could be sample identifiers, concentrations of chemical, time from start of treatment, etc.

The SimFIT analysis of proportions procedure accepts a matrix of such y, N data then calculates the binomial parameters and derived parameters such as the Odds

$$\text{Odds } = \hat{p}/(1-\hat{p}), \text{ where } 0 < \hat{p} < 1,$$

and log(Odds), along with standard errors and confidence limits. It also performs a chi-square contingency table test and a likelihood ratio test for common binomial parameters as in the next table.

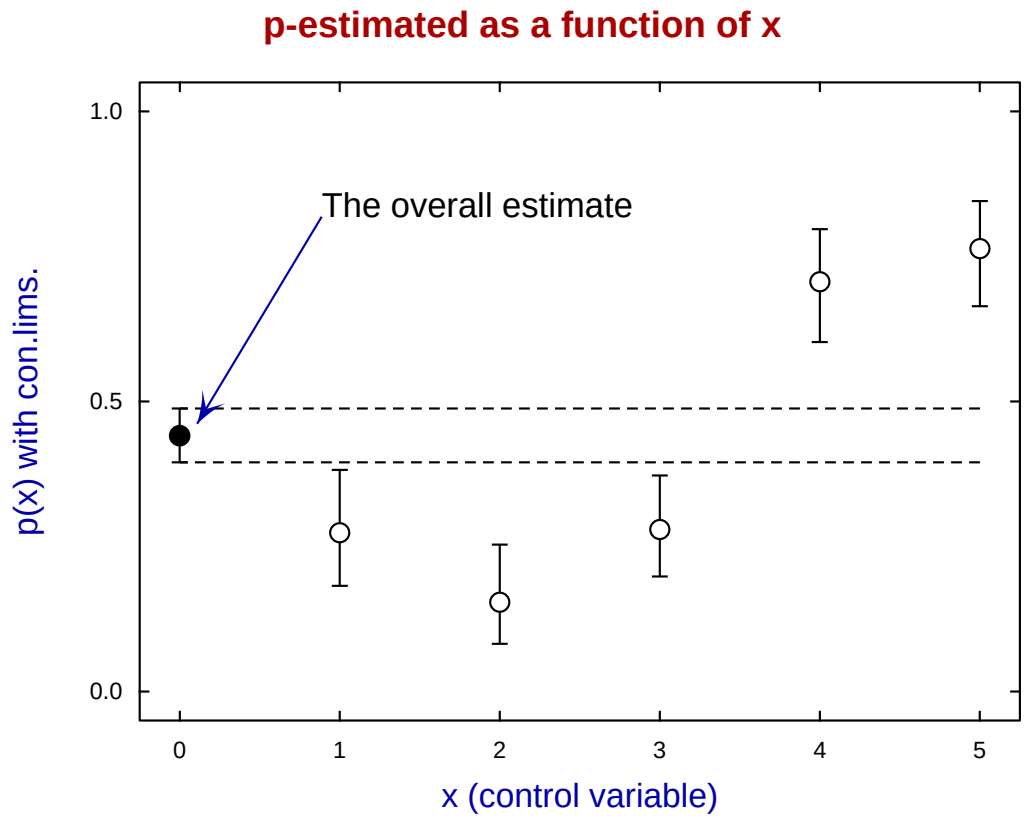
To test H_0 : equal binomial p-values for data in test file `binomial.tf3`		
Sample-size i.e. number of pairs	5	
Overall sum of y	202	
Overall sum of N	458	
Overall estimate of p	0.4410	
Lower 95% confidence limit	0.3950	
Upper 95% confidence limit	0.4879	
$-2\log\lambda$ $(-2LL)$	118.3	$NDOF = 4$
p = $P(\chi^2 \geq -2LL)$	0.0000	Reject H_0 at 1% significance level
Chi-square test statistic (C)	112.9	$NDOF = 4$
p = $P(\chi^2 \geq C)$	0.0000	Reject H_0 at 1% significance level

After choosing to analyze the parameter estimates, the next table with the data, p estimates, and exact 95% confidence limits is displayed.

y	N	lower-95%	$\hat{p}$	upper-95%
23	84	0.18214	0.27381	0.38201
12	78	0.08210	0.15385	0.25332
31	111	0.19829	0.27928	0.37241
65	92	0.60242	0.70652	0.79688
71	93	0.66404	0.76344	0.84542

Plotting binomial proportions

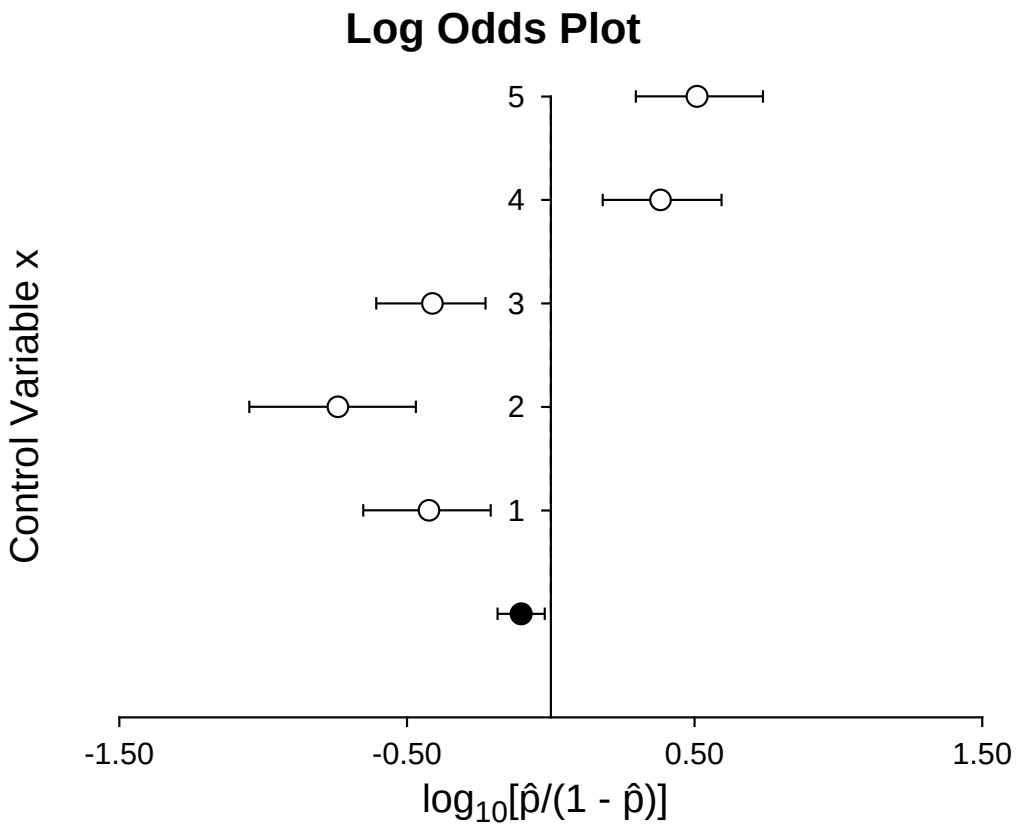
These results can then be plotted as individual sample estimates with 95% confidence limits, as in this graph where the overall estimate with overall confidence limits is also displayed.



A useful rule of thumb is to analyze such plots for the position of confidence limits for the individual estimates with respect to the other individual confidence limits and the overall confidence limits shown as dotted lines. It is clear that samples 1, 2, and 3 lie below the overall confidence limits, while samples 4 and 5 lie above the overall confidence limits, confirming the previous conclusions from the χ^2 test.

Plotting log odds

It should be emphasized that everything that can be done with binomial proportions can be done by calculating and plotting parameter estimates $\hat{p}$ and confidence limits as just discussed. However, many experimentalists prefer to work with log odds in order to emphasize differences in order of magnitudes. For instance, the data can be plotted as log odds as in this next figure. However, to perform advanced graphics editing SIMFIT always transfers raw data not transformed data into the advanced editing, so it is necessary to transfer x, $y/(N-y)$ into the advanced editing option first of all, followed by choosing a reverse y-semilog interactive transformation using logs to base ten. In addition, for finishing touches, the legends were edited and the y-axis moved to the central position indicated.



Binomial parameter confidence limits

It is obvious that a binomial parameter estimate $\hat{p} = y/N$ for the true population parameter p must satisfy

$$0 \leq \hat{p} \leq 1$$

and so the confidence limits should also be constrained to this range. Hence any accurate confidence limits cannot be symmetrical but must be skewed and so, when a binomial parameter is estimated, it is not possible to report the result in the usual way as $\hat{p} \pm \hat{s}$, or as $\hat{p}(\hat{p} - \hat{s}, \hat{p} + \hat{s})$, where $\hat{s}$ is estimated from the sample and percentiles of a standard normal distribution. Nevertheless, many users of computer packages do not understand this and prefer an approximate expression using the normal distribution because, as long as the sample is large and $p \approx 0.5$, a binomial distribution can be approximated by a normal distribution. For that reason a large sample 95% approximate central confidence range for the true population parameter p is often constructed using

$$\tilde{p} - \tilde{s} \leq p \leq \tilde{p} + \tilde{s}, \text{ where } \tilde{s} = Z_{\alpha/2}\sqrt{\tilde{p}(1-\tilde{p})/\tilde{N}}$$

with $\tilde{N} = N + 4$, and $\tilde{p} = (y + 2)/\tilde{N}$.

It is clear that for large samples with $y \approx N/2$ the normal approximation will be adequate but, in order to check the closeness of the approximate limits to the exact ones in any given case, SIMFIT provides tables to check the values. For instance, analysis of the test file `binomial.tf4` yields the following comparison.

$\hat{p} = (y/N)$ with exact unsymmetrical small sample limits

y	N	Lower-95%	$\hat{p}$	Upper-95%
23	84	0.182144	0.273810	0.382008
12	78	0.082102	0.153846	0.253321
31	111	0.198289	0.279279	0.372414
91	92	0.940922	0.989130	0.999725
1	93	0.000272	0.010753	0.058458

$\tilde{p} = (y+2)/(N+4)$ with approximate central limits $[\tilde{p} \pm \tilde{s}]$

y	N	Lower-95%	$\tilde{p}$	Upper-95%	$\tilde{s}$
23	84	0.189866	0.284091	0.378316	0.094225
12	78	0.089290	0.170732	0.252173	0.081442
31	111	0.204283	0.286957	0.369630	0.082673
91	92	0.933945	0.968750	1.003555	0.034805 ***
1	93	-0.003524	0.030928	0.065380	0.034452 ***

*** Indicates parameter limits outside range (0,1)

The column $\hat{s}$ indicates the amount $\hat{s}$ added to and subtracted from $\hat{p}$ to derive the limits so that the results can be reported as $\hat{p} \pm \hat{s}$. It will be seen that modifying the data in test file `binomial.tf3` to make test file `binomial.tf4` by editing in a couple of extreme values causes the approximate method to overflow or underflow as indicated by ***. Actually the numerical calculation to estimate the exact confidence takes much longer than estimation of the normal approximation, so SIMFIT allows users to choose the method to use when analyzing large samples.

Differences between probability estimates

For cases where the number of samples is relatively small, it is also sometimes helpful to examine tables that highlight significant differences between estimates as follows, using the test file `binomial.tf4`.

$d(i,j) = \hat{p}_i - \hat{p}_j, \quad NNT = 1/|d(i,j)|$

i	j	Lower-95%	$d(i,j)$	Upper-95%	Result	p_sig.	$Var(d(i,j))$	NNT	(95%cl)
1	2	-0.00455	0.11996	0.24448	Not significant	0.0590	0.00404	8	(*.*) NS
1	3	-0.13219	-0.00547	0.12125	Not significant	0.9326	0.00418	183	(*.*) NS
1	4	-0.81300	-0.71532	-0.61764	(1) < (4)	0.0000	0.00248	1	(*.*) NC
1	5	0.16542	0.26306	0.36069	(1) > (5)	0.0000	0.00248	4	(3,6)
2	3	-0.24109	-0.12543	-0.00977	(2) < (3)	0.0355	0.00348	8	(4,102) NNH
2	4	-0.91811	-0.83528	-0.75246	(2) < (4)	0.0000	0.00179	1	(*.*) NC
2	5	0.06033	0.14309	0.22586	(2) > (5)	0.0007	0.00178	7	(4,17) NC
3	4	-0.79596	-0.70985	-0.62374	(3) < (4)	0.0000	0.00193	1	(*.*) NC
3	5	0.18247	0.26853	0.35458	(3) > (5)	0.0000	0.00193	4	(3,5)
4	5	0.94857	0.97838	1.00818	(4) > (5)	0.0000	0.00023	1	(*.*) NC

p_sig. = significance, NNH = No. needed to harm, NS = Not significant, NC = Not calculated

Note that when the lower limit is negative and the upper limit is positive the confidence range includes zero so that the difference between estimates is not significantly different from zero. When the parameters are listed as different, the result can be interpreted as stricter (since $\alpha/2$ is used) than a one–sided lower tail or upper tail test (where α would normally be used). A purist would argue that, as three tests are being done on the same data, the Bonferroni principle would require that significance levels should be divided by three anyway.

It should be noted that the number needed to treat (NNT) is simply the reciprocal of the absolute difference $d(i,j) = p_i - p_j$, except that, to avoid overflow, this is constrained to the range $1 \le NNT \le 10^6$. Where confidence limits for NNT cannot be estimated, this is indicated by NC, and when the probability difference is negative the number needed to harm is indicated by NNH instead of the confidence range, as will be seen in the above table.

Often it is required to calculate pairwise differences between adjacent lines of the data file, this will happen automatically if the sample size exceeds a certain limiting size. For instance a sample size of size N requires $\binom{N}{2}$ lines of table to output the results from all pairwise comparisons whereas restricting analysis to adjacent pairs only requires $N/2$. Here, for instance, is the result from analyzing the test file `binomial.tf5` where there are twenty lines of data requiring a table with only ten rows.

$d(i,j) = \hat{p}_i - \hat{p}_j, \quad NNT = 1/|d(i,j)|$

i	j	Lower-95%	$d(i,j)$	Upper-95%	Result	p_sig.	$Var(d(i,j))$	NNT	(95%cl)
1	2	0.06707	0.20000	0.33293	(1) > (2)	0.0032	0.00460	5	(3,15)
3	4	0.01315	0.15000	0.28685	(3) > (4)	0.0317	0.00488	7	(3,76)
5	6	0.06707	0.20000	0.33293	(5) > (6)	0.0032	0.00460	5	(3,15)
7	8	0.07604	0.20000	0.32396	(7) > (8)	0.0016	0.00400	5	(3,13)
9	10	0.08322	0.20000	0.31678	(9) > (10)	0.0008	0.00355	5	(3,12)
11	12	0.11684	0.21000	0.30316	(11) > (12)	0.0000	0.00226	5	(3,9)
13	14	-0.23118	-0.14000	-0.04882	(13) < (14)	0.0026	0.00216	7	(4,20) NNH
15	16	-0.46386	-0.35000	-0.23614	(15) < (16)	0.0000	0.00338	3	(2,4) NNH
17	18	-0.15509	-0.05000	0.05509	Not sig.	0.3511	0.00288	20	(*.*) NS
19	20	-0.12002	0.00000	0.12002	Not sig.	1.0000	0.00375	>999999	(*.*) NS

p_sig. = significance, NNH = No. needed to harm, NS = Not significant, NC = Not calculated

Confidence limits for analysis of two proportions

Given two proportions p_i and p_j estimated as

$$\hat{p}_i = y_i/N_i$$
$$\hat{p}_j = y_j/N_j$$

it is often wished to estimate confidence limits for the relative risk RR_{ij}, the difference between proportions DP_{ij}, and the odds ratio OR_{ij}, defined as

$$RR_{ij} = \hat{p}_i/\hat{p}_j$$
$$DP_{ij} = \hat{p}_i - \hat{p}_j$$
$$OR_{ij} = \frac{\hat{p}_i/(1-\hat{p}_i)}{\hat{p}_j/(1-\hat{p}_j)}.$$

First of all note that, for small proportions, the odds ratios and relative risks are similar in magnitude. Further, unlike the case of single proportions, exact confidence limits for these derived parameters can not be calculated. However, approximate central $100(1-\alpha)\%$ confidence limits can be obtained using the large sample normal approximations

$$\log(RR_{ij}) \pm Z_{\alpha/2}\sqrt{\frac{1-\hat{p}_i}{N_i\hat{p}_i} + \frac{1-\hat{p}_j}{N_j\hat{p}_j}}$$
$$DP_{ij} \pm Z_{\alpha/2}\sqrt{\frac{\hat{p}_i(1-\hat{p}_i)}{N_i} + \frac{\hat{p}_j(1-\hat{p}_j)}{N_j}}$$
$$\log(OR_{ij}) \pm Z_{\alpha/2}\sqrt{\frac{1}{y_i} + \frac{1}{N_i-y_i} + \frac{1}{y_j} + \frac{1}{N_j-y_j}}$$

provided $\hat{p}_i$ and $\hat{p}_j$ are not too close to 0 or 1. Here $Z_{\alpha/2}$ is the upper $100\alpha/2$ percentage point, i.e., the lower $100(1-\alpha/2)$ percentage point for the standard normal distribution, and confidence limits for RR_{ij} and OR_{ij} can be obtained using the exponential function.

If the confidence regions estimated by this procedure include zero the significance is reported in the table of differences as not significant. Otherwise only the relative magnitudes of the pair in question are indicated.

When the difference between two probabilities is positive, a very approximate estimate for the confidence limits for NNT can be obtained using the values for DP_{ij}.

As elsewhere in SIMFIT the significance level can be set by the user, and either natural or base ten logarithms can be plotted.

4.5.3 Trinomial proportions (trichotomous data)

The trinomial distribution is encountered in the analysis of trichotomous data, where N observations are made in a situation where there are only three possible disjoint categories, say x, y, or z. Data triples of counts in categories x, y, z can be any partitions, such as the number of male, female or dead hatchlings from a batch of eggs where it is hoped to determine a shift from equi-probable sexes.

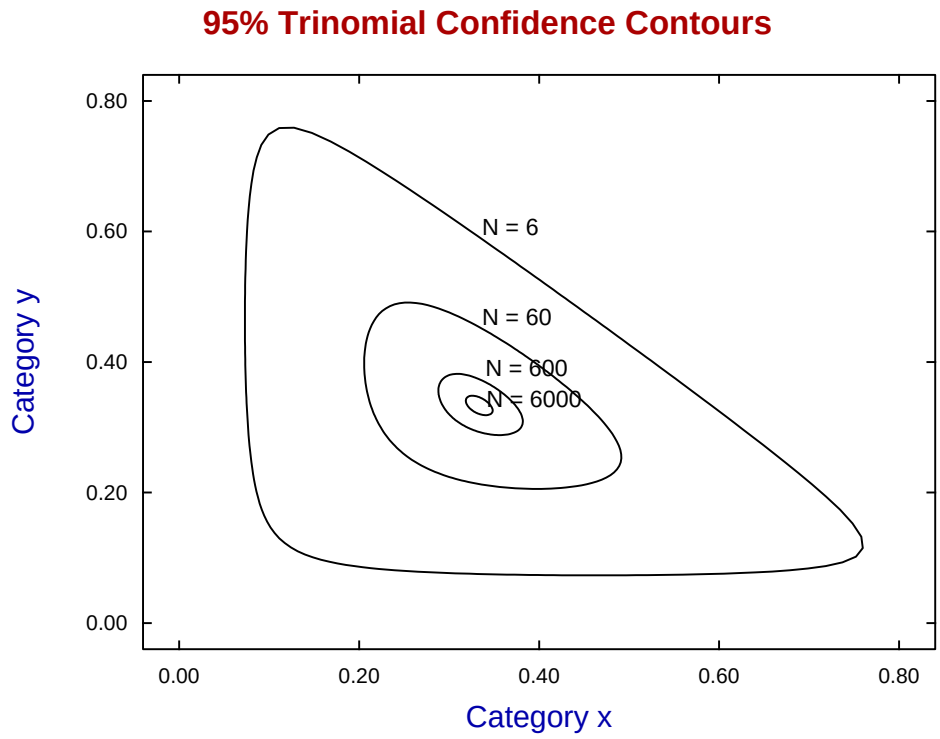
From the main SIMFIT menu select [A/Z], open program **binomial**, choose to plot trinomial confidence limits, then analyze the default data set `trinom.tf1` which has the following format.

N_x	N_y	N
2	2	6
20	20	60
200	200	600
2000	2000	6000

The format for trinomial analysis must be as in this data set as now summarized.

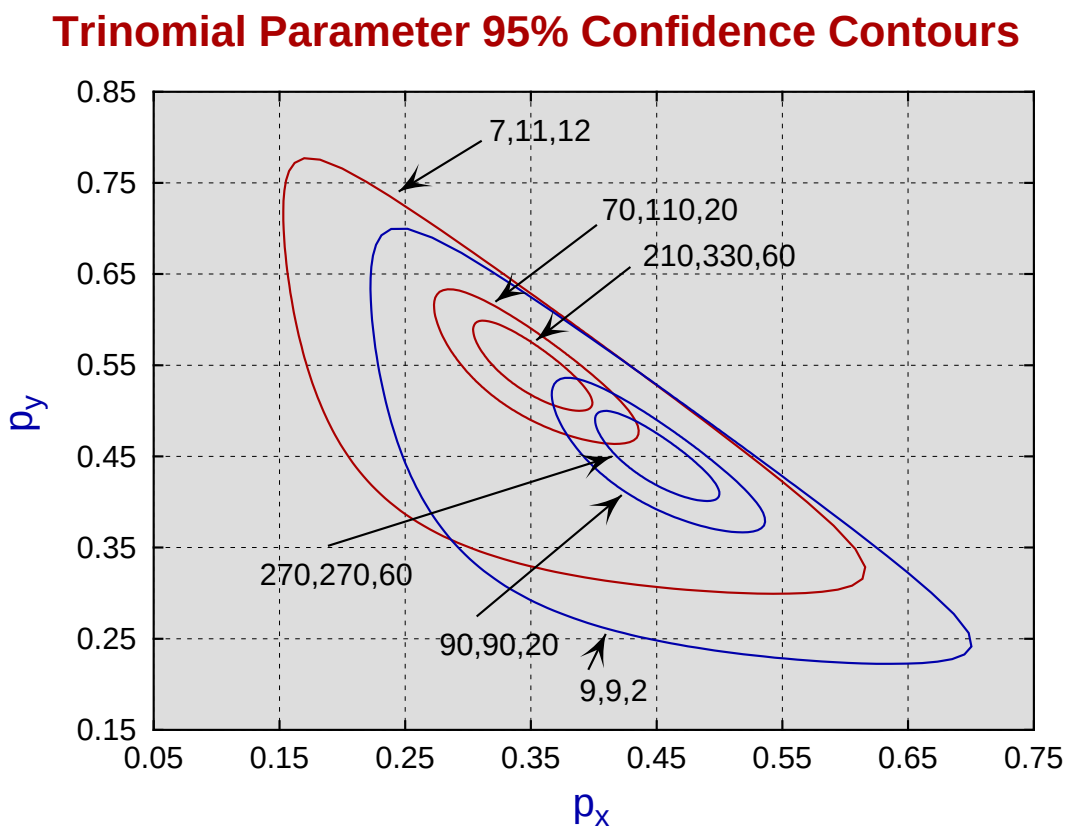
- Column 1: $N_x > 0$ . . . The number of times category x was observed
- Column 2: $N_y > 0$ . . . The number of times category y was observed
- Column 3: $N = N_x + N_y + N_z$ . . . The total number of observations

Clearly, the values in such a data matrix must all be non-negative integers subject to the constraint that column 1 plus column 2 cannot exceed column 3 in any row. After performing a chi-square test the following plot is displayed which powerfully demonstrates the contraction of the confidence regions as the sample size increases



The main value of this graphical technique is to examine confidence regions for overlap in order to better interpret the results from the chi-square test that SIMFIT always performs on such data matrices. Also, the plot can provide insight into the comparison of several different experiments using the following argument.

A useful rule of thumb to see if parameter estimates differ significantly is to check their approximate central 95% confidence regions. If the regions are disjoint it indicates that the parameters differ significantly and, in fact, parameters can differ significantly even with limited overlap. If two or more parameters are estimated, it is valuable to inspect the joint confidence regions defined by the estimated covariance matrix and appropriate chi-square critical value. Consider, for example, this figure generated by the contour plotting function of program **binomial**.



The contours are defined by

$$((\hat{p_x} - p_x), (\hat{p_y} - p_y)) \begin{bmatrix} p_x(1-p_x)/N & -p_x p_y/N \\ -p_x p_y/N & p_y(1-p_y)/N \end{bmatrix}^{-1} \begin{pmatrix} \hat{p_x} - p_x \\ \hat{p_y} - p_y \end{pmatrix} = \chi^2_{2:0.05}$$

where

$$N = N_x + N_y + N_z$$
$$\hat{p_x} = N_x/N$$
$$\text{and } \hat{p_y} = N_y/N$$

When $N = 20$ the triples 9,9,2 and 7,11,2 cannot be distinguished, but when $N = 200$ the orbits are becoming elliptical and converging to asymptotic values. By the time $N = 600$ the triples 210,330,60 and 270,270,60 can be seen to differ significantly.

Theory

If, in a trinomial distribution, the probability of category i is p_i for $i = 1, 2, 3$, then the probability P of observing n_i in category i in a sample of size $N = n_1 + n_2 + n_3$ from a homogeneous population is given by

$$P = \frac{N!}{n_1!n_2!n_3!} p_1^{n_1} p_2^{n_2} p_3^{n_3}$$

and the maximum likelihood estimates, of which only two are independent, are

$$\begin{aligned} \hat{p_1} &= n_1/N, \\ \hat{p_2} &= n_2/N, \\ \text{and } \hat{p_3} &= 1 - \hat{p_1} - \hat{p_2}. \end{aligned}$$

The bivariate estimator is approximately normally distributed, when N is large, so that

$$\begin{bmatrix} \hat{p_1} \\ \hat{p_2} \end{bmatrix} \sim MN_2 \left(\begin{bmatrix} p_1 \\ p_2 \end{bmatrix}, \begin{bmatrix} p_1(1-p_1)/N & -p_1p_2/N \\ -p_1p_2/N & p_2(1-p_2)/N \end{bmatrix} \right)$$

where MN_2 signifies the bivariate normal distribution. Consequently

$$((\hat{p_1} - p_1), (\hat{p_2} - p_2)) \begin{bmatrix} p_1(1-p_1)/N & -p_1p_2/N \\ -p_1p_2/N & p_2(1-p_2)/N \end{bmatrix}^{-1} \begin{pmatrix} \hat{p_1} - p_1 \\ \hat{p_2} - p_2 \end{pmatrix} \sim \chi_2^2$$

and hence, with probability 95%,

$$\frac{(\hat{p_1} - p_1)^2}{p_1(1-p_1)} + \frac{(\hat{p_2} - p_2)^2}{p_2(1-p_2)} + \frac{2(\hat{p_1} - p_1)(\hat{p_2} - p_2)}{(1-p_1)(1-p_2)} \leq \frac{(1-p_1-p_2)}{N(1-p_1)(1-p_2)} \chi^2_{2;0.05}.$$

Such inequalities define regions in the (p_1, p_2) parameter space which can be examined for statistically significant differences between $p_{i(j)}$ in samples from populations subjected to treatment j.

Hence, where regions are clearly disjoint for groups treated differently or for different samples, it can be concluded that parameters have been significantly affected by the treatments, as illustrated previously.

4.5.4 Cochran-Mantel-Haenszel meta analyis

Meta analysis is widely used in areas such as evidence based medicine in order to examine several studies of the same problem by different analysts, then extract the most plausible and objective overall conclusions. One common situation is where there are k alternative 2 by 2 contingency tables available, and worked examples to demonstrate the options available in SIMFIT to analyze this type of data set will now be presented.

Open the SIMFIT main menu, choose [Statistics], [Analysis of proportions], then [Meta Analysis] and examine the default test file `meta.tf1` which is formatted as follows.

y	N	x
126	226	1
35	96	1
908	1596	2
497	1304	2
913	1660	3
336	934	3
235	407	4
58	179	4
402	710	5
121	336	5
182	338	6
72	170	6
60	159	7
11	54	7
104	193	8
21	57	8

The format for SIMFIT meta analysis data files must be exactly as now summarized.

- The number of rows in the data matrix must be an even number.
- Distinct 2 by 2 contingency tables are included as sequential pairs of adjacent rows.
- Column 1 at row i must contain the number of critical outcomes $y_i \geq 0$, e.g. successful recovery.
- Column 2 at row i must contain the total number of observations $N_i \geq y_i$, and not $N_i - y_i$ which would be the complement of y_i, i.e. the number of failures to respond to treatment.
- Column 3 at row i must contain the control variable x for use in plotting.

Note that control variable x is not used in subsequent calculations, and it is only used for identifying the adjacent 2 by 2 contingency tables, and as a coordinate for plotting, which will be explained subsequently. Obviously, the value of x in rows j and $j + 1$ must be the same for $j = 1, 3, \ldots, k - 1$.

For instance, the first 2 by 2 contingency table that can be constructed from the data set is

126	100	*and not*	126	226
35	61		35	96

so we would have the probability estimates $\hat{p}_1 = 126/226$ and $\hat{p}_2 = 35/96$ and the odds ratio for this 2 by 2 contingency table would then be

$$2.196 = \frac{p_1/(1-p_1)}{p_2/(1-p_2)}.$$

Reading in this data set produces the following summary table.

To test H_0: equal binomial p-values

Number of 2 by 2 tables	8	
Overall sum of Y	4081	
Overall sum of N	8419	
Overall estimate of p	0.4847	95% confidence limits = (0.4740,0.4955)
$-2\log\lambda \quad (-2LL)$	310.9	$NDOF = 15$
$P(\chi^2 \geq -2LL)$	0.0000	*Reject H_0 at 1% significance level*
Chi-square test statistic (C)	306.9	$NDOF = 15$
$P(\chi^2 \geq C)$	0.0000	*Reject H_0 at 1% significance level*

Subsequent analysis leads to these results

Cochran-Mantel-Haenszel 2 by 2 by k Meta Analysis

y	N	Odds Ratio	$E[n(1,1)]$	$Var[n(1,1)]$
126	226	2.19600	113.00000	16.89720
35	96			
908	1596	2.14296	773.23448	179.30144
497	1304			
913	1660	2.17526	799.28296	149.27849
336	934			
235	407	2.85034	203.50000	31.13376
58	179			
402	710	2.31915	355.00000	57.07177
121	336			
182	338	1.58796	169.00000	28.33333
72	170			
60	159	2.36915	53.00000	9.00000
11	54			
104	193	2.00321	96.50000	11.04518
21	57			

H_0: conditional independence (all odds ratios = 1)
CMH Test Statistic = 279.4
$P(\chi^2 \geq CMH)$ = 0.0000 *Reject H_0 at 1% significance level*
Common Odds Ratio = 2.174, 95% confidence limits = (1.914,2.471)

Overall 2 by 2 contingency table

y	$N-y$
2930	2359
1151	1979

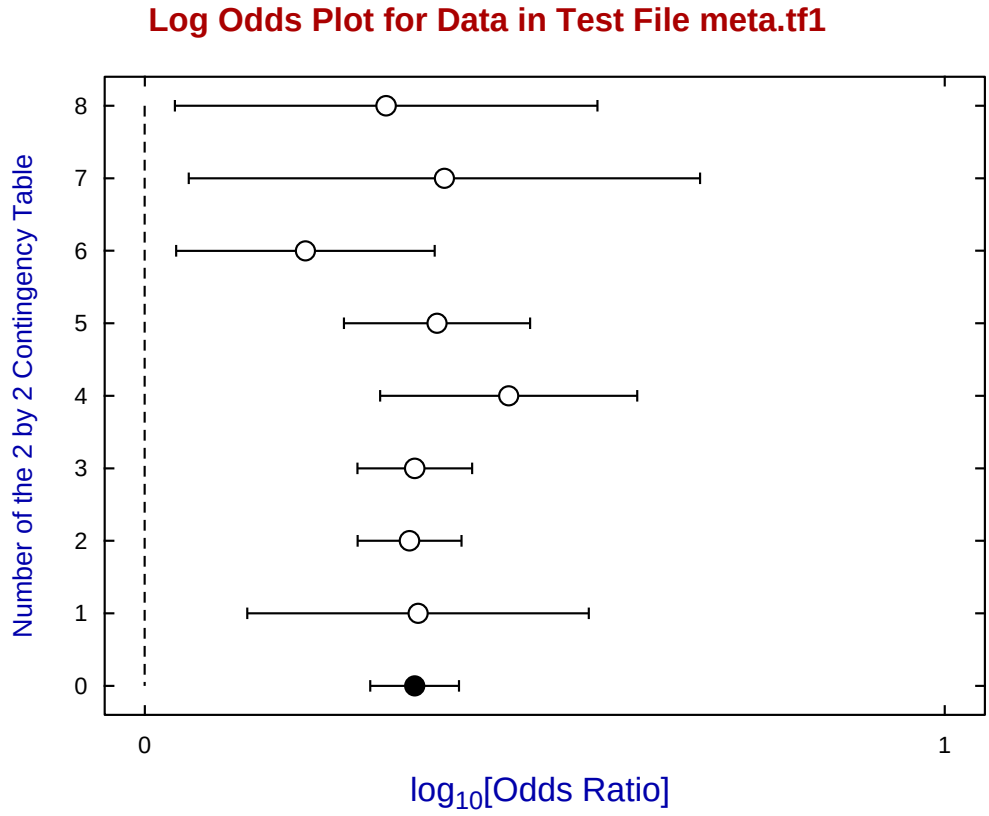
Overall Odds Ratio = 2.1360, 95% confidence limits = (1.950, 2.338)

The default log-odds plot for these 2 by 2 contingency tables can be easily viewed but to perform the editing necessary to create the next plot the following procedure has to be used.

1. Read in data and perform the meta analysis.
2. Display the default log odds plot using logarithms to base e or 10 as required.
3. Choose the [Advanced] option.
4. Select the [Avanced editing] option to transfer the data into the **simplot** procedure.
5. Note: this always transfers data into **simplot** in original not transformed coordinates.
6. Select the [Tansform] option, then the reverse y-semilog transformation.
7. The [Titles], [Labels], and [Legends] options can then be used for fine tuning as required.

Note that the sold circle represents the overall log odds ratio, while the dotted vertical line represents the reference position corresponding to the special case $p_1 = p_2$, which serves to indicate orders of magnitude deviation of the odds from the ideal case where the Odds = 1.

As the Odds are all greater than 1 with these data, the points displayed all lie to the right of this reference line.



Various other tables can be displayed, such as the next one which summarizes the differences and calculate NNT, the approximate number needed to treat.

$d_{i,j} = \hat{p}_i - \hat{p}_j, \quad NNT = 1/|d_{i,j}|$

Row(i)	Row(j)	$d_{i,j}$	lower-95%	upper-95%	Conclusion	$Var(d_{i,j})$	NNT	(95%c.l.)
1	2	0.19294	0.07691	0.30897	$p_1 > p_2$	0.00350	6	(3,14)
3	4	0.18779	0.15194	0.22364	$p_3 > p_4$	0.00033	6	(4,7)
5	6	0.19026	0.15127	0.22924	$p_5 > p_6$	0.00040	6	(4,7)
7	8	0.25337	0.16969	0.33706	$p_7 > p_8$	0.00182	4	(2,6)
9	10	0.20608	0.14312	0.26903	$p_9 > p_{10}$	0.00103	5	(3,7)
11	12	0.11493	0.02360	0.20626	$p_{11} > p_{12}$	0.00217	9	(4,43)
13	14	0.17365	0.04245	0.30486	$p_{13} > p_{14}$	0.00448	6	(3,24)
15	16	0.17044	0.02682	0.31406	$p_{15} > p_{16}$	0.00537	6	(3,38)

Zero cells

Contingency table analysis is compromised when cells have zero frequencies, as many of the usual summary statistics become undefined. Structural zeros can be handled by applying loglinear GLM analysis but sampling zeros presumably arise from small samples with extreme probabilities. Such tables can be analyzed by exact methods, but usually a positive constant is added to all the frequencies to avoid the problems.

The next table illustrates how this problem is handled in SIMFIT when analyzing data in the test file `meta.tf4`; the correction of adding 0.01 to all contingency tables frequencies being indicated.

Values ranging from 0.00000001 to 0.5 have been suggested elsewhere for this purpose, but all such choices are a compromise and, if possible, sampling should be continued until all frequencies are nonzero.

Cochran-Mantel-Haenszel 2 x 2 x k Meta Analysis

y	N	Odds Ratio	$E[n(1,1)]$	$Var[n(1,1)]$
*** 0.01 added to all cells for next calculation				
0	6	0.83361	0.01091	0.00544
0	5			
*** 0.01 added to all cells for next calculation				
3	6	601.00000	1.51000	0.61686
0	6			
*** 0.01 added to all cells for next calculation				
6	6	1199.00995	4.01000	0.73008
2	6			
*** 0.01 added to all cells for next calculation				
5	6	0.00825	5.51000	0.25454
6	6			
*** 0.01 added to all cells for next calculation				
2	2	0.40120	2.01426	0.00476
5	5			

H_0: conditional independence (all odds ratios = 1)

CMH Test Statistic = 386.2
$P(\chi^2 \geq CMH)$ = 0.0494, *Reject H_0 at 5% significance level*
Common Odds Ratio = 6.749, 95% confidence limits = (1.144, 39.81)

Overall 2 by 2 table

y	$N-y$
16	10
13	15

Overall Odds Ratio = 1.842, 95% confidence limits = (0.6241,5.435)

Creating composite log odds plots

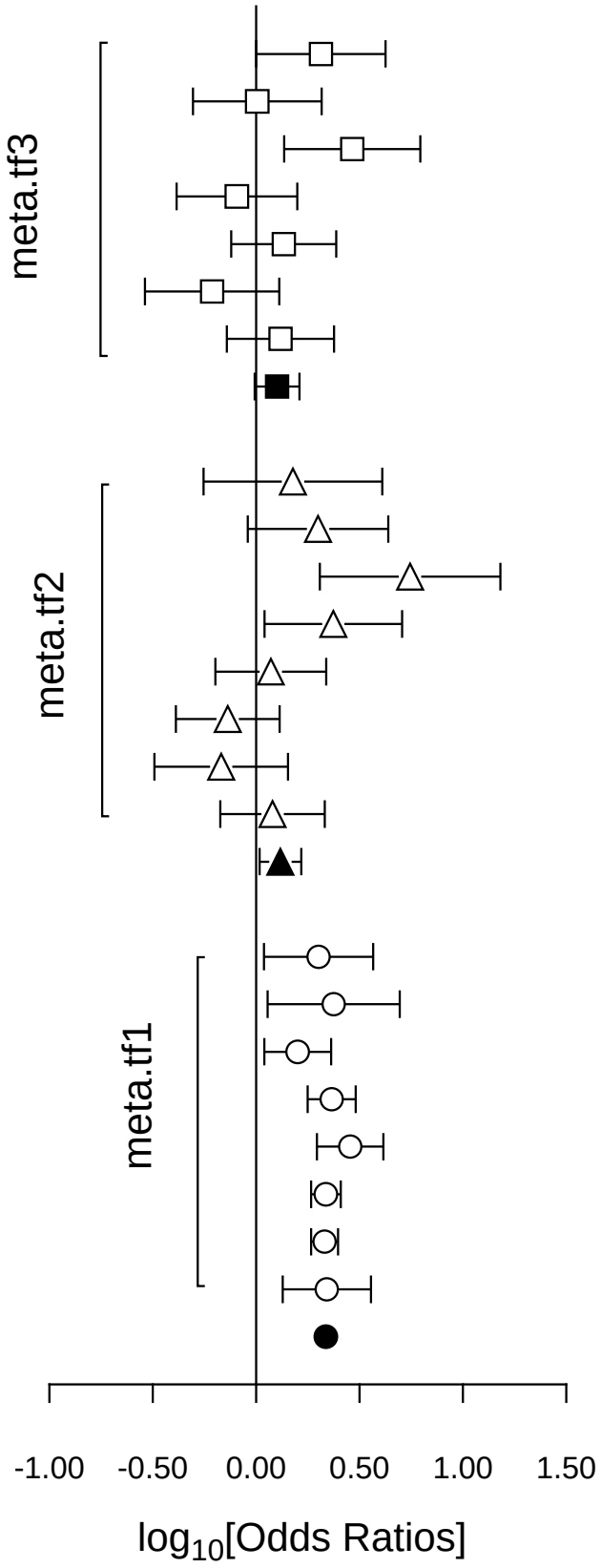
It is often necessary to create extensive log odds plots for three main reasons.

1. A single large data set is presented for analysis.
   This presents no problems if the control variables have been set correctly. However, if the graph becomes crowded it will need to be stretched.

2. Several data sets are available.
   These can be combined into a single data set by copying and pasting, or by using the SIMFIT program **editmt**. However the control variables must already be consistent for this purpose or can be made so by editing at the same time.

3. Several individual log odds plots are available.
   In this case individual coordinate files can be saved then combined as a library file for SIMFIT program **simplot** to make a composite plot. For this purpose the control variables on the individual data sets must be consistent to control spacing.

To illustrate these issues of spacing and stretching a worked example follows.



(1) **The data**
Test files `meta.tf1`, `meta.tf2`, and `meta.tf3` were analyzed in sequence using the SIMFIT Meta Analysis procedure. Note that, in these files, column 3 contains spacing coordinates so that data will be plotted consecutively.

(2) **The ASCII coordinate files**
During Meta Analysis, $100(1-\alpha)\%$ confidence limits on the Log-Odds-Ratio resulting from a 2 by 2 contingency tables with cell frequencies n_{ij} can be constructed from the approximation $\hat{e}$ where

$$\hat{e} = Z_{\alpha/2}\sqrt{\frac{1}{n_{11}}+\frac{1}{n_{12}}+\frac{1}{n_{21}}+\frac{1}{n_{22}}}.$$

When Log-Odds-Ratios with error bars are displayed, the overall values (shown as filled symbols) with error bars are also plotted with a x coordinate one less than smallest x value on the input file. For this figure, error bar coordinates were transferred into the project archive using the [Advanced] option to save ASCII coordinate files.

(3) **Creating the composite plot**
Program **simplot** was opened and the six error bar coordinate files were retrieved from the project archive. Experienced users would do this more easily using a library file of course. Reverse y-semilog transformation was selected, symbols were chosen, axes, title, and legends were edited, then half bracket hooks identifying the data were added as arrows and extra text.

(4) **Creating the PostScript file**
Vertical format was chosen then, using the option to stretch PostScript files, the y coordinate was stretched by a factor of two.

(5) **Editing the PostScript file**
To create the final PostScript file for LaTeX a tighter bounding box was calculated using **gsview** then, using **notepad**, clipping coordinates at the top of the file were set equal to the BoundingBox coordinates, to suppress excess white space. This can also be done using the [Style] option to omit painting a white background, so that PostScript files are created with transparent backgrounds, i.e. no white space, and clipping is irrelevant.

Theory

A pair of success/failure classifications with y successes in N trials, i.e. with frequencies $n_{11} = y_1$, $n_{12} = N_1 - y_1$, $n_{21} = y_2$, and $n_{22} = N_2 - y_2$, results in a 2 by 2 contingency table, and meta analysis is used for exploring k sets of such 2 by 2 contingency tables. That is, each row of each table is a pair of numbers of successes and number of failures, so that the Odds ratio in contingency table k can be defined as

$$\begin{aligned}\text{Odds ratio}_k &= \frac{y_{1k}/(N_{1k} - y_{1k})}{y_{2k}/(N_{2k} - y_{2k})} \\ &= \frac{n_{11k}n_{22k}}{n_{12k}n_{21k}}.\end{aligned}$$

Typically, the individual contingency tables would be for partitioning of groups before and after treatment, and a common situation would be where the aim of the meta analysis would be to assess differences between the results summarized in the individual contingency tables, or to construct a best possible Odds ratio taking into account the sample sizes for appropriate weighting. Suppose, for instance, that contingency table number k is

n_{11k}	n_{12k}	n_{1+k}
n_{21k}	n_{22k}	n_{2+k}
n_{+1k}	n_{+2k}	n_{++k}

where the marginals are indicated by plus signs in the usual way. Then, assuming conditional independence and a hypergeometric distribution, the mean and variance of n_{11k} are given by

$$\begin{aligned}E(n_{11k}) &= n_{1+k}n_{+1k}/n_{++k} \\ V(n_{11k}) &= \frac{n_{1+k}n_{2+k}n_{+1k}n_{+2k}}{n_{++k}^2(n_{++k} - 1)},\end{aligned}$$

and, to test for significant differences between m contingency tables, the Cochran-Mantel-Haenszel test statistic CMH, given by

$$CMH = \frac{\left\{\left|\sum_{k=1}^{m}(n_{11k} - E(n_{11k}))\right| - \frac{1}{2}\right\}^2}{\sum_{k=1}^{m} V(n_{11k})}$$

can be regarded as an approximately chi-square variable with one degree of freedom. Some authors omit the continuity correction and sometimes the variance estimate is taken to be

$$\hat{V}(n_{11k}) = n_{1+k}n_{2+k}n_{+1k}n_{+2k}/n_{++k}^3.$$

The estimated common odds ratio $\hat{\theta}_{MH}$ presented in the previous tables is calculated allowing for random effects using

$$\hat{\theta}_{MH} = \frac{\sum_{k=1}^{m}(n_{11k}n_{22k}/n_{++k})}{\sum_{k=1}^{m}(n_{12k}n_{21k}/n_{++k})},$$

while the variance is used to construct the confidence limits from

$$\hat{\sigma}^2[\log(\hat{\theta}_{MH})] = \frac{\sum_{k=1}^{m}(n_{11k}+n_{22k})n_{11k}n_{22k}/n_{++k}^2}{2\left(\sum_{k=1}^{m}n_{11k}n_{22k}/n_{++k}\right)^2}$$
$$+\frac{\sum_{k=1}^{m}[(n_{11k}+n_{22k})n_{12k}n_{21k}+(n_{12k}+n_{n21k})n_{11k}n_{22k}]/n_{++k}^2}{2\left(\sum_{k=1}^{m}n_{11k}n_{22k}/n_{++k}\right)\left(\sum_{k=1}^{m}n_{12k}n_{21k}/n_{++k}\right)}$$
$$+\frac{\sum_{k=1}^{m}(n_{12k}+n_{21k})n_{12k}n_{21k}/n_{++k}^2}{2\left(\sum_{k=1}^{m}n_{12k}n_{21k}/n_{++k}\right)^2}.$$

Also, in these tables, the overall 2 by 2 contingency table using the pooled sample assuming a fixed effects model is listed for reference, along with the overall odds ratio and estimated confidence limits calculated using the expressions presented elsewhere for an arbitrary log odds ratio.

The table of differences illustrates another technique to study sets of 2 by 2 contingency tables. SimFIT can calculate all the standard probability statistics for sets of paired experiments. In this case the pairwise differences are illustrated along with the number needed to treat i.e. $NNT = 1/|d|$, but it should be remembered that such estimates have to be interpreted with care. For instance, the differences and log ratios change sign when the rows are interchanged.

Again, it should be emphasized that SimFIT outputs values and confidence limits both for the differences $d_{1,2} = \hat{p}_1 - \hat{p}_2$ and the calculated $NNT = 1/d_{1,2}$ values, but the choice between these quantities for data interpretation is controversial. To appreciate the reason why a value of NNT calculated from a sample is just a coarse estimate of the size of a sample needed to treat in order to obtain one additional cure, and could be very misleading, consider the situation of binomial trials with exactly known probabilities p_1 and p_2, and $p_1 > p_2$. The condition that the expectation of a binomial variable X_1 with probability p_1 should be one greater than than a binomial variable X_2 with probability p_2 given a sample size N is

$$E(X_1) = E(X_2) + 1$$
$$Np_1 = Np_2 + 1, \text{ so that}$$
$$N = \frac{1}{p_1 - p_2}.$$

Of course NNT calculated from data is not the exact N as just derived but is given by the random function

$$NNT = \frac{1}{\hat{p}_1 - \hat{p}_2}$$

where there is experimental uncertainty in the parameter estimates. This is one reason why many experts recommend relying on conclusions based directly on the difference $d_{1,2}$, because this quantity is more robust for the purpose of hypothesis testing than NNT where reciprocation exaggerates random effects. Another reason is that it is possible to calculate accurate confidence limits for the difference $d_{1,2}$, but confidence limits calculated for NNT are unsymmetrical and much less intuitive. It just seems more informative to say, for instance, that with a possible error of up to 5%, a treatment improves the chance of cure from approximately 10 to 20%, or say from 60 to 70%, than to simply report NNT = 10 to cover all possible 10% improvements

4.5.5 Bioassay, dose response curves and LD50

It is often of interest to fit a model to data in order to estimate parameters such as the 50% point from dose-response curves, and SimFIT provides several dedicated programs for this purpose such as the following.

- **exfit**: fits one or sums of exponentials and calculates the area under the curve (AUC).
- **mmfit**: fits one or sums of Michaelis-Menten models and calculates the apparent K_m.
- **hlfit**: fits one or sums of binding models and calculates the apparent K_a.
- **sffit**: fits cooperative binding models and calculates half saturation points.
- **gcfit**: fits nonlinear growth models and calculates maximal growth rates.
- **inrate**: fits several models and calculates initial rates.
- **polnom**: fits polynomials and calculates y given x.
- **calcurve**: fits cubic splines and calculates y given x.
- **qnfit**: fits user defined models and calculates y given x.

These programs all assume uncorrelated normally distributed errors, but there are many procedures, such as bioassay, dose-response curves, determination of LD50, or EC50 etc. where binomially distributed errors would be more appropriate, and so it would be better to fit general linear models (GLM)

This would be a situation such as the following dose-response data set contained in the default test file `ld50.tf1` which can be inspected after opening the main SimFIT menus, followed by selecting [Statistics], [Analysis of proportions], then [Bioassay, dose response curves and LD50].

y	N	x
1	10	1
4	20	2
4	10	3
5	10	4
15	30	5
7	10	6
9	10	7
12	15	8
9	10	9
8	10	10

Data for determination of LD50 by GLM requires the above format as follows for k groups and $i = 1, 2, \ldots, k$.

- Column 1: $y_i \geq 0$, the number of animals dying in group i
- Column 2: $N_i \geq y_i$, the number of animals tested in group i
- Column 3: $x_i \geq 0$, the amount of poison being tested on group i

If the k groups are all independent and each group is homogeneous, i.e., each animal in the group has exactly the same probability p of dying given the same time of exposure to poison at amount x for the same period of time, then y is binomially distributed and p_i can be estimated as $\hat{p}_i = y_i/N_i$, together with exact confidence limits.

It is usual to investigate a data set to choose a model with the lowest deviance and the next table shows the results from analysis of the data in the default test file using the three GLM link functions indicated.

Method: GLM with binomial errors, Link: Logistic
Number of groups = 10, Deviance = 4.246

Parameter	Value	Standard error	Lower 95%cl	Upper 95%cl	*p*
Constant	-2.0986	0.4733	-3.190	-1.007	0.0022
Slope	0.45070	0.08725	0.2495	0.6519	0.0009
50% point	4.6564	0.4441	3.632	5.681	0.0000

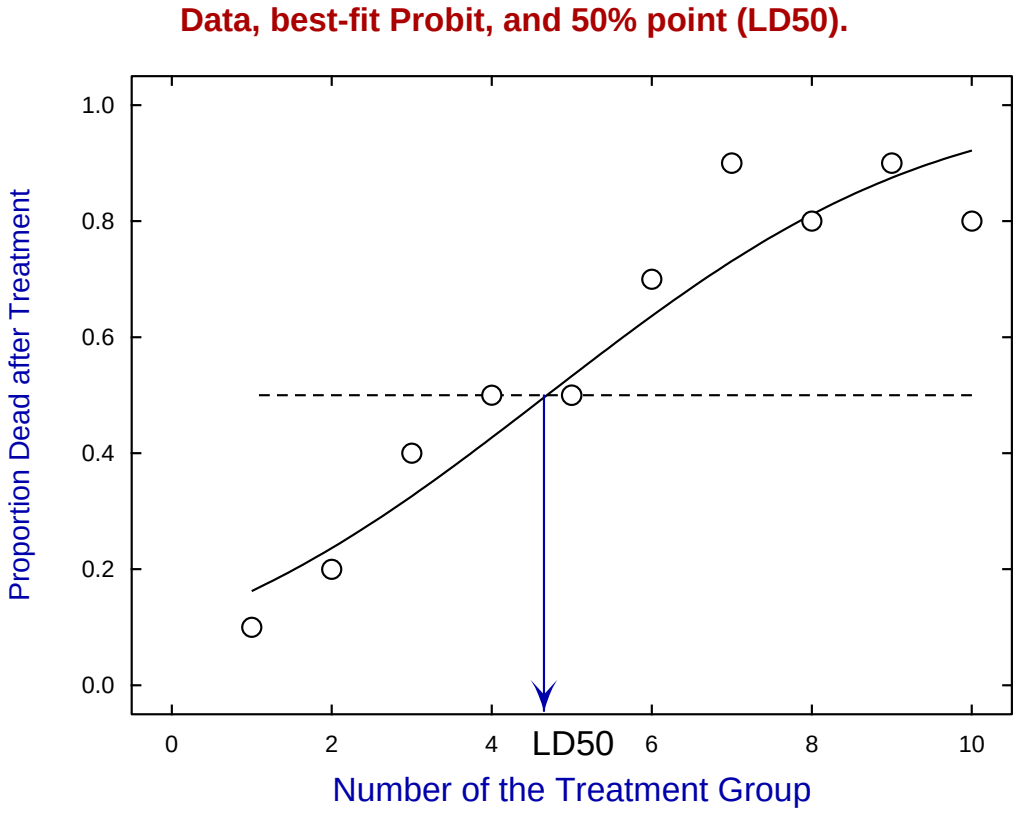
Method: GLM with binomial errors, Link: Probit
Number of groups = 10, Deviance = 4.564

Parameter	Value	Standard error	Lower 95%cl	Upper 95%cl	*p*
Constant	-1.2513	0.2708	-1.876	-0.6269	0.0017
Slope	0.26678	0.04855	0.1548	0.3787	0.0006
50% point	4.6902	0.4463	3.661	5.719	0.0000

Method: GLM with binomial errors, Link: Complementary log-log
Number of groups = 10, Deviance = 6.600

Parameter	Value	Standard error	Lower 95%cl	Upper 95%cl	*p*
Constant	-1.6696	0.3295	-2.429	-0.9097	0.0010
Slope	0.26635	0.05079	0.1492	0.3835	0.0008
50% point	4.89220	0.5182	3.697	6.087	0.0000

In this case the logistic and probit models give a similar fit, which is somewhat better than the complementary log-log, and so the standard probit graph is shown next.

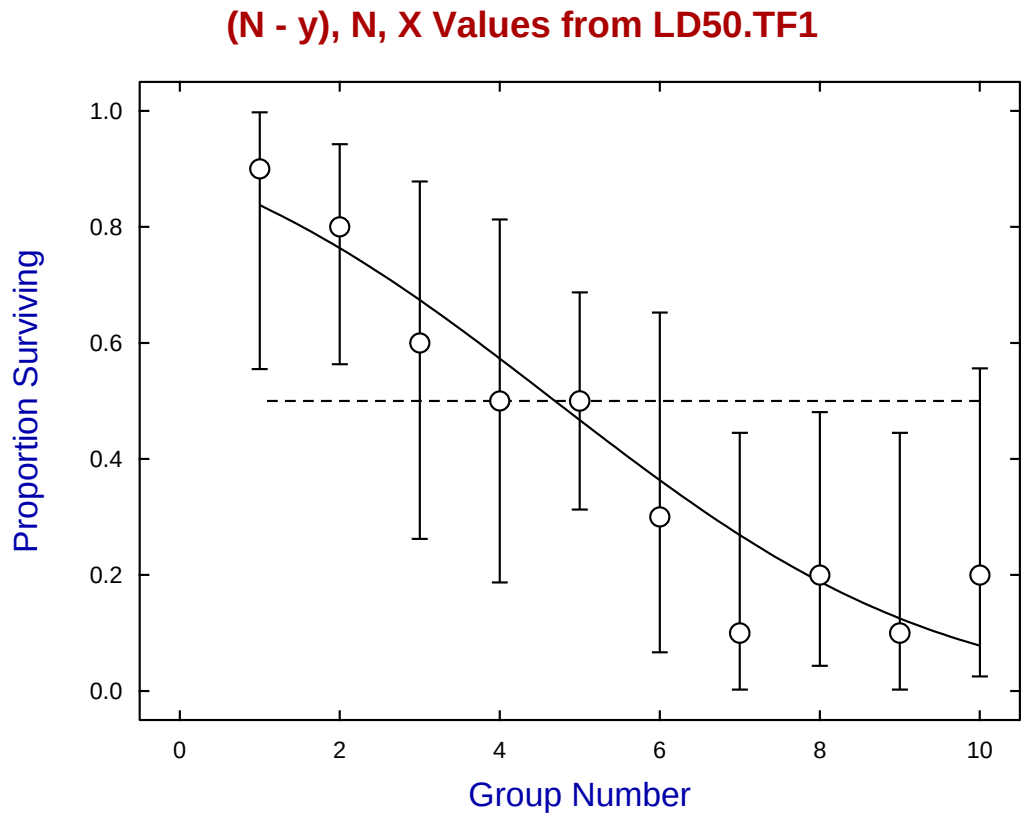


Various options are available for testing the goodness of fit by plotting residuals or inspecting tables of residuals as shown next.

Number	Y-value	Theory	Deviance	Leverages
1	1	1.624	-0.5692	0.2308
2	4	4.729	-0.3912	0.3731
3	4	3.260	0.4907	0.1391
4	5	4.270	0.4645	0.1030
5	15	15.99	-0.3612	0.2652
6	7	6.366	0.4227	0.09853
7	9	7.311	1.328	0.1282
8	12	12.17	-0.1118	0.2492
9	9	8.749	0.2476	0.1985
10	8	9.217	-1.219	0.2143

Note that exact 95% confidence limits can also be plotted but, as these can be large and very distracting with small samples, they can be switched off.

A further point can be made about this GLM procedure. Suppose that, instead of a file with y, N, x for the proportion failing, we input a file with $N - y, N, x$. This would then be the proportion surviving as plotted below.



Note that this change from proportion failing to the complement, that is the proportion surviving, leads to exactly the same estimate for LD50.

Another variant of the this technique is that a parameter can be changed in order to estimate other percentiles than the 50% point, e.g., LD25, LD75, EC25, ID25, ED75, etc.

Theory 1: GLM

It is important to understand that fitting dose-response curves in the manner just described does not correspond to the usually understood technique of adjusting the parameters of a deterministic mathematical model by optimization to obtain a best-fit curve that minimizes the sum of squared residuals. For this reason a brief overview of generalized linear modeling (GLM) is now presented.

To understand the motivation for this technique, it is usual to refer to a typical doubling dilution experiment in which diluted solutions from a stock containing infected organisms are plated onto agar in order to count infected plates, and hence estimate the number of organisms in the stock. Suppose that before dilution the stock had N organisms per unit volume, then the number per unit volume after $x = 0, 1, \ldots, m$ dilutions will follow a Poisson dilution with $\mu_x = N/2^x$. Now the chance of a plate receiving no organisms at dilution x is the first term in the Poisson distribution , that is $\exp(-\mu_x)$, so if p_x is the probability of a plate becoming infected at dilution x, then

$$p_x = 1 - \exp(-\mu_x),\ x = 1, 2, \ldots, m.$$

Evidently, where the p_x have been estimated as proportions from y_x infected plates out of n_x plated at dilution x, then N can be estimated using

$$\log[-\log(1 - p_x)] = \log N - x \log 2$$

considered as a maximum likelihood fitting problem of the type

$$\log[-\log(1 - p_x)] = \beta_0 + \beta_1 x$$

where the errors in estimated proportions $p_x = y_x/n_x$ are binomially distributed. So, to fit a generalized linear model, you must have independent evidence to support your choice for an assumed error distribution for the dependent variable Y from the normal, binomial, Poisson, or gamma distributions, in which it is supposed that the expectation of Y is to be estimated, i.e.,

$$E(Y) = \mu.$$

The associated *pdfs* are parameterized as follows.

$$\begin{aligned}
\text{normal}: f_Y &= \frac{1}{\sqrt{2\pi}\,\sigma} \exp\left(-\frac{(y-\mu)^2}{2\sigma^2}\right) \\
\text{binomial}: f_Y &= \binom{N}{y} \pi^y (1-\pi)^{N-y} \\
\text{Poisson}: f_Y &= \frac{\mu^y \exp(-\mu)}{y!} \\
\text{gamma}: f_Y &= \frac{1}{\Gamma(\nu)} \left(\frac{\nu y}{\mu}\right)^{\nu} \exp\left(-\frac{\nu y}{\mu}\right) \frac{1}{y}
\end{aligned}$$

It is a mistake to make the usual unwarranted assumption that measurements imply a normal distribution, while proportions imply a binomial distribution, and counting processes imply a Poisson distribution, unless the error distribution assumed has been verified for your data. Another very questionable assumption that has to made is that a predictor function η exists, which is a linear function of the m covariates, i.e., independent explanatory variables, as in

$$\eta = \sum_{j=1}^{m} \beta_j x_j.$$

Finally, yet another dubious assumption must be made, that a link function $g(\mu)$ exists between the expected value of Y and the linear predictor. The choice for

$$g(\mu) = \eta$$

depends on the assumed distribution as follows. For the binomial distribution, where y successes have been observed in N trials, the link options are the logistic, probit or complementary log-log

$$\text{logistic: } \eta = \log\left(\frac{\mu}{N-\mu}\right)$$
$$\text{probit: } \eta = \Phi^{-1}\left(\frac{\mu}{N}\right)$$
$$\text{complementary log-log: } \eta = \log\left(-\log\left(1-\frac{\mu}{N}\right)\right).$$

Where observed values can have only one of two values, as with binary or quantal data, it may be wished to perform binary logistic regression. This is just the binomial situation where y takes values of 0 or 1, N is always set equal to 1, and the logistic link is selected. However, for the normal, Poisson and gamma distributions the link options are

$$\text{exponent: } \eta = \mu^a$$
$$\text{identity: } \eta = \mu$$
$$\text{log: } \eta = \log(\mu)$$
$$\text{square root: } \eta = \sqrt{\mu}$$
$$\text{reciprocal: } \eta = \frac{1}{\mu}.$$

In addition to these possibilities, you can supply weights and install an offset vector along with the data set, the regression can include a constant term if requested, the constant exponent a in the exponent link can be altered, and variables can be selected for inclusion or suppression in an interactive manner. However, note that the same strictures apply as for all regressions: you will be warned if the SVD has to be used due to rank deficiency and you should redesign the experiment until all parameters are estimable and the covariance matrix has full rank, rather than carry on with parameters and standard errors of limited value.

Theory 2: 95% confidence range in inverse prediction

The calculation of confidence limits for derived values, such as LD50 in the present case, that are obtained from the parameter estimates from fitting along with the estimated parameter covariance matrix should be noted.

polnom estimates non-symmetrical confidence limits assuming that the N values of y for inverse prediction and weights supplied for weighting are exact, and that the model fitted has n parameters that are justified statistically. **calcurve** uses the weights supplied, or the estimated coefficient of variation, to fit confidence envelope splines either side of the best fit spline, by employing an empirical technique developed by simulation studies. Root finding is employed to locate the intersection of the y_i supplied with the envelopes. The AUC, LD50, half-saturation, asymptote and other inverse predictions in SimFIT use a t distribution with $N-n$ degrees of freedom, and the variance-covariance matrix estimated from the regression. That is, assuming a prediction parameter defined by $p = f(\theta_1, \theta_2, \ldots, \theta_n)$, a central 95% confidence region is constructed using the prediction parameter variance estimated by the propagation of errors formula

$$\hat{V}(p) = \sum_{i=1}^{n}\left(\frac{\partial f}{\partial \theta_i}\right)^2 \hat{V}(\theta_i) + 2\sum_{i=2}^{n}\sum_{j=1}^{i-1}\frac{\partial f}{\partial \theta_i}\frac{\partial f}{\partial \theta_j}\hat{CV}(\theta_i, \theta_j).$$

Note that this formula for the propagation of errors can be used to calculate parameter standard errors for parameters that are calculated as functions of parameters that have been estimated by fitting, such as apparent maximal velocity when fitting sums of Michaelis-Menten functions. However, such estimated standard errors will only be very approximate.

5 Statistical calculations

Tutorials and worked examples for simulation, curve fitting, statistical analysis, and plotting.
https://simfit.uk
https://simfit.org.uk
https://simfit.silverfrost.com

5.1 Power and sample size

Hypothesis testing is based upon specifying a null hypothesis H_0 then testing to see if a statistic calculated from the data is sufficiently extreme to justify rejecting the null hypothesis. There are two possible errors.

- **Type I error**
  The null hypothesis is rejected when it is true and the probability of this happening is α.
- **Type II error**
  The null hypothesis is accepted when it is false and the probability of this happening is β.

The significance level is α while the power is $1-\beta$, often expressed as a percentage. The situation can be summarized in the following table.

Decision	H_0 is true	H_0 is false
Reject H_0	Type I error α	Correct $1-\beta$
Accept H_0	Correct $1-\alpha$	Type II error β

Calculations related to power as a function of sample size can be performed as long as the statistical distributions and parameters required for the null hypothesis are correct and specified. Unfortunately, while calculation of α is straightforward, calculation of β requires that an alternative hypothesis H_A be specified and can be much more difficult.

In any given situation it may be necessary to estimate the sample size n required given α and β, or to estimate β given α and n, as it is not possible to simultaneously minimize α and β. If n is fixed, then increasing α decreases β, while decreasing α increases β. The following cases are discussed here.

1. One binomial sample
2. Two binomial samples
3. One normal sample
4. Two normal samples
5. Multiple normal samples
6. One and two variances
7. One and two correlations
8. The chi-squared test

For each of these cases, the minimum essential theoretical details are given followed by typical examples. Plots of power as a function of sample size can also be created. Finally, a more comprehensive description of the underlying theory is given.

1. Power calculations for 1 binomial sample

The calculations are based on the binomial test, the binomial distribution, and the normal approximation to it for large samples and p not close to 0 or 1.
If the theoretical binomial parameters p_0 and $q_0 = 1 - p_0$ are not too close to 0 or 1 and it is wished to estimate this with an error of at most δ, then the sample size required is

$$n = \frac{Z^2_{\alpha/2} p_0 q_0}{\delta^2},$$
$$\text{where } P(Z > Z_{\alpha/2}) = \alpha/2,$$
$$\text{or } \Phi(Z_{\alpha/2}) = 1 - \alpha/2,$$

which, for many purposes, can be approximated by $n \approx 1/\delta^2$. The power in a binomial or sign test can be approximated, again if the sample estimates p_1 and $q_1 = 1 - p_1$ are not too close to 0 or 1, by

$$1 - \beta = P\left(Z < \frac{p_1 - p_0}{\sqrt{p_0 q_0 / n}} - Z_{\alpha/2}\sqrt{\frac{p_1 q_1}{p_0 q_0}}\right) + P\left(Z > \frac{p_1 - p_0}{\sqrt{p_0 q_0 / n}} + Z_{\alpha/2}\sqrt{\frac{p_1 q_1}{p_0 q_0}}\right).$$

Example 1

This demonstrates calculations that are possible when, in a sample of size n, x successes are used to estimate the binomial parameter $\hat{p} = x/n$. Given a binomial distribution with H_0 : parameter $p = p_0$, H_A : parameter $p = p_1$, and α specified, then three calculations are possible, namely

1. Calculate $n(\delta)$, i.e. n giving an error at most δ
2. Calculate $n(\beta)$, i.e. n given β
3. Calculate $\beta(n)$, i.e. β given n

as summarized in the next table.

Example 1: Power analysis for 1 binomial sample

For $n(\delta)$	α = 0.05	p = 0.5	δ = 0.1		n = 96
For $n(\beta)$	α = 0.05	β = 0.2	p_0 = 0.5	p_1 = 0.6	n = 192
For $\beta(n)$	α = 0.05	p_0 = 0.5	p_1 = 0.6	power = 80%	n = 192

The conclusion is that a sample size of 96 is required to ensure that the parameter estimated is within 0.1 of the true parameter $p = 0.5$ for 95% of the results from repeated samples, while to confirm that the parameter is distinct from $p = 0.6$ with 80% power requires a sample size of 192.

2. Power calculations for 2 binomial samples

For two sample proportions p_1 and p_2 that are similar and not too close to 0 or 1, as in a 2 by 2 contingency table, the sample size n and power $1 - \beta$ associated with a binomial test for $H_0 : p_{01} = p_{02}$ can be estimated using one of numerous methods based upon normal approximations. For example

$$n = \frac{(p_1 q_1 + p_2 q_2)(Z_{\alpha/2} + Z_\beta)^2}{(p_1 - p_2)^2},$$
$$Z_\beta = \sqrt{\frac{n(p_1 - p_2)^2}{p_1 q_1 + p_2 q_2}} - Z_{\alpha/2},$$
$$\beta = P(Z \geq Z_\beta),$$
$$1 - \beta = \Phi(Z_\beta).$$

Example 2a

This deals with the situation where two samples of size n are analyzed to determine if the estimates $\hat{p}_1 = x_1/n$ and $\hat{p}_2 = x_2/n$ differ significantly. Parameters can also be input as log odds ratios.

Example 2a. Power analysis for 2 binomial samples

$\beta(n)$	α = 0.05	p_1 = 0.6	p_2 = 0.7	power = 32%	n = 100
$\beta(n)$	α = 0.05	p_1 = 0.6	p_2 = 0.7	power = 56%	n = 200
$\beta(n)$	α = 0.05	p_1 = 0.6	p_2 = 0.7	power = 73%	n = 300
$n(\beta)$	α = 0.05	β = 0.2	p_1 = 0.6	p_2 = 0.7	n = 353

Note that, for $p_1 = 0.6$ and $p_2 = 0.7$ the power increases from 32%, to 56%, to 73% as n increases from 100, to 200, to 300, while a sample of size $n = 353$ is required to achieve 80% power.

Example 2b

Power for the Fisher exact test with sample size n used to estimate both p_1 and p_2, as for the binomial test, can be calculated using

$$1 - \beta = 1 - \sum_{r=0}^{2n} \sum_{C_r} \binom{n}{x}\binom{n}{r-x},$$

where $r =$ total successes,
$x =$ number of successes in the group,
and $C_r =$ the critical region.

This can be inverted by SIMFIT to estimate n, but unfortunately the sample sizes required may be too large to implement by the normal procedure of enumerating probabilities for all 2 by 2 contingency tables with consistent marginals.

Example 2b. Power analysis for the Fisher Exact Test

$\beta(n)$	α = 0.05	p_1 = 0.6	p_2 = 0.7	power = 37%	n = 100
$\beta(n)$	α = 0.05	p_1 = 0.6	p_2 = 0.7	power = 64%	n = 200
$n(\beta)$	α = 0.05	β = 0.2	p_1 = 0.6	p_2 = 0.7	n = 304

For sample sizes of 100 and 200 the power is 37% and 64% but sample sizes of 304 are required for 80% power.

3. Power calculations for 1 normal sample

The calculations for a 1 sample t test are based upon the confidence limit formula for the population mean μ from a sample of size n, using the sample mean $\bar{x}$, sample variance s^2 and the t distribution, as follows

$$P\left(\bar{x} - t_{\alpha/2,n-1}\frac{s}{\sqrt{n}} \le \mu \le \bar{x} + t_{\alpha/2,n-1}\frac{s}{\sqrt{n}}\right) = 1 - \alpha,$$

$$\text{where } \bar{x} = \sum_{i=1}^{n} x_i/n,$$

$$s^2 = \sum_{i=1}^{n} (x_i - \bar{x})^2/(n-1),$$

$$P(t \le t_{\alpha/2,\nu}) = 1 - \alpha/2,$$

$$\text{and } \nu = n - 1.$$

You input the sample variance, which should be calculated using a sample size comparable to those predicted above. Power calculations can be done using the half width $h = t_{\alpha/2,n-1}\, s/\sqrt{n}$, or using the absolute difference δ between the population mean and the null hypothesis mean as argument. The following options are available:

- $n(h)$, i.e. to calculate the sample size necessary to estimate the true mean within a half width h

$$n = \frac{s^2 t^2_{\alpha/2,n-1}}{h^2};$$

- $n(\delta)$, i.e. to calculate the sample size necessary for an absolute difference δ

$$n = \frac{s^2}{\delta^2}(t_{\alpha/2,n-1} + t_{\beta,n-1})^2; \text{ or}$$

- $\beta(n)$, i.e. to estimate the power

$$t_{\beta,n-1} = \frac{\delta}{\sqrt{s^2/n}} - t_{\alpha/2,n-1}.$$

It should be noted that the sample size occurs in the degrees of freedom for the t distribution, necessitating an iterative solution to estimate n.

Example 3

Example 3. Power analysis for 1 sample t test

$n(h)$	h = 1	α = 0.05		s^2 = 1	n = 7
$n(\delta)$	δ = 1	α = 0.05	β = 0.2	s^2 = 1	n = 10
$\delta(n,\beta)$	n = 10	α = 0.05	β = 0.2	s^2 = 1	δ = 0.9947
$\beta(n)$	n = 10	δ = 1	α = 0.05	s^2 = 1	β = 0.1958

4. Power calculations for 2 normal samples

These calculations are based upon the same type of t test approach as just described for 1 normal sample, except that the pooled variance s_p^2 should be input as the estimate for the common variance σ^2, i.e.,

$$s_p^2 = \frac{\sum_{i=1}^{n_x}(x_i - \bar{x})^2 + \sum_{j=1}^{n_y}(y_j - \bar{y})^2}{n_x + n_y - 2}$$

where X has sample size n_x and Y has sample size n_y. The following options are available:

- To calculate the sample size necessary to estimate the difference between the two population means within a half width h

$$n = \frac{2s_p^2 t^2_{\alpha/2,2n-2}}{h^2};$$

- To calculate the sample size necessary to detect an absolute difference δ between population means

$$n = \frac{2s_p^2}{\delta^2}(t_{\alpha/2,2n-2} + t_{\beta,2n-2})^2; \text{ or}$$

- To estimate the power

$$t_{\beta,2n-2} = \frac{\delta}{\sqrt{2s_p^2/n}} - t_{\alpha/2,2n-2}.$$

The t test has maximum power when $n_x = n_y$ but, if the two sample sizes are unequal, calculations based on the the harmonic mean n_h should be used, i.e.,

$$n_h = \frac{2n_x n_y}{n_x + n_y},$$
$$\text{so that } n_y = \frac{n_h n_x}{2n_x - n_h}.$$

Example 4

In order to perform calculations it is necessary to assume that both samples are from normal distributions with the same variance and then input those of the following parameters as required.

- The significance level α
- An accurate estimate for the common variance s^2
- Choice of a 2-tail test or 1-tail test
- The half width h to determine a 95% confidence range $2h$ for the difference between the sample means
- The minimum absolute difference δ between the sample means that can be detected
- The power $100(1-\beta)\%$
- The sample size n

The following table was created using the analysis of power and sample size option from the statistical calculations procedure available from the [Statistics] menu on the SimFIT main menu, or by using the [A/Z] option to open program **simstat**.

Example 4. Power analysis for the t test

$n(h)$	h = 1	α = 0.05		s^2 = 1	n = 9
$n(\delta)$	δ = 1	α = 0.05	β = 0.2	s^2 = 1	n = 17
$\delta(n)$	n = 17	α = 0.05	β = 0.2	s^2 = 1	δ = 0.9912
$\beta(n)$	n = 17	δ = 1	α = 0.05	s^2 = 1	β = 0.1931
$n(h)$	h = 0.5	α = 0.05		s^2 = 0.5193	n = 18
$n(\delta)$	δ = 0.5	α = 0.05	β = 0.1	s^2 = 0.5193	n = 45
$\beta(n)$	n = 15	δ = 1	α = 0.05	s^2 = 0.5193	β = 0.0454

The last three entries in the above table would be typical. They are for two samples of size $n = 15$ with pooled variance $s^2 = 0.5193$, and the results would be interpreted as follows.

- $n(h)$ shows that a sample size of $n = 18$ would be required to have a 95% confidence interval for the difference between the true means no larger than 1, that is with $h = 0.5$.
- $n(\delta)$ illustrates that a sample size of $n = 45$ is necessary in order for a 90% chance of detecting a difference δ between the true means as small as 0.5.
- $\beta(n)$ demonstrate that the power for detecting a difference of $\delta = 1$ between the true means has a power of 95.46%.

5. Power calculations for k normal samples

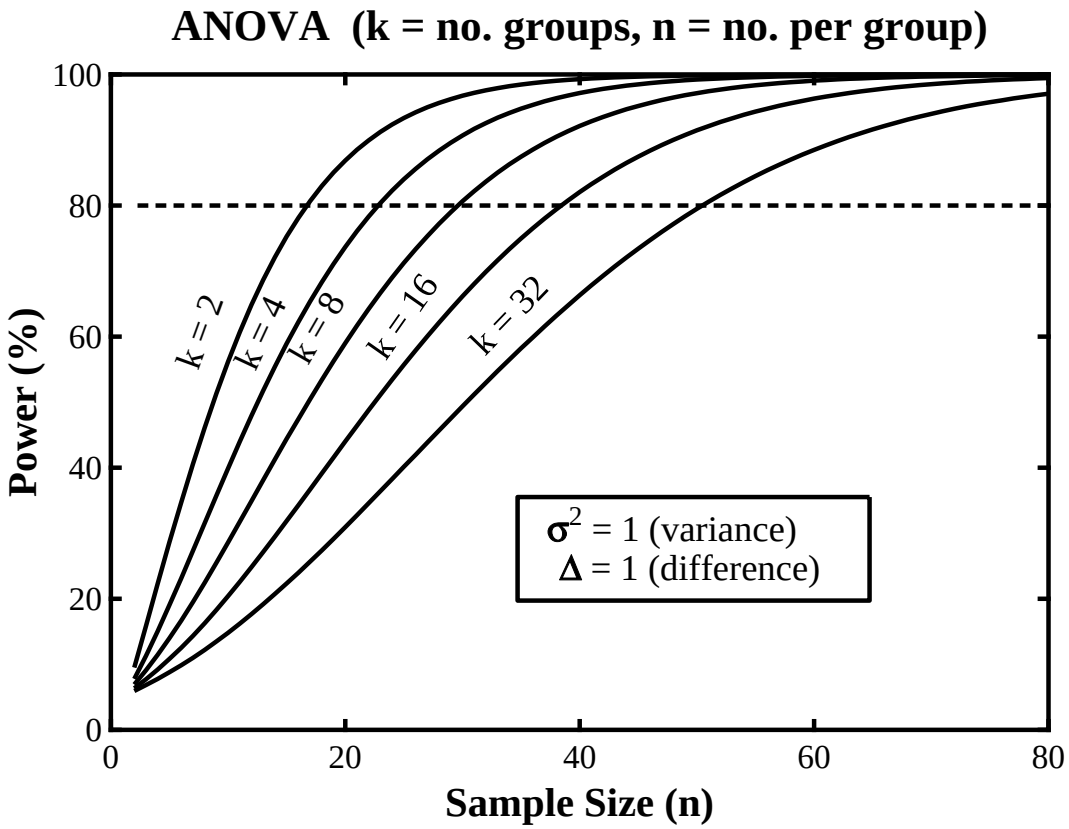
The calculations are based on the 1-way analysis of variance technique, i.e. ANOVA. Note that the SimFIT power as a function of sample size procedure also allows you to plot power as a function of sample size, which is particularly useful with ANOVA designs where the number of columns k can be of interest, in addition

to the number per sample n. The power calculation involves the F and non-central F distributions and you calculate the required n values by using graphical estimation to obtain starting estimates for the iteration. If you choose a n value that is sufficient to make the power as a function on n plot cross the critical power, the program then calculates the power for sample sizes adjacent to the intersection, which is of use when studying k and n for ANOVA.

Example 5

All the power procedures available in SIMFIT provide the ability to plot power as a function of sample size but this particularly useful with ANOVA, as will now be explained.

It is important in the design of experiments to be able to estimate the sample size needed to detect a significant effect. For such calculations you must specify all the parameters of interest except one, then calculate the unknown parameter using numerical techniques. For example, the problem of deciding whether one or more samples differ significantly is a problem in the Analysis of Variance, as long as the samples are all normally distributed and with the same variance. You specify the known variance, σ^2, the minimum detectable difference between means, Δ, the number of groups, k, the significance level, α, and the sample size per group, n. Then, using nonlinear equations involving the F and noncentral F distributions, the power, $100(1-\beta)$ can be calculated. It can be very confusing trying to understand the relationship between all of these parameters so, in order to obtain an impression of how these factors alter the power, a graphical technique is very useful, as in this figure.



simstat was used to create this graph. The variance, significance level, minimum detectable difference and number of groups were fixed, then power was plotted as a function of sample size. The ASCII text coordinate files from several such plots were collected together into a library file to compose the joint plot using **simplot**. Note that, if a power plot reaches the current power level of interest, the critical power level is plotted (80% in the above plot) and the n values either side of the intersection point are displayed.

6. Power calculations for 1 and 2 variances

The calculations depend on the fact that, for a sample of size n from a normal distribution with true variance σ_0^2, the function χ^2 defined as

$$\chi^2 = \frac{(n-1)s^2}{\sigma_0^2}$$

is distributed as a chi-square variable with $n-1$ degrees of freedom. Also, given variance estimates s_x^2 and s_y^2 obtained with sample sizes n_x and n_y from the same normal distribution, the variance ratio F defined as

$$F = \max\left(\frac{s_x^2}{s_y^2}, \frac{s_y^2}{s_x^2}\right)$$

is distributed as an F variable with either n_x, n_y or n_y, n_x degrees of freedom. If possible n_x should equal n_y, of course. The 1-tailed options available are:

- $H_0 : \sigma^2 \leq \sigma_0^2$ against $H_1 : \sigma^2 > \sigma_0^2$

$$1 - \beta = P(\chi^2 \geq \chi^2_{\alpha,n-1}\sigma_0^2/s^2);$$

- $H_0 : \sigma^2 \geq \sigma_0^2$ against $H_1 : \sigma^2 < \sigma_0^2$

$$1 - \beta = P(\chi^2 \leq \chi^2_{1-\alpha,n-1}\sigma_0^2/s^2); \text{ or}$$

- Rearranging the samples, if necessary, so that $s_x^2 > s_y^2$ then $H_0 : \sigma_x^2 = \sigma_y^2$ against $H_1 : \sigma_x^2 \neq \sigma_y^2$

$$Z_\beta = \sqrt{\frac{2m(n_y-2)}{m+1}}\log\left(\frac{s_x^2}{s_y^2}\right) - Z_\alpha$$

$$\text{where } m = \frac{n_x - 1}{n_y - 1}.$$

Example 6a

This example shows the results from performing a one-tail test on a variance estimate s_1^2 of 2.6898 obtained with a sample size of 8 compared to a theoretical value of s_0^2 of 1.5. A confidence interval for the sample variance is calculated as well as the actual power, followed by the sample size needed for 90% power.

Example 6a: Test for $H_0 : \sigma^2 \leq 1.5, H_1 : \sigma^2 > 1.5$	
Sample variance s_1^2	2.6898
Sample size used to calculate s_1^2	8
95% confidence interval for s_1^2	1.1758, 11.142
Test statistic (C)	12.552
$P(\chi^2 \geq C)$	0.0838
Power of this test	34.6486% (for α = 0.05)
Minimal sample size	51 (for α = 0.05, β = 0.1)

Example 6b

This example explores the power as a function of sample size for a two-tail variance ratio test using samples with variances $s_1^2 = 21.87 (n_1 = 11)$ and $s_2^2 = 15.36 (n_2 = 8)$, i.e. for the null hypothesis $H_0 : \sigma_1^2 = \sigma_2^2$. Details for two-tail tests are also given for the additional hypothetical cases with the same variances but estimated with sample sizes $n_1 = n_2 = 60$ and $n_1 = 20, n_2 = 30$.

Example 6b. Two-tailed variance ratio (F) test	
Sample variance s_1^2	21.87
Sample variance s_2^2	15.36
Sample size n_1	11
Sample size n_2	8
Variance ratio (VR)	1.4238
$P(F \geq VR)$	0.3284
Power of this actual test	15.359% (for α = 0.05)
Minimal sample size	86 (for α = 0.05, β = 0.1)
Hypothetical sample size n_1	60
Hypothetical sample size n_2	60
Variance ratio (VR)	1.4238
$P(F \geq VR)$	0.0889
Power of this hypothetical test	76.763% (for α = 0.05)
Hypothetical sample size n_1	20
Hypothetical sample size n_2	30
Variance ratio (VR)	1.4238
$P(F \geq VR)$	0.1908
Power of this hypothetical test	38.348% (for α = 0.05)

7. Power calculations for 1 and 2 correlations

The correlation coefficient r calculated from a normally distributed sample of size n has a standard error

$$s_r = \sqrt{\frac{1-r^2}{n-2}}$$

and is an estimator of the population correlation ρ. A test for zero correlation, i.e., $H_0 : \rho = 0$, can be based on the statistics

$$t = \frac{r}{s_r},$$
$$\text{or } F = \frac{1+|r|}{1-|r|},$$

where t has a t distribution with $n-2$ degrees of freedom, and F has an F distribution with $n-2$ and $n-2$ degrees of freedom. The Fisher z transform and standard error s_z, defined as

$$z = \tanh^{-1} r,$$
$$= \frac{1}{2} \log\left(\frac{1+r}{1-r}\right),$$
$$s_z = \sqrt{\frac{1}{n-3}},$$

are also used to test $H_0 : \rho = \rho_0$, by calculating the unit normal deviate

$$Z = \frac{z - \zeta_0}{s_z}$$

where $\zeta_0 = \tanh^{-1} \rho_0$. The power is calculated using the critical value

$$r_c = \sqrt{\frac{t^2_{\alpha/2,n-2}}{t^2_{\alpha/2,n-2} + n - 2}}$$

which leads to the transform $z_c = \tanh^{-1} r_c$ and

$$Z_\beta = (z - z_c)\sqrt{n-3}$$

then the sample size required to reject $H_0 : \rho = 0$, when actually ρ is nonzero, can be calculated using

$$n = \left(\frac{Z_\beta + Z_{\alpha/2}}{\zeta_0}\right)^2 + 3.$$

For two samples, X of size n_x and Y of size n_y, where it is desired to test $H_0 : \rho_x = \rho_y$, the appropriate Z statistic is

$$Z = \frac{z_x - z_y}{s_{xy}}$$

$$\text{where } s_{xy} = \sqrt{\frac{1}{n_x - 3} + \frac{1}{n_y - 3}}$$

and the power and sample size are calculated from

$$Z_\beta = \frac{|z_x - z_y|}{s_{xy}} - Z_{\alpha/2},$$

$$\text{and } n = 2\left(\frac{Z_{\alpha/2} + Z_\beta}{z_x - z_y}\right)^2 + 3.$$

Example 7a

A sample correlation coefficient $R_1 = 0.87$ was calculated from a sample of size 12 and a population correlation $R_0 = 0.5$ was assumed, resulting in the following calculations.

Example 7a. $H_0 : \rho = 0, H_1 : \lvert\rho\rvert > 0$(or $> R_0$)	
Current α	0.05
Current β	0.01
Correlation coefficient R_1	0.87
Correlation coefficient R_0	0.5
Sample size	12
95% confidence limits	0.5893, 0.9633
Two tailed t-test, p	0.0002
Power of this test	97.882%
N for $H_1 : \lvert\rho\rvert > 0$	14 (R_1 given)
N for $H_1 : \lvert\rho\rvert > R_0$	64 (R_1 arbitrary)

These results indicate that a 95% confidence interval for R_1 is (0.5893, 0.9633) and that the power for this test was 97.882%. The sample size would have to be increased to 14 to obtain 99% power with the current sample correlation coefficient R_1, while a sample size of 64 would be required to ensure 99% power in a test for $|\rho| \geq 0.5$ before a sample is taken in order to calculate R_1.

Example 7b

Two samples were analyzed to investigate equality of correlation coefficients and estimate the power and sample size needed for specified power for hypothetical samples with the same correlation coefficients but with equal hypothetical sample sizes.

Example 7b. $H_0 : \rho_1 = \rho_2, H_1 : \lvert\rho_1 - \rho_2\rvert > 0$	
Current α	0.05
Current β	0.2
R_1, N_1, Z_1	0.78, 98, 1.045
R_2, N_2, Z_2	0.84, 95, 1.221
Two-tailed Z-test, p	0.2294
Power of this test	22.42%
For a Z-diff $\lvert z_1 - z_2\rvert$	0.1758
sample size N required	511

These results show that, for samples with R_1 estimated from a sample of size 98, and R_2 estimated from a sample of size 95 the power was 22.42%. For a hypothetical sample with the same correlation coefficients but estimated from samples with size N then, for 80% power, N would have to be at least 511.

8. Power calculations for a chi-square test

The calculations are based on the chi-square test for either a contingency table, or sets of observed and expected frequencies. However, irrespective of whether the test is to be performed on a contingency table or on samples of observed and expected frequencies, the null hypotheses can be stated in terms of k probabilities as
H_0 : the probabilities are $p_0(i)$, for $i = 1, 2, \ldots, k$,
H_1 : the probabilities are $p_1(i)$, for $i = 1, 2, \ldots, k$.
The power can then be estimated using the non-central chi-square distribution with non-centrality parameter λ and ν degrees of freedom given by

$$\lambda = nQ,$$
$$\text{where } Q = \sum_{i=1}^{k} \frac{(p_0(i) - p_1(i))^2}{p_0(i)},$$
$$n = \text{ total sample size,}$$
$$\text{and } \nu = k - 1 - \text{ no. of parameters estimated.}$$

You can either input the Q values directly, or read in vectors of observed and expected frequencies. If you do input frequencies $f_i \geq 0$ they will be transformed internally into probabilities, i.e., the frequencies only have to be positive integers as they are normalized to sum unity using

$$p_i = f_i \Big/ \sum_{i=1}^{k} f_i.$$

In the case of contingency table data with r rows and c columns, the probabilities are calculated from the marginals $p_{ij} = p(i)p(j)$ in the usual way, so you must input $k = rc$, and the number of parameters estimated as $r + c - 2$, so that $\nu = (r - 1)(c - 1)$.

Example 8

We demonstrate this procedure using the example of a weighted die discussed by William C Guenther in The American Statistician 31 (1977) pp 83–85 using table look-up. Here the sum of squares was $Q = 0.05$ with a sample size of $n = 120$ so that the degrees of freedom were $\lambda = 6$ and β was calculated to be approximately 0.5671. Further calculations showed that a sample size of $n > 330$ was required to achieve $\beta = 0.1$, while the results calculated by SIMFIT were as follows.

Example 8. Power analysis for a chi-square test $H_0 : p_0(i) = p_1(i)$					
α = 0.05	β = 0.5761	Q = 0.05	λ = 6	χ^2 = 12.59	N = 128
α = 0.05	β = 0.1000	Q = 0.05	λ = 6	χ^2 = 12.59	N = 352

Theory

Experiments often generate random samples from a population so that parameters estimated from the samples can be use to test hypotheses about the population parameters. So it is natural to investigate the relationship between sample size and the absolute precision of the estimates, given the expectation $E(X)$ and variance $\sigma^2(X)$ of the random variable. For a single observation, i.e., $n = 1$, the Chebyshev inequality

$$P\left(|X - E(X)| < \epsilon\right) \geq 1 - \frac{\sigma^2(X)}{\epsilon^2}$$

with $\epsilon > 0$, indicates that, for an unspecified distribution,

$$P\left(|X - E(X)| < 4.5\sigma(X)\right) \geq 0.95,$$
$$\text{and } P\left(|X - E(X)| < 10\sigma(X)\right) \geq 0.99,$$

but, for an assumed normal distribution,

$$P\left(|X - E(X)| < 1.96\sigma(X)\right) \geq 0.95,$$
$$\text{and } P\left(|X - E(X)| < 2.58\sigma(X)\right) \geq 0.99.$$

However, provided that $E(X) \neq 0$, it is more useful to formulate the Chebyshev inequality in terms of the relative precision, that is, for $\delta > 0$

$$P\left(\left|\frac{X - E(X)}{E(X)}\right| < \delta\right) \geq 1 - \frac{1}{\delta^2}\frac{\sigma^2(X)}{E^2(X)}.$$

Now, for an unspecified distribution,

$$P\left(\left|\frac{X - E(X)}{E(X)}\right| < 4.5\frac{\sigma(X)}{|E(X)|}\right) \geq 0.95,$$
$$\text{and } P\left(\left|\frac{X - E(X)}{E(X)}\right| < 10\frac{\sigma(X)}{|E(X)|}\right) \geq 0.99,$$

but, for an assumed normal distribution,

$$P\left(\left|\frac{X - E(X)}{E(X)}\right| < 1.96\frac{\sigma(X)}{|E(X)|}\right) \geq 0.95,$$
$$\text{and } P\left(\left|\frac{X - E(X)}{E(X)}\right| < 2.58\frac{\sigma(X)}{|E(X)|}\right) \geq 0.99.$$

So, for high precision, the coefficient of variation $cv\%$

$$cv\% = 100\frac{\sigma(X)}{|E(X)|}$$

must be as small as possible, while the signal-to-noise ratio $SN(X)$

$$SN(X) = \frac{|E(X)|}{\sigma(X)}$$

must be as large as possible. For instance, for the single measurement to be within 10% of the mean 95% of the time requires $SN \geq 45$ for an arbitrary distribution, or $SN \geq 20$ for a normal distribution. A particularly

valuable application of these results concerns the way that the signal-to-noise ratio of sample means depends on the sample size n. From

$$\begin{aligned}\bar{X} &= \frac{1}{n}\sum_{i=1}^{n} x_i, \\ Var(\bar{X}) &= \frac{1}{n^2}\sum_{i=1}^{n} Var(X) \\ &= \frac{1}{n}\sigma^2(X),\end{aligned}$$

it follows that, for arbitrary distributions, the signal-to-noise ratio of the sample mean $SN(\bar{X})$ is given by $SN(\bar{X}) = \sqrt{n}SN(X)$, that is

$$SN(\bar{X}) = \sqrt{n}\frac{E(X)}{\sigma(X)}.$$

This result, known as the law of $\sqrt{n}$, implies that the signal-to-noise ratio of the sample mean as an estimate of the population mean increases as $\sqrt{n}$, so that the the relative error in estimating the mean decreases like $1/\sqrt{n}$.

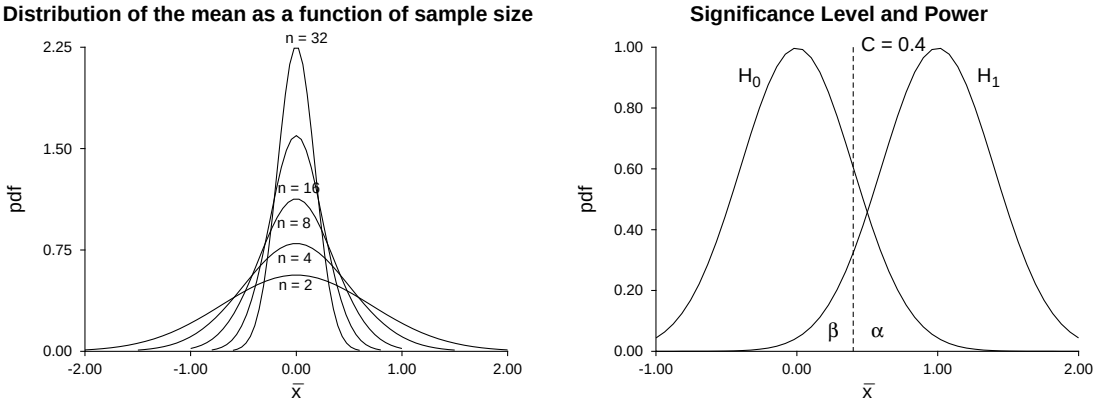
If $f(x)$ is the density function for a random variable X, then the null and alternative hypotheses can sometimes be expressed as

$$\begin{aligned}H_0 &: f(x) = f_0(x) \\ H_1 &: f(x) = f_1(x)\end{aligned}$$

while the error sizes, given a critical region C, are

$$\begin{aligned}\alpha &= P_{H_0}(\text{reject } H_0) \text{ (i.e., the Type I error)} \\ &= \int_C f_0(x)\,dx \\ \beta &= P_{H_1}(\text{accept } H_0) \text{ (i.e., the Type II error)} \\ &= 1 - \int_C f_1(x)\,dx.\end{aligned}$$

Usually α is referred to as the significance level, β is the operating characteristic, while $1-\beta$ is the power, frequently expressed as a percentage, i.e., $100(1-\beta)\%$, and these will both alter as the critical region is changed.



This figure illustrates the concepts of signal-to-noise ratio, significance level, and power. The family of curves on the left are the probability density functions for the distribution of the sample mean $\bar{x}$ from a normal

distribution with mean $\mu = 0$ and variance $\sigma^2 = 1$. The curves on the right illustrate the significance level α, and operating characteristic β for the null and alternative hypotheses

$$H_0 : \mu = 0, \sigma^2 = 4$$
$$H_1 : \mu = 1, \sigma^2 = 4$$

for a test using the sample mean from a sample of size $n = 25$ from a normal distribution, with a critical point $C = 0.4$. The significance level is the area under the curve for H_0 to the right of the critical point, while the operating characteristic is the area under the curve for H_1 to the left of the critical point. Clearly, increasing the critical value C will decrease α and increase β, while increasing the sample size n will decrease both α and β.

Often it is wished to predict power as a function of sample size, which can sometimes be done if distributions $f_0(x)$ and $f_1(x)$ are assumed, necessary parameters are provided, the critical level is specified, and the test procedure is defined. Essentially, given an implicit expression in k unknowns, this option solves for one given the other $k - 1$, using iterative techniques. For instance, you might set α and β, then calculate the sample size n required, or you could input α and n and estimate the power. Note that 1-tail tests can sometimes be selected instead of 2-tail tests (e.g., by replacing $Z_{\alpha/2}$ by Z_α in the appropriate formula) and also be very careful to make the correct choice for supplying proportions, half-widths, absolute differences, theoretical parameters or sample estimates, etc.

A word of warning is required on the subject of calculating n required for a given power. The values of n will usually prove to be very large, probably much larger than can be used. So, for pilot studies and typical probing investigations, the sample sizes should be chosen according to cost, time, availability of materials, past experience, and so on. Sample size calculations are only called for when Type II errors may have serious consequences, as in clinical trials, so that large samples are justified.

Of course, the temptation to choose 1-tail instead of 2-tail tests, or use variance estimates that are too small, in order to decrease the n values should be avoided, but it happens.

5.2 Parameter confidence limits

Given a sample from a known distribution it is generally easy to estimate the population parameters using the sample estimates, but it is not always so easy to determine the confidence limits, such as a 95% confidence interval. From the main SimFIT menu you can select [Statistics] then the option to perform statistical calculations. Here you can choose the distribution required and the significance level of interest, then input the estimates and sample sizes required. Note that the well-known case of a normal distribution leads many to believe that a confidence interval is always symmetrical about a parameter estimate, but many confidence intervals will be asymmetric for those distributions (Poisson, binomial) where exact methods are used, not calculations based on the normal approximation.

Confidence limits for a Poisson parameter

Given a sample $x_1, x_2, \ldots, x_n$ of n non-negative integers from a Poisson distribution with parameter λ, the parameter estimate $\hat{\lambda}$, i.e., the sample mean, and confidence limits λ_1, λ_2 are calculated as follows

$$K = \sum_{i=1}^{n} x_i,$$

$$\hat{\lambda} = K/n,$$

$$\lambda_1 = \frac{1}{2n}\chi^2_{2K,\alpha/2},$$

$$\lambda_2 = \frac{1}{2n}\chi^2_{2K+2,1-\alpha/2},$$

$$\text{so that } \exp(-n\lambda_1)\sum_{x=K}^{\infty}\frac{(n\lambda_1)^x}{x!} = \frac{\alpha}{2},$$

$$\exp(-n\lambda_2)\sum_{x=0}^{K}\frac{(n\lambda_2)^x}{x!} = \frac{\alpha}{2},$$

$$\text{and } P(\lambda_1 \le \lambda \le \lambda_2) = 1-\alpha,$$

using the lower tail critical points of the chi-square distribution. The following very approximate rule-of-thumb can be used to get a quick idea of the range of a Poisson mean λ given a single count x and exploiting the fact that the Poisson variance equals the mean

$$P(x - 2\sqrt{x} \le \lambda \le x + 2\sqrt{x}) \approx 0.95.$$

Example

The number of weed seeds in 98 samples of meadow grass yielded these counts with a mean of 3.0204.

Number	0	1	2	3	4	5	6	7	8	9	10
Frequency	3	17	26	16	18	9	3	5	0	1	0

The 95% and 99% noncentral confidence intervals from the estimate were found to be as follows.

Sample size	Mean	Level	Interval
98	3.0204	95%	$2.68608 \le \lambda \le 3.38483$
98	3.0204	99%	$2.58737 \le \lambda \le 3.50272$

Confidence limits for a binomial parameter

For k successes in n trials, the binomial parameter estimate $\hat{p}$ is k/n and three methods are used to calculate confidence limits p_1 and p_2 so that

$$\sum_{x=k}^{n} \binom{n}{x} p_1^x (1-p_1)^{n-x} = \alpha/2,$$
$$\text{and } \sum_{x=0}^{k} \binom{n}{x} p_2^x (1-p_2)^{n-x} = \alpha/2.$$

- If $\max(k, n-k) < 10^6$, the lower tail probabilities of the beta distribution are used as follows

$$p_1 = \beta_{k,n-k+1,\alpha/2},$$
$$\text{and } p_2 = \beta_{k+1,n-k,1-\alpha/2}.$$

- If $\max(k, n-k) \geq 10^6$ and $\min(k, n-k) \leq 1000$, the Poisson approximation with $\lambda = np$ and the chi-square distribution are used, leading to

$$p_1 = \frac{1}{2n}\chi^2_{2k,\alpha/2},$$
$$\text{and } p_2 = \frac{1}{2n}\chi^2_{2k+2,1-\alpha/2}.$$

- If $\max(k, n-k) > 10^6$ and $\min(k, n-k) > 1000$, the normal approximation with mean np and variance $np(1-p)$ is used, along with the lower tail normal deviates $Z_{1-\alpha/2}$ and $Z_{\alpha/2}$, to obtain approximate confidence limits by solving

$$\frac{k - np_1}{\sqrt{np_1(1-p_1)}} = Z_{1-\alpha/2},$$
$$\text{and } \frac{k - np_2}{\sqrt{np_2(1-p_2)}} = Z_{\alpha/2}.$$

The following very approximate rule-of-thumb can be used to get a quick idea of the range of a binomial mean np given x and exploiting the fact that the binomial variance variance equals $np(1-p)$

$$P(x - 2\sqrt{x} \leq np \leq x + 2\sqrt{x}) \approx 0.95.$$

Example

In a study the number of deaths among pensioners in a six year period were as follows.

	Sample size	Deaths	Probability	95% confidence interval
Non-smokers	1067	117	0.109653	$0.091533 \leq p \leq 0.129957$
Smokers	402	54	0.134328	$0.102548 \leq p \leq 0.171609$

Again, note the noncentral 95% confidence intervals for the probability estimates $\hat{p}$ as summarized below.

Deaths/Subjects	$\hat{p}$	95% Confidence Interval	Group
117/1067	0.1097	0.1097 - 0.0182, 0.1097 + 0.0203	Non-smokers
54/402	0.1343	0.1343 - 0.0318, 0.1343 + 0.0373	Smokers

Confidence limits for a normal mean and variance

If the sample mean is $\bar{x}$, and the sample variance is s^2, with a sample of size n from a normal distribution having mean μ and variance σ^2, the confidence limits are defined by

$$P(\bar{x} - t_{\alpha/2,n-1}\, s/\sqrt{n} \le \mu \le \bar{x} + t_{\alpha/2,n-1}\, s/\sqrt{n}) = 1 - \alpha,$$
$$\text{and } P((n-1)s^2/\chi^2_{\alpha/2,n-1} \le \sigma^2 \le (n-1)s^2/\chi_{1-\alpha/2,n-1}) = 1 - \alpha$$

where the upper tail probabilities of the t and chi-square distribution are used.

Example

The body temperature of 25 intertidal crabs was recorded in ℃ as follows: 24.3, 25.8, 24.6, 26.1, 22.9, 25.1, 27.3, 24.0, 24.5, 23.9, 26.2, 24.3, 24.6, 23.3, 25.5, 28.1, 24.8, 23.5, 26.3, 25.4, 25.5, 23.9, 27.0, 24.8, 22.9, 25.4. The sample mean, variance and standard deviation were $\bar{x} = 25.03$, $s^2 = 1.8$, and $s = 1.3416408$ leading to the following central confidence intervals for the mean and unsymmetrical confidence limits for the variance.

Sample size	Level	Parameter	Estimate	Interval
25	95%	Mean	25.03	$24.4762 \le \mu \le 25.5838$
25	99%	Mean	25.03	$24.2795 \le \mu \le 25.7805$
25	95%	Variance	1.8	$1.09745 \le \sigma^2 \le 3.48355$
25	99%	Variance	1.8	$0.948231 \le \sigma^2 \le 4.36971$

Confidence limits for a correlation coefficient

If a Pearson product-moment correlation coefficient r is calculated from two samples of size n that are jointly distributed as a bivariate normal distribution, the confidence limits for the population parameter ρ are given by

$$P\left(\frac{r - r_c}{1 - r r_c} \le \rho \le \frac{r + r_c}{1 + r r_c}\right) = 1 - \alpha,$$
$$\text{where } r_c = \sqrt{\frac{t^2_{\alpha/2,n-2}}{t^2_{\alpha/2,n-2} + n - 2}}.$$

Example

The wing and tail lengths in cm for 12 birds were as in this next table.

Wing	10.4	10.8	11.1	10.2	10.3	10.2	10.7	10.5	10.8	11.2	10.6	11.4
Tail	7.4	7.6	7.9	7.2	7.4	7.1	7.4	7.2	7.8	7.7	7.8	8.3

This gives a correlation coefficient of $r = 0.87$ with a sample size of $n = 12$, leading to the nonsymmetrical 95% confidence interval.

$$0.589337 \le \rho \le 0.963279$$

Confidence limits for trinomial parameters

If, in a trinomial distribution, the probability of category i is p_i for $i = 1, 2, 3$, then the probability P of observing n_i in category i in a sample of size $N = n_1 + n_2 + n_3$ from a homogeneous population is given by

$$P = \frac{N!}{n_1! n_2! n_3!} p_1^{n_1} p_2^{n_2} p_3^{n_3}$$

and the maximum likelihood estimates, of which only two are independent, are

$$\begin{aligned}\hat{p_1} &= n_1/N,\\ \hat{p_2} &= n_2/N,\\ \text{and } \hat{p_3} &= 1-\hat{p_1}-\hat{p_2}.\end{aligned}$$

The bivariate estimator is approximately normally distributed, when N is large, so that

$$\begin{bmatrix}\hat{p_1}\\ \hat{p_2}\end{bmatrix} \sim MN_2\left(\begin{bmatrix}p_1\\ p_2\end{bmatrix}, \begin{bmatrix}p_1(1-p_1)/N & -p_1p_2/N\\ -p_1p_2/N & p_2(1-p_2)/N\end{bmatrix}\right)$$

where MN_2 signifies the bivariate normal distribution. Consequently

$$((\hat{p_1}-p_1),(\hat{p_2}-p_2))\begin{bmatrix}p_1(1-p_1)/N & -p_1p_2/N\\ -p_1p_2/N & p_2(1-p_2)/N\end{bmatrix}^{-1}\begin{pmatrix}\hat{p_1}-p_1\\ \hat{p_2}-p_2\end{pmatrix} \sim \chi^2_2$$

and hence, with probability 95%,

$$\frac{(\hat{p_1}-p_1)^2}{p_1(1-p_1)}+\frac{(\hat{p_2}-p_2)^2}{p_2(1-p_2)}+\frac{2(\hat{p_1}-p_1)(\hat{p_2}-p_2)}{(1-p_1)(1-p_2)} \le \frac{(1-p_1-p_2)}{N(1-p_1)(1-p_2)}\chi^2_{2;0.05}.$$

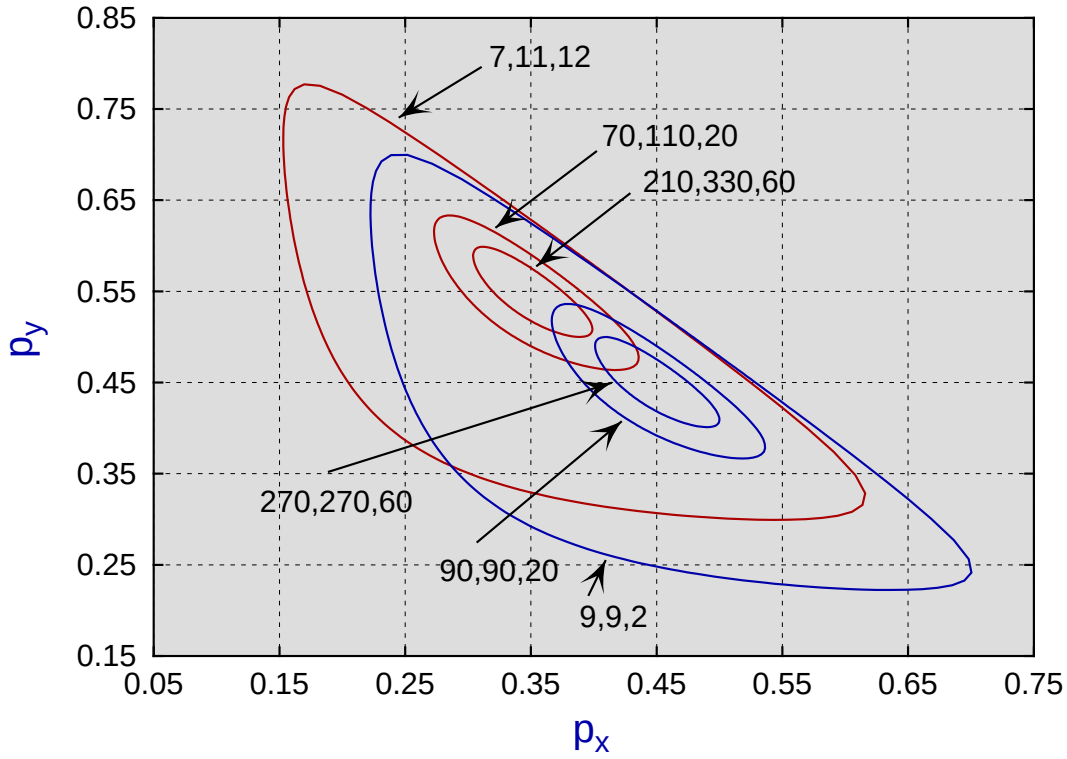
Such inequalities define regions in the (p_1, p_2) parameter space which can be examined for statistically significant differences between $p_{i(j)}$ in samples from populations subjected to treatment j. Where regions are clearly disjoint, parameters have been significantly affected by the treatments, as illustrated next.

Plotting trinomial parameter joint confidence regions

A useful rule of thumb to see if parameter estimates differ significantly is to check their approximate central 95% confidence regions. If the regions are disjoint it indicates that the parameters differ significantly and, in fact, parameters can differ significantly even with limited overlap. If two or more parameters are estimated, it is valuable to inspect the joint confidence regions defined by the estimated covariance matrix and appropriate chi-square critical value. Consider, for example, this figure generated by the contour plotting function of **binomial**. Data triples x, y, z can be any partitions, such as number of male, female or dead hatchlings from a batch of eggs where it is hoped to determine a shift from equi-probable sexes. The contours are defined by

$$((\hat{p_x}-p_x),(\hat{p_y}-p_y))\begin{bmatrix}p_x(1-p_x)/N & -p_xp_y/N\\ -p_xp_y/N & p_y(1-p_y)/N\end{bmatrix}^{-1}\begin{pmatrix}\hat{p_x}-p_x\\ \hat{p_y}-p_y\end{pmatrix} = \chi^2_{2:0.05}$$

where $N = x + y + z$, $\hat{p_x} = x/N$ and $\hat{p_y} = y/N$ as discussed in connection with the trinomial distribution. When $N = 20$ the triples 9,9,2 and 7,11,2 cannot be distinguished, but when $N = 200$ the orbits are becoming elliptical and converging to asymptotic values. By the time $N = 600$ the triples 210,330,60 and 270,270,60 can be seen to differ significantly.


Trinomial Parameter 95% Confidence Contours
0.85
0.75
0.65
0.55
0.45
0.35
0.25
0.15
0.05
0.15
0.25
0.35
0.45
0.55
0.65
0.75
p_y
p_x
7,11,12
70,110,20
210,330,60
270,270,60
90,90,20
9,9,2

5.3 Robust analysis of 1 sample

It is obvious that outliers in a sample lead to biased parameters estimates. In some instances an experimenter is able to examine the data and make a decision to eliminate certain observations, usually extremely low or high values, that indicate a systematic source of variation beyond the usual spread of observational errors. Alternatively, to avoid subjective doctoring of data, a robust method can be used which generally involves discarding extreme values and using more appropriate numerical methods that do not assume that the sample is normally distributed.

As an example, choose statistics from the main SIMFIT menu, navigate to [Data exploration] and open the option for [Robust analysis of one sample]. The results from examining the test file `robust.tf1` after trimming 10% off the extreme values are shown below, followed by the results from handling the full data set without any trimming in the exhaustive analysis procedure.

Robust analysis	
Data: 50 N(0,1) random numbers with 5 outliers	
Total sample size	50
Median value	0.2019
Median absolute deviation	1.0311
Robust standard deviation	1.5288
Trimmed mean (TM)	0.2227
Variance estimate for TM	0.0192
Winsorized mean (WM)	0.2326
Variance estimate for WM	0.0192
Number of discarded values	10
Number of included values	40
Percentage of sample used	80% (for TM and WM)
Hodges-Lehmann estimate (HL)	0.2586

Exhaustive analysis	
Minimum, Maximum values	-2.208, 7.000
Lower and Upper Hinges	-0.829, 1.307
Coefficient of skewness	1.690
Coefficient of kurtosis	3.566
Median value	0.202
Sample mean	0.512
Sample standard deviation	1.853: CV% = 361.736%
Standard error of the mean	0.262
Upper 2.5% t-value	2.010
Lower 95% confidence limit for mean	-0.014
Upper 95% confidence limit for mean	1.039
Variance of the sample	3.435
Lower 95% confidence limit for variance	2.397
Upper 95% confidence limit for variance	5.335
Shapiro-Wilks W statistic	0.851
Significance level for W	0.000 Reject normality at 1% sig.level

Clearly the exhaustive analysis indicates that the presence of outliers has created a sample that is not normally distributed and the results from robust analysis yield better estimates for the population mean and variance which, before adding outliers, were $\mu = 0, \sigma^2 = 1$. An outline of the theory and definitions used in this robust analysis follows.

Theory

If the sample vector is $x_1, x_2, \ldots, x_n$ the following calculations are done.

1. Using the whole sample and the inverse normal function $\Phi^{-1}(.)$, the median M, median absolute deviation D and a robust estimate of the standard deviation S are calculated as

$$\begin{aligned} M &= \text{median}(x_i) \\ D &= \text{median}(|x_i - M|) \\ S &= D/\Phi^{-1}(0.75). \end{aligned}$$

2. The percentage of the sample chosen by users to be eliminated from each of the tails is $100\alpha\%$, then the trimmed mean TM, and Winsorized mean WM, together with variance estimates VT and VW, are calculated as follows, using $k = [\alpha n]$ as the integer part of αn.

$$\begin{aligned} TM &= \frac{1}{n-2k}\sum_{i=k+1}^{n-k} x_i \\ WM &= \frac{1}{n}\left\{\sum_{i=k+1}^{n-k} x_i + kx_{k+1} + kx_{n-k}\right\} \\ VT &= \frac{1}{n^2}\left\{\sum_{i=k+1}^{n-k} (x_i - TM)^2 + k(x_{k+1} - TM)^2 + k(x_{n-k} - TM)^2\right\} \\ VW &= \frac{1}{n^2}\left\{\sum_{i=k+1}^{n-k} (x_i - WM)^2 + k(x_{k+1} - WM)^2 + k(x_{n-k} - WM)^2\right\}. \end{aligned}$$

3. If the assumed sample density is symmetrical, the Hodges-Lehman location estimator HL can be used to estimate the center of symmetry. This is

$$HL = \text{median}\left\{\frac{x_i + x_j}{2}, 1 \le i \le j \le n\right\},$$

and it is calculated along with 95% confidence limit. This would be useful if the sample was a vector of differences between two samples X and Y for a Wilcoxon signed rank test that X is distributed $F(x)$ and Y is distributed $F(x - \theta)$.

5.4 Robust analysis of 2 samples

Sometimes a robust estimate is required for the difference in location (with corresponding confidence limits) for two samples, not necessarily of the same size, but without assuming normality or any other distribution.

From the main SIMFIT menu choose [Statistics], navigate to [Data exploration] and open the option for [Robust analysis of two samples]. The two default test files are `ttest.tf4` and `ttest.tf5` with these values

ttest.tf4	ttest.tf5
134	70
146	118
104	101
119	85
124	107
161	132
107	94
83	
113	
129	
97	
123	

while analysis produces the following results.

Robust analysis of two samples	
X-sample size	12
Y-sample size	7
Difference in location	-18.501
Lower confidence limit	-40.009
Upper confidence limit	2.997
Percentage confidence limit	95.30%
Lower Mann-whitney U-value	19.000
Upper Mann-Whitney U-value	66.000

The procedure is based on the assumption that X of size n_x is distributed as $F(x)$ and Y of size n_y as $F(x-\theta)$, so an estimate $\hat{\theta}$ for the difference in location is calculated as

$$\hat{\theta} = \text{ median}(y_j - x_i, i = 1, 2, \ldots, n_x, j = 1, 2, \ldots, n_y).$$

$100\alpha\%$ confidence limits U_L and U_H are then estimated by inverting the Mann-Whitney U statistic so that

$$\begin{aligned} P(U \le U_L) &\le \alpha/2 \\ P(U \le U_L + 1) &> \alpha/2 \\ P(U \ge U_H) &\le \alpha/2 \\ P(U \ge U_H - 1) &> \alpha/2. \end{aligned}$$

5.5 Shannon-Brillouin-Simpson indices of diversity

It is often required to define functions that estimate or merely summarize the entropy or degree of randomness in the distribution of observations into categories.

Typically there would be n observations in total divided into k categories with nonnegative frequencies f_i, for $i = 1, 2, \ldots, k$, so that $n = \sum_{i=1}^{k} f_i$, as for example with this extreme sample where there is clearly no evidence of differences between the groups

Group	Frequency
1	5
2	5
3	5
4	5

or this equally extreme example where the non-homogeneity is obvious.

Group	Frequency
1	1
2	1
3	1
4	17

To analyze such data choose [Statistics] from the main SIMFIT menu then [Statistical calculations] followed by selecting [Shannon/Brilllouin indices of diversity]. Note that data sets must be supplied as samples of frequencies and these can be as vector files, or columns of frequencies pasted in from the clipboard. However, it is often the case when repetitive analysis of small data sets is required, that it is useful to temporarily switch off the speed-up option that suppresses input from the terminal and simply enter the frequencies manually.

Of course such problems arise constantly in data analysis but especially in ecology where several well known indices of diversity are used, as illustrated in the next table for these two extreme cases.

Data: 5,5,5,5	
Number of groups	4
Total sample size	20
Pielou J' evenness	1.0000 [complement = 0.0000]
Brillouin J evenness	1.0000 [complement = 0.0000]
Shannon H'	0.6021($\log_{10}$) 1.386($\log_e$) 2.000($\log_2$)
Brillouin H	0.5035($\log_{10}$) 1.159($\log_e$) 1.672($\log_2$)
Simpson λ	0.2500 [complement = 0.7500]
Simpson λ'	0.2105 [complement = 0.7895]

Data: 1,1,1,17	
Number of groups	4
Total sample size	20
Pielou J' evenness	0.4238 [complement = 0.5762]
Brillouin J evenness	0.3809 [complement = 0.6191]
Shannon H'	0.2551 ($\log_{10}$) 0.5875($\log_e$) 0.8476($\log_2$)
Brillouin H	0.1918 ($\log_{10}$) 0.4415($\log_e$) 0.6370($\log_2$)
Simpson λ	0.7300 [complement = 0.2700]
Simpson λ'	0.7158 [complement = 0.2842]

Definitions

Given positive integer frequencies $f_i > 0$ in $k > 1$ groups with n observations in total, then proportions $p_i = f_i/n$ can be defined, leading to the Shannon H', Brillouin H, and Simpson λ and λ' indices, and the evennness parameters J and J' defined as follows.

$$\begin{aligned}
\text{Shannon diversity } H' &= -\sum_{i=1}^{k} p_i \log p_i \\
&= [n \log n - \sum_{i=1}^{k} f_i \log f_i]/n \\
\text{Pielou evenness } J' &= H'/\log k \\
\text{Brilloin diversity } H &= [\log n! - \log \prod_{i=1}^{k} f_i!]/n \\
\text{Brilloin evenness } J &= nH/[\log n! - (k-d)\log c! - d\log(c+1)!] \\
\text{Simpson lambda } \lambda &= \sum_{i=1}^{k} p_i^2 \\
\text{Simpson lambda prime } \lambda' &= \sum_{i=1}^{k} f_i(f_i-1)/[n(n-1)]
\end{aligned}$$

where $c = [n/k]$ and $d = n - ck$. Note that H and H' are given using logarithms to bases ten, e, and two, while the forms J and J' have been normalized by dividing by the corresponding maximum diversity and so are independent of the base. The complements $1 - J$, $1 - J'$, $1 - \lambda$, and $1 - \lambda'$ are also tabulated within the square brackets.

In the above tables we see that evenness is maximized when all categories are equally occupied, so that $f_i = 1/k$ and $H' = \log k$, and is minimized when one category dominates.

Of course SimFIT provides numerous techniques to test hypotheses about the distribution of frequencies into groups, e.g. a χ^2 test on observed and expected frequencies.

5.6 Non-central statistical distributions

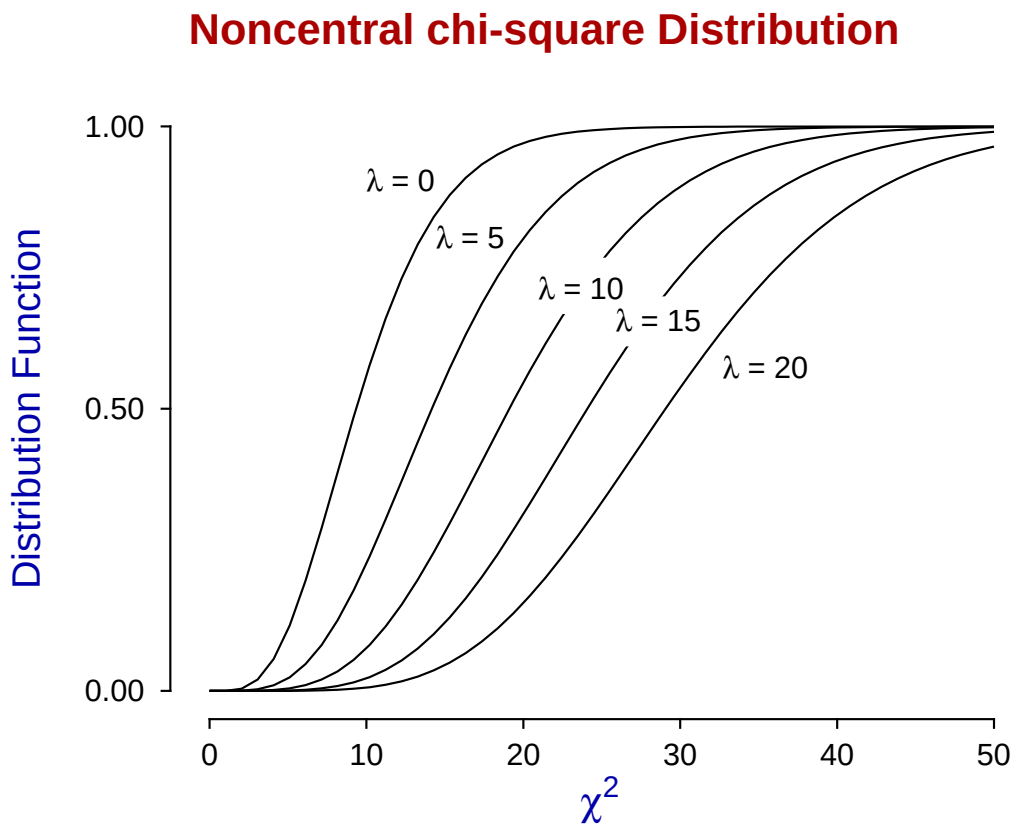
Non-central distributions are frequently used in statistical analysis, especially for studies to estimate the power of hypothesis tests as functions of sample size.

SIMFIT uses many discrete and continuous central and non-central distributions for modelling and hypothesis tests, and provides numerous options as well as dedicated programs such as **binomial**, **chisqd**, **F**, and **normal** to plot or obtain percentage points to replace table look-up. However, you can also obtain values and plots for the following special distributions, given the appropriate arguments.

- non-central β
- non-central χ^2
- non-central F
- non-central t

To obtain percentage points and create plots for non-central distributions choose [Statistics] followed by [Statistical calculations] from the main SIMFIT menu.

For instance, this figure illustrates the chi-square distribution with 10 degrees of freedom for non-centrality parameter λ at values of $\lambda = 0, 5, 10, 15,$ and 20.



5.7 Ligand-binding cooperativity analysis

Cooperative ligand binding models are used in the situation where a protein or receptor has more than one type of binding site and these are linked in such a way as to display deviations from normal hyperbolic binding. If a receptor has $n > 1$ binding sites that differ in binding constants but are independent this can only give rise to apparent negative cooperativity. If the sites are linked in that the binding to one site influences the subsequent binding of further ligands then positive or mixed cooperativity can be exhibited.

Ligand binding theory will be presented under the following headings.

1. Historical introduction
2. Binding polynomials
3. The Hessian of a binding polynomial
4. Definition of cooperativity
5. Zeros of the binding polynomial
6. Statistical interpretation of saturation functions
7. Cooperativity analysis

Historical Introduction

In 1910 Hill [1] proposed that the sigmoid binding curve for oxygen binding to haemoglobin Hb could be analyzed in terms of the binding of n ligands X in one step with no appreciable intermediates, i.e. the mass action description

$$Hb + nX \rightleftharpoons HbX_n.$$

This leads to the Hill equation describing the fractional saturation y as a function of concentration x, and the Hill plot of $\log[y/(1-y)]$ as a function of $\log x$ as follows

$$y = \frac{Kx^n}{1 + Kx^n}$$

$$\log\left(\frac{y}{1-y}\right) = n\log x + \log K.$$

It is now realized that the Hill equation is simply an empirical equation that is at best a poor approximation to any real binding situation since:

1. it is only an appropriate representation for a one-site binding process, i.e. for $n = 1$;
2. when $n < 1$ it has an infinite slope at the origin and cannot model any realistic binding situation;
3. when $n > 1$ it has zero slope at the origin and cannot model any realistic binding situation;
4. when n is not a positive integer it is pure nonsense; and
5. using it to discuss the effect of cooperativity on graphical features such as sigmoidicity in the $y(x)$ curve, or convexity in Lineweaver-Burke or Scatchard space, has resulted in considerable confusion.

Of course, before the days of computers and nonlinear regression, fitting a straight line to a Hill plot to get a non-integer value for the estimated slope was all that could be done, and this non-integer value was correctly taken to mean that this was a result of a cooperative binding model.

Nowadays no one would dream of discussing cooperative binding in terms of the Hill equation or fitting a straight line to a Hill plot but, by a serendipitous coincidence, it turns out that the variable slope of the curve obtained by transforming a saturation curve into Hill space still provides an unambiguous definition of the

sign and magnitude of cooperativity that has got nothing at all to do with the Hill equation. That is because, to use receptor terminology,

$$\frac{y}{1-y} = \frac{[\text{Bound}]}{[\text{Free}]}.$$

Binding polynomials and their Hessians

In 1925 Adair [2] improved the description of binding isotherms by defining binding constants for the individual binding events, and later it came to be appreciated that these have to be normalized by statistical factors in order to discuss the affinity of receptor for ligand in adjacent binding events. In 1967 Wyman [3] rationalized the situation by pointing out that, for a non-aggregating macromolecule with n binding sites and only one ligand x varied, there would be binding polynomial which would act like a partition function in that successive terms of degree i in the polynomial are proportional to the amount of macromolecule with i ligands attached.

So now the binding of ligands to receptors can be defined for all possible cooperative binding schemes in terms of a binding polynomial $p(x)$ in the free ligand activity x, as follows

$$\begin{aligned} p(x) &= 1 + K_1x + K_2x^2 + \cdots + K_nx^n \\ &= 1 + A_1x + A_1A_2x^2 + \cdots + \prod_{i=1}^{n} A_ix^n \\ &= 1 + \binom{n}{1}B_1x + \binom{n}{2}B_1B_2x^2 + \cdots + \binom{n}{n}\prod_{i=1}^{n} B_ix^n, \end{aligned}$$

where the only difference between these alternative expressions concerns the meaning and interpretation of the binding constants. The fractional saturation $y(x)$ is just the scaled derivative of the log of the polynomial with respect to $\log(x)$, and an important auxiliary function is $h(x)$, the scaled Hessian of the binding polynomial and these are defined as follows

$$\begin{aligned} y(x) &= \left(\frac{1}{n}\right)\frac{d\log p(x)}{d\log x} \\ &= \left(\frac{1}{n}\right)\frac{xp'(x)}{p(x)}, \text{ and} \\ h(x) &= npp'' - (n-1)p'^2. \end{aligned}$$

Definition of the Hessian of a binding polynomial

To investigate the algebraic properties of arbitrary polynomials $f(x)$ of degree n it is useful to consider the homogeneous form $U(x, y)$ as in

$$\begin{aligned} f(x) &= p_0 + p_1x + p_2x^2 + \ldots + p_nx^n, \text{ in the equivalent form} \\ U(x,y) &= \binom{n}{0}A_0x^n + \binom{n}{1}A_1x^{n-1}y + \binom{n}{2}A_2x^{n-2}y^2 + \ldots + \binom{n}{n}A_ny^n \end{aligned}$$

where y is a dummy variable. Then the Hessian of the polynomial $H(f)$ can be derived from the symmetrical formula for $H(U)$ leading to the expression for $h(x)$ in the Hill plot slope as follows

$$\begin{aligned} H(U) &= \frac{1}{n^2(n-1)^2}[U_{xx}U_{yy} - U_{xy}^2] \\ H(f) &= \frac{h(x)}{n^2(n-1)} \end{aligned}$$

using Euler's theorem on homogeneous functions and setting $y = 1$ after differentiating.

Definition of cooperativity

Given a binding polynomial of degree n there are $n - 1$ cooperativity coefficients c_i defined as

$$c_i = B_{i+1} - B_i \text{ for } i = 1, 2, \ldots, n-1,$$

or alternatively as $\log(B_{i+1}/B_i)$, and the interpretation of these is perfectly clear: in a situation where $c_i > 0$ the macromolecule has greater affinity for binding the $i + 1$th ligand after the ith ligand has been bound and it is perfectly reasonable to describe this as mechanistic positive cooperativity. Hence every binding situation for n ligands can be summarized by a succession of $n - 1$ signs and it might be thought that during the actual saturation of macromolecule with ligand there would be a succession of phases with possibly differing cooperativity. For instance, the sequence $+ - +$ might be supposed to give a saturation curve with positive, then negative, then positive cooperativity. Unfortunately the cooperativity coefficients cannot be interpreted in this way and they are not a unique indicator of the sign and magnitude of the type of cooperativity exhibited during the saturation process. The reason for this is simply that binding does not occur in a succession of isolated steps and at every stage for $0 < x < \infty$ every species that is possible is present, that is no ligands bound, one ligand bound, two ligands bound, etc. up to n ligands bound.

At every point in the range $0 < x < \infty$ there is a one site binding curve y_{app} with a uniquely defined apparent binding constant K_{app} according to the scheme

$$[\text{Free sites}] + X \rightleftharpoons [\text{Occupied sites}]$$

that is

$$y_{app}(x) = \frac{K_{app}x}{1 + K_{app}x}.$$

Surely all would agree that the sign and magnitude of cooperativity at that point in the saturation curve would depend on whether K_{app} is increasing or decreasing as a function of x. It turns out that

$$K_{app} = \frac{p'(x)}{np(x) - xp'(x)} \text{ and}$$
$$\frac{dK_{app}}{dx} = \frac{h(x)}{(np(x) - xp'(x))^2}$$

so that increasing affinity (i.e. positive cooperativity) requires that the Hessian of the binding polynomial $h(x)$ has $h(x) > 0$, decreasing affinity (i.e. negative cooperativity) requires $h(x) < 0$ while at a point where $h(x) = 0$ cooperativity changes sign. Bardsley and Wyman [4] emphasized that the magnitude of the Hill slope with respect to 1 is the unambiguous indicator of cooperativity which also depends on the sign of the Hessian $h(x)$ as follows

$$\frac{d\log[y/(1-y)]}{d\log x} = 1 + \frac{xh(x)}{p'(x)(np(x) - xp'(x))}.$$

and Wood and Bardsley [5] proved that the Hessian can have at most $n - 2$ positive zeros.

Zeros of the binding polynomial

If the n zeros of the binding polynomial are α_i then the fractional saturation y can be expressed as

$$y = \left(\frac{x}{n}\right) \sum_{i=1}^{n} \frac{1}{x - \alpha_i},$$

but further discussion depends on the nature of the zeros.

First observe that, for a set of m groups of receptors, each with n_i independent binding sites and binding constant k_i, then the zeros are all real and

$$p(x) = \prod_{i=1}^{m} (1 + k_i x)^{n_i},$$

$$\text{and } y = \frac{1}{\sum_{i=1}^{m} n_i} \sum_{i=1}^{m} \frac{n_i k_i x}{1 + k_i x},$$

so y is just the sum of simple binding curves, giving concave down double reciprocal plots, etc.

Actually Bardsley et al [6] and [7] proved that, if a binding polynomial factorizes into m polynomials p_i with positive coefficients according to

$$p(x) = p_1(x) p_2(x) \ldots p_m(x)$$

then the Hill plot slope cannot exceed that of the Hill plot slope for any of the individual factors. As a binding polynomial can always be factorized into a product of linear factors with real negative zeros and complex conjugate pairs forming quadratic factors it might be supposed that the Hill slope can never exceed two. However, if a binding polynomial of degree > 2 has complex conjugate zeros, the Hill slope may exceed two and there may be evidence of strong positive cooperativity. That is why Hill plot slopes up to a maximum of the degree of the binding polynomial can be achieved if there are quadratic factors with negative coefficients, corresponding to a group of at least three linked binding sites.

For instance, the binding polynomial for a four site Monod-Wyman-Changeux model is

$$p(\alpha) = \frac{1}{1 + L} \left((1 + \alpha)^4 + L(1 + c\alpha)^n \right)$$

and this can factorize into the form

$$q(x) = (1 + a_1 x + b_1 x^2)(1 - a_2 x + b_2 x^2)$$

with $a_1 > 0, a_2 > 0, b_1 > 0, b_2 > 0$ under certain constraints so that the meaningless quadratic factor with a negative term allows Hill slopes greater than two.

Edelstein and Bardsley [8] subsequently explored the relationship between the Hill slope at half-saturation and the Hessian of the binding polynomial.

Statistical interpretation of saturation functions

The species fractional populations s_i which are defined for $i = 0, 1, \ldots, n$ as

$$s_i = \frac{K_i x^i}{K_0 + K_1 x + K_2 x^2 + \cdots + K_n x^n}$$

with $K_0 = 1$, are interpreted as the proportions of the receptors in the various states of ligation as a function of ligand activity. The species fractions defined as $y_i = i s_i / n$ for $i = 1, 2, \ldots, n$ are the contributions of the

species to the overall saturation. Note that

$$\sum_{i=0}^{n} s_i = 1, , \text{ while}$$
$$\sum_{i=1}^{n} y_i = (1/n) d \log p / d \log x.$$

Such expressions are very useful when analyzing cooperative ligand binding data and they can be generated from the best fit binding polynomial after fitting binding curves with program **sffit**, or by interactive input of binding constants into program **simstat**. At the same time other important analytical results like factors of the Hessian and minimax Hill slope are also calculated.

The species fractional populations can be also used in a probability model to interpret ligand binding in several interesting ways. For this purpose, consider a random variable U representing the probability of a receptor existing in a state with i ligands bound. Then the the probability mass function, expected values and variance are

$$\begin{aligned} P(U = i) &= s_i \ (i = 0, 1, 2, \ldots, n), \\ E(U) &= \sum_{i=0}^{n} i s_i, \\ E(U^2) &= \sum_{i=0}^{n} i^2 s_i, \\ V(U) &= E(U^2) - [E(U)]^2 \\ &= x \left(\frac{p'(x) + x p''(x)}{p(x)} \right) - \left(\frac{x p'(x)}{p(x)} \right)^2 \\ &= n \frac{dy}{d \log x}, \end{aligned}$$

as fractional saturation y is $E(U)/n$. In other words, the slope of a semi-log plot of fractional saturation data indicates the variance of the number of occupied sites, namely; all unoccupied when $x = 0$, distribution with variance increasing as a function of x up to the maximum semi-log plot slope, then finally approaching all sites occupied as x tends to infinity. You can input binding constants into the statistical calculations procedure to see how they are mapped into all spaces, cooperativity coefficients are calculated, zeros of the binding polynomial and Hessian are estimated, Hill slope is reported, and species fractions and binding isotherms are displayed, as is done automatically after every $n > 1$ fit by program **sffit**.

Cooperativity analysis

After fitting a model, program **sffit** outputs the binding constant estimates in all the conventions and, when $n > 2$ it also outputs the zeros of the best fit binding polynomial and those of the Hessian of the binding polynomial $h(x)$.

The positive zeros of $h(x)$ indicate points where the theoretical one-site binding curve coinciding with the actual saturation curve at that x value has the same slope as the higher order saturation curve, which are therefore points of cooperativity change. The SimFIT cooperativity procedure allows users to input binding constant estimates retrospectively to calculate zeros of the binding polynomial and Hessian, and also to plot species population fractions.

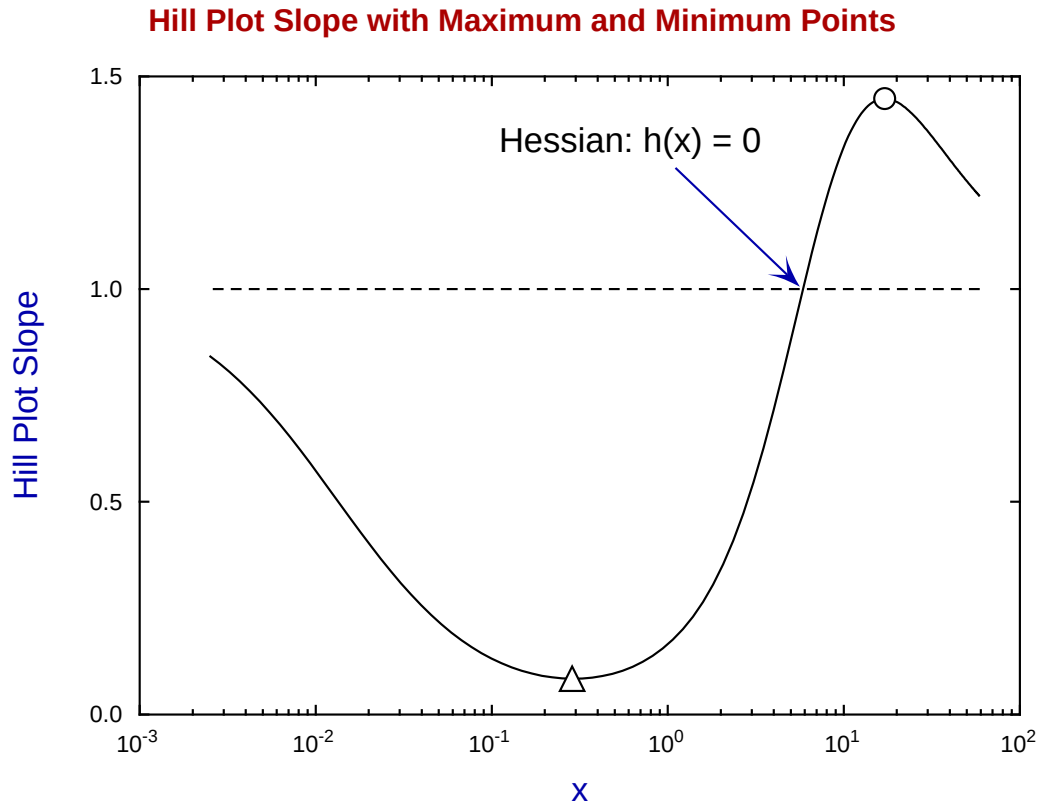
For instance, for 4 sites with

$$K_1 = 100, K_2 = 10, K_3 = 1, \text{and } K_4 = 0.1,$$

the Hessian has these characteristic features

positive zero at $x = 5.86139$
minimum Hill slope in the range plotted is 0.0842, at $x = 0.28607$
maximum Hill slope is 1.44479, at $x = 17.059$, and
the slope at half saturation is 1.0847, at $x = 6.5808$.

The next graph shows the plot of the Hill slope and illustrates how it varies for these K_i values leading to the maximum and minimum slopes indicated along with the point where the positive zero of the Hessian occurs.

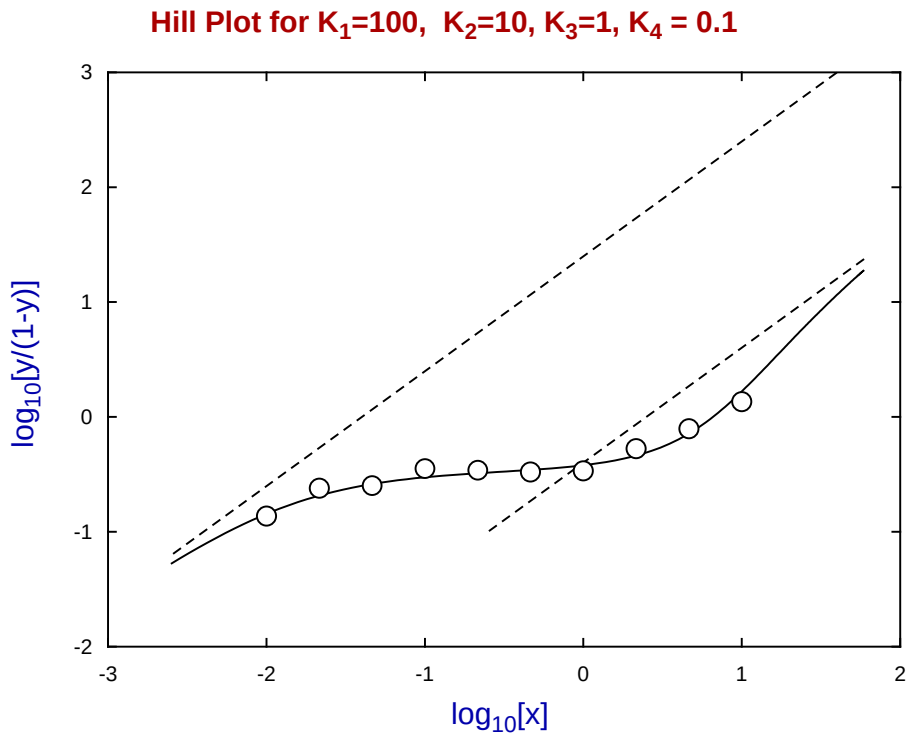


The next graph is the actual Hill plot obtained using these K_i values which shows the sort of complicated Hill plots that can be obtained when there are more than two cooperatively linked sites, that is, where up to $n - 2$ zeros of the Hessian of the binding polynomial can occur.

The asymptotes are for the equation

$$y = \frac{kx}{1 + kx}$$

with $k = K_1/n$ as $x \to 0$ and $k = nK_n/K_{n-1}$ as $x \to \infty$, and the zero of the Hessian is where the slope changes from less than 1 to greater than one,



References

[1] The possible effects of the aggregation of the molecules of haemoglobin on its dissociation curves.
Hill,A.V. (1910), *J.Physiol.* **40**, 4-7.

[2] The hemoglobin system. VI. The oxygen dissociation curve of hemoglobin.
Adair,G.S. (1925) *J.Biol. Chem.* **63**, 529-545.

[3] Allosteric Linkage.
Wyman,J. (1967), *J.Amer.Chem.Soc. 89.9* 2202-2218.

[4] Concerning the thermodynamic definition and graphical manifestations of positive and negative co-operativity.
Bardsley,W.G. & Wyman,J. (1978) *J. theor. Biol.* **72**, 373-376

[5] Critical points and sigmoidicity of positive rational functions.
Wood,R.M.W. & Bardsley,W.G. (1985) *Amer. Math. Month.* **92(1)**, 37-48

[6] Relationships between the magnitude of Hill plot slopes, apparent binding constants and factorability of binding polynomials and their Hessians.
Bardsley,W.G., Woolfson,R. & Mazat,J,-P. (1980) *J. theor. Biol.* **85**, 247-284

[7] Factorability of the Hessian of the binding polynomial. The central issue concerning statistical ratios between binding constants, Hill plot slope and positive and negative co-operativity.
Bardsley,W.G. & Waight,R.D. (1978) *J. theor. Biol.* **72**, 321-372

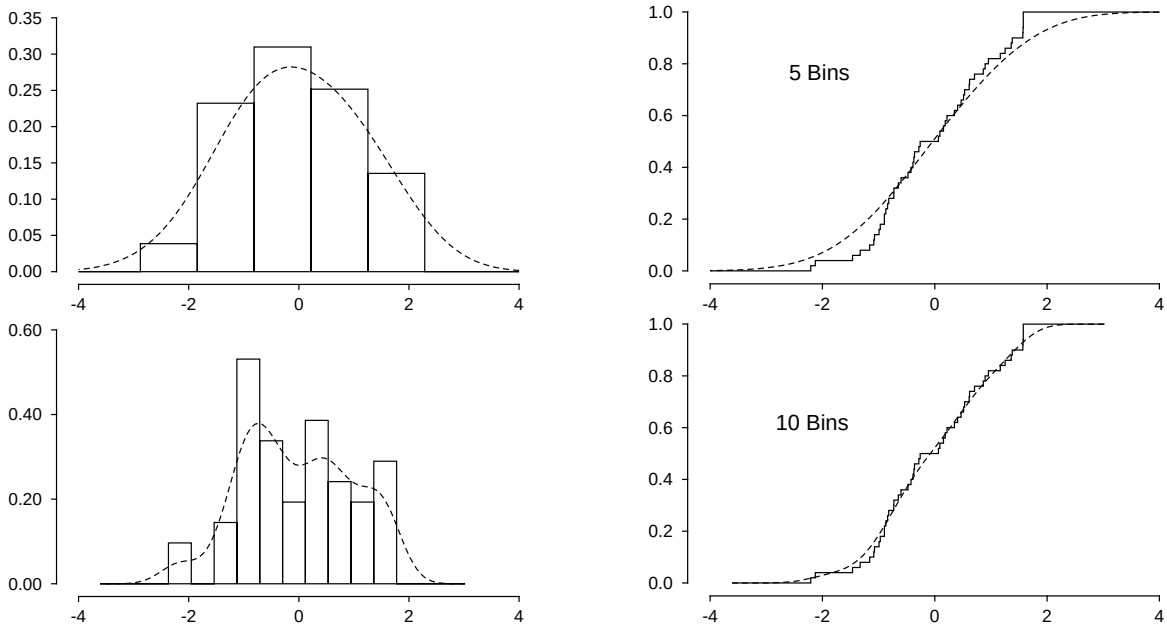
[8] Contributions of individual molecular species to the Hill coefficient for ligand binding by an oligomeric protein.
Edelstein,S.J & Bardsley,W.G. *J. Mol. Biol* (1997) **267**, 10–16

5.8 Gaussian kernel density estimation using FFT

Kernel density estimation is a technique used to create a numerical approximation to a density function given a random sample of observations for which there is no known density.

To use this method choose [Statistics] from the main SimFIT menu then [Statistical calculations] and select [Kernel density estimation].

At this stage it is necessary to input a sample of observations and the following figure illustrates the results when this was done with a data set simulated from a normal distribution with $\mu = 0$ and $\sigma^2 = 1$, using 5 bins for the histogram in the top row of figures, but using 10 bins for the histogram in the bottom row.



The parameters used for the method are adjusted until a satisfactory fit has been obtained when it is then possible to save the best-fit kernel to be used retrospectively as a representation of the data set. However it should be noted that users have to exert considerable control over the parameters chosen as these will greatly affect the kernel estimated. Understanding of the meaning of the parameters selected helps, but naturally a visual display of the fit of the kernel estimate using a reasonable number of bins is recommended.

For instance, in this example changing the number of bins k alters the density estimate since, given a sample of n observations $x_1, x_2, \ldots, x_n$ with $A \le x_i \le B$, the Gaussian kernel density estimate $\hat{f}(x)$ is defined as

$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right)$$

$$\text{where } K(t) = \frac{1}{\sqrt{2\pi}} \exp(-t^2/2)$$

$$\text{and in this case } h = (B - A)/(k - 2).$$

Clearly, a window width h similar to the bin width, as in the top row, can generate an unrealistic over-smoothed density estimate, while using narrower many bins, as in the second row, can lead to over-fitting.

Details are as follows.

- The calculation involves four steps.

  1. From the n data points x_i choose a lower limit a, an upper limit b, and m equally spaced points t_i where
  $$a = A - 3h \leq t_i \leq B + 3h = b,$$
  and m is power of 2. The value of m can be altered interactively from the default value of 128 if necessary for better representation of multi-modal profiles. Data are discretized by binning the x_i at points t_i to generate weights ξ_l.
  2. Compute FFT of the weights, ξ_l to give Y_l.
  3. Compute $\xi_l = Y_l \exp(h^2 s_l^2/2)$ where $s_l = 2\pi l/(b-a)$
  4. Find the inverse FFT of ξ_l to give $\hat{f}(x)$.

- The histograms shown on the left use k bins to contain the sample, and the height of each bin is the fraction of sample values in the bin. The value of k can be changed interactively, and the dotted curves are the density estimates for the m values of t. The program generates additional empty bins for the FFT outside the range set by the data to allow for tails. However, the total area under the histogram is one, and the density estimate integrates to one between $-\infty$ and ∞.

- In addition to the definition of the smoothing parameter h depending on the number of bins chosen for display in the above figure the default setting, which is
  $$h = 1.06\hat{\sigma}n^{-1/5},$$
  uses the sample standard deviation and sample size, as recommended for a normal distribution. Users can also set arbitrary smoothing parameters and, with these two options, the histograms plotted simply illustrate the fit of the kernel density estimate to the data and do not alter the smoothing parameter h.

- The sample cumulative distributions shown on the right have a vertical step of $1/n$ at each sample value, and so they increase stepwise from zero to one. The density estimates are integrated numerically to generate the theoretical cdf functions, which are shown as dashed curves. They will attain an asymptote of one if the number of points m is sufficiently large to allow accurate integration, say ≥ 100.

- The density estimates are unique given the data, h and m, but they will only be meaningful if the sample size is fairly large, say ≥ 50 and preferably much more. Further, the histogram bins will only be representative of the data if they have a reasonable content, say $n/k \geq 10$.

- The histogram, sample distribution, pdf estimate and cdf estimate can be saved to file by selecting the [Advanced] option then creating ASCII text coordinate files.

5.9 False discovery rates FDR(BH)

Multiple testing is when several statistical tests are performed on the same data so it is necessary to control false results. One procedure is the Bonferroni correction where, for m tests and p values, results are considered significant at level α if $p \leq \alpha/m$, rather than $p \leq \alpha$ for single tests.

If at least one of the p values satisfies the Bonferroni restriction, the FDR(BH) false discovery rate technique (Benjamini and Hochberg J.R.satist.Soc. B (1995) 57,1, 289–300, and Benjamini et al Behavioural Brain Research 125 (2001) 279–284) is available to see if there other p values, not necessarily satisfying the Bonferroni restriction, that could also be regarded as possibly significant.

Example 1: FDR(BH) for a vector of p values

From the main SIMFIT menu choose [Statistics] then [Statistical calculations] and then [False discovery rates from a vector p(i)] and scrutinize the default test file `fdr_bh.tf1` provided. After selecting to calculate the false discovery rates, view the table of results for all the data arranged into order of rank which is displayed next. Here m is the number of tests and i is the rank of the sample in terms of the ordered p values.

False discovery rates for a vector of p(i) values: 1
Title: Data for BH False Discovery rate calculation
Sample size = 17
Number rejected = 0
Number analysed = 17
Significance level, α = 0.05

Rank	Sample	p p-value	$m * p/i$ p-adjusted	$\alpha * i/m$ BH-level	Result
1	12	0.000001	0.000017	0.002941	1
2	1	0.000013	0.000110	0.005882	1
3	3	0.000065	0.000368	0.008824	1
4	6	0.000630	0.002678	0.011765	1
5	5	0.000800	0.002720	0.014706	1
6	16	0.001700	0.004817	0.017647	1
7	2	0.003200	0.007771	0.020588	1
8	7	0.006500	0.013813	0.023529	1
9	11	0.014800	0.027956	0.026471	1
10	13	0.049000	0.083300	0.029412	0
11	14	0.094000	0.145273	0.032353	0
12	17	0.110000	0.155833	0.035294	0
13	9	0.150000	0.196154	0.038235	0
14	8	0.240000	0.291429	0.041176	0
15	15	0.450000	0.510000	0.044118	0
16	10	0.560000	0.595000	0.047059	0
17	4	0.870000	0.870000	0.050000	0

There are other options to view the results in sample order or to just show significant results, but the above table is the easiest to understand and follows the example given by Benjamini et al on this same data set.

In order to understand the FDR(BH) technique we shall explain the meanings of the above columns and, in particular, the interpretation of the colors and meaning of the 1's and 0's in the last column.

1. **Column 1**

   This is the rank i of the sample with respect to the p values. That is, the rows of the table are arranged so that the samples in row i are arranged in order of increasing p values.

2. **Column 2**

   This registers the actual number of the sample in the original order.

3. **Column 3**

   Here are the p values corresponding to the rank recorded in column 1 for the sample identified in column 2.

4. **Column 4**

   If this is table line for rank i then this is the p value adjusted by the rank and the sample size m. In other words, the adjusted p value is mp/i. Note that this column only depends on p, i and m, and the last adjusted p value is always the same as the uncorrected p value since $i = m$.

5. **Column 5**

   Here are listed the BH-levels, i.e., the BH threshold values $\alpha i/m$. Note that these only depend on α, i and m, and they have the following sequence. The value at row 1 is the Bonferroni corrected level for significance testing, and the value at line m is the significance level α, while between these extremes the values slowly increase as a function of the rank.

6. **Column 6**

   This column has a 1 if the sample is in the FDR(BH) set and a 0 otherwise

The systematic FDR(BH) procedure

The technique starts at row m and advances up the table until the first rank is encountered, say k, where the p value is less than or equal to the BH threshold. We then conclude that all samples from line 1 up to line k must be considered as possibly significant. So the set of possibly significant samples contains those where

$$p \leq \alpha i/m,$$
$$\text{or equivalently } mp/i \leq \alpha.$$

So now the importance of the color change will be clear and the interpretation of the table is obvious.

All samples numbered in column 2 up to level k with a 1 in column 6 are colored blue, which makes identification of the set of possibly significant samples easy to recognize.

The table can also be rearranged into sample order and can be displayed in such a way as to only identify the set of possibly significant samples. Also, for very large samples it is possible to scroll through the table to select sections or even to write the whole table to file.

Example 2: FDR(BH) for a matrix of p values

Some procedures result in matrices of p values, and this requires a more complicated approach because we have to keep track of the row and column indices. As a typical example, select the option for false discovery rate for a matrix and read in the default test file `matrix_p.tf1` which is as follows

0.00023	0.00060	0.40906	0.41318
0.00050	0.00005	0.32055	0.23282
0.00560	0.01362	0.43751	0.06327

This results from the directed correlation procedure in the multivariate statistics options using the default test files `matrix_a.tf1` for the A matrix which has dimensions 30 by 3, and `matrix_b.tf1` for the B matrix which has dimensions 30 by 4.

Proceeding with the false discovery option we obtain the following table in rank order.

False discovery rates for a matrix of p(i,j) values: 1
Title: Data from directed correlation
Number of columns = 4
Number of rows = 3
Number out of range = 0
Significance level, α = 0.05

$A(i)$	$B(j)$	p-value	p-adjusted	BH-level	Result
2	2	0.000053	0.000632	0.004167	1
1	1	0.000231	0.001387	0.008333	1
2	1	0.000500	0.002000	0.012500	1
1	2	0.000598	0.001793	0.016667	1
3	1	0.005602	0.013446	0.020833	1
3	2	0.013624	0.027247	0.025000	1
3	4	0.063269	0.108461	0.029167	0
2	4	0.232822	0.349234	0.033333	0
2	3	0.320548	0.427398	0.037500	0
1	3	0.409063	0.490875	0.041667	0
1	4	0.413176	0.450738	0.045833	0
3	3	0.437508	0.437508	0.050000	0

As before the set of possibly significant samples is easy to identify by the 1 in the last column or blue color, but columns 1 and 2 need some explanation.

In this example column 1 indicates what the row indices of p values are, because the matrix of p values had 3 rows which originated from the 3 columns of the A matrix in the directed correlation. The second column identifies column indices for the 4 columns corresponding to the 4 columns of the B matrix. This situation is valid only for this particular matrix, and results from the convention dictating the way that the matrix of p values was constructed.

6 Multivariate analysis

Tutorials and worked examples for simulation, curve fitting, statistical analysis, and plotting.
https://simfit.uk
https://simfit.org.uk
https://simfit.silverfrost.com

6.1 Introduction

Multivariate analysis is used to study n by m data matrices where the n rows represent subjects while the m columns are values for variables observed for the n subjects.

To be precise, consider the possible outcome from testing eight people exposed to mosquito attacks with five different types of clothing as follows, where a 1 indicates attacked by mosquitos and a 0 indicates freedom from attack.

Blocks	Groups (Clothing Type)				
(Subjects)	Light-loose	Light-tight	Dark-long	Dark-short	None
1	0	0	0	1	0
2	1	1	1	1	1
3	0	0	0	1	1
4	1	1	0	1	0
5	0	1	1	1	1
6	0	1	0	0	1
7	0	0	1	1	1
8	0	0	1	1	0

Here there are 8 subjects and 5 variables, but in addition there is a first column identifying the subjects. So the actual data matrix used for analysis would have dimensions $n = 8$ and $m = 5$, whereas the above table has $n = 8$ and $m = 6$ because, with some multivariate techniques provided by SimFIT, the additional first column can be used to identify subjects if the data are rearranged into groups, or if some subjects are excluded from analysis. Note also that it is often useful to exclude selected variables from an analysis and so, if this is done, the remaining columns will be re-numbered.

So, from now on, we shall consider a matrix X with elements x_{ij} as follows

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1m} \\ x_{21} & x_{22} & \cdots & x_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ x_{n1} & x_{n2} & \cdots & x_{nm} \end{bmatrix}$$

where all the values are to be used in a current analysis.

Almost all multivariate techniques require that the vector of column means and the covariance matrix should be estimated from X and, in addition, subsequent analysis will usually require a singular value decomposition (SVD) because it is the most reliable method for determining the rank of a matrix. SimFIT provides the ability to check the rank of any matrix in this way, and this should be done with any data matrices that prove problematical to analyze.

It should be obvious that the units of measurements for the variables should lead to similar values for the x_{ij} so that all m variables have comparable means and variances, otherwise columns with large values will dominate columns with small values. This can be achieved by centralizing the matrix by subtracting the

column means, and then normalizing by dividing by the column standard deviation. If this is required then the SIMFIT program **editmt** can be used to pre-process data matrices, or it can be done interactively before the data are submitted for analysis, or performed automatically by the routine. However, care must be exercised when centralizing and normalizing because some techniques can give biased results if this is done uncritically. For instance, partial least squares will give biased predictions if data sets for calibration and prediction are pre-processed uncritically before analysis.

6.2 Correlation

Tutorials and worked examples for simulation, curve fitting, statistical analysis, and plotting.
https://simfit.uk
https://simfit.org.uk
https://simfit.silverfrost.com

6.2.1 introduction

Correlation analysis is used to study the possible dependence of two or more columns in a n by m data matrix. For instance, consider any two columns in a n by m data matrix such as

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix}$$

where we can select column j where $1 \le j \le m$, and refer to it as X, and column k where $1 \le k \le m$, and refer to it as Y, as long as $j \ne k$. As the data matrix will consist of observations subject to random variation and experimental error the following situations are possible.

1. X and Y are completely independent and there is no relationship whatsoever between them.
2. X and Y are linearly dependent, that is, components x_i and y_i are related in that $y_i \approx \alpha x_i$ for some parameter α.
3. X and Y are monotonically dependent, that is, components x_i and y_i are related in that, roughly speaking, y_i tend to be large when x_i are large, or some similar nonlinear tendency exists.
4. X and Y are nonlinearly dependent, that is, components x_i and y_i are related in that $f(x_i, y_i) \approx 0$ for some nonlinear implicit function $f(x, y) = 0$.
5. X and Y are dependent because they are separately dependent on another column or columns in the data matrix, or else some other factor not represented in the data matrix.

Actually, given any set of n nonsingular (x_i, y_i) pairs, a correlation coefficient r can always be calculated as

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

where $-1 \le r \le 1$ and, using b_{xy} for the slope of the regression of X on Y, and b_{yx} for the slope of the regression of Y on X

$$r^2 = b_{yx} b_{xy}.$$

However, only when X is normally distributed given Y, and Y is normally distributed given X can simple statistical tests be used for significant linear correlation. The most well known facts about r are as follows.

- When X and Y are linearly related with $y_i \approx \alpha x_i$ and $\alpha > 0$ then $r \to 1$.
- When X and Y are linearly related with $y_i \approx \alpha x_i$ and $\alpha < 0$ then $r \to -1$.
- When X and Y are not linearly related then $r \to 0$.

- When the (x_i, y_i) pairs are from a bivariate normal distribution with population correlation coefficient ρ_0 equal to zero, then the statistic

$$t = r\sqrt{\frac{n-2}{1-r^2}}$$

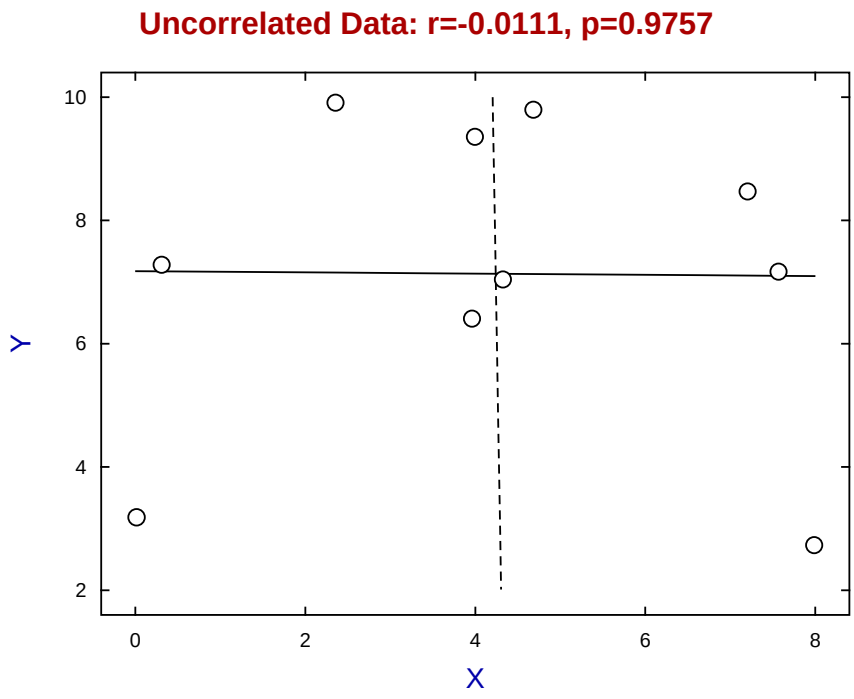
has a Student's t-distribution with $n - 2$ degrees of freedom.

Example 1: Uncorrelated data

Consider this data set

x	y
2.3556	9.9096
0.0165	3.1851
0.3103	7.2811
3.9954	9.3582
7.9854	2.7311
4.3243	7.0423
4.6832	9.7970
7.2031	8.4710
7.5664	7.1706
3.9607	6.4083

which can be displayed as the following scattergram.



Note that, in a correlation scattergram, it is arbitrary which column of the data matrix is chosen for X, and which is chosen for Y. Hence, as it makes no sense to just show the regression line for Y as a function of X, or X as a function of Y, SIMFIT allows you to plot both regression lines. If these regression lines are approximately at right angles it indicates that X and Y are not linearly correlated. Of course the visual check for perpendicularity is best when the same range and scale is used for the coordinates axes, and when a square aspect ratio is employed.

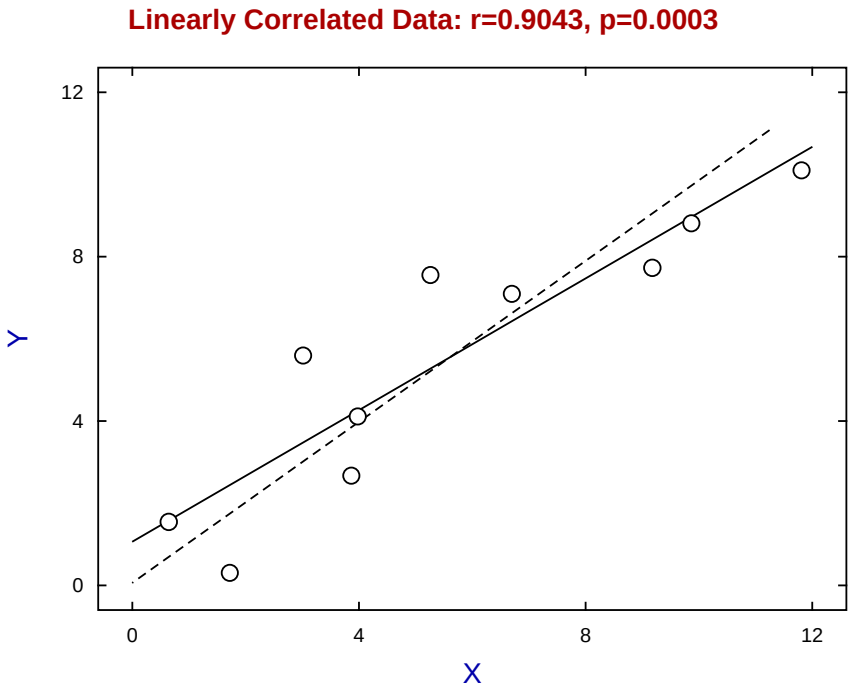
The conclusion is obvious from the scattergram, r value, p value, and almost perpendicular regression lines that these two data columns are not linearly correlated. Note that the usual type of regression line is based on the supposition that the X values are known exactly, but SIMFIT also provides other techniques for plotting a best-fit single regression line when there is variation in both X and Y.

Example 2: Linearly correlated data

Consider this data set

x	y
1.7215	0.3048
0.6453	1.5455
3.8647	2.6689
3.9793	4.1100
3.0151	5.5931
5.2616	7.5528
9.1775	7.7276
6.6972	7.0932
9.8648	8.8121
11.8088	10.0993

which can be displayed as the following scattergram.



The conclusion is obvious from the scattergram, r value, p value, and almost parallel regression lines that these two data columns are linearly correlated.

Note however, that such strong evidence for linear correlation does not imply that Y is really a linear function of X in the sense that X causes Y or vice versa. Often a strong correlation will be due to the dependence of both variables on some other factor such as time, population size, or age. For instance, one study examined that incidence of crime in several cities along with other variables such as the number of churches and reported

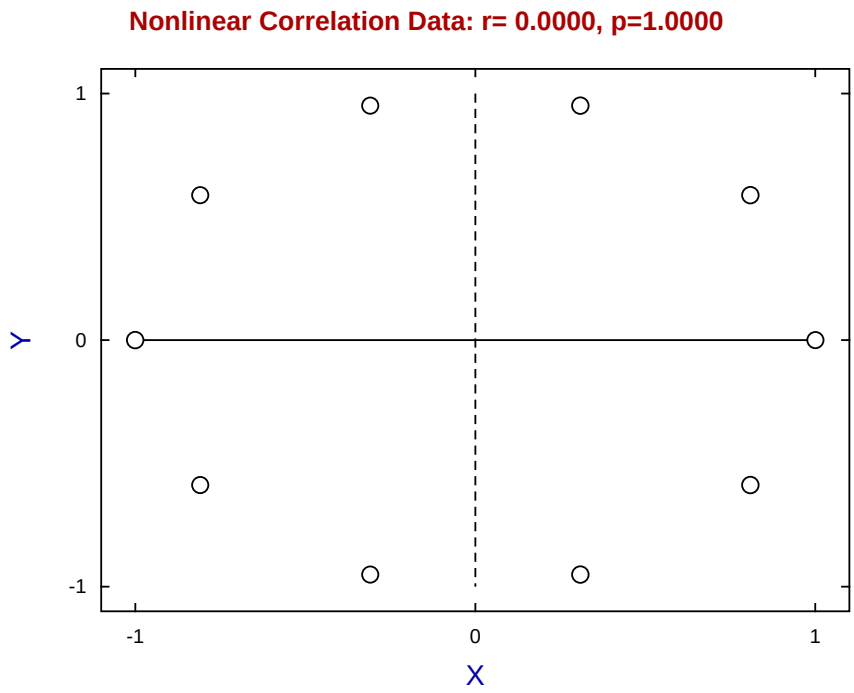
a strong positive correlation between the incidence of crime and the number of churches. This does not, of course, mean that churches cause crime, but merely reflects the fact that large cities will tend to have more crimes but also more churches. SIMFIT provides techniques for studying these sorts of induced correlations.

Example 3: non-linearly correlated data

Consider this data set

x	y
1.0000	0.0000
0.8090	0.5878
0.3090	0.9511
-0.3090	0.9511
-0.8090	0.5878
-1.0000	0.0000
-0.8090	-0.5878
-0.3090	-0.9511
0.3090	-0.9511
0.8090	-0.5878

which can be displayed as the following scattergram.



The conclusion is obvious from the scattergram, r value, p value, and perpendicular regression lines that these two data columns are not linearly correlated.

This example emphasizes an extremely widespread misunderstanding in the application of correlation analysis. It would be harder to find a more obvious example of a data set displaying such extreme nonlinear correlation as this one. Yet the standard technique of relying on r and p values would only conclude an absence of linear correlation, and would not exclude nonlinear correlation. Scatter diagrams showing both regression lines, as in these examples, should always be inspected before making conclusions about possible correlations.

6.2.2 Pearson product-moment correlation

The Pearson product-moment method is used to estimate the amount of linear correlation between paired columns, say X and Y, of a n by m data matrix where it is assumed that the values are of the continuous type from a normal bivariate distribution, and not integers such as frequencies or categorical variables. The null hypothesis is that X and Y are independent, i.e. have zero covariance, that is

$$H_0 : X \text{ and } Y \text{ are from a bivariate normal distribution with } \rho = 0.$$

Example 1

From the SIMFIT main menu choose [Statistics], [Multivariate], [Correlation], then analyze `g02baf.tf1`, the test file provided, using the Pearson product-moment technique. This file contains the following 5 by 3 data matrix

2.0	3.0	3.0
4.0	6.0	4.0
9.0	9.0	0.0
0.0	12.0	2.0
12.0	-1.0	5.0

and analysis leads first to the correlation coefficients and corresponding p values

Matrix A, Pearson correlation results

Upper triangle = r		
Lower triangle = corresponding two-tail p values		
.....	-0.5704	0.1670
0.3153	.....	-0.7486
0.7883	0.1455	.....

which is in the following simplified but comprehensive format

$$A = \begin{bmatrix} \cdots & r_{12} & r_{13} \\ p_{12} & \cdots & r_{23} \\ p_{13} & p_{23} & \cdots \end{bmatrix}$$

where the values a_{ij} for matrix A in the table are interpreted as now described. For $j > i$ in the strict upper triangle, then $a_{ij} = r_{ij} = r_{ji}$ are the correlation coefficients, while for $i > j$ in the strict lower triangle $a_{ij} = p_{ij} = p_{ji}$ are the corresponding two-tail probabilities. In other words, since $r_{ij} = r_{ji}$, $p_{ij} = p_{ji}$, while $r_{ii} = 1$, there will only be $m(m-1)/2$ independent correlations coefficients, and so the diagonal $r_{ii} = 1$ are shown as dots. For instance $r_{12} = -0.5704$ is the correlation coefficient for columns 1 and 2, while $p_{12} = 0.3153$ is the two-tail p value for this correlation coefficient. The table indicates that none of the correlations are significant in this case, that is, the probability of obtaining such pairwise linearity in a random swarm of points from a multivariate normal distribution is not low.

This is then followed by a likelihood ratio test that the full correlation matrix $R = r_{ij}$ for the data matrix is the identity matrix with the following results.

Test for absence of any significant correlations

H_0: correlation matrix is the identity matrix	
Determinant	0.2290
Test statistic (TS)	3.194
Degrees of freedom	3
$P(\chi^2 \geq TS)$	0.3627

To test the hypothesis of no significant correlations, i.e.
H_0: the covariance matrix is diagonal, or equivalently
H_0: the correlation matrix R is the identity matrix, the likelihood ratio test statistic TS, i.e.

$$-2\log\lambda = -(n - (2m + 11)/6)\log|R|$$

is used, where $|R|$ is the determinant of the full correlation matrix (not the previous A matrix) which has the asymptotic chi-square distribution with $m(m-1)/2$ degrees of freedom.

Example 2

This example illustrates the analysis of SIMFIT test file `cluster.tf1` which contains the following data set

1.0	4.0	2.0	11.0	6.0	4.0	3.0	9.0
8.0	5.0	1.0	14.0	19.0	7.0	13.0	21.0
3.0	1.0	3.0	1.0	3.0	6.0	23.0	37.0
9.0	0.0	7.0	7.0	1.0	2.0	21.0	2.0
7.0	12.0	9.0	5.0	14.0	9.0	12.0	14.0
2.0	13.0	15.0	2.0	23.0	6.0	34.0	8.0
11.0	7.0	2.0	1.0	4.0	17.0	11.0	4.0
6.0	3.0	7.0	12.0	11.0	8.0	8.0	0.0
8.0	21.0	1.0	10.0	31.0	9.0	3.0	18.0
19.0	14.0	12.0	9.0	16.0	10.0	0.0	27.0
17.0	18.0	10.0	6.0	19.0	14.0	1.0	24.0
15.0	21.0	8.0	7.0	17.0	12.0	4.0	22.0

leading to this correlation and probability matrix

Upper triangle = r, Lower = corresponding two-tail p values

.....	0.5295	0.2874	0.0662	0.1941	0.6255	-0.5876	0.3010
0.0766	.....	0.3285	-0.0219	0.7930	0.5338	-0.4230	0.3006
0.3650	0.2971	.....	-0.2833	0.2165	0.0264	0.2314	-0.0304
0.8381	0.9460	0.3723	.....	0.2787	-0.2837	-0.5238	-0.1166
0.5455	0.0021	0.4992	0.3804	.....	0.2029	-0.1949	0.2144
0.0296	0.0738	0.9351	0.3715	0.5271	.....	-0.4532	0.1360
0.0445	0.1706	0.4694	0.0805	0.5439	0.1390	.....	-0.1696
0.3418	0.3424	0.9253	0.7181	0.5035	0.6735	0.5983	.....

followed by the results displayed next for a likelihood ratio test.

Test for absence of any significant correlations

H_0: correlation matrix is the identity matrix		
Determinant	0.002476	
Test statistic (TS)	45.01	
Degrees of freedom	28	
$P(\chi^2 \geq TS)$	0.0220	*Reject H_0 at 5% significance level*

From the r values in the strict upper triangle, the p values in the strict lower triangle, and the chi-square test there are linear correlations, and in such cases it would be usual to select pairs of columns for closer analysis.

Analyzing selected pairs of columns

For example, the results for analyzing columns 1 and 2 will be considered.

For the next analysis: X is column 1, Y is column 2

Linear regression: $y(x) = A + B * x, x(y) = C + D * y$			
Sample	size = 12		
For X	mean = 8.8333	std. dev. = 5.7814	var. = 33.424
For Y	mean = 9.9167	std. dev. = 7.5973	var. = 57.720

First the parameter estimates for linear regression are calculated, where Estimate/Standard Error are t values to test for parameters significantly different from zero, Ppmcc is the Pearson product-moment correlation coefficient, and the Fisher z value is used to estimate a 95% confidence region for ρ. In this type of table $p \leq 0.05$ would be required to suggest a nonzero parameter at the 5% significance level.

Parameter	Estimate	Standard Error	Estimate/Standard Error	p
B (slope)	0.69583	0.35252	1.9739	0.0766
A (const)	3.7702	3.6748	1.0260	0.3291
r (Ppmcc)	0.52951	0.26826	1.9739	0.0766
r^2	0.28038			
y-variation due to x = 28.04%				
z(Fisher)	0.58946			

Note: $z = (1/2)\log[(1+r)/(1-r)]$

$r^2 = B * D$, and $t = r * \sqrt{[(n-2)/(1-r^2)]}$ = Estimate/Standard Error for B and D

The Pearson product-moment correlation coefficient r estimates ρ and

95% confidence limits using z are $-0.0771 \le \rho \le 0.8500$

Then this analysis of variance (ANOVA) table is displayed, where the F value is used to test for a significant regression slope. In this type of table $p \le 0.05$ would be required to suggest a nonzero regression slope at the 5% significance level.

Source	Sum of squares	$ndof$	Mean square	F-value	p
due to regression	178.02	1	178.02	3.8962	0.0766
about regression	456.90	10	45.690		
total	634.92	11			

Conclusions:

B is not significantly different from zero ($p > 0.05$)

A is not significantly different from zero ($p > 0.05$)

The two best-fit unweighted regression lines are:

$y(x) = 3.7702 + 0.69583x$, and $x(y) = 4.8375 + 0.40294y$

Various options for plotting follow, and the theory necessary to interpret such correlation tests and visual displays will be presented next.

Theory

Given any set of n nonsingular (x_i, y_i) pairs, a correlation coefficient r can be calculated as

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

where $-1 \le r \le 1$ and, using b_{xy} for the slope of the regression of X on Y, and b_{yx} for the slope of the regression of Y on X

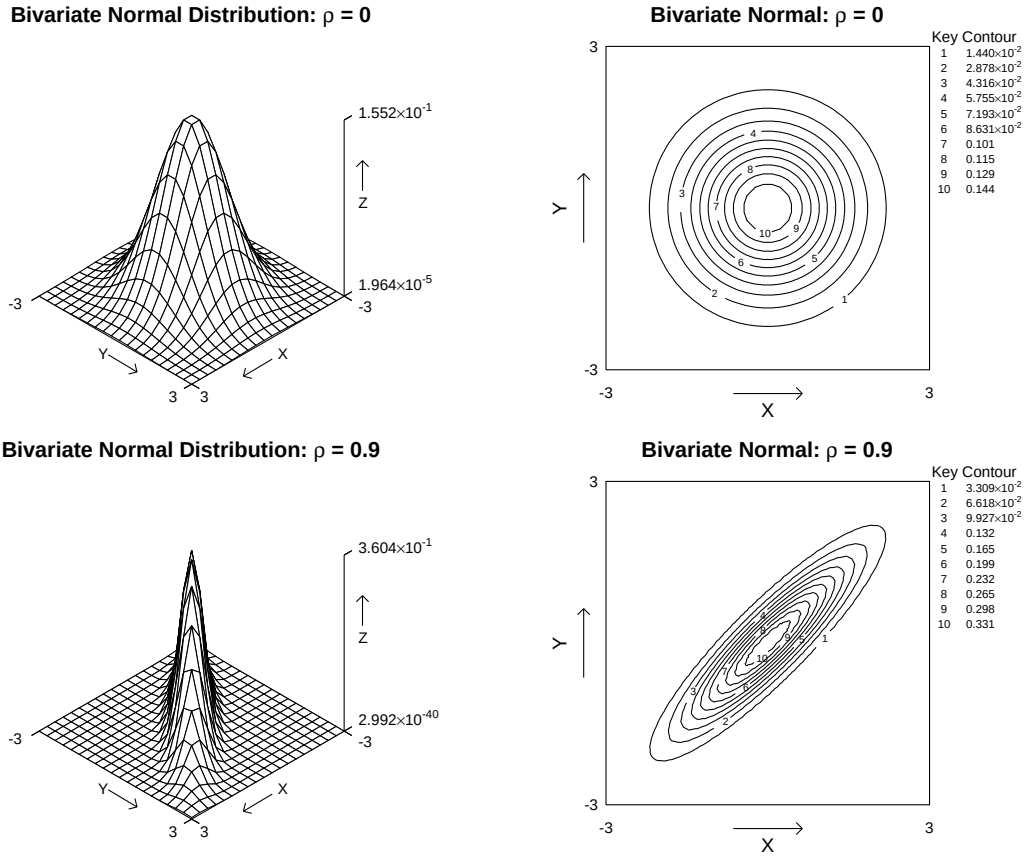
$$r^2 = b_{yx} b_{xy}.$$

However, only when X is normally distributed given Y, and Y is normally distributed given X can simple statistical tests be used for significant linear correlation. For instance, when the (x_i, y_i) pairs are from such a bivariate normal distribution, the statistic

$$t = r\sqrt{\frac{n-2}{1-r^2}}$$

has a Student's t-distribution with $n-2$ degrees of freedom. It is also the t value required to test for nonzero slope in the regression of Y on X, and X on Y, for which a p value can be calculated.

The next figure illustrates how the elliptical contours of constant probability for a bivariate normal distribution are aligned with the X and Y axes when X and Y are uncorrelated, i.e., $\rho = 0$ but are inclined otherwise. In this example $\mu_X = \mu_Y = 0$ and $\sigma_x = \sigma_Y = 1$, but in the upper figure $\rho = 0$, while in the lower figure $\rho = 0.9$. The Pearson product-moment correlation coefficient r is an estimator of ρ, and it can can be used to test for independence of X and Y.



The SIMFIT product-moment correlation procedure can be used when you have a data matrix X consisting of $m > 1$ columns of $n > 1$ measurements (not counts or categorical data) and wish to test for pairwise linear correlations, i.e., where pairs of columns can be regarded as consistent with a bivariate normal distribution. In matrix notation, the relationships between such a n by m data matrix X, the same matrix Y after centering by subtracting each column mean from the corresponding column, the sum of squares and products matrix C, the covariance matrix S, the correlation matrix R, and the diagonal matrix D of standard deviations are

$$\begin{aligned} C &= Y^T Y \\ S &= \frac{1}{n-1} C \\ D &= \mathrm{diag}(\sqrt{s_{11}}, \sqrt{s_{22}}, \ldots, \sqrt{s_{mm}}) \\ R &= D^{-1} S D^{-1} \\ S &= DRD. \end{aligned}$$

So, for all pairs of columns, the sample correlation coefficients r_{jk} are given by

$$r_{jk} = \frac{s_{jk}}{\sqrt{s_{jj} s_{kk}}},$$

$$\text{where } s_{jk} = \frac{1}{n-1} \sum_{i=1}^{n} (x_{ij} - \bar{x}_j)(x_{ik} - \bar{x}_k),$$

and the corresponding t_{jk} values and significance levels p_{jk} are calculated then output in matrix format with the correlations as a strict upper triangular matrix, and the significance levels as a strict lower triangular matrix.

6.2.3 Plotting lines on correlation diagrams

You can plot either both unweighted regression lines, the unweighted reduced major axis line, or the unweighted major axis line on such scattergrams and the difference between these types will now be outlined.

For n pairs (x_i, y_i) with mean $x = \bar{x}$ and mean $y = \bar{y}$, the variances and covariance required are

$$S_{xx} = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$$
$$S_{yy} = \frac{1}{n-1}\sum_{i=1}^{n}(y_i - \bar{y})^2$$
$$S_{xy} = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y}).$$

Also, for an arbitrary point (x_i, y_i) and a straight line defined by $y = a + bx$ the squares of the vertical, horizontal, and orthogonal (i.e. perpendicular) distances, v_i^2, h_i^2, and o_i^2 between the point and the line are

$$v_i^2 = [y_i - (a + bx_i)]^2$$
$$h_i^2 = v_i^2/b^2$$
$$o_i^2 = v_i^2/(1 + b^2).$$

Ordinary least squares

If x is regarded as an exact variable free from random variation or measurement error while y has random variation, then the best fit line from minimizing the sum of v_i^2 is

$$y_1(x) = \hat{\beta}_1 x + [\bar{y} - \hat{\beta}_1 \bar{x}]$$

where $\hat{\beta}_1 = S_{xy}/S_{xx}$. However, if y is regarded as an exact variable while x has random variation, then the best fit line for x as a function of y from minimizing the sum of h_i^2 would be

$$x_2(y) = (1/\hat{\beta}_2)y + [\bar{x} - (1/\hat{\beta}_2)\bar{y}]$$

where $\hat{\beta}_2 = S_{yy}/S_{xy}$ or, rearranging to express the line as $y_2(x)$,

$$y_2(x) = \hat{\beta}_2 x + [\bar{y} - \hat{\beta}_2 \bar{x}],$$

emphasizing that the slope of the regression line for $y_2(x)$ is the reciprocal of the slope for $x_2(y)$. Since neither of these two best fit lines can be regarded as satisfactory, SIMFIT plots both lines such that $y_1(x)$ covers the range of x values while $x_2(y)$ covers the range of y values. However these two lines intersect at $(\bar{x}, \bar{y})$ and, from the fact that the ratio of slopes equals the square of the correlation coefficient, that is,

$$r^2 = \hat{\beta}_1/\hat{\beta}_2,$$

then two best fit lines with similar slopes suggests strong linear correlation, whereas one line almost parallel to the x axis and the other almost parallel to the y axis would indicate negligible linear correlation. For instance, if there is no linear correlation between x and y, then the slope of the regression line for $y(x)$ i.e. $\hat{\beta}_1$ would be zero, as would be the slope of the regression line for $x(y)$ i.e. $1/\hat{\beta}_2$ leading to $r^2 = 0$. Conversely strong linear correlation would lead to $\hat{\beta}_1 = \hat{\beta}_2$ and $r^2 = 1$.

The major axis and reduced major axis lines to be discussed next are attempts to get round the necessity to plot two lines and just have one best fit line intermediate between these two lines to represent the correlation.

The major axis line

Here it is the sum of o_i^2, the squares of the orthogonal distances between the points and the best fit line, that is minimized to yield the slope as

$$\hat{\beta}_3 = \frac{1}{2}\left(\hat{\beta}_2 - (1/\hat{\beta}_1) + \gamma\sqrt{4 + (\hat{\beta}_2 - (1/\hat{\beta}_1))^2}\right)$$

where $\gamma = 1$ if $S_{xy} > 0$, $\gamma = 0$ if $S_{xy} = 0$, and $\gamma = -1$ if $S_{xy} < 0$, so that the major axis line is

$$y_3(x) = \hat{\beta}_3 x + [\bar{y} - \hat{\beta}_3 \bar{x}].$$

Actually $\hat{\beta}_3$ is the slope of the first principal component axis and so it points in the direction of maximum variability.

The reduced major axis line

Instead of minimizing the sum of squares of the vertical distances v_i^2, or horizontal distances h_i^2, it is possible to minimize the sum of the areas of the triangles formed by the v_i, h_i with the best fit line as hypotenuse, i.e. $v_i h_i/2$, to obtain the reduced major axis line as

$$y_4(x) = \hat{\beta}_4 x + [\bar{y} - \hat{\beta}_4 \bar{x}].$$

Here

$$\begin{aligned}\hat{\beta}_4 &= \gamma\sqrt{S_{yy}/S_{xx}} \\ &= \gamma\sqrt{\hat{\beta}_1\hat{\beta}_2}\end{aligned}$$

so that the slope of the reduced major axis line is the geometric mean of the slopes of the regression of y on x and x on y.

6.2.4 Recommendations for plotting lines on scattergrams

1. Plotting both both simple regression lines is the most useful and least controversial. Such lines tending to coincidence indicate strong linear correlation, while lines approaching perpendicularity indicate absence of significant linear correlation.

2. If a single line must be plotted to summarize the overall correlation it should be the reduced major axis line, as this allows for uncertainty in both variables and is not so controversial as the major axis line, which requires both axes to have similar units, as in allometry.

3. It should not be just one of the simple regression lines, since the line plotted must be independent of which variable is regarded as x and which is regarded as y.

6.2.5 Plotting bivariate confidence ellipses: basic theory

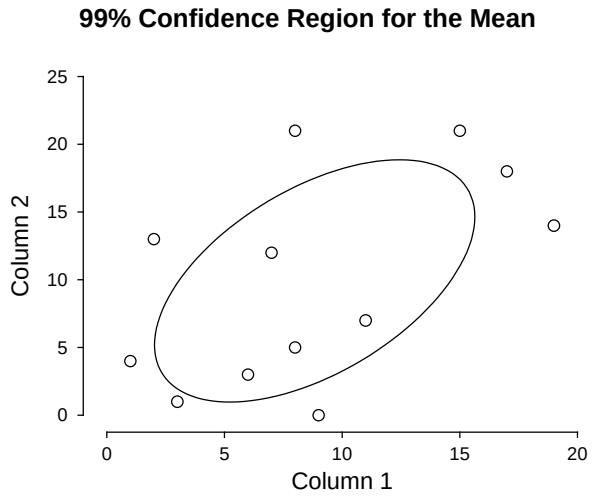
For a p-variate normal sample of size n with mean $\bar{x}$ and variance matrix estimate S, the region

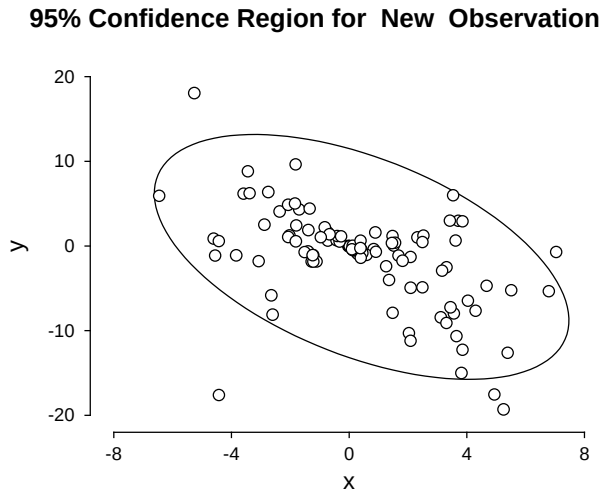
$$P\left\{(\bar{x}-\mu)^T S^{-1}(\bar{x}-\mu) \le \frac{p(n-1)}{n(n-p)} F^{\alpha}_{p,n-p}\right\} \le 1-\alpha$$

can be regarded as a $100(1-\alpha)\%$ confidence region for μ. The next figure illustrates this for columns 1 and 2 of `cluster.tf1` discussed previously. Alternatively, the region satisfying

$$P\left\{(x-\bar{x})^T S^{-1}(x-\bar{x}) \le \frac{p(n^2-1)}{n(n-p)} F^{\alpha}_{p,n-p}\right\} \le 1-\alpha$$

can be interpreted as a region that with probability $1-\alpha$ would contain another independent observation x, as shown for the swarm of points in the next figure.

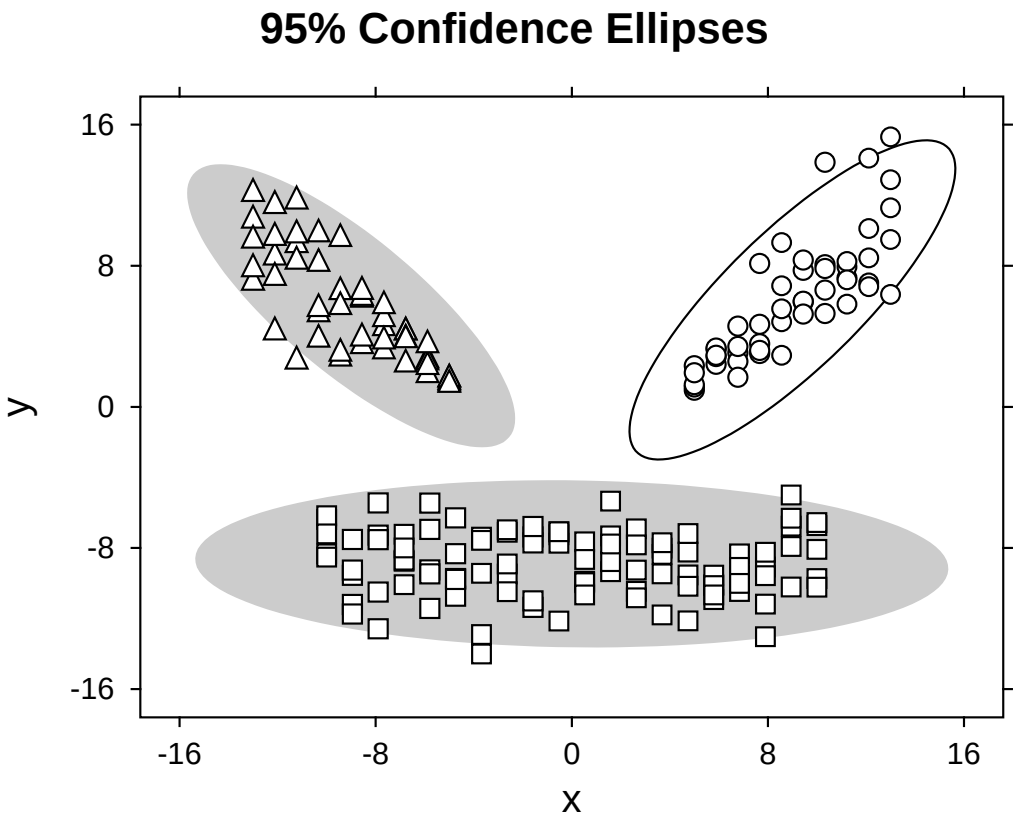




The μ confidence region contracts with increasing n, limiting application to small samples, but the new observation ellipse does not, making it useful for visualizing if data do represent a bivariate normal distribution, while inclination of the principal axes away from parallel with the plot axes demonstrates linear correlation. This technique is only justified if the data are from a bivariate normal distribution and are independent of the variables in the other columns, as indicated by the correlation matrix.

Plotting bivariate confidence ellipses: regions

Often a two dimensional swarm of points results from projecting data that have been partitioned into groups into a subspace of lower dimension in order to visualize the distances between putative groups, e.g., after principal components analysis or similar. If the projections are approximately bivariate normal then confidence ellipses can be added, as in the figure below.



The following steps were used to create this figure and can be easily adapted for any number of sets of two dimensional group coordinates.

1. For each group a file of values for x and y coordinates in the projected space was saved.
2. Each file was analyzed for correlation using the SimFIT correlation analysis procedure.
3. After each correlation analysis, the option to create a 95% confidence ellipse for the data was selected, and the ellipse coordinates were saved to file.
4. A library file was created with the ellipse coordinates as the first three files, and the groups data files as the next three files.
5. The library file was read into **simplot**, then colors and symbols were chosen.

Note that, because the ellipse coordinates are read in as the first coordinates to be plotted, the option to plot lines as closed polygons can be used to represent the confidence ellipses as colored background regions.

6.2.6 Kendall tau and Spearman rank nonparameteric correlation

Nonparametric correlation is required when the data are not distributed according to a multivariate normal distribution, so the Kendall-tau or else the Spearman-rank method is to be preferred. As with the Pearson product-moment correlation technique a n by m data matrix (but now with ranked or ordinal scaled data) is supplied, then SIMFIT calculates all possible pairwise correlation coefficients, and all possible two tail probabilities.

From the SIMFIT main menu choose [Statistics], [Multivariate], [Nonparametric correlation] then analyze the test file `npcorr.tf1` which contains the following data set with $n = 9$ and $m = 3$

1.70	1.00	0.50
2.80	4.00	3.00
0.60	6.00	2.50
1.80	9.00	6.00
0.99	4.00	2.50
1.40	2.00	5.50
1.80	9.00	7.50
2.50	7.00	0.00
0.99	5.00	3.00

to obtain these results.

Matrix A: Correlation coefficients
Upper triangle = Spearman's rank
Lower triangle = Kendall's tau

.....	0.2246	0.1186
0.0294	.....	0.3814
0.1176	0.2353	.....

Matrix B: Two tail p-values

.....	0.5613	0.7611
0.9121	.....	0.3112
0.6588	0.3772	.....

To be more precise, matrices A and B in this table are to be interpreted as follows. In the first matrix A, for $j > i$ in the strict upper triangle, then $a_{ij} = c_{ij} = c_{ji}$ are Spearman correlation coefficients (in black), while for $i > j$ in the strict lower triangle $a_{ij} = \tau_{ij} = \tau_{ji}$ are the corresponding Kendall coefficients (in red).

In the second matrix B, for $j > i$ in the strict upper triangle, then $b_{ij} = p_{ij} = p_{ji}$ are two-tail probabilities for the corresponding c_{ij} coefficients (in black), while for $i > j$ in the strict lower triangle $b_{ij} = p_{ij} = p_{ji}$ (in red) are the corresponding two-tail probabilities for the corresponding τ_{ij}.

For instance, because of symmetry,

- $a_{12} = c_{12} = c_{21} = 0.2246$ with $b_{12} = p - \text{Spearman}_{12} = p - \text{Spearman}_{21} = 0.5613$ refer to the Spearman rank correlation and two-tail p-values for analyzing columns 1 and 2, while
- $a_{32} = \tau_{32} = \tau_{23} = 0.2353$ with $b_{32} = p - \text{Kendall}_{32} = p - \text{Kendall}_{23} = 0.3772$ refer to the Kendall τ correlation and two-tail p-values for analyzing columns 2 and 3.

Note that, from these matrices, τ_{jk}, c_{jk} and p_{jk} values are given for all possible correlations j, k. Also, note that these nonparametric correlation tests are tests for monotonicity rather that linear correlation but, as with the Pearson parametric test, the columns of data must be of the same length and the values must be ordered according to some correlating influence such as multiple responses on the same animals. If the number of categories is small or there are many ties, then Kendall's Tau is to be preferred and conversely. Since you are not testing for linear correlation you should not add regression lines when plotting such correlations.

It should be obvious that SIMFIT displays both sets of results for convenience, and so there are just two possible ways to proceed.

1. Decide in advance which correlation coefficients and corresponding p values to accept, or
2. Apply the Bonferroni or similar correction required for two tests on the same data.

Theory

These nonparametric procedures can be used when the data matrix does not consist of columns of normally distributed measurements, but may contain counts or categorical variables, etc. so that the conditions for Pearson product-moment correlation are not satisfied and ranks have to be used. Suppose, for instance, that the data matrix, say X, has n rows (observations) and m columns (variables) with $n > 1$ and $m > 1$, then the x_{ij} are replaced by the corresponding column-wise ranks y_{ij}, where groups of tied values are replaced by the average of the ranks that would have been assigned in the absence of ties. Kendall's tau τ_{jk} for variables j and k is then defined as

$$\tau_{jk} = \frac{\sum_{h=1}^{n}\sum_{i=1}^{n} f(y_{hj} - y_{ij}) f(y_{hk} - y_{ik})}{\sqrt{[n(n-1) - T_j][n(n-1)T_k]}},$$

$$\text{where } f(u) = 1 \text{ if } u > 0,$$
$$= 0 \text{ if } u = 0,$$
$$= -1 \text{ if } u < 0,$$
$$\text{and } T_j = \sum t_j(t_j - 1).$$

Here t_j is the number of ties at successive tied values of variable j, and the summation is over all tied values. For large samples τ_{jk} is approximately normally distributed with

$$\mu = 0$$
$$\sigma^2 = \frac{4n + 10}{9n(n-1)}$$

which can be used as a test for the absence of correlation.

Another alternative is to calculate Spearman's rank coefficient c_{jk}, defined as

$$c_{jk} = \frac{n(n^2-1) - 6\sum_{i=1}^{n}(y_{ij} - y_{ik})^2 - (T_j + T_k)/2}{\sqrt{[n(n^2-1) - T_j][n(n^2-1)T_k]}}$$

$$\text{where now } T_j = \sum t_j(t_j^2 - 1)$$

and a test can be based on the fact that, for large samples, the statistic

$$t_{jk} = c_{jk}\sqrt{\frac{n-2}{1 - c_{jk}^2}}$$

is approximately t-distributed with $n - 2$ degrees of freedom.

6.2.7 Partial correlation

Partial correlation analysis is used to evaluate the extent to which the correlations between two or more columns (called Y-variables) of a n by m data matrix with $m > 2$ depend on correlations between these columns and other columns in the matrix (called X-variables). Either a data set or a correlation matrix together with sample size can be input, and it is most often used to study the way that the correlations between two columns depend on a third column.

Example 1

From the main SIMFIT menu select [Statistics], [Multivariate], [Partial correlation] and then read in the test file `g02byf.tf1` provided. In the special case when $n = m$ you have to specify whether a data file or correlation matrix is being input, but this is a data matrix with fifteen rows and three columns as follows.

Column 1: number of deaths
Column 2: smoke(mg/m^3)
Column 3: sulphur dioxide(parts/million)

```
112   0.30   0.09
140   0.49   0.16
143   0.61   0.22
120   0.49   0.14
196   2.64   0.75
294   3.45   0.86
513   4.46   1.34
518   4.46   1.34
430   1.22   0.47
274   1.22   0.47
255   0.32   0.22
236   0.29   0.23
256   0.50   0.26
222   0.32   0.16
213   0.32   0.16
```

However the following important trailer section has been added to the data.

```
begin{indicators}
-1   -1   1
end{indicators}
```

Negative indicator values denote Y-variables, zero values indicate suppression, while positive indictor values identify X variables. In other words, the default partial correlation between deaths and smoke is required when sulphur dioxide is considered as fixed. However, it should be noted that the assigning of columns to Y or X groups can also be done interactively.

First the overall Pearson product-moment correlation matrix is calculated and displayed along with the two-tail p-values.

```
Pearson product moment correlation results:
Strict upper triangle: r
Strict lower triangle: corresponding two-tail p values
.....      0.7560   0.8309
0.0011     .....    0.9876
0.0001     0.0000   .....
```

This is then followed by a likelihood ratio test

Test for absence of any significant correlations
H_0: correlation matrix is the identity matrix

Determinant	0.003484	
Test statistic (TS)	68.86	
Degrees of freedom	3	
$P(\chi^2 \geq TS)$	0.0000	*Reject H_0 at 1% significance level*

but, in addition, the partial correlation matrix is displayed as in the next table for variables indicated as YYX. That is, correlation for columns 1 and 2, regarding column 3 as fixed.

Partial correlation results for variables: YYX
Strict upper triangle: partial r
Strict lower triangle: corresponding 2-tail p values

...	-0.7381
0.0026	...

Example 2

This is the test file `pacorr.tf1` which contains a correlation matrix.

Correlation matrix: sample size = 30

3	3	
1.0000	0.6162	0.8267
0.6162	1.0000	0.7321
0.8267	0.7321	1.0000

3
variable 1: Intelligence
variable 2: Weight
variable 3: Age

By systematically altering the definition for Y variables and X variables SIMFIT can calculate all the correlations and partial correlations as follows.

$r(1,2) = 0.6162$
$r(1,3) = 0.8267$
$r(2,3) = 0.7321$
...
$r(1,2|3) = 0.0286$ (95% confidence limits $= -0.3422, 0.3918$)
$t = 0.1488, ndof = 27, p = 0.8828$
...
$r(1,3|2) = 0.7001$ (95% confidence limits $= 0.4479, 0.8490$)
$t = 5.094, ndof = 27, p = 0.0000$ *Reject H_0 at 1% significance level*
...
$r(2,3|1) = 0.5025$ (95% confidence limits $= 0.1659, 0.7343$)
$t = 3.020, ndof = 27, p = 0.0055$ *Reject H_0 at 1% significance level*

From this table it is clear that when variable 3 is regarded as fixed, the correlation between variables 1 and 2 is not significant but, when either variable 1 or variable 2 are regarded as fixed, there is evidence for significant correlation between the other variables. Exactly what commonsense would predict.

Theory

Assuming a multivariate normal distribution and linear correlations, the partial correlations between any two variables from the set i, j, k conditional upon the third can be calculated using the usual correlation coefficients

as

$$r_{i,j|k} = \frac{r_{ij} - r_{ik}r_{jk}}{\sqrt{(1 - r_{ik}^2)(1 - r_{jk}^2)}}.$$

If there are p variables in all but $p - q$ are fixed then the sample size n can be replaced by $n - (p - q)$ in the usual significance tests and estimation of confidence limits, e.g. $n - (p - q) - 2$ for a t test.

The situation is more involved when there are more than three variables, say n_x X variables which can be regarded as fixed, and the remaining n_y Y variables for which partial correlations are required conditional on the fixed variables.

Then the variance-covariance matrix Σ can be partitioned as in

$$\Sigma = \begin{bmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy} \end{bmatrix}$$

when the variance-covariance of Y conditional upon X is given by

$$\Sigma_{y|x} = \Sigma_{yy} - \Sigma_{yx}\Sigma_{xx}^{-1}\Sigma_{xy},$$

while the partial correlation matrix R is calculated by normalizing as

$$R = \operatorname{diag}(\Sigma_{y|x})^{-\frac{1}{2}}\, \Sigma_{y|x}\, \operatorname{diag}(\Sigma_{y|x})^{-\frac{1}{2}}.$$

Exactly as for the full correlation matrix, the strict upper triangle of the output from the partial correlation analysis contains the partial correlation coefficients r_{ij}, while the strict lower triangle holds the corresponding two tail probabilities p_{ij} where

$$p_{ij} = P\left(t_{n-n_x-2} \leq -|r_{ij}|\sqrt{\frac{n - n_x - 2}{1 - r_{ij}^2}}\right) + P\left(t_{n-n_x-2} \geq |r_{ij}|\sqrt{\frac{n - n_x - 2}{1 - r_{ij}^2}}\right).$$

However, for convenience, the output table may display the subscripted partial correlation coefficients with indicated conditional variables together with confidence limits as in Example 2.

6.2.8 Canonical correlation

Canonical correlation is used to explore the correlations between selected columns of a matrix by calculating transformations into lower-dimensional subspaces where the transformed variables have maximum correlation, and can thus be quantified and visualized

Consider a n by m matrix A with elements a_{ij} as follows

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix}$$

where a subset of n_x columns (i.e. x-variables) will be defined as X, another disjoint subset of n_y columns (i.e. y-variables) will be defined as Y, while n_s columns may be suppressed (i.e. not used in the analysis). Clearly

$$m = n_x + n_y + n_s \text{ where } n_x \geq 1, n_y \geq 1 \text{ and } n_s \geq 0.$$

Example 1

From the main SimFIT menu choose [Statistics], [Multivariate], then [Canonical correlation] and observe the format for the test file `g03adf.tf1` shown below.

```
80.0    58.4    14.0    21.0
75.0    59.2    15.0    27.0
78.0    60.3    15.0    27.0
75.0    57.4    13.0    22.0
79.0    59.5    14.0    26.0
78.0    58.1    14.5    26.0
75.0    58.0    12.5    23.0
64.0    55.5    11.0    22.0
80.0    59.2    12.5    22.0
begin{indicators}
 -1      1       1      -1
end{indicators}
```

The final section after the data matrix specifies the meaning of the above data as follows.

- Column 1: variable 1 ($y(1)$ in this case as indicator(1) = -1)
- Column 2: variable 2 ($x(1)$ in this case as indicator(2) = 1)
- Column 3: variable 3 ($x(2)$ in this case as indicator(3) = 1)
- Column 4: variable 4 ($y(2)$ in this case as indicator(4) = -1)

In other words, the red data values are Y variables while the blue values are X variables. Note that, in this example, there are no variables to be suppressed by setting the corresponding indicator to zero, but in any case the assignment of columns to types X or Y or suppressed can also be done interactively. Analysis leads to the next table of results.

Results from analysis of data in test file `g03adf.tf1`
Variables: $yxxy$
Number of X variables = 2, Number of Y variables = 2, Number unused = 0
Minimum of rank of X and rank of Y = 2

Correlations	Eigenvalues	Proportions	χ^2	$NDOF$	p
0.9570	0.91591	0.8746	14.391	4	0.0061
0.3624	0.13133	0.1254	0.77438	1	0.3789

CVX: Canonical coefficients for centralized X

-0.4261	1.034
-0.3444	-1.114

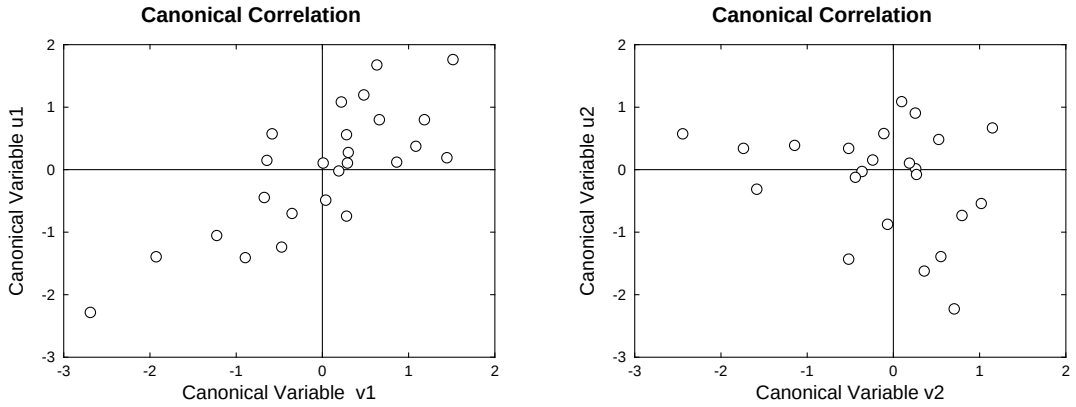
CVY: Canonical coefficients for centralized Y

-0.1415	0.1504
-0.2384	-0.3424

In this table the eigenvalues are proportional to the correlation explained by the corresponding canonical variable, while the χ^2 values and corresponding p values indicate the significance of the successive canonical variables. The results indicate that, with these data, the first canonical variate is sufficient to summarize the correlations between the X and Y variables. Scree diagrams can also be plotted for this purpose.

Example 2

The figure below illustrates two possible graphical displays for the canonical



variates defined by the SIMFIT test file `matrix.tf5`, where columns 1 and 2 are designated the Y sub-matrix, while columns 3 and 4 hold the X matrix. Note that, as eigenvectors do not have unique signs, it is often necessary to reverse the signs of canonical variates for plotting in order to agree with graphs calculated by alternative software. This feature, and also the ability to label the components in such diagrams according to labels added to the data file, is also supported.

Theory

This technique is employed when a n by m data matrix includes at least two groups of variables, say n_x variables of type X, and n_y variables of type Y, measured on the same n subjects, so that $m \geq n_x + n_y$. The idea is to find two transformations, one for the X variables to generate new variables V, and one for the Y variables to generate new variables U, with l components each for $l \leq \min(n_x, n_y)$, such that the canonical variates u_1, v_1 calculated from the data using these transformations have maximum correlation, then u_2, v_2, and so on. Now the variance-covariance matrix of the X and Y data can be partitioned as

$$\begin{pmatrix} S_{xx} & S_{xy} \\ S_{yx} & S_{yy} \end{pmatrix}$$

and it is required to find transformations that maximize the correlations between the X and Y data sets. Actually, the equations

$$(S_{xy}S_{yy}^{-1}S_{yx} - R^2S_{xx})a = 0$$
$$(S_{yx}S_{xx}^{-1}S_{xy} - R^2S_{yy})b = 0$$

have the same nonzero eigenvalues as the matrices $S_{xx}^{-1}S_{xy}S_{yy}^{-1}S_{yx}$ and $S_{yy}^{-1}S_{yx}S_{xx}^{-1}S_{xy}$, and the square roots of these eigenvalues are the canonical correlations, while the eigenvectors of the two above equations define the canonical coefficients, i.e. loadings.

Note that the eigenvalues are proportional to the correlation explained by the corresponding canonical variates, so a scree diagram can be plotted to determine the minimum number of canonical variates needed to adequately represent the data. This diagram plots the eigenvalues together with the average eigenvalue, and the canonical variates with eigenvalues above the average should be retained. Alternatively, assuming multivariate normality, the likelihood ratio test statistics

$$-2\log\lambda = -(n - (k_x + k_y + 3)/2)\sum_{j=i+1}^{l}\log(1 - R_j^2)$$

can be calculated for $i = 0, 1, \ldots, l-1$, where $k_x \leq n_x$ and $k_y \leq n_y$ are the ranks of the X and Y data sets and $l = \min(k_x, k_y)$. These are asymptotically chi-square distributed with $(k_x - i)(k_y - i)$ degrees of freedom, so that the case $i = 0$ tests that none of the l correlations are significant, the case $i = 1$ tests that none of the remaining $l - 1$ correlations are significant, and so on. If any of these tests in sequence are not significant, then the remaining tests should, of course, be ignored.

The previous figure illustrates two possible graphical displays for the canonical variates defined by `matrix.tf5`, where columns 1 and 2 are designated the Y sub-matrix, while columns 3 and 4 hold the X matrix. The canonical variates for X are constructed from the n_x by n_{cv} loading or coefficient matrix CVX, where $CVX(i, j)$ contains the loading coefficient for the ith x variable on the jth canonical variate u_j. Similarly CVY) is the n_y by n_{cv} loading coefficient matrix for the ith y variable on the jth canonical variate v_j. More precisely, if cvx_j is column j of CVX, and cvy_j is column j of CVY, while $x(k)$ is the vector of centralized X observations for case k, and $y(k)$ is the vector of centralized Y observations for case k, then the components $u(k)_j$ and $v(k)_j$ of the n vector canonical variates u_j and v_j are

$$v(k)_j = cvx_j^T x(k), \quad k = 1, 2, \ldots, n$$
$$u(k)_j = cvy_j^T y(k), \quad k = 1, 2, \ldots, n.$$

It is important to realize that the canonical variates for U and V do not represent any sort of regression of Y on X, or X on Y, they are just new coordinates chosen to present the existing correlations between the original X and Y in a new space where the correlations are then ordered for convenience as

$$R^2(u_1, v_1) \geq R^2(u_2, v_2) \geq \ldots \geq R^2(u_l, v_l).$$

Clearly, the left hand plot shows the highest correlation, that is, between u_1 and v_1, whereas the right hand plot illustrates weaker correlation between u_2 and v_2. Note that further linear regression and correlation analysis can also be performed on the canonical variates if required, and also the loading matrices can be saved to construct canonical variates using the SimFIT matrix multiplication routines, and vectors of canonical variates can be saved directly from plots like those displayed.

6.3 Cluster analysis

Tutorials and worked examples for simulation, curve fitting, statistical analysis, and plotting.
https://simfit.uk
https://simfit.org.uk
https://simfit.silverfrost.com

6.3.1 Introduction

In order to separate a set of objects into categories according to some measure of similarity between individual items there has to be some concept of the distance between them. For instance, for two sets of coordinates $\alpha = (x_1, y_1)$ and $\beta = (x_2, y_2)$ we could use the square of the Euclidean distance between them, that is

$$||\alpha - \beta|| = (x_1 - x_2)^2 + (y_1 - y_2)^2$$

as this is the squared length of the hypotenuse of a right angle triangle with coordinates (x_1, y_1) and (x_2, y_2). We could then group items together depending on such a distance measure between them or according to distances from some fixed points. Cluster analysis extends such a concept to situations involving more than two dimensions, and using alternative measures of distance.

Calculating a distance matrix

The idea is, as in data mining, where you have a n by m matrix a_{ij} of m variables (columns) for each of n cases (rows) and wish to explore clustering, that is groupings together of like entities. To do this, you choose an appropriate pre-analysis transformation of the data, a suitable distance measure, a meaningful scaling procedure, and a sensible linkage function. SIMFIT will then calculate a distance matrix, or a similarity matrix, and plot the clusters as a dendrogram. As an example, from the main SIMFIT menu choose [Statistics], [Multivariate], then [Distance matrix] and analyze the test file `cluster.tf1` giving the results displayed in this table.

Variables included:
1 2 3 4 5 6 7 8
Transformation: Untransformed
Distance: Euclidean distance
Scaling: Unscaled
Linkage: Group average
Weighting: [weights r not used]
Distance matrix (strict lower triangle) is:

2)	22.0								
3)	36.2	28.8							
4)	22.9	29.7	36.6						
5)	1.95	16.6	31.1	24.5					
6)	39.8	32.7	40.6	31.8	26.1				
7)	21.7	28.3	38.2	21.3	19.3	36.2			
8)	14.1	24.1	42.6	18.8	18.9	34.2	18.5		
9)	32.7	23.0	45.4	44.9	23.6	38.7	36.6	33.4	
10)	31.6	23.9	37.2	41.0	22.2	43.9	33.5	33.9	(+)
10)	24.7								
11)	32.2	24.4	39.1	41.8	20.2	41.4	31.3	33.4	(+)
11)	19.9	8.25							
12)	29.9	22.7	37.7	39.0	17.2	38.4	29.2	31.4	(+)
12)	18.1	11.4	6.24						

Note that, as a distance matrix is symmetrical with diagonals = 0, only the strict lower triangle is displayed. The header to this table indicates that all eight variables were included in the analysis using untransformed data, the Euclidean distance, no data scaling, group average linkage, and no weights. The symbol (+) merely indicates wrap round due to long lines. The meaning of the parameter settings in the table header will now be explained.

Distance matrix norms

The distance d_{jk} between objects j and k for variable i is just a chosen variant of the weighted L_p norm

$$d_{jk} = \{\sum_{i=1}^{m} w_{ijk} D(a_{ji}/s_i, a_{ki}/s_i)\}^p.$$

for some D and weighting factors w_{ijk}.

For example, for two vectors α and β there would be one of three possibilities.

(a) The Euclidean distance $D(\alpha, \beta) = ||\alpha - \beta||$ with $p = 1/2$

(b) The Euclidean squared difference $D(\alpha, \beta) = ||\alpha - \beta||$ with $p = 1$

(c) The absolute distance $D = |\alpha - \beta|$ with $p = 1$, otherwise known as the Manhattan or city block metric.

However, as the values of the variables may differ greatly in size, so that large values would dominate the analysis, it is usual to subject the data to a preliminary transformation or to apply a suitable weighting s_i for variable i. Often it is best to transform the data to standardized $(0, 1)$ form before constructing the dendrogram, or at least to use some sort of scaling procedure such as:

(i) use the sample standard deviation as s_i,

(ii) use the sample range as s_i, or

(iii) supply precalculated values of s_i.

Usually the weighting factor w_{ijk} would have the default value 1 but there are exceptions as follows. Bray-Curtis dissimilarity uses the absolute distance except that the weighting factor is given by

$$w_{ijk} = \frac{1}{\sum_{i=1}^{m}(a_{ji}/s_i + a_{ki}/s_i)}$$

which is independent of the variables i and only depends on the cases j and k, and distances are usually multiplied by 100 to represent percentage differences. Bray-Curtis similarity is the complement, i.e., 100 minus the dissimilarity.

The Canberra distance measure, like the Bray-Curtis one, also derives from the absolute distance except that the weighting factor is now

$$w_{ijk} = \frac{1}{\lambda(a_{ji}/s_i + a_{ki}/s_i)}.$$

There are various conventions for defining λ and deciding what to do when values or denominators are zero with the Bray-Curtis and Canberra distance measures, and the scheme used by SIMFIT is as follows.

- If any values are negative the calculation is terminated.
- If any Bray-Curtis denominator is zero the calculation is terminated.
- If there are no zero values, then λ is equal to the number of variables in the Canberra measure.

- If both members of a pair are zero, then λ is decreased by one for each occurrence of such a pair, and the pairs are ignored.
- If one member of a pair is zero, then it is replaced by the smallest non-zero value in the data set divided by five, then scaled if required.

Distance matrix linkage

The values in a distance matrix will affect subsequent analysis. For instance, the shape of a dendrogram depends on the choice of analytical techniques and the order of objects plotted is arbitrary: groups at a given fixed distance can be rotated and displayed in either orientation. Another choice which will affect the dendrogram shape is the method used to recalculate distances after each merge has occurred. Suppose there are three clusters i, j, k with n_i, n_j, n_k objects in each cluster and let clusters j and k be merged to give cluster jk. Then the distance from cluster i to cluster jk can be calculated in several ways.

[1] Single link: $d_{i,jk} = \min(d_{ij}, d_{ik})$

[2] Complete link: $d_{i,jk} = \max(d_{ij}, d_{ik})$

[3] Group average: $d_{i,jk} = (n_j d_{ij} + n_k d_{ik})/(n_j + n_k)$

[4] Centroid: $d_{i,jk} = (n_j d_{ij} + n_k d_{ik} - n_j n_k d_{jk}/(n_j + n_k))/(n_j + n_k)$

[5] Median: $d_{i,jk} = (d_{ij} + d_{ik} - d_{jk}/2)/2$

[6] Minimum variance: $d_{i,jk} = \{(n_i + n_j)d_{ij} + (n_i + n_k)d_{ik} - n_i d_{jk}\}/(n_i + n_j + n_k)$

Distance matrix nearest neighbors

Once a distance matrix has been calculated, it is sometimes useful to calculate the nearest neighbors, as illustrated in the next table for the previous data.

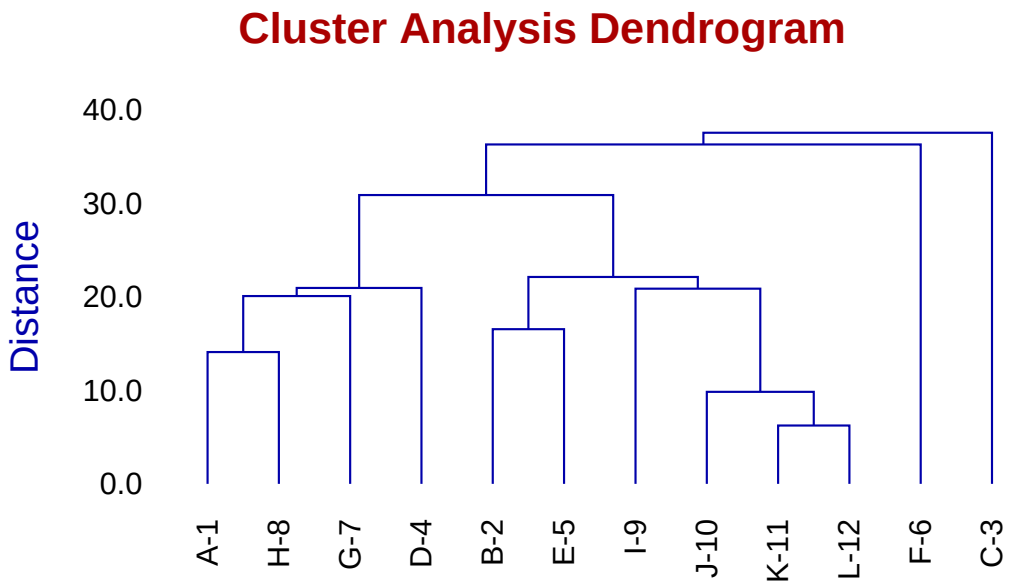
Object	Nearest	Distance
1	8	14.1067
2	5	16.5529
3	2	28.7576
4	8	18.7617
5	2	16.5529
6	5	26.0960
7	8	18.4932
8	1	14.1067
9	12	18.1384
10	11	8.24621
11	12	6.24500
12	11	6.24500

In this table, column 1 refers to the objects in logical order, column 2 indicates the object that is closest, i.e., the nearest neighbor, while column 3 records these minimum distances. Clearly, the nearest neighbors will depend upon the parameters used to configure the calculation of the distance matrix.

6.3.2 Dendrograms

Dendrograms can be plotted after a distance matrix has been calculated and a linkage technique has been selected in order to build up a picture as to how merging can be used to partition samples into subgroups as defined by distance thresholds.

For example, open the main SimFIT menu choose [Statistics], [Multivariate], then [Dendrograms] and read in the test file `cluster.tf1`, which should also be examined to see how to provide labels, as illustrated when this figure is displayed.



Of course the precise shape of such a figure depends on the metric and weights, etc. used to calculate the distance matrix and the linkage assumed when building up the groups. Further details about using SimFIT to construct dendrograms will now be discussed.

Partial clustering

An important application of distance matrices and dendrograms is in partial clustering. Unlike the situation with full clustering where we start with n groups, each containing a single case, and finish with just one group containing all the cases, in partial clustering the clustering process is not allowed to be completed. There are two distinct ways to arrest the clustering procedure.

1. A number, K, between 1 and $n-1$ is chosen, and clustering is allowed to proceed until just K subgroups have been formed. It may not always be possible to satisfy this requirement, e.g. if there are ties in the data.

2. A threshold, D, is set somewhere between the first clustering distance and the last clustering distance, and clustering terminates when this threshold is reached. The position of such clustering thresholds will be plotted on the dendrogram, unless D is set equal to zero.

As an example of this technique consider the results in this table

Group assignments for Fisher Iris data
Data file: `iris.tf1`, 3 groups, variables included: 1 2 3 4
Transformation: Untransformed, Distance: Euclidean, Scaling: Unscaled,
Linkage: Group average, [weights not used], sub-clusters for K = 3
Odd rows: data ... Even rows: corresponding group number

1	2	3	4	5	6	7	8	9	10	11	12
1	1	1	1	1	1	1	1	1	1	1	1
13	14	15	16	17	18	19	20	21	22	23	24
1	1	1	1	1	1	1	1	1	1	1	1
25	26	27	28	29	30	31	32	33	34	35	36
1	1	1	1	1	1	1	1	1	1	1	1
37	38	39	40	41	42	43	44	45	46	47	48
1	1	1	1	1	1	1	1	1	1	1	1
49	50	51	52	53	54	55	56	57	58	59	60
1	1	2	2	2	2	2	2	2	2	2	2
61	62	63	64	65	66	67	68	69	70	71	72
2	2	2	2	2	2	2	2	2	2	2	2
73	74	75	76	77	78	79	80	81	82	83	84
2	2	2	2	2	2	2	2	2	2	2	2
85	86	87	88	89	90	91	92	93	94	95	96
2	2	2	2	2	2	2	2	2	2	2	2
97	98	99	100	101	102	103	104	105	106	107	108
2	2	2	2	2*	2*	3	2*	2*	3	2*	3
109	110	111	112	113	114	115	116	117	118	119	120
2*	3	2*	2*	2*	2*	2*	2*	2*	3	3	2*
121	122	123	124	125	126	127	128	129	130	131	132
2*	2*	3	2*	2*	3	2*	2*	2*	3	3	3
133	134	135	136	137	138	139	140	141	142	143	144
2*	2*	2*	3	2*	2*	2*	2*	2*	2*	2*	2*
145	146	147	148	149	150						
2*	2*	2*	2*	2*	2*						

This resulted from analysis of the famous Fisher iris data set in `iris.tf1` when $K = 3$ subgroups were requested.

We note that groups 1 (setosa) and 2 (versicolor) contained the all the cases from the known classification, but most of the known group 3 (virginica) cases (those identified by asterisks) were also assigned to subgroup 2. This table should also be compared to a table resulting from K-means clustering analysis of the same data set.

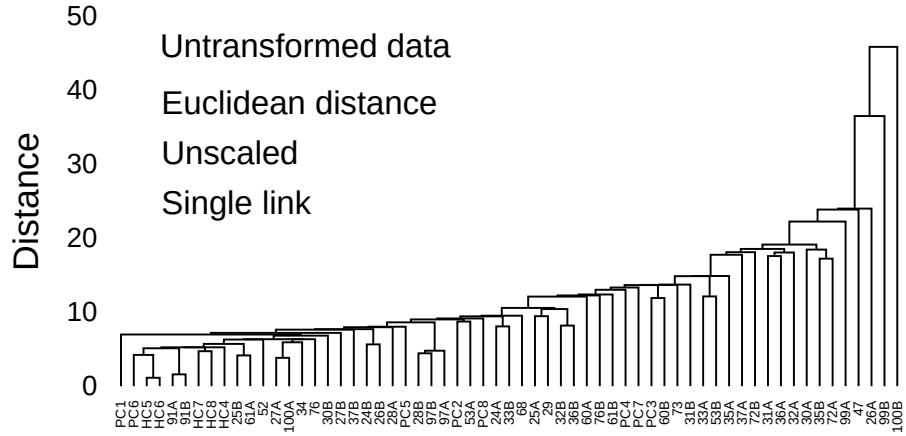
From the SIMFIT dendrogram partial clustering procedure it is also possible to create a SIMFIT MANOVA type file for any type of subsequent MANOVA analysis and, to aid in the use of dendrogram clusters as training sets for allocating new observations to groups, the subgroup centroids are also appended to such files. Alternatively a file ready for K-means cluster analysis can be saved, with group centroids appended to serve as starting estimates.

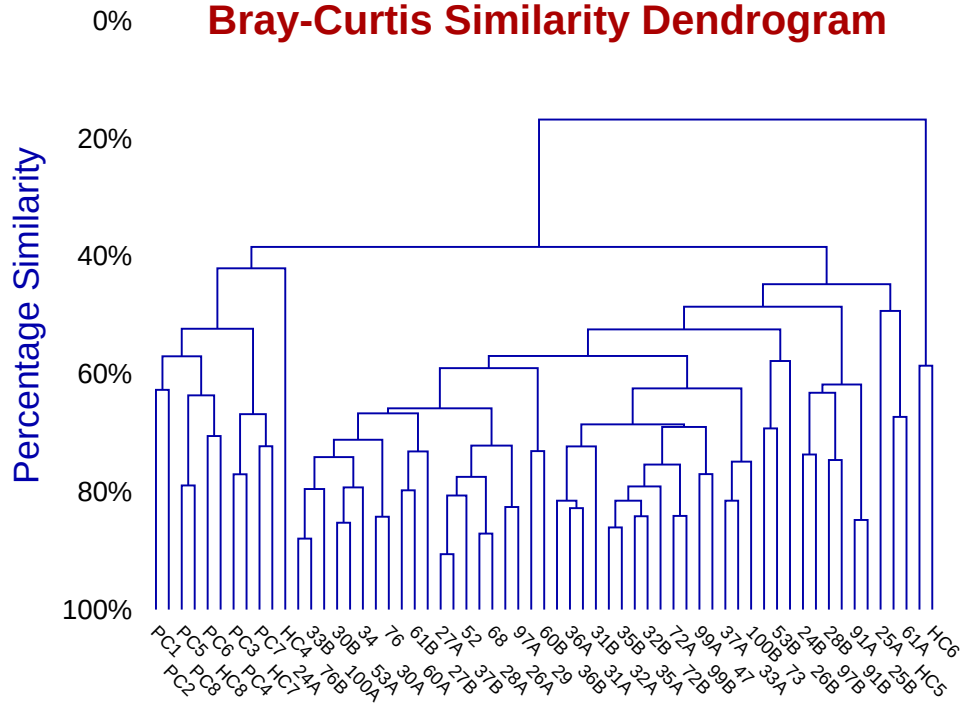
Finally, attention should be drawn to the advanced techniques provided by SIMFIT for plotting dendrogram thresholds and subgroups illustrated next.

Plotting dendrograms: standard format

Dendrogram shape is arbitrary in two ways; the x axis order is arbitrary as clusters can be rotated around any clustering distance leading to 2^{n-1} different orders, and the distance matrix depends on the settings used. For instance, a square root transformation, Bray-Curtis similarity, and a group average link generates the second dendrogram in this figure from the first. The data were contained in test file `cluster.tf2`, y plotted are dissimilarities, while labels are $100 - y$, which should be remembered when changing the y axis range.

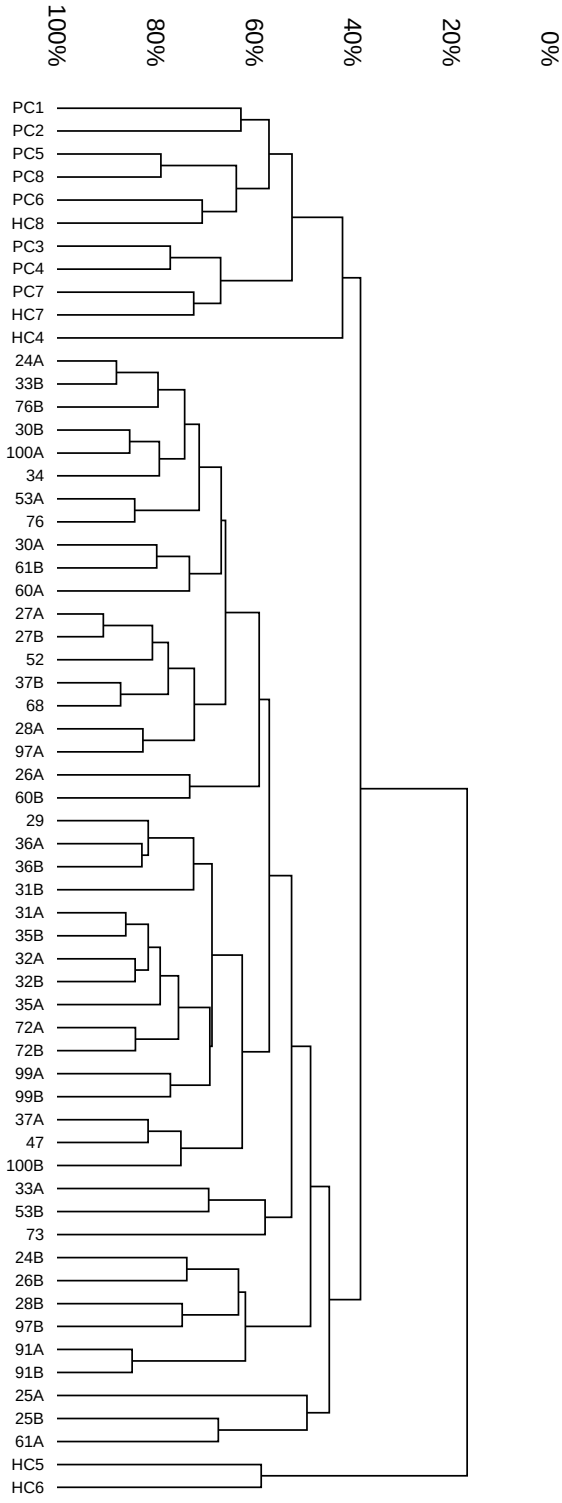
Users should not manipulate dendrogram parameters to create a dendrogram supporting some preconceived clustering scheme. You can set a label threshold and translation distance from the [X-axis] menu so that, if the number of labels exceeds the threshold, even numbered labels are translated, and font size is decreased.





Plotting dendrograms: stretched format

Sometimes dendrograms are more readable if the white space is stretched without distorting the labels. So SIMFIT PostScript graphs have a very useful feature: you can stretch or compress the white space between plotted lines and symbols without changing the line thickness, symbol size, or font size and aspect ratio. For instance, stretching, clipping and sliding procedures are valuable in graphs which are crowded due to overlapping symbols or labels, as in previous figures. If such dendrograms are stretched retrospectively using **editps**, the labels will not separate as the fonts will also be stretched so letters become ugly due to altered aspect ratios. SIMFIT can increase white space between symbols and labels while maintaining correct aspect ratios for the fonts in PostScript hardcopy and, to explain this, the creation of this figure using the data in `cluster.tf2` will be described.

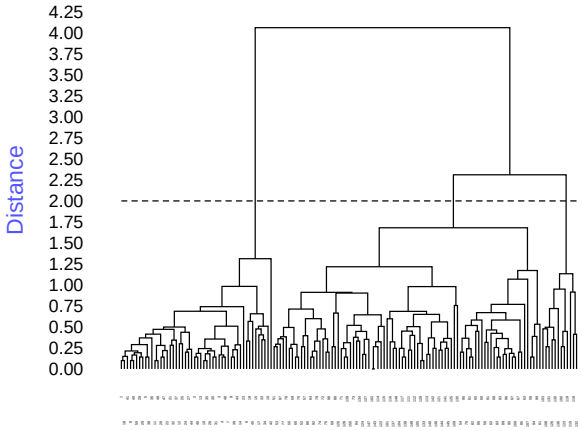


The title, legend and double x labeling were suppressed, and landscape mode with stretching, clipping and sliding was selected from the PostScript control using the [Shape] then [Landscape +] options, with an x stretching factor of two. Stretching increases the space between each symbol, or the start of each character string, arrow or other graphical object, but does not turn circles into ellipses or distort letters. As graphs are often stretched to print on several sheets of paper, sub-sections of the graph can be clipped out, then the clipped sub-sections can be slid to the start of the original coordinate system to facilitate printing.

If stretch factors greater than two are used, legends tend to become detached from axes, and empty white space round the graph increases. To remedy the former complication, the default legends should be suppressed or replaced by more closely positioned legends while, to cure the later effect, GSview can be used to calculate new BoundingBox coordinates (by transforming .ps to .eps). If you select the option to plot an opaque background even when white (by mistake), you may then find it necessary to edit the resulting .eps file in a text editor to adjust the clipping coordinates (identified by %#clip in the .eps file) and background polygon filling coordinates (identified by %#pf in the .ps file) to trim away unwanted white background borders that are ignored by GSview when calculating BoundingBox coordinates. Another example of this technique is with meta analysis plots, where it is also pointed out that creating transparent backgrounds by suppressing the painting of a white background obviates the need to clip away extraneous white space.

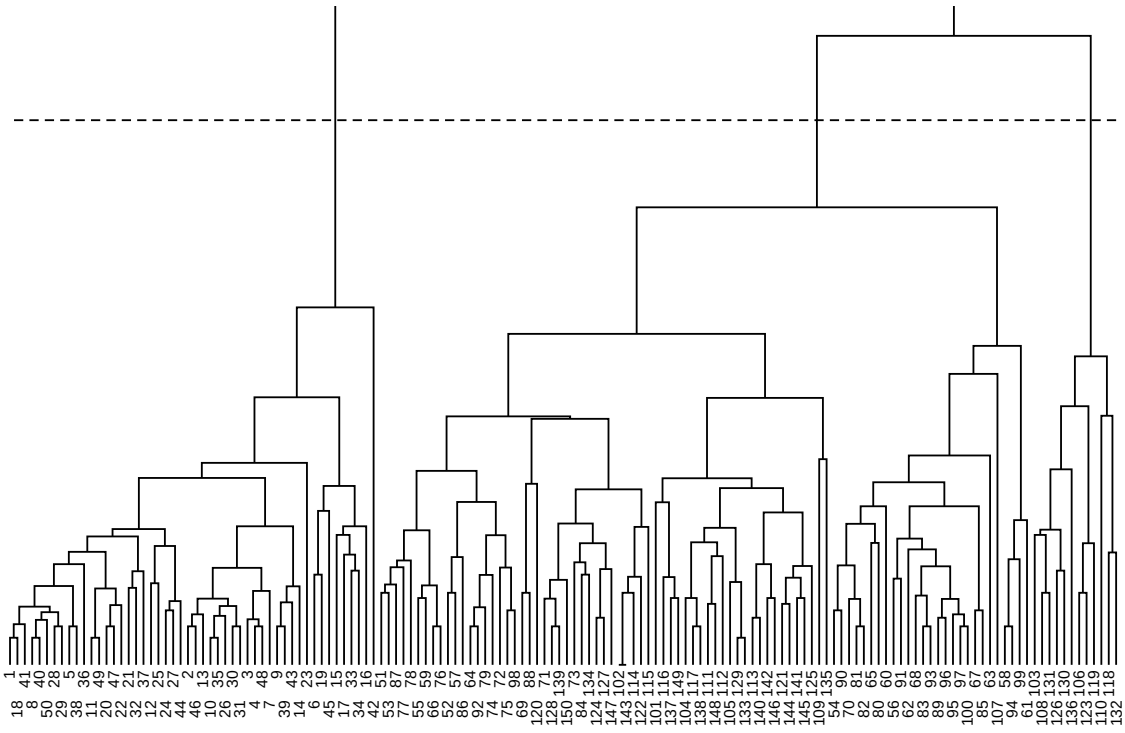
Plotting dendrograms: subgroups

The procedures described can also be used to improve the readability of dendrograms where subgroups have been assigned by partial clustering. The next figure shows a graph from `iris.tf1` when three subgroups are requested, or a threshold is set corresponding to the horizontal dotted line. The figure was created by these steps.



First the title was suppressed, the y-axis range was changed to $(0, 4.25)$ with 18 tick marks, the (x, y) offset was canceled as this suppresses axis moving, the label font size was increased from 1 to 3, and the x-axis was translated to 0.8.
Then the PostScript stretch/slide/clip procedure was used with these parameters

$$x_{\text{stretch}} = 1.5$$
$$y_{\text{stretch}} = 2.0$$
$$x_{\text{clip}} = 0.15, 0.95$$
$$y_{\text{clip}} = 0.10, 0.60.$$



Windows users without PostScript printing facilities must create a `*.eps` file using this technique, then use the SIMFIT procedures to create a graphics file they can use, e.g. `*.jpg`. Use of a larger font and increased x-stretching would be required to read the labels, of course.

6.3.3 Classical metric and non-metric (ordinal) scaling

Multi-dimensional scaling (MDS) provides various alternatives to dendrograms for visualizing distances between cases, so facilitating the recognition of potential groupings in a space of lower dimension than the number of variables. Given a n by m data set, the idea is to generate a set of n points in a Euclidean sub-space of dimension $1 < k << n - 1$ that have a distance matrix as close as possible to the distance matrix for the original data, so that distances can be visualized in the subspace for say $k = 2$, or $k = 3$. There are two cases.

1. **Classical metric scaling**
   This technique is used when the original data are in the form of observed quantities measured in terms of coordinates where distance is meaningful.

2. **Non-metric (ordinal) scaling**
   This technique is resorted to when the original data are of categorical or similar type that have been observed on a scale where only ranking is important and not actual differences.

SimFIT can perform classical metric and/or non-metric (ordinal) scaling using a distance matrix calculated interactively or by supplying a pre-calculated distance matrix.

From the main SimFIT menu choose [Statistics], [Multivariate], then [Scaling] using a distance matrix, read in the test file `g03faf.tf1`, and analyze using both metric and non-metric techniques to obtain the results as follows.

Eigenvalues from MDS (divided by the trace of the E matrix)
0.787130
0.280850
0.159630
0.077476
0.031624
0.020654
0.000000
-0.012186
-0.013685
-0.030479
-0.045469
-0.056206
-0.079207
-0.117400
[Sum 1 to 2]/[sum 1 to 13] = 0.9558 (95.58%) (actual values)
[Sum 1 to 2]/[sum 1 to 13] = 0.6709 (67.09%) (absolute values)
STRESS = 0.12557 (start = Metric 0%)
S-STRESS = 0.14962 (start = Metric 0%)

This table first lists the eigenvalues from classical metric scaling, where each eigenvalue has been normalized by dividing by the sum of all the eigenvalues, then the $STRESS$ and $SSTRESS$ values are listed.

Note that the type of starting estimates used, together with the percentages of the metric values used in any random starts, are output by SimFIT and it will be seen that, with this distance matrix, there are small but negative eigenvalues, and hence the proportion of the distances captured by the lower dimensional subspace would be inflated, and in addition two-dimensional plotting could be misleading. However it is usual to consider such small negative eigenvalues as being effectively zero, so that metric scaling in two dimensions is probably justified in this case as most of the proportion is in the first two eigenvalues.

The indication (actual values) is for the case where the sum of eigenvalues is used in the denominator when calculating the proportion P, i.e.

$$P = \sum_{i=1}^{k} \lambda_i / \sum_{i=1}^{n-1} \lambda_i,$$

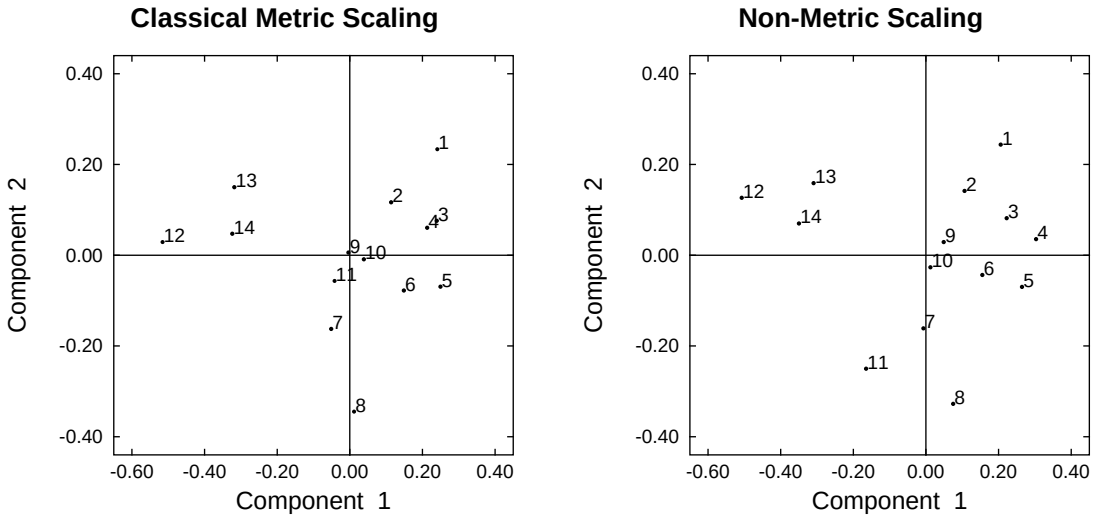
while the indication (absolute values) is for the case where the sum of the absolute values is used as the denominator, i.e.

$$P = \sum_{i=1}^{k} \lambda_i / \sum_{i=1}^{n-1} |\lambda_i|,$$

as discussed later.

In an ideal case all the eigenvalues would be positive and these two values would be the same. For this reason a warning is issued when negative eigenvalues are encountered to alert users that caution is required when regarding the subspace plot as a valid representation of the distance between cases.

The next figures confirm the validity of using metric scaling with these data by showing considerable agreement between the two dimensional plots from metric scaling, and also non-metric scaling involving the $STRESS$ calculation. Note that the default labels in such plots may be integers corresponding to the case numbers, and not case labels, but such plot labels can be edited interactively, or overwritten from a labels file if required.



Format for data input

The data required for scaling must either be in the form of a multivariate matrix from which a distance matrix is calculated interactively, then either used directly or saved to a file for retrospective analysis. Of course a distance matrix $D = d_{ij}$ from a n by m data matrix is a symmetric n by n matrix but, because the diagonals are zero and generally $d_{ij} = d_{ji}$, then only $n(n-1)/2$ differences need to be available. For that reason, distance matrices are stored and analyzed by SimFIT in strict lower triangular format as now described.

For instance, the data contained in test file `cluster.tf1` is the following 12 by 8 matrix

```
 1.0   4.0   2.0  11.0   6.0   4.0   3.0   9.0
 8.0   5.0   1.0  14.0  19.0   7.0  13.0  21.0
 3.0   1.0   3.0   1.0   3.0   6.0  23.0  37.0
 9.0   0.0   7.0   7.0   1.0   2.0  21.0   2.0
 7.0  12.0   9.0   5.0  14.0   9.0  12.0  14.0
 2.0  13.0  15.0   2.0  23.0   6.0  34.0   8.0
11.0   7.0   2.0   1.0   4.0  17.0  11.0   4.0
 6.0   3.0   7.0  12.0  11.0   8.0   8.0   0.0
 8.0  21.0   1.0  10.0  31.0   9.0   3.0  18.0
19.0  14.0  12.0   9.0  16.0  10.0   0.0  27.0
17.0  18.0  10.0   6.0  19.0  14.0   1.0  24.0
15.0  21.0   8.0   7.0  17.0  12.0   4.0  22.0
```

leading to the strict lower triangle of the 12 by 12 distance matrix below.

```
22.0
36.2  28.8
22.9  29.7  36.6
1.95  16.6  31.1  24.5
39.8  32.7  40.6  31.8  26.1
21.7  28.3  38.2  21.3  19.3  36.2
14.1  24.1  42.6  18.8  18.9  34.2  18.5
32.7  23.0  45.4  44.9  23.6  38.7  36.6  33.4
31.6  23.9  37.2  41.0  22.2  43.9  33.5  33.9  24.7
32.2  24.4  39.1  41.8  20.2  41.4  31.3  33.4  19.9  8.25
29.9  22.7  37.7  39.0  17.2  38.4  29.2  31.4  18.1  11.4  6.24
```

However this lower triangle would be stored packed by rows as follows

```
22.0
36.2
28.8
22.9
29.7
36.6
...
6.24
```

and distance matrices supplied for analysis by SIMFIT must be formatted in this way.

Theory for metric scaling

For instance, once a distance matrix $D = (d_{ij})$ has been calculated for n cases with m variables, as described for dendrograms, it may be possible to calculate principal coordinates. This involves constructing a matrix E defined by

$$e_{ij} = -\tfrac{1}{2}(d_{ij}^2 - d_{i.}^2 - d_{.j}^2 + d_{..}^2),$$

where $d_{i.}^2$ is the average of d_{ij}^2 over the suffix j, etc., in the usual way. The idea is to choose an integer k, where $1 < k << n - 1$, so that the data can be represented approximately in a space of dimension less than the number of cases, but in such a way that the distance between the points in that space correspond to the distances represented by the d_{ij} of the distance matrix as far as possible. If E is positive semi-definite, then the ordered eigenvalues $\lambda_i > 0$ of E will be nonnegative and the proportionality expression

$$P = \sum_{i=1}^{k} \lambda_i \Big/ \sum_{i=1}^{n-1} \lambda_i$$

will show how well the cases of dimension n are represented in this subspace of dimension k. The most useful case is when $k = 2$, or $k = 3$, and the d_{ij} satisfy

$$d_{ij} \leq d_{ik} + d_{jk},$$

so that a two or three dimensional plot will display distances corresponding to the d_{ij}.

If this analysis is carried out but some relatively large negative eigenvalues result, then the proportion P may not adequately represent the success in capturing the values in distance matrix in a subspace of lower dimension that can be plotted meaningfully.

It should be pointed out that the principal coordinates will actually be the same as the principal components scores when the distance matrix is based on Euclidean norms. Further, where metrical scaling succeeds, the distances between points plotted in say two or three dimensions will obey the triangle inequality and so correspond reasonably closely to the distances in the dissimilarity matrix, but if it fails it could be useful to proceed to non-metrical scaling, which is discussed next.

Theory for non-metric (ordinal) scaling

Often a distance matrix is calculated where some or all of the variables are ordinal, so that only the relative order is important, not the actual distance measure. Non-metric (i.e. ordinal) scaling is similar to the metric scaling previously discussed, except that the representation in a space of dimension $1 < k << n - 1$ is sought in such a way as to attempt to preserve the relative orders, but not the actual distances. The closeness of a fitted distance matrix to the observed distance matrix can be estimated as either $STRESS$, or $SSTRESS$, given by

$$STRESS = \sqrt{\frac{\sum_{i=1}^{n}\sum_{j=1}^{i-1}(\hat{d}_{ij} - \tilde{d}_{ij})^2}{\sum_{i=1}^{n}\sum_{j=1}^{i-1}\hat{d}_{ij}^2}}$$

$$SSTRESS = \sqrt{\frac{\sum_{i=1}^{n}\sum_{j=1}^{i-1}(\hat{d}_{ij}^2 - \tilde{d}_{ij}^2)^2}{\sum_{i=1}^{n}\sum_{j=1}^{i-1}\hat{d}_{ij}^4}},$$

where $\hat{d}_{ij}$ is the Euclidean squared distance between points i and j, and $\tilde{d}_{ij}$ is the fitted distance when the $\hat{d}_{ij}$ are monotonically regressed on the d_{ij}. This means that $\tilde{d}_{ij}$ is monotonic relative to d_{ij} and is obtained from $\hat{d}_{ij}$ with the smallest number of changes.

It should be noted that this is a nonlinear optimization problem which may depend critically on starting estimates, and so can only be relied upon to locate a local, not a global solution. For this reason, starting estimates can be obtained in SimFIT by a preliminary metric scaling, or alternatively the values from such a scaling can be randomly perturbed before the optimization, in order to explore possible alternative solution points.

As mentioned previously, SimFIT can save distance matrices to files, so that dendrogram creation, classical metric, and non-metric scaling can be carried out retrospectively, without the need to generate distance matrices repeatedly from multivariate data matrices. Such distance matrices will be stored as vectors, corresponding to the strict lower triangle of the distance matrix packed by rows, (i.e. the strict upper triangle packed by columns).

6.3.4 K-means clustering

Given a swarm of n multivariate points, K-means clustering attempts to assign these data into K non-empty clusters where $1 < K < n$. The clusters are created by assigning cases to those groups that minimize the within-cluster sum of squared distances of the data from the means of the clusters. Starting values from which to commence the iterations to find such clusters must be provided.

Example 1

From the SIMFIT main menu choose [Statistics], [Multivariate], then [K-means clustering], and observe the format for the test data contained in `g03eff.tf1` which are observations of five variables on twenty soil types as follows.

```
77.3   13.0    9.7   1.5   6.4
82.5   10.0    7.5   1.5   6.5
66.9   20.6   12.5   2.3   7.0
47.2   33.8   19.0   2.8   5.8
65.3   20.5   14.2   1.9   6.9
83.3   10.0    6.7   2.2   7.0
81.6   12.7    5.7   2.9   6.7
47.8   36.5   15.7   2.3   7.2
48.6   37.1   14.3   2.1   7.2
61.6   25.5   12.9   1.9   7.3
58.6   26.5   14.9   2.4   6.7
69.3   22.3    8.4   4.0   7.0
61.8   30.8    7.4   2.7   6.4
67.7   25.3    7.0   4.8   7.3
57.2   31.2   11.6   2.4   6.5
67.2   22.7   10.1   3.3   6.2
59.2   31.2    9.6   2.4   6.0
80.2   13.2    6.6   2.0   5.8
82.2   11.1    6.7   2.2   7.2
69.7   20.7    9.6   3.1   5.9
begin{values}
82.5   10.0    7.5   1.5   6.5
47.8   36.5   15.7   2.3   7.2
67.2   22.7   10.1   3.3   6.2
end{values}
```

Note that starting cluster coordinates are appended to this data set in the section identified by

begin{values} . . . end{values}

as this is the most convenient way to perform K-means clustering. However, these can be supplied independently, e.g. in a file like `g03eff.tf2`, or generated randomly. Note that K-means clustering is an iterative technique and the outcome will depend on the starting clusters.

From the analysis the following results are displayed, where for each case (in the odd-numbered rows) the cluster number to which it is assigned is the corresponding figure below it (in the even-numbered rows).

Results for K-means clustering with `g03eff.tf1`
Variables included: 1 2 3 4 5
Number of clusters K = 3
Transformation: Untransformed
Weighting: Unweighted for replicates
Cases (odd rows) and Clusters (even rows)

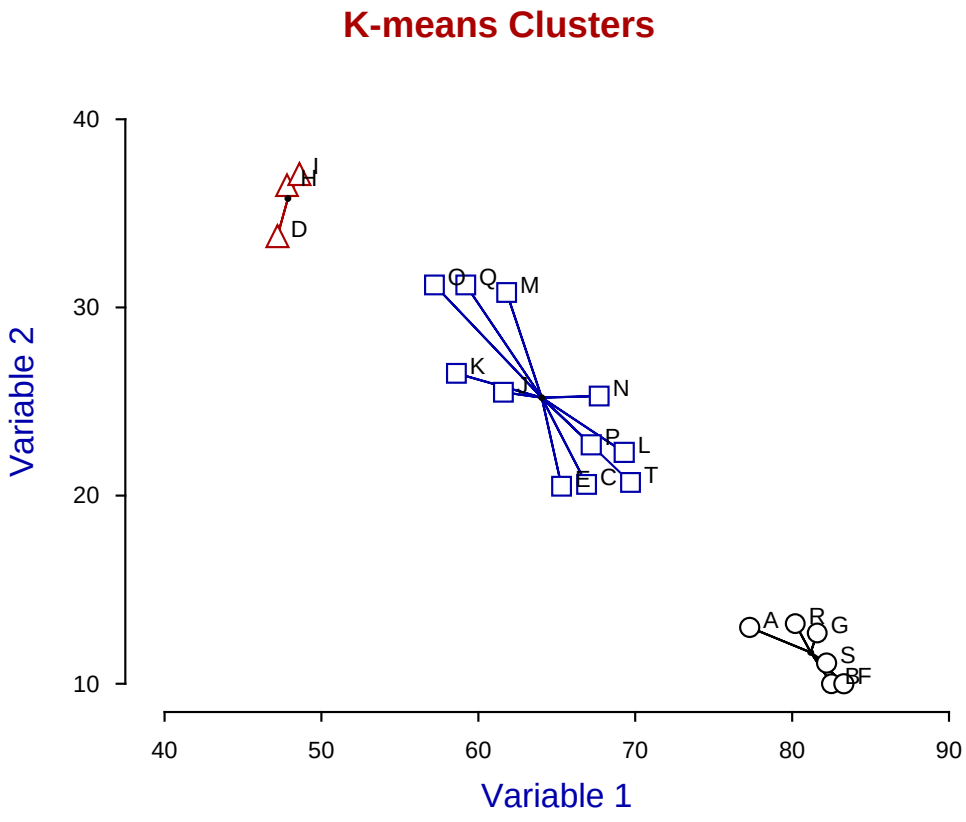
1	2	3	4	5	6	7	8	9	10	11	12
1	1	3	2	3	1	1	2	2	3	3	3
13	14	15	16	17	18	19	20				
3	3	3	3	3	1	1	3				

Final cluster centroids

81.183	11.667	7.1500	2.0500	6.6000
47.867	35.800	16.333	2.4000	6.7333
64.045	25.209	10.745	2.8364	6.6545

Note that the final cluster centroids minimizing the objective function, given the starting estimates supplied, are calculated, and the cases are assigned to these final clusters.

Plots of the clusters and final cluster centroids can be created as in the next figure for variables x_1 and x_2, with the optional labels as these were also supplied on the data file `g03eff.tf1` (as for dendrograms).



With two dimensional data representing actual distances, outline maps can be added and other special effects can be created, as shown later. Further, techniques are provided to perturb the default positions of labels if this is required in order to clarify the labeling.

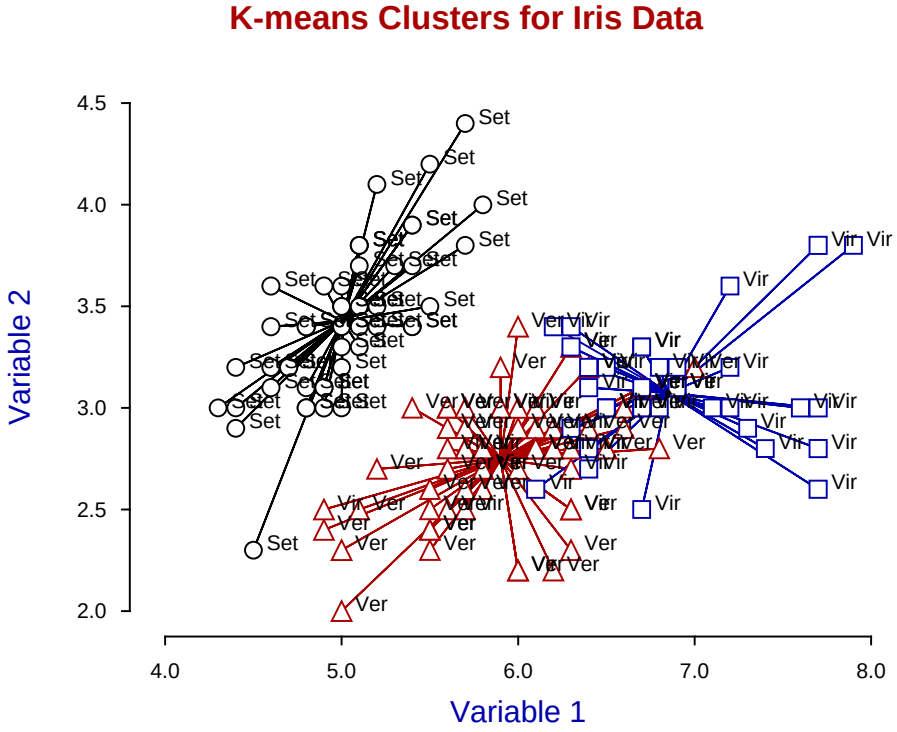
Example 2

The next table is for analysis of the Fisher Iris data set in `iris.tf1`, using starting clusters in `iris.tf2`.

```
1  1  1  1  1  1  1  1  1  1  1  1
1  1  1  1  1  1  1  1  1  1  1  1
1  1  1  1  1  1  1  1  1  1  1  1
1  1  1  1  1  1  1  1  1  1  1  1
1  1  2  2  3* 2  2  2  2  2  2  2
2  2  2  2  2  2  2  2  2  2  2  2
2  2  2  2  2  3* 2  2  2  2  2  2
2  2  2  2  2  2  2  2  2  2  2  2
2  2  2  2  3  2* 3  3  3  3  2* 3
3  3  3  3  3  2* 2* 3  3  3  3  2*
3  2* 3  2* 3  3  2* 2* 3  3  3  3
3  2  3  3  3  3  2* 3  3  3  2* 3
3  3  2* 3  3  2*
```

```
Cluster     Size     WSSQ
   1          50     15.15
   2          62     39.82
   3          38     23.88
Final cluster centroids
5.0060   3.4280   1.4620   0.2460
5.9016   2.7484   4.3935   1.4339
6.8500   3.0737   5.7421   2.0711
```

The data were maintained in the known group order (as in `manova1.tf5`), and the clusters assigned are seen to be identical to the known classification for group 1 (setosa), while limited misclassification has occurred for groups 2 (versicolor, 2 assigned to group 3), and 3 (viginica, 14 assigned to group 2), as shown by the starred values. Clearly group 1 is distinct from groups 2 and 3 which show some similarities to each other, a conclusion also illustrated in the next figure.

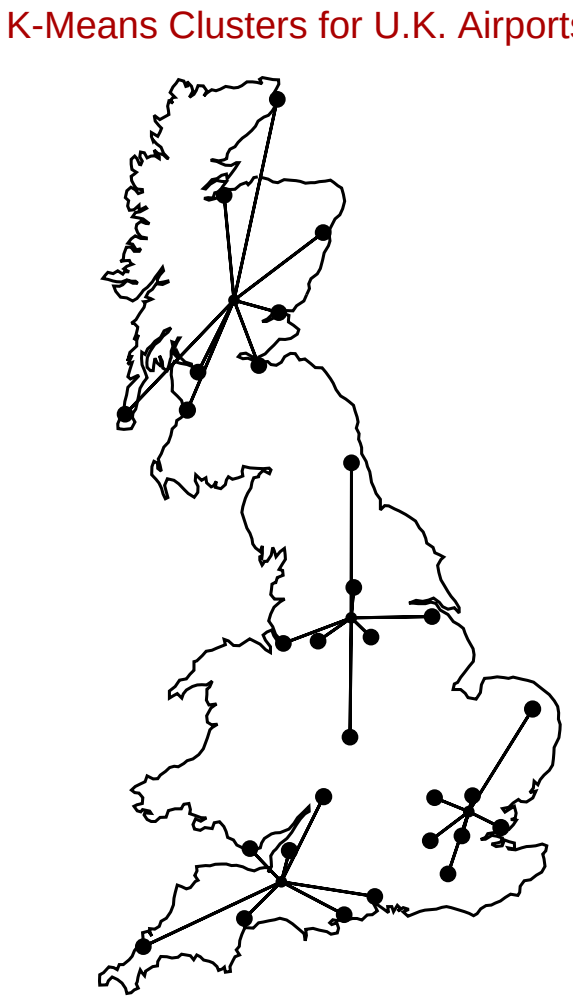


Example 3

This example explains how to include additional features such as maps which are often added to plots to emphasize the meaning of clusters.

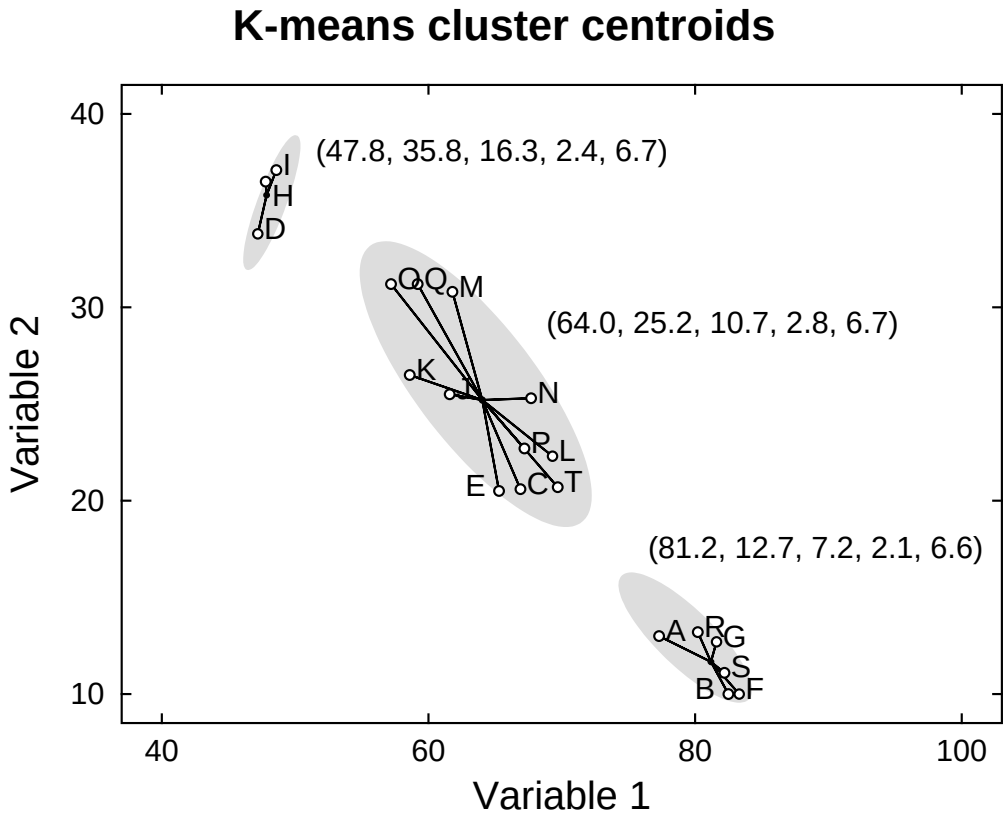
Stretching and clipping are also valuable when graphs have to be re-sized to achieve geometrically correct aspect ratios, as in the map shown in this next figure, which can be generated by the K-means clustering procedure using program **simstat** as follows.

- Input `ukmap.tf1` with coordinates for UK airports.
- Input `ukmap.tf2` with coordinates for starting centroids.
- Calculate centroids then transfer the plot to advanced graphics.
- Read in the UK coastal outline coordinates as an extra file from `ukmap.tf3`.
- Suppress axes, labels, and legends, then clip away extraneous white space.
- Stretch the PS output using the [Shape] then [Portrait +] options, and save the stretched eps file.



Example 4

It is frequently useful to be able highlight groups of data points in a two dimensional swarm, as in this figure.



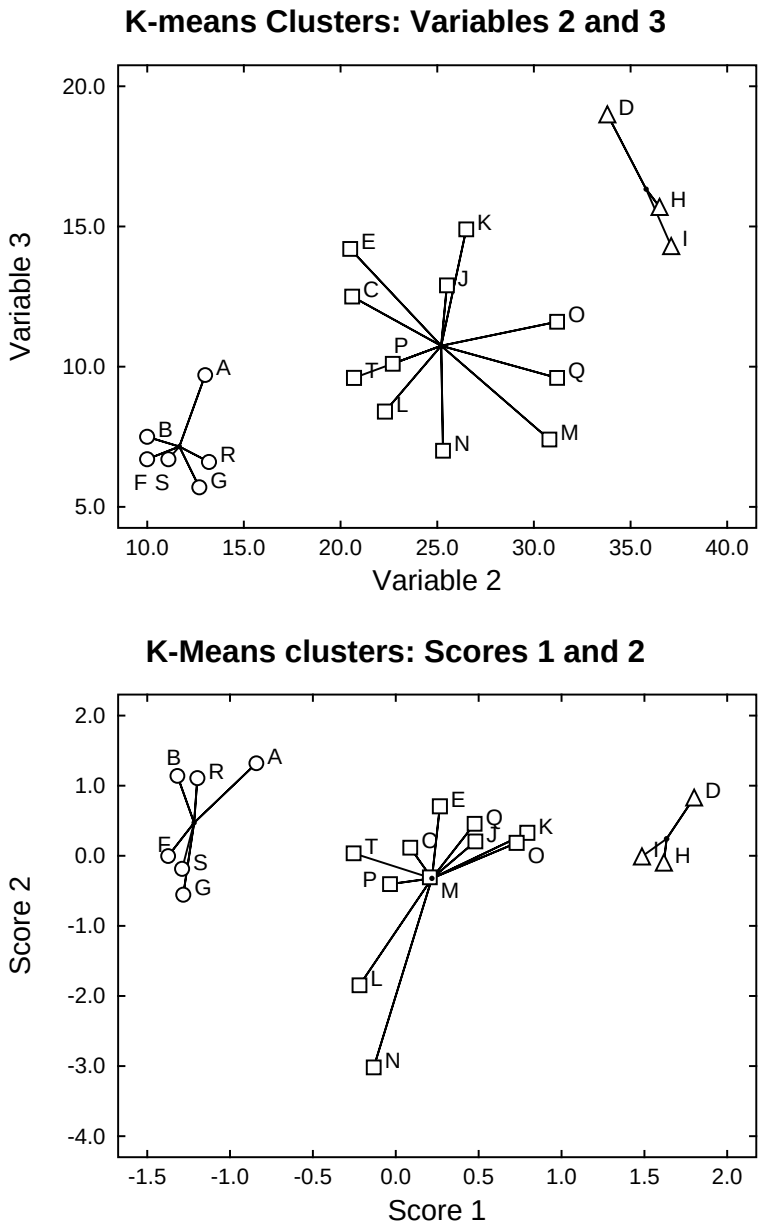
In this case a partition into three groups has been done by K-means clustering, and to appreciate how to use this technique, note that this figure can be generated by the K-means clustering procedure using program **simstat** as follows.

- Input the K-means clustering test file `kmeans.tf1`.
- Calculate the centroids, using the starting estimates appended to the test file. View them, which then adds them to the results file, then record the centroid coordinates from the results file.
- Select to plot the groups with associated labels, but then it will prove necessary to move several of the labels by substituting new labels, or shifting the *x* or *y* coordinates to clarify the graph.
- Add the solid background ellipses using the lines/arrows/boxes option because both head and tail coordinate must be specified using the red arrow, as well as an eccentricity value for the ellipses. Of course, any filled shapes such as circles, squares, or triangles can be chosen, and any size or color can be used.
- Add the centroid coordinates as extra text strings.

Of course, this technique can be used to highlight or draw attention to any subsets of data points, for instance groups in principal component analysis.

Example 5

This example considers the plotting of principal component scores instead of original variables, illustrated in the next figure for `kmeans.tf1`.



Note that, in the upper figure, symbols F, S, and P have been translated for clarity, and it should be compared to an earlier figure, for the same data with variables 1 and 2. This highlights an important point when plotting clusters for more than 2 variables: the plot shape depends on the variables chosen. So, for a more representative plot when there are more than 2 variables it is better to plot principal component scores instead of variables. The SIMFIT default is to plot the scores obtained using the correlation matrix technique, as this can prevent the analysis being dominated by columns with unduly large values.

In the lower figure, symbols B, O, M, and P have been translated for clarity, but now the principal component scores 1 and 2 have been plotted, which will usually be a better representation of the clustering, as the shape of the plot is not so strongly influenced by the variables chosen.

Theory

Once a n by m matrix of values a_{ij} for n cases and m variables has been provided, the cases can be sub-divided into K non-empty clusters where $K < n$, provided that a K by m matrix of starting estimates b_{ij} has been specified. The procedure is iterative, and proceeds by moving objects between clusters to minimize the objective function

$$\sum_{k=1}^{K} \sum_{i \in S_k} \sum_{j=1}^{m} w_i (a_{ij} - \bar{a}_{kj})^2$$

where S_k is the set of objects in cluster k and $\bar{a}_{kj}$ is the weighted sample mean for variable j in cluster k. The weighting factors w_i can allow for situations where the objects may not be of equal value, e.g., if replicates have been used to determine the a_{ij}.

Certain other aspects of the SIMFIT implementation of K-means clustering should be made clear.

1. If variables differ greatly in magnitude, data should be transformed before cluster analysis but note that, if this is done interactively, the same transformation will be applied to the starting clusters. If a transformation cannot be applied to data, clustering will not be allowed at all, but if a starting estimate cannot be transformed (e.g., square root of a negative number), then that particular value will remain untransformed.

2. If, after initial assignment of data to the starting clusters some are empty, clustering will not start, and a warning will be issued to decrease the number of clusters requested, or edit the starting clusters.

3. Clustering is an iterative procedure, and different starting clusters may lead to different final cluster assignments. So, to explore the stability of a cluster assignment, you can perturb the starting clusters by adding or multiplying by a random factor, or you can even generate a completely random starting set. For instance, if the data have been normalized to zero mean and unit variance, then choosing uniform random starting clusters from $U(-1, 1)$, or normally distributed values from $N(0, 1)$ might be considered.

4. After clusters have been assigned you may wish to pursue further analysis, say using the groups for canonical variate analysis, or as training sets for allocation of new observations to groups. To do this, you can create a SIMFIT MANOVA type file with group indicator in column 1. Such files also have the centroids appended, and these can be overwritten by new observations (not forgetting to edit the extra line counter following the last line of data) for allocating to the groups as training sets.

5. If weighting, variable suppression, or interactive transformation is used when assigning K-means clusters, all results tables, plots and MANOVA type files will be expressed in coordinates of the transformed space.

6. When viewing two dimensional plots of clusters where there are more than two variables, users can choose which coordinates to display, and this can give a misleading impression where it can seem that some cases have been wrongly assigned. This is to forget that the assignment is based on a selection process that uses all of the variables, and is a good reason to view using several pairs of coordinates to get a better overall picture, or to plot using principal components.

7. When displaying clusters as principal components, the loadings used to plot scores for data and centroids are calculated interactively from the data correlation matrix and standardized for unit variance. The scores are not used for further iterations to refine the clustering procedure.

6.4 Multivariate projection and display techniques

Tutorials and worked examples for simulation, curve fitting, statistical analysis, and plotting.
https://simfit.uk
https://simfit.org.uk
https://simfit.silverfrost.com

6.4.1 Principal components

Principal component analysis attempts to express a n by m data set with $m > 2$ in new coordinates Y obtained by rotating the original coordinates X so that the overall variance of the observations is contained in decreasing order in the new variables.

If this is successful in that most of the variance is contained in the first 2 or 3 of the Y variables then this allows inferences to be drawn about the data in a sub-space of dimension $< m$.

Example 1

From the main SIMFIT menus choose [Statistics], [Multivariate], then [Principal components] and analyze the default test file provided (`g03aaf.tf1`) which contains the following data

7	4	3
4	1	8
6	3	5
8	6	1
8	5	7
7	2	9
5	3	3
9	5	8
7	4	5
8	2	2

leading to these results.

Variables included: 1 2 3
Transformation: Untransformed
Matrix type: Variance-covariance matrix
Score type: Score variance = eigenvalue
Replicates: Unweighted for replicates

Eigenvalues	Proportion	Cumulative	χ^2	DOF	p
8.274	0.6515	0.6515	8.613	5	0.1255
3.676	0.2895	0.9410	4.118	2	0.1276
0.750	0.0590	1.0000	0.000	0	0.0000

Loadings (by column)

-0.138	0.699	0.702
-0.250	0.661	-0.707
0.958	0.273	-0.084

Scores (by column)

-2.150	-0.173	-0.107
3.800	-2.890	-0.510
0.153	-0.987	-0.269
-4.710	1.300	-0.652
1.290	2.280	-0.449
4.100	0.144	0.803
-1.630	-2.230	-0.803
2.110	3.250	0.168
-0.235	0.373	-0.275
-2.750	-1.070	2.090

The significance of the options used for the analysis and the results listed in this table are now explained.

- **Variables included**
  All three variables were included as this was defined in the trailer section of `g03aaf.tf1`, but the variables to be included can also be adjusted interactively.
- **Transformation**
  The data were used without any transformation.
- **Matrix type**
  If the magnitude of the variables are similar so that the data do not need to be centralized and scaled, then the covariance matrix can be used. Otherwise the correlation matrix should be used.
- **Score type**
  Several options are available, to provide consistency and facilitate comparison with published data.
- **Replicates**
  SIMFIT provides the facility to supply a weighting vector to permit data suppression (setting a weight to zero), or to allow for replicates (setting a weight equal to the number of replicates used in the observation).
- **Eigenvalues**
  These are listed in decreasing order, the proportion of variance and cumulative sum of variances captured by each component is listed, and a chi-square test is performed to check the significance of each component. The significance levels are not valid if the correlation matrix is used instead of the covariance matrix. Clearly the first two principal components are sufficient to represent the three original variables.
- **Loading**
  Column j of the loading matrix contains the coefficients required to express y_j as linear function of the variables $x_1, x_2, \ldots, x_m$. The values can be used to indicate the importance of the contribution of the original variables to the rotated variables.
- **Scores**
  Row i of the scores matrix contains the values for row i of the original data expressed in variables $y_1, y_2, \ldots, y_m$. If most of the variance can be explained by the first two or three variables, the values can be used instead of the original observations to visualize grouping and clustering, etc.

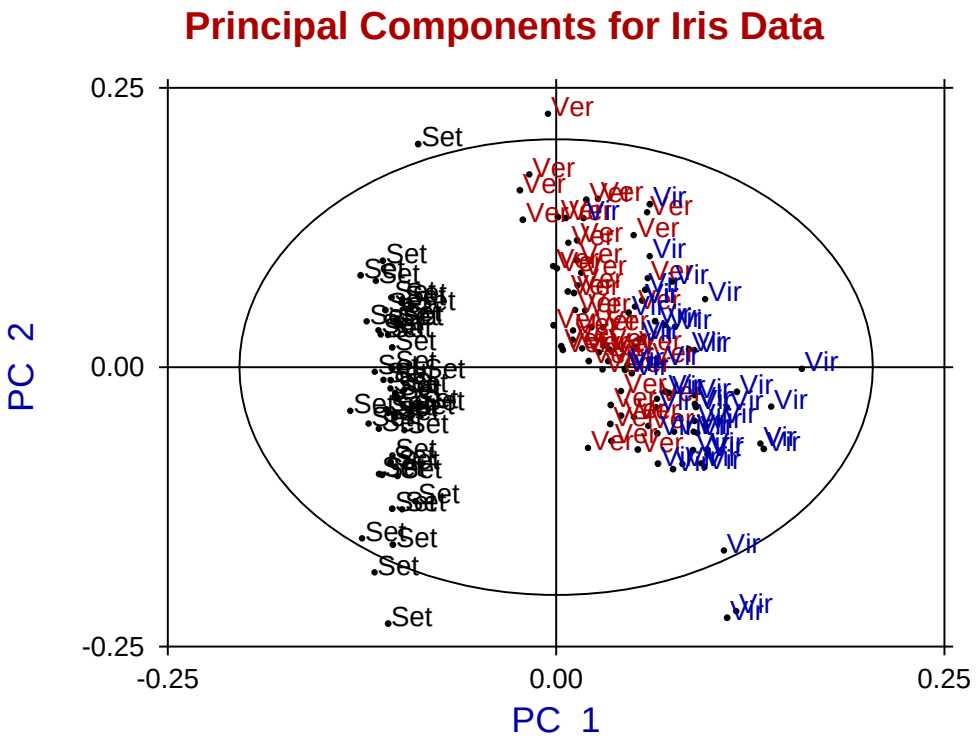
Example 2

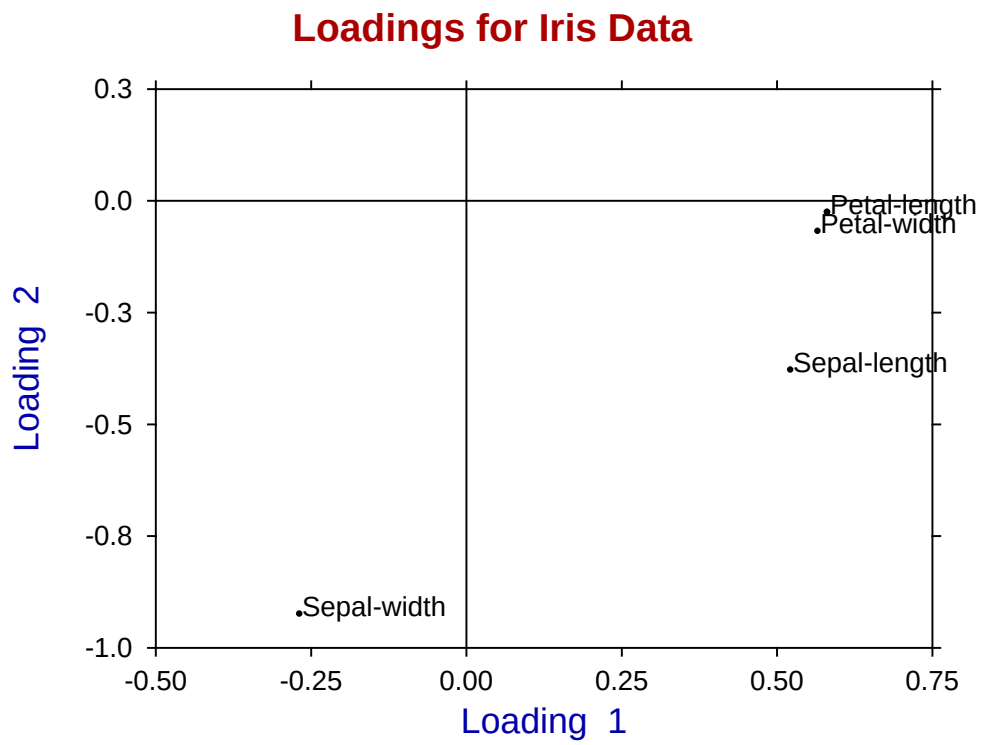
The next example concerns the analysis of the Fisher iris data with 150 cases and 4 variables (Sepal length, Sepal width, Petal length, and Petal width) contained in the test file `iris.tf1`. The figures below show the scores and loadings for these data after analyzing the correlation matrix.

The score plot displays the score components for all samples using the selected principal components, so some may prefer to label the legends as principal components instead of scores, and this plot is used to search for possible groupings among the sample. The components can be labeled using any labels supplied at the end of the data file, but this can cause confusion where, as in the present case, the labels overlap leading to crowding. A method for moving labels to avoid such confusion is provided. However, with such dense labels it is best to just plot the scores using different symbols and colors for the three groups.

The loading plot displays the coefficients that express the selected principal components y_j as linear functions of the original variables $x_1, x_2, \ldots, x_m$, so this plot is used to observe the contributions of the original variables x to the new ones y.

Note that figures also illustrate an application of the SIMFIT technique for adding extra data interactively to create the cross-hairs intersecting at $(0, 0)$, and it also shows how labels can be added to identify the variables in a loadings plot. It should be noted that, as the eigenvectors are of indeterminate sign and only the relative magnitudes of coefficients are important, the scattergrams can be plotted with either the scores calculated from the SVD, or else with the scores multiplied by minus one, which is equivalent to reversing the direction of the corresponding axis in a scores or loadings plot.





The confidence ellipse plotted on the scores will be explained later.

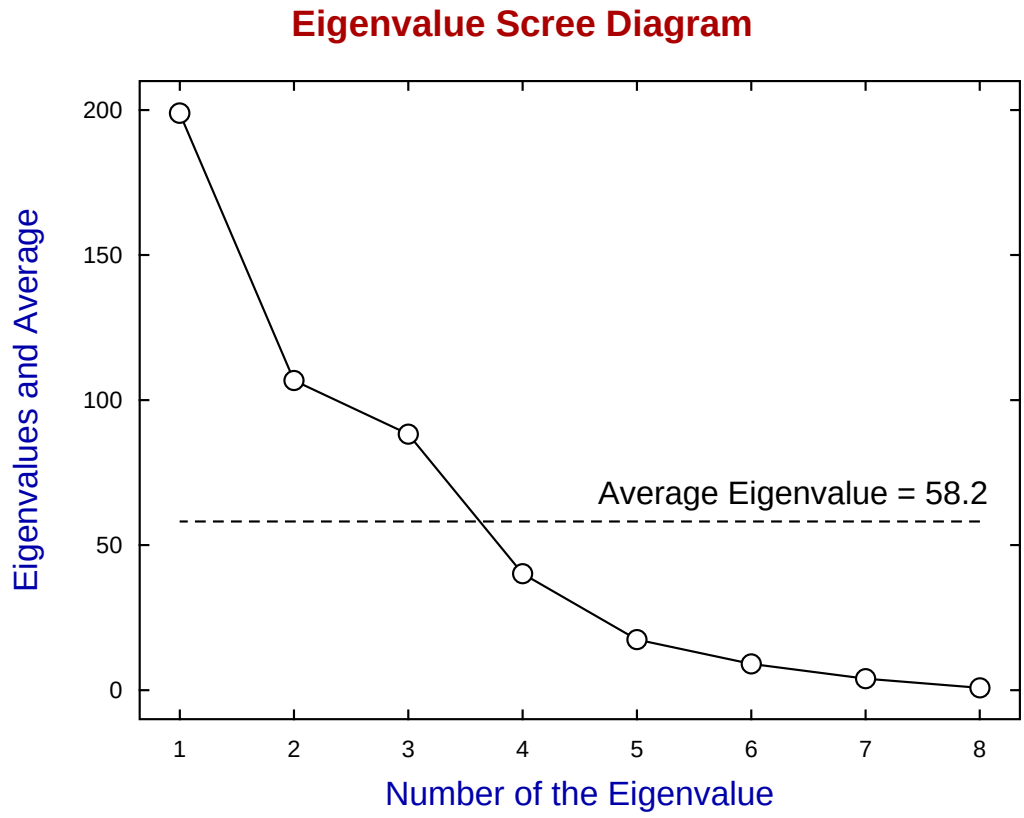
Example 3

An important topic in principal component analysis is deciding how to choose a sufficient number of principal components to represent the data adequately. As the eigenvalues are proportional to the fractions of variance along the principal component axes, a table of the cumulative proportions is calculated, and some users may find it useful to include sufficient principal components to account for a given amount of the variance, say 70%. Consider these results from the analysis of data with eight variables contained in test file `cluster.tf1` and analyzed using the covariance matrix.

Eigenvalues	Proportion	Cumulative	χ^2	DOF	p
198.9	0.4274	0.4274	61.13	35	0.0041
106.8	0.2294	0.6568	48.09	27	0.0075
88.29	0.1897	0.8465	39.73	20	0.0054
40.18	0.0863	0.9328	25.39	14	0.0309
17.46	0.0375	0.9703	15.04	9	0.0898
9.036	0.0194	0.9898	9.149	5	0.1033
3.966	0.0085	0.9983	4.349	2	0.1137
0.803	0.0017	1.0000	0.000	0	0.0000

The next figure shows how scree plots can be displayed to illustrate the number of components needed to represent the data adequately.

For instance, in this case, it seems that approximately three of the principal components are required. A useful rule of thumb for selecting the minimum number of components is to observe where the scree diagram crosses the average eigenvalue or becomes flattened indicating that all subsequent eigenvalues contribute to a comparable extent. Use of the chi-square statistics for this type of investigation will be described later.



Theory 1: The calculation of principal components

In the principal components analysis of a n by m data matrix, new coordinates y are selected by rotation of the original coordinates x so that the proportion of the variance projected onto the new axes decreases in the order $y_1, y_2, \ldots, y_m$. The hope is that most of the variance can be accounted for by a subset of the data in y coordinates, so reducing the number of dimensions required for data analysis.

It is usual to scale the original data so that the variables are all of comparable dimensions and have similar variances, otherwise the analysis will be dominated by variables with large values. Basing principal components analysis on the correlation matrix rather than the covariance or sum of squares and cross product matrices is often recommended as it also prevents the analysis being unduly dominated by variables with large values. The data format for principal components analysis is exactly the same as for cluster analysis; namely a data matrix with n rows (cases) and m columns (variables).

If the data matrix is X with covariance, correlation or scaled sum of squares and cross products matrix S, then the quadratic form

$$a_1^T S a_1$$

is maximized subject to the normalization $a_1^T a_1 = 1$ to give the first principal component

$$c_1 = \sum_{i=1}^{m} a_{1i} x_i.$$

Similarly, the quadratic form

$$a_2^T S a_2$$

is maximized, subject to the normalization and orthogonality conditions $a_2^T a_2 = 1$ and $a_2^T a_1 = 0$, to give the second principal component

$$c_2 = \sum_{i=1}^{m} a_{2i} x_i$$

and so on. The vectors a_i are the eigenvectors of S with eigenvalues λ_i^2, where the proportion of the variation accounted for by the ith principal component can be estimated as

$$\lambda_i^2 / \sum_{j=1}^{m} \lambda_j^2.$$

Actually SIMFIT uses a singular value decomposition (SVD) of a centered and scaled data matrix, say $X_s = (X - \bar{X})/\sqrt{(n-1)}$ as in

$$X_s = V \Lambda P^T$$

to obtain the diagonal matrix Λ of singular values, the matrix of left singular vectors V as the n by m matrix of scores, and the matrix of right singular vectors P as the m by m matrix of loadings.

Theory 2: Confidence ellipses in scores plots

Note that a 95% confidence Hotelling T^2 ellipse is also plotted, which assumes a multivariate normal distribution for the original data and uses the F distribution.

The confidence ellipse is based on the fact that, if $\bar{y}$ and S are the estimated mean vector and covariance matrix from a sample of size n and, if x is a further independent sample from an assumed $p-$variate normal distribution, then

$$(x - \bar{y})^T S^{-1} (x - \bar{y}) \sim \frac{p(n^2 - 1)}{n(n - p)} F_{p,n-p},$$

where the significance level for the confidence region can be altered interactively.

Theory 3: The chi-square test for significant components

In cases where the correlation matrix is not used, a chi-square test statistic is also provided along with appropriate probability estimates to make the decision more objective. In this case, if k principal components are selected, the chi-square statistic

$$(n-1-(2m+5)/6)\left\{-\sum_{i=k+1}^{m}\log(\lambda_i^2)+(m-k)\log\left(\sum_{i=k+1}^{m}\lambda_i^2/(m-k)\right)\right\}$$

with $(m-k-1)(m-k+2)/2$ degrees of freedom can be used to test for the equality of the remaining $m-k$ eigenvalues.

If one of these test statistics, say the $k+1$th, is not significant then it is usual to assume k principal components should be retained and the rest regarded as of little importance. So, if it is concluded that the remaining eigenvalues are of comparable importance, then a decision has to be made whether to eliminate all or preserve all. For instance, from the last column of p values referring to the above chi-square test for `g03aaf.tf1`, it might be concluded that a minimum of two components are required to represent this data set adequately. However, for the case of `iris.tf1`, three components would be required.

The common practise of always using two or three components just because these can be visualized is to be deplored.

Theory 4: Calculating scores from loadings

The data used by SIMFIT are automatically centered at run time, and sometimes also scaled if requested, so it is not usually necessary to transform the original data for principal component analysis, especially if the correlation matrix method is used. However, in order to calculate scores using the loadings retrospectively the following points should be noted.

1. The original data matrix must be centralized by subtracting column sample means.
2. If the correlation matrix technique was used to calculate the scores, then the data must also be scaled by dividing columns by the column sample standard deviations.
3. If the covariance matrix technique was used no further scaling is required.
4. If the sum of squares and cross-product matrix method was used, then the centralized data must also be multiplied by $\sqrt{n-1}$.
5. The final scaling of the scores will be that used when generating the loadings.
6. The average of a group of k scores is the same as using loadings with the means from the same k values.
7. The scores are unspecified up to multiples of -1.

To illustrate this procedure consider the following steps that are required to calculate the scores for a covariance matrix with scores normalized to have variance equal to the corresponding eigenvalue, using the notation for subroutine `g03aaf` in the NAG library documentation.

- Obtain the data matrix X
- Transform X to obtain the centered matrix Y
- Generate the loading matrix P
- Calculate the scores $V = YP$ as shown next.

$$X = \begin{pmatrix} 7.0 & 4.0 & 3.0 \\ 4.0 & 1.0 & 8.0 \\ 6.0 & 3.0 & 5.0 \\ 8.0 & 6.0 & 1.0 \\ 8.0 & 5.0 & 7.0 \\ 7.0 & 2.0 & 9.0 \\ 5.0 & 3.0 & 3.0 \\ 9.0 & 5.0 & 8.0 \\ 7.0 & 4.0 & 5.0 \\ 8.0 & 2.0 & 2.0 \end{pmatrix}$$

$$Y = \begin{pmatrix} 0.1 & 0.5 & -2.1 \\ -2.9 & -2.5 & 2.9 \\ -0.9 & -0.5 & -0.1 \\ 1.1 & 2.5 & -4.1 \\ 1.1 & 1.5 & 1.9 \\ 0.1 & -1.5 & 3.9 \\ -1.9 & -0.5 & -2.1 \\ 2.1 & 1.5 & 2.9 \\ 0.1 & 0.5 & -0.1 \\ 1.1 & -1.5 & -3.1 \end{pmatrix}$$

$$P = \begin{pmatrix} -0.1376 & 0.6990 & 0.7017 \\ -0.2505 & 0.6609 & -0.7075 \\ 0.9583 & 0.2731 & -0.0842 \end{pmatrix}$$

$$V = YP$$

$$= \begin{pmatrix} 0.1 & 0.5 & -2.1 \\ -2.9 & -2.5 & 2.9 \\ -0.9 & -0.5 & -0.1 \\ 1.1 & 2.5 & -4.1 \\ 1.1 & 1.5 & 1.9 \\ 0.1 & -1.5 & 3.9 \\ -1.9 & -0.5 & -2.1 \\ 2.1 & 1.5 & 2.9 \\ 0.1 & 0.5 & -0.1 \\ 1.1 & -1.5 & -3.1 \end{pmatrix} \begin{pmatrix} -0.1376 & 0.6990 & 0.7017 \\ -0.2505 & 0.6609 & -0.7075 \\ 0.9583 & 0.2731 & -0.0842 \end{pmatrix}$$

$$= \begin{pmatrix} -2.1514 & -0.1731 & -0.1068 \\ 3.8042 & -2.8875 & -0.5104 \\ 0.1532 & -0.9869 & -0.2694 \\ -4.7065 & 1.3015 & -0.6517 \\ 1.2938 & 2.2791 & -0.4492 \\ 4.0993 & 0.1436 & 0.8031 \\ -1.6258 & -2.2321 & -0.8028 \\ 2.1145 & 3.2512 & 0.1684 \\ -0.2348 & 0.3730 & -0.2751 \\ -2.7464 & -1.0689 & 2.0940 \end{pmatrix}$$

Note that, as the sign of eigenvectors is arbitrary and can change with relatively small perturbations of a data set, SIMFIT provides the option to reflect plots of loadings and scores in order to retain consistency of spatial distribution for the visual presentations of results.

6.4.2 Factor analysis

Factor analysis seeks to explore the relationships between multivariate observations with m variables in terms of a set of k hypothetical factors, where $k < m$. It is widely used in social and psychological research where the factors could be things such as intelligence which are difficult to quantify and model, but it is not used much in the physical sciences where the construction of deterministic models is preferred where possible.

Example 1

From the main SIMFIT menu choose [Statistics], [Multivariate], then [Factor analysis] and read in the default test file `g03caf.tf1` which contains the following correlation matrix from a sample of 211 subjects where 9 variables were measured. Actually, due to the symmetry and unit diagonals, only the strict lower or strict upper triangle is needed, but the SIMFIT data input requires a full matrix because the factor analysis procedure can also read in a data matrix then calculate the correlation matrix interactively.

1	0.523	0.395	0.471	0.346	0.426	0.576	0.434	0.639
0.523	1	0.479	0.506	0.418	0.462	0.547	0.283	0.645
0.395	0.479	1	0.355	0.270	0.254	0.452	0.219	0.504
0.471	0.506	0.355	1	0.691	0.791	0.443	0.285	0.505
0.346	0.418	0.270	0.691	1	0.679	0.383	0.149	0.409
0.426	0.462	0.254	0.791	0.679	1	0.372	0.314	0.472
0.576	0.547	0.452	0.443	0.383	0.372	1	0.385	0.680
0.434	0.283	0.219	0.285	0.149	0.314	0.385	1	0.470
0.639	0.645	0.504	0.505	0.409	0.472	0.680	0.470	1

This matrix is discussed in the book *Factor Analysis as a Statistical Method by D.N.Lawley and E.A.Maxwell London Butterworths (2nd Edition) 1971* which must be consulted in order to understand the following results.

Results from analysis of test file *g03caf.tf1*	
Number of variables	9
Transformation	Untransformed
Matrix type	Input correlation matrix
Number of factors	3
Replicates	Unweighted for replicates
$F(\hat{\Psi})$	0.0350
Test statistic TS	7.1494
Degrees of Freedom	12 (Number of cases = 211)
$P(\chi^2 \geq TS)$	0.8476

Eigenvalues	Communalities	$\hat{\Psi}$
15.968	0.54954	0.45046
4.3577	0.57293	0.42707
1.8475	0.38345	0.61655
1.1560	0.78767	0.21233
1.1190	0.61947	0.38053
1.0271	0.82308	0.17692
0.92574	0.60046	0.39954
0.89508	0.53846	0.46154
0.87710	0.76908	0.23092

Residual correlations

0.0004							
-0.0128	0.0220						
0.0114	-0.0053	0.0231					
-0.0100	-0.0194	-0.0162	0.0033				
-0.0046	0.0113	-0.0122	-0.0009	-0.0008			
0.0153	-0.0216	-0.0108	0.0023	0.0294	-0.0123		
-0.0011	-0.0105	0.0134	0.0054	-0.0057	-0.0009	0.0032	
-0.0059	0.0097	-0.0049	-0.0114	0.0020	0.0074	0.0033	-0.0012

Factor loadings by columns

0.6642	-0.3209	-0.0735
0.6888	-0.2471	-0.1933
0.4926	-0.3022	-0.2224
0.8372	0.2924	-0.0354
0.7050	0.3148	-0.1528
0.8187	0.3767	0.1045
0.6615	-0.3960	-0.0778
0.4579	-0.2955	0.4914
0.7657	-0.4274	-0.0117

Example 2

Test file `g03ccf.tf1` contains the following correlation matrix that is also discussed by Lawley and Maxwell. It is from an analysis of 220 students on the six subjects indicated in column 1. They suggest that "*the fact that all the correlations between the variates are positive indicates that students who get scores above average on any one of the subjects tend also to get scores above average on the other subjects.*"

Gaelic	1	0.439	0.410	0.288	0.329	0.248
English	0.439	1	0.351	0.354	0.320	0.329
History	0.410	0.351	1	0.164	0.190	0.181
Arithmetic	0.288	0.354	0.164	1	0.595	0.470
Algebra	0.329	0.320	0.190	0.595	1	0.464
Geometry	0.248	0.329	0.181	0.470	0.464	1

The next table shows the results from analysis of this correlation matrix for two factors.

Results from analysis of test file *g03ccf.tf1*

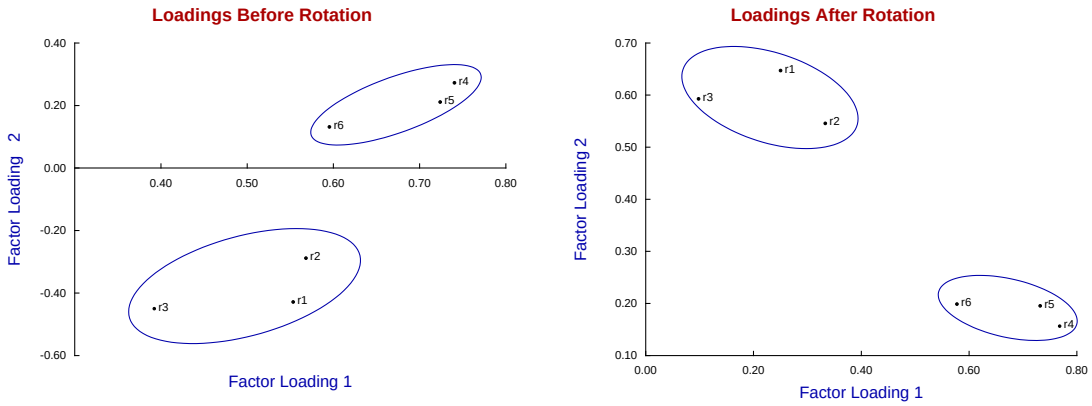
Number of variables	6
Transformation	Untransformed
Matrix type	Input correlation matrix
Number of factors	2
Replicates	Unweighted for replicates
$F(\hat{\Psi})$	0.1088
Test statistic TS	2.3346
Degrees of Freedom	4 (Number of cases = 220)
$P(\chi^2 \geq TS)$	0.6754

Eigenvalues	Communalities	$\hat{\Psi}$
5.6142	0.48983	0.51017
2.1428	0.40593	0.59407
1.0923	0.35627	0.64373
1.0264	0.62264	0.37736
0.9908	0.56864	0.43136
0.8905	0.37179	0.62821
Factor loadings by columns		
0.55332	-0.42856	
0.56816	-0.28832	
0.39218	-0.44996	
0.74042	0.27280	
0.72387	0.21131	
0.59536	0.13169	

The score coefficients are now shown but also a further possibility should be mentioned. As the factors are only unique up to rotation, it is possible to perform a Varimax or Quartimax rotation to calculate a rotation matrix R before working out the score coefficients, which may simplify the interpretation of the observed variables in terms of the unobservable variables.

Factor score coefficients	
Method	Regression
Rotation	None
0.19318	-0.39203
0.17035	-0.22649
0.10852	-0.32621
0.34950	0.33738
0.29891	0.22861
0.16881	0.09783

The next figures illustrate the rows from the loading matrix labeled as $r1, r2, \cdots, r6$ both before and after a Varimax rotation with $\gamma = 1$ and reflection of the y-axis and indicating the presence of two clusters.



Many workers find it convenient to rotate loadings in this way until all are positive so that the relative magnitudes and potential groupings can be visualized more easily. The example illustrated above indicates that factor 2 is what is known as a bi-polar factor with approximately half positive and half negative, but that the obvious grouping is still preserved by rotation.

It should be pointed out that this procedure may also require the use of reflection of axes in order to achieve positive loadings, as in the present case where the second set of loadings were reflected by the automatic technique provided by SIMFIT to do such transformations interactively.

Theory

This technique is used when it is wished to express a multivariate data set in m manifest, or observed variables, in terms of k latent variables, where $k < m$. Latent variables are variables that by definition are unobservable, such as social class or intelligence, and thus cannot be measured but must be inferred by estimating the relationship between the observed variables and the supposed latent variables. The statistical treatment is based upon a very restrictive mathematical model that, at best, will only be a very crude approximation and, most of the time, will be quite inappropriate. For instance, Krzanowski (in *W.J.Krzanowski Principles of Multivariate Analysis, Oxford, revised edition, 2000*) explains how the technique is used in the psychological and social sciences, but then goes on to state

> *At the extremes of, say, Physics or Chemistry, the models become totally unbelievable. p477*
> *It should only be used if a positive answer is provided to the question, "Is the model valid?" p503*

However, despite such warnings, the technique is now widely used, either to attempt to explain observables in terms of hypothetical unobservables, or as just another technique for expressing multivariate data sets in a space of reduced dimension. In this respect it is similar to principal components analysis, except that the technique attempts to capture the covariances between the variables, not the variances. If the observed variables x can be represented as a linear combination of the unobservable variables or factors f, so that the partial correlation $r_{ij.l}$ between x_i and x_j with f_l fixed is effectively zero, then the correlation between x_i and x_j can be said to be explained by f_l. The idea is to estimate the coefficients expressing the dependence of x on f in such a way that the the residual correlation between the x variables is a small as possible, given the value of k.

The assumed relationship between the mean-centered observable variables x_i and the factors is

$$x_i = \sum_{j=1}^{k} \lambda_{ij} f_j + e_i \text{ for } i = 1, 2, \ldots, m, \text{ and } j = 1, 2, \ldots, k$$

where λ_{ij} are the loadings, f_i are independent normal random variables with unit variance, and e_i are independent normal random variables with variances ψ_i. If the variance covariance matrix for x is Σ, defined as

$$\Sigma = \Lambda\Lambda^T + \Psi,$$

where Λ is the matrix of factor loadings λ_{ij}, and Ψ is the diagonal matrix of variances ψ_i, while the sample covariance matrix is S, then maximum likelihood estimation requires the minimization of

$$F(\Psi) = \sum_{j=k+1}^{m} (\theta_j - \log \theta_j) - (m - k),$$

where θ_j are eigenvalues of $S^* = \Psi^{-1/2} S \Psi^{-1/2}$. Finally, the estimated loading matrix $\hat{\Lambda}$ is given by

$$\hat{\Lambda} = \Psi^{1/2} V (\Theta - I)^{1/2},$$

where V are the eigenvectors of S^*, Θ is the diagonal matrix of θ_i, and I is the identity matrix.

The proportion of variation for each variable x_i accounted for by the k factors is the communality $\sum_{j=1}^{k} \lambda_{ij}^2$, the Psi-estimates are the variance estimates, and the residual correlations are the off-diagonal elements of

$$C - (\Lambda\Lambda^T + \Psi)$$

where C is the sample correlation matrix. If a good fit has resulted and sufficient factors have been included, then the off-diagonal elements of the residual correlation matrix should be small with respect to the diagonals (listed with arbitrary values of unity to avoid confusion). Subject to the normality assumptions of the model,

the minimum dimension k can be estimated by fitting sequentially with $k = 1$, $k = 2$, $k = 3$, and so on, until the likelihood ratio test statistic

$$TS = [n - 1 - (2m + 5)/6 - 2k/3]F(\hat{\Psi})$$

is not significant as a chi-square variable with $[(m - k)^2 - (m + k)]/2$ degrees of freedom. Note that data for factor analysis can be input as a general n by m multivariate matrix, or as either a m by m covariance or correlation matrix. However, if a square covariance or correlation matrix is input then there are two further considerations: the sample size must be supplied independently, and it will not be possible to estimate or plot the sample scores in factor space, as the original sample matrix will not be available.

It remains to explain the estimation of scores, which requires the original data of course, and not just the covariance or correlation matrix. This involves the calculation of a m by k factor score coefficients matrix Φ, so that the estimated vector of factor scores $\hat{f}$, given the x vector for an individual can be calculated from

$$\hat{f} = x^T \Phi.$$

However, when calculating factor scores from the factor score coefficient matrix in this way, the observable variables x_i must be mean centered, and also scaled by the standard deviations if a correlation matrix has been analyzed. The regression method uses

$$\Phi = \Psi^{-1}\Lambda(I + \Lambda^T \Psi^{-1}\Lambda)^{-1},$$

while the Bartlett method uses

$$\Phi = \Psi^{-1}\Lambda(\Lambda^T \Psi^{-1}\Lambda)^{-1}.$$

6.4.3 Procrustes analysis

Procrustes analysis is useful when there are two matrices X and Y with the same dimensions, and it wished to see how closely the X matrix can be made to fit the target matrix Y using only distance preserving transformations, like translation and rotation. For instance, X could be a matrix of loadings, and the target matrix Y could be a reference matrix of loadings from another data set.

From the main SimFIT menu choose [Statistics], [Multivariate], then [Procrustes] and analyze the following two matrices

$$X = \begin{bmatrix} 0.63 & 0.58 \\ 1.36 & 0.39 \\ 1.01 & 1.76 \end{bmatrix}$$

$$Y = \begin{bmatrix} 0.00 & 0.00 \\ 1.00 & 0.00 \\ 0.00 & 2.00 \end{bmatrix}$$

contained in the default test files `g03bcf.tf1` with X-data to be rotated, and `g03bcf.tf2` containing the target matrix Y, to obtain the following results.

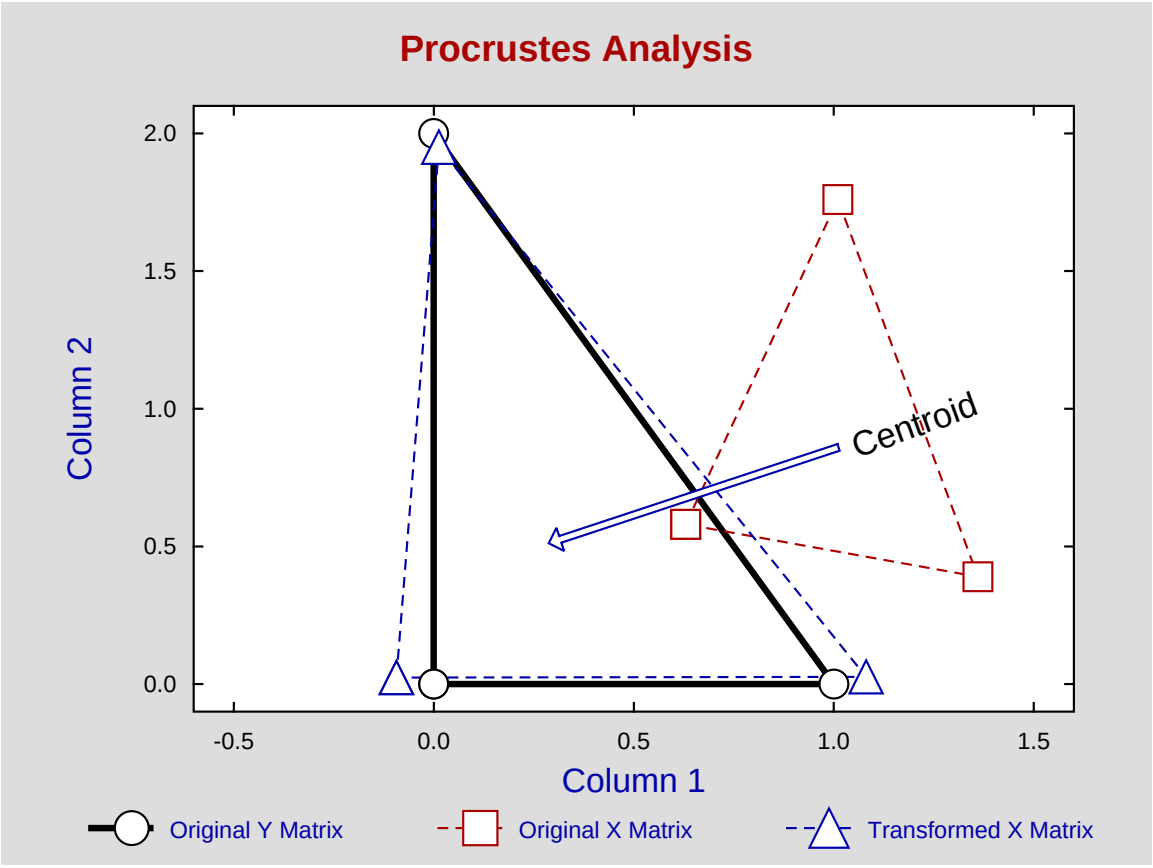
Results from Procrustes analysis
X-data for rotation: `g03bcf.tf1`
Y-data for target: `g03bcf.tf2`
Number of rows: 3
Number of columns: 2
Type: To origin then Y-centroid
Scaling: Least squares scaling
α = 1.5563
Residual sum of squares = 0.019098
Residuals from Procrustes rotation
0.09644
0.08455
0.05145
Rotation matrix from Procrustes rotation
 0.9673 0.2536
-0.2536 0.9673
Y-hat matrix from Procrustes rotation
-0.0934 0.0239
 1.0805 0.0259
 0.0130 1.9502

There are numerous options for performing this calculation including the following alternative types

- No translation or normalization
- Translation to the origin
- Translation to the origin then to the Y centroid after rotation
- Unit normalization
- Translation and normalization (i.e. standardization)

with or without least squares scaling after the rotation.

The following diagram shows the result from transforming the original X matrix which is initially distant from the target Y matrix into the transformed X matrix $\hat{Y}$ which has been brought almost into coincidence with the target Y matrix by movement of the centroid, rotating and scaling.



Also, as well as displaying the residuals, the sum of squares, the rotation and best fit matrices, options are provided to plot arbitrary rows or columns of these matrices.

Theory

First the centroids of X and Y are translated to the origin to give X_c and Y_c. Then the matrix of rotations R that minimize the sum of squared residuals is found from the singular value decomposition as

$$X_c^T Y_c = UDV^T$$
$$R = UV^T,$$

and after rotation a dilation factor α can be estimated by least squares, if required, to give the estimate

$$\hat{Y}_c = \alpha X_c R.$$

Additional options from the SimFIT Procrustes interface include normalizing both matrices to have unit sums of squares, normalizing the X matrix to have the same sum of squares as the Y matrix, and translating to the original Y centroid after rotation. Note that these Procrustes options can often be done interactively in SimFIT whenever loadings are calculated.

6.4.4 Varimax and Quartimax rotation

Generalized orthomax rotation techniques can be used to simplify the interpretation of loading matrices, e.g. from canonical variates or factor analysis. These are only unique up to rotation so, by applying rotations according to stated criteria, different contributions of the original variables can be assessed.

Example 1

From the main SIMFIT menu choose [Statistics], [Multivariate], then [Varimax and Quartimax] and analyze the default test file provided, `g03baf.tf1`, to obtain the following results.

Results from Varimax rotation
Number of rows: 10
Number of columns: 3
Type: Unstandardised
Scaling: Varimax ($\gamma = 1$)
Original data Λ

0.788	-0.152	-0.352
0.874	0.381	0.041
0.814	-0.043	-0.213
0.798	-0.170	-0.204
0.641	0.070	-0.042
0.755	-0.298	0.067
0.782	-0.221	0.028
0.767	-0.091	0.358
0.733	-0.384	0.229
0.771	-0.101	0.071

Rotation matrix R

0.63347	-0.53367	-0.56029
0.75803	0.57333	0.31095
0.15529	-0.62169	0.76772

Rotated matrix $\Lambda^* = \lambda R$

0.32929	-0.28884	-0.75901
0.84882	-0.27348	-0.33974
0.44997	-0.32664	-0.63297
0.34496	-0.39651	-0.65659
0.45259	-0.27584	-0.36962
0.26278	-0.61542	-0.46424
0.33219	-0.56144	-0.48537
0.47248	-0.68406	-0.18319
0.20881	-0.75370	-0.35429
0.42287	-0.51350	-0.40888

The input loading matrix Λ has m rows and k columns and results from the analysis of an original data matrix with n rows (i.e. cases) and m columns (i.e. variables), where k factors have been calculated for $k \leq m$.

Example 2

If the input loading matrix is not standardized to unit length rows, this can be done interactively, as in the next example which also illustrates the use of Varimax rotation to simplify the interpretation of loadings containing negative values by forming a rotated loading matrix with most values positive.

Results from Varimax rotation
Number of rows: 10
Number of columns: 2
Type: Row standardized
Scaling: Varimax ($\gamma = 1$)
Original data Λ

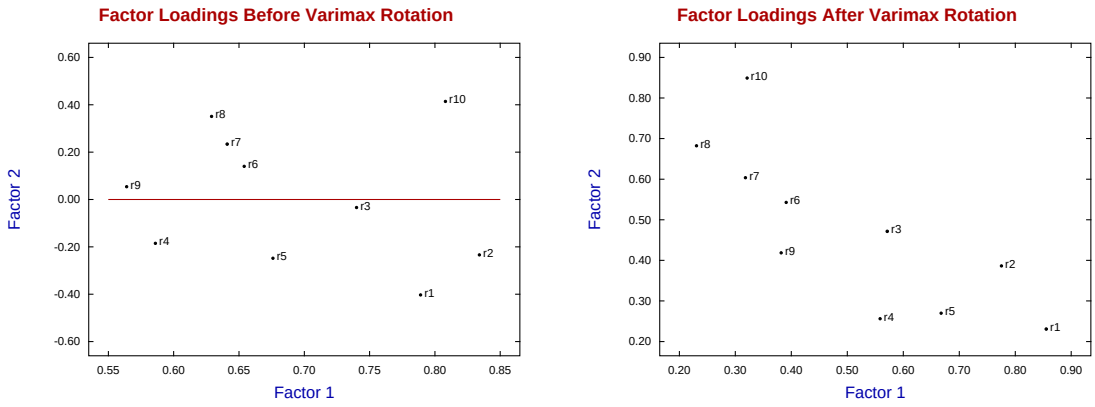
0.789	-0.403
0.834	-0.234
0.740	-0.034
0.586	-0.185
0.676	-0.248
0.654	0.140
0.641	0.234
0.629	0.351
0.564	0.054
0.808	0.414

Rotation matrix R

0.7279	0.6857
-0.6857	0.7279

Rotated matrix $\Lambda^* = \lambda R$

0.8506	0.2477
0.7675	0.4016
0.5619	0.4827
0.5534	0.2672
0.6621	0.2830
0.3800	0.5504
0.3061	0.6099
0.2171	0.6868
0.3735	0.4260
0.3042	0.8554



In these diagrams it is clear that factor 1 loads most heavily on variables 1 to 5 (i.e. as rows 1 to 5 of the loading matrix with symbols r1 to r5), while factor 2 loads most heavily on variables 6 to 10 (i.e. as rows 6 to 10 of the loading matrix with symbols r6 to r10). However it should be noted that before rotation there were both positive positive and negative loadings in the left hand figure as emphasized by the red line, while after rotation all loadings are now positive. So, the dependence of factors and loadings may be thought easier to appreciate after the rotation to make all values positive as in the right hand figure.

The SimFIT Varimax procedure is made available when loading matrices are calculated and provides the facility to plot any selection of two or three columns of a loading matrix before or after rotation.

Theory

The rotated matrix Λ^* is calculated so that the elements λ^*_{ij} are either relatively large or small. This involves maximizing the function

$$V = \sum_{j=1}^{k} \sum_{i=1}^{m} (\lambda^*_{ij})^4 - \frac{\gamma}{m} \sum_{j=1}^{k} \left[\sum_{i=1}^{m} (\lambda^*_{ij})^2 \right]^2$$

where there were m variables originally and k factors were chosen for $k < m$.

There are several cases as follows

- Varimax rotation: $\gamma = 1$
- Quartimax rotation: $\gamma = 0$.
- Equamax rotation: $\gamma = k/2$.
- Parsimax rotation: $\gamma = m(k-1)/(m+k+2)$.
- User chosen rotation: γ input.

The resulting rotation matrix R satisfies $\Lambda^* = \Lambda R$ and, when the matrices have been calculated they can be viewed, written to the results log file, saved to a text file, or plotted.

6.4.5 Biplots in two or three dimensions

Biplots are widely used to view multivariate data in a space with smaller dimensions.

The data would normally be held in a spreadsheet program like Microsoft Office Excel or LibreOffice Calc, as in this example of an unselected table of multivariate statistical data from K.R. Gabriel in *Biometrika* 1971, **58**, 453–67.

Percent	Christian	Armenian	Jewish	Moslem	American	Shaafat	A-Tur	Silwan	Sur-Bahar
Toilet	98.2	97.2	97.3	96.9	97.6	94.4	90.2	94	70.5
Kitchen	78.8	81	65.6	73.3	91.4	88.7	82.2	84.2	55.1
Bath	14.4	17.6	6	9.6	56.2	69.5	31.8	19.5	10.7
Electricity	86.2	82.1	54.5	74.7	87.2	80.4	68.6	65.5	26.1
Water	32.9	30.3	21.1	26.9	80.1	74.3	46.3	36.2	9.8
Radio	73	70.4	53	60.5	81.2	78	67.9	64.8	57.1
TV set	4.6	6	1.5	3.4	12.7	23	5.6	2.7	1.3
Refrigerator	29.2	26.3	4.3	10.5	52.8	49.7	21.7	9.5	1.2

Such tables must be rectangular, with optional row and column labels, and all other cells filled with numerical data (missing data must be replaced by estimates). Often it would only be necessary to select cells containing numerical values from such a table by highlighting as follows.

Percent	Christian	Armenian	Jewish	Moslem	American	Shaafat	A-Tur	Silwan	Sur-Bahar
Toilet	98.2	97.2	97.3	96.9	97.6	94.4	90.2	94	70.5
Kitchen	78.8	81	65.6	73.3	91.4	88.7	82.2	84.2	55.1
Bath	14.4	17.6	6	9.6	56.2	69.5	31.8	19.5	10.7
Electricity	86.2	82.1	54.5	74.7	87.2	80.4	68.6	65.5	26.1
Water	32.9	30.3	21.1	26.9	80.1	74.3	46.3	36.2	9.8
Radio	73	70.4	53	60.5	81.2	78	67.9	64.8	57.1
TV set	4.6	6	1.5	3.4	12.7	23	5.6	2.7	1.3
Refrigerator	29.2	26.3	4.3	10.5	52.8	49.7	21.7	9.5	1.2

However, sometimes row and column labels could also be needed, when a labeled table with cells containing either labels or numerical values would be selected, as follows.

Percent	Christian	Armenian	Jewish	Moslem	American	Shaafat	A-Tur	Silwan	Sur-Bahar
Toilet	98.2	97.2	97.3	96.9	97.6	94.4	90.2	94	70.5
Kitchen	78.8	81	65.6	73.3	91.4	88.7	82.2	84.2	55.1
Bath	14.4	17.6	6	9.6	56.2	69.5	31.8	19.5	10.7
Electricity	86.2	82.1	54.5	74.7	87.2	80.4	68.6	65.5	26.1
Water	32.9	30.3	21.1	26.9	80.1	74.3	46.3	36.2	9.8
Radio	73	70.4	53	60.5	81.2	78	67.9	64.8	57.1
TV set	4.6	6	1.5	3.4	12.7	23	5.6	2.7	1.3
Refrigerator	29.2	26.3	4.3	10.5	52.8	49.7	21.7	9.5	1.2

Note that the dummy label in cell(1,1) is not used.

The structure of the default SIMFIT test file `houses.tf1` available after selecting [Statistics] from the SIMFIT main menu, followed by [Multivariate], then [Biplots] will now be explained.

First of all, note that SIMFIT is not constrained to work with spread sheet programs, and the data file format is more universal and much simpler, being a simple ASCII table of space separated numerical values with optional row and column labels. So the default test file `houses.tf1` contains the following table of observations.

98.2	97.2	97.3	96.9	97.6	94.4	90.2	94.0	70.5
78.8	81.0	65.6	73.3	91.4	88.7	82.2	84.2	55.1
14.4	17.6	6.0	9.6	56.2	69.5	31.8	19.5	10.7
86.2	82.1	54.5	74.7	87.2	80.4	68.6	65.5	26.1
32.9	30.3	21.1	26.9	80.1	74.3	46.3	36.2	9.8
73.0	70.4	53.0	60.5	81.2	78.0	67.9	64.8	57.1
4.6	6.0	1.5	3.4	12.7	23.0	5.6	2.7	1.3
29.2	26.3	4.3	10.5	52.8	49.7	21.7	9.5	1.2

Also, as the row and column labels would be required for a biplot, these are added to the test file as follows.

```
begin{labels}
Toilet
Kitchen
Bath
Electricity
Water
Radio
TV set
Refrigerator
Christian
Armenian
Jewish
Moslem
Am.Colony Sh.Jarah
Shaafat Bet-Hanina
A-Tur Isawyie
Silwan Abu-Tor
Sur-Bahar Bet-Safafa
end{labels}
```

An Excel macro called `simfit6.xls` is distributed with the SIMFIT package and it can output spreadsheet tables as correctly formatted SIMFIT data files from within Excel. Another easy way is to copy and paste the whole table directly into SIMFIT using the [Paste] option from the file selection control, or to copy and paste into program **maksim** which will then output a correctly formatted SIMFIT data file. However, if this course of action is to be followed, the following important restrictions may have to be noted.

1. There must be no missing values and every cell in the numeric part of the table must contain a valid number, except cell(1,1) which is ignored.

2. Data copied to the clipboard from a spreadsheet program will have tab separated columns and so SIMFIT will be able to perform numerous format conversion procedures interactively.

3. If spaces are used as column separators instead of tabs, the data must be in scientific format using full stops for decimal points not commas.

4. If spaces are used as column separators instead of tabs, there must no spaces in the labels, and any must be replaced by undercores before copying to the clipboard from a standard ASCII text editor such as **notepad**. For example, replace `time of day` by `time_of_day`, or `cycles per second` by `cycles_per_second`. This restriction does not matter with formatted SIMFIT data files as the row labels followed by the column labels are added as sequential lines between the begin{labels} and end{labels} section of the data file trailer.

The next figures illustrate typical biplots derived from `houses.tf1`.

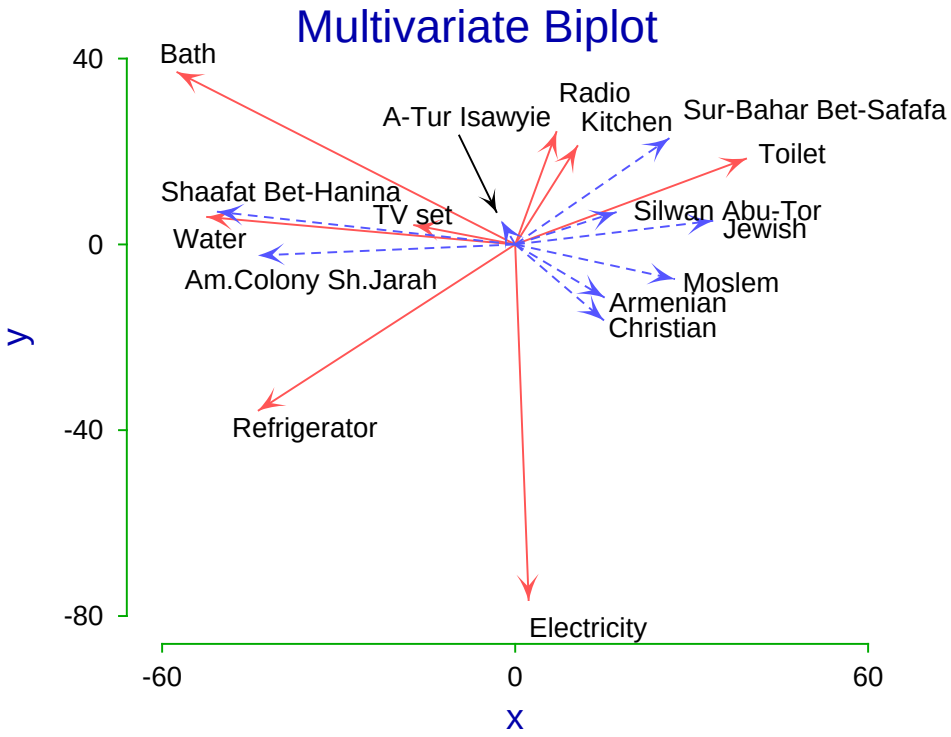


Figure 1: 2D Biplot

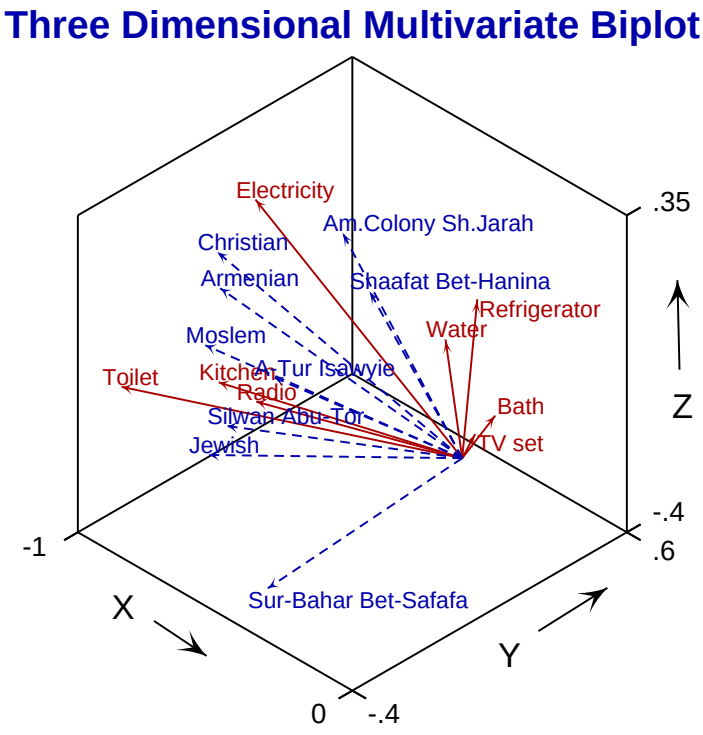


Figure 2: 3D Biplot

As with other projection techniques, such as principal components, it is necessary to justify that the number of singular values used to display a biplot does represent the data matrix adequately. To do this, consider the next table from the singular value decomposition of `houses.tf1`.

Proportion of total variance captured by singular values
Data file: `houses.tf1`, rank = 8

Index	σ_i	Fraction	Cumulative	σ_i^2	Fraction	Cumulative
1	499.393	0.7486	0.7486	249394	0.9631	0.9631
2	88.3480	0.1324	0.8811	7805.36	0.0301	0.9933
3	33.6666	0.0505	0.9315	1133.44	0.0044	0.9977
4	17.8107	0.0267	0.9582	317.222	0.0012	0.9989
5	12.8584	0.0193	0.9775	165.339	0.0006	0.9995
6	10.4756	0.0157	0.9932	109.738	0.0004	1.0000
7	3.37372	0.0051	0.9983	11.3820	0.0000	1.0000
8	1.15315	0.0017	1.0000	1.32974	0.0000	1.0000

In this example, it is clear that the first two or three singular values do represent the data adequately, and this is further reinforced by Figure 3 where the percentage variance represented by the successive singular values is plotted as a function of the singular value index. Here we see the cumulative variance $CV(i)$

$$CV(i) = \frac{100 \sum_{j=1}^{i} \sigma_j^2}{\sum_{j=1}^{k} \sigma_j^2}$$

plotted as a function of the index i, and such tables or plots should always be inspected to make sure that $CV(i)$ is greater than some minimum value (say 70 percent, for instance) for $i = 2$ or $i = 3$ as appropriate.

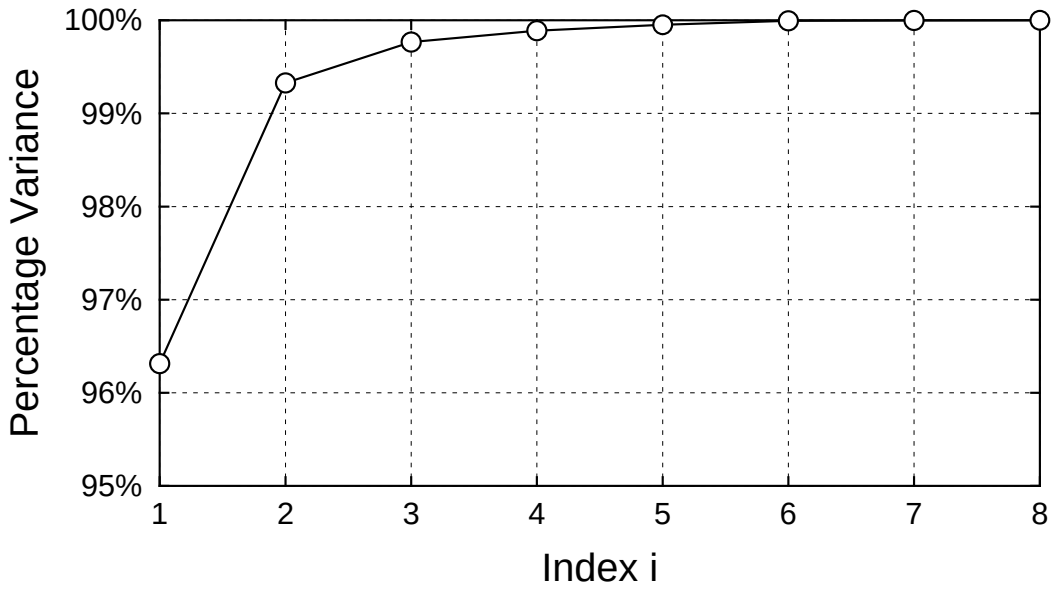


Figure 3: Cumulative Variance Plot

The theory behind the biplot options available in SIMFIT will now be described.

Theory

The biplot is used to explore relationships between the rows and columns of any arbitrary matrix, by projecting the matrix onto a space of smaller dimensions using the singular value decomposition (SVD). It is based upon the fact that, as a n by m matrix X of rank k can be expressed as a sum of k rank 1 matrices as follows

$$X = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \cdots + \sigma_k u_k v_k^T,$$

then the best fit rank r matrix Y with $r < k$ which minimizes the objective function

$$\begin{aligned} S &= \sum_{i=1}^{m} \sum_{j=1}^{n} (x_{ij} - y_{ij})^2 \\ &= \text{trace}[(X - Y)(X - Y)^T] \end{aligned}$$

is the sum of the first r of these rank 1 matrices. Further, such a least squares approximation results in the minimum value

$$S_{min} = \sigma_{r+1}^2 + \sigma_{r+2}^2 + \cdots + \sigma_k^2$$

so that the rank r least squares approximation Y accounts for a fraction

$$\frac{\sigma_1^2 + \cdots + \sigma_r^2}{\sigma_1^2 + \sigma_2^2 + \cdots + \sigma_k^2}$$

of the total variance, where k is less than or equal to the smaller of n and m, k is greater than or equal to r, and $\sigma_i = 0$ for $i > k$.

Figure 1 illustrates a biplot for the data in test file `houses.tf1`. The technique is based upon creating one of several possible rank-2 representations of of a n by m matrix X with rank k of at least two as follows. Let the SVD of X be

$$\begin{aligned} X &= U \Sigma V^T \\ &= \sum_{i=1}^{k} \sigma_i u_i v_i^T \end{aligned}$$

so that the best fit rank-2 matrix Y to the original matrix X will be

$$Y = \begin{pmatrix} u_{11} & u_{21} \\ u_{12} & u_{22} \\ \vdots & \vdots \\ u_{1n} & u_{2n} \end{pmatrix} \begin{pmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{pmatrix} \begin{pmatrix} v_{11} & v_{12} & \ldots & v_{1m} \\ v_{21} & v_{22} & \ldots & v_{2m} \end{pmatrix}.$$

Then Y can be written in several ways as GH^T, where G is a n by 2 matrix and H is a m by 2 matrix as follows.

1. General representation

$$Y = \begin{pmatrix} u_{11}\sqrt{\sigma_1} & u_{21}\sqrt{\sigma_2} \\ u_{12}\sqrt{\sigma_1} & u_{22}\sqrt{\sigma_2} \\ \vdots & \vdots \\ u_{1n}\sqrt{\sigma_1} & u_{2n}\sqrt{\sigma_2} \end{pmatrix} \begin{pmatrix} v_{11}\sqrt{\sigma_1} & v_{12}\sqrt{\sigma_1} & \ldots & v_{1m}\sqrt{\sigma_1} \\ v_{21}\sqrt{\sigma_2} & v_{22}\sqrt{\sigma_2} & \ldots & v_{2m}\sqrt{\sigma_2} \end{pmatrix}$$

2. Representation with row emphasis

$$Y = \begin{pmatrix} u_{11}\sigma_1 & u_{21}\sigma_2 \\ u_{12}\sigma_1 & u_{22}\sigma_2 \\ \vdots & \vdots \\ u_{1n}\sigma_1 & u_{2n}\sigma_2 \end{pmatrix} \begin{pmatrix} v_{11} & v_{12} & \ldots & v_{1m} \\ v_{21} & v_{22} & \ldots & v_{2m} \end{pmatrix}$$

3. Representation with column emphasis

$$Y = \begin{pmatrix} u_{11} & u_{21} \\ u_{12} & u_{22} \\ \vdots & \vdots \\ u_{1n} & u_{2n} \end{pmatrix} \begin{pmatrix} v_{11}\sigma_1 & v_{12}\sigma_1 & \dots & v_{1m}\sigma_1 \\ v_{21}\sigma_2 & v_{22}\sigma_2 & \dots & v_{2m}\sigma_2 \end{pmatrix}$$

4. User-defined representation

$$Y = \begin{pmatrix} u_{11}\sigma_1^{\alpha} & u_{21}\sigma_2^{\alpha} \\ u_{12}\sigma_1^{\alpha} & u_{22}\sigma_2^{\alpha} \\ \vdots & \vdots \\ u_{1n}\sigma_1^{\alpha} & u_{2n}\sigma_2^{\alpha} \end{pmatrix} \begin{pmatrix} v_{11}\sigma_1^{\beta} & v_{12}\sigma_1^{\beta} & \dots & v_{1m}\sigma_1^{\beta} \\ v_{21}\sigma_2^{\beta} & v_{22}\sigma_2^{\beta} & \dots & v_{2m}\sigma_2^{\beta} \end{pmatrix}$$

where $0 < \alpha < 1$, and $\beta = 1 - \alpha$.

To construct a biplot we take the n row effect vectors g_i and m column effect vectors h_j as vectors with origin at $(0, 0)$ and defined in the general representation as

$$\begin{aligned} g_i^T &= (u_{1i}\sqrt{\sigma_1}, u_{2i}\sqrt{\sigma_2}) \\ h_j^T &= (v_{1j}\sqrt{\sigma_1}, v_{2j}\sqrt{\sigma_2}) \end{aligned}$$

with obvious identities for the alternative row emphasis and column emphasis factorizations. The biplot consists of n vectors with end points at $(u_{1i}\sqrt{\sigma_1}, u_{2i}\sqrt{\sigma_2})$ and m vectors with end points at $(v_{1j}\sqrt{\sigma_1}, v_{2j}\sqrt{\sigma_2})$ so that interpretation of the biplot is then in terms of the inner products of vector pairs. That is, vectors with the same direction correspond to proportional rows or columns, while vectors approaching right angles indicate near orthogonality, or small contributions. Another possibility is to display a difference biplot in which a residual matrix R is first created by subtracting the best fit rank-1 matrix so that

$$\begin{aligned} R &= X - \sigma_1 u_1 v_1^T \\ &= \sum_{i=2}^{k} \sigma_i u_i v_i^T \end{aligned}$$

and this is analyzed, using appropriate vectors calculated with σ_2 and σ_3 of course. Again, the row vectors may dominate the column vectors or vice versa whatever representation is used and, to improve readability, additional scaling factors may need to be introduced. For instance, the previous figures used the residual matrix and scaling factors of -100 for rows and -1 for columns to reflect and stretch the vectors until comparable size was attained. To do this over-rides the default auto-scaling option, which is to scale each set of vectors so that the largest row and largest column vector are of unit length, whatever representation is chosen.

Biplots are most useful when the number of rows and columns is not too large, and when the rank-2 approximation is satisfactory as an approximation to the data or residual matrix. Note that biplot labels should be short, and they can be appended to the data file as with `houses.tf1`, or pasted into the plot as a table of label values. Fine tuning to re-position labels was necessary with these figures, and this can be done by editing the PostScript file in a text editor, or by using techniques described elsewhere for moving labels in scattergrams.

Sometimes, as with Figure 2, it is useful to inspect biplots in three dimensions. This has the advantage that three singular values can be used, but the plot may have to be viewed from several angles to get a good idea of which vectors of like type are approaching a parallel orientation (indicating proportionality of rows or columns) and which pairs of vectors i, j of opposite types are orthogonal (i.e., at right angles, indicating small contributions to x_{ij})

6.5 Multivariate analysis of variance (MANOVA)

Tutorials and worked examples for simulation, curve fitting, statistical analysis, and plotting.
https://simfit.uk
https://simfit.org.uk
https://simfit.silverfrost.com

6.5.1 Introduction

The multivariate analysis of variance technique is an extension of the standard univariate analysis of variance procedure (ANOVA) to the situation where observations of more than one variable are made for each subject. So, for instance, a simple application of ANOVA would be to assume that multiple observations have been made of a single variable in several groups and, assuming that this variable is distributed normally in each group with the same variance, to test if all the population means are identical. In the corresponding MANOVA case it would be to assume that all the observations are from multivariate normal distributions with the same covariance matrix, and to test for identical mean vectors in the populations.

For example, sometimes a designed experiment is conducted in which more than one response is measured at each treatment, so that there are two possible courses of action.

1. Do a separate ANOVA analysis for each variable.
   The disadvantages of this approach are that it is tedious, and also it relies upon the questionable assumption that each variable is statistically independent of every other variable, with a fixed variance for each variable. The advantages are that the variance ratio tests are intuitive and unambiguous, and also there is no requirement that sample size per group should be greater than the number of variables.

2. Do an overall MANOVA analysis for all variables simultaneously.
   The disadvantages of this technique are that it relies on the assumption of a multivariate normal distribution with identical covariance matrices across groups, it requires a sample size per group greater than the number of variables, and also there is no unique and intuitive best test statistic. Further, the power will tend to be lower than the power of the corresponding ANOVA. The advantages are that analysis is compact, and several useful options are available which simplify situations like the analysis of repeated measurements.

Central to a MANOVA analysis are the assumptions that there are n observations of a random m dimensional vector divided into g groups, each with n_i observations, so that $n = \sum_{i=1}^{g} n_i$ where $n_i \geq m$ for $i = 1, 2, \ldots, g$.

If y_{ij} is the m vector for individual j of group i, then the sample mean $\bar{y}_i$, corrected sum of squares and products matrix C_i, and covariance matrix S_i for group i are

$$\bar{y}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} y_{ij}$$

$$C_i = \sum_{j=1}^{n_i} (y_{ij} - \bar{y}_i)(y_{ij} - \bar{y}_i)^T$$

$$S_i = \frac{1}{n_i - 1} C_i.$$

For each ANOVA design there will be a corresponding MANOVA design in which corrected sums of squares and product matrices replace the ANOVA sums of squares, but where other test statistics are required in place of the ANOVA F distributed variance ratios. This will be clarified by dealing with typical MANOVA procedures, such as testing for equality of means and equality of covariance matrices across groups.

6.5.2 MANOVA examples

From the main SIMFIT menu choose [Statistcs], [Multivariate], then [MANOVA], noting that several test files named `manova1.tfk` are provided for $k = 1$ to $k = 5$. It is important to realize that the first column in all data sets provided for MANOVA analysis must have the group numbers as successive integers in column 1 in nondecreasing order, with further columns for observations.

MANOVA example 1. Testing for equality of all means

Example 1 describes the results from analyzing these data for three groups and two variables contained in test file `manova1.tf3`.

1	3	10
1	5	16
1	5	16
1	4	14
1	1	9
2	8	12
2	4	8
2	4	6
2	2	6
2	9	14
3	10	16
3	4	10
3	10	18
3	4	14
3	10	16

Column 1 is the group number (in nondecreasing order), while columns 2 and 3 are the observations.

If all groups have the same multivariate normal distribution, then estimates for the mean μ and covariance matrix Σ can be obtained from the overall sample statistics $\hat{\mu} = \bar{y}$ and $\hat{\Sigma}$

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{g}\sum_{j=1}^{n_i} y_{ij}$$

$$\hat{\Sigma} = \frac{1}{n-1}\sum_{i=1}^{g}\sum_{j=1}^{n_i}(y_{ij}-\hat{\mu})(y_{ij}-\hat{\mu})^T$$

obtained by ignoring group means $\bar{y}_i$ and summing across all groups. Alternatively, the pooled between-groups B, within-groups W, and total sum of squares and products matrices T can be obtained along with the within-groups covariance matrix S using the group mean estimates $\bar{y}_i$ as

$$\begin{aligned}
B &= \sum_{i=1}^{g} n_i(\bar{y}_i-\bar{y})(\bar{y}_i-\bar{y})^T \\
W &= \sum_{i=1}^{g}\sum_{j=1}^{n_i}(y_{ij}-\bar{y}_i)(y_{ij}-\bar{y}_i)^T \\
&= \sum_{i=1}^{g}(n_i-1)S_i \\
&= (n-g)S \\
T &= B+W \\
&= (n-1)\hat{\Sigma}.
\end{aligned}$$

This table is typical, and clearly strong differences between groups will be indicated if B is much larger than W.

Source of variation	d.f.	ssp matrix
Between groups	$g-1$	B
Within groups	$n-g$	W
Total	$n-1$	T

The usual likelihood ratio test statistic is Wilk's lambda defined as

$$\Lambda = \frac{|W|}{|B|+|W|}$$

but other statistics can also be defined as functions of the eigenvalues of BW^{-1}. Unlike B and W separately, the matrix BW^{-1} is not symmetric and positive definite but, if the m eigenvalues of BW^{-1} are θ_i, then Wilk's lambda, Roy's largest root R, the Lawley-Hotelling trace T, and the Pillai trace P can be defined as

$$\Lambda = \prod_{i=1}^{m} \frac{1}{1+\theta_i}$$
$$R = \max(\theta_i)$$
$$T = \sum_{i=1}^{m} \theta_i$$
$$P = \sum_{i=1}^{m} \frac{\theta_i}{1+\theta_i}.$$

The next table of results was obtained when `manova1.tf3` was analyzed, and the methods used to calculate the significance levels will then be outlined.

MANOVA H_0: all mean vectors are equal

Number of groups	3
Number of variables	2
Number of observations	15

Statistic	Value	Transform	$NDOF$	p	
Wilks lambda	0.1917	7.062	4, 22	0.0008	*Reject H_0 at 1%*
Roys largest root	2.801				
Lawley-Hotelling T	3.173	8.727	4, 11	0.0017	*Reject H_0 at 1%*
Pillais trace	1.008				

The next table indicates conditions on the number of groups g, variables m, and total number of observations n that lead to exact F variables for appropriate transforms of Wilk's Λ.

Parameters	F statistic	Degrees of freedom
$g=2$, any m	$\frac{(2g-m-1)(1-\Lambda)}{m\Lambda}$	$m, 2g-m-1$
$g=3$, any m	$\frac{(3g-m-2)(1-\sqrt{\Lambda})}{m\sqrt{\Lambda}}$	$2m, 2(n-m-2)$
$m=1$, any g	$\frac{(n-g)(1-\Lambda)}{(g-1)\Lambda}$	$g-1, n-g$
$m=2$, any g	$\frac{(n-g-1)(1-\sqrt{\Lambda})}{(g-1)\sqrt{\Lambda}}$	$2(g-1), 2(n-g-1)$

For other conditions the asymptotic expression

$$-\left(\frac{2n-2-m-g}{2}\right)\log\Lambda \sim F_{m,g-1}$$

is generally used. The Lawley-Hotelling trace is a generalized Hotelling's T_0^2 statistic, and so the null distribution of this can be approximated as follows.

Defining the degrees of freedom and multiplying factors α and β by

$$\begin{aligned}
\nu_1 &= g - 1 \\
\nu_2 &= n - g \\
\nu &= \frac{m\nu_1(\nu_2 - m)}{\nu_1 + \nu_2 - m\nu_1 - 1} \\
\alpha &= \frac{(\nu_2 - 1)(\nu_1 + \nu_2 - m - 1)}{(\nu_2 - m)(\nu_2 - m - 1)(\nu_2 - m - 3)} \\
\beta &= \frac{m\nu_1}{\nu_2 - m + 1},
\end{aligned}$$

then the case $\nu > 0$ leads to the approximation

$$T \sim \beta F_{\nu,\nu_2-m+1},$$

otherwise the alternative approximation

$$T \sim \alpha\chi_f^2$$

is employed, where $f = m\nu_1/\{\alpha(\nu_2 - m - 1)\}$. The null distributions for Roy's largest root and Pillai's trace are more complicated to approximate, which is one reason why Wilk's Λ is the most widely used test statistic.

MANOVA example 2. Testing for equality of selected means

The next table resulted when groups 2 and 3 were tested for equality of selected means, another example of a Hotelling's T^2 test.

MANOVA H_0: selected group means are equal		
First group	2 (5 cases)	
Second group	3 (5 cases)	
Number of observations	15 (to estimate CV)	
Number of variables	2	
Hotelling T^2	12.00	
Test statistic S	5.498	
Numerator DOF	2	
Denominator DOF	11	
$P(F \geq S)$	0.0221	*Reject H_0 at 5% significance level*

MANOVA H_0: selected group means are equal		
First group	2 (5 cases)	
Second group	3 (5 cases)	
Number of observations	10 (to estimate CV)	
Number of variables	2	
Hotelling T^2	15.18	
Test statistic S	6.640	
Numerator DOF	2	
Denominator DOF	7	
$P(F \geq S)$	0.0242	*Reject H_0 at 5% significance level*

The first result uses the difference vector $d_{2,3}$ between the means estimated from groups 2 and 3 with the matrix $W = (n - g)S$ estimated using the pooled sum of squares and products matrix to calculate and test T^2

according to

$$T^2 = \left(\frac{(n-g)n_2n_3}{n_2+n_3}\right) d_{2,3}^T W^{-1} d_{2,3}$$
$$\frac{n-g-m+1}{m(n-g)} T^2 \sim F_{m,n-g-m+1},$$

while the second result uses the data from samples 2 and 3 as if they were the only groups as follows

$$S_{2,3} = \frac{(n_2-1)S_2 + (n_3-1)S_3}{n_2+n_3-2}$$
$$T^2 = \left(\frac{n_2n_3}{n_2+n_3}\right) d_{2,3}^T S_{2,3}^{-1} d_{2,3}$$
$$\frac{n_2+n_3-m-1}{m(n_2+n_3-2)} T^2 \sim F_{m,n_2+n_3-m-1}.$$

The first method could be used if all covariance matrices are equal (see next) but the second might be preferred if it was only likely that the selected covariance matrices were identical.

MANOVA example 3. Testing for equality of all covariance matrices

The next data set in `manova1.tf2` has three groups for three types of Cushing's syndrome, the variables are logarithms of urinary excretion rates (mg/hr) for two steroid metabolites, and the table below the data shows the results from testing that the within-group variance-covariance matrices are equal.

1	1.1314	2.4596
1	1.0986	0.2624
1	0.6419	-2.3026
1	1.3350	-3.2189
1	1.4110	0.0953
1	0.6419	-0.9163
2	2.1163	0.0000
2	1.3350	-1.6094
2	1.3610	-0.5108
2	2.0541	0.1823
2	2.2083	-0.5108
2	2.7344	1.2809
2	2.0412	0.4700
2	1.8718	-0.9163
2	1.7405	-0.9163
2	2.6101	0.4700
3	2.3224	1.8563
3	2.2192	2.0669
3	2.2618	1.1314
3	3.9853	0.9163
3	2.7600	2.0281

MANOVA H_0: all covariance matrices are equal

Number of groups	3	
Number of observations	21	
Number of variables	2	
Test statistic C	19.24	
Degrees of freedom	6	
$P(\chi^2 \geq C)$	0.0038	*Reject H_0 at 1% significance level*

These results refer to using Box's test to analyze `manova1.tf2` for equality of covariance matrices. This depends on the likelihood ratio test statistic C defined by

$$C = M\left\{(n-g)\log|S| - \sum_{i=1}^{g}(n_i-1)\log|S_i|\right\},$$

where the multiplying factor M is

$$M = 1 - \frac{2m^2+3m-1}{6(m+1)(g-1)}\left(\sum_{i=1}^{g}\frac{1}{n_i-1} - \frac{1}{n-g}\right)$$

and, for large n, C is approximately distributed as χ^2 with $m(m+1)(g-1)/2$ degrees of freedom. Just as tests for equality of variances are not very robust, this test should be used with caution, and then only with large samples, i.e. $n_i >> m$.

MANOVA example 4. Profile analysis

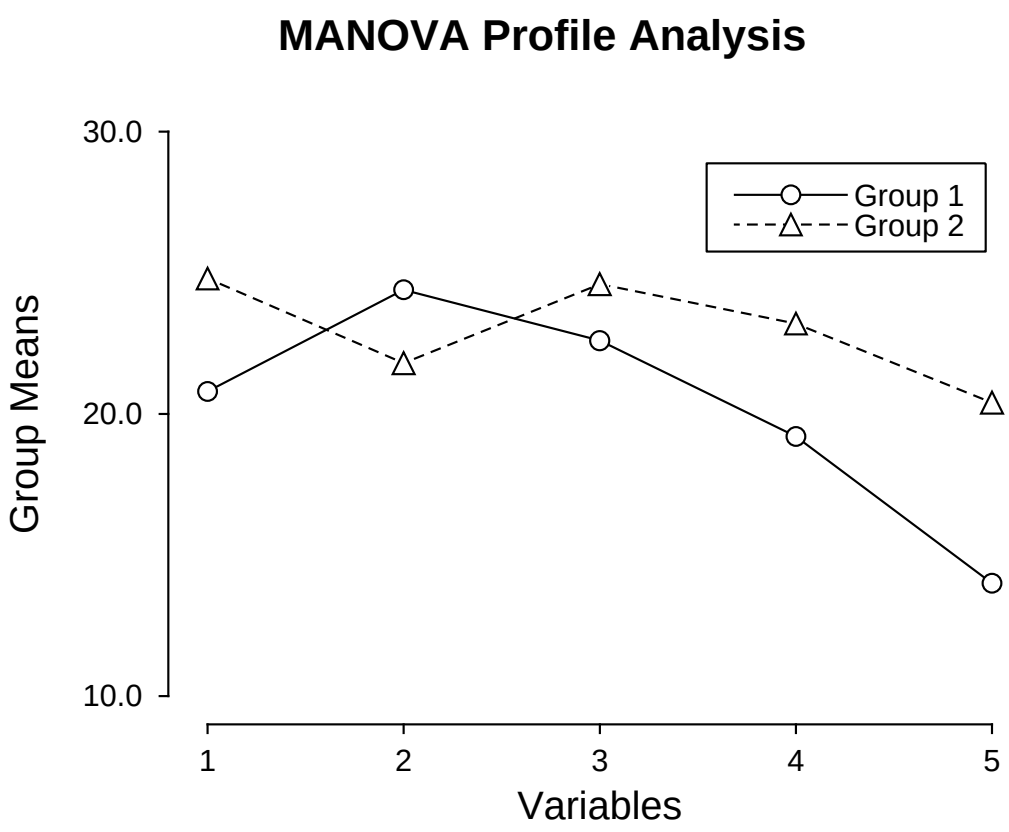
Test file `manova1.tf1` has two groups and five variables as follows,

1	11	18	15	18	15
1	33	27	31	21	17
1	20	28	27	23	19
1	18	26	18	18	9
1	22	23	22	16	10
2	18	17	20	18	18
2	31	24	31	26	20
2	14	16	17	20	17
2	25	24	31	26	18
2	36	28	24	26	29

while the following table shows the results of statistical analysis using the profile option.

MANOVA H_0: selected group profiles are equal		
First group	1 (5 cases)	
Second group	2 (5 cases)	
Number of observations	10 (to estimate CV)	
Number of variables	5	
Hotelling T^2	35.65	
Test statistic S	5.570	
Numerator DOF	4	
Denominator DOF	5	
$P(F \geq S)$	0.0438	*Reject H_0 at 5% significance level*

The next figure illustrates the results from plotting the group means from `manova1.tf1` using the profile analysis option, noting that error bars are not added as a multivariate distribution is assumed,



Profile analysis attempts to explore a common question that often arises in repeated measurements ANOVA namely, can two profiles be regarded as parallel. This amounts to testing if the sequential differences between adjacent means for groups i and j are equal, that is, if the slopes between adjacent treatments are constant across the two groups, so that the two profiles represent a common shape.

To do this, we first define the $m-1$ by m transformation matrix K by

$$K = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 & \dots \\ 0 & 1 & -1 & 0 & 0 & \dots \\ 0 & 0 & 1 & -1 & \dots & \\ \dots & \dots & \dots & \dots & \dots & \dots \end{pmatrix}.$$

Then a Hotelling's T^2 test is conducted using the pooled estimate for the covariance matrix $S_{ij} = [(n_i - 1)S_i + (n_j - 1)S_j]/(n_i + n_j - 2)$ and mean difference vector $d_{ij} = \bar{y}_i - \bar{y}_j$ according to

$$T^2 = \left(\frac{n_i n_j}{n_i + n_j}\right)(Kd_{ij})^T (KS_{ij}K^T)^{-1}(Kd_{ij})$$

and comparing the transformed statistic

$$\frac{n_i + n_j - m}{(n_i + n_j - 2)(m-1)} T^2 \sim F_{m-1, n_1+n_2-m}$$

to the corresponding F distribution.

Clearly, from the above table, the profiles are not parallel for the data in test file `manova1.tf1`.

6.6 Comparing groups

Tutorials and worked examples for simulation, curve fitting, statistical analysis, and plotting.
https://simfit.uk
https://simfit.org.uk
https://simfit.silverfrost.com

6.6.1 Canonical variates (discriminant functions)

Canonical variates is a technique used to transform multivariate data into new coordinates in order to highlight differences and similarities between groups of observations.

If MANOVA investigation suggests that at least one group mean vector differs from the the rest, it is usual to proceed to canonical variates analysis, although this technique can be also be used for data exploration when the assumption of multivariate normality with equal covariance matrices is not justified.

Example 1

From the main SIMFIT menu choose [Statistics], [Multivariate], then [Canonical variates], and read in the test file `manova1.tf4` from the `C:\Program Files\simfit\dem` folder, which contains the following data for three groups of three subjects, each with observations on three variables.

```
1   13.3   10.6   21.2
1   13.4    9.4   21.0
1   12.9   10.0   20.5
2   13.6   10.2   21.0
2   13.2    9.6   20.1
2   12.2    9.9   20.7
3   14.2   10.7   21.1
3   13.9   10.4   19.8
3   13.9   11.0   19.1
begin{values}
    14.0   11.0   22.0
    12.0    9.0   19.0
    13.0    9.0   20.0
end{values}
```

Column 1 has the group numbers as nondecreasing integers, while columns 2, 3, and 4 are observations on three variables. Because canonical variates are also used to see how closely additional unassigned observations compare to the defined groups of the training set, these can be added as additional values to the end of the data file as shown.

The table below shows the results from analyzing these data, and this is followed by a summary that will be explained in more detail later.

Results from analysis of `manova.tf4`: rank = 3

Correlations	Eigenvalues	Proportions	χ^2	$NDOF$	p
0.8826	3.5238	0.9795	7.9032	6	0.2453
0.2623	0.0739	0.0205	0.3564	2	0.8368

Canonical variate means

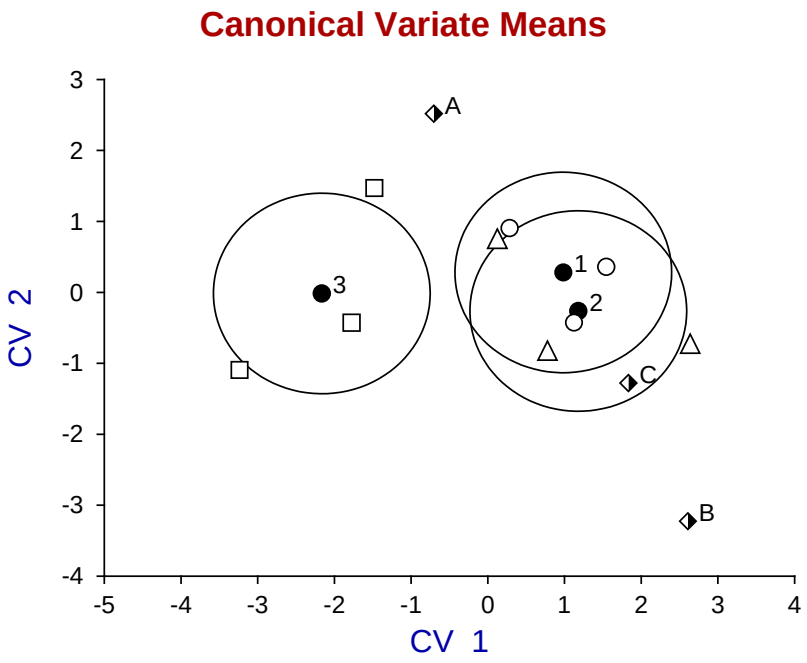
0.9841	0.2797
1.181	-0.2632
-2.165	-0.01642

Canonical coefficients

-1.707	0.7277
-1.348	0.3138
0.9327	1.220

- The number of correlations is the larger of the rank and the number of groups less one.
- The eigenvalues are for the within group sum of squares matrix, and these are used to estimate the proportion of variation explained by the canonical variates.
- The chi-square statistic is used to decide the number of canonical variates required to represent the data.
- The degrees of freedom and a p value for the significance of this chi-square statistic are presented.

Perhaps the most useful application of this technique is to plot the group means together with the data and 95% confidence regions in canonical variate space in order to visualize how close or how far apart the groups are. This is done for the first two canonical variates in the next figure.

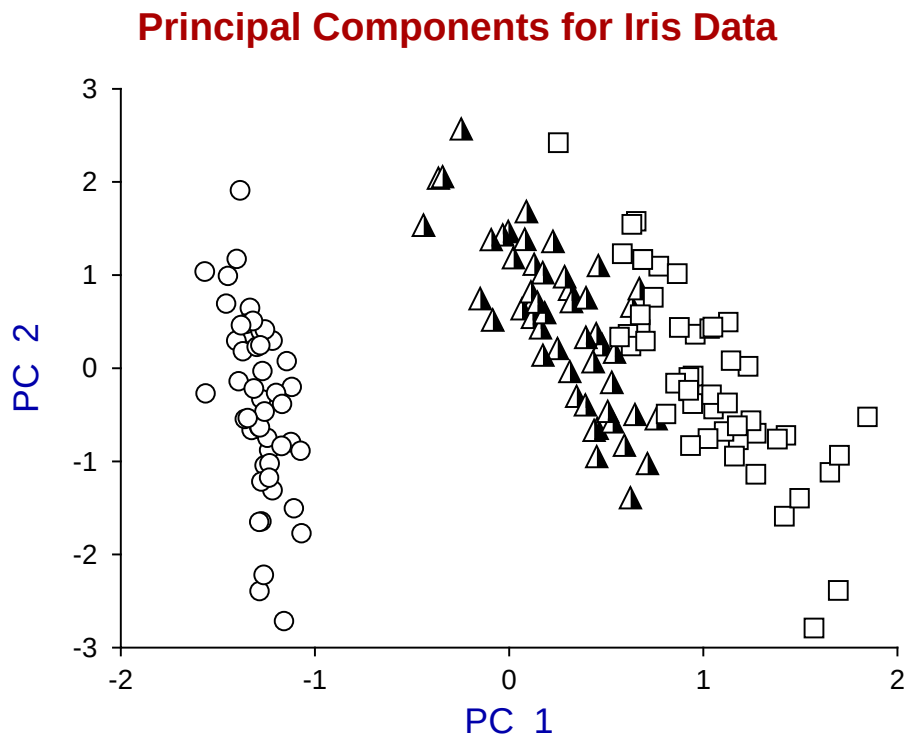


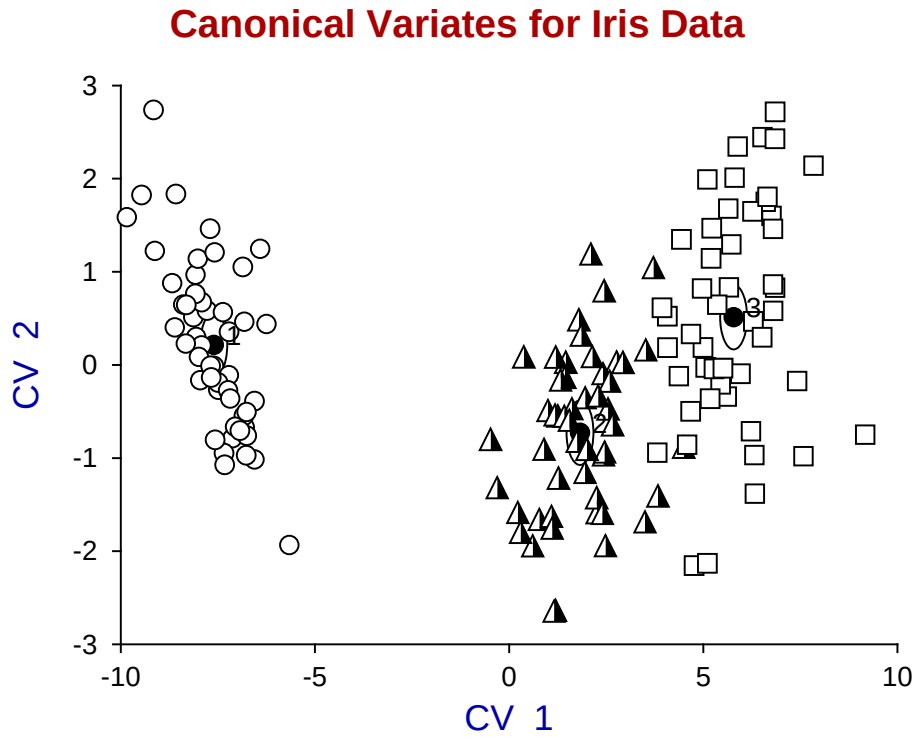
Using the option to edit plot parameters the above figure was constructed to show the labeled canonical variate means (filled circles) together with 95% confidence confidence regions, along with the original data and the three additional observations. Finally the ranges of data plotted were adjusted in order to display the confidence ranges as circles instead of ellipses.

From this graph it is evident that groups 1 and 2 (circles and triangles) are similar but both groups are distinct from group 3 (squares). Additional observation C (half filled diamonds) can be assigned the groups 1 and 2 but additional observations A and B do not belong to any of the training sets.

Example 2

To appreciate the use of canonical variates to distinguish groups with a larger data set, consider the next figures, which illustrate the famous Fisher Iris data set contained in `manova1.tf5` using the first two principal components, and also the first two canonical variates for comparison.





Theory

First of all, note that canonical variates, unlike principal components, are not simply obtained by a distance preserving rotation: the transformation is non-orthogonal and best represents the Mahalanobis distance between groups.

The confidence range

In the first figure of Example 1 we see the group means identified by the filled symbols labeled as 1, 2 and 3, each surrounded by a 95% confidence region, which in this case is circular as equally scaled physical distances are plotted along the axes. The canonical variates are uncorrelated and have unit variance so, assuming normality, the $100(1-\alpha)\%$ confidence region for the population mean is a circle radius

$$r = \sqrt{\chi^2_{\alpha,2}/n_i},$$

where group i has n_i observations and $\chi^2_{\alpha,2}$ is the value exceeded by $100\alpha\%$ of a chi-square distribution with 2 degrees of freedom.

Note that, alternatively, a circle radius $\sqrt{\chi^2_{\alpha,2}}$ can be plotted as this defines a tolerance region, i.e. the region within which $100(1-\alpha)\%$ of the whole population is expected to lie.

The additional observations

Also, the test file `manova1.tf4` has three other observations appended which are to be compared with the main groups in order to assign group membership, that is, to see to which of the main groups 1, 2 and 3 the extra observations should be assigned. The half-filled diamonds representing these are identified by the labels A, B and C which, like the identifying numbers 1, 2, and 3, are plotted automatically by SIMFIT to identify group means and extra data. In this case, as the data sets are small, the transformed observations from groups 1, 2 and 3 are also shown as circles, triangles and squares respectively, which is easily done by saving the coordinates from the plotted transforms of the observations in ASCII text files which are then added interactively as extra data files to the means plot.

The calculation of canonical variates

The aim of canonical variate analysis is to find the transformations a_i that maximize F_i, the ratios of B (the between group sum of squares and products matrices) to W (the within-group sum of squares and products matrix), i.e.

$$F_i = \frac{a_i^T B a_i/(g-1)}{a_i^T W a_i/(n-g)}$$

where there are g groups and n observations with m covariates each, so that $i = 1, 2, \ldots, l$ where l is the lesser of the number of groups minus one and the rank of the data matrix. The canonical variates are obtained by solving the symmetric eigenvalue problem

$$(B - \lambda^2 W)x = 0,$$

where the eigenvalues λ_i^2 define the ratios F_i, and the eigenvectors a_i corresponding to the λ_i^2 define the transformations. So, just as with principal components, a scree diagram of the eigenvalues in decreasing order indicates the proportion of the ratio of between-group to within-group variance captured by the canonical variates.

Note that the previous results table lists the rank k of the data matrix, the number of canonical variates $l = \min(k, g-1)$, the eigenvalues λ_i^2, the canonical correlations $\lambda_i^2/(1+\lambda_i^2)$, the proportions $\lambda_i^2/\sum_{j=1}^{l}\lambda_j^2$, the group means, the loadings, and the results of a chi-square test.

The number of canonical variates

It is important to realize that the first two canonical variates may be insufficient to represent the data adequately. A scree diagram can be plotted to estimate the minimum number required, or the eigenvalues, proportions, or chi-square statistics calculated from the data can be used.

For instance. If the data are assumed to be from a common multivariate distribution, then to test for a significant dimensionality greater than some level i, the statistic

$$\chi^2 = (n - 1 - g - (k - g)/2) \sum_{j=i+1}^{l} \log(1 + \lambda_j^2)$$

has an asymptotic chi-square distribution with $(k-i)(g-1-i)$ degrees of freedom. If the test is not significant for some level h, then the remaining tests for $i > h$ should be ignored. It should be noted that the group means and loadings are calculated for data after column centering and the canonical variates have within group variance equal to unity. Also, if the covariance matrices $\beta = B/(g-1)$ and $\omega = W/(n-g)$ are used, then $\omega^{-1}\beta = (n-g)W^{-1}B/(g-1)$, so eigenvectors of $W^{-1}B$ are the same as those of $\omega^{-1}\beta$, but eigenvalues of $W^{-1}B$ are $(g-1)/(n-g)$ times the corresponding eigenvalues of $\omega^{-1}\beta$.

In the iris plot of Example 2 there are only two canonical variates, so the canonical variates diagram is fully representative of the data set, and both techniques illustrate the distinct separation of group 1 (circles = setosa) from groups 2 (triangles = versicolor) and 3 (squares = virginica), and the lesser separation between groups 2 and 3.

Users of these techniques should always remember that, as eigenvectors are only defined up to an arbitrary scalar multiple and different matrices may be used in the principal component calculation, principal components and canonical variates may have to be reversed in sign and re-scaled to be consistent with calculations reported using software other than SIMFIT. To see how to compare extra data to groups involved in the calculations, the test file `manova1.tf4` should be examined.

6.6.2 Discriminant analysis: Mahalanobis distances

Discriminant analysis is based on comparing multivariate observations made with different groups of subjects in order to define the distances between the groups, and also to assign new observations to appropriate groups.

From the main SIMFIT menus choose [Statistics], [Multivariate, [Discriminant analysis] then read in the default test file `g03daf.tf1` which has the following data.

1	1.1314	2.4596
1	1.0986	0.2624
1	0.6419	-2.3026
1	1.3350	-3.2189
1	1.4110	0.0953
1	0.6419	-0.9163
2	2.1163	0.0000
2	1.3350	-1.6094
2	1.3610	-0.5108
2	2.0541	0.1823
2	2.2083	-0.5108
2	2.7344	1.2809
2	2.0412	0.4700
2	1.8718	-0.9163
2	1.7405	-0.9163
2	2.6101	0.4700
3	2.3224	1.8563
3	2.2192	2.0669
3	2.2618	1.1314
3	3.9853	0.9163
3	2.7600	2.0281

This data set has three groups, indicated by the nondecreasing integers in columns 1, for three types of Cushing's syndrome, the variables in columns 2 and 3 are logarithms of urinary excretion rates (mg/hr) for two steroid metabolites.

The following options are then available.

- Calculate the group sample means and the pooled sample means.
  The numerical values for the vectors of means can be displayed.
- Test for equality of the vectors of population means.
  If required, this can be done using the MANOVA options provided by SIMFIT.
- Test if all population variance-covariance matrices are equal.
  The results from discriminant analysis will differ depending on whether it is assumed that the variables all have the same population covariance matrix (as estimated from the pooled samples) or different covariance matrices (as estimated from the group samples).
- Calculate distances between the groups.
  The Mahalanobis distance matrix D^2_{ij} can be calculated assuming equal or unequal variance-covariance matrices.
- Plot the groups.
  The centroids can also be plotted to indicate the center of gravity of the groups while, for cases with more than two variables, the principal components can be plotted instead.

The results from such a systematic investigation are now presented.

First of all here are the group and pooled means followed by a MANOVA test for equality of means.

Table 1. Mean vectors

Group 1	1.0433	-0.6034
Group 2	2.0073	-0.2060
Group 3	2.7097	1.5998
Pooled	1.8991	0.1104

Table 2. MANOVA test for H_0: population mean vectors are equal

Number of groups	3				
Number of variables	2				
Number of observations	21				
Statistic	Value	Transform	$NDOF$	p	
Wilks lambda	0.3144	6.660	4, 34	0.0005	*Reject H_0 at 1% significance level*
Roys largest root	1.801				
Lawley-Hotelling T	1.937	8.231	4, 17	0.0006	*Reject H_0 at 1% significance level*
Pillais trace	0.7625				

A mean vector for a group is simply the vector consisting of sample means for each variable within that group, and the results suggest that the population mean vectors for these three groups are not the same, so that regarding the subjects as forming three distinct groups seems to be justified in this case.

The results in this next table for testing if the population variance-covariance matrices for the groups are identical suggests that we should consider rejecting the null hypothesis.

Table 3. Testing H_0: population variance-covariance matrices are equal

Number of groups	3	
Number of observations	21	
Number of variables	2	
Test statistic C	19.24	
Degrees of freedom	6	
$P(\chi^2 \geq C)$	0.0038	*Reject H_0 at 1% significance level*

Note that, in the following values for the Mahalanobis distances D_i^2 between groups, assuming a common population variance-covariance matrix leads to a symmetric distance matrix, but for unequal variance-covariance matrices the distance matrix is not symmetric.

Table 4. Mahalanobis distances

D_{ij}^2 assuming equal CV

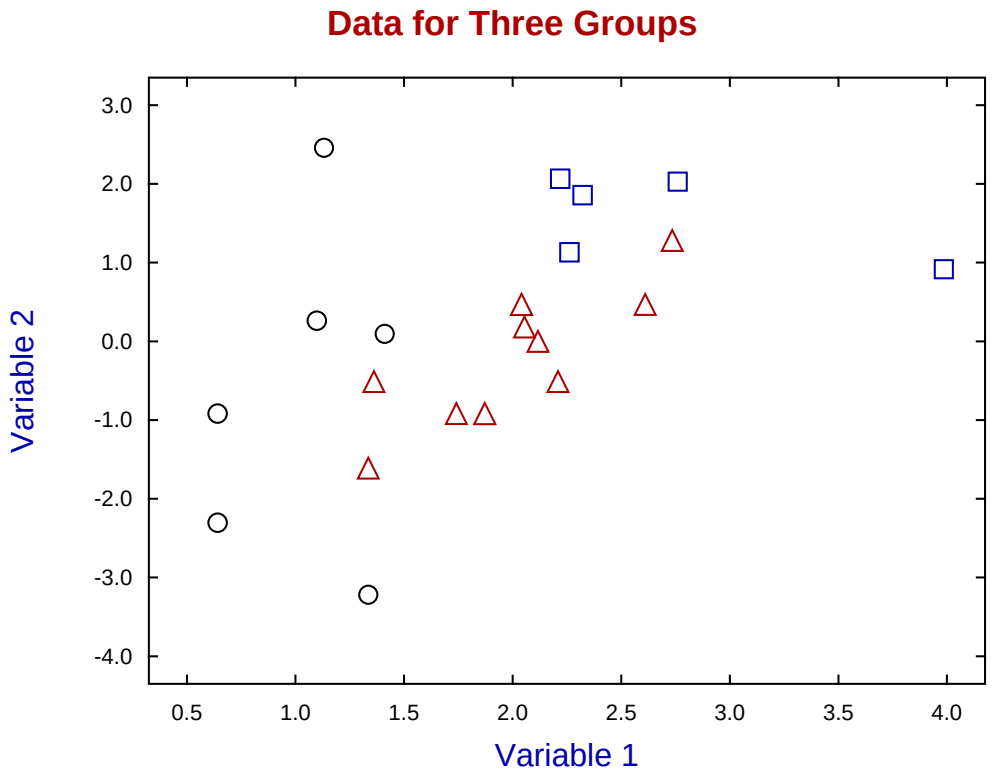
0.00000	3.58476	11.7998
3.58476	0.00000	3.25922
11.7998	3.25922	0.00000

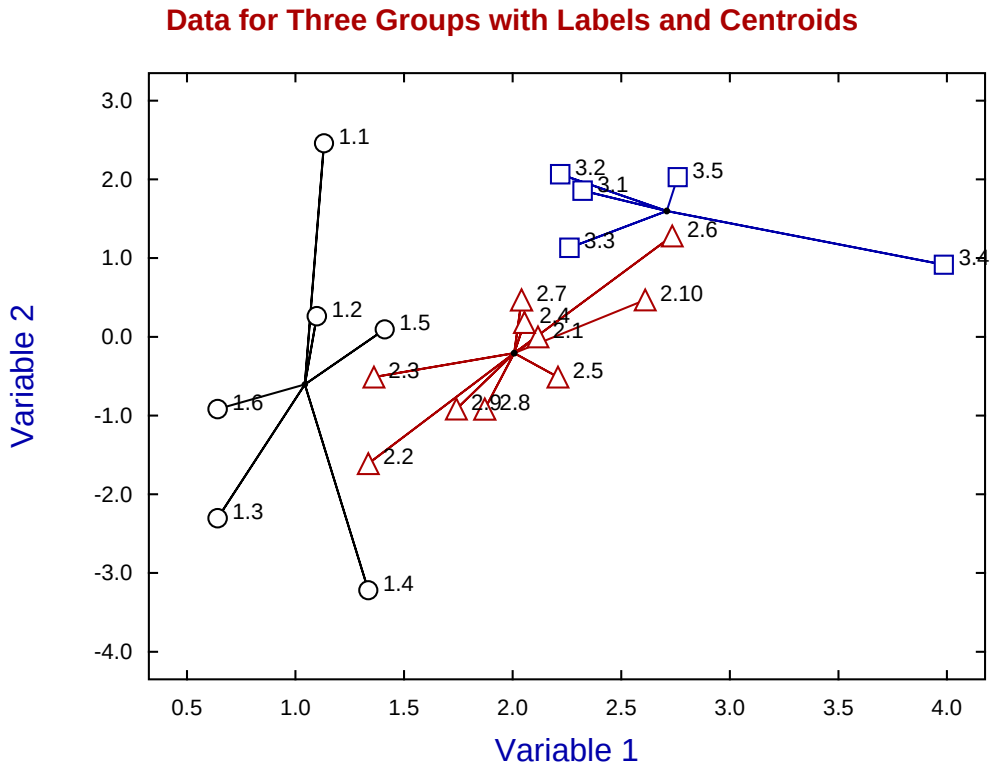
D_{ij}^2 assuming unequal CV

0.00000	9.55703	51.9737
8.51398	0.00000	25.2973
25.1215	4.71142	0.00000

Finally, the next figure displays the observations for the groups followed by the same data but with centroids added together with spokes to emphasize the groups.

In the present case there are only two variables so these can be used as axes but, for more than two variables, the option to plot principal components should be used


Data for Three Groups
Variable 2
Variable 1
3.0
2.0
1.0
0.0
-1.0
-2.0
-3.0
-4.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0


Data for Three Groups with Labels and Centroids
Variable 2
Variable 1
1.1
1.2
1.3
1.4
1.5
1.6
2.1
2.2
2.3
2.4
2.5
2.6
2.7
2.8
2.9
2.10
3.1
3.2
3.3
3.4
3.5

Theory

Defining the mean vector

Consider a group of X of size n with m variables as in

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1m} \\ x_{21} & x_{22} & \cdots & x_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ x_{n1} & x_{n2} & \cdots & x_{nm} \end{bmatrix}$$

for then the vector of column means

$$\bar{X} = \begin{bmatrix} \bar{x}_1, & \bar{x}_2, & \cdots & \bar{x}_m \end{bmatrix}^T$$

where $\bar{x}_j$ is the mean of column j is generally referred to as the mean vector as in Table 1 and Table 2. Alternatively, regarding the points as having unit mass, this is would be the center of mass or centroid of the data regarded as a multivariate swarm. For two mean vectors to be equal requires all corresponding component means to be equal.

Testing for equality of covariance matrices

The results from analyzing `g03daf.tf1` in Table 3 refer to using Box's test to analyze for equality of population covariance matrices. This depends on n the overall sample size, m the number of variables, g the number of groups, n_i the sample size in group i, S the pooled variance-covariance matrix with determinant $|S|$, S_i the within-group variance-covariance matrices with determinants $|S_i|$, and the likelihood ratio test statistic C defined by

$$C = M \left\{ (n-g) \log |S| - \sum_{i=1}^{g} (n_i - 1) \log |S_i| \right\}.$$

Here the multiplying factor M is

$$M = 1 - \frac{2m^2 + 3m - 1}{6(m+1)(g-1)} \left(\sum_{i=1}^{g} \frac{1}{n_i - 1} - \frac{1}{n-g} \right)$$

and, for large n, C is approximately distributed as χ^2 with $m(m+1)(g-1)/2$ degrees of freedom. Just as tests for equality of variances are not very robust, this test should be used with caution, and then only with large samples, i.e. $n_i >> m$.

The squared Mahalanobis distance between two groups

The squared Mahalanobis distance D_{ij}^2 between two group means $\bar{x}_i$ and $\bar{x}_j$ referred to in Table 4 can be defined as either

$$D_{ij}^2 = (\bar{x}_i - \bar{x}_j)^T S^{-1} (\bar{x}_i - \bar{x}_j)$$
$$\text{or } D_{ij}^2 = (\bar{x}_i - \bar{x}_j)^T S_j^{-1} (\bar{x}_i - \bar{x}_j)$$

depending on whether the covariance matrices are assumed to be equal, when the pooled estimate S is used and $D_{ij}^2 = D_{ji}^2$, or unequal when the group estimate S_j is used and $D_{ij}^2 \neq D_{ji}^2$. This distance is a useful quantitative measure of similarity between groups, but often there will be extra measurements which can then be appended to the data file, as with `g03daf.tf1`, so that the distance between measurement k and group j can be calculated as either

$$D_{kj}^2 = (x_k - \bar{x}_j)^T S^{-1} (x_k - \bar{x}_j)$$
$$\text{or } D_{kj}^2 = (x_k - \bar{x}_j)^T S_j^{-1} (x_k - \bar{x}_j).$$

6.6.3 Discriminant analysis: Allocating observations to training sets

Discriminant analysis provides methods for allocating new observations to existing training sets, i.e. groups that have been defined on the basis of previous investigations.

From the main SIMFIT menus choose [Statistics], [Multivariate, [Discriminant analysis] then read in the default test file `g03daf.tf1` which has the following data.

```
1    1.1314    2.4596
1    1.0986    0.2624
1    0.6419   -2.3026
1    1.3350   -3.2189
1    1.4110    0.0953
1    0.6419   -0.9163
2    2.1163    0.0000
2    1.3350   -1.6094
2    1.3610   -0.5108
2    2.0541    0.1823
2    2.2083   -0.5108
2    2.7344    1.2809
2    2.0412    0.4700
2    1.8718   -0.9163
2    1.7405   -0.9163
2    2.6101    0.4700
3    2.3224    1.8563
3    2.2192    2.0669
3    2.2618    1.1314
3    3.9853    0.9163
3    2.7600    2.0281
begin{values}
     1.6292   -0.9163
     2.5572    1.6094
     2.5649   -0.2231
     0.9555   -2.3026
     3.4012   -2.3026
     3.0204   -0.2231
end{values}
```

This data set has three groups, indicated by the nondecreasing integers in columns 1, for three types of Cushing's syndrome, the variables in columns 2 and 3 are logarithms of urinary excretion rates (mg/hr) for two steroid metabolites, and the values below the data are additional observations for allocating to one of the three groups. Such extra observations can also be added interactively and expanded training sets containing the newly assigned data can be saved as SIMFIT MANOVA type files.

Assigning new observations to groups defined by training sets can be made more objective by employing Bayesian techniques than by simply using distance measures, but only if a multivariate normal distribution can be assumed. For instance, the next table displays the results from assigning the six observations appended to `g03daf.tf1` to groups defined by using the data as a training set, under the assumption of unequal variance-covariance matrices and equal priors.

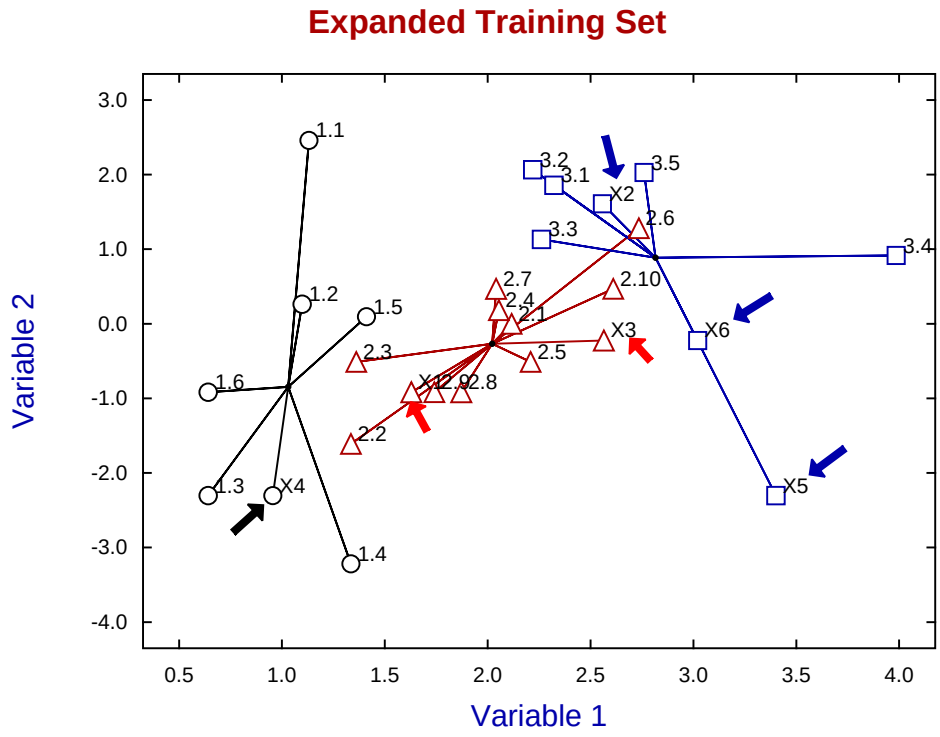
Method	Predictive
CV-mat	Unequal
Priors	Equal

Observation	Group-allocated
1	2
2	3
3	2
4	1
5	3
6	3

Posterior probabilities			Atypicality indices		
0.0939	0.9046	0.0015	0.5956	0.2539	0.9747
0.0047	0.1682	0.8270	0.9519	0.8360	0.0184
0.0186	0.9196	0.0618	0.9540	0.7966	0.9122
0.6969	0.3026	0.0005	0.2073	0.8599	0.9929
0.3174	0.0130	0.6696	0.9908	1.0000	0.9843
0.0323	0.3664	0.6013	0.9807	0.9779	0.8871

Plotting training sets and assigned observations

The next figure displays the training set from `g03daf.tf1`, together with the assignment of the extra observations appended to this test file as described previously. The additional observations allocated to the existing training set to create this expanded training set are emphasized by solid arrows, which confirm what the atypicality indices suggest, i.e. additional observation 5 is not particularly close to group 3.



Theory

The results from discriminant analysis will differ depending on whether it is assumed that the variables all have the same population covariance matrix so that this can be estimated from the pooled samples. Alternatively estimates from the separate groups can used. However, estimating variance-covariance matrices from multivariate samples requires sample sizes very much greater than the number of variables and, if this condition is not met, poor estimates can lead to incorrect allocations. So, unless sample sizes in all training sets are very much larger than the number of variables, it is probably best to use pooled estimates and ignore the tests suggesting unequal variance-covariance matrices.

The calculation is for g groups, each with n_j observations on m variables, and it is necessary to make assumptions about the identity or otherwise of the variance-covariance matrices, as well as assigning prior probabilities. Then Bayesian arguments lead to expressions for posterior probabilities q_j, under a variety of assumptions, given prior probabilities π_j as follows.

- Estimative with equal variance-covariance matrices (Linear discrimination)

$$\log q_j \propto -\tfrac{1}{2}D_{kj}^2 + \log \pi_j$$

- Estimative with unequal variance-covariance matrices (Quadratic discrimination)

$$\log q_j \propto -\tfrac{1}{2}D_{kj}^2 + \log \pi_j - \tfrac{1}{2}\log |S_j|$$

- Predictive with equal variance-covariance matrices

$$q_j \propto \frac{\pi_j}{((n_j+1)/n_j)^{m/2}\{1+[n_j/((n-g)(n_j+1))]D_{kj}^2\}^{(n-g+1)/2}}$$

- Predictive with unequal variance-covariance matrices

$$q_j \propto \frac{\pi_j \Gamma(n_j/2)}{\Gamma((n_j-m)/2)((n_j^2-1)/n_j)^{m/2}|S_j|^{1/2}\{1+(n_j/(n_j^2-1))D_{kj}^2\}^{n_j/2}}$$

Subsequently the posterior probabilities are normalized so that $\sum_{j=1}^{g} q_j = 1$ and the new observations are assigned to the groups with the greatest posterior probabilities. In this analysis the priors can be assumed to be all equal, proportional to sample size, or user defined. Also, atypicality indices I_j are computed to estimate how well an observation fits into an assigned group. These are

- Estimative with equal or unequal variance-covariance matrices

$$I_j = P(D_{kj}^2/2, m/2)$$

- Predictive with equal variance-covariance matrices

$$I_j = R(D_{kj}^2/(D_{kj}^2 + (n-g)(n_j-1)/n_j), m/2, (n-g-m+1)/2)$$

- Predictive with unequal variance-covariance matrices

$$I_j = R(D_{kj}^2/(D_{kj}^2 + (n_j^2-1)/n_j), m/2, (n_j-m)/2),$$

where $P(x,\alpha)$ is the incomplete gamma function, and $R(x,\alpha,\beta)$ is the incomplete beta function. Values of atypicality indices close to one for all groups suggest that the corresponding new observation does not fit well into any of the training sets, since one minus the atypicality index can be interpreted as the probability of encountering an observation as or more extreme than the one in question given the training set.

The assignment of extra observations to the training sets depends on the data transformation selected and variables suppressed or included in the analysis, and this must be considered when supplying extra observations interactively.

7 Survival analysis

Tutorials and worked examples for simulation, curve fitting, statistical analysis, and plotting.
https://simfit.uk
https://simfit.org.uk
https://simfit.silverfrost.com

7.1 Introduction

Survival analysis attempts to develop a statistical model for situations where a group is observed from time $t = 0$ onwards until a number of the subjects in the original group no longer survive. It can be used to model numerous situations ranging from the failure of machinery by wear and tear to the death of individuals due to disease or old age.

SimFIT provides several techniques to analyze the following types of survival data.

1. Estimates of proportions of a population surviving as a function of time are available by some technique which does not directly estimate the number surviving in a populations of known initial size, rather, proportions surviving are inferred by indirect techniques such as light scattering for bacterial density or enzyme assay for viable organisms. In such instances the estimated proportions are not binomial variables so fitting survival models directly by weighted least squares is justified, especially where destructive sampling has to be used so that autocorrelations are less problematical. Program **gcfit** is used in mode 2 for this type of fitting.

2. A population of individuals is observed and information on the times of censoring (i.e. leaving the group) or failure are recorded, but no covariates are measured. In this case, survival density functions, such as the Weibull model, can be fitted by maximum likelihood, and there are numerous statistical and graphical techniques to test for goodness of fit. Program **gcfit** is used in mode 3 for this type of fitting.

3. When there are covariates as well as survival times and censored data, then survival models can be fitted as generalized linear models. The SimFIT GLM simplified interface module is used for this type of analysis.

4. The Cox proportional hazards model does not attempt to fit a complete model, but a partial model can be fitted by the method of partial likelihood as long as the proportional hazards assumption is justified independently. Actually, after fitting by partial likelihood, a piece-wise hazard function can be estimated and residuals can then be calculated. The SimFIT GLM simplified interface module is also used for this type of analysis.

To summarize, in the context of survival analysis, the random survival time T with density $f(t)$, cumulative distribution function $F(t)$, survivor function $S(t)$, hazard function $h(t)$, and cumulative hazard function $H(t)$ are defined for $t \geq 0$ by

$$
\begin{aligned}
f(t) &\geq 0 \\
F(t) &= \int_0^t f(u)\,du \\
S(t) &= 1 - F(t) \\
h(t) &= f(t)/S(t) \\
H(t) &= \int_0^t h(u)\,du \\
f(t) &= h(t)\exp\{-H(t)\}.
\end{aligned}
$$

Clearly, the survivor function $S(t) = \exp\{-H(t)\}$ is the probability of surviving up to time t, which decreases monotonically from 1 to zero as t increases, while the hazard function is the probability of failure at t given survival up to this time. However, analysis is often complicated by left censoring when new individuals join the group at some $t > 0$, or right censoring when individuals leave the original group without failing. The alternative methods used to quantify the behavior of any particular group simply depend on the model assumed, while any predictions made from estimated parameters also depend on the size and homogeneity of the group under investigation in terms of covariates.

7.2 1-sample Kaplan-Meier survivor function

Given observations of some critical event such as survival, recovery from illness, failure of a machine, or death of a subject, as a function of time within a given group, the Kaplan-Meier or product moment nonparametric estimator for a suitable step function to model the survivor function has gained wide acceptance.

Example 1

From the main SimFIT menu choose [Statistics], [Time series and survival], then [Kaplan-Meier], and study the default test file `survive.tf2` which has the following format.

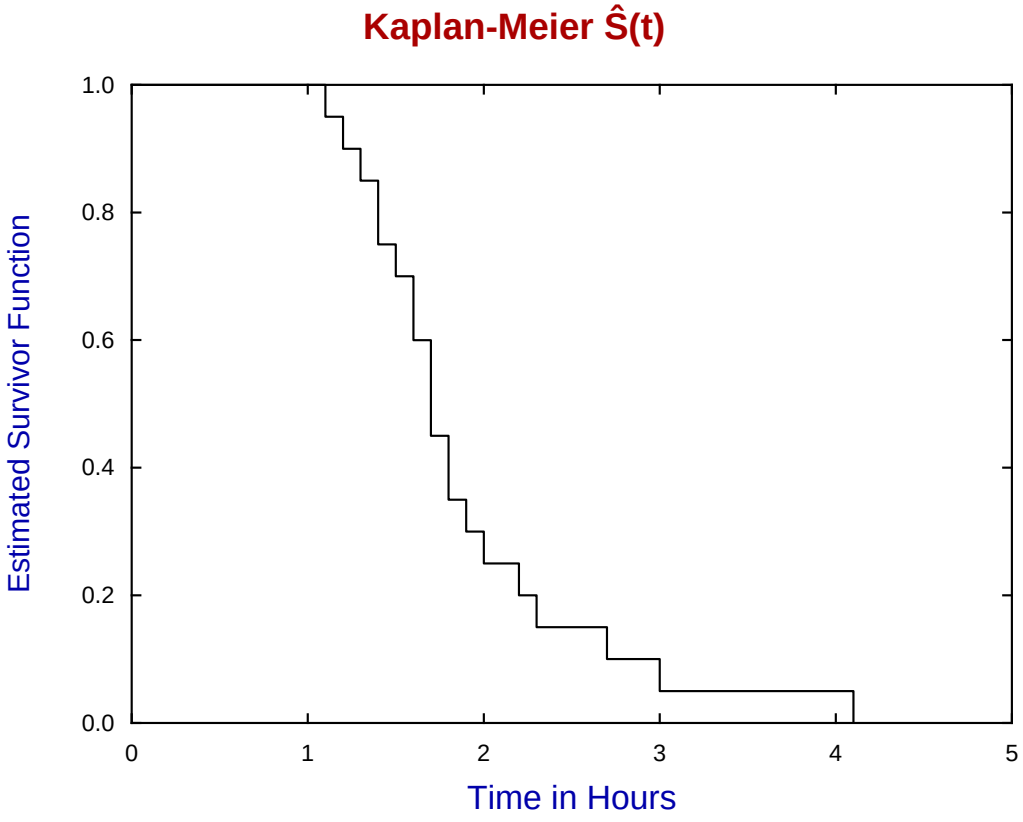
Time to relief	Censorship	Frequency
1.1	0	1
1.4	0	1
1.3	0	1
1.7	0	1
1.9	0	1
1.8	0	1
1.6	0	1
2.2	0	1
1.7	0	1
2.7	0	1
4.1	0	1
1.8	0	1
1.5	0	1
1.2	0	1
1.4	0	1
3.0	0	1
1.7	0	1
2.3	0	1
1.6	0	1
2.0	0	1

These data were for twenty patients taking an analgesic to relieve headache pain and the data have been formatted according to this scheme, where the critical event in this case is freedom from pain.

1. **First column**
   Time in hours (not necessarily ordered)
2. **Second column**
   Censoring code (0 = occurrence of the critical event, 1 = right-censored)
3. **Third column**
   Frequency of the observation
4. **Note**
   The starting sample size will be taken as the sum of all the frequencies in column 3. So subjects remaining at or after the last failure should be included as right-censored with the appropriate frequency. Failure to do this will lead to an underestimate of the starting size for the group and a biased estimator.

These data do not contain censored observations and all the subjects were observed until the headache ceased, so the estimated survivor function $\hat{S}(t)$ was as displayed in the next table and graph.

Time to relief	Estimate $\hat{S}(t)$	Standard Error
1.1	0.95	0.0487
1.2	0.90	0.0671
1.3	0.85	0.0798
1.4	0.75	0.0968
1.5	0.70	0.1025
1.6	0.60	0.1095
1.7	0.45	0.1112
1.8	0.35	0.1067
1.9	0.30	0.1025
2.0	0.25	0.0968
2.2	0.20	0.0894
2.3	0.15	0.0798
2.7	0.10	0.0671
3.0	0.05	0.0487
4.1	0.00	0.0000

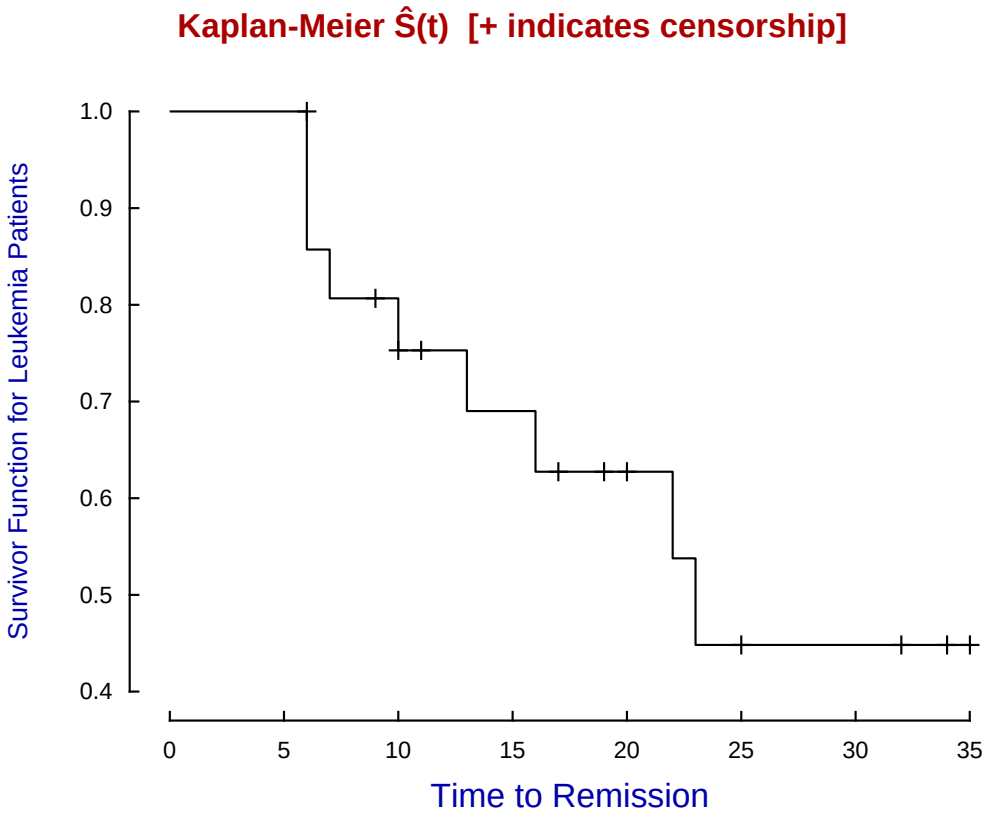


Example 2

The next table and graph are for the data contained in test file `survive.tf1` which is for time to remission in 21 leukemia patients.

This data set contains right-censored observations with a 1 instead of 0 in column two, and also records the number of replicates with frequencies of 2 or 3 instead of 1 in column three.

Time to remission	Censorship	Frequency
6	1	1
6	0	3
7	0	1
9	1	1
10	0	1
10	1	1
11	1	1
13	0	1
16	0	1
17	1	1
19	1	1
20	1	1
22	0	1
23	0	1
25	1	1
32	1	2
34	1	1
35	1	1



Note that the eleven points at times 6, 9, 10, 11, 17, 19, 20, 25, 32, 34, and 35 where loss by censoring happened are indicated in the above diagram by plus signs (+).

Theory

The nomenclature regarding the Kaplan-Meier estimator for a survivor function arose because it is most widely used in reliability studies, where machinery is operated until failure occurs, and in clinical studies where a group of patients under treatment is observed until some critical event like recovery from illness, relief from suffering, or death happens. It is often complicated by the fact that right censoring occurs, where a subject leaves the group without the critical event occurring, e.g. when a clinical trial is discontinued.

For these reasons it is well to remember that as $S(t) = 1 - F(t)$ then $F(t) = 1 - S(t)$ and there may be some occasions where it could be more logical to regard $F(t)$ as the *survivor function*.

Suppose that there are exactly k distinct times where at least one critical event, e.g. a failure occurred. Then the calculation is based on ordering these k distinct times for failure into an increasing sequence

$$t_1 < t_2 < t_3 < \cdots < t_k$$

and recording the number that failed at each time, but also taking note of the number lost at each time due to censoring.

To understand the method, note that, as the times t_i are distinct and ordered failure times, i.e. $t_{i-1} < t_i$, and the number in the sample that have not failed by time t_i is n_i, while the number that do fail is d_i, then the estimated probabilities of failure and survival at time t_i are given by

$$\hat{p}(\text{failure}) = d_i/n_i$$
$$\hat{p}(\text{survival}) = (n_i - d_i)/n_i.$$

The Kaplan-Meier product limit nonparametric estimate of the survivor function is then defined as a step function which is given in the interval t_i to t_{i+1} by the product of survival probabilities up to time t_i, that is

$$\hat{S}(t) = \prod_{j=1}^{i} \left(\frac{n_j - d_j}{n_j} \right)$$

with variance estimated by Greenwood's formula as

$$\hat{V}(\hat{S}(t)) = \hat{S}(t)^2 \sum_{j=1}^{i} \frac{d_j}{n_j(n_j - d_j)}.$$

It is understood in this calculation that, if failure and censoring occur at the same time, the failure is regarded as having taken place just before that time and the censoring just after it.

It should be pointed out that steps in the survivor function only occur at failure points and, in the absence of any censored points, the Kaplan-Meier estimate for the survivor function at $t = t_i$ is just the usual binomial parameter estimate.

7.3 1-sample Weibull survivor function

Often experimentalists prefer to fit a continuous parameteric survivor function to survival data in order to quantify the data by best-fit parameters instead a nonparameteric step-wise approximation, and the Weibull distribution is often used for this purpose.

From the main SIMFIT menu choose [Statistics], [Time series and survival], then [Kaplan-Meier], and study the default test file `survive.tf2` which has the following format.

Time to relief	Censorship	Frequency
1.1	0	1
1.4	0	1
1.3	0	1
1.7	0	1
1.9	0	1
1.8	0	1
1.6	0	1
2.2	0	1
1.7	0	1
2.7	0	1
4.1	0	1
1.8	0	1
1.5	0	1
1.2	0	1
1.4	0	1
3.0	0	1
1.7	0	1
2.3	0	1
1.6	0	1
2.0	0	1

These data were for twenty patients taking an analgesic to relieve headache pain and the data have been formatted according to this scheme, where the critical event in this case is freedom from pain.

1. **First column**
   Time in hours (not necessarily ordered)

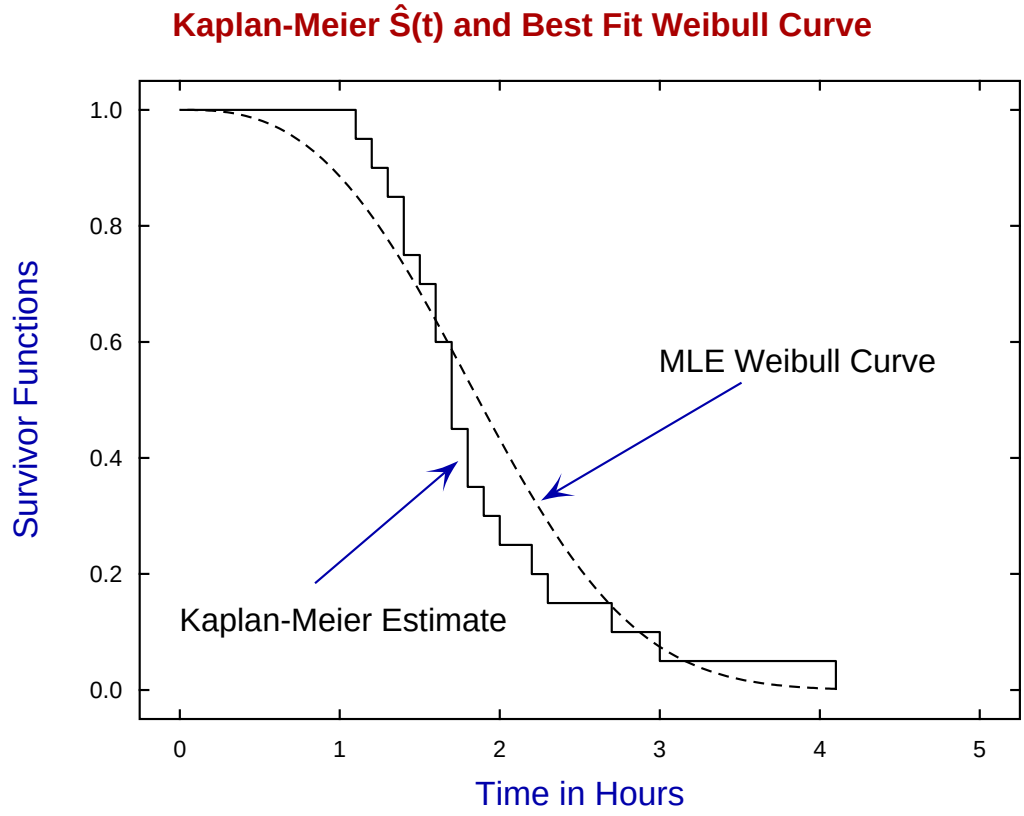
2. **Second column**
   Censoring code (0 = occurrence of the critical event, 1 = right-censored)

3. **Third column**
   Frequency of the observation

4. **Note**
   The starting sample size will be taken as the sum of all the frequencies in column 3. So subjects remaining at or after the last failure should be included as right-censored with the appropriate frequency. Failure to do this will lead to an underestimate of the starting size for the group and a biased estimator.

These data do not contain censored observations and all the subjects were observed until the headache ceased.

The Kaplan-Meier menu then provides the option to fit a Weibull survivor function and to overlay it on the Kaplan-Meier step function as in the next figure.



Of course the maximum likelihood Weibull estimate displayed is only one possible curve, and SIMFIT offers the opportunity to fit alternative curves by weighted least squares, or generalized linear interactive modeling (GLIM).

Tests for goodness of fit and for comparing parameter estimates require the actual parameter estimates and standard errors, i.e. estimates for their their standard errors, where the numerical values for these will depend upon the particular parameterization scheme used to define the Weibull model. So, as the next table shows, the Weibull model can be fitted with several parameterizations, and the one used by SIMFIT is designed to stabilize the calculations for the maximum likelihood estimate, as described subsequently.

Alternative MLE Weibull parameterizations

$S(t) = \exp[-\exp(\beta)t^B]$
$S(t) = \exp[-\lambda t^B]$
$S(t) = \exp[-(At)^B]$

Parameter	Value	Standard error	Lower95%cl	Upper95%cl	p
B	2.7870	0.4273	1.889	3.685	0.0000
β	-2.1073	0.4627	-3.079	-1.135	0.0002
λ	0.1216	0.0563	0.003	0.240	0.0444
A	0.4695	0.0401	0.385	0.554	0.0000
$t_{1/2}$	1.8675	0.1761	1.497	2.238	0.0000

Correlation coefficient(β, B) = -0.8755

Theory

To understand fitting the Weibull distribution, note that maximum likelihood parameter and standard error estimates are reported for three alternative parameterizations, namely

$$\begin{aligned} S(t) &= \exp(-\exp(\beta)t^B) \\ &= \exp(-\lambda t^B) \\ &= \exp(-(At)^B). \end{aligned}$$

Since the density and survivor function are

$$\begin{aligned} f(t) &= B\lambda t^{B-1}\exp(-\lambda t^B) \\ S(t) &= \exp(-\lambda t^B), \end{aligned}$$

and there are d failures and $n-d$ right censored observations, the likelihood function $l(B,\lambda)$ is proportional to the product of the d densities for the failures in the overall set of n observations and the survivor functions, that is

$$l(B,\lambda) \propto (B\lambda)^d \left(\prod_{i\in D} t_i^{B-1}\right) \exp\left(-\lambda \sum_{i=1}^{n} t_i^B\right)$$

where D is the set of failure times.

Actually, the log-likelihood function objective function

$$L(B,\beta) = d\log(B) + d\beta + (B-1)\sum_{i\in D}\log(t_i) - \exp(\beta)\sum_{i=1}^{n} t_i^B$$

with $\lambda = \exp(\beta)$ is better conditioned, so it is maximized and the partial derivatives

$$\begin{aligned} L_1 &= \partial L/\partial\beta \\ L_2 &= \partial L/\partial B \\ L_{11} &= \partial^2 L/\partial\beta^2 \\ L_{12} &= \partial^2 L/\partial B\partial\beta \\ L_{22} &= \partial^2 L/\partial B^2 \end{aligned}$$

are used to form the standard errors and correlation coefficient according to

$$\begin{aligned} \mathrm{se}(\hat{B}) &= \sqrt{-L_{11}/(L_{11}L_{22}-L_{12}^2)} \\ \mathrm{se}(\hat{\beta}) &= \sqrt{-L_{22}/(L_{11}L_{22}-L_{12}^2)} \\ \mathrm{corr}(\hat{B},\hat{\beta}) &= L_{12}/\sqrt{L_{11}L_{22}}. \end{aligned}$$

Given the maximum likelihood estimates for B and β, the estimates for A and λ are calculated and the standard errors worked out by the propagation of errors technique, while the half-time $t_{1/2}$, which is the time to 50% survival, is calculated numerically using the survivor function with parameters equal to the maximum likelihood estimates.

7.4 1-sample GLM survivor function with covariates

The general linear modeling technique (GLM) can be used to analyze survival data when there are covariates. It should be emphasized that GLM is a very powerful technique, but it must be used with great care as it requires more understanding from users than most analytical techniques. It defines an error type for the observations, and assumes that the distribution of mean values is described in a link function which is a linear combination of covariates. Further, additional model information in the form of data transformation, offsets, weights, and strata may be required. For this reason SIMFIT provides a simplified interface for fitting survival data which will now be described

From the main SIMFIT menu choose [Statistics], [Time series and survival], then [GLM], and study the default test file `cox.tf1` which has data from P. Feigel and M. Zelen Biometrics 21, 826-838 (1965) in the following format.

covariate x_1	covariate x_2	covariate x_3	observation y	time in weeks t	indicator s
0.8329	0	0	0	65.00	1
-0.2877	0	0	0	156.0	1
1.4586	0	0	0	100.0	1
0.9555	0	0	0	134.0	1
1.7918	0	0	0	16.00	1
2.3514	0	0	0	108.0	1
2.3026	0	0	0	121.0	1
2.8332	0	0	0	4.000	1
1.6864	0	0	0	39.00	1
1.9459	0	0	0	143.0	1
2.2407	0	0	0	56.00	1
3.4657	0	0	0	26.00	1
3.5553	0	0	0	22.00	1
4.6052	0	0	0	1.000	1
4.6052	0	0	0	1.000	1
3.9512	0	0	0	5.000	1
4.6052	0	0	0	65.00	1
1.4816	1	1.4816	0	56.00	1
1.0986	1	1.0986	0	65.00	1
1.3863	1	1.3863	0	17.00	1
0.4055	1	4.0547	0	7.000	1
2.1972	1	2.1972	0	16.00	1
1.6677	1	1.6677	0	22.00	1
2.3026	1	2.3026	0	3.000	1
2.9444	1	2.9444	0	4.000	1
3.2958	1	3.2958	0	2.000	1
3.3322	1	3.3322	0	3.000	1
3.4340	1	3.4340	0	8.000	1
3.2581	1	3.2581	0	4.000	1
3.0445	1	3.0445	0	3.000	1
4.3694	1	4.3694	0	30.00	1
4.6052	1	4.6052	0	4.000	1
4.6052	1	4.6052	0	43.00	1

The above data format, i.e. the meaning of these six columns of data for this example of GLM survival analysis with three covariates must be thoroughly understood as will be explained.

If there are m covariates the first m columns must be the covariates, then column $m + 1$ must be either 0 (failure) or 1 (right censoring), column $m + 2$ must be the nonnegative survival time, while column (m + 3) could be a default value of 1, or the weight for replicates or (in some case) the stratum indicator.

For these data the particular details are as follows.

- **Column 1:**
  covariate x_1 = log white blood cell count (in thousands)
- **Column 2:**
  covariate x_2 = AG-factor positive or negative (0 or 1)
- **Column 3:**
  covariate x_3 (in this special case $x_3 = x_1 x_2$ i.e. column 1 multiplied by column 2)
- **Column 4:**
  observation y (where $y = 0$ for failure, or $y = 1$ for censored)
- **Column 5:**
  t = survival time in weeks (t must be > 0)
- **Column 6:**
  $s = 1$ this should usually be 1. However, it could be interpreted as a weighting factor for replicates, except for the SimFIT advanced Cox regression procedure when it would be assumed to be the stratum indicator.

In order to fit survival data using generalized linear models (GLM) by maximum likelihood four components must be defined.

1. A random variable, say Y with mean $E(Y) = \mu$, and variance $V(Y)$
2. A set of covariates $x_1, x_2, \ldots, x_m$ recorded at the same time as Y
3. A link function $g(.)$ which is a function of μ
4. A linear predictor function of the covariates $\eta = \sum_{j=1}^{m} \beta_j x_i$

In addition it is supposed that the relationship between $E(Y)$ and η is

$$g(\mu) = \eta$$

and the fit is achieved by an iterative process.

As the GLM technique for fitting survival models is very complicated, requiring careful choices for the distribution of Y and the link function $g(.)$ as well as the calculation of offsets and use of data transformations, SimFIT supplies a simplified interface to handle the following four special cases.

- The exponential model
- The Weibull model
- The extreme distribution model
- The Cox model

The following table displays the results from analyzing the same test file `cox.tf1` using each of these models sequentially.

Model: exponential survival

No. parameters = 4, Rank = 4, No. points = 33, Deg. freedom = 29

Parameter	Value	Lower95%cl	Upper95%cl	Std.error	p	
$Constant$	-5.1498	-6.201	-4.098	0.5142	0.0000	
$B(1)$	0.4818	0.115	0.849	0.1795	0.0119	
$B(2)$	1.8705	0.374	3.367	0.7317	0.0161	
$B(3)$	-0.3278	-0.831	0.175	0.2460	0.1931	**

Deviance = 38.55, A = 1

Model: Weibull survival

No. parameters = 4, Rank = 4, No. points = 33, Deg. freedom = 29

Parameter	Value	Lower95%cl	Upper95%cl	Std.error	p	
$Constant$	-5.0405	-6.182	-3.899	0.5580	0.0000	
$B(1)$	0.4761	0.108	0.844	0.1800	0.0131	
$B(2)$	1.8413	0.338	3.344	0.7349	0.0181	
$B(3)$	-0.3244	-0.829	0.180	0.2465	0.1985	**
α	0.9777	0.889	1.066	0.0434	0.0000	

Deviance = 37.06

Deviance - $2n\log[alpha] = 38.55$

Model: Extreme value survival

No. parameters = 4, Rank = 4, No. points = 33, Deg. freedom = 29

Parameter	Value	Lower95%cl	Upper95%cl	Std.error	p
$Constant$	-5.2457	-6.502	-3.989	0.6143	0.0000
$B(1)$	0.9024	0.520	1.284	0.1868	0.0000
$B(2)$	3.8711	2.272	5.471	0.7821	0.0000
$B(3)$	-0.7195	-1.241	-0.198	0.2549	0.0085
α	0.0344	0.030	0.039	0.0020	0.0000

Deviance = 35.69

Deviance - $2n\log[alpha] = 258.1$

Model: Cox proportional hazards

Deviance = 131.48, Number of time points = 33

Parameter	Estimate	Score	Lower95%cl	Upper95%cl	Std.error	p	
$B(1)$	0.7325	5.138E-06	0.248	1.217	0.2371	0.0043	
$B(2)$	2.7557	1.886E-06	0.731	4.780	0.9913	0.0093	
$B(3)$	-0.5792	5.062E-06	-1.188	0.030	0.2981	0.0615	*

It is very difficult to check goodness of fit when using the simplified GLM procedure in a situation where, as in this case, the number of covariates is greater than zero, because only a limited number of techniques are available for checking the deviance residuals as the technique is not simply estimating the parameters of a theoretical equation for survival as a function of time. The most useful technique is probably to examine the half-normal residuals plot for apparent linearity. Another indication is the final deviance, and the pattern of convergence displayed during the iteration to find the minimum deviance. Again, the statistical significance of the parameter estimates should be taken into account. The p values reported in the above table refer to a, approximate two-tailed t test on the ratio of parameter estimate to the corresponding standard error in order to test the null hypothesis

$$H_0: \text{ The population parameter is not significantly different from zero.}$$

In other words, a p value less than 0.05 suggests that the parameter estimate could be meaningful, i.e. the corresponding parameter has been estimated reasonably well and it seems to be significantly different from zero. However, when p values exceed 0.05 this is indicated by stars as in the above table, drawing attention to the fact that the 95% confidence region for that parameter includes zero.

Theory

Many survival models can be fitted to N_u uncensored together with N_r right censored survival times with associated explanatory variables using the GLM technique from SimFIT programs **linfit**, **gcfit** in mode 4, or **simstat**.

For instance, the SimFIT simplified GLM interface allows you to read in data for the covariates, x, the variable y which can be either 1 for right-censoring or 0 for failure, together with the times t in order to fit survival models. With a density $f(t)$, survivor function $S(t) = 1 - F(t)$ and hazard function $h(t) = f(t)/S(t)$ a proportional hazards model is assumed for $t \geq 0$ with

$$\begin{aligned} h(t_i) &= \lambda(t_i)\exp(\sum_j \beta_j x_{ij}) \\ &= \lambda(t_i)\exp(\beta^T x_i) \\ \Lambda(t) &= \int_0^t \lambda(u)\,du \\ f(t) &= \lambda(t)\exp(\beta^T x - \Lambda(t)\exp(\beta^T x)) \\ S(t) &= \exp(-\Lambda(t)\exp(\beta^T x)). \end{aligned}$$

The SimFIT comprehensive GLM procedure allows almost any model to be fitted to survival data, but it requires that users must understand the numerous choices that have to be made concerning distributions to be assumed, starting estimates to provide, link functions required, offsets that have to be provided, etc.

For these reasons the SimFIT simplified GLM interface can fit several survival models using the appropriate choices for error distribution, link function, offset, data transformation, etc. required, as long as data are provided in the format demonstrated for the SimFIT test file `cox.tf1`.

The exponential survival model

The exponential model has constant hazard and is particularly easy to fit, since

$$\begin{aligned} \eta &= \beta^T x \\ f(t) &= \exp(\eta - t\exp(\eta)) \\ F(t) &= 1 - \exp(-t\exp(\eta)) \\ \lambda(t) &= 1 \\ \Lambda(t) &= t \\ h(t) &= \exp(\eta) \\ \text{and } E(t) &= \exp(-\eta), \end{aligned}$$

so this simply involves fitting a GLM model with Poisson error type, a log link, and a calculated offset of $\log(t)$.

The selection of a Poisson error type, the log link and the calculation of offsets are all done automatically by the simplified interface from the data provided, as will be appreciated on fitting the test file `cox.tf1`. It should be emphasized that the values for y in the simplified GLM procedure for survival analysis must be either $y = 0$ for failure or $y = 1$ for right censoring, and the actual time for failure t must be supplied paired with the y values.

Internally, the SimFIT simplified GLM interface reverses the y values to define the Poisson variables and uses the t values to calculate offsets automatically. Users who wish to use the advanced GLM interface for survival analysis must be careful to declare the Poisson variables correctly and provide the appropriate offsets as offset vectors.

The Weibull survival model

Weibull survival is similarly easy to fit, but is much more versatile than the exponential model on account of the extra shape parameter α as in the following equations.

$$\begin{aligned} f(t) &= \alpha t^{\alpha-1} \exp(\eta - t^{\alpha} \exp(\eta)) \\ F(t) &= 1 - \exp(-t \exp(\eta)) \\ \lambda(t) &= \alpha t^{\alpha-1} \\ \Lambda(t) &= t^{\alpha} \\ h(t) &= \alpha t^{\alpha-1} \exp(\eta) \\ E(t) &= \Gamma(1 + 1/\alpha) \exp(-\eta/\alpha). \end{aligned}$$

However, this time, the offset is $\alpha \log(t)$, where α has to be estimated iteratively and the covariance matrix subsequently adjusted to allow for the extra parameter α that has been estimated. The iteration to estimate α and covariance matrix adjustments are done automatically by the SimFIT simplified GLM interface, and the deviance is also adjusted by a term $-2n \log \hat{\alpha}$.

The extreme value survival model

Extreme value survival is defined by

$$f(t) = \alpha \exp(\alpha t) \exp(\eta - \exp(\alpha t + \eta))$$

which is easily fitted, as it is transformed by $u = \exp(t)$ into Weibull form, and so can be fitted as a Weibull model using t instead of $\log(t)$ as offset. However it is not so useful as a model since the hazard increases exponentially and the density is skewed to the left.

The Cox proportional hazards model

This model assumes an arbitrary baseline hazard function $\lambda_0(t)$ so that the hazard function is

$$h(t) = \lambda_0(t) \exp(\eta).$$

It should first be noted that Cox regression techniques may often yield slightly different parameter estimates, as these will often depend on the starting estimates, and also since there are alternative procedures for allowing for ties in the data. In order to allow for Cox's exact treatment of ties in the data, i.e., more than one failure or censoring at each time point, this model is fitted by the SimFIT GLM techniques after first calculating the risk sets at failure times t_i, that is, the sets of subjects that fail or are censored at time t_i plus those who survive beyond time t_i. Then the model is fitted using the technique for conditional logistic analysis of stratified data. The model does not involve calculating an explicit constant as that is subsumed into the arbitrary baseline function. However, the model can accommodate strata in two ways. With just a few strata, dummy indicator variables can be defined as in test files `cox.tf2` and `cox.tf3` but, with large numbers of strata, data should be prepared as for `cox.tf4`.

As an example, consider the results shown in the previous table from fitting an exponential, Weibull, then Cox model to data in the test file `cox.tf1`. In this case there is little improvement from fitting a Weibull model after an exponential model, as shown by the deviances and half normal residuals plots. The deviances from the full models (exponential, Weibull, extreme value) can be compared for goodness of fit, but they can not be compared directly to the Cox deviance.

7.5 2-sample Mantel-Haenszel log-rank test

The Mantel-Haenszel test and related procedures are used to compare two sets of survival data and to test for the suitability of the exponential, Weibull, Gompertz, Cox, and extreme value survival models.

From the main SIMFIT menu choose [Statistics], [Time series and survival], then [Kaplan-Meier] for two samples, and study the default test files `survive.tf3` and `survive.tf4` with data for remission from Leukemia from Frierich et al, Blood, 21, 699-716, 1963 in the following formats.

`survive.tf3`: 6-MP data

Time	Code	Number
6	0	3
6	1	1
7	0	1
9	1	1
10	0	1
10	1	1
11	1	1
13	0	1
16	0	1
17	1	1
19	1	1
20	1	1
22	0	1
23	0	1
25	1	1
32	1	2
34	1	1
35	1	1

`survive.tf4`: Placebo data

1	0	2
2	0	2
3	0	1
4	0	2
5	0	2
8	0	4
11	0	2
12	0	2
15	0	1
17	0	1
22	0	1
23	0	1

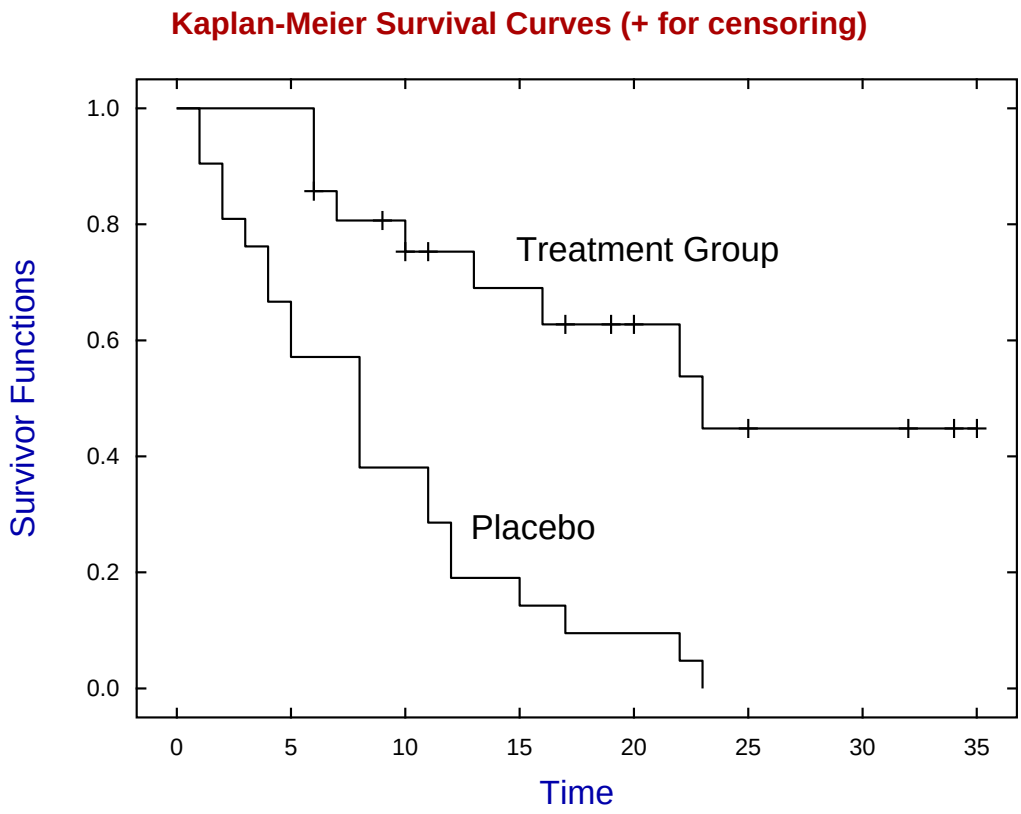
- Column 1: time (not necessarily ordered)
- Column 2: censoring code (0 = failure, 1 = right-censored)
- Column 3: frequency of the observation
  Note: The starting sample size will be taken as the sum of all the frequencies in column 3. So subjects remaining at or after the last failure should be included as right-censored with the appropriate frequency.

The results from the Mantel-Haenszel log-rank test are recorded in the next table.

For `survive.tf3`: 9 failures, 12 censored
For `survive.tf4`: 21 failures, 0 censored
$H_0 : h_A(t) = h_B(t)$ (equal hazards)
$H_1 : h_A(t) = \theta h_B(t)$ (proportional hazards)

QMH test statistic	16.79	
$P(\chi^2 \geq QMH)$	0.0000	*Reject H_0 at 1% significance level*
Estimate for θ	0.1915	
95% confidence range	0.0828, 0.4429	

The conclusion that the two groups differ significantly is reinforced by the next figure showing the Kaplan-Meier survivor functions, including censored data, from this analysis.

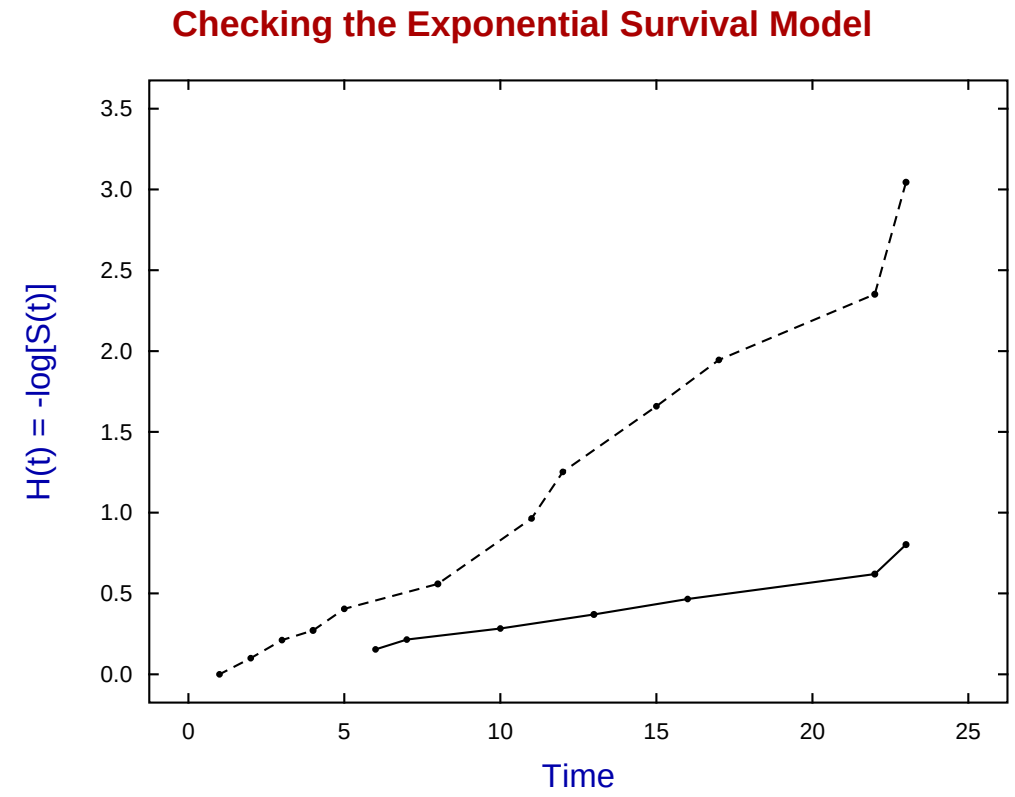


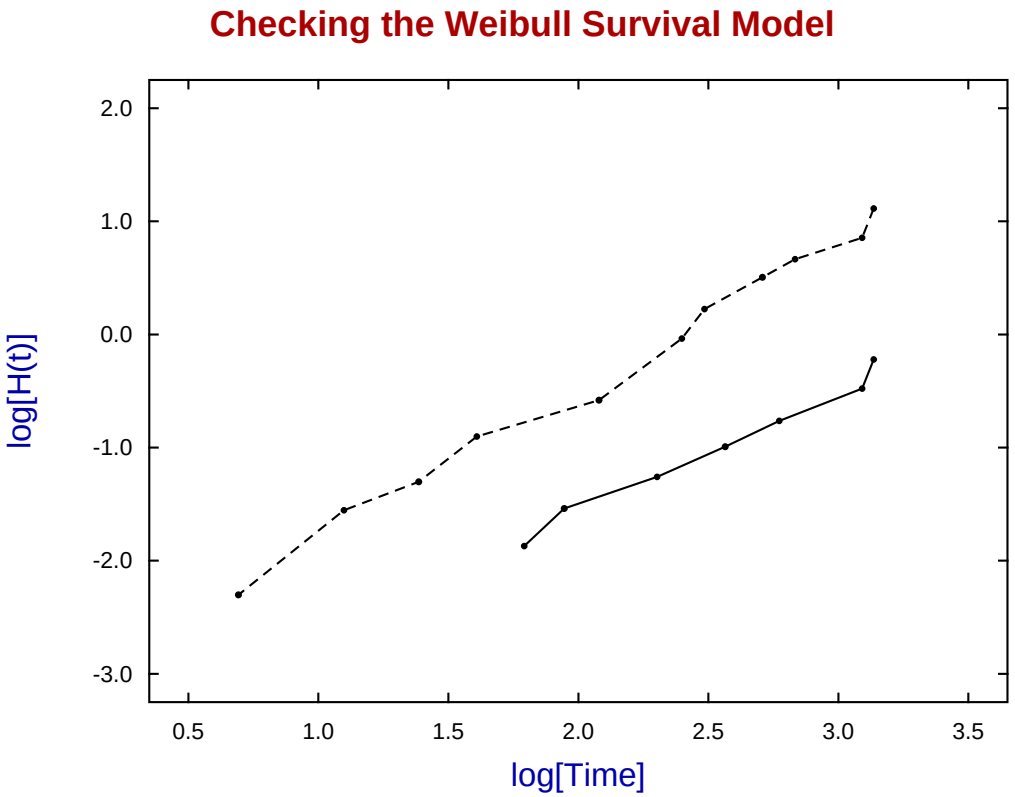
Also, various graphs can be plotted to explore the form of the estimated hazard function $\hat{h}(t)$ and estimated cumulative hazard function $\hat{H}(t)$ for the commonly used models based on the identities

$$\begin{aligned}
\text{Exponential} &: H(t) = At \\
\text{Weibull} &: \log(H(t)) = \log A^B + B \log t \\
\text{Gompertz} &: \log(h(t)) = \log B + At \\
\text{Extreme value} &: \log(H(t)) = \alpha(t - \beta).
\end{aligned}$$

For instance, $\hat{H}(t)$ would be linear for the exponential model, for the Weibull distribution a plot of $\log(-\log(\hat{S}(t))$ against $\log t$ should be linear, while the proportional hazards assumption would merely alter the constant term since, for $h(t) = \theta AB(At)^{B-1}$,

$$\log(-\log(S(t)) = \log\theta + \log A^B + B \log t.$$


Checking the Exponential Survival Model
H(t) = -log[S(t)]
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
5
10
15
20
25
Time


Checking the Weibull Survival Model
log[H(t)]
2.0
1.0
0.0
-1.0
-2.0
-3.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
log[Time]

Theory

To understand the graphical and statistical tests used to compare two samples, and to appreciate the results displayed in the previous results table, consider the relationship between the cumulative hazard function $H(t)$ and the hazard function $h(t)$ defined as follows

$$\begin{aligned} h(t) &= f(t)/S(t) \\ H(t) &= \int_0^t h(u)\,du \\ &= -\log(S(t)). \end{aligned}$$

Testing for the presence of a constant of proportionality in the proportional hazards assumption amounts to testing the value of θ with respect to unity. If the confidence limits in the results table enclose 1, this can be taken as suggesting equality of the two hazard functions, and hence equality of the two distributions, since equal hazards implies equal distributions.

The QMH statistic given in the results table can be used in a chi-square test with one degree of freedom for equality of distributions, and it arises by considering the 2 by 2 contingency tables at each distinct time point t_j of the following type.

	Died	Survived	Total
Group A	d_{jA}	$n_{jA} - d_{jA}$	n_{jA}
Group B	d_{jB}	$n_{jB} - d_{jB}$	n_{jB}
Total	d_j	$n_j - d_j$	n_j

Here the total number at risk n_j at time t_j also includes subjects subsequently censored, while the numbers d_{jA} and d_{jB} actually dying can be used to estimate expectations and variances such as

$$\begin{aligned} E(d_{jA}) &= n_{jA}d_j/n_j \\ V(d_{jA}) &= \frac{d_j(n_j - d_j)n_{jA}n_{jB}}{n_j^2(n_j - 1)}. \end{aligned}$$

Now, using the sums

$$\begin{aligned} O_A &= \sum d_{jA} \\ E_A &= \sum E(d_{jA}) \\ V_A &= \sum V(d_{jA}) \end{aligned}$$

as in the Mantel-Haenszel test, the log rank statistic can be calculated as

$$QMH = \frac{(O_A - E_A)^2}{V_A}.$$

Clearly, the graphs, the value of θ with 95% confidence range not enclosing 1, and the chi-square test with one degree of freedom all support the hypothesis that the the assumption of a Weibull distribution with proportional hazards but not equal hazards cannot be rejected with these data.

7.6 n-sample Cox regression

Cox regression can be used with data sets containing strata with covariates where it is not convenient to use a nonparametric method or a fully defined statistical model. Rather it makes the somewhat restrictive assumption of proportional hazards.

From the main SIMFIT menu choose [Statistics], [Time series and survival], then [Cox regression], and study the default test file `cox.tf4` which has three covariates and three strata as shown next.

x_1	x_2	x_3	y	t	s
0	0	0	1	1.0072	2
1	1	0	0	0.0209	1
0	1	0	0	0.7954	1
1	1	1	0	0.0582	2
0	0	1	1	0.0611	2
1	0	0	0	0.1750	2
1	0	1	0	0.2593	2
0	1	0	0	0.9463	1
0	1	1	1	0.0898	3
1	0	1	0	0.0787	2
0	0	1	0	0.1378	3
0	0	0	1	0.6303	2
1	0	0	1	0.2115	1
1	1	1	1	0.1085	3
0	0	0	1	0.5227	2
0	0	1	0	0.0164	3
1	1	0	0	0.6804	1
0	1	0	0	1.1091	2
0	0	0	0	0.0154	1
1	1	1	0	0.0816	2
1	0	1	0	0.4498	3
0	0	0	1	0.0847	2
1	1	1	0	1.0198	1
1	1	1	0	0.0607	2
0	0	0	0	0.0968	2
1	0	1	1	0.2083	2
0	0	0	1	5.0050	1
0	0	1	0	0.0243	2
1	0	0	1	1.0054	3
1	0	0	1	0.1810	1
0	1	1	1	0.0512	3
0	1	0	1	0.2579	2
1	0	0	1	0.5309	2
0	0	1	0	0.2753	3
0	1	1	0	0.1252	3
1	0	1	0	0.0664	3
0	1	0	1	0.6108	2
0	0	1	0	0.0187	1
0	0	1	0	0.1026	1
1	0	1	0	0.1016	2
0	0	1	0	0.4314	1

Continued on next page

x_1	x_2	x_3	y	t	s
1	0	1	0	0.7174	1
0	0	1	1	0.2297	1
1	1	0	1	0.0346	2
0	1	1	1	0.0478	2
0	1	0	0	0.1261	2
0	1	0	1	0.0827	3
1	0	0	0	0.8903	1
0	0	1	0	0.2746	1
1	0	0	1	0.9722	1

Note that Cox regression data files for m covariates must be formatted as follows $x_1, x_2, ..., x_m, y, t, s$. Here, for example, with test file `cox.tf4` we have the following format.

- column 1, 2, and 3: covariates x_1, x_2, and x_3
- column 4: $y = 0$ (failure) or $y = 1$ (right censored)
- column 5: time t
- column 6: strata s

The results table is shown next.

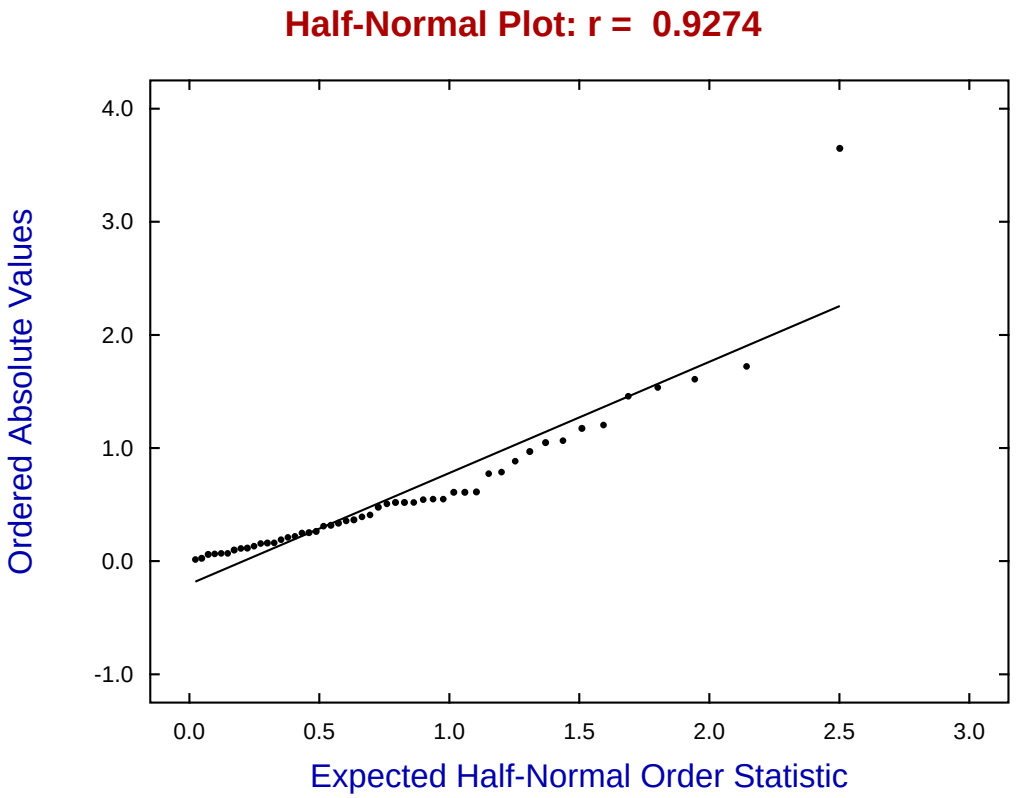
Deviance = 109.25, Number of time points = 50

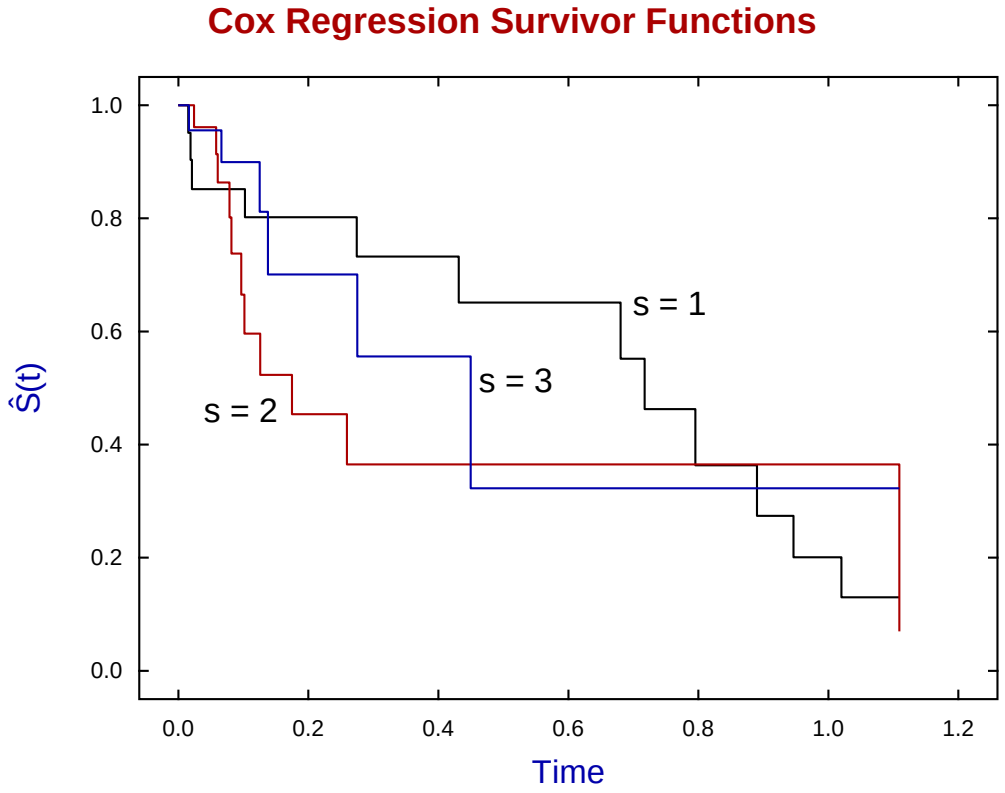
$B(i)$	Estimate	Score	Lower95%cl	Upper95%cl	Std.error	p	
1	-0.4893	8.156E-05	-1.423	0.445	0.464	0.2973	***
2	0.1609	-2.865E-05	-0.724	1.046	0.440	0.7162	***
3	1.5749	2.992E-04	0.562	2.588	0.504	0.0030	

Before proceeding to demonstrate further features of the SIMFIT Cox regression procedure it is necessary to caution about the fact that analyzing the same data using different software packages may give differing results. This is inevitable with all nonlinear iterative methods and is because of several factors.

1. The Cox regression procedure does not completely specify a unique statistical model.
2. The solution point found is not unique but will depend on the method used and the starting values used.
3. The results obtained will depend on the technique used to deal with ties.
4. It is fairly common to find that some of the parameters are not well defined, i.e. not statistically different from zero as shown by stars in the last column.
5. The scores are derived from the partial derivatives estimated at the solution point so any values much less than about 10^{-6} can be regarded as effectively undefined due to rounding errors.

Probably the easiest way to check the goodness of fit is to inspect the half-normal residuals plot, while to compare strata the collected survivor function estimates should be viewed. These two plots for the current analysis are shown next.


Half-Normal Plot: r = 0.9274
Ordered Absolute Values
4.0
3.0
2.0
1.0
0.0
-1.0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
Expected Half-Normal Order Statistic


Cox Regression Survivor Functions
Ŝ(t)
1.0
0.8
0.6
0.4
0.2
0.0
s = 1
s = 3
s = 2
0.0
0.2
0.4
0.6
0.8
1.0
1.2
Time

Theory

It should be pointed out that parameter estimates using the comprehensive Cox procedure may be slightly different from parameter estimates obtained by the GLM procedure if there are ties in the data, as the Breslow approximation for ties may sometimes be used by the comprehensive procedure, unlike the Cox exact method which is employed by the GLM procedures.

Another advantage of the comprehensive procedure is that experienced users can input a vector of offsets, as the assumed model is actually

$$\lambda(t, x) = \lambda_0(t) \exp(\beta^T x + \omega)$$

for parameters β, covariates x and offset ω.

Then the maximum likelihood estimates for β are obtained by maximizing the Kalbfleisch and Prentice approximate marginal likelihood

$$L = \prod_{i=1}^{n_d} \frac{\exp(\beta^T s_i + \omega_i)}{[\sum_{l \in R(t_{(i)})} \exp(\beta^T x_l + \omega_l)]^{d_i}}$$

where, n_d is the number of distinct failure times, s_i is the sum of the covariates of individuals observed to fail at $t_{(i)}$, and $R(t_{(i)})$ is the set of individuals at risk just prior to $t_{(i)}$.

In the case of multiple strata, the likelihood function is taken to be the product of such expressions, one for each stratum. For example, with ν strata, the marginal likelihood will be

$$L = \prod_{k=1}^{\nu} L_k.$$

Once parameters have been estimated the survivor function $\exp(-\hat{H}(t_{(i)})$ and residuals $r(t_l)$ are then calculated using

$$\begin{aligned} \hat{H}(t_{(i)}) &= \sum_{t_j \le t_i} \left(\frac{d_j}{\sum_{l \in R(t_{(j)})} \exp(\hat{\beta}^T x_l + \omega_l)} \right) \\ r(t_l) &= \hat{H}(t_l) \exp(\hat{\beta}^T x_l + \omega_l), \end{aligned}$$

where there are d_j failures at t_j.

Note that the deviance is minus twice the log of marginal likelihood and the significance of nested models with different parameters contributing can be assessed by chi-square tests, as an alternative to the two-tailed t test given in the results table. Also, stratum differences (i.e. differences between groups) can be examined using the log-ranks test.

8 Curve and surface fitting

Tutorials and worked examples for simulation, curve fitting, statistical analysis, and plotting.
https://simfit.uk
https://simfit.org.uk
https://simfit.silverfrost.com

8.1 Introduction

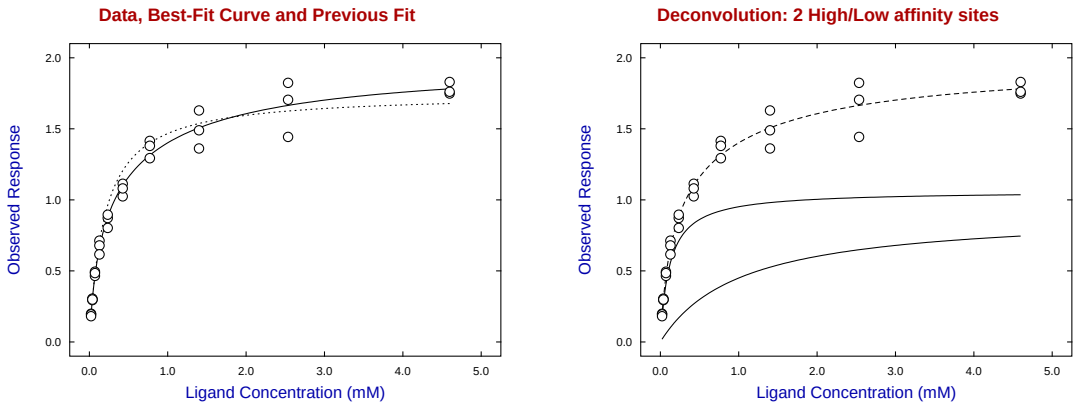
Curve and surface fitting aims to fit mathematical models described by equations or systems of equations to observations in order to estimate parameters that can be used to interpret the experimental data.

An example to illustrate the advantages of curve fitting

As a typical example consider the results from using SimFIT program **hlfit** to fit the dose response data in test file `hlfit.tf4` to one then two binding sites, where the aim is to decide if the assumption of two sites can be justified on statistical grounds and, if so, to estimate the parameters for the model

$$f(x) = \frac{A_1 x}{1 + Ka_1 x} + \frac{A_2 x}{1 + Ka_2 x} + C$$

but with $C = 0$. **hlfit** displays the following plots to illustrate that fitting two sites gives significant improvement over fitting one site, and **hlfit** also provides convincing evidence of this by showing how the overall fit results as the sum of the distinct contributions from the low and high affinity sites.



Not only does the evidence support the hypothesis that there are two classes of binding sites of differing affinity and activity, but this is substantiated by the following results table for estimated parameters, their standard errors, 95% confidence ranges, and significance levels.

Number	Parameter	Value	Std.error	Lower95%cl	Upper95%cl	p	
1	A_1	0.91175	0.2451	0.4079	1.416	0.0010	
2	A_2	1.0625	0.3055	0.4344	1.691	0.0018	
3	Ka_1	0.97501	0.6857	-0.4345	2.385	0.1669	*
4	Ka_2	8.5829	2.004	4.463	12.70	0.0002	

Apparent V_{max} (i.e. $A_1 + A_2 + \cdots + A_n$) = 1.9742
Apparent K_m (i.e. x_0 where $f(x_0) - C = V_{max}/2$) = 0.31272

Here parameters A_1 and A_2 are proportional to the responses from two populations of binding sites with binding constants Ka_1 and Ka_2, and the apparent overall response and half saturation point are calculated by numerical techniques, which removes the subjective element involved in data interpretation. A brief survey of the nomenclature used and procedures provided by SimFIT follows.

The Data

In the simplest case an experimentalist would have N pairs of observations $x(i), y(i)$, possibly together with $s(i)$, the estimated standard deviations of $y(i)$ to use as weights $w(i) = 1/s(i)^2$, as follows

$$X = x(1), x(2), x(3), \ldots, x(N)$$
$$Y = y(1), y(2), x(3), \ldots, y(N)$$
$$S = s(1), s(2), s(3), \ldots, s(N)$$

and there could be three possibilities.

1. **Case 1**
   Values of $x(i)$ are known with high accuracy, as fixed by experiment, and the error distribution of $y(i)$ is assumed to be one of constant variance. In this case it is usual to set all $s(i) = 1$.

2. **Case 2**
   Values of $x(i)$ are known with high accuracy, as fixed by experiment, and the error distribution of $y(i)$ is assumed to vary as a function of the experimental conditions, so values for $s(i)$ are required.

3. **Case 3**
   Values of $x(i)$ and $y(i)$ would both be measured, i.e. there could be error or variation in X and Y.

Of course there are endless variations on this simple scheme, for instance, X and/or Y could be multidimensional, and the model might have to be defined as an implicit function, $\Phi(x, y) = 0$, or require numerical integration of a system of nonlinear differential equations.

The weighting

It is important to realize that all curve fitting is actually weighted curve fitting. The only issue is whether the weighting is assumed to have a defined form, to be estimated from the sample, or to be estimated independently.

- **Case 1**
  This is the simplest and most used technique because it assumes that X is an independent variable and Y values result from a random error ϵ with constant variance added to an exact function value, i.e.

  $$y(i) = f(x(i)) + \epsilon(i).$$

  With this approach no separate attempt is made to estimate the variance of Y as the sample variance of Y is used. It has the great attraction of simplicity but there are two things to observe:
  a) This assumption is almost never true as, in general, error variance is an increasing function $|Y|$.
  b) It diminishes the importance of low $|Y|$ *values so that the resulting fit is dominated by large* $|Y|$ *values.*

- **Case 2**
  This is more realistic in that it accepts that the variance of Y is not constant and attempts to remedy this by providing or calculating a set of weighting factors $s(i)$. However, if the $s(i)$ are inaccurate, the resulting fit can be even more biased than with Case 1. In particular, using a weighting scheme based simply on the experimental Y values or the estimated function values can lead to a situation the reverse of Case 1, where the fit can be dominated by small values of the observations.

- **Case 3**
  This requires that the variance-covariance matrix $CV(X, Y)$ be estimated, and sample estimates for variance-covariance matrices are notoriously unreliable. For this reason this approach is often reserved for the analysis of very large samples with very simple model equations, together with prior knowledge of the covariance structure.

The model

The model to be fitted will involve parameters that have to be estimated, e.g. a parameter vector Θ as in

$$\Theta = \theta_1, \theta_2, \theta_3, \ldots, \theta_m$$

and models are usually described as linear or nonlinear.

Linear models

Linear models are of the form $f(\Theta, x) = \theta_1 f_1(x) + \theta_2 f_2(x) + \theta_3 f_3(x) + \cdots + \theta_m f_m(x)$ and examples could be

$$\begin{aligned} \text{A simple straight line:} \quad & f(\theta, x) = \theta_1 + \theta_2 x \\ \text{A multilinear model:} \quad & f(\Theta, x) = \theta_1 x_1 + \theta_2 x_2 + \theta_3 x \cdots + \theta_m x_m \\ \text{A polynomial:} \quad & f(\Theta, x) = \theta_1 + \theta_2 x + \theta_3 x^2 + \cdots + \theta_m x^{m-1}. \end{aligned}$$

These all have partial derivatives of $f(\Theta, x)$ with respect to θ_i that are independent of Θ.

The advantage of linear models is that they can be fitted very easily and usually lead to unique solutions. Another advantage is that the assumption of normally distributed errors zero means and constant variance allows the application of convenient statistical tests for goodness of fit and parameter significance based on the χ^2, t, and F distributions. The disadvantage is that the real world is nonlinear and linear models are not based on scientific laws but are used for convenience when a meaningful mathematical model is not available.

Nonlinear models

Nonlinear models do not have have partial derivatives of $f(\Theta, x)$ with respect to θ_i that are independent of Θ. Examples could be the following.

$$\begin{aligned} \text{Michaelis-Menten functions:} \quad & f(\Theta, x) = \frac{\theta_1 x}{\theta_2 + x} + \frac{\theta_3 x}{\theta_4 + x} + \cdots + \frac{\theta_{m-1} x}{\theta_m + x} \\ \text{Exponential functions:} \quad & f(\Theta, x) = \theta_1 \exp(-\theta_2 x) + \theta_3 \exp(-\theta_4 x) + \cdots + \theta_{m-1} \exp(-\theta_m x) \\ \text{Rational functions:} \quad & f(\Theta, x) = \frac{\theta_1 x + \theta_2 x^2 + \cdots + \theta_{m/2} x^{m/2}}{1 + \theta_{m/2+1} x + \theta_{m/2+2} x^2 + \cdots + \theta_m x^{m/2}}. \end{aligned}$$

The advantage of using nonlinear models is that they may be a good approximation to reality based on scientific laws. The disadvantage is that they have to be fitted by iterative techniques which depend critically on sensible scaling, good starting estimates, and meaningful limits on parameter values. Because of this, local rather than global solutions may be located.

However it is well to remember a distinct limitation of nonlinear regression: all the numerical techniques used to fit the models, and all the statistical methods used to interpret the results, are based on the assumptions that the model can be regarded as approximately linear at a solution point, and that the weighting factors are known exactly.

The objective function

This will usually be $WSSQ$, the weighted sum of squared residuals defined as

$$WSSQ = \sum_{i=1}^{N} w(i)[y(i) - f(\Theta, x(i)]^2,$$

and the hope is that minimizing this expression with respect to parameters Θ will be equivalent to finding the maximum likelihood estimates.

Options

The SIMFIT package provides many options to prepare data, define models, fit data, and test for model discrimination, goodness of fit, and parameter redundancy as briefly summarized below.

- **Programs for linear models**
  Simple linear models can be fitted by program **linfit** which also provides techniques for orthogonal regression, generalized linear models (GLM), and partial least squares (PLS). Polynomials can be fitted by program **polnom**, which also allows inverse prediction, i.e. generating calibration curves. Several varieties of cubic splines can be fitted by program **spline**, and used to compare curves by **compare**, while program **calcurve** is dedicated to using splines to construct calibration curves followed by inverse prediction of x given y.

- **Programs for simple nonlinear regression**
  The following programs attempt to guess starting estimates then fit models and output tables of statistical results and graphs.

  - **mmfit** fits Michaelis-Menten models.
  - **hlfit** fits high and low affinity binding site models.
  - **sffit** fits cooperative ligand binding isotherms.
  - **rffit** fits positive rational functions.
  - **exfit** fits several types of exponentials functions.
  - **gcfit** fits classical nonlinear growth models.

- **Advanced nonlinear regression**
  Program **qnfit** provides the following facilities but, as it is extremely comprehensive, it requires considerable expertise and should only be used by experienced analysis.

  - Models can be functions of one or several independent variables.
  - Multiple linked or independent models can be fitted simultaneously.
  - Parameters can be constrained interactively within user-defined limits.
  - Three dimensional plots and contours of the objective function can be plotted at solution points.
  - The best fit models can be used for evaluation or inverse prediction.
  - Models can be used from a built-in library or supplied as text files.
  - Single nonlinear differential equations can be fitted.
  - Models defined as convolutions of two defined functions can be fitted.

- **Differential equations**
  Program **deqsol** allows the simulation and fitting of systems of nonlinear differential equations but, like **qnfit**, it should only be used by experts.

- **Simulation**
  An essential technique required for advanced curve fitting is the ability to simulate exact data using program **makdat** then add random error to simulate reality using program **adderr**. SIMFIT also provides numerous additional facilities to confirm the robustness of results from regression with respect to sensitivity of the results to perturbations of parameter values, change in the range of variables, nature of the error, etc.

How to interpret tables of parameter estimates

The meaning of the results generated by program **exfit** after fitting two exponentials to `exfit.tf4` will now be explained, as a similar type of analysis is generated by all the user-friendly curve fitting programs. Consider, first of all the next table listing parameter estimates which result from fitting the two exponential function

$$f(t) = A_1 \exp(-k_1 t) + A_2 \exp(-k_2 t).$$

Parameter	Value	Std.error	Lower95%cl	Upper95%cl	p
A_1	0.8526	0.0677	0.713	0.992	0.0000
A_2	1.1764	0.0747	1.023	1.330	0.0000
k_1	6.7933	0.8541	5.038	8.549	0.0000
k_2	1.1121	0.0511	1.007	1.217	0.0000
AUC	1.1834	0.0147	1.153	1.214	0.0000

AUC is the area under the curve from $t = 0$ to $t = \infty$
Initial time point $(A) = 0.03598$
Final time point $(B) = 1.611$
Area from $t = A$ to $t = B = 0.9383$
Average over range $(A, B) = 0.5958$

The first column gives the estimated values for the parameters, i.e., the amplitudes A_i and decay constants k_i, although it must be appreciated that the pairwise order of these is arbitrary. Actually program **exfit** will always try to rearrange the output so that the amplitudes are in increasing order, and a similar rearrangement will also occur with programs **mmfit** and **hlfit**. For situations where $A_i > 0$ and $k_i > 0$, the area from zero to infinity, i.e. the AUC, can be estimated, as can the area under the data range and the average function value calculated from it. The parameter AUC is not estimated directly from the data, but is a secondary parameter estimated algebraically from the primary parameters. The standard errors of the primary parameters are obtained from the inverse of the estimated Hessian matrix at the solution point, but the standard error of the AUC is estimated from the partial derivatives of AUC with respect to the primary parameters, along with the estimated variance-covariance matrix. The 95% confidence limits are calculated from the parameter estimates and the t distribution, while the p values are the two-tail probabilities for the estimates, i.e., the probabilities that parameters as extreme or more extreme than the estimated ones could have resulted if the true parameter values were zero. The windows defined by the confidence limits are useful for a quick rule of thumb comparison with windows from fitting the same model to another data set; if the windows are disjoint then the corresponding parameters differ significantly, although there are more meaningful tests. Clearly, parameters with $p < 0.05$ are well defined, while parameters with $p > 0.05$ must be regarded as ill-determined.

Expert users may sometimes need the estimated correlation matrix

$$C_{ij} = \frac{CV_{i,j}}{\sqrt{CV_{ii} CV_{jj}}},$$

where $-1 \le C_{ij} \le 1$, $C_{ii} = 1$, which is shown in the next table, and where the index i refers to the natural order of parameters, that is $i = 1, 2, 3, 4$ corresponds to A_1, k_1, A_2, k_2.

Parameter correlation matrix

1			
-0.8756	1		
-0.5961	0.8995	1	
-0.8478	0.9485	0.8199	1

How to interpret tables for goodness of fit

The next table, displaying the results from analyzing the residuals after fitting two exponentials to `exfit.tf4`, is typical of many SimFIT programs. Residuals tables should always be consulted when assessing goodness of fit.

Analysis of residuals: $WSSQ$	24.397
$P(\chi^2 \geq WSSQ)$	0.5533
R^2, cc(theory,data)^2	0.9934
Largest absolute relative residual	11.99%
Smallest absolute relative residual	0.52%
Average absolute relative residual	3.87%
Absolute relative residuals in range 0.1–0.2	3.33%
Absolute relative residuals in range 0.2–0.4	0.00%
Absolute relative residuals in range 0.4–0.8	0.00%
Absolute relative residuals > 0.8	0.00%
Number of negative residuals (n_1)	15
Number of positive residuals (n_2)	15
Number of runs observed (r)	16
P(runs $\leq r$: given n_1 and n_2)	0.5759
5% lower tail point	11
1% lower tail point	9
P(runs $\leq r$: given n_1 plus n_2)	0.6445
P(signs $\leq$ least number observed)	1.000
Durbin-Watson test statistic	1.8061
Shapiro-Wilks W statistic	0.9387
Significance level of W	0.0841
Akaike AIC (Schwarz SC) statistics	1.7979 (7.4027)

Verdict on goodness of fit: incredible

Several points should be remembered when assessing such residuals tables, where there are N observations $y(i)$, with weighting factors $s(i)$, theoretical values $f(x(i))$, residuals $r(i) = y(i) - f(x(i))$, weighted residuals $r(i)/s(i)$, and where m parameters have been estimated. Theoretical details for the statistical tests will be found in the SimFIT reference manual `w_manual.pdf`, or the appropriate tutorial documents.

- $WSSQ$
  The χ^2 test on $N - m$ degrees of freedom using $WSSQ$, the objective function at the solution point where
  $$WSSQ = \sum_{i=1}^{N} \left(\frac{y(i) - f(x_i)}{s(i)} \right)^2,$$
  is only meaningful if the weights defined by the $s(i)$ supplied for fitting are good estimates of the standard deviations of the observations at that level of the independent variable; say means of at least five replicates. Inappropriate weighting factors will result in a biased chi-square test. Also, if all the $s(i)$ are set equal to 1, unweighted regression will be performed and an alternative analysis test based on the coefficient of variation will be performed.

- R^2
  The R^2 value is the square of the correlation coefficient between data and best fit points. It only represents a meaningful estimate of that proportion of the fit explained by the regression for simple unweighted linear models, and should be interpreted with restraint when nonlinear models have been fitted.

- **Absolute relative residuals**

The results based on the absolute relative residuals $a(i)$ defined using machine precision ϵ as

$$a_i = \frac{2|r(i)|}{\max(\epsilon, |y(i)| + |f(x(i))|)}$$

do not have statistical relevance, but they do have obvious empirical justification, and they must be interpreted with commonsense, especially where the data and/or theoretical values are very small.

- **Run and sign tests**
  The probability of the number of runs observed given n_1 negative and n_2 positive residuals is a very useful test for randomly distributed runs, but the probability of runs given $N = n_1 + n_2$, and also the overall sign test are weak, except for very large data sets.

- **Durbin-Watson test**
  The Durbin-Watson test statistic

$$DW = \frac{\sum_{i=1}^{N-1} (r(i+1) - r(i)^2}{\sum_{i=1}^{N} r(i)^2}$$

  is useful for detecting serially correlated residuals, which could indicate correlated data or an inappropriate model. The expected value is 2.0, and values less than 1.5 suggest positive correlation, while values greater than 2.5 suggest negative serial correlation.

- **Shapiro-Wilks test**
  Where N, the number of data points, significantly exceeds m, the number of parameters estimated, the weighted residuals are approximately normally distributed, and so the Shapiro-Wilks test should be taken seriously.

- **Akaike and Schwarz criteria**
  The Akaike AIC statistic

$$AIC = N \log(WSSQ/N) + 2m$$

  and Schwarz Bayesian criterion SC

$$SC = N \log(WSSQ/N) + m \log N$$

  are only really meaningful if minimizing $WSSQ$ is equivalent to Maximum Likelihood Estimation. Note that only differences between AIC with the same data, i.e. fixed N, are relevant, as in the evidence ratio ER, defined as $ER = \exp[(AIC(1) - AIC(2))/2]$.

- **The qualitative conclusion**
  The final verdict is calculated from an empirical look-up table, where the position in the table is a weighted mean of scores allocated for each of the tests listed above. It is qualitative and rather conservative, and has no precise statistical relevance, but a good result will usually indicate a well-fitting model.

- **Residuals plots**
  As an additional measure, plots of residuals against theory, and half-normal residuals plots can be displayed after such residuals analysis, and they should always be inspected before concluding that any model fits satisfactorily.

- **Leverages**
  With linear models, SimFIT also calculates studentized residuals and leverages, while with generalized linear models, deviance residuals can be tabulated.

How to interpret tables for model discrimination results

After a sequence of models have been fitted, tables like the next one are generated.

$WSSQ$-previous	224.9
$WSSQ$-current	24.4
Number of parameters-previous	2
Number of parameters-current	4
Number of x-values	30
Akaike AIC-previous	64.44
Akaike AIC-current	1.798, $ER = 3.998E + 13$
Schwarz SC-previous	67.24
Schwarz SC-current	7.403
Mallows C_p	213.7, $C_p/m_1 = 106.9$
Numerator degrees of freedom	2
Denominator degrees of freedom	26
F test statistic (FS)	106.9
$P(F \geq FS)$	0.0000
$P(F \leq FS)$	1.0000
5% upper tail point	3.369
1% upper tail point	5.526

Conclusion based on the F test
Reject the previous model at 1% significance level
There is strong support for the extra parameters
Tentatively accept the current best fit model

First of all, note that the above model discrimination analysis is only strictly applicable for nested linear models with known error structure, and should be interpreted with restraint otherwise. Now, if $WSSQ_1$ with m_1 parameters is the previous (possibly deficient) model, while $WSSQ_2$ with m_2 parameters is the current (possibly superior) model, so that $WSSQ_1 > WSSQ_2$, and $m_1 < m_2$, then

$$F = \frac{(WSSQ_1 - WSSQ_2)/(m_2 - m_1)}{WSSQ_2/(N - m_2)}$$

should be F distributed with $m_2 - m_1$ and $N - m_2$ degrees of freedom, and the F test for excess variance can be used. Alternatively, if $WSSQ2/(N - m_2)$ is equivalent to the true variance, i.e., model 2 is equivalent to the true model, the Mallows C_p statistic

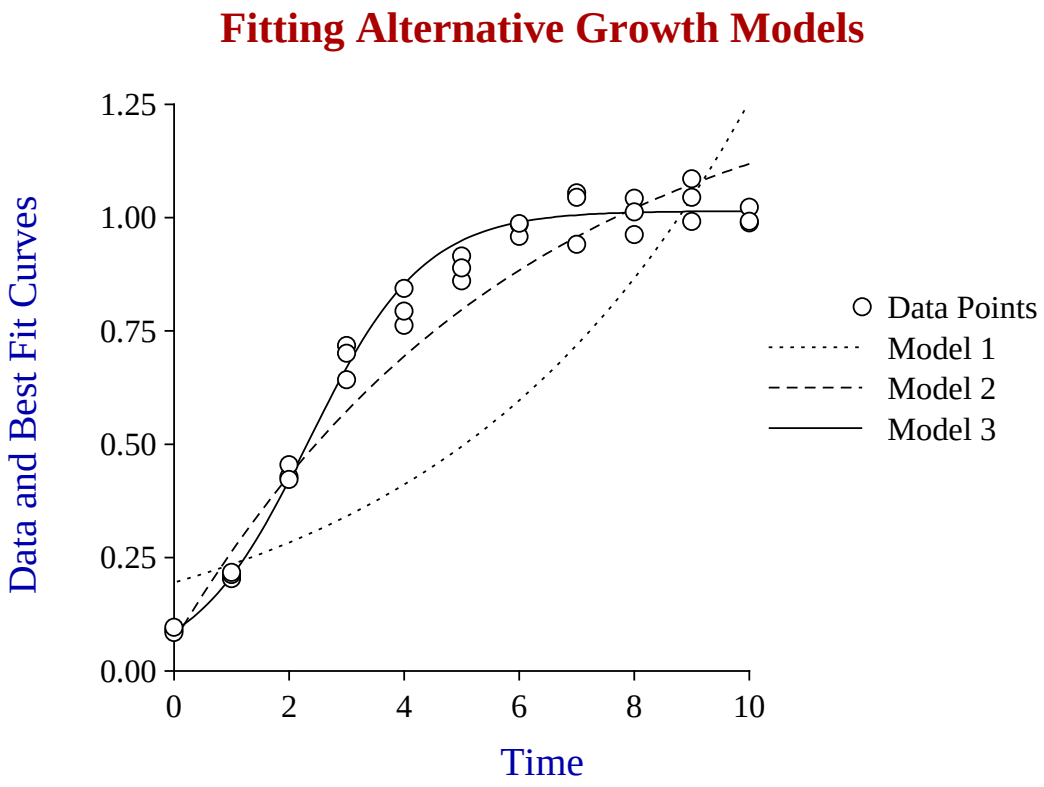
$$C_p = \frac{WSSQ_1}{WSSQ_2/(N - m_2)} - (N - 2m_1)$$

can be considered. This has expectation m_1 if the previous model is sufficient, so values greater than m_1, that is $C_p/m_1 > 1$, indicate that the current model should be preferred over the previous one. However, graphical deconvolution should always be done wherever possible, as with sums of exponentials, Michaelis-Mentens, High-Low affinity sites, sums of Gaussians or trigonometric functions, etc., before concluding that a higher order model is justified on statistical grounds.

8.2 Goodness of fit

Goodness of fit analysis is always required before a model can be considered to be satisfactory, otherwise using parameter estimates to characterize experimental observations can lead to false interpretation.

As an example consider the use of program **gcfit** to fit nonlinear growth models to data in the test file `gcfit.tf2` as shown in the next figure. A typical situation would be when an experimentalist would want to fit growth curves to data with the main aim being to estimate parameters like the maximum growth rate, the time at which this was achieved, and the final size attained in order to characterize a group under observation, say bacterial colonies of several species incubated with alternative antibiotics.



The models fitted were Model 1 (exponential), Model 2 (monomolecular) and Model 3 (logistic) as follows.

$$\text{Model 1: } f_1(t) = A_1 \exp(k_1 t)$$
$$\text{Model 2: } f_2(t) = A_2(1 - \exp(-k_2 t))$$
$$\text{Model 3: } f_3(t) = \frac{A_3}{1 + B\exp(-k_3 t)}$$

It is perfectly clear in this case that Model 1 is completely unsatisfactory, Model 2 would give a rough estimate for the final asymptotic size, while Model 3 would accurately fit all features of the data set. The aim of this document is to show how SimFIT could be used to make a decision in situations where the outcome is not so clear cut.

SimFIT program **gcfit** also displays the following summary.

Model	$WSSQ/NDOF$	$P(C \geq W)$	$P(\text{Runs} \leq r)$	$N > 10\%$	$N > 40\%$	$Av.r\%$	Verdict
1	1.52E+02	0.000	0.000	29	17	40.03	Very bad
2	1.81E+01	0.000	0.075	20	0	12.05	Very poor
3	1.32E+00	0.113	0.500	0	0	3.83	Incredible

The way to interpret this goodness of fit summary table will now explained, but it must always be remembered that the only situations where statistical tests are justified, and when minimizing $WSSQ$ is equivalent to maxmimum likelihood, are when the following four conditions are met.

1. The values $x(i)$ must be known exactly, and not subject to errors of estimation or natural variation. In other words, X can be regarded as an independent variable and not a covariate.
2. The error of measurement $\epsilon(i)$ must be normally distributed with mean zero.
3. The variance of $\epsilon(i)$ has one of two forms.
   (a) The homoscedastic case where weighting factors $s(i)$ are all equal to one and $WSSQ/NDOF$ estimates the constant variance.
   (b) The heteroscedastic case where the variance of $\epsilon(i)$ is a function of X and/or Y and exact values for the standard deviation of the $\epsilon(i)$ are supplied as $s(i)$. In other words, values of $s(i)$ are supplied to reduce this case to the homoscedastic case with error variance = 1.
4. The model is correct and linear.

As experimental errors are more like a Cauchy distribution than a normal distribution, variance of the experimental error is usually an increasing function of the absolute value of the observations, values of $s(i)$ supplied are at best only determined with limited precision from independent studies or at worst are determined from replicates, and the model is nonlinear and often only an approximation anyway, such results tables must be interpreted with restraint.

- $WSSQ/NDOF$
  This is the objective function estimated by SimFIT and $WSSQ/NDOF$ should be approximately one at the solution point, as the expectation of a chi-square variable is the number of degrees of freedom.
- $P(C \geq W)$
  This is the very approximate result of a performing a χ^2 test on the weighted sum of squared residuals. An alternative test is usually done by SimFIT based on the estimated coefficient of variation here and for the previous result when all $s(i) = 1$.
- $P(Runs \leq r)$
  This is the probability of runs less than the number obtained, given the number of negative and positive residuals.
- $N > 10\%$
  This is the number of data points where the ratio of absolute residual to absolute value of observation exceeds 0.1.
- $N > 40\%$
  This is the number of data points where the ratio of absolute residual to absolute value of observation exceeds 0.4.
- $Av.r\%$
  This is the average of absolute residual divided by absolute observation as a percentage.
- Verdict
  This is a somewhat arbitrary decision based on a formula involving all of these, and also some other factors.

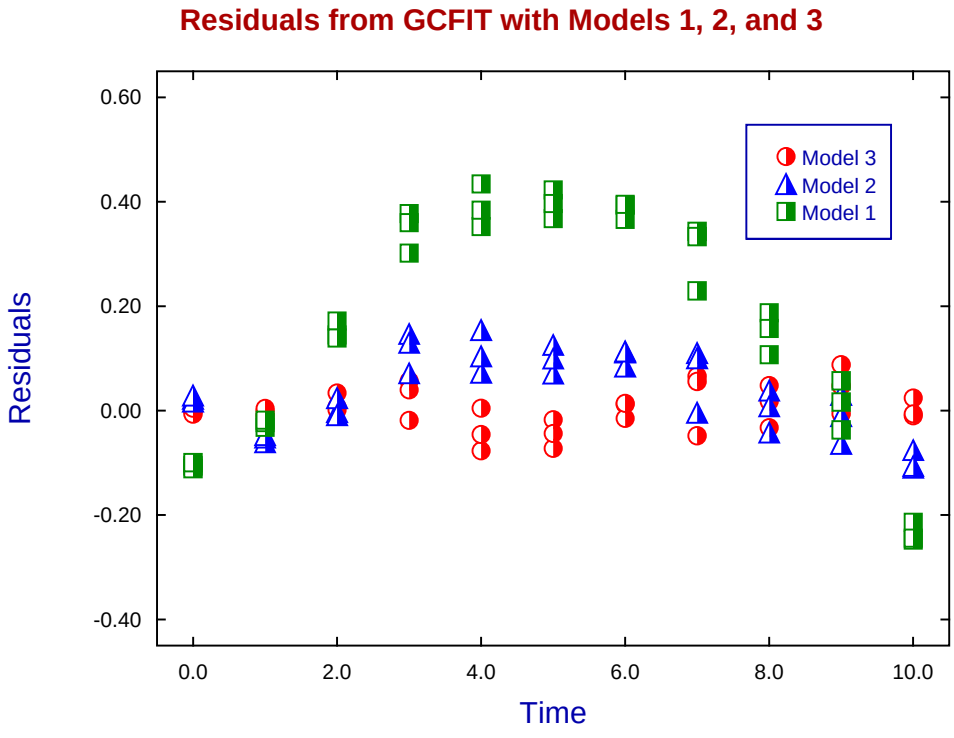
Analysis of residuals

Analysis of residuals and/or weighted residuals is a very important way to judge goodness of fit, especially when there is only one independent variable, and SIMFIT provides numerous ways to do this as follows.

1. Tables of residuals
   These highlight residuals which indicate poor fit by colour changes and stars.
2. Tables summarizing goodness of fit based on residuals
3. Test for runs and serial correlations
   These rely on the residuals being in a systematic order, such as in order of the independent variable.
4. Test for a normal distribution
   Residuals cannot be normally distributed due to correlations induced by parameter estimation, nevertheless the Shapiro-Wilks test is quiet robust when the number of observations greatly exceeds the number of parameters estimated.
5. Methods for plotting residuals
   (a) Residuals plotted against the independent variable
   (b) Residuals plotted against the observations
   (c) Residuals plotted against the best-fit model
   (d) Normal and half-normal plots

   Probably option (a) is the easiest to interpret and residuals have to deviate wildly from normality before option (d) picks this up. Unfortunately this is the only option available when there are multiple independent variables.

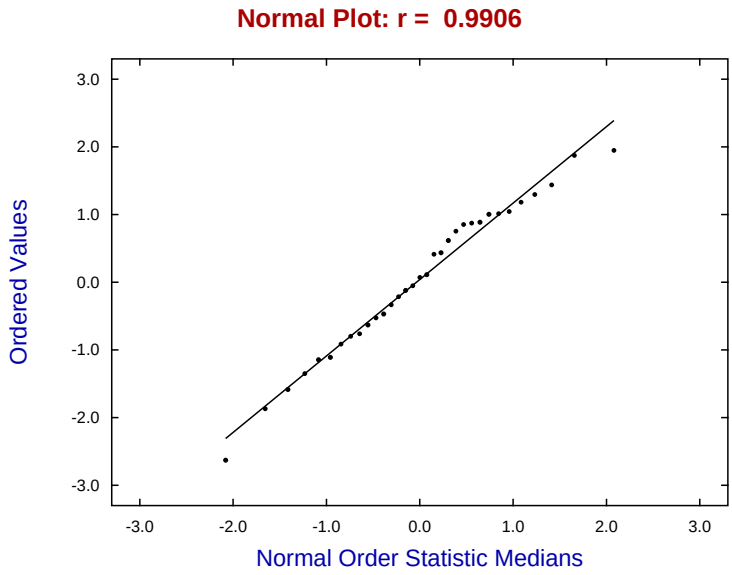
Residuals and/or weighted residuals should be scattered randomly about zero, and the next plot shows very clearly that with Model 1 there is a systematic nonlinear drift which is much less with model 2, while model 3 shows a much more acceptable pattern

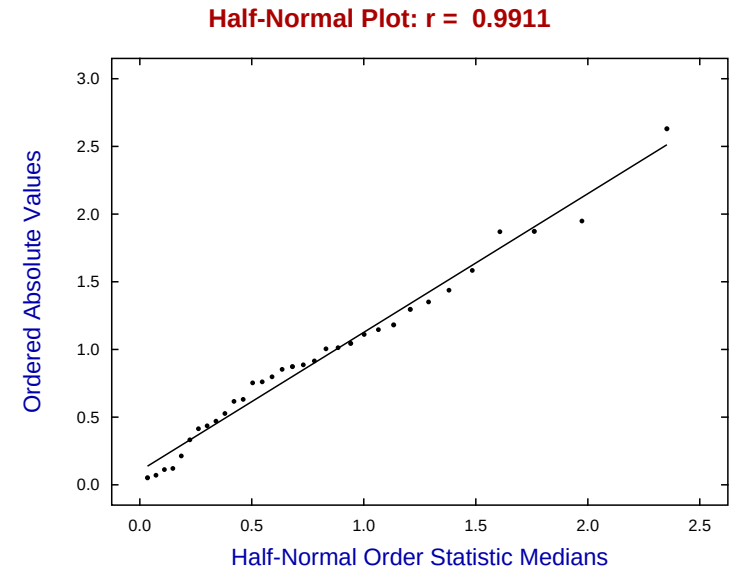


A plot of the n ordered residuals or weighted residuals as Y against normal order statistic medians as X should be very close to linearity, since residuals should be approximately normally distributed when the number of points is much greater than the number of parameters estimated. The medians are approximated using

$$x_i = \Phi^{-1}(z_i) \text{ where } z_i = (i - 3/8)/(n + 1/4) \text{ for } i = 1, 2, \cdots, n.$$

Here Φ^{-1} is the inverse standard normal distribution function. It is also possible to create a half-normal plot where Y are the ordered absolute residuals, and X values are calculated by a similar approximation but using $z_i = (n + 0.5 + i)/(2n + 9/8)$ to allow for the wrapping round of the negative residuals. If correctly weighted residuals are plotted, as in the next figure for the fitted logistic model, the Y values should be in the range -3 to 3 for the normal plot, but 0 to 3 for the half normal plot.





Best-fit lines for the regression of Y on X are also plotted on these graphs along with the Pearson product-moment correlation coefficient r. The significance level p for the r will also be displayed when $p < 0.05$, but this only happens when the residuals show clear departure from linearity in these plots.

Testing for differences between two parameter estimates

This can sometimes be a useful simple procedure when you wish to compare two parameters resulting from a regression, e.g., the final size from fitting a growth curve model, or perhaps two parameters that have been derived from regression parameters e.g., AUC from fitting an exponential model, or LD50 from bioassay.

You input the two parameter estimates θ and ϕ, the standard error estimates s_θ and s_ϕ, the number of experimental observations N_θ and N_ϕ, and the number of parameters estimated from the regression M_θ and M_ϕ. A t test for equality is then performed with the correction for unequal variances by the the Satterthwaite procedure, using a t_c statistic with ν degrees of freedom calculated with the Welch correction for unequal variances given by

$$t_c = \frac{\theta - \phi}{\sqrt{s_\theta^2 + s_\phi^2}}$$

$$\nu = \frac{(s_\theta^2 + s_\phi^2)^2}{s_\theta^4/(N_\theta - M_\theta) + s_\phi^4/(N_\phi - M_\phi)}.$$

Here θ and ϕ refer to the same parameter using the same mathematical model but estimated from two distinct data sets of sizes N_θ and N_ϕ.

Such t tests depend on the asymptotic normality of maximum likelihood parameters, and will only be meaningful if the data set is fairly large and the best fit model adequately represents the data.

Note that t tests on parameter estimates can be especially unreliable because they ignore non-zero covariances in the estimated parameter variance-covariance matrix.

Testing for differences between several parameter estimates

To take some account of the effect of significant off-diagonal terms in the estimated parameter variance-covariance matrix you will need to calculate a Mahalanobis distance between parameter estimates e.g., to test if two or more curve fits using the same model but with different data sets support the presence of significant treatment effects. For instance, after fitting the logistic equation to growth data by nonlinear regression, you may wish to see if the growth rates, final asymptotic size, half-time, etc. have been affected by the treatment.

Note that, after every curve fit, you can select an option to add the current parameters and covariance matrix to your parameter covariance matrix project archive, and also you have the opportunity to select previous fits to compare with the current fit. For instance, you may wish to compare two fits with m parameters, A in the first set with estimated covariance matrix C_A and B in the second set with estimated covariance matrix C_B. The parameter comparison procedure will then perform a t test for each pair of parameters, and also calculate the quadratic form

$$Q = (A - B)^T (C_A + C_B)^{-1} (A - B)$$

which has an approximate chi-square distribution with m degrees of freedom. You should realize that the rule of thumb test using non-overlapping confidence regions is more conservative than the above t test: parameters can still be significantly different despite a small overlap of confidence windows.

This technique must be used with care when the models fitted are themselves sums of k identical sub-functions such as

$$f(\Theta x) = f_1(\Theta, x) + f_2(\Theta, x) + \cdots + f_k(\Theta, x).$$

Examples of where this can occur could be sums of exponentials, Michaelis-Menten terms, High-Low affinity site binding isotherms, Gaussians, trigonometric terms, and so on. This is because the parameters are only unique up to a permutation.

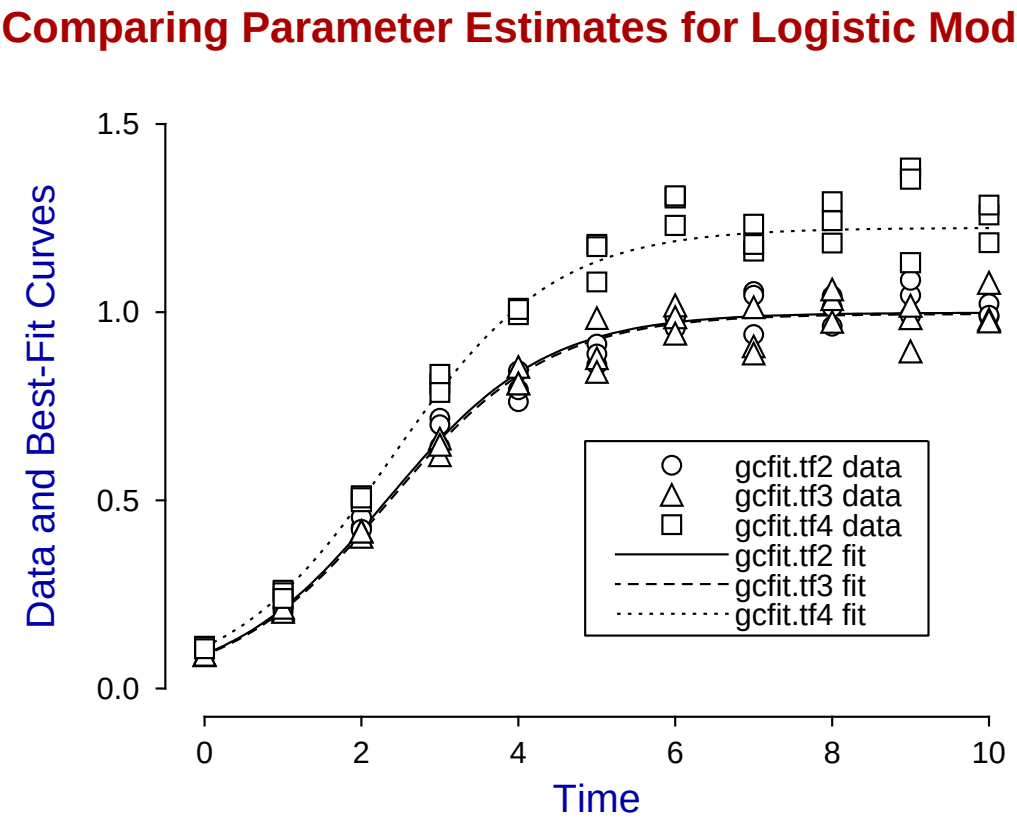
For instance, the terms A_i and k_i are linked in the exponential function

$$f(t) = \sum_{i=1}^{m} A_i \exp(-k_i t)$$

but the order implied by the index i is arbitrary. So, when testing if A_1 from fitting a data set is the same as A_1 from fitting another data set it is imperative to compare the same terms.

The user friendly programs **exfit**, **mmfit**, and **hlfit** attempt to assist this testing procedure by rearranging the results into increasing order of amplitudes A_i but, to be sure, it is best to use **qnfit**, where starting estimates and parameter constraints can be used from a parameter limits file. That way there is a better chance that parameters and covariance matrices saved to project archives for retrospective testing for equality of parameters will be consistent, i.e. the parameters will be compared in the correct order.

The next figure illustrates a common problem, where the same model has been fitted to alternative data sets and it is wished to decide if one or more parameters differ significantly.



In this case, the logistic model defined as

$$f(t) = \frac{\theta_1}{1 + \theta_2 \exp(-\theta_3 t)}$$

was simulated using **makdat** and **adderr** then fitted by **gcfit**, and the main interest is to decide if the estimated final asymptote i.e. $\hat{\theta}_1$ differs significantly for the test files `gcfit.tf2` and `gcfit.tf3` which actually have identical parameters $\theta_1 = 1$, while `gcfit.tf4` has a slightly larger asymptotic value $\theta_1 = 1.25$, the other parameters being identical $\theta_2 = 10$ and $\theta_3 = 1$.

The next table illustrates how this technique works.

Table of Mahalanobis χ^2, and corrected pairwise t tests for differences between parameters(A, B) and covariances(Ca, Cb).

Comparison 1: Parameters from `gcfit.tf3` (A) and `gcfit.tf2` (B)
$Q = (A-B)^T (Ca+Cb)^{-1}(A-B) = 2.193E+00, NDOF = 3$
$P(\chi^2 \geq Q) = 0.5333$

Index	A	B	A - B	t	DOF	p	
1	0.996	0.999	-0.0033	-2.567E-01	53	0.7984	
2	10.15	9.890	0.2600	7.224E-01	40	0.4743	
3	0.985	0.988	-0.0033	-1.164E-02	37	0.9908	

Comparison 2: Parameters from `gcfit.tf4` (A) and `gcffit.tf2` (B)
$Q = (A-B)^T (Ca+Cb)^{-1}(A-B) = 7.492E+02, NDOF = 3$
$P(\chi_2 \geq Q) = 0.0000$: *Reject H_0 at 1% significance level*

Index	A	B	A - B	t	DOF	p	
1	1.224	0.999	0.2251	19.17	57	0.0000	*****
2	10.04	9.890	0.1500	0.382	50	0.7038	
3	0.969	0.988	-0.0191	-0.063	46	0.9501	

Comparison 3: Parameters from `gcfit.tf4` (A) and `gcfit.tf3` (B)
$Q = (A-B)^T (Ca+Cb)^{-1}(A-B) = 1.064E+03, NDOF = 3$
$P(\chi^2 \geq Q) = 0.0000$: *Reject H_0 at 1% significance level*

Index	A	B	A - B	t	DOF	p	
1	1.224	0.996	0.2284	16.21	59	0.0000	*****
2	10.04	10.15	-0.1100	-0.443	52	0.6596	
3	0.969	0.985	-0.0158	-0.093	52	0.9265	

The data were fitted using **gcfit** using the option to store parameter estimates and covariance matrices. Then the global tests for different parameter sets, and t tests for individual parameter differences were performed, leading to the results indicated.

Clearly the parameter estimates for test files `gcfit.tf2` and `gcfit.tf3` indicate no significant differences, while `gcfit.tf4` differed significantly from both of these, due to a larger value for the asymptote θ_1 for `gcfit.tf4`.

Graphical deconvolution

There are occasions when a model to be fitted consists of a sum of sub-functions and it is wished to estimate the contribution of the sub-functions to the overall regression. In some instances it may be possible to plot the overall function fitted to the data along with plots for the sub-functions.

This is particularly valuable with models such as the sum of Gaussians, which for three components is

$$f(x) = \frac{A_1}{\sqrt{2}\sigma_1}\exp -\frac{1}{2}\left(\frac{x-\mu_1}{\sigma_1}\right)^2 + \frac{A_2}{\sqrt{2}\sigma_2}\exp -\frac{1}{2}\left(\frac{x-\mu_2}{\sigma_2}\right)^2 + \frac{A_3}{\sqrt{2}\sigma_3}\exp -\frac{1}{2}\left(\frac{x-\mu_3}{\sigma_3}\right)^2.$$

This model is notoriously difficult to fit unless the amplitudes A_i and variances σ_i^2 are of similar size, but the means μ_i are distinct. However it is one of several models where the ability to do such plotting, which is loosely referred to as graphical deconvolution in SimFIT, is provided.

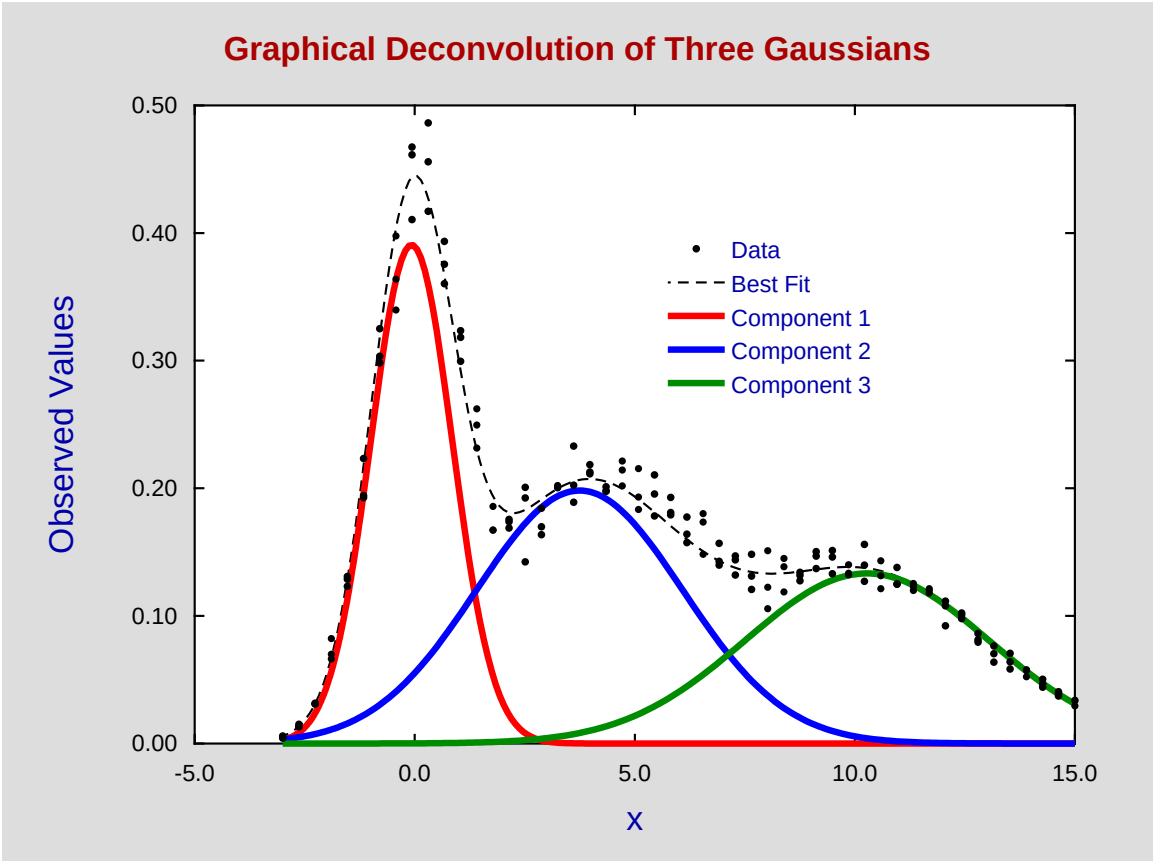
Using SIMFIT program **qnfit** to analyze the data in test file `gauss3.tf1` leads to the following table of parameter estimates for parameters defined in terms of Θ as

$$f(x) = \frac{\theta_1}{\sqrt{2}\theta_7}\exp -\frac{1}{2}\left(\frac{x-\theta_4}{\theta_7}\right)^2 + \frac{\theta_2}{\sqrt{2}\theta_8}\exp -\frac{1}{2}\left(\frac{x-\theta_5}{\theta_8}\right)^2 + \frac{\theta_3}{\sqrt{2}\theta_9}\exp -\frac{1}{2}\left(\frac{x-\theta_6}{\theta_9}\right)^2.$$

The columns indicate: the parameter number, the lowest value allowed for the parameter, the highest value allowed for the parameter, the value of the parameter estimate, the standard error of the parameter estimate, the lower 95% confidence limit for the estimate, the upper 95% confidence limit for the estimate, and the significance level for the estimate. The small p values and absence of stars after the last column in the next table of results indicates that all 9 parameters were well determined.

Number	Low-Limit	High-Limit	Value	Std. Error	Lower95%cl	Upper95%cl	p
1	0.000	2.000	0.90754	0.021624	0.8648	0.9503	0.0000
2	0.000	2.000	1.16433	0.042173	1.0810	1.2477	0.0000
3	0.000	2.000	0.92519	0.030130	0.8656	0.9848	0.0000
4	-2.000	2.000	-0.07298	0.015572	-0.1038	-0.0422	0.0000
5	2.000	6.000	3.74510	0.050816	3.6446	3.8456	0.0000
6	8.000	12.00	10.2774	0.096413	10.087	10.468	0.0000
7	0.100	2.000	0.92641	0.014331	0.8981	0.9547	0.0000
8	1.000	3.000	2.34330	0.070567	2.2038	2.4828	0.0000
9	2.000	4.000	2.76906	0.062637	2.6452	2.8929	0.0000

This conclusion is reinforced by the next graphical deconvolution plot showing the data as dots, the best-fit curve as a dotted line, and the components contributing to the best-fit curve as red, green, and blue curves.



8.3 Linear regression

Tutorials and worked examples for simulation, curve fitting, statistical analysis, and plotting.
https://simfit.uk
https://simfit.org.uk
https://simfit.silverfrost.com

8.3.1 Fitting a straight line: simple

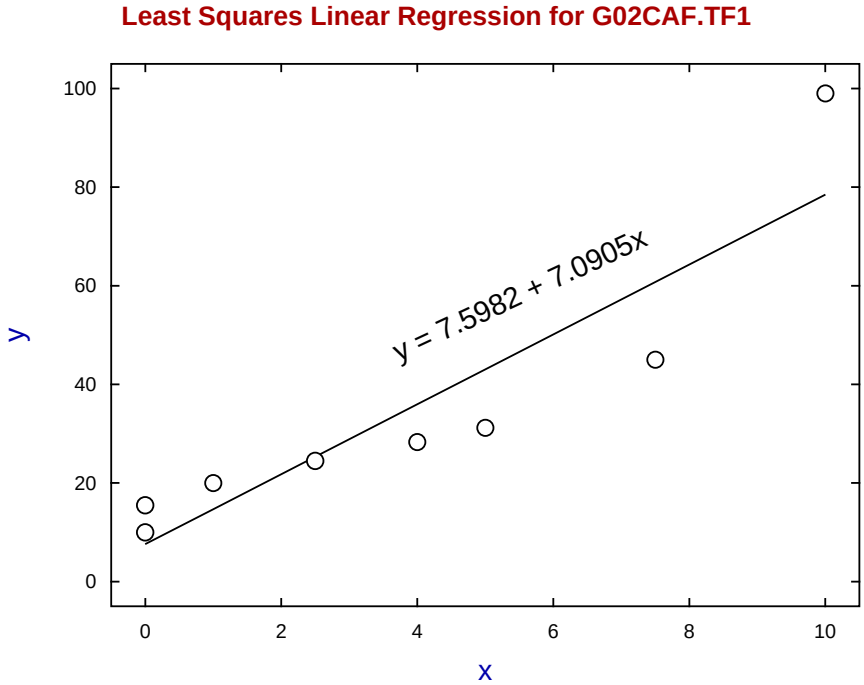
Simple least squares linear regression is used when there are two variables, X which is known accurately and can be regarded as an independent variable, and Y which is a linear function of X except that there is measurement error or random variation which is normally distributed with zero mean and constant variance. From the SIMFIT main menu choose [A/Z], open program **linfit**, choose simple linear regression and inspect the default test file `g02caf.tf1` which has the following data.

x	y
0.0	10.0
0.0	15.5
1.0	20.0
2.5	24.5
4.0	28.3
5.0	31.2
7.5	45.0
10.0	99.0

Analysis yields the following results table and plot for the least squares best-fit straight line.

Parameter	Value	Std. Error	Lower95%cl	Upper95%cl	p	
constant (c)	7.5982	6.6858	-8.7613	23.958	0.2991	**
slope (m)	7.0905	1.3224	3.8548	10.326	0.0017	

($r^2 = 0.8273, r = 0.9096, p = 0.0017$)



The way to interpret this table is as follows.

Column 1 This indicates that the equation fitted is $y = mx + c$.

Column 2 Values for the estimated parameters ($\hat{m}$ and $\hat{c}$).

Column 3 The standard errors for the parameter estimates ($\hat{se}_m$ and $\hat{se}_c$).

Column 4 The lower 95% confidence limit for the true parameters.

Column 5 The upper 95% confidence limit for the true parameters.

Column 6 The significance level for the t variables $t_m = \hat{m}/\hat{se}_m$ and $t_c = \hat{c}/\hat{se}_c$.

Column 7 The stars indicate that the constant is not significantly different from zero.

Last line This records the Pearson product-moment correlation coefficient r, and the significance level p, indicating that the probability of these data resulting from a bivariate distribution with zero correlation parameter ρ is less than 1%.

Theory

The assumed model is that $y_i = mx_i + c + \epsilon_i$ for $n > 2$ observations, where ϵ_i is normally distributed with zero mean and variance σ^2, and the best fit parameters are those at the minimum value of SSQ defined as the sum of squared residuals, that is

$$\begin{aligned} SSQ &= \sum_{i=1}^{n} \epsilon_i^2 \\ &= \sum_{i=1}^{n} (y_i - \hat{m}x_i - \hat{c})^2. \end{aligned}$$

The sample means $\bar{x}, \bar{y}$, standard deviations s_x, s_y, Pearson product-moment correlation coefficient r, and estimates $\hat{m}, \hat{c}$ are as follows.

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

$$\bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i$$

$$s_x = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2}$$

$$s_y = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (y_i - \bar{y})^2}$$

$$r = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n} (x_i - \bar{x})^2 \sum_{i=1}^{n} (y_i - \bar{y})^2}}$$

$$\hat{m} = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2}$$

$$\hat{c} = \bar{y} - \hat{m}\bar{x}$$

In order to perform an analysis of variance and estimate parameter standard errors further quantities are required. The total sum of squares SST with degrees of freedom $n-1$, the sum of squares of deviations about the regression SSD with degrees of freedom $n-2$, the sum of squares attributable to the regression SSR with degrees of freedom 1, and the mean square of deviations about the regression MSD are defined as follows.

$$\begin{aligned} SST &= \sum_{i=1}^{n}(y_i - \bar{y})^2 \\ SSD &= SSQ \\ SSR &= SST - SSD \\ MSD &= SSQ/(n-2) \end{aligned}$$

MSD is used as an estimate for the constant variance of y_i in order to estimate the standard errors of the slope and constant. Then the standard errors of the slope se_m and constant se_c are

$$\hat{se}_m = \sqrt{\frac{MSD}{\sum_{i=1}^{n}(x_i - \bar{x})^2}}$$

$$\hat{se}_c = \sqrt{\frac{MSD\sum_{i=1}^{n}x_i^2}{n\sum_{i=1}^{n}(x_i - \bar{x})^2}}.$$

Another quantity that is sometimes required is the multiple correlation coefficient

$$R^2 = \frac{\sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

where $\hat{y}_i$ is the best-fit value evaluated at x_i, and R is the correlation coefficient for y_i and $\hat{y}_i$. R^2 is said to measure the proportion of the total variation about $\hat{y}$ explained by the regression.

In the special case of fitting a straight line by least squares then we also have

$$R^2 = \frac{\left(\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})\right)^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2\sum_{i=1}^{n}(y_i - \bar{y})^2},$$

and so the multiple correlation coefficient equals the square of the Pearson product-moment correlation coefficient r between X and Y.

It should be emphasized that the equation

$$R^2 = r^2$$

is only true for the special situation where the best-fit equation is assumed to be the least squares line, that is

$$y(x) = \hat{m}x + \hat{c}.$$

8.3.2 Fitting a straight line: comprehensive

Comprehensive least squares linear regression is used when there are two variables, X which is known accurately and can be regarded as an independent variable, and Y which is a linear function of X, except that there is measurement error or random variation which is normally distributed with zero mean and constant variance. This option provides procedures to check for goodness of fit which are not available with the simple linear regression option.

From the SimFIT main menu choose [A/Z], open program **linfit**, choose advanced linear regression and inspect the default test file `line.tf2` which has the following data.

x	y
28.10	11.88
28.60	11.08
28.90	12.19
29.70	11.13
30.80	12.51
33.40	10.36
35.30	10.98
39.10	9.570
44.60	8.860
46.40	8.240
46.80	10.94
48.50	9.580
57.50	9.140
58.10	8.470
58.80	8.400
59.30	10.09
61.40	9.270
70.00	8.110
70.00	6.830
70.70	7.820
71.30	8.730
72.10	7.680
74.40	6.360
74.50	8.880
76.70	8.500

The two columns of data have the following meanings.

1. Column one is the independent x (with no error), the temperature in degrees Fahrenheit.
2. Column two is the dependent variable y (with error), in pounds of steam per month.

This options then fits a straight line in the form $y = mx + c$ leading to the following results.

Table 1: Parameter estimates

Parameter	Value	Std. Error	Lower95%cl	Upper95%cl	p
constant (c)	13.623	0.58146	12.420	14.826	0.0000
slope (m)	-0.079829	0.010524	-0.1016	-0.058059	0.0000
$r^2 = 0.7144, r = -0.8452, p = 0.0000$					

Table 2: Residuals

x	y	Theory	Residuals	
28.1	1.188	1.138	0.5002	
28.6	1.108	1.134	-0.2599	
28.9	1.219	1.132	0.8741	*
29.7	1.113	1.125	-0.1221	
30.8	1.251	1.116	1.3460	**
33.4	1.036	1.096	-0.5967	*
35.3	1.098	1.081	0.1750	
39.1	9.570	1.050	-0.9317	*
44.6	8.860	1.006	-1.2030	**
46.4	8.240	9.919	-1.6790	**
46.8	1.094	9.887	1.0530	**
48.5	9.580	9.751	-0.1713	
57.5	9.140	9.033	0.1072	*
58.1	8.470	8.985	-0.5149	*
58.8	8.400	8.929	-0.5291	*
59.3	1.009	8.889	1.2010	**
61.4	9.270	8.722	0.5485	*
70.0	8.110	8.035	-0.0750	
70.0	6.830	8.035	-1.2050	**
70.7	7.820	7.979	-0.1591	
71.3	8.730	7.931	0.7988	*
72.1	7.680	7.867	-0.1873	
74.4	6.360	7.684	-1.3240	**
74.5	8.880	7.676	1.2040	**
76.7	8.500	7.500	0.9999	**

Table 3: Analysis of residuals

Sum of squared residuals: SSQ	18.223
Estimated average % coefficient of variation	9.45%
R^2, correlation coefficient(theory,data)^2	0.7144
Largest Absolute relative residual	18.85%
Smallest Absolute relative residual	0.93%
Average Absolute relative residual	7.80%
Percentage of absolute relative residuals in range 0.1–0.2	36.00%
Percentage of absolute relative residuals in range 0.2–0.4	0%
Percentage of absolute relative residuals in range 0.4–0.8	0%
Percentage of absolute relative residuals > 0.8	0%
Number of residuals < 0 (m)	13
Number of residuals > 0 (n)	12
Number of runs observed (r)	17
$P(runs \leq r:$ given m and $n)$	0.9502
5% lower tail point	9
1% lower tail point	7
$P(runs \leq r:$ given m plus $n)$	0.9680
$P(signs \leq$ least number observed)	1.0000
Durbin-Watson test statistic	1.9930
Shapiro-Wilks W statistic	0.9596
Significance level of W	0.4064
Akaike AIC (Schwarz SC) statistics	-3.904 (-1.467)
Verdict on goodness of fit: *fantastic*	

Table 1
This illustrates that there was a strong linear correlation between x and y with well determined parameters, as all p values were less than 0.01.

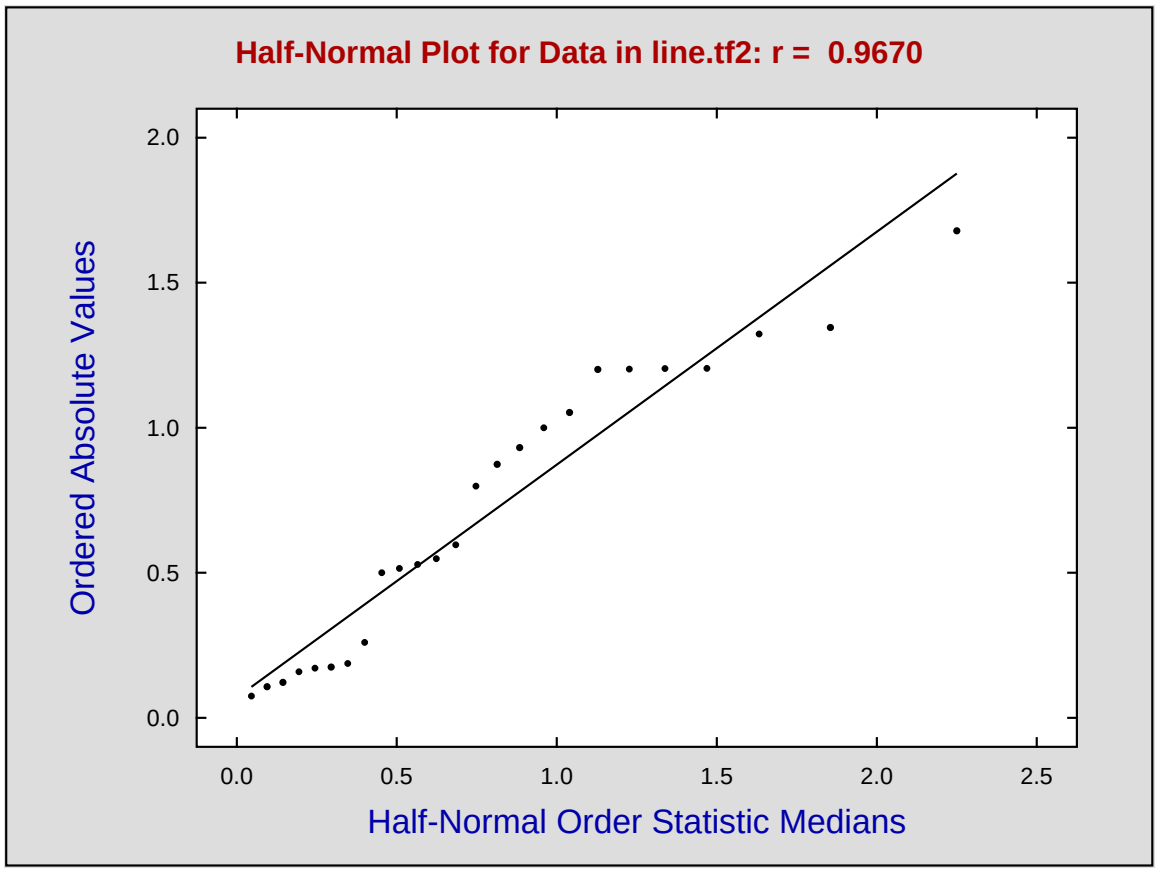
Table 2
This highlights large absolute relative residuals by the following scheme
$******$ > 160%, $*****$ > 80%, $****$ > 40%, $***$ > 20%, $**$ > 10%, $*$ > 5%
indicating that the fit is fairly reasonable, as there are only a few large values and no extremely large absolute relative residuals. Absolute relative residuals are the absolute values of the ratios of residuals to the average of experimental observations and best-fit values, that is

$$\frac{2|y_i - \hat{m}x_i - \hat{c}|}{\max(\epsilon, |y_i| + |\hat{m}x_i + \hat{c}|)}$$

where ϵ is machine precision. These are very useful because they summarize what, to most experimentalists, would be an indicator of how well a model fits the data, even though they do not have any standard statistical interpretation.

Table 3
This presents all the statistics that SimFIT uses to characterize goodness of fit leading to the qualitative, but probably over-enthusiastic, conclusion of a fantastic fit.

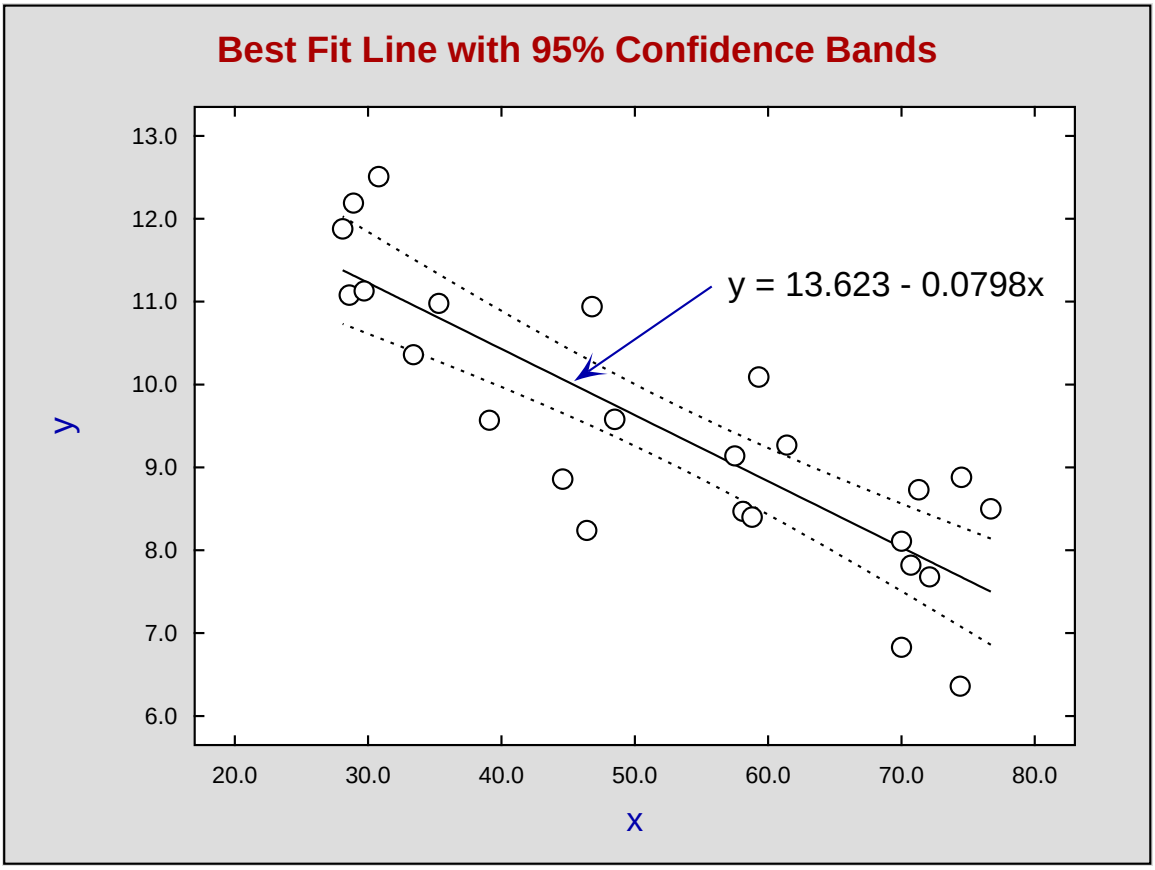
The Half-Normal plot



This shows a typical result with data winding around the best-fit line, and no sign of systematic deviation.

The Best-Fit Line

The next plot shows the data and best fit line $y = \hat{m}x + \hat{c}$ together with the 95% confidence envelope.



A $100(1-\alpha/2)\%$ confidence envelope can be added using the advanced line fitting and calibrating procedure in **linfit**, or by reading the data file into **polnom** and fitting a polynomial of degree one then requesting the addition of confidence limit curves. The confidence envelope is created using the two-valued function

$$f(x) = \hat{m}x + \hat{c} \pm t(n-2, 1-\alpha/2)\left(1 + \frac{1}{n} + \frac{(x-\bar{x})^2}{\sum_{i=1}^{n}(x_i-\bar{x})^2}\right)^{1/2} s$$

where t is the upper 0.975 point of a distribution with $n-2$ degrees of freedom and $\alpha = 0.05$, while s is the variance estimate $SSQ/(n-2)$.

The confidence curves are used by **polnom** to estimate confidence limits for predicting x from y when a best-fit curve us used as a calibration curve.

8.3.3 Fitting a straight line: orthogonal

Orthogonal linear regression is used when there are two variables, X and Y, which both have error of measurement, and/or natural variation due to sampling from a population. Because of this there is no sense in which one variable can be regarded as an independent variable, and the other a dependent variable with added noise: they are really covariates, but they could be sufficiently related to justify fitting a straight line. The problem is to decide the criteria to use when selecting a best-fit line.

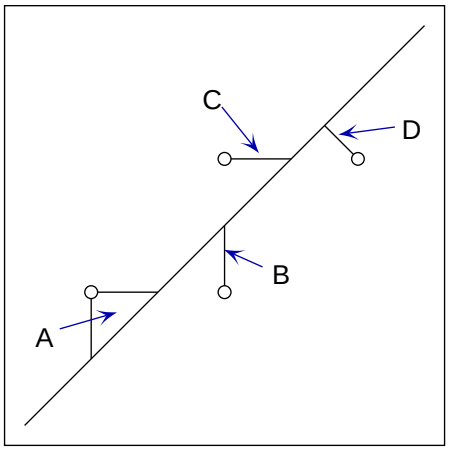
From the SimFIT main menu choose [A/Z], open program **linfit**, choose one of the options for orthogonal or reduced major axis regression and inspect the default test file `swarm.tf1` which has the following data.

x	y
-5.0754	6.4669
-0.1053	11.6754
3.4949	15.4471
3.9864	5.0136
5.1110	7.8573
5.4251	-0.1269
5.7351	2.5006
5.9965	12.8566
6.5293	13.0522
6.6922	11.7522
6.7427	7.9817
8.9142	6.0645
9.6825	19.2638
12.0221	14.5156
14.5866	19.9856
15.4610	17.7134
16.3355	22.4164
16.8102	9.7428
16.9810	23.3692
17.2585	17.3129
18.9608	5.2797
20.2275	16.5656
24.2327	26.7548
25.0702	12.7738
27.6169	25.1028

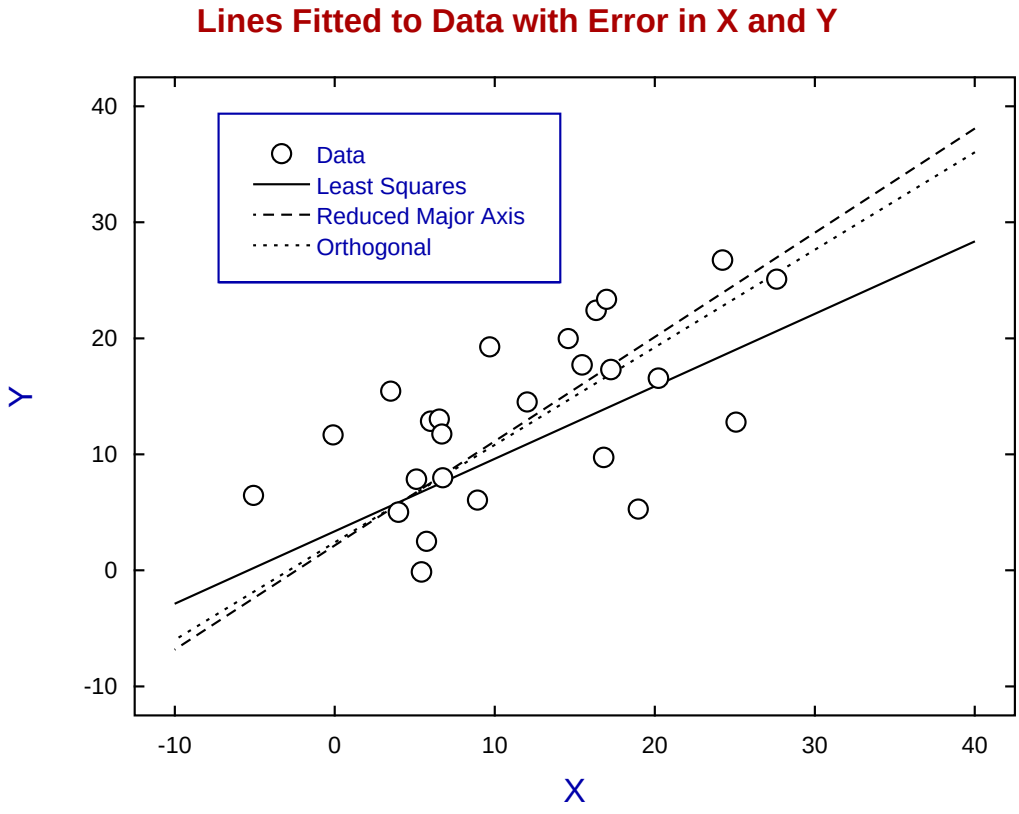
The following straight line procedures could be considered.

1. Least squares fit for $y = ax + b$, i.e. $Y(X)$.
2. Least squares fit for $x = \alpha y + \beta$, i.e. $X(Y)$.
3. Reduced major axis regression.
   This minimizes the sum of the areas of the triangles formed by projecting across and up or down from the data points to the best-fit line.
4. Orthogonal or major axis regression.
   This minimizes the sum of squares of the orthogonal projections from the points to the best-fit line.

Reduced major axis regression minimizes the sum of the areas of the triangles **A**, least squares $Y(X)$ minimizes the sum of squares of the lengths **B**, least squares $X(Y)$ minimizes the sum of squares of lengths **C**, while orthogonal regression, minimizes the sum of squares of the lengths **D** as in the next diagram.



The following plot shows the fit of lines by least squares, reduced major axis, and major axis (orthogonal) regression to the data in test file `swarm.tf1`. In general it seems that if the X and Y data are similarly scaled then the choice depends on the variance of X and Y. If the variance of Y is very much greater than the variance of X then least squares regression of Y on X could be preferred, and when the variance of Y is very much less than that of X then least squares regression of X on Y might be better. With similar variance in Y and Y, as in correlation analysis, then either both linear regression lines should be plotted, or one of the alternatives described in this tutorial should be used if only one line is to be plotted.



Theory

For n pairs (x_i, y_i) with mean $x = \bar{x}$ and mean $y = \bar{y}$, the variances and covariance required are

$$S_{xx} = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$$
$$S_{yy} = \frac{1}{n-1}\sum_{i=1}^{n}(y_i - \bar{y})^2$$
$$S_{xy} = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y}).$$

Also, for an arbitrary point (x_i, y_i) and a straight line defined by $y = a + bx$ the squares of the vertical, horizontal, and orthogonal (i.e. perpendicular) distances, v_i^2, h_i^2, and o_i^2 between the point and the line are

$$v_i^2 = [y_i - (a + bx_i)]^2$$
$$h_i^2 = v_i^2/b^2$$
$$o_i^2 = v_i^2/(1 + b^2).$$

Ordinary least squares

If x is regarded as an exact variable free from random variation or measurement error while y has random variation, then the best fit line from minimizing the sum of v_i^2 is

$$y_1(x) = \hat{\beta}_1 x + [\bar{y} - \hat{\beta}_1 \bar{x}]$$

where $\hat{\beta}_1 = S_{xy}/S_{xx}$. However, if y is regarded as an exact variable while x has random variation, then the best fit line for x as a function of y from minimizing the sum of h_i^2 would be

$$x_2(y) = (1/\hat{\beta}_2)y + [\bar{x} - (1/\hat{\beta}_2)\bar{y}]$$

where $\hat{\beta}_2 = S_{yy}/S_{xy}$ or, rearranging to express the line as $y_2(x)$,

$$y_2(x) = \hat{\beta}_2 x + [\bar{y} - \hat{\beta}_2 \bar{x}],$$

emphasizing that the slope of the regression line for $y_2(x)$ is the reciprocal of the slope for $x_2(y)$. Since neither of these two best fit lines can be regarded as satisfactory, SIMFIT plots both lines such that $y_1(x)$ covers the range of x values while $x_2(y)$ covers the range of y values. However these two lines intersect at $(\bar{x}, \bar{y})$ and, from the fact that the ratio of slopes equals the square of the correlation coefficient, that is,

$$r^2 = \hat{\beta}_1/\hat{\beta}_2,$$

then two best fit lines with similar slopes suggests strong linear correlation, whereas one line almost parallel to the x axis and the other almost parallel to the y axis would indicate negligible linear correlation. For instance, if there is no linear correlation between x and y, then the slope of the regression line for $y(x)$ i.e. $\hat{\beta}_1$ would be zero, as would be the slope of the regression line for $x(y)$ i.e. $1/\hat{\beta}_2$ leading to $r^2 = 0$. Conversely strong linear correlation would lead to $\hat{\beta}_1 = \hat{\beta}_2$ and $r^2 = 1$.

The major axis and reduced major axis lines to be discussed next are attempts to get round the necessity to plot two lines and just have one best fit line intermediate between these two lines to represent the correlation.

The major axis line

Here it is the sum of o_i^2, the squares of the orthogonal distances between the points and the best fit line, that is minimized to yield the slope as

$$\hat{\beta}_3 = \frac{1}{2}\left(\hat{\beta}_2 - (1/\hat{\beta}_1) + \gamma\sqrt{4 + (\hat{\beta}_2 - (1/\hat{\beta}_1))^2}\right)$$

where $\gamma = 1$ if $S_{xy} > 0$, $\gamma = 0$ if $S_{xy} = 0$, and $\gamma = -1$ if $S_{xy} < 0$, so that the major axis line is

$$y_3(x) = \hat{\beta}_3 x + [\bar{y} - \hat{\beta}_3 \bar{x}].$$

Actually $\hat{\beta}_3$ is the slope of the first principal component axis and so it points in the direction of maximum variability.

The reduced major axis line

Instead of minimizing the sum of squares of the vertical distances v_i^2, or horizontal distances h_i^2, it is possible to minimize the sum of the areas of the triangles formed by the v_i, h_i with the best fit line as hypotenuse, i.e. $v_i h_i/2$, to obtain the reduced major axis line as

$$y_4(x) = \hat{\beta}_4 x + [\bar{y} - \hat{\beta}_4 \bar{x}].$$

Here

$$\begin{aligned}\hat{\beta}_4 &= \gamma\sqrt{S_{yy}/S_{xx}} \\ &= \gamma\sqrt{\hat{\beta}_1\hat{\beta}_2}\end{aligned}$$

so that the slope of the reduced major axis line is the geometric mean of the slopes of the regression of y on x and x on y.

Weighting

In the unlikely case that weighting of one of the set is observations is desired, then the variable to be weighted would have to be specified as the Y variable, and weighted fitting could then performed using program **qnfit**.

8.3.4 Fitting a polynomial: weighted least squares polynomial regression

Polynomial regression is used for data smoothing, detecting trends in noisy data, and for creating calibration curves for inverse prediction. It is not much used for modeling data, as polynomial curves are too flexible, they do not accommodate horizontal asymptotes, and they cannot be used for extrapolation. In many applications nowadays they have been replaced by piecewise cubic splines.

From the main SimFIT menu choose the [A/Z] option, open program **polnom**, then browse the default test file `polnom.tf1` which contains the following data set.

x	y	s
0.0	0.098421	0.0056072
0.0	0.10950	0.0056072
0.0	0.10248	0.0056072
2.0	3.8448	0.052139
2.0	3.8647	0.052139
2.0	3.9434	0.052139
4.0	6.8490	0.38867
4.0	6.1469	0.38867
4.0	6.2091	0.38867
6.0	8.5864	0.22982
6.0	9.0156	0.22982
6.0	8.6585	0.22982
8.0	9.8616	0.45524
8.0	9.8748	0.45524
8.0	9.0798	0.45524
10.0	9.5218	0.51790
10.0	9.3098	0.51790
10.0	10.294	0.51790

The columns are for data simulated by SimFIT according to $y = 0.1 + 2.0x + 0.1x^2$ and have the following meanings.

1. The first column contains the independent variable x_i in triplicate.
2. The second column contains the dependent variable y_i arising from evaluating the model equation using SimFIT program **makdat**, then adding 5% relative error using SimFIT program **adderr** to simulate experimental error.
3. The third column are the sample standard deviations s_i calculated by SimFIT program **adderr** to use for weights $w_i = 1/s_i^2$. In the absence of replicates to calculate sample standard deviations for y_i at fixed x_i, the third column could be replaced by $s_i = 1$, or simply omitted, whereupon a default value of $s_i = 1$ would be used for unweighted regression.

Program **polnom** will then proceed to fit polynomials of degree m according to

$$f(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \cdots + \theta_6 x^6$$

for $m = 0, 1, 2, \ldots, k$ where $k \leq 6$ depends on the number of distinct values of x. That is, $m = 0$ for a constant term, $m = 1$ for a straight line, $m = 2$ for a quadratic, $m = 3$ for a cubic, and so on. After fitting each degree, several statistics are output to assess goodness of fit and determine the highest degree that can be justified.

The idea of this systematic procedure is to determine if there is statistical evidence to justify a trend line or progressive curvature in noisy data, or to select a model equation to use as a calibration curve for inverse prediction. To appreciate this aspect consider the following results tables when the data are analyzed.

Table 1: Degree fitted and Chebyshev coefficients

m	A_0	A_1	A_2	A_3	A_4	A_5
0	0.31113					
1	16.034	7.9080				
2	12.737	4.8194	-1.4456			
3	12.735	4.8132	-1.4591	-0.0083774		
4	12.762	4.8342	-1.4387	-0.055083	-0.059600	
5	12.654	4.6602	-1.3858	-0.087456	-0.035275	0.22979

Another table of statistics required to determine the degree of the polynomial required is also displayed as follows.

Table 2: Statistics to determine degree of the fitted polynomial

m	σ	%change	$WSSQ$	%change	$P(\chi^2 \geq WSSQ)$	5%	FV	$P(F \geq FV)$	5%
0	36.703		22901		0.0000	no			
1	8.0833	77.98	1045.4	95.44	0.0000	no	334.50	0.0000	yes
2	0.9914	87.73	14.744	98.59	0.4700	yes	1048.6	0.0000	yes
3	1.0253	3.42	14.718	0.18	0.3977	yes	0.0249	0.8769	no
4	1.0511	2.52	14.363	2.41	0.3488	yes	0.3213	0.5805	no
5	1.0000	4.87	11.999	16.46	0.4457	yes	2.3639	0.1501	no

Here m is the degree fitted, $\sigma = \sqrt{WSSQ/NDOF}$, and FV is the F value for assessing the significance of variance reduction by adding higher degree terms.

There are many results displayed in Tables 1 and 2 in order to suggest the highest degree that can be justified statistically. The qualitative conclusions do not use a Bonferroni correction, but the actual significance levels are also provided for purists. At this point SIMFIT program **polnom** outputs the next table to aid decision.

Table 3: information to help you select a best-fit polynomial

Lowest degree where < 10% change in σ	2
Lowest degree where < 10% change in $WSSQ$	2
Lowest degree by chi-sq. at 5% significance level	2
Lowest degree by chi-sq. at 1% significance level	2
Lowest degree by F test at 5% significance level	2
Lowest degree by F test at 1% significance level	2

Accepting the recommendations of Table 3 leads to Table 4 for the best-fit quadratic.

Table 4: Results for weighted fitting ($w = 1/s^2$)

Parameter	Value	Std. error	Lower95%cl	Upper95%cl	p
θ_0	0.10347	0.0032091	0.096630	0.11031	0.0000
θ_1	2.1203	0.019731	2.0783	2.1624	0.0000
θ_2	-0.11565	0.0035714	-0.12326	-0.10803	0.0000

Correlation matrix

1		
-0.0960	1	
0.0516	-0.8432	1

If you selected to predict x from y the following warning is issued.

> You must be very careful if you wish to use this best-fit curve as a calibration curve for predicting x given y since there are turning points for $X_{min} \leq x \leq X_{max}$ as follows:
>
x-value	y-value
9.1673	9.8224

This is because the quadratic has a turning point within the range of the data, and so predicting x from y could be misleading if a horizontal line for $y = y_0$ for some y_0 intersected the best fit curve twice. So you have to choose whether to search upwards or downwards along the x axis for the prediction required. If a spurious prediction results you have to change the search order. For degrees greater than two there may be multiple turning points, so using degrees greater than two is not normally recommended for inverse prediction. Table 5 results from choosing to predict x from y and evaluate y given x along with 95% confidence ranges using the data supplied in test files `polnom.tf2` and `polnom.tf3`.

Table 5: Predicting x given y and evaluating y given x

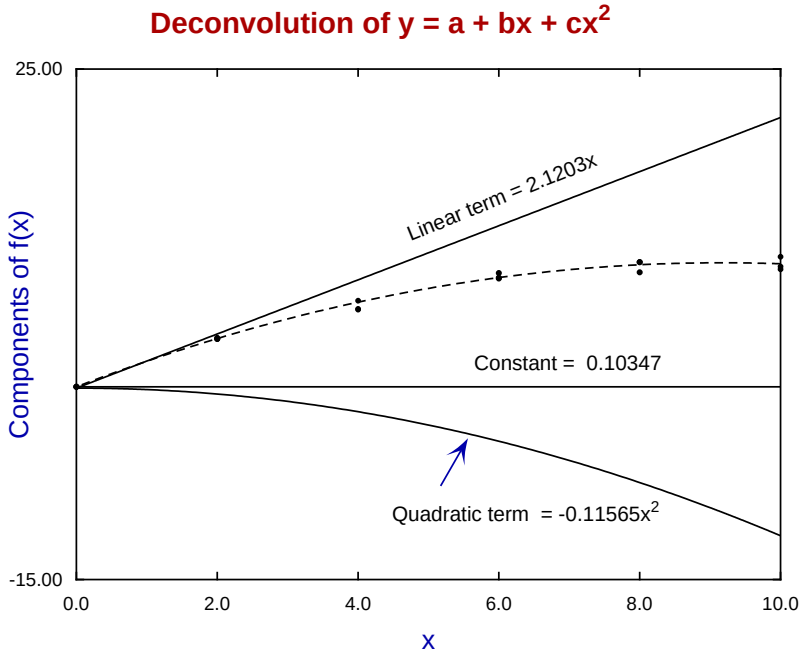
Evaluation data for program **polnom** : x = 2, 4, 6, 8

x-input	y-calculated	95% confidence limits
2.0	3.8816	3.8212, 3.9419
4.0	6.7345	6.6424, 6.8267
6.0	8.6623	8.5137, 8.8108
8.0	9.6649	9.3927, 9.9370

Inverse prediction data for program **polnom** : y = 2, 4, 6, 8

y-measured	x-predicted	95% confidence limits
2.0	0.94293	0.92529, 0.96118
4.0	2.0718	2.0347, 2.1100
6.0	3.4182	3.3566, 3.4819
8.0	5.1976	5.0739, 5.3342

This next graph, constructed using SIMFIT program **qnfit**, shows the data and best-fit quadratic along with contributions of the individual components to the fit.



Theory

It is possible to fit polynomials using multilinear regression with a constant term but with variables defined as $x_1 = x, x_2 = x^2, x_3 = x^3, \ldots, x_m = x^m$. However, this is regarded as an inefficient and numerically inaccurate technique. The best technique is to transform the original variables x into new variables $-1 \leq \tilde{x} \leq 1$ according to

$$\tilde{x} = \frac{2x - x_{max} - x_{min}}{x_{max} - x_{min}}$$

Then a polynomial of degree m is fitted using Chebyshev polynomials as follows

$$g_m(\tilde{x}) = 0.5A_{m+1,1}T_0(\tilde{x}) + A_{m+1,2}T_1(\tilde{x}) + A_{m+1,3}T_2(\tilde{x}) + \cdots + A_{m+1,m+1}T_m(\tilde{x}).$$

In this expression the $T_k(\tilde{x})$ are Chebyshev polynomials of the first kind of degree k defined as follows.

$$\begin{aligned} T_k(x) &= \cos(k\cos^{-1}(x)), k \geq 0 \\ T_{k+1}(x) &= 2xT_k(x) - T_{k-1}x, k \geq 1 \\ \text{For instance, } T_0(x) &= 1 \\ T_1(x) &= x \\ T_2(x) &= 2x^2 + 1 \\ T_3(x) &= 4x^3 - 3x. \end{aligned}$$

The magnitude of the coefficients $A_{m+1,j}$ indicates the contribution of the corresponding Cheyshev polynomial to the corresponding power of x. When fitting polynomials sequentially the coefficients $A_{m+1,j}$ will tend to stabilize for powers of x that are contributing to the fit, but will often tend to diminish as further irrelevant powers are added to the polynomial. So Table 1 provides a quick method for assessing the highest degree polynomial required for a satisfactory fit. Of course, the coefficients and best-fit curve are transformed back into the original space after a satisfactory degree has been decided.

The techniques used by SIMFIT for calculating confidence limits for evaluation and inverse prediction are based on extending the methods used for standard unweighted straight line fitting to the case of fitting polynomials to weighted data.

8.3.5 Multilinear least squares regression

Multilinear regression is resorted to in situations where the value of a variable Y is believed to depend on one or more fixed variables X, but it has not proved possible to develop a mathematical model based on scientific principles. Usually the variation in Y due to experimental error or sampling variation is considerably greater than the variation in X, and in addition variables X are assumed to be uncorrelated, so that Y can be regarded as a dependent variable, and X as independent variables.

From the main SimFIT menu choose the [A/Z] option, open program **linfit**, choose [multilinear regression] using least squares, then browse the default test file `linfit.tf2` which contains the following data set.

x_1	x_2	x_3	x_4	y	s
7.00	26.0	6.00	60.0	78.50	1
1.00	29.0	15.0	52.0	74.30	1
11.0	56.0	8.00	20.0	104.3	1
11.0	31.0	8.00	47.0	87.60	1
7.00	52.0	6.00	33.0	95.90	1
11.0	55.0	9.00	22.0	109.2	1
3.00	71.0	17.0	6.00	102.7	1
1.00	31.0	22.0	44.0	72.50	1
2.00	54.0	18.0	22.0	93.10	1
21.0	47.0	4.00	26.0	115.9	1
1.00	40.0	23.0	34.0	83.80	1
11.0	66.0	9.00	12.0	113.3	1
10.0	68.0	8.00	12.0	109.4	1

The columns have the following meanings.

1. Column 1: % tricalcium aluminate
2. Column 2: % tricalcium silicate
3. Column 3: % tetracalcium alumino ferrite
4. Column 4: % dicalcium silicate
5. Column 5: Heat evolved in calories per gram of cement
6. Column 5: weighting factor.

Note that the weighting factor s must be supplied as the last column so that SimFIT knows how many variables are present. It is usual to set all the values of s to one as in the above example, but if accurate estimates for the standard deviations of Y are known these could be used so that weighted least squares fitting can be done.

To conclude: if the data set supplied has k columns, then it will be presumed that there are $k-2$ independent variables X in columns $1, 2, \ldots, k-2$, the dependent variable Y is in column $k-1$, and the weighting factors are in column k. If these are all equal to one then unweighted regression will be carried out, but otherwise the values s_i will be assumed to be standard errors for the y_i and weighted regression will be performed using $w_i = 1/s_i^2$. Setting $s = 0$ suppresses corresponding rows of data but this is not recommended.

Analysis of these data then leads to the following tables of results using this model

$$y = \beta_0 x_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_{k-2} x_{k-2}$$

where $x_0 = 1$ if β_0 is to be estimated and a constant term is required, or $\beta_0 = 0$ otherwise.

Table 1: Parameter estimates
Number of parameters: 5, Rank: 5, Number of points: 13, Degrees of freedom: 8
Residual-SSQ: 47.864, Mallows' C_p: 5.0, R^2: 0.9824

Parameter	Value	Lower95%cl	Upper95%cl	Std. Error	p	
β_0 (Constant)	62.405	-99.179	223.99	70.071	0.3991	***
β_1	1.5511	-0.16634	3.2685	0.74477	0.0708	*
β_2	0.51017	-1.1589	2.1792	0.72379	0.5009	***
β_3	0.10191	-1.6385	1.8423	0.75471	0.8959	***
β_4	-0.14406	-1.7791	1.4910	0.70905	0.8441	***

The stars shown against the parameter estimates in Table 1 are displayed when the parameter estimates are not significantly different from zero, so this table indicates that none of the five parameters were well determined.

Table 2: Residuals

Number	y-value	Theory	Residual	Leverage	Studentized
1	78.500	78.495	0.0047604	0.55028	0.0029021
2	74.300	72.789	1.5112	0.33324	0.75662
3	104.30	105.97	-1.6709	0.57694	-1.0503
4	87.600	89.327	-1.7271	0.29524	-0.84108
5	95.900	95.649	0.25076	0.35760	0.12791
6	109.20	105.27	3.9254	0.12416	1.7148
7	102.70	104.15	-1.4487	0.36708	-0.74445
8	72.500	75.675	-3.1750	0.40854	-1.6878
9	93.100	91.722	1.3783	0.29431	0.67080
10	115.90	115.62	0.28155	0.70040	0.21029
11	83.800	81.809	1.9910	0.42551	1.0739
12	113.30	112.33	0.97299	0.26298	0.46335
13	109.40	111.69	-2.2943	0.30372	-1.1241

However Table 2 does show a good scatter of residuals about zero with no particular bias or runs indicated.

Table 3: Analysis of Variance

Source	NDOF	SSQ	Mean SSQ	F-value	p
Total	12	2715.8			
Regression	4	2667.9	666.97	111.48	0.0000
Residual	8	47.864	5.9830		

Table 3 is used in much the same way as for simple linear regression and is defined using $\hat{y}_i$ for the best-fit model as follows.

$$SSQ_{total} = \sum_{i=1}^{n} (y_i - \bar{y})^2$$

$$SSQ_{regression} = \sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2$$

$$SSQ_{residual} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

The F value is the ratio of mean regression SSQ to mean residual SSQ, and the significance level p is used to test the null hypothesis

$$H_0 : \beta_i = 0 \text{ for all } i$$

against the alternative hypothesis

$$H_A : \beta_i \neq 0 \text{ for one or more } i.$$

Clearly it must be concluded that, although none of the individual parameters were well determined as judged by the t tests on the ratios of estimates to standard errors, there is a significant overall reduction in the sum of squares by some combinations of variables.

At this point it is customary to see if a satisfactory regression could be achieved with fewer parameters, and a variety of techniques are available to perform such subset regression to find the best explanation of the data in terms of the smallest number of variables. When this is done systematically with large data sets it generates an enormous amount of analysis, which is not normally justified because usually the experimentalist would have a good idea which subsets of variables to try. This is fairly easy to do interactively in SIMFIT by suppressing variables until a fit is achieved where all the parameters are significantly different from zero with the two-tail t test, and the fit is justified by the C_p values.

The way to interpret the Mallows' C_p values should be explained. Program **linfit** first fits a full model and the results from this analysis are saved. This fit is assumed to be the best possible for estimating the variance and the effect of suppressing any variable can then be seen by comparing the effect on the C_p value. From fitting the full model the C_p value will be equal to the total number of parameters and subsequent subset regressions can be judged by the ratio of the C_p values to the number of parameters, where values much greater than the number of parameters estimated suggest a deficient model.

The next table summarizes the results from fitting a constant only, followed by fitting subsets of the additional 1, 2, and 3 variables.

Table 4: C_p values

Variables	C_p	Parameters
Constant only	442.9	1
+1,+2,+3,+4	202.5, 142.5, 315.2, 138.7	2
+12, +13, +14	2.7, 198.1, 5.5	3
+23, +24, +34	62.4, 138.2, 22.4	3
+123, +124, +134, +234	3.0, 3.0, 3.5, 7.3	4
+1234	5.0	5

In Table 4 the first column indicates the subscripts of the variables added to the constant term, column 2 holds the corresponding C_p variables, while column 3 contains the total number of parameters varied including the constant term. Values where C_p divided by the number of parameters in the regression are less than or equal to one are highlighted, and it is perfectly clear that the combination of a constant plus variables 1 and 2 seems to be strongly recommended.

Here, for example, are the results from suppressing variables 3 and 4.

Table 1A: parameter estimates with variables 3 and 4 suppressed
Number of parameters: 3, Rank: 3, Number of points: 13, Degrees of freedom: 10
Residual-SSQ: 57.904, Mallows' C_p: 2.6782, R^2: 0.9787

Parameter	Value	Lower95%cl	Upper95%cl	Std. Error	p
β_0 (Constant)	52.577	47.483	57.671	2.2862	0.0000
β_1	1.4683	1.1980	1.7386	0.12130	0.0000
β_2	0.66225	0.56008	0.76442	0.045855	0.0000

Table 2A: Residuals with variables 3 and 4 suppressed

Number	y-value	Theory	Residual	Leverage	Studentized
1	78.500	80.074	-1.5740	0.25119	-0.75590
2	74.300	73.251	1.0491	0.26189	0.50745
3	104.30	105.81	-1.5147	0.11890	-0.67061
4	87.600	89.258	-1.6585	0.24225	-0.79175
5	95.900	97.293	-1.3925	0.83616	-0.60451
6	109.20	105.15	4.0475	0.11512	1.7881
7	102.70	104.00	-1.3021	0.36180	-0.67732
8	72.500	74.575	-2.0754	0.24119	-0.99011
9	93.100	91.275	1.8245	0.17915	0.83687
10	115.90	114.54	1.3625	0.55002	0.84405
11	83.800	80.536	3.2643	0.18402	1.5018
12	113.30	112.44	0.86276	0.19666	0.40002
13	109.40	112.29	-2.8934	0.21420	-1.3564

Table 3A: Analysis of Variance with variables 3 and 4 suppressed

Source	NDOF	SSQ	Mean SSQ	F-value	p
Total	12	2715.8			
Regression	2	2657.9	1328.9	229.50	0.0000
Residual	10	57.904	5.7904		

Comparing the results with all variables present to those with variables 3 and 4 suppressed leads to the following conclusions.

1. **Table 1 compared to Table 1A**
   With all variables present no parameters were well-determined by a two-tail t test, but with variables 3 and 4 suppressed all parameters were well-determined.

2. **Table 2 compared to Table 2A**
   There are no differences to indicate a poorer fit with the simpler model.

3. **Table 3 compared to Table 3A**
   There are no differences to indicate a poorer fit with the simpler model.

Sometimes it is useful to evaluate the best-fit model and, as variables 3 and 4 do not seem to be making an important contribution prediction, this will be demonstrated using the model with variables 3 and 4 suppressed. A vector of default values equal to 1 is supplied and this can be edited interactively to change the variables as follows with variable 2.

Using the best-fit model to predict y given x
x_0 = 1.0, coefficient = 52.577 (the constant term)
x_1 = 1.0, coefficient = 1.4683
x_2 = 1.0, coefficient = 0.66225
$y(x)$ = 54.708

x_0 = 1.0, coefficient = 52.577 (the constant term)
x_1 = 1.0, coefficient = 1.4683
x_2 = 5.0, coefficient = 0.66225
$y(x)$ = 57.357
Of course, users cannot alter the value of x_0 which is always equal to 1, and included or excluded from the regression depending on whether a constant (i.e. intercept) term is included.

Note that, for more advanced analysis of such data sets (including prediction and inverse prediction) the SimFIT partial least squares procedure should be used.

Theory

Program **linfit** fits a multilinear model in the form

$$y = \beta_0 x_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_m x_m,$$

where $x_0 = 1$, but you can choose interactively whether or not to include a constant term β_0, you can decide which variables are to be included, and you can use a weighting scheme if this is required. For each regression sub-set, you can observe the parameter estimates and standard errors, R-squared, Mallows C_p , and $ANOVA$ table, to help you decide which combinations of variables are the most significant. Unlike nonlinear regression, multilinear regression, is based on the assumptions

$$\begin{aligned} Y &= X\beta + \epsilon, \\ \mathrm{E}(\epsilon) &= 0, \\ \mathrm{Var}(\epsilon) &= \sigma^2 I, \end{aligned}$$

where X is the over-determined data matrix (e.g., the 13 rows and first 4 columns of test file `linfit.tf2`), Y is the observation vector (e.g., column 5 of test file `linfit.tf2`), β is the parameter vector and ϵ is the error vector. This allows us to introduce the hat matrix

$$H = X(X^T X)^{-1} X^T,$$

then define the leverages h_{ii}, which can be used to asses influence, and the studentized residuals

$$R_i = \frac{r_i}{\hat{\sigma}\sqrt{1 - h_{ii}}},$$

which may offer some advantages over ordinary residuals r_i for goodness of fit assessment from residuals plots. In the event of weighting being required, Y, X and ϵ above are simply replaced by $W^{\frac{1}{2}}Y$, $W^{\frac{1}{2}}X$, and $W^{\frac{1}{2}}\epsilon$, where W is the diagonal weighting matrix.

Note that examining parameter reliability using the t test as in Tables 1 and 1A and also model discrimination analysis using the F test is applicable for nested linear models as fitted by SimFIT program **linfit**. So several additional options are provide by **linfit** to perform such further investigations. For instance, to perform an F test for excess variance note that, if $WSSQ_1$ with m_1 parameters is the previous (possibly deficient) model, while $WSSQ_2$ with m_2 parameters is the current (possibly superior) model, so that $WSSQ_1 > WSSQ_2$, and $m_1 < m_2$, then

$$F = \frac{(WSSQ_1 - WSSQ_2)/(m_2 - m_1)}{WSSQ_2/(N - m_2)}$$

should be F distributed with $m_2 - m_1$ and $N - m_2$ degrees of freedom, and the F test for excess variance can be used. Alternatively, if $WSSQ2/(N - m_2)$ is equivalent to the true variance, i.e., model 2 is equivalent to the true model, the Mallows' C_p statistic

$$C_p = \frac{WSSQ_1}{WSSQ_2/(N - m_2)} - (N - 2m_1)$$

can be considered. This has expectation m_1 if the previous model is sufficient, so values greater than m_1, that is $C_p/m_1 > 1$, indicate that the current model should be preferred over the previous one. In the **linfit** results tables C_p values refer to the full model being fitted as the reference case.

Finally it should be noted that, for successful analysis of data, the units used to provide values of X should be such that the numerical values are of similar size. If categorical data are mixed with continuous data, or the data set is ill-conditioned, or less than full rank for any reason, a linear model will still be fitted using the singular value decomposition. However, in such cases **linfit** will issue a warning that the estimated parameters are not independent, and the data should be re-scaled, or the number or variables should be reduced until the columns of the X matrix are independent, i.e. the rank is at least as large as the number of parameters estimated.

8.3.6 Partial least squares (PLS)

Partial least squares is also known as regression by projection to latent structures, or simply PLS, and it is sometimes useful when a n by r matrix of responses Y, with $r \geq 1$, is observed with a n by m matrix of predictor variables X, with $m > 1$, and one or more of the following conditions may apply:

1. There is no deterministic model to express the r columns of Y as functions of the m columns of the matrix X.
2. The number of columns of X is too large for convenient analysis, or the number of observations n is not significantly greater than the number of predictor variables m, e.g. the rank of X is less than m.
3. The X variables may be correlated and/or the Y variables may be correlated.

The idea behind PLS is to express the X and Y matrices in terms of sets of k factors, with $k \leq m$, derived from the matrices by projection and regression techniques. The X scores would have maximum covariance with the Y scores, and the principal problem is to decide on a sufficiently small dimension l, with $l \leq k$, that would be needed to represent the relationship between Y and X adequately. Having obtained satisfactory expressions for approximating X and Y using these factors, they can then be used to treat X as a training matrix, then predict what new Y would result from a new n by m matrix Z that is expressed in the same variables as the training matrix X. Hence the use of this technique in multivariate calibration, or quantitative structure activity relationships (QSAR).

Data format

From the main SimFIT menu choose [A/Z], open program **linfit**, then select [PLS] and inspect the two test files. The file `g02laf.tf1` contains the following 15 by 15 matrix of X data

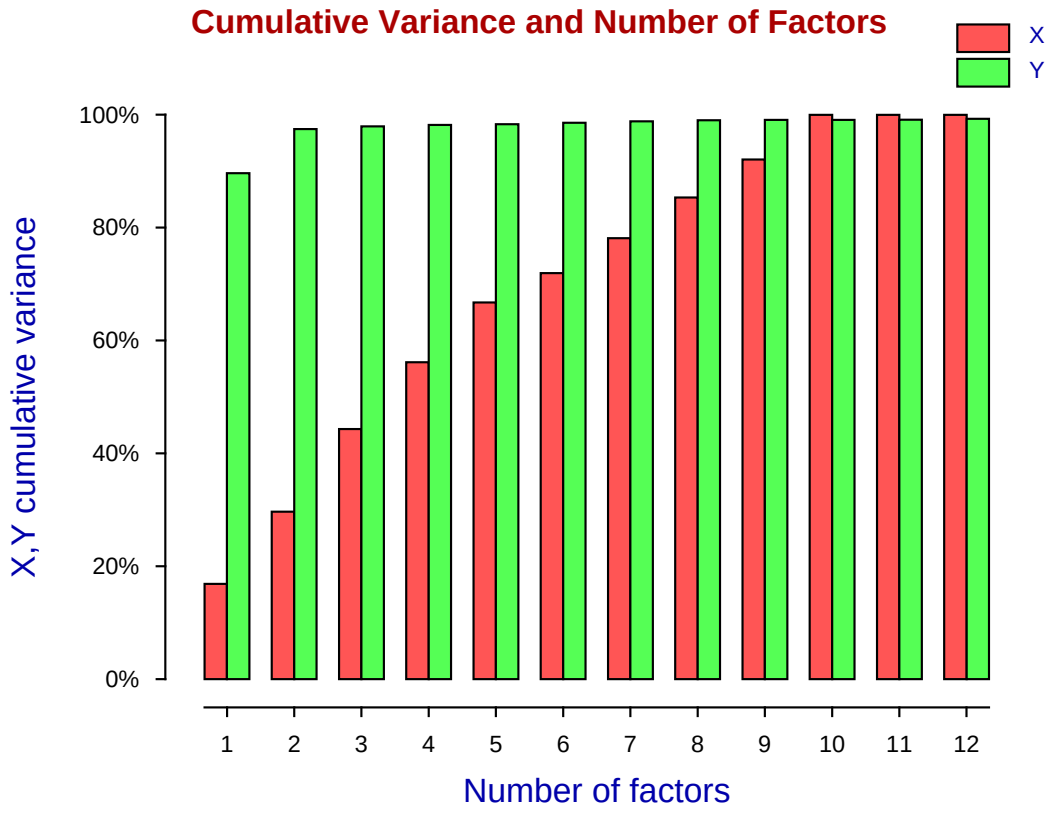
X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	X_{11}	X_{12}	X_{13}	X_{14}	X_{15}
-2.6931	-2.5271	-1.2871	3.0777	0.3891	-0.0701	1.9607	-1.6324	0.5746	1.9607	-1.6324	0.5740	2.8369	1.4092	-3.1398
-2.6931	-2.5271	-1.2871	3.0777	0.3891	-0.0701	1.9607	-1.6324	0.5746	0.0744	-1.7333	0.0902	2.8369	1.4092	-3.1398
-2.6931	-2.5271	-1.2871	3.0777	0.3891	-0.0701	0.0744	-1.7333	0.0902	1.9607	-1.6324	0.5746	2.8369	1.4092	-3.1398
-2.6931	-2.5271	-1.2871	3.0777	0.3891	-0.0701	0.0744	-1.7333	0.0902	0.0744	-1.7333	0.0902	2.8369	1.4092	-3.1398
-2.6931	-2.5271	-1.2871	2.8369	1.4092	-3.1398	0.0744	-1.7333	0.0902	0.0744	-1.7333	0.0902	2.8369	1.4092	-3.1398
-2.6931	-2.5271	-1.2871	3.0777	0.3891	-0.0701	-4.7548	3.6521	0.8524	0.0744	-1.7333	0.0902	2.8369	1.4092	-3.1398
-2.6931	-2.5271	-1.2871	3.0777	0.3891	-0.0701	0.0744	-1.7333	0.0902	0.0744	-1.7333	0.0902	-1.2201	0.8829	2.2253
-2.6931	-2.5271	-1.2871	3.0777	0.3891	-0.0701	2.4064	1.7438	1.1057	0.0744	-1.7333	0.0902	2.8369	1.4092	-3.1398
-2.6931	-2.5271	-1.2871	0.0744	-1.7333	0.0902	0.0744	-1.7333	0.0902	0.0744	-1.7333	0.0902	2.8369	1.4092	-3.1398
2.2261	-5.3648	0.3049	3.0777	0.3891	-0.0701	0.0744	-1.7333	0.0902	0.0744	-1.7333	0.0902	2.8369	1.4092	-3.1398
-4.1921	-1.0285	-0.9801	3.0777	0.3891	-0.0701	0.0744	-1.7333	0.0902	0.0744	-1.7333	0.0902	2.8369	1.4092	-3.1398
-4.9217	1.2977	0.4473	3.0777	0.3891	-0.0701	0.0744	-1.7333	0.0902	0.0744	-1.7333	0.0902	2.8369	1.4092	-3.1398
-2.6931	-2.5271	-1.2871	3.0777	0.3891	-0.0701	2.2261	-5.3648	0.3049	2.2261	-5.3648	0.3049	2.8369	1.4092	-3.1398
-2.6931	-2.5271	-1.2871	3.0777	0.3891	-0.0701	-4.9217	1.2977	0.4473	0.0744	-1.7333	0.0902	2.8369	1.4092	-3.1398
-2.6931	-2.5271	-1.2871	3.0777	0.3891	-0.0701	-4.1921	-1.0285	-0.9801	0.0744	-1.7333	0.0902	2.8369	1.4092	-3.1398

while the test file `g02laf.tf2` contains the following 15 by 1 matrix with Y data.

Y_1
0.00
0.28
0.20
0.51
0.11
2.73
0.18
1.53
-0.10
-0.52
0.40
0.30
-1.00
1.57
0.59

Choosing the number of factors

Select a maximum of 12 factors then plot the cumulative variance plot as in the next figure.



This graph shows that most of the variance in the Y data (green bars) can be explained by only two factors while at least six to eight factors are required to account for a significant proportion of the variance in the X data (red bars).

The main point of PLS analysis is to choose the number of factors that subsequently will be used in the predictive procedures, as the number of factors will be the dimension of the subspace used in the projection of the X data into a space of smaller dimension.

It must be stressed that these factors are like principal components in that every factor is a linear combination of all the variables, and SIMFIT provides several ways to determine the influence of the original variables in the factors.

Contribution of variables to the projection

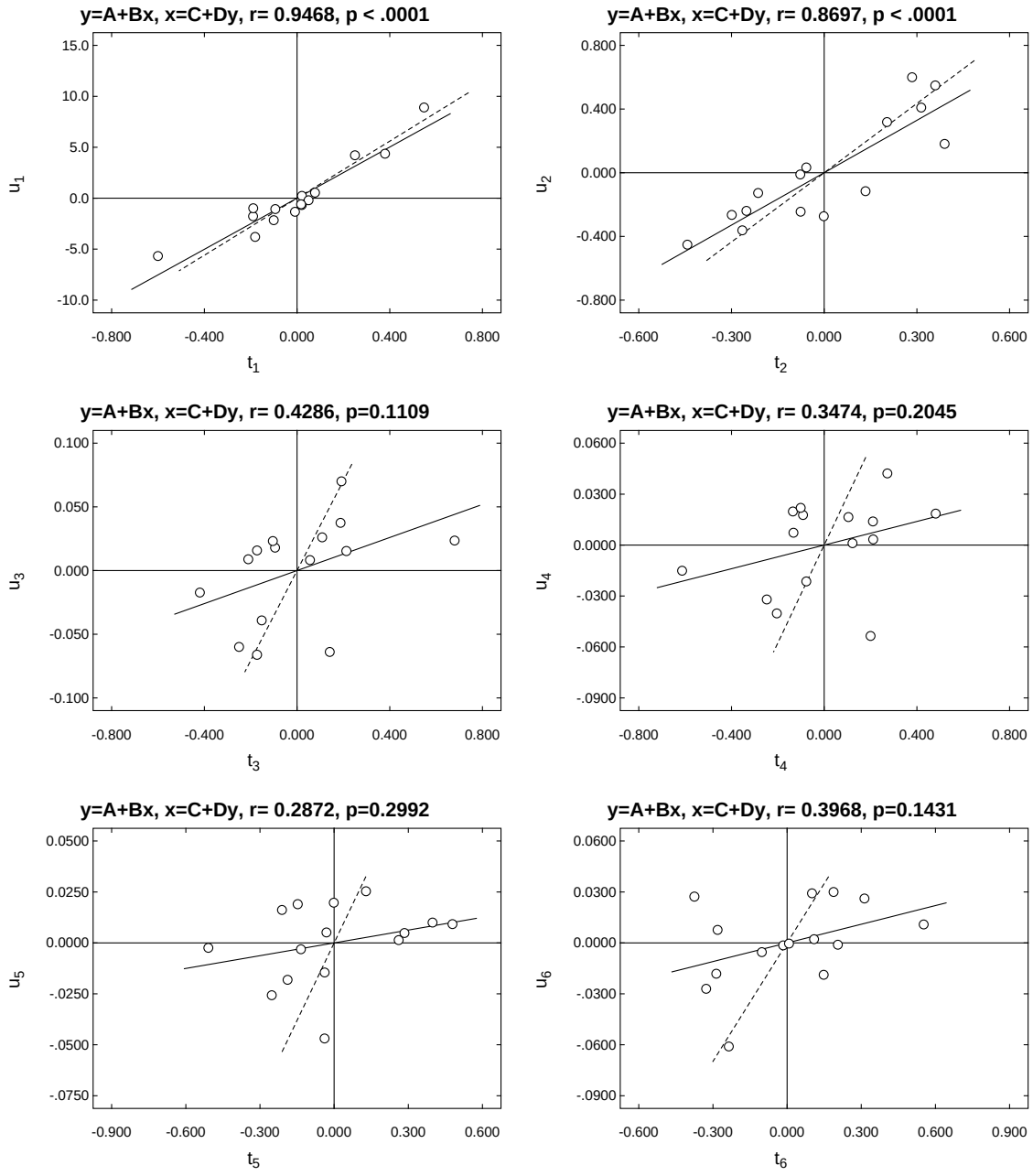
For example, the next table shows the variable influence on projection (VIP) results which indicate that variables 7, 8, 9, 10, and 11 seem to make the most significant contribution to the factors. That is because the sum of squared VIP values equals the number of X variables and a large $VIP(i)$ value indicates that variable i has an important influence on projection.

Variable	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
VIP	0.611	0.318	0.751	0.504	0.272	0.359	1.577	2.435	1.132	1.223	1.180	0.884	0.213	0.213	0.213
							*	*	*	*	*				

Correlation between scores

Another way to assess the number of factors required to adequately represent the model is to examine the correlation between the scores as, unlike with principal components which are select to maximize variance, PLS factors are selected to maximize covariance between factors.

In the next figure are plotted the successive correlations between the X and Y scores. Each plot shows the best fit linear regression for the u_i i.e. Y scores on the t_i i.e. X scores, and also the best fit linear regression of the X scores on the Y scores, together with the correlation coefficients r and and significance levels p.



Clearly the scores corresponding to the first two factors are highly correlated, but thereafter the correlation is very weak.

Predicting Y given new X

Once a model has been selected with an appropriate number of factors, a set of parameters can be calculated to express Y as a function of the X matrix. In other words, the original X and Y matrices can be regarded as a training set, then a new X data matrix can be input to predict a new Y matrix of responses.

For instance, select a model with 7 factors and then read in the default test file `g02laf.tf3` as a Z matrix which has the following data.

Z_1	Z_2	Z_3	Z_4	Z_5	Z_6	Z_7	Z_8	Z_9	Z_{10}	Z_{11}	Z_{12}	Z_{13}	Z_{14}	Z_{15}
-2.6931	-2.5271	-1.2871	3.0777	0.3891	-0.0701	1.9607	-1.6324	0.5746	1.9607	-1.6324	0.574	2.8369	1.4092	-3.1398
-2.6931	-2.5271	-1.2871	3.0777	0.3891	-0.0701	1.9607	-1.6324	0.5746	0.0744	-1.7333	0.0902	2.8369	1.4092	-3.1398
-2.6931	-2.5271	-1.2871	3.0777	0.3891	-0.0701	0.0744	-1.7333	0.0902	1.9607	-1.6324	0.5746	2.8369	1.4092	-3.1398
-2.6931	-2.5271	-1.2871	3.0777	0.3891	-0.0701	0.0744	-1.7333	0.0902	0.0744	-1.7333	0.0902	2.8369	1.4092	-3.1398
-2.6931	-2.5271	-1.2871	2.8369	1.4092	-3.1398	0.0744	-1.7333	0.0902	0.0744	-1.7333	0.0902	2.8369	1.4092	-3.1398
-2.6931	-2.5271	-1.2871	3.0777	0.3891	-0.0701	-4.7548	3.6521	0.8524	0.0744	-1.7333	0.0902	2.8369	1.4092	-3.1398
-2.6931	-2.5271	-1.2871	3.0777	0.3891	-0.0701	0.0744	-1.7333	0.0902	0.0744	-1.7333	0.0902	-1.2201	0.8829	2.2253
-2.6931	-2.5271	-1.2871	3.0777	0.3891	-0.0701	2.4064	1.7438	1.1057	0.0744	-1.7333	0.0902	2.8369	1.4092	-3.1398
-2.6931	-2.5271	-1.2871	0.0744	-1.7333	0.0902	0.0744	-1.7333	0.0902	0.0744	-1.7333	0.0902	2.8369	1.4092	-3.1398
2.2261	-5.3648	0.3049	3.0777	0.3891	-0.0701	0.0744	-1.7333	0.0902	0.0744	-1.7333	0.0902	2.8369	1.4092	-3.1398
-4.1921	-1.0285	-0.9801	3.0777	0.3891	-0.0701	0.0744	-1.7333	0.0902	0.0744	-1.7333	0.0902	2.8369	1.4092	-3.1398
-4.9217	1.2977	0.4473	3.0777	0.3891	-0.0701	0.0744	-1.7333	0.0902	0.0744	-1.7333	0.0902	2.8369	1.4092	-3.1398
-2.6931	-2.5271	-1.2871	3.0777	0.3891	-0.0701	2.2261	-5.3648	0.3049	2.2261	-5.3648	0.3049	2.8369	1.4092	-3.1398
-2.6931	-2.5271	-1.2871	3.0777	0.3891	-0.0701	-4.9217	1.2977	0.4473	0.0744	-1.7333	0.0902	2.8369	1.4092	-3.1398
-2.6931	-2.5271	-1.2871	3.0777	0.3891	-0.0701	-4.1921	-1.0285	-0.9801	0.0744	-1.7333	0.0902	2.8369	1.4092	-3.1398

This can then be used to predict a new set of responses, say $\tilde{Y}$, with the following values.

$\tilde{Y}$
0.1408592
0.2991056
0.1383467
0.2967188
0.1304868
2.6278021
0.1862098
1.4820439
-0.1140479
-0.4532927
0.4052814
0.2922055
-1.0294603
1.7814955
0.5962458

When using this technique it is important to realize that the training matrices and the prediction matrices must be centered and scaled using exactly the same mean vectors and scaling factors otherwise biased predictions will result. That is why it is best to submit the data without centering and scaling then SIMFIT will automatically use the centering and scaling from the training set with the prediction data and will map the predicted results back into the original coordinates. This will be explained with more detail in the theoretical section that follows.

Theory

The idea behind PLS is to express the n by m matrix X and n by r matrix Y in terms of sets of k factors, with $k \leq m$, derived from the matrices by projection and regression techniques. The X scores would have maximum covariance with the Y scores, and the principal problem is to decide on a sufficiently small dimension l, with $l \leq k$, that would be needed to represent the relationship between Y and X adequately.

If X_1 is the centered matrix obtained from X by subtracting the X column means, and Y_1 is obtained from Y by subtracting the Y column means, then the first factor is obtained by regressing on a column vector of n normalized scores t_1, as in

$$\begin{aligned} \hat{X}_1 &= t_1 p_1^T \\ \hat{Y}_1 &= t_1 c_1^T \\ t_1^T t_1 &= 1, \end{aligned}$$

where the column vectors of m x-loadings p_1 and r y-loadings c_1 are calculated by least squares, i.e.

$$\begin{aligned} p_1^T &= t_1^T X_1 \\ c_1^T &= t_1^T Y_1. \end{aligned}$$

The x-score vector $t_1 = X_1 w_1$ is the linear combination of X_1 that has maximum covariance with the y-scores $u_1 = Y_1 c_1$, where the x-weights vector w_1 is the normalized first left singular vector of $X_1^T Y_1$. The further $k - 1$ orthogonal factors are then calculated successively using

$$\begin{aligned} X_i &= X_{i-1} - \hat{X}_{i-1} \\ Y_i &= Y_{i-1} - \hat{Y}_{i-1}, \ i = 2, 3, \ldots, k \\ t_i^T t_j &= 0, \ j = 1, 2, \ldots, i-1. \end{aligned}$$

Once a set of k factors has been calculated, these can be used to generate the parameter estimates necessary to predict a new Y matrix from a Z matrix, given the original training matrix X. Usually k would be an upper limit on the number of factors to consider, and the m by r parameter estimates matrix B required for l factors, where $l \leq k$, would be given by

$$B = W(P^T W)^{-1} C^T.$$

Here W is the m by k matrix of x-weights, P is the m by k matrix of x-loadings, and C is the r by k matrix of y-loadings. Note that B calculated in this way is for the centered matrices X_1 and Y_1, but parameter estimates appropriate for the original data are also calculated.

Before proceeding further it is important to emphasize a complication which can arise when predicting a new Y matrix using the parameter estimates. In most multivariate techniques it is immaterial whether the data are scaled and centered before submitting a sample for analysis, or whether the data are scaled and centered internally by the software. In the case of PLS, the Y predicted will be incorrect if the data are centered and scaled independently before analysis, but then the Z matrix for prediction is centered and scaled using its own column means and variances.

So there are just two ways to make sure PLS predicts correctly.

1. You can submit X and Y matrices that are already centered and scaled, but then you must submit a Z matrix that has not been centered and scaled using its own column means and standard deviations, but one that has been processed by subtracting the original X column means and scaled using the original X column standard deviations.

2. Do not center or scale any data. Just submit the original data for analysis, request automatic centering and scaling if necessary, but allow the software to then center and scale internally.

As the first method is error prone and will predict scaled and centered predictions, which could be confusing, the advice to PLS users would be:

> *Do not center or scale any training sets, or Z-data for predicting new Y, before PLS analysis.*
> *Always submit raw data and allow the software to perform centering and scaling.*
> *That way predictions will be in coordinates corresponding to the original Y-coordinates.*

Several techniques are available to decide how many factors l out of the maximum calculated k should be selected when using a training set for prediction.

For instance, the previous figure displaying cumulative variance was obtained by using test file `g02laf.tf1` with 15 rows and 15 columns as the source of X prediction data, and test file `g02laf.tf2` with 15 rows and just 1 column as the source of Y response data, then fitting a PLS model with up to a maximum of $k = 12$ factors. It illustrates how the cumulative percentage of variance in X and a column of Y is accounted for the factor model as the number of factors is steadily increased. It is clear that two factors are sufficient to account for the variance of the single column of Y in this case but more, probably about 6 to 8, are required to account for the variance in the X matrix, i.e. we should choose $6 \leq l \leq 8$.

Alternatively, the previous figures showing the successive correlations between the X and Y scores should be inspected. Each plot shows the best fit linear regression for the u_i i.e. Y scores on the t_i i.e. X scores, and also the best fit linear regression of the X scores on the Y scores, together with the correlation coefficients r and and significance levels p. Clearly the scores corresponding to the first two factors are highly correlated, but thereafter the correlation is very weak.

Note that the PLS model can be summarized as follows

$$\begin{aligned} X &= \bar{X} + TP^T + E \\ Y &= \bar{Y} + UC^T + F \\ U &= T + H \end{aligned}$$

where E, F, and H are matrices of residuals.

So the SIMFIT PLS routines also allow users to study such residuals, to see how closely the fitted model predicts the original Y data for increasing numbers of factors before the number of factors to be used routinely is decided. Various tests for goodness of fit can be derived from these residuals and, in addition, variable influence on projection (VIP) statistics can also be calculated.

8.4 Generalized linear models (GLM)

Tutorials and worked examples for simulation, curve fitting, statistical analysis, and plotting.
https://simfit.uk
https://simfit.org.uk
https://simfit.silverfrost.com

8.4.1 Summary of GLM techniques

Generalized linear modeling (GLM) is used to fit statistical models to observations that have a known error distribution, and where covariates can be assumed to contribute in a linear manner via a specified intermediate link function.

Introduction

The GLM technique is intermediate between linear regression, which is trivial and gives uniquely determined parameter estimates but is rarely appropriate, and nonlinear regression, which is very hard and does not usually give unique parameter estimates, but is justified with normal errors and a known model.

To understand the motivation for this technique, it is usual to refer to a typical doubling dilution experiment in which diluted solutions from a stock containing infected organisms are plated onto agar in order to count infected plates, and hence estimate the number of organisms in the stock. Suppose that before dilution the stock had N organisms per unit volume, then the number per unit volume after $x = 0, 1, \ldots, m$ dilutions will follow a Poisson dilution with $\mu_x = N/2^x$. Now the chance of a plate receiving no organisms at dilution x is the first term in the Poisson distribution, that is $\exp(-\mu_x)$, so if p_x is the probability of a plate becoming infected at dilution x, then

$$p_x = 1 - \exp(-\mu_x),\ x = 1, 2, \ldots, m.$$

Evidently, where the p_x have been estimated as proportions from y_x infected plates out of n_x plated at dilution x, then N can be estimated using

$$\log[-\log(1 - p_x)] = \log N - x \log 2$$

considered as a maximum likelihood fitting problem of the type

$$\log[-\log(1 - p_x)] = \beta_0 + \beta_1 x$$

where the errors in estimated proportions $p_x = y_x/n_x$ are binomially distributed.

The SimFIT generalized models interface can be used from **gcfit**, **linfit** or **simstat** as it finds many applications, ranging from bioassay to survival analysis.

Basic theory

So, to fit a generalized linear model, you must have independent evidence to support your choice for an assumed error distribution for the dependent variable Y from the following possibilities:

- normal
- binomial
- Poisson
- gamma

in which it is supposed that the expectation of Y is to be estimated, i.e.,

$$E(Y) = \mu.$$

The associated *pdfs* are parameterized as follows.

$$\begin{aligned}
\text{normal}: f_Y &= \frac{1}{\sqrt{2\pi}\,\sigma}\exp\left(-\frac{(y-\mu)^2}{2\sigma^2}\right)\\
\text{binomial:}\ f_Y &= \binom{N}{y}\pi^y(1-\pi)^{N-y}\\
\text{Poisson:}\ f_Y &= \frac{\mu^y\exp(-\mu)}{y!}\\
\text{gamma:}\ f_Y &= \frac{1}{\Gamma(\nu)}\left(\frac{\nu y}{\mu}\right)^{\nu}\exp\left(-\frac{\nu y}{\mu}\right)\frac{1}{y}
\end{aligned}$$

It is a mistake to make the usual unwarranted assumption that measurements imply a normal distribution, while proportions imply a binomial distribution, and counting processes imply a Poisson distribution, unless the error distribution assumed has been verified for your data. Another very questionable assumption that has to made is that a predictor function η exists, which is a linear function of the m covariates, i.e., independent explanatory variables, as in

$$\eta = \sum_{j=1}^{m}\beta_j x_j.$$

Finally, yet another dubious assumption must be made, that a link function $g(\mu)$ exists between the expected value of Y and the linear predictor. The choice for

$$g(\mu) = \eta$$

depends on the assumed distribution as follows. For the binomial distribution, where y successes have been observed in N trials, the link options are the logistic, probit or complementary log-log

$$\begin{aligned}
\text{logistic:}\ \eta &= \log\left(\frac{\mu}{N-\mu}\right)\\
\text{probit:}\ \eta &= \Phi^{-1}\left(\frac{\mu}{N}\right)\\
\text{complementary log-log:}\ \eta &= \log\left(-\log\left(1-\frac{\mu}{N}\right)\right).
\end{aligned}$$

Where observed values can have only one of two values, as with binary or quantal data, it may be wished to perform binary logistic regression. This is just the binomial situation where y takes values of 0 or 1, N is always set equal to 1, and the logistic link is selected. However, for the normal, Poisson and gamma distributions the link options are

$$\begin{aligned}
\text{exponent:}\ \eta &= \mu^a\\
\text{identity:}\ \eta &= \mu\\
\text{log:}\ \eta &= \log(\mu)\\
\text{square root:}\ \eta &= \sqrt{\mu}\\
\text{reciprocal:}\ \eta &= \frac{1}{\mu}.
\end{aligned}$$

In addition to these possibilities, you can supply weights and install an offset vector along with the data set, the regression can include a constant term if requested, the constant exponent a in the exponent link can be altered, and variables can be selected for inclusion or suppression in an interactive manner. However, note that the same strictures apply as for all regressions: you will be warned if the SVD has to be used due to rank deficiency and you should redesign the experiment until all parameters are estimable and the covariance matrix has full rank, rather than carry on with parameters and standard errors of limited value.

The simplified GLM interface

Although generalized linear models have widespread use, specialized knowledge is sometimes required to prepare the necessary data files, weights, offsets, etc.

For this reason, there is a simplified SIMFIT interface to facilitate the use of GLM techniques in such fields as the following.

- Bioassay, assuming a binomial distribution and using logistic, probit, or log-log models to estimate percentiles, such as the LD50.
- Logistic regression and binary logistic regression.
- Logistic polynomial regression, generating new variables interactively as powers of an original covariate.
- Contingency table analysis, assuming Poisson errors and using log-linear analysis to quantify row and column effects.
- Survival analysis, using the exponential, Weibull, extreme value, and Cox (i.e., proportional hazard) models.

Of course, by choosing the advanced interface, users can always take complete control of the GLM analysis, but for many purposes the simplified interface will prove much easier to use for many routine applications.

Warning

The GLM procedure involves an iterative technique to estimate parameters from starting estimates and, in this respect, it is similar to nonlinear regression in that it will only succeed if the following conditions are satisfied.

1. The error type and link function (i.e. the model) must be chosen sensibly.
2. The data must be formatted in a specific manner depending on the error type and link function selected, as will be explained in subsequent worked examples.
3. The data must be sufficiently accurate and cover a wide enough range to allow the parameters to be estimated.
4. Error messages about failure to fit or poor parameter estimates must be interpreted sensibly, and then appropriate action taken.

Only when all these conditions are satisfied will SIMFIT be able to fit a GLM model.

8.4.2 GLM: Examples using standard formats

This section illustrates fitting generalized linear models (GLM) with various error types and link functions.

Test files and data formats

From the main SIMFIT menu choose [Statistics], [Generalized linear models], then [Comprehensive GLM options], and after selecting an error and link type view the test file provided which will be one of these.

`glm.tf1 (≡ g02gaf.tf1)`: normal error and reciprocal link
`glm.tf2 (≡ g02gbf.tf1)`: binomial error and logistic link (logistic regression)
`glm.tf3 (≡ g02gcf.tf1)`: Poisson error and log link
`glm.tf4 (≡ g02gdf.tf1)`: gamma error and reciprocal link

Here the data format for k variables, observations y and weightings s is

$$x_1, x_2, \ldots, x_k, y, s$$

except for the binomial error which has

$$x_1, x_2, \ldots, x_k, y, N, s$$

for y successes in N independent Bernoulli trials.

It is absolutely essential to have a final column of s values in the data as the number of columns is used to indicate the number of covariates. In most cases these values would be $s = 1$, but note that the weights w used are actually $w = 1/s^2$ if advanced users wish to employ weighting, e.g., using s as the reciprocal of the square root of the number of replicates for replicate weighting, except that when $s \leq 0$ the corresponding data points are suppressed. Also, observe the alternative measures of goodness of fit, such as residuals, leverages and deviances. The residuals r_i, sums of squares SSQ and deviances d_i and overall deviance depend on the error types as indicated in the examples.

GLM example 1: G02GAF, normal errors and reciprocal link

The test file `glm.tf1` contains the following data.

x	y	s
1.0	25.0	1
2.0	10.0	1
3.0	6.0	1
4.0	4.0	1
5.0	3.0	1

The next table has the results from fitting a reciprocal link with mean but no offsets to `glm.tf1`,

No. parameters = 2, Rank = 2, No. points = 5, Deg. freedom = 3

Parameter	Value	Lower95%cl	Upper95%cl	Std. error	p
Constant	-0.0238725	-0.0327174	-0.0150276	0.00277926	0.0033
B(1)	0.0638107	0.0554160	0.0722054	0.00263782	0.0002

$WSSQ = 0.387173$, $S = 0.129058$, $A = 1$

while the table of deviance residuals and leverages was as follows.

Number	Y-value	Theory	Dev-resid	Leverage
1	25.0	25.0387	-0.038665	0.995407
2	10.0	9.63865	0.361348	0.457746
3	6.0	5.96802	0.031977	0.268103
4	4.0	4.32207	-0.322074	0.166606
5	3.0	3.38775	-0.387751	0.112138

Note that the scale factor ($S = \sigma^2$) can be input or estimated using the residual sum of squares SSQ defined as follows

$$\text{For normal errors: } d_i = y_i - \hat{\mu}_i$$
$$\text{Deviance residuals: } r_i = d_i$$
$$SSQ = \sum_{i=1}^{n} r_i.$$

GLM example 2: G02GBF, binomial errors with logistic link

The next table shows the results from fitting a logistic link and mean but no offsets to test file `glm.tf2` which contains the following data for covariate x, number of successes y in N Bernoulli trials, with no weighting (i.e. all $s = 1$).

x	y	N	s
1.0	19	516	1
0.0	29	560	1
-1.0	24	293	1

No. parameters = 2, Rank = 2, No. points = 3, Deg. freedom = 1

Parameter	Value	Lower95%cl	Upper95%cl	Std. error	p
Constant	-2.86822	-4.41463	-1.32180	0.121705	0.0270
B(1)	-0.42637	-2.45654	1.60380	0.159778	0.2283 ***

Deviance = 0.0735389

Number	Y-value	Theory	Dev-resid	Leverage
1	19.0	18.4508	0.129596	0.768720
2	29.0	30.0984	-0.207027	0.422046
3	24.0	23.4508	0.117828	0.809234

The estimates are defined as follows

$$\text{For binomial errors: } d_i = 2\left\{y_i \log\left(\frac{y_i}{\hat{\mu}_i}\right) + (t_i - y_i)\log\left(\frac{t_i - y_i}{t_i - \hat{\mu}_i}\right)\right\}$$
$$\text{Deviance residuals: } r_i = \text{sign}(y_i - \hat{\mu}_i)\sqrt{d_i}$$
$$\text{Deviance } = \sum_{i=1}^{n} d_i.$$

Note that, unlike the situation with normal errors as in Example 1, the deviance residuals in the column headed as Dev-resid in the previous residuals table are not the same as the usual residuals from a regression.

GLM example 3: G02GCF, Poisson errors with a log link

This example illustrates using the choice for Poisson error and a log link to analyze a contingency table. and the test file for this option is `glm.tf3` which has columns for 8 variables x_i, then a column y for the Poisson variable, and a final column of weights $s = 1$. However, to understand the format for these data it must be pointed out that this is a representation of a 3 by 5 contingency table contained in test file `loglin.tf1`. Because there are 3 rows and 5 columns in the contingency table there will be 8 categorical variables with a 1 representing true and a 0 representing false. To clarify the situation consider the following table displaying the contingency table along with equivalent data file

Test file loglin.tf1 Test file glm.tf3

	c_1	c_2	c_3	c_4	c_5	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	y	s
r_1	141	67	114	79	39	1	0	0	1	0	0	0	0	141	1
r_2	131	66	143	72	35	1	0	0	0	1	0	0	0	67	1
r_3	36	14	38	28	16	1	0	0	0	0	1	0	0	114	1
						1	0	0	0	0	0	1	0	79	1
						1	0	0	0	0	0	0	1	39	1
						0	1	0	1	0	0	0	0	131	1
						0	1	0	0	1	0	0	0	66	1
						0	1	0	0	0	1	0	0	143	1
						0	1	0	0	0	0	1	0	72	1
						0	1	0	0	0	0	0	1	35	1
						0	0	1	1	0	0	0	0	36	1
						0	0	1	0	1	0	0	0	14	1
						0	0	1	0	0	1	0	0	38	1
						0	0	1	0	0	0	1	0	28	1
						0	0	1	0	0	0	0	1	16	1

Hence, because cell 1,1 indicates 141 number of times that category 1,1 occurred then row 1 of the data file will have a 0 everywhere except for $x_1 = 1$ and $x_4 = 1$ indicating row 1 and column 1 of the contingency table. In other words variables x_1, x_2, x_3 represent rows 1, 2, 3 in the contingency table, while variables x_4, x_5, x_6, x_7, x_8 represent columns 1, 2, 3, 4, 5 in the contingency table.

To summarize. If a contingency table T has r rows and c columns then the equivalent data file D will have rc rows and $r + c + 2$ columns. The value in contingency table cell T_{ij} will be the value in data cell D_{kl} with $k = (i - 1)c + j$ and $l = r + c + 1$. However, all the data cells D_{kl} will be zero for $l \leq r + c$ except for $l = i$ and $l = j + r$ which will be one.

The next tables show the results from fitting a log link and mean but no offsets to `glm.tf3`.

No. parameters = 9, Rank = 7, No. points = 15, Deg. freedom = 8

Parameter	Value	Lower95%cl	Upper95%cl	Std. error	p
Constant	2.59766	2.53813	2.65719	0.0258152	0.0000
B(1)	1.26195	1.16091	1.36299	0.0438171	0.0000
B(2)	1.27773	1.17714	1.37833	0.0436224	0.0000
B(3)	0.05798	-0.09595	0.21190	0.0667511	0.4104 ***
B(4)	1.03069	0.90365	1.15773	0.0550913	0.0000
B(5)	0.29102	0.12229	0.45976	0.0731714	0.0041
B(6)	0.98757	0.85859	1.11654	0.0559316	0.0000
B(7)	0.48798	0.33224	0.64371	0.0675352	0.0001
B(8)	-0.19960	-0.40795	0.00875	0.0903524	0.0582 *

Deviance = 9.03788, A = 1

Number	Y-value	Theory	Dev-resid	Leverage
1	141	132.993	0.68750	0.603533
2	67	63.4740	0.43857	0.513759
3	114	127.380	-1.20721	0.596285
4	79	77.2915	0.19363	0.531602
5	39	38.8616	0.02218	0.481976
6	131	135.109	-0.35531	0.608326
7	66	64.4838	0.18808	0.519638
8	143	129.406	1.17492	0.601167
9	72	78.5211	-0.74647	0.537265
10	35	39.4799	-0.72715	0.488239
11	36	39.8979	-0.62759	0.392649
12	14	19.0422	-1.21309	0.255123
13	38	38.2139	-0.03464	0.381546
14	28	23.1874	0.96754	0.282457
15	16	11.6585	1.20279	0.206435

The definitions are

$$\text{For Poisson errors: } d_i = 2\left\{y_i \log\left(\frac{y_i}{\hat{\mu}_i}\right) - (y_i - \hat{\mu}_i)\right\}$$

$$\text{Deviance residuals: } r_i = \text{sign}(y_i - \hat{\mu}_i)\sqrt{d_i}$$

$$\text{Deviance } = \sum_{i=1}^{n} d_i,$$

but note that an error message is output to warn you that the solution is over-determined, i.e., the parameters and standard errors are not unique.

Thus, in order to obtain unique parameter estimates, it is necessary to impose constraints so that the resulting constrained system is of full rank. Let the singular value decomposition (SVD) P^* be represented, as in G02GKF, by

$$P^* = \begin{pmatrix} D^{-1}P_1^T \\ P_0^T \end{pmatrix},$$

and suppose that there are m parameters and the rank is r, so that there need to be $n_c = m - r$ constraints, for example, in a m by n_c matrix C where

$$C^T\beta = 0.$$

Then the constrained estimates $\hat{\beta}_c$ are given in terms of the SVD parameters $\hat{\beta}_{svd}$ by

$$\begin{aligned}\hat{\beta}_c &= A\hat{\beta}_{svd} \\ &= (I - P_0(C^T P_0)^{-1}C^T)\hat{\beta}_{svd},\end{aligned}$$

while the variance-covariance matrix V is given by

$$V = AP_1D^{-2}P_1^T A^T,$$

provided that $(C^T P_0^{-1})$ exists.

This approach is commonly used in log-linear analysis of contingency tables, but it can be tedious to first fit the overdetermined Poisson GLM model then apply a matrix of constraints as just described. For this reason SimFiT provides an automatic procedure to calculate the dummy indicator matrix from the contingency table then fit a log-linear model and apply the further constraints that the sum of row effects and sum of column effects are zero.

This simplified GLM log-linear analysis of contingency tables is available from the SIMFIT main menu [Statistics] option using either the [Standard statistical tests] sub-menu or the [Generalized linear models] options.

For instance, the next table illustrates how this is done with `loglin.tf1` using the GLM log-linear contingency table analysis procedure to read in a contingency table, fit a Poisson model, then apply the correction to apply the equations of constraint

$$\sum_{i=1}^{ncol} \text{Column parameter}_i = 0$$

$$\sum_{i=1}^{nrow} \text{Row parameter}_i = 0$$

to obtain well-defined parameter estimates.

No. rows = 3, No. columns = 5
Deviance (D) = 9.03788E+00, Deg. freedom = 8
$P(\chi^2 \geq D)$ = 0.3391

Parameter	Value	Lower95%cl	Upper95%cl	Std. error	p	
Constant	3.98308	0.0395833	3.89180	4.07435	0.0000	
Row 1	0.39606	0.0458291	0.29038	0.50175	0.0000	
Row 2	0.41185	0.0456995	0.30646	0.51723	0.0000	
Row 3	-0.80791	0.0621905	-0.95132	-0.66450	0.0000	
Col 1	0.51116	0.0561557	0.38166	0.64065	0.0000	
Col 2	-0.22851	0.0727114	-0.39618	-0.06084	0.0137	*
Col 3	0.46804	0.0569148	0.33679	0.59933	0.0000	
Col 4	-0.03156	0.0675080	-0.18723	0.12412	0.6527	***
Col 5	-0.71913	0.0887225	-0.92373	-0.51454	0.0000	

Data	Model	Delta	Dev-resid	Leverage
141	132.9931	8.0069	0.6875	0.6035
67	63.4740	3.5260	0.4386	0.5138
114	127.3798	-13.3798	-1.2072	0.5963
79	77.2915	1.7085	0.1936	0.5316
39	38.8616	0.1384	0.0222	0.4820
131	135.1089	-4.1089	-0.3553	0.6083
66	64.4838	1.5162	0.1881	0.5196
143	129.4063	13.5937	1.1749	0.6012
72	78.5211	-6.5211	-0.7465	0.5373
35	39.4799	-4.4799	-0.7271	0.4882
36	39.8979	-3.8979	-0.6276	0.3926
14	19.0422	-5.0422	-1.2131	0.2551
38	38.2139	-0.2139	-0.0346	0.3815
28	23.1874	4.8126	0.9675	0.2825
16	11.6585	4.3415	1.2028	0.2064

GLM example 4: G02GDF, gamma errors with a reciprocal link

The next tables show the results from fitting a reciprocal link and mean but no offsets to `glm.tf4`.

No. parameters = 2, Rank = 2, No. points = 10, Deg. freedom = 8

Parameter	Value	Lower95%cl	Upper95%cl	Std. error	p
Constant	1.44085	-0.08812	2.96981	0.663037	0.0615 *
B(1)	-1.28653	-2.82436	0.25131	0.666882	0.0898 *

Adjusted Deviance = 35.0344, S = 1.07418, A = 1

Number	Y-value	Theory	Dev-resid	Leverage
1	1.00	6.48000	-1.39085	0.2
2	0.30	6.48000	-1.92278	0.2
3	10.5	6.48000	0.52365	0.2
4	9.70	6.48000	0.43179	0.2
5	10.9	6.48000	0.56784	0.2
6	0.62	0.69404	-0.11071	0.2
7	0.12	0.69404	-1.32870	0.2
8	0.09	0.69404	-1.48152	0.2
9	0.50	0.69404	-0.31063	0.2
10	2.14	0.69404	1.36648	0.2

Note that with gamma errors, the scale factor (ν^{-1}) can be input or estimated using the degrees of freedom, k, and

$$\hat{\nu}^{-1} = \sum_{i=1}^{n} \frac{[(y_i - \hat{\mu}_i]/\hat{\mu}_i]^2}{N-k}$$

$$\text{For gamma errors: } d_i = 2\left\{\log(\hat{\mu}_i) + \left(\frac{y_i}{\hat{\mu}_i}\right)\right\}$$

$$\text{Deviance residuals: } r_i = \frac{3(y_i^{\frac{1}{3}} - \hat{\mu}_i{}^{\frac{1}{3}})}{\hat{\mu}_i{}^{\frac{1}{3}}}$$

$$\text{Deviance: } = \sum_{i=1}^{n} d_i$$

8.4.3 GLM: Loglinear contingency table analysis

In addition to chi-square and Fisher exact analysis of contingency tables, using generalized linear models (GLM) to perform loglinear analysis is often preferred as it provides more insight into the structure of the table, and can be extended to contingency tables with more than two dimensions.

From the main SIMFIT menu select [Statistics], [Generalized linear models], [Contingency table analysis], then observe the format of the default data file `loglin.tf1` which contains the following contingency table.

	c_1	c_2	c_3	c_4	c_5
r_1	141	67	114	79	39
r_2	131	66	143	72	35
r_3	36	14	38	28	16

When these data are analyzed, SIMFIT creates a temporary data file formatted for GLM analysis using Poisson error with a log link then applies the constraint that the sum of row coefficients and also the sum of column coefficients add to zero to output the next tables of results.

No. rows = 3, No. columns = 5
Deviance (D) = 9.03788E+00, Deg. freedom = 8
$P(\chi^2 \geq D)$ = 0.3391

Parameter	Value	Lower95%cl	Upper95%cl	Std. error	p
Constant	3.98308	0.0395833	3.89180	4.07435	0.0000
Row 1	0.39606	0.0458291	0.29038	0.50175	0.0000
Row 2	0.41185	0.0456995	0.30646	0.51723	0.0000
Row 3	-0.80791	0.0621905	-0.95132	-0.66450	0.0000
Col 1	0.51116	0.0561557	0.38166	0.64065	0.0000
Col 2	-0.22851	0.0727114	-0.39618	-0.06084	0.0137 *
Col 3	0.46804	0.0569148	0.33679	0.59933	0.0000
Col 4	-0.03156	0.0675080	-0.18723	0.12412	0.6527 ***
Col 5	-0.71913	0.0887225	-0.92373	-0.51454	0.0000

Data	Model	Delta	Dev-resid	Leverage
141	132.9931	8.0069	0.6875	0.6035
67	63.4740	3.5260	0.4386	0.5138
114	127.3798	-13.3798	-1.2072	0.5963
79	77.2915	1.7085	0.1936	0.5316
39	38.8616	0.1384	0.0222	0.4820
131	135.1089	-4.1089	-0.3553	0.6083
66	64.4838	1.5162	0.1881	0.5196
143	129.4063	13.5937	1.1749	0.6012
72	78.5211	-6.5211	-0.7465	0.5373
35	39.4799	-4.4799	-0.7271	0.4882
36	39.8979	-3.8979	-0.6276	0.3926
14	19.0422	-5.0422	-1.2131	0.2551
38	38.2139	-0.2139	-0.0346	0.3815
28	23.1874	4.8126	0.9675	0.2825
16	11.6585	4.3415	1.2028	0.2064

Theory

A contingency table is an array of nonnegative frequencies with n rows and m columns, such as this table contained in SimFIT test file `chisqd.tf4`, for 15 observations carried out on two populations to test for equal probabilities of success.

	Success	Failure	
Sample 1	3	3	6
Sample 2	7	2	9
	10	5	15

Here, the cell frequencies f_{ij} are $(3, 3, 7, 2)$, the sum of row frequencies known as row marginals are $(6, 9)$, the sum of column frequencies known as column marginals are $(10, 5)$, and obviously the row and column marginals must separately both add up to the total number of frequencies (15). The null hypothesis is usually to test for homogeneity or independence, which is the condition that the f_{ij} only depend on row i and column j, and there are no additional influences affecting frequencies in special cells.

To be precise, in the general case there will be frequencies f_{ij} where $i = 1, 2, \ldots, n$, and $j = 1, 2, \ldots, m$, and it is wished to test for homogeneity, i.e. independence, or no association between the variables, which can be stated as the null hypothesis

$$H_0 : \mu_{ij} = \mu_{i+}\mu_{+j}, \text{ for } i = 1, 2, \ldots, n, \text{ and } j = 1, 2, \ldots, m$$

where each cell probability μ_{ij} is completely determined by the corresponding row marginal μ_{i+}, and the column marginal μ_{+j} probabilities.

To do this, SimFIT defines dummy indicator variables for the rows and columns, then fits a generalized linear model assuming a Poisson error distribution and log link, but imposing the constraints that the sum of row coefficients is zero and the sum of column coefficients is zero, to avoid fitting an over-determined model, and to be consistent with an assumed loglinear model.

The advantage of this approach is that the deviance, predicted frequencies, deviance residuals, and leverages can be calculated for the model

$$\log(\mu_{ij}) = \theta + \alpha_i + \beta_j,$$

where μ_{ij} are the expected cell frequencies expressed as functions of an overall mean θ, row coefficients α_i, and column coefficients β_j. The row and column coefficients reflect the main effects of the categories, according to the above model, where

$$\sum_{i=1}^{n} \alpha_i = \sum_{j=1}^{m} \beta_j = 0$$

and the deviance, which is a likelihood ratio test statistic, can be used to test the justification for a mixed term γ_{ij} in the saturated model

$$\log(\mu_{ij}) = \theta + \alpha_i + \beta_j + \gamma_{ij},$$

which fits exactly, i.e., with zero deviance.

SimFIT performs a chi-square test on the deviance to test the null hypotheses of homogeneity, which is the same as testing that all γ_{ij} are zero, the effect of individual cells can be assessed from the leverages, and various deviance residuals plots can be done to estimate goodness of fit of the assumed loglinear model.

Clearly, the chi-square test for data in test file `loglin.tf1` presented in the previous table does not support rejection of the null hypothesis of homogeneity.

8.4.4 GLM: Logistic regression

Logistic regression is widely used to model experiments with only one of two outcomes such as success or failure which, unlike the simple method for analysis of binomial proportions, depend on the values of k covariates $x_1, x_2, \ldots, x_k$, where $k \geq 1$.

Example 1: Alcohol and congenital abnormalities

From the main SIMFIT menu choose [Statistics], [Generalized linear models], then [Logistic regression], and examine the default test file `logistic.tf3` containing the following data.

x	y	N	s
0.0	48	17066	1
0.5	38	14464	1
1.5	5	788	1
4.0	1	126	1
7.0	1	37	1

These data were taken from a study of the effects of consuming alcoholic drinks on congenital abnormalities noted in infants after birth.

1. Column 1: x, alcoholic drinks consumed per day by mother
2. Column 2: y, infants born with abnormalities
3. Column 3: N, sample size
4. Column 4: s, weighting factors (s = 1 indicates unweighted analysis)

Logistic regression by was used to fit the GLM model

$$\log[y/(N-y)] \approx \beta_0 + \beta_1 x$$

which yielded the following parameter estimates, residuals, then observed and estimated frequencies.

Number of parameters = 2, Rank = 2, Number of points = 5, Degrees of freedom = 3

Parameter	Value	Lower95%cl	Upper95%cl	Std. error	p	$exp(\beta_1)$
Constant	-5.95840	-6.32583	-5.59097	0.115454	0.0000	
β_1	0.31927	-0.08038	0.71889	0.125574	0.0845 *	1.37611

Deviance = 1.96760

Number	Y-value	Theory	Dev-resid.	Leverage
1	48	43.9856	0.597220	0.584800
2	38	43.7119	-0.885127	0.476721
3	5	3.27338	0.886968	0.097194
4	1	1.15684	-0.149976	0.246568
5	1	0.87238	0.135174	0.594717

Observed	Estimated
0.0028	0.0026
0.0026	0.0030
0.0064	0.0041
0.0079	0.0092
0.0270	0.0236

Example 2: The symmetrical case with one variable

Logistic regression is frequently used to model the variation in binomial probability p as a simple linear function of a variable x often without realizing that, because of the necessary symmetry of the logistic function, this will usually lead to a biased fit.

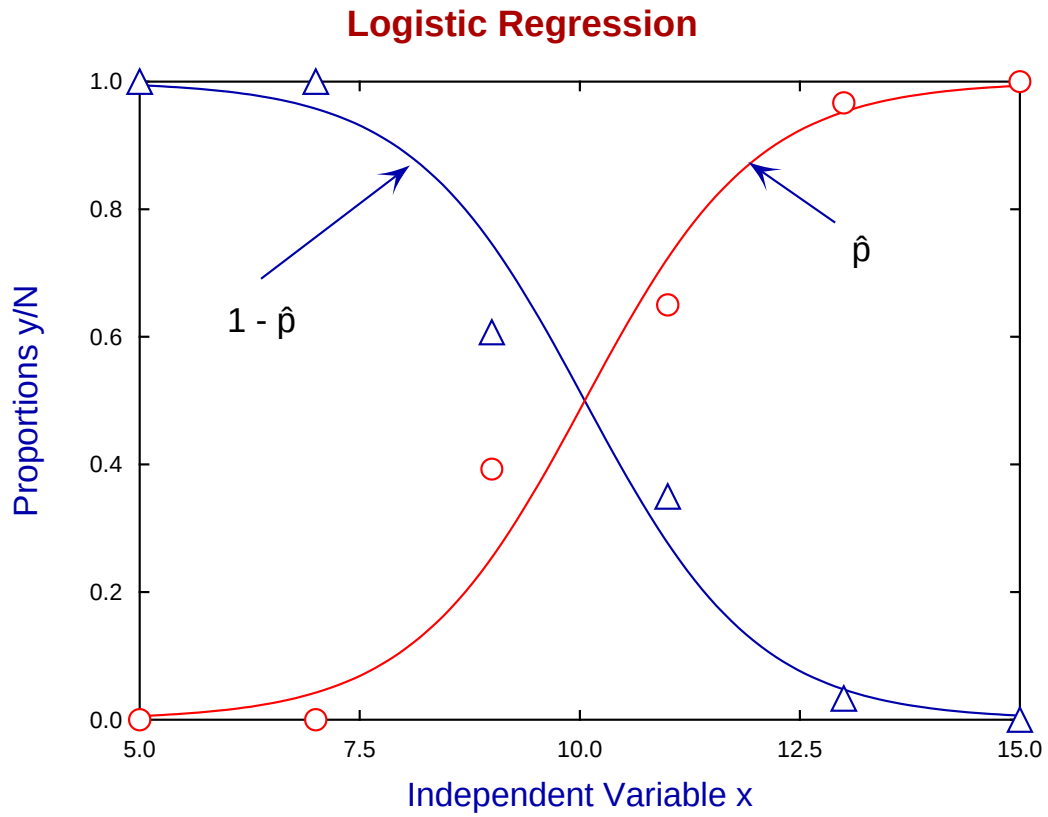
For instance, consider the data file `logistic.tf4` below

5	0	39	1
7	0	30	1
9	11	28	1
11	26	40	1
13	29	30	1
15	20	20	1

and its complement `logistic.tf5`

5	39	39	1
7	30	30	1
9	17	28	1
11	14	40	1
13	1	30	1
15	0	20	1

simulated by SimFIT using a random choice from an integer uniform distribution for N $(20 \leq N \leq 40)$ followed by a random choice from a binomial distribution for y given N and $p(x)$ which were then fitted as indicated in the next graph.



Exact parameters were $\beta_0 = -10, \beta_1 = 1$ for $p(x)$ and $\beta_0 = 10, \beta_1 = -1$ for $1 - p(x)$ and the best fit parameters are in the next tables.

Number of parameters = 2, Rank = 2, Number of points = 6, Degrees of freedom = 4

Parameter	Value	Lower95%cl	Upper95%cl	Std. error	p	$exp(\beta_1)$
Constant	-10.2573	-14.5246	-5.98991	1.53699	0.0026	
β_1	1.01996	0.60551	1.43440	0.14927	0.0024	2.77307

Deviance = 7.07433

Number of parameters = 2, Rank = 2, Number of points = 6, Degrees of freedom = 4

Parameter	Value	Lower95%cl	Upper95%cl	Std. error	p	$exp(\beta_1)$
Constant	10.2573	5.98391	14.5307	1.53915	0.0026	
β_1	-1.01996	-1.43498	-0.60494	0.14948	0.0024	0.360610

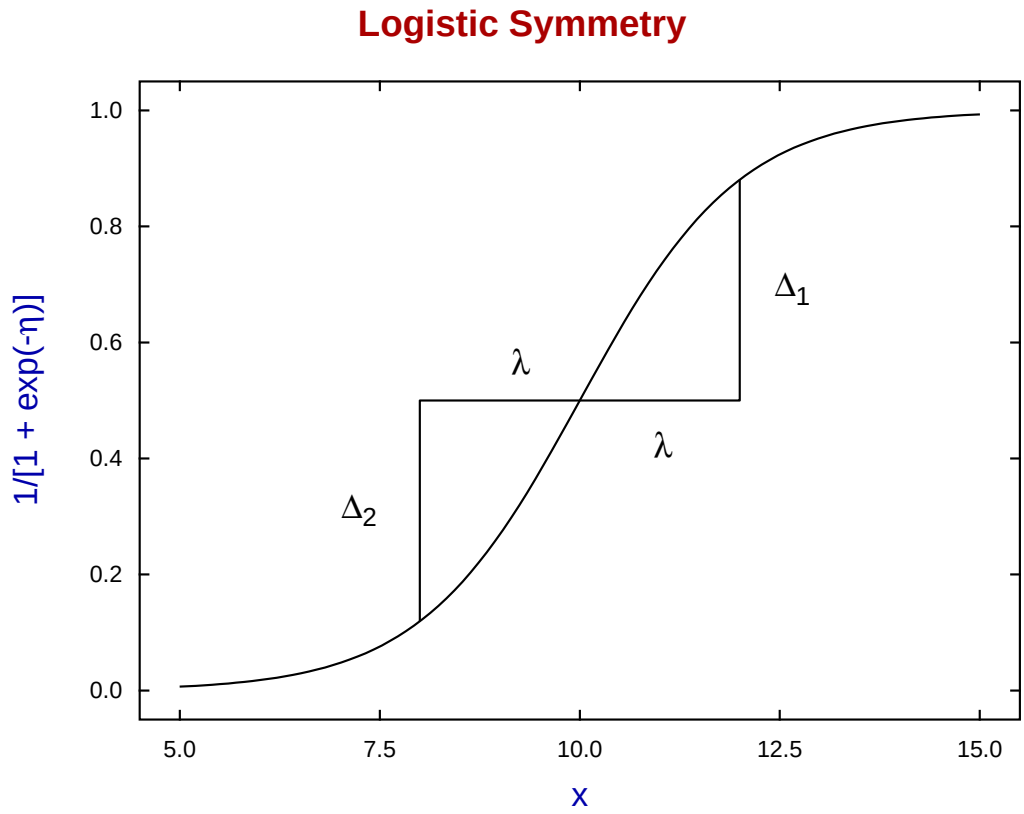
Deviance = 7.07433

Due to the symmetry of the logistic curves for p and $1-p$ about their midpoints $x_{1/2}$

$$p = \frac{1}{1+\exp(-\eta)} = 1 - p = \frac{1}{1+\exp(\eta)} = \frac{1}{2}$$

so that the mid point requires $\eta = 0$, that is $x_{1/2} = -\beta_0/\beta_1$.

Moving a horizontal distance λ to either side of the midpoint generates two vertical distances Δ_1 and Δ_2 which are equal as shown in this next graph.



This is because

$$\Delta_1 = \frac{1}{1+\exp(-\beta_1\lambda)} - \frac{1}{2} = \Delta_2 = \frac{1}{2} - \frac{1}{1+\exp(\beta_1\lambda)}.$$

Theory

In a situation where the probability of success is a fixed constant p and Y is the number of successes in N trials ($N \geq 1$), then the probability the Y equals any specific value y where $0 \leq y \leq N$ is

$$P(Y = y) = \binom{N}{y} p^y (1-p)^{N-y}$$

and p can be estimated as $\hat{p}$ where the estimate, expectation and variance are

$$\begin{aligned} \hat{p} &= y/N \\ E(y) &= Np \\ V(y) &= Np(1-p). \end{aligned}$$

When the binomial parameter is a function of some variables $x_1, x_2, \ldots, x_k$ then a functional relationship must be proposed to model $p(x)$ and this must be fitted to estimate parameters accounting for the variation in p. It is usual to do this by fitting a generalized linear model (GLM) with assumed binomial error and logistic link, but the reason for this model is not because it is the correct model but because of the following fact. In the simple case of one variable x the log odds ratio from fitting a GLM model with y_1 at x and y_2 at $x+1$ can then be expressed as

$$\begin{aligned} \log\left(\frac{y_1}{N-y_1}\right) &\approx \beta_0 + \beta_1 x \\ \log\left(\frac{y_2}{N-y_2}\right) &\approx \beta_0 + \beta_1 (x+1) \end{aligned}$$

and hence the odds ratio can be estimated as

$$\frac{\hat{p}_2/(1-\hat{p}_2)}{\hat{p}_1/(1-\hat{p}_1)} \approx \exp(\beta_1).$$

From this we could conclude that an increase of one alcoholic drink per day can be estimated to change the odds ratio for congenital malformation by about 1.376.

It is this seemingly easy method for interpreting the parameter estimates from logistic regression that has been responsible for the widespread and often uncritical adoption of this technique and its extension into areas such as

- Including multiple variables
- Analyzing cases with categorical variables.

So, in order to confirm goodness of fit, SimFIT outputs the deviance and deviance residuals defined as follows

$$\begin{aligned} \text{For binomial errors: } d_i &= 2\left\{y_i \log\left(\frac{y_i}{\hat{\mu}_i}\right) + (N_i - y_i)\log\left(\frac{N_i - y_i}{N_i - \hat{\mu}_i}\right)\right\} \\ \text{Deviance residuals: } r_i &= \text{sign}(y_i - \hat{\mu}_i)\sqrt{d_i} \\ \text{Deviance } &= \sum_{i=1}^{n} d_i. \end{aligned}$$

where there are n observations and $\hat{\mu}_i = N_i \hat{p}_i$, and these should always be considered before accepting a fit.

8.4.5 GLM: Binary logistic regression

Binary logistic regression is widely used to model experiments with only one of two outcomes such as success or failure which, unlike the simple method for analysis of binomial proportions, depend on the values of k covariates $x_1, x_2, \ldots, x_k$, where $k \geq 1$.

Example 1: Fitting a binary logistic model

From the main SIMFIT menu choose [Statistics], [Generalized linear models], then [Binary logistic regression] (with no strata), then examine the default test file `logistic.tf1` containing the following data.

x_1	x_2	y	N	s
3.70	0.825	1	1	1
3.50	1.090	1	1	1
0.75	1.500	1	1	1
1.25	2.500	1	1	1
0.80	3.200	1	1	1
0.70	3.500	1	1	1
0.60	0.750	0	1	1
1.10	1.700	0	1	1
0.90	0.750	0	1	1
0.90	0.450	0	1	1
0.80	0.570	0	1	1
0.55	2.750	0	1	1
0.60	3.000	0	1	1
1.40	2.330	1	1	1
0.75	3.750	1	1	1
2.34	1.640	1	1	1
3.20	1.600	1	1	1
0.85	1.415	1	1	1
1.70	1.060	0	1	1
1.80	1.800	1	1	1
0.40	2.000	0	1	1
0.95	1.360	0	1	1
1.35	1.350	0	1	1
1.50	1.360	0	1	1
1.60	1.780	1	1	1
0.60	1.500	0	1	1
1.80	1.500	1	1	1
0.95	1.900	0	1	1
1.90	0.950	1	1	1
1.60	0.400	0	1	1
2.70	0.750	1	1	1
2.35	0.030	0	1	1
1.10	1.830	0	1	1
1.10	2.200	1	1	1
1.20	2.000	1	1	1
0.80	3.330	1	1	1
0.95	1.900	0	1	1
0.75	1.900	0	1	1
1.30	1.625	1	1	1

The format of this data file will now be explained. These are vasoconstriction data from Finney D. J. (1947) Biometrika, 34, 320-34 with the following meanings.

- Column 1: x_1 (volume of air inspired)
- Column 2: x_2 (rate of air inspiration)
- Column 3: $y = 1$ (vasoconstriction), or $y = 0$ (no vasoconstriction)
- Column 4: $N = 1$ (sample size)
- Column 5: $s = 1$ (unweighted)

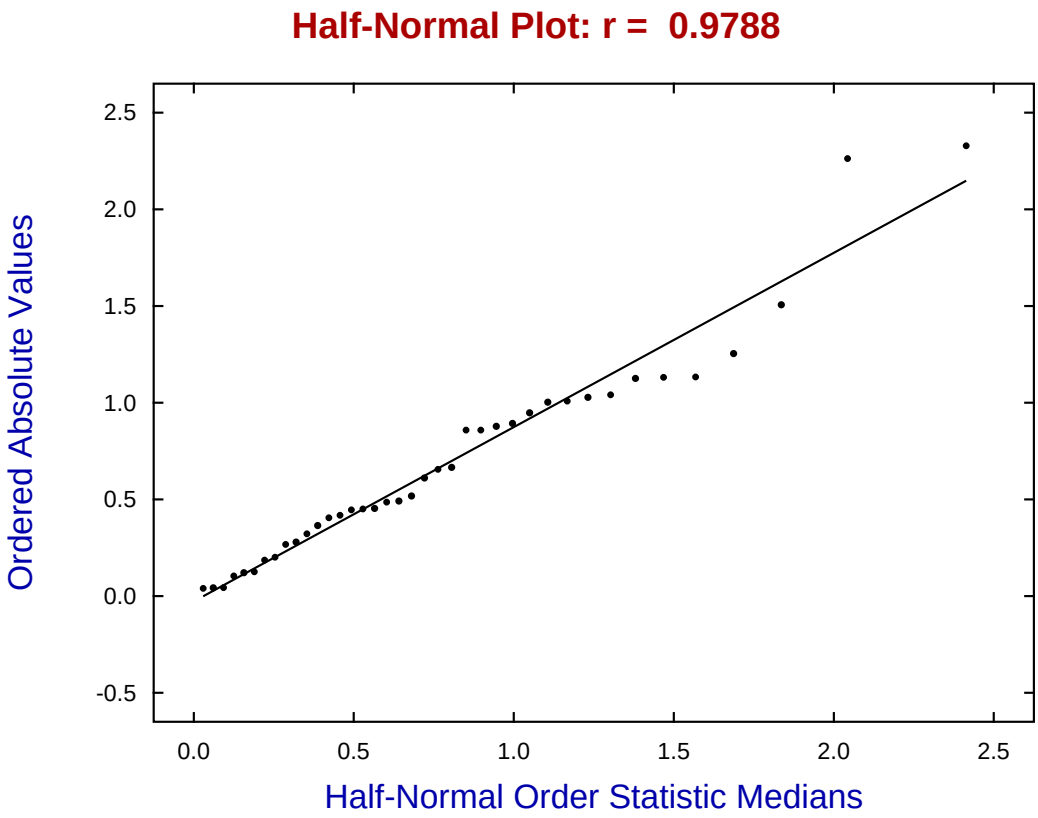
It is important to note that for binary logistic regression, y must be 1 (e.g. for success) or 0 (e.g. for failure), and N must be 1 because y is the outcome from a single Bernoulli trial, whereas for normal logistic regression, y must be in the range $0 \leq y \leq N$ with $N \geq 1$ for the number of trials resulting in y (e.g. the number of successful outcomes). The weighting factors would normally be $s = 1$ except for experienced users.

Fitting a generalized linear model with a mean, no offsets, binomial error and a logistic link leads to the following results table and half-normal residuals plot.

Number of parameters = 3, Rank = 3, Number of points = 39, Degrees of freedom = 36

Parameter	Value	Lower95%cl	Upper95%cl	Std. error	p	$\exp(\beta_i)$
Constant	-9.51999	-16.0587	-2.98131	3.22405	0.0055	
β_1	3.87719	0.986847	6.76753	1.42515	0.0100	48.2882
β_2	2.64683	0.797466	4.49619	0.91187	0.0063	14.1092

Deviance = 29.7656



Example 2: Predicting probabilities

Binary logistic regression seeks to find an approximation $\hat{p}(x)$ to a population binomial probability $p(x)$ that is not a constant probability but one that depends on covariates x as in the following model for the logodds

$$\log\left(\frac{p(x)}{1-p(x)}\right) \approx \eta(x)$$
$$\text{where } \eta(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k$$

and where the approximation results from fitting a generalized linear model (GLM) with binomial error, using a logistic link, when there are k covariates. The constant parameter β_0 in this polynomial simply estimates the logodds when all k covariates are zero, and can be included or omitted from the model.

Having estimated best-fit parameters and confirmed that the model is satisfactory it is often useful to predict what the probability would be given a set of covariates using the best-fit parameters in the next expressions

$$\hat{\eta} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \cdots + \hat{\beta}_k x_k$$
$$\hat{p} = \frac{\exp(\hat{\eta})}{1+\exp(\hat{\eta})}$$
$$= \frac{1}{1+\exp(-\hat{\eta})}.$$

So, after fitting has been completed, SimFIT offers the possibility to do this either by inputting a set of covariates from the terminal or from a data file containing a matrix of covariates. Here, for instance, are the data contained in the test file `logistic.tf2`

x_1	x_2
0.4	1.0
0.6	1.0
0.8	1.0
1.0	1.0
1.0	0.8
1.0	0.6
1.0	0.4

which lead to these predictions after evaluation using the best-fit model from fitting `logistic.tf1`.

File: logistic.tf2
Data: Covariates to evaluate p after fitting logistic.tf1
$Y(x)$ evaluated for x_1 to x_2
Model includes a constant term
Binomial $N = 1$

y predicted	Probability estimated	Range of covariates
y_1 = 4.85785E-03	Binomial p = 0.004858	x = 0.4, ..., 1.0
y_2 = 1.04893E-02	Binomial p = 0.010489	x = 0.6, ..., 1.0
y_3 = 2.25015E-02	Binomial p = 0.022502	x = 0.8, ..., 1.0
y_4 = 4.76080E-02	Binomial p = 0.047608	x = 1.0, ..., 1.0
y_5 = 2.85997E-02	Binomial p = 0.028600	x = 1.0, ..., 0.8
y_6 = 1.70450E-02	Binomial p = 0.017045	x = 1.0, ..., 0.6
y_7 = 1.01099E-02	Binomial p = 0.010110	x = 1.0, ..., 0.4

As binary logistic regression is so widely used, often uncritically and without justification, to predict probabilities given covariates it is as well to consider the basic principles behind this method which will now be done.

Theory

The ideas behind binary logistic regression will be explained under several headings, namely

1. definitions;
2. one quantitative variable;
3. several quantitative variables;
4. categorical variables;
5. fitting technique; then
6. conclusion.

1. Definitions

A random integer variable Y can be formulated as taking a value depending on the result of an experiment with only one of two possible outcomes, such as heads/tails in coin tossing, death/survival following a serious illness, positive/negative of a value with respect to a baseline, etc. Arbitrarily calling one outcome a success and the other a failure we can sometimes define Y as taking two possible values, 1 or 0, depending on a probability p where $0 \leq p \leq 1$ that is

$$\begin{aligned} \text{Probability}(y = 1) &= p \\ \text{Probability}(y = 0) &= 1 - p. \end{aligned}$$

When the certain and impossible outcomes are excluded (i.e. $0 < p < 1$) the Odds can be defined as the ratio of success to failure as can the Log Odds, its natural logarithm, that is

$$\begin{aligned} \text{Odds} &= \frac{p}{1-p} \\ \text{Log Odds} &= \log\left(\frac{p}{1-p}\right). \end{aligned}$$

Given one trial with probability p_1 where $0 < p_1 < 1$ and one with probability p_2 where $0 < p_2 < 1$ the Odds Ratio and Log Odds Ratio are then given by

$$\begin{aligned} \text{Odds Ratio} &= \frac{p_2/(1-p_2)}{p_1/(1-p_1)} \\ \text{Log Odds Ratio} &= \log\left(\frac{p_2/(1-p_2)}{p_1/(1-p_1)}\right). \end{aligned}$$

At this point it should be emphasized that exponentials and logarithms to base e are used in theoretical developments and computational implementation, but many users prefer to present results using logarithms to base 10 in order to immediately clarify changes in orders of magnitude as powers of 10.

Of course such probabilities are never known exactly but must be estimated by sampling. In the case of N successive Bernoulli trials which are independent with identical probability it is usual to define a binomial variable W as

$$W = y_1 + y_2 + \cdots + y_N$$

so that the probability that $W = w$ where $0 \leq w \leq N$ is

$$P(W = w) = \binom{N}{w} p^w (1-p)^{N-w}$$

with expectation and variance given by

$$E(W) = Np$$
$$V(W) = Np(1-p).$$

From such a series of trials an estimate for the binomial parameter $\hat{p}$ is easily seen to be

$$\hat{p} = \frac{w}{N}$$

and SIMFIT provides dedicated analysis of proportions routines to calculate $\hat{p}$ with unsymmetrical confidence limits and plot these as a function of a parameter such as time, which only serves to order the observations for plotting and does not enter into the calculations.

Logistic regression using GLM is an extension of this subject to the case where additional variables x affect the probabilities under the assumption that each set of additional variables alters the binomial distribution distribution, i.e. $p = p(x)$, while binary logistic regression is just the special case where $N = 1$.

2. One quantitative variable

The GLM technique is used to adjust the two parameters β_0 and β_1 until the best-fit values are located to satisfy the approximation

$$\log\left(\frac{\hat{p}(x)}{1-\hat{p}(x)}\right) \approx \hat{\beta}_0 + \hat{\beta}_1 x$$

according to the maximum likelihood criterion.

A special case is where the continuous variable is used at just two levels differing by one unit, say x and $x+1$, for then we have that, since

$$\begin{aligned}\text{Odds} &= \frac{p}{1-p}\\ &= \exp(\eta)\end{aligned}$$

then for the two levels x and $x+1$

$$\begin{aligned}\text{Odds Ratio} &= \frac{\text{Odds}(x+1)}{\text{Odds}(x)}\\ &= \frac{\exp(\hat{\beta}_0 + \hat{\beta}_1(x+1))}{\exp(\hat{\beta}_0 + \hat{\beta}_1 x)}\\ &= \exp(\hat{\beta}_1)\end{aligned}$$

so that the Odds multiply by the factor $\exp(\hat{\beta}_1)$ for every one unit increase in the variable x or, alternatively,

$$\hat{\beta}_1 = \text{Log Odds Ratio}.$$

3. Several quantitative variables

Now, for k variables we have

$$\eta = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k$$

which becomes difficult to fit as k increases, especially if the variables are not expressed in units so that the values are of similar size. Further, the arguments about interpreting the estimated parameters in terms of Log Odds Ratios, imply that only one variable is increased at a time with the others remaining fixed.

4. Categorical variables

Frequently the covariates are qualitative variables which can be included in the model by defining appropriate dummy indicator variables. For instance, suppose a factor has m levels, then we can define m dummy indicator variables $x_1, x_2, \ldots, x_m$ as in the next table.

Level	x_1	x_2	x_3	...	x_m
1	1	0	0	...	0
2	0	1	0	...	0
3	0	0	1	...	0
...	...	...	...	...	...
m	0	0	0	...	1

The data file would be set up as if to estimate all m parameters for the m factor levels but because only $m-1$ of the dummy indicator variables are independent, one of them would have to be suppressed if a constant were to be fitted, to avoid aliasing, i.e., the model would be overdetermined and the parameters could not be estimated uniquely. Suppose, for instance, that the model to be fitted was for a factor with three levels, i.e.,

$$\log\left\{\frac{p(x)}{1-p(x)}\right\} = a_0 + a_1x_1 + a_2x_2 + a_3x_3$$

but with any one of the x_i suppressed, x_1 for instance, since

$$x_{1i} + x_{2i} + x_{3i} = 1$$

for every i.

Then the estimated parameters could be interpreted as log odds ratios for the factor levels with respect to level 1, the suppressed reference level. This is because for probability estimates $\hat{p}_1$, $\hat{p}_2$ and $\hat{p}_3$ we would have the odds estimates

$$\begin{aligned}
\frac{\hat{p}_1}{1-\hat{p}_1} &= \exp(\hat{a}_0) && (x_1 = 1, x_2 = 0, x_3 = 0)\\
\frac{\hat{p}_2}{1-\hat{p}_2} &= \exp(\hat{a}_0 + \hat{a}_2) && (x_1 = 0, x_2 = 1, x_3 = 0)\\
\frac{\hat{p}_3}{1-\hat{p}_3} &= \exp(\hat{a}_0 + \hat{a}_3) && (x_1 = 0, x_2 = 0, x_3 = 1)
\end{aligned}$$

and estimates for the corresponding log odds ratios involving only the corresponding estimated coefficients

$$\begin{aligned}
\log\left\{\frac{\hat{p}_2/(1-\hat{p}_2)}{\hat{p}_1/(1-\hat{p}_1)}\right\} &= \hat{a}_2\\
\log\left\{\frac{\hat{p}_3/(1-\hat{p}_3)}{\hat{p}_1/(1-\hat{p}_1)}\right\} &= \hat{a}_3.
\end{aligned}$$

5. Fitting technique

The first thing to note about binary logistic regression is that we cannot fit the model

$$\log\left(\frac{y}{1-y}\right) = \beta_0 + \beta_1x_1 + \beta_2x_2 + \cdots + \beta_kx^k$$

directly as all the y values are 0 or 1. Instead starting estimates are generated and an iterative procedure is used to find the maximum likelihood solution point and estimate the deviance and deviance residuals. In the

event that the data matrix is not of full rank this will be reported and the singular value decomposition will be used, so parameter estimates will still be generated but will not then be unique.

Note that, for standard logistic regression where w_i is the i'th binomial variable for m samples of size N_i, and not either 0 or 1 with a sample size of 1 as with binary logistic regression, the deviance is

$$\sum_{i=1}^{m} \text{dev}(w_i, \hat{\mu}_i) = 2 \sum_{i=1}^{m} \left\{ w_i \log\left(\frac{w_i}{\hat{\mu}_i}\right) + (N_i - w_i) \log\left(\frac{N_i - w_i}{N_i - \hat{\mu}_i}\right) \right\}$$

and the deviance residuals r_i are

$$r_i = \text{sign}(w_i - \hat{\mu}_i)\sqrt{\text{dev}(w_i, \hat{\mu}_i)}.$$

Of course these expressions are corrected for the extreme cases $w_i = 0$ or $w_i = N_i$ which will happen from time to time with standard logistic regression, but will happen all the time with binary logistic regression.

6. Conclusions

Binary logistic regression is widely used to analyze large data sets, sometimes even containing mixtures of qualitative and quantitative variables, and often in order to estimate Log Odds Ratios. It is incumbent upon users that any conclusions drawn about predicting probabilities are justified by taking account of the following suggestions.

- Add a constant term to the regression unless it is clear that $p = 0.5$ when all the covariates are zero.
- Scale all the variables to similar orders of magnitude prior to regression.
- Take care about the need with categorical variables to suppress a variable if a constant is fitted.
- Check the deviance, deviance residuals, and leverages to make sure the model gives a sensible fit.
- Do not ignore warnings if the rank is less than full or iteration has not converged.
- Only using parameters to estimate log odds ratio if the number of variables is small and that a unit change in a variable makes sense.
- Be careful not to confuse exponentials and logarithms to base e with powers of ten and logarithms to base 10.

8.5 Nonlinear regression: simple

Tutorials and worked examples for simulation, curve fitting, statistical analysis, and plotting.
https://simfit.uk
https://simfit.org.uk
https://simfit.silverfrost.com

8.5.1 Fitting Michaelis-Menten enzyme kinetic models

Michaelis–Menten models are used to analyze quasi steady state data when the enzymes concerned do not exhibit substrate inhibition, substrate activation, or any other complicating features such as allosterism. The one site version can be extended to cover the case of an enzyme with several kinetically differing but independent sites, or mixtures of isoenzymes.

Example 1: Substrate varied mode

From the main SIMFIT menu select [A/Z], open program **mmfit**, select the substrate-varied option, and view the default test file `mmfit.tf4` which has the following data.

S	v	$se(v)$
0.21759	0.20273	0.0054324
0.21759	0.20050	0.0054324
0.21759	0.19241	0.0054324
0.39440	0.31925	0.015018
0.39440	0.34123	0.015018
0.39440	0.31252	0.015018
0.71490	0.50336	0.011163
0.71490	0.48104	0.011163
0.71490	0.49241	0.011163
1.2958	0.67103	0.018464
1.2958	0.70535	0.018464
1.2958	0.67639	0.018464
2.3488	0.90847	0.015994
2.3488	0.93885	0.015994
2.3488	0.91501	0.015994
4.2575	1.1107	0.021537
4.2575	1.1439	0.021537
4.2575	1.1035	0.021537
7.7172	1.3639	0.048544
7.7172	1.2947	0.048544
7.7172	1.3882	0.048544
13.988	1.6565	0.042217
13.988	1.5894	0.042217
13.988	1.5785	0.042217
25.355	1.6468	0.078963
25.355	1.7954	0.078963
25.355	1.6748	0.078963
45.959	1.8712	0.029314
45.959	1.8568	0.029314
45.959	1.8148	0.029314

The columns contain data in the following format.

Column 1: S, the non–negative substrate concentration which must be in non-decreasing order.
Column 2: v, the non–negative initial rate measured for the concentration in column 1.
Column 3: se, the positive sample standard deviation of the replicate rate measurements.
Note that column 3 can be omitted or set to 1 if unweighted regression is required.

To illustrate the functionality of the SIMFIT program **mmfit** we shall fit a one site model followed by a two site model (or mixture of two isoenzymes) and see if any improvement in fit can be supported by statistical analysis. The two models are as follows.

$$f_1(S) = \frac{V_{max}S}{K_m + S}$$
$$f_2(S) = \frac{V_{max_1}S}{K_{m_1} + S} + \frac{V_{max_2}S}{K_{m_2} + S}$$

To fit these two models choose to start fitting at order 1 and end fitting at order 2, using the further default settings, to obtain the following results tables.

Table 1: For best-fit 1:1 Michaelis-Menten function f_1

Number	Parameter	Value	Std. Error	Lower95%cl	Upper95%cl	p
1	V_{max}	1.7861	0.040866	1.7024	1.8698	0.0000
2	K_m	1.9734	0.097463	1.7738	2.1731	0.0000

Parameter correlation matrix for model f_1

1	
0.8185	1

Table 2: For best-fit 2:2 Michaelis-Menten function f_2

Number	Parameter	Value	Std. Error	Lower95%cl	Upper95%cl	p
1	V_{max_1}	1.0254	0.10377	0.81211	1.2387	0.0000
2	V_{max_2}	1.0290	0.13352	0.75455	1.3035	0.0000
3	K_{m_1}	9.7460	2.8652	3.8566	15.636	0.0022
4	K_{m_2}	1.0433	0.11698	0.80283	1.2837	0.0000

Predicted maximum rate (i.e. apparent V_{max}) = 2.0544
Predicted half saturation point (i.e. apparent K_m) = 3.1811

Parameter correlation matrix for model f_2

1			
-0.9568	1		
-0.8573	0.9638	1	
-0.9631	0.9810	0.9088	1

In order to determine if a significant improvement in fit has resulted we need to consider the following questions.

1. Are the parameters well-determined with both fits ?
2. Does the residuals analysis indicate satisfactory fits ?
3. Does the F test for excess variance support model f_2 in preference to f_1 ?
4. Can the best-fit curves be seen to differ when plotted against the data ?
5. Does the graphical deconvolution display convincing evidence that both components of f_2 are contributing to the overall fit ?

The results displayed in Tables 1 and 2 show that both models fit well with parameters that differ significantly from zero. Table 3 indicates that an excellent fit has resulted for model f_2 and Table 4 supports the conclusion that there is statistical evidence that model f_2 should be accepted as explaining the data better than model f_1. This is then further emphasized by the graphical displays showing the data with best-fit curves for f_1 and f_2, and the deconvolution of the f_2 fit into the two contributing components.

Table 3: Goodness of fit for model f_2

Analysis of residuals: $WSSQ$	28.293
$P(\chi^2 \geq WSSQ)$	0.3442
$R^2, cc(theory, data)^2$	0.9963
Largest Absolute relative residual	5.88%
Smallest Absolute relative residual	0.25%
Average Absolute relative residual	2.28%
Absolute relative residuals in range 0.1-0.2	0.00%
Absolute relative residuals in range 0.2-0.4	0.00%
Absolute relative residuals in range 0.4-0.8	0.00%
Absolute relative residuals > 0.8	0.00%
Number of negative residuals (m)	13
Number of positive residuals (n)	17
Number of runs observed (r)	22
$P(\text{runs} \leq r : \text{ given } m \text{ and } n)$	0.9957
5% lower tail point	10
1% lower tail point	9
$P(\text{runs} \leq r : \text{ given } m \text{ plus } n)$	0.9959
$P(\text{signs} \leq \text{ least number observed})$	0.5847
Durbin-Watson test statistic	2.5008 >2.5, -ve serial correlation?
Shapiro-Wilks W statistic	0.9678
Significance level of W	0.4806
Akaike AIC (Schwarz SC) stats	6.2425 (11.847)
Verdict on goodness of fit: incredible	

Table 4: F test results for model f_2 against f_1

$WSSQ$ previous	257.18
$WSSQ$ current	28.293
Number of parameters previous	2
Number of parameters current	4
Number of x values	30
Akaike AIC previous	68.458
Akaike AIC current	6.2425, ER = 3.2346E+13
Schwarz SC previous	71.260
Schwarz SC current	11.847
Mallows' C_p	210.34, $C_p/2$ = 105.17
Numerator degrees of freedom	2
Denominator degrees of freedom	26
F test statistic (FS)	105.17
$P(F \geq FS)$	0.0000
$P(F \leq FS)$	1.0000
5% upper tail point	3.3690
1% upper tail point	5.5263

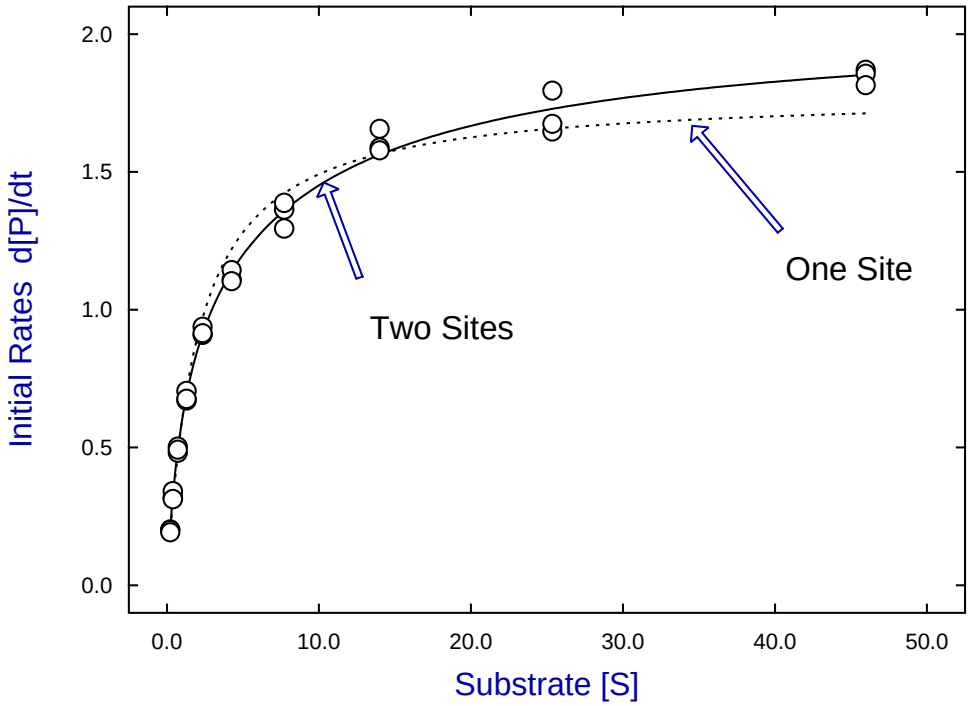
Conclusion based on F test

Reject previous model at 1% significance level

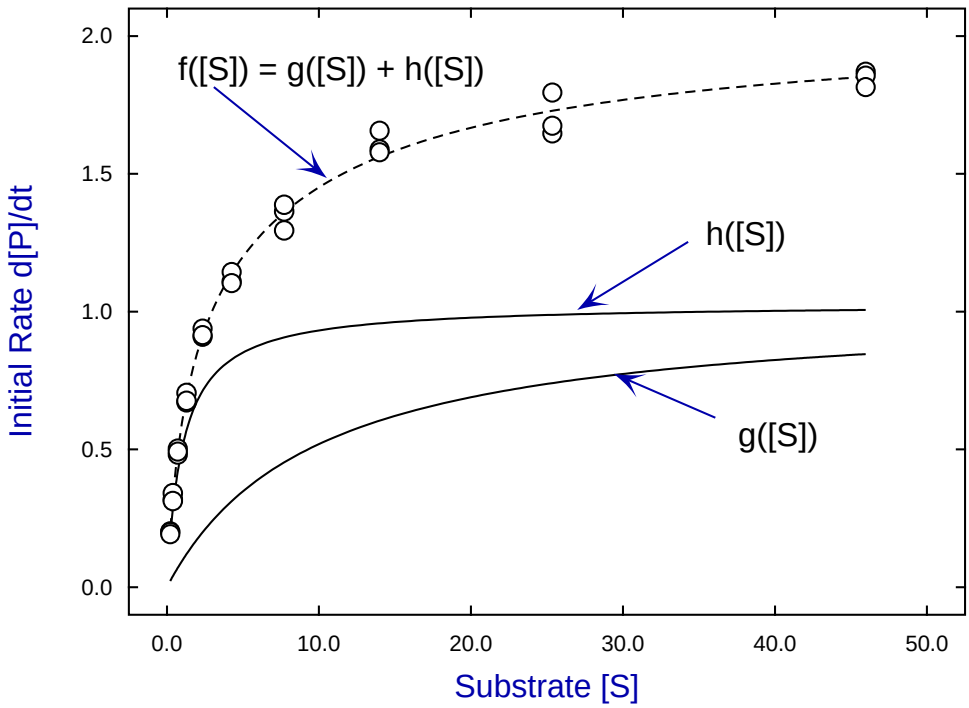
There is strong support for the extra parameters

Tentatively accept the current best fit model

Fitting 1 and 2 Site Michaelis-Menten Models



Deconvolution: 2 Michaelis-Mentens



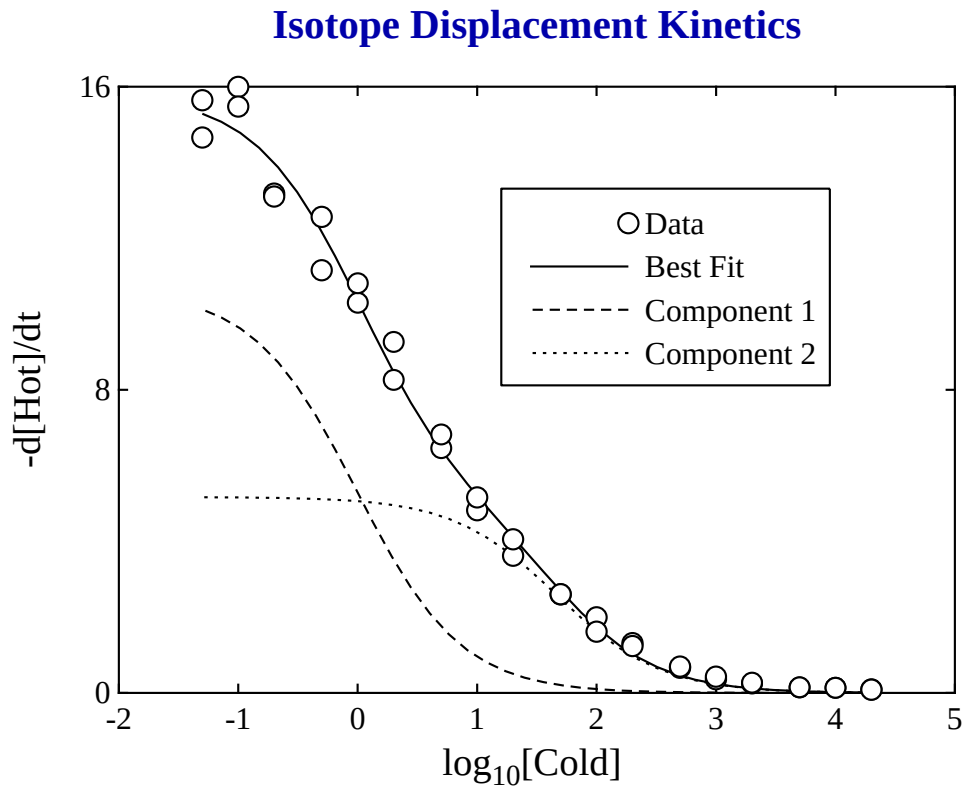
Example 2: Isotope displacement mode

When there is no appreciable kinetic isotope effect, that is, the binding or kinetic transformation process is the same whether the substrate is labeled or not, this allows experiments in which labeled ligand is displaced by unlabeled ligand, or where the flux of labeled substrate is inhibited by unlabeled substrate. Since the ratios of labeled ligand to unlabeled ligand in the bound state, free state, and in the total flux are equal, a modified form of Michaelis-Menten equations can be used to model the binding or kinetic processes. For instance, suppose that total substrate, S say, consists of labeled substrate, $[Hot]$ say, and unlabeled substrate, $[Cold]$ say. Then the flux of labeled substrate for $k \geq 1$ active sites will be given by

$$-\frac{d[Hot]}{dt} = \frac{V_{max_1}[Hot]}{K_{m_1} + [Hot] + [Cold]} + \frac{V_{max_2}[Hot]}{K_{m_2} + [Hot] + [Cold]} + \cdots + \frac{V_{max_k}[Hot]}{K_{m_k} + [Hot] + [Cold]}.$$

So, if $[Hot]$ is kept fixed and $[Cold]$ is regarded as the independent variable, then program **mmfit** can be used to fit the resulting data. In other words, cold substrate is being used as a competitive inhibitor of the flux of hot substrate in such experiments.

Using the isotope displacement option in program **mmfit** with the default test file `hotcold.tf1` establishes that two sites is a statistically significant improvement over one site, and leads to the following deconvolution plot to display the best-fit curve together with the separate components.



Note that an important difference between using **mmfit** in this mode rather than in straightforward kinetic mode is that the kinetic constants are modified in the following sense: the apparent V_{max} values estimated are actually the true values multiplied by the concentration of labeled substrate, while the apparent K_m values estimated are the true ones plus the concentration of labeled substrate.

Where the actual concentration of $[Hot]$ is known it is possible to fit such data in a more satisfactory and discerning manner by using SimFIT program **qnfit**, where the $[Hot]$ can be input as a fixed constant term so that the actual kinetic constants can be estimated rather than the apparent ones mentioned above.

Theory

Quasi steady-state enzyme kinetics is actually based on the assumption that the substrate concentration remains constant, i.e. $dS/dt = 0$, while the initial rate of product production $dP/dt \geq 0$ is measured. Although it is a contradiction of nomenclature it is a widely used short hand convention nevertheless that an initial rate $v(S)$ can defined as a flux from substrate S into product P as follows

$$v(S) = \frac{dP}{dt} = -\frac{dS}{dt}$$

and, in the case of $k \geq 1$ independent active sites, the appropriate model equation is

$$v(S) = \frac{V_{max_1} S}{K_{m_1} + S} + \frac{V_{max_2} S}{K_{m_2} + S} + \cdots + \frac{V_{max_k} S}{K_{m_k} + S}.$$

In bygone times before the advent of computers, experimentalists had to fit such equations by plotting in transformed spaces, such as the Lineweaver-Burke double reciprocal plot, and then extrapolating to estimate slopes and intercepts, but thankfully this era is long since gone. However, this does not mean that fitting such an equation by constrained weighted least squares is a simple process. It is not. In fact the case with $k = 1$ is trivial, the case with $k = 2$ is reasonable, but the cases $k > 2$ require data that is very extensive and accurate, and where the parameters are sufficiently distinct to allow model discrimination. For this particular model that requires V_{max_i} values to be similar, but K_{m_i} to be distinct.

Program **mmfit** performs the following steps.

1. The v values are first weighted using $w_i = 1/se_i^2$, or used unweighted if all $se_i = 1$.
2. Using the ranges of S_i and v_i the data are transformed into internal coordinates of order unity.
3. Possible starting estimates are calculated for the parameters based on the internal coordinates, and then these are altered by adding pseudo-random perturbations until an approximate minimum value for the weighted sum of squares is located.
4. The parameters are then transformed into internal coordinates that will hopefully be of order unity to stabilize the optimization.
5. From these random starting estimates the lowest and highest possible limits are calculated, then constrained optimization is performed by the quasi-Newton technique.
6. The internal parameters are transformed back into user-space, and the Hessian is estimated at the solution point then inverted to calculate the parameter covariance matrix.
7. The order of parameters is permuted so that the subscripts for $i = 1, 2, \ldots, k$ refer to best-fit parameters in the order $V_{max_1} \leq V_{max_2} \leq \ldots \leq V_{max_k}$. This is to allows retrospective comparison of fits to alternative data sets.
8. The apparent (overall) V_{max} is calculated as the sum of the V_{max_i} and the apparent (overall) K_m is calculated numerically.
9. Analysis of the residuals is performed together with numerous statistical procedures to ascertain goodness of fit, parameter reliability, and model discrimination.
10. Results tables and graphs are then provided.

Program **mmfit** allows users to control the random search for starting estimates and the technique to be used for calculating the gradient vector, and should the cases with $k > 2$ be required, users can perform extensive random searches to obtain starting estimates that can be input retrospectively for manual starts. If these steps do not succeed it is time to try the SIMFIT advanced curve-fitting program **qnfit**.

8.5.2 Fitting High–Low affinity ligand binding models

Ligand binding curves can be fitted by one binding site models or multiple binding sites with different affinity. A distinction has to be made between high/low affinity receptor sites that are independent and can only show negative cooperativity, and allosteric and other site-site interactions that can also give positive cooperativity.

Example 1: Ligand varied mode

From the main SIMFIT menu select [A/Z], open program **hlfit**, select the ligand-varied option, and view the default test file `hlfit.tf4` which has the following data.

x	y	$se(y)$
0.021759	0.19832	0.0091144
0.021759	0.19438	0.0091144
0.021759	0.18094	0.0091144
0.039440	0.30473	0.0047306
0.039440	0.29537	0.0047306
0.039440	0.29883	0.0047306
0.071490	0.46465	0.015273
0.071490	0.49460	0.015273
0.071490	0.48484	0.015273
0.12958	0.71278	0.048762
0.12958	0.67885	0.048762
0.12958	0.61663	0.048762
0.23488	0.87238	0.048295
0.23488	0.80269	0.048295
0.23488	0.89546	0.048295
0.42575	1.0246	0.044998
0.42575	1.1137	0.044998
0.42575	1.0806	0.044998
0.77172	1.4145	0.062457
0.77172	1.2934	0.062457
0.77172	1.3806	0.062457
1.3988	1.3619	0.13387
1.3988	1.6295	0.13387
1.3988	1.4897	0.13387
2.5355	1.7047	0.19446
2.5355	1.4435	0.19446
2.5355	1.8236	0.19446
4.5959	1.7486	0.043681
4.5959	1.7613	0.043681
4.5959	1.8298	0.043681

The columns contain data in the following format.

1. **Column 1**: the non–negative ligand concentration x which must be in non-decreasing order.
2. **Column 2**: the non–negative response y presumed to be dependent on fractional saturation of receptor or binding site at the concentration in column 1.
3. **Column 3**: the positive sample standard deviation of the replicate response measurements.
   Note that column 3 can be omitted or set to 1 if unweighted regression is required.

To illustrate the functionality of the SimFIT program **hlfit** we shall fit a one site model followed by a two site model (or mixture of two receptor types) and see if any improvement in fit can be supported by statistical analysis. The two models are as follows.

$$f_1(x) = \frac{AK_a x}{1 + K_a x} + C$$
$$f_2(L) = \frac{A_1 K_{a_1} x}{1 + K_{a_1} x} + \frac{A_2 K_{a_2} x}{1 + K_{a_2} x} + C$$

To fit these two models, choose to start fitting at order 1 and end fitting at order 2, using the further default settings but with $C = 0$ as there is no background signal with these data. This leads to the following results tables.

Table 1: For best-fit order 1 saturation function f_1

Number	Parameter	Value	Std. Error	Lower95%cl	Upper95%cl	p	
1	A	1.7482	0.038529	1.6693	1.8271	0.0000	
2	K_a	5.2161	0.17513	4.8574	5.5749	0.0000	

Apparent Y_{max} (i.e. $A_1 + A_2 + ... + A_n$) = 1.7482
Apparent K_a (i.e. x_0 where $f(x_0) - C = Y_{max}/2$) = 0.19171

Parameter correlation matrix

1	
-0.8715	1

Table 2: For best-fit order 2 saturation function f_2

Number	Parameter	Value	Std. Error	Lower95%cl	Upper95%cl	p	
1	A_1	0.91175	0.24512	0.40790	1.4156	0.0010	
2	A_2	1.0625	0.30555	0.43439	1.6905	0.0018	
3	K_{a_1}	0.97501	0.68571	-0.43449	2.3845	0.1669	*
4	K_{a_2}	8.5829	2.0044	4.4629	12.703	0.0002	

Apparent Y_{max} (i.e. $A_1 + A_2 + ... + A_n$) = 1.9742
Apparent K_a (i.e. x_0 where $f(x_0) - C = Y_{max}/2$) = 0.31272

Parameter correlation matrix

1			
-0.9770	1		
0.9019	-0.9685	1	
0.9845	-0.9936	0.9385	1

In order to determine if a significant improvement in fit has resulted we need to consider the following questions.

1. Are the parameters well-determined with both fits ?
2. Does the residuals analysis indicate satisfactory fits ?
3. Does the F test for excess variance support model f_2 in preference to f_1 ?
4. Can the best-fit curves be seen to differ when plotted against the data ?
5. Does the graphical deconvolution display convincing evidence that both components of f_2 are contributing to the overall fit ?

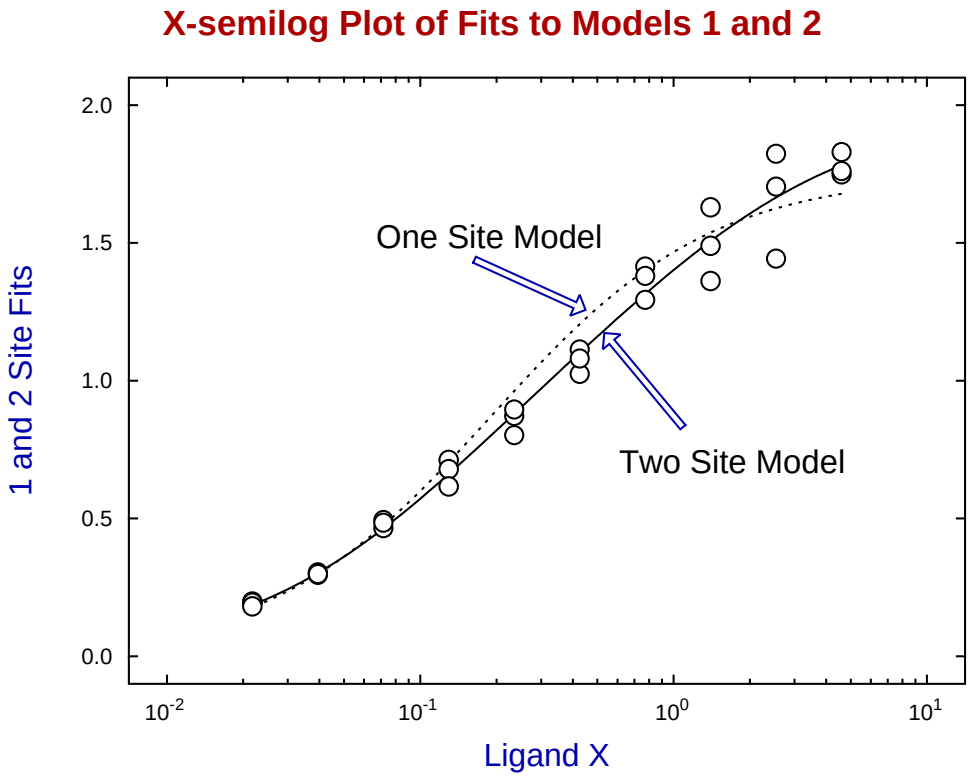
The results displayed in Tables 1 and 2 show that both models fit well with parameters that differ significantly from zero. Table 3 indicates that an excellent fit has resulted for model f_2, and Table 4 supports the conclusion that there is statistical evidence that model f_2 should be accepted as explaining the data better than model f_1. This is then further emphasized by the graphical displays showing the data with best-fit curves for f_1 and f_2, and the deconvolution of the f_2 fit into the two contributing components. The concentration is often plotted on a logarithmic scale which is then proportional to chemical potential.

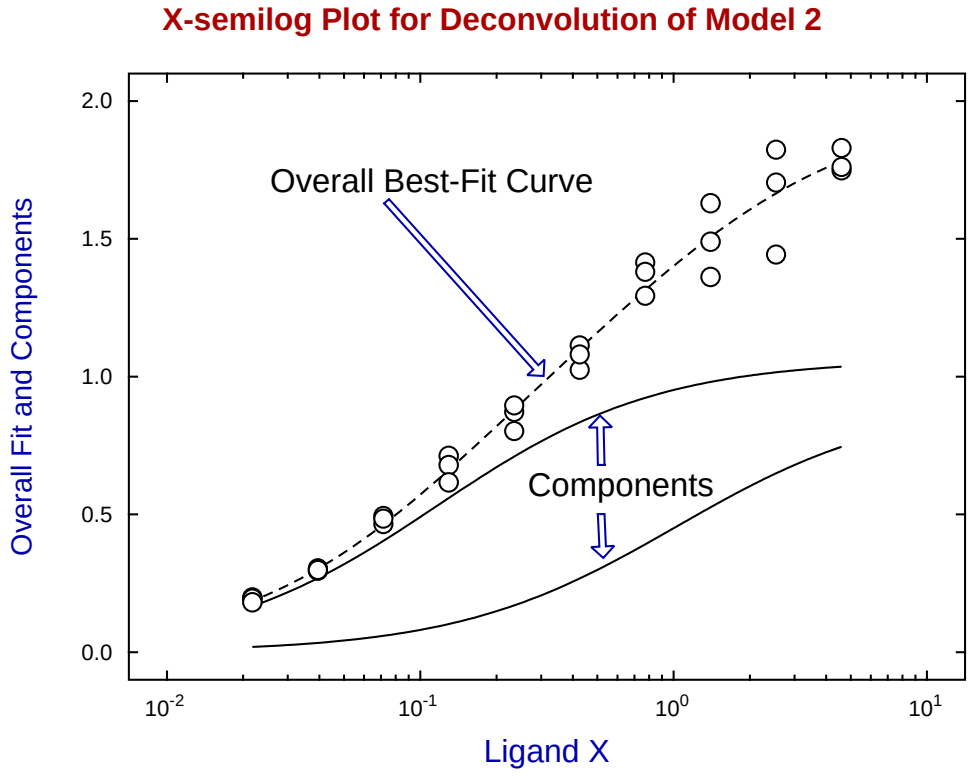
Table 3: Goodness of fit for model f_2

Analysis of residuals: $WSSQ$	29.952
$P(\chi^2 \geq WSSQ)$	0.2696
$R^2, cc(theory, data)^2$	0.9834
Largest Absolute relative residual	14.25%
Smallest Absolute relative residual	0.26%
Average Absolute relative residual	4.42%
Absolute relative residuals in range 0.1-0.2	6.67%
Absolute relative residuals in range 0.2-0.4	0.00%
Absolute relative residuals in range 0.4-0.8	0.00%
Absolute relative residuals > 0.8	0.00%
Number of negative residuals (m)	14
Number of positive residuals (n)	16
Number of runs observed (r)	21
$P(\text{runs} \leq r : \text{ given } m \text{ and } n)$	0.9820
5% lower tail point	11
1% lower tail point	9
$P(\text{runs} \leq r : \text{ given } m \text{ plus } n)$	0.9879
$P(\text{signs} \leq \text{ least number observed})$	0.8555
Durbin-Watson test statistic	3.1269 > 2.5, -ve serial correlation?
Shapiro-Wilks W statistic	0.9754
Significance level of W	0.6948
Akaike AIC (Schwarz SC) stats	7.9520 (13.557)
Verdict on goodness of fit: incredible	

Table 4: F test results for model f_2 against f_1

$WSSQ$ previous	86.634
$WSSQ$ current	29.952
Number of parameters previous	2
Number of parameters current	4
Number of x values	30
Akaike AIC previous	35.815
Akaike AIC current	7.9520, ER = 1.1230e+06
Schwarz SC previous	38.617
Schwarz SC current	13.557
Mallows' C_p	49.203, $C_p/2$ = 24.602
Numerator degrees of freedom	2
Denominator degrees of freedom	26
F test statistic (FS)	24.602
$P(F \geq FS)$	0.0000
$P(F \leq FS)$	1.0000
5% upper tail point	3.3690
1% upper tail point	5.5263
Conclusion based on F test	
Reject previous model at 1% significance level	
There is strong support for the extra parameters	
Tentatively accept the current best fit model	


X-semilog Plot of Fits to Models 1 and 2
1 and 2 Site Fits
2.0
1.5
1.0
0.5
0.0
One Site Model
Two Site Model
10^{-2}
10^{-1}
10^{0}
10^{1}
Ligand X


X-semilog Plot for Deconvolution of Model 2
Overall Fit and Components
2.0
1.5
1.0
0.5
0.0
Overall Best-Fit Curve
Components
10^{-2}
10^{-1}
10^{0}
10^{1}
Ligand X

Example 2: Isotope displacement mode

When there is no appreciable kinetic isotope effect, that is, the binding and response process process is the same whether the ligand is labeled or not, this allows experiments in which labeled ligand is displaced by unlabeled ligand. Since the ratios of labeled ligand to unlabeled ligand in the bound state, and free state are equal, a modified form of high-low affinity sites equations can be used to model the binding processes. For instance, suppose that total ligand, L say, consists of labeled ligand held constant, $[Hot]$ say, and unlabeled ligand varied, $[Cold]$ say. Then the response of labeled substrate for $n \geq 1$ active sites will be given by

$$f([Cold]) = \frac{A_1 K_{a_1}[Hot]}{1 + K_{a_1}([Hot] + [Cold])} + \frac{A_2 K_{a_2}[Hot]}{1 + K_{a_2}([Hot] + [Cold])} + \cdots + \frac{A_n K_{a_n}[Hot]}{1 + K_{a_n}([Hot] + [Cold])}.$$

So, if $[Hot]$ is kept fixed and $[Cold]$ is regarded as the independent variable, then program **hlfit** can be used to fit the resulting data. In other words, cold substrate is being used as a competitive inhibitor of the saturation by hot ligand in such experiments. Note that the parameters estimated will be clear when writing the saturation with $[Hot] = u$ and $[Cold] = v$ as follows

$$\begin{aligned} f(u) &= \frac{AK_a u}{1 + K_a u} \\ g(u, v) &= \frac{AK_a u}{1 + K_a(u + v)} \\ &= \frac{\alpha\beta}{1 + \beta v} \\ \alpha &= Au \\ \beta &= \frac{K_a}{1 + K_a u}. \end{aligned}$$

This is how the estimated parameters displayed by program **hlfit** as in Table 5 must be interpreted, that is, $\hat{A}$ estimated is really an estimate for Au and $\hat{K_a}$ estimated is really an estimate for $K_a/(1 + K_a u)$.

Using the isotope displacement option in program **hlfit** with the default test file `hotcold.tf1` establishes that two sites is a statistically significant improvement over one site, and leads to the following deconvolution plot to display the best-fit curve together with the separate components.

Table 5: For best-fit order 2 isotope displacement function

Number	Parameter	Value	Std. Error	Lower95%cl	Upper95%cl	p
1	B_1	10.485	2.4284	5.5380	15.431	0.0001
2	B_2	239.06	46.121	145.11	333.00	0.0000
3	K_1	1.0124	0.19498	0.61521	1.4095	0.0000
4	K_2	0.021593	0.0063982	0.0085604	0.034626	0.0019

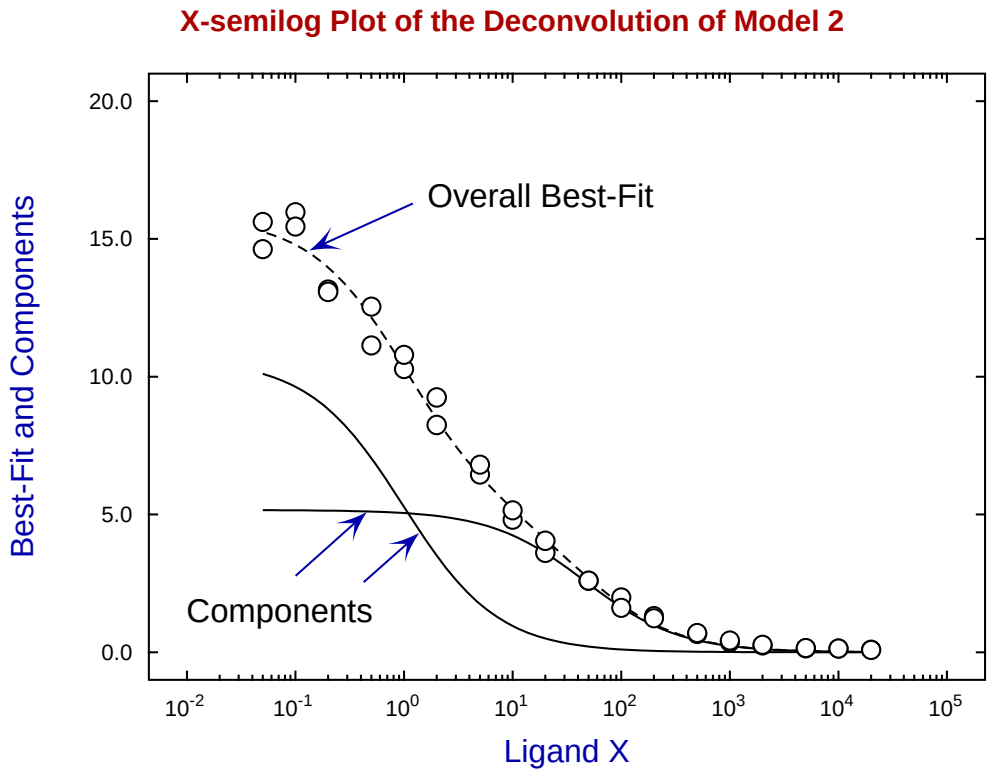
Apparent Y_{max} (i.e. $B_1K_1 + B_2K_2 + \ldots + B_nK_n$) = 15.776
Apparent K_a (i.e. x_0 where $f(x_0) - C = Y_{max}/2$) = 2.5163

Parameter correlation matrix

1			
0.5405	1		
-0.9799	-0.4559	1	
-0.7615	-0.9450	0.6712	1

Note that an important difference between using **hlfit** in this mode rather than in straightforward binding mode is that the binding constants are modified in the following sense, as previously described.

Where the actual concentration of $[Hot]$ is known it is possible to fit such data in a more satisfactory and discerning manner by using SimFIT program **qnfit**, where the $[Hot]$ can be input as a fixed constant term so that the actual amplitudes A_i and binding constants K_{a_i} can be estimated, rather than the apparent ones mentioned above.



Theory

SimFIT program **hlfit** assumes that a response is measured that depends on the fractional saturation of binding sites with possibly differing affinity. The amplitude factors A_i can be interpreted as being proportional to the population of the receptor types, possibly complicated by the situation where the fractional receptor occupancy does not give the same response for the different receptor types. Program **hlfit** also allows for the situation where there is background noise at level C that has to be estimated then substracted from the data so that the response is zero at zero ligand concentration.

The first thing to point out is that this model does not have a standard binding polynomial to act as a partition function, as it is a weighted sum of individual independent sites and can therefore only show negative cooperativity. To understand the meaning of the parameters being estimated by program **hlfit** consider the binding of a single ligand X to a protein P at equlibrium so that this is the binding process

$$P + X = PX$$

with the association constant K_a defined as

$$K_a = \frac{[PX]}{[P][X]}$$

and the fractional saturation of the protein with ligand X is $0 \le y \le 1$ defined as

$$y = \frac{K_a[X]}{1 + K_a X}.$$

However the response measured will be the fractional saturation multiplied by an arbitrary amplitude factor A, unless fractional saturation is measured when the individual amplitude factors would be nonnegative and

would have sum one. Some versions of programs **hlfit** and **qnfit** provide this feature as an additional option. In addition, in some experiments there is an unavoidable background level C which can be estimated during the fitting, or better estimated independently and then subtracted from the measured response, so that $Y(0) = 0$.

In bygone days before the advent of computers, experimentalists had to fit binding equations by plotting in transformed spaces, such as the Scatchard plot, and then extrapolating to estimate slopes and intercepts, but thankfully this era has long since gone. However, this does not mean that fitting such an equation by constrained weighted least squares is a simple process. It is not. In fact the case with $k = 1$ is trivial, the case with $k = 2$ is reasonable, but the cases $k > 2$ require data that is very extensive and accurate, and where the parameters are sufficiently distinct to allow model discrimination. For this particular model that requires amplitudes A_i values to be similar, but binding constants K_i to be distinct.

Program **hlfit** performs the following steps.

1. The y_i values are first weighted using $w_i = 1/se_i^2$, or used unweighted if all $se_i = 1$.
2. Using the ranges of x_i and y_i the data are transformed into internal coordinates of order unity.
3. Possible starting estimates are calculated for the parameters based on the internal coordinates, and then these are altered by adding pseudo-random perturbations until an approximate minimum value for the weighted sum of squares is located.
4. The parameters are then transformed into internal coordinates that will hopefully be of order unity to stabilize the optimization.
5. From these random starting estimates the lowest and highest possible limits are calculated, then constrained optimization is performed by the quasi-Newton technique.
6. The internal parameters are transformed back into user-space, and the Hessian is estimated at the solution point then inverted to calculate the parameter covariance matrix.
7. The order of parameters is permuted so that the subscripts for $i = 1, 2, \ldots, k$ refer to best-fit parameters in the order $A_1 \leq A_2 \leq \ldots \leq A_n$. This is to allows retrospective comparison of fits to alternative data sets.
8. The apparent (overall) A is calculated as the sum of the A_i (or $A_i K_{a_i} i$ for isotope displacement) and the apparent (overall) K_a is calculated numerically.
9. Analysis of the residuals is performed together with numerous statistical procedures to ascertain goodness of fit, parameter reliability, and model discrimination.
10. Results tables and graphs are then provided.

Program **hlfit** allows users to control the random search for starting estimates and the technique to be used for calculating the gradient vector, and should the cases with $k > 2$ be required, users can perform extensive random searches to obtain starting estimates that can be input retrospectively for manual starts. If these steps do not succeed it is time to try the SimFIT advanced curve-fitting program **qnfit**.

Although **hlfit** can be used to fit more than two classes of sites it must be stressed that this requires extremely accurate data over a large range of ligand concentration, and the automatic estimation of suitable starting estimates may have to be replaced by user-supplied estimates. In any case it will be extremely difficult to interpret binding data in terms of more than two classes of binding sites by curve-fitting alone unless there is additional experimental evidence.

8.5.3 Fitting allosteric and cooperative ligand binding models

Cooperative ligand binding models are used in the situation where a protein or receptor has more than one type of binding site and these are linked in such a way as to display deviations from normal hyperbolic binding. If a receptor has $n > 1$ binding sites that differ in binding constants but are independent this can only give rise to apparent negative cooperativity. If the sites are linked in that the binding to one site influences the subsequent binding of further ligands then positive or mixed cooperativity can be exhibited. These terms will be defined subsequently.

From the main SimFIT menu choose [A/Z] then open program **sffit** and examine the default test file `sffit.tf4` which contains the following data.

x	*y*	*se*
0.085504	0.10022	0.0026739
0.085504	0.10533	0.0026739
0.085504	0.10142	0.0026739
0.11434	0.14319	0.010065
0.11434	0.16178	0.010065
0.11434	0.14578	0.010065
0.15291	0.24510	0.012043
0.15291	0.22191	0.012043
0.15291	0.22786	0.012043
0.20449	0.30735	0.019939
0.20449	0.32957	0.019939
0.20449	0.28978	0.019939
0.27346	0.43824	0.0071355
0.27346	0.43342	0.0071355
0.27346	0.44746	0.0071355
0.36569	0.57197	0.014359
0.36569	0.56004	0.014359
0.36569	0.58863	0.014359
0.48903	0.64381	0.030621
0.48903	0.63820	0.030621
0.48903	0.69382	0.030621
0.65398	0.75455	0.017667
0.65398	0.78973	0.017667
0.65398	0.77504	0.017667
0.87456	0.81456	0.030889
0.87456	0.82605	0.030889
0.87456	0.76774	0.030889
1.1695	0.95153	0.029772
1.1695	0.89315	0.029772
1.1695	0.91216	0.029772

The columns contain data in the following format.

1. **Column 1**: the non–negative ligand concentration x which must be in non-decreasing order.
2. **Column 2**: the non–negative response y presumed to be dependent on fractional saturation of receptor or binding site at the concentration in column 1.
3. **Column 3**: the positive sample standard deviation of the replicate response measurements. This column can be omitted or set to 1 if unweighted regression is required.

The model $f(x)$ fitted by SIMFIT program **sffit** for n binding sites in the presence of ligand at concentration x is based on a binding polynomial $p(x)$ and fractional saturation function $y(x)$ expressed using overall binding constants K_i as follows

$$\begin{aligned} p(x) &= 1 + K_1x + K_2x^2 + \cdots + K_nx^n \\ y(x) &= \frac{1}{n}\frac{d\log p(x)}{d\log x} \\ &= \frac{1}{n}\frac{xp'(x)}{p(x)} \\ f(x) &= Zy(x) + C. \end{aligned}$$

Here Z is an arbitrary factor relating the observed response to fractional saturation, while C is a possible background noise in the absence of any ligand. It is supposed that the number of sites n would be known in advance while the arbitrary scaling factor Z and background noise C would be estimated in a preliminary run and used to normalize the data so that $0 \le f(x) \le 1$ for $x \ge 0$.

Before proceeding to discuss the results from analyzing the test data two things should be noted.

1. When the data have been normalized so that $Z = 1$ and $C = 0$, as with the data in test file `sffit.tf4`, program **sffit** begins by scaling the data internally, performing a L_1 norm fitting procedure for starting estimates followed by a refinement by random searching. For low order models with accurate data over a large range of x this will often mean that the starting estimates are very close to the best-fit parameters, and the quasi-Newton constrained regression program will draw attention to the fact that only a small percentage reduction in the objective function $WSSQ$ has been achieved. This simply indicates how good the **sffit** algorithm has been in calculating starting estimates.

2. Users are given the option to fit several values of n in sequence with statistical tests for model discrimination and goodness of fit. This facility is provided for preliminary analysis in the case where Z is estimated and/or n is not known in advance, and also so that **sffit** can be used as an empirical model for data smoothing

Proceeding to start optimization at $n = 2$ and end optimization at $n = 2$ with $Z = 1$ and $C = 0$ means that the following model was fitted

$$f(x) = \left(\frac{1}{2}\right)\frac{K_1x + 2K_2x^2}{1 + K_1x + K_2x^2}$$

leading to this table of results.

For best-fit order 2 function (f for fixed parameter)

Number	Parameter	Value	Std. error	Lower95%cl	Upper95%cl	p	
1	K_1	1.0734	0.063088	0.94414	1.2026	0.0000	
2	K_2	10.042	0.17298	9.6881	10.397	0.0000	
3	Z	1.0000	0.0000	1.0000	1.0000	...	f
4	C	0.0000	0.0000	0.0000	0.0000	...	f

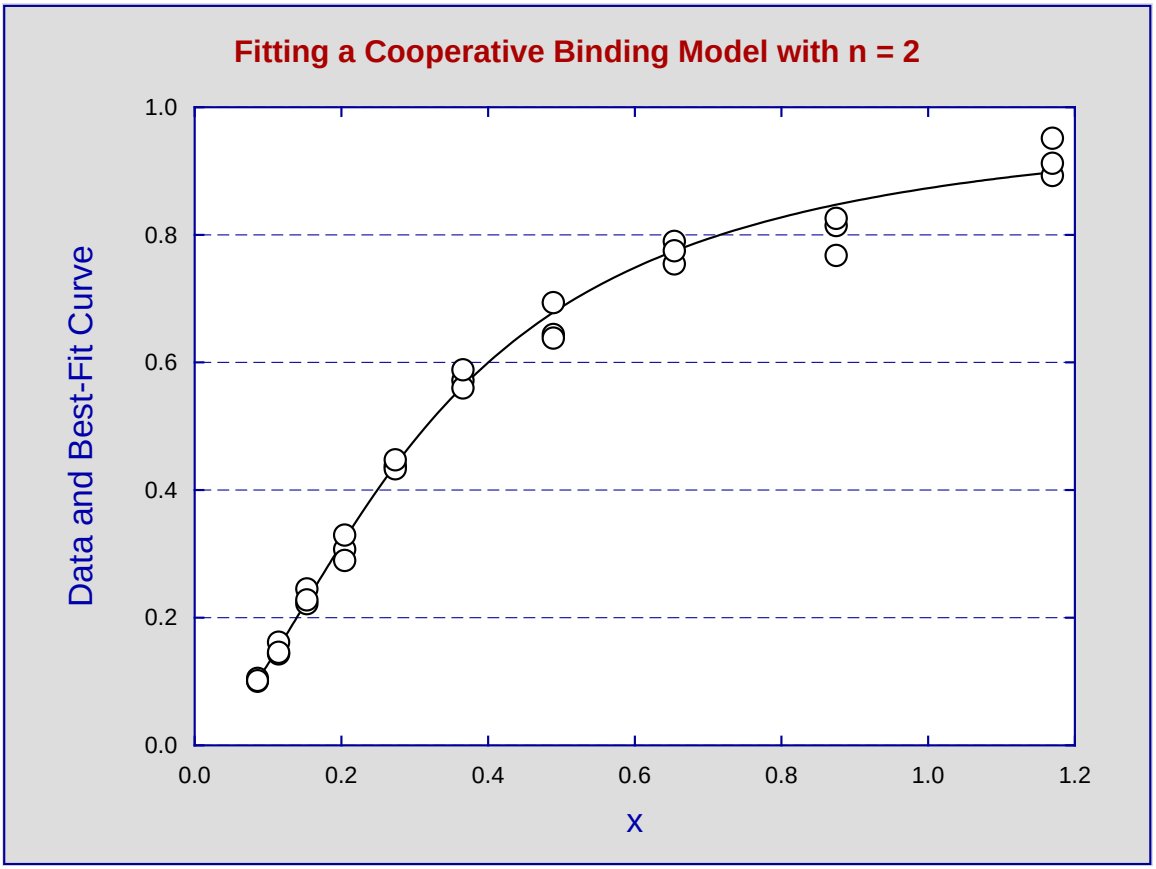
Apparent V_{max} (i.e. Z or $f(\infty) - C$) = 1.0000
Apparent K_m (i.e. x where $f(x) - C = Z/2$) = 0.31554

Parameter correlation matrix (f for fixed parameter)

1			
-0.5562	1		
f	f	f	
f	f	f	f

The excellent fit will be clear from the analysis of residuals and plot of data and best-fit curve as shown next.

Goodness of fit for model with $n = 2$	
Analysis of residuals: $WSSQ$	33.100
$P(\chi^2 \geq WSSQ)$	0.2321
$R^2, cc(theory, data)^2$	0.9928
Largest Absolute relative residual	10.87%
Smallest Absolute relative residual	0.04%
Average Absolute relative residual	3.47%
Absolute relative residuals in range 0.1-0.2	3.33%
Absolute relative residuals in range 0.2-0.4	0.00%
Absolute relative residuals in range 0.4-0.8	0.00%
Absolute relative residuals > 0.8	0.00%
Number of negative residuals (m)	17
Number of positive residuals (n)	13
Number of runs observed (r)	22
P(runs $\leq r$: given m and n)	0.9957
5% lower tail point	10
1% lower tail point	9
P(runs $\leq r$: given m plus n)	0.9959
P(signs $\leq$ least number observed)	0.5857
Durbin-Watson test statistic	2.441
Shapiro-Wilks W statistic	0.9781
Significance level of W	0.7729
Akaike AIC (Schwarz SC) stats	6.9509 (9.7529)
Verdict on goodness of fit: incredible	



At this point various options are available for further study of the best fitting order two saturation function. Choosing cooperativity analysis we first observe the percentage saturation points given the K_i values, which allows users to see at a glance the range of saturation spanned by the range of the data.

Overall association constants and % saturation points		
K_1	1.0734	
K_2	10.042	
X start	at $x = 0.085504$	
X stop	at $x = 1.695$	
$y = 0.05$	at $x = 0.0051378$	The 5% saturation point
$y = 0.10$	at $x = 0.084086$	The 10% saturation point
$y = 0.50$	at $x = 0.31556$	The 50% saturation point
$y = 0.90$	at $x = 1.1843$	The 90% saturation point
$y = 0.95$	at $x = 1.9381$	The 95% saturation point

Evidently the range of these experimental data spans the range from around 10% saturation to just over the point of 90% saturation. Perhaps it will not be often that experimentalists will be able to achieve such a wide span.

The next table displays the values of the association constants and reciprocals for overall association constants K_i, Adair constants A_i and Adair constants corrected for statistical factors B_i so that the results can be compared between alternative computer packages. Note that SimFIT program **qnfit** can be used to fit these alternative model formulations if confidence limits on parameter estimates and parameter correlation matrices are required. All other goodness of fit and model discrimination results are the same irrespective of the model formulation.

Alternative expressions for binding constants						
Number	K	$1/K$	A	$1/A$	B	$1/B$
1	1.0734	0.93165	1.0734E+00	0.93165	0.53668	1.8633
2	10.042	0.099577	9.3560E+00	0.10688	18.712	0.053442

Intrinsic cooperativity coefficient	Value	Sign
$B_2 - B_1$	18.175	+

Intrinsic cooperativity coefficients are particularly easy to interpret in molecular terms. For instance, if $B_i > B_{i-1}$ this indicates that when ligand is bound to $i-1$ sites the affinity increase for when the the next ligand binds, whereas when $B_i < B_i$ the affinity decreases. So, when there are only two binding sites, the condition $B_2 > B_1$ is equivalent to positive cooperativity and the condition $B_2 < B_1$ is correctly referred to as negative cooperativity. Unfortunately, for more than two sites this argument breaks down due to the additional complication of species fractional populations S_i defined as

$$S_i = \frac{K_i x_i}{K_0 + K_1 X + K_2 X^2 + \cdots + K_n x^n}.$$

for $i = 0, 1, 2 \ldots, n$. Here $K_0 = 1$, $0 \le S_i \le 1$, $S_0 + S_1 + S_2 + \cdots + S_n = 1$ and the S_i measure the proportion of the macromolecule with i ligands bound as the concentration of ligand x varies. A similar measure, the species fraction S_{fi}, defined as

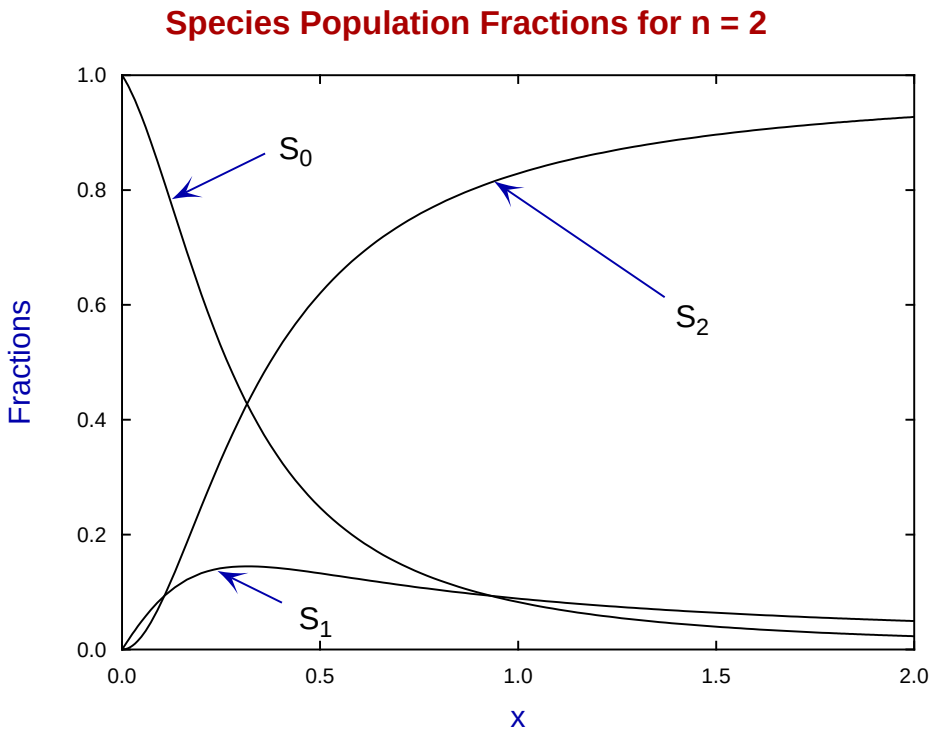
$$S_{fi} = iS_i/n$$

for $i = 1, 2, \ldots, n$ takes account of the stoichiometry and measures the contribution of the species with i ligands attached to the overall fractional saturation $y(x)$ as

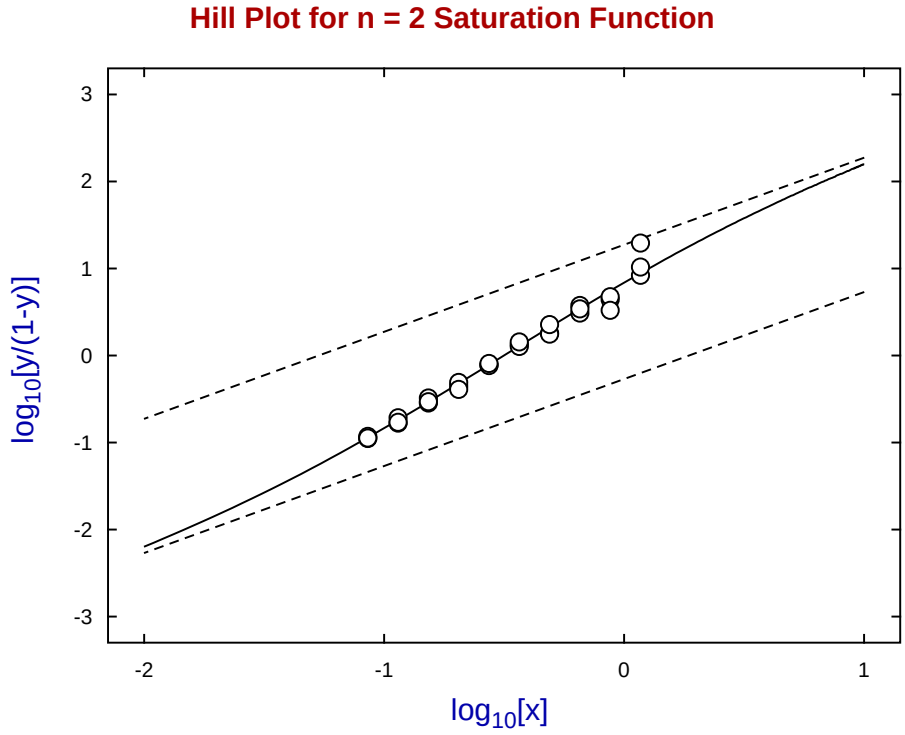
$$S_{f1} + S_{f2} + \ldots, +S_{fn} = y(x).$$

The next plot displays the population fractions for the current best-fit model showing that all the macromolecule is free from ligand at $x = 0$, then the macromolecule with one ligand appears then disappears as x increases

until eventually as $x \rightarrow \infty$ all the binding sites are occupied. It is this fact that makes the interpretation of the sign of cooperativity ambiguous when the order exceeds 2 which is where the Hill plot slope is a less ambiguous measure of the sign and extent of cooperativity when viewed as a function of ligand activity.



The Hill plot slope shown next is very simple to interpret in the case of fitting an order $n = 2$ saturation function but, as will be discussed subsequently, the situation is not so simple for order $n > 2$, and this is an area of unnecessarily great confusion.


Hill Plot for n = 2 Saturation Function
log10[y/(1-y)]
log10[x]
3
2
1
0
-1
-2
-3
-2
-1
0
1

Theory

Ligand binding theory will be presented under the following headings.

1. Historical introduction
2. Binding polynomials
3. Definition of cooperativity
4. Factorability of the binding polynomial
5. Statistical interpretation of saturation functions
6. Cooperativity analysis

Historical Introduction

In 1910 Hill [1] proposed that the sigmoid binding curve for oxygen binding to haemoglobin could be analyzed in terms of the binding of n ligands in one step with no appreciable intermediates, i.e. the mass action description

$$Hb + nX \rightleftharpoons HbX_n.$$

This leads to the Hill equation describing the fractional saturation y as a function of concentration x, and the Hill plot of $\log[y/(1-y)]$ as a function of $\log x$ as follows

$$y = \frac{Kx^n}{1 + Kx^n}$$

$$\log\left(\frac{y}{1-y}\right) = n \log x + \log K.$$

It is now realized that the Hill equation is simply an empirical equation that is at best a poor approximation to any real binding situation since:

1. it is only an appropriate representation for a one-site binding process, i.e. for $n = 1$;
2. when $n < 1$ it has an infinite slope at the origin and cannot model any realistic binding situation;
3. when $n > 1$ it has zero slope at the origin and cannot model any realistic binding situation;
4. when n is not a positive integer it is pure nonsense; and
5. using it to discuss the effect of cooperativity on graphical features such as sigmoidicity in the $y(x)$ curve, or convexity in Lineweaver-Burke or Scatchard space, has resulted in considerable confusion.

Of course, before the days of computers and nonlinear regression, fitting a straight line to a Hill plot to get a non-integer value for the estimated slope was all that could be done, and this non-integer value was correctly taken to mean that this was a result of the model being incorrect.

Nowadays no one would dream of discussing cooperative binding in terms of the Hill equation or fitting a straight line to a Hill plot but, by a serendipitous coincidence, it turns out that the variable slope of the curve obtained by transforming a saturation curve into Hill space still provides an unambiguous definition of the sign and magnitude of cooperativity that has got nothing at all to do with the Hill equation. That is because, to use receptor terminology,

$$\frac{y}{1-y} = \frac{[\text{Bound}]}{[\text{Free}]}.$$

Binding polynomials and their Hessians

In 1925 Adair [2] improved the description of binding isotherms by defining binding constants for the individual binding events, and later it came to be appreciated that these have to be normalized by statistical factors in order to discuss the affinity of receptor for ligand in adjacent binding events. In 1967 Wyman [3] rationalized the situation by pointing out that, for a non-aggregating macromolecule with n binding sites and only one ligand x varied, there would be binding polynomial which would act like a partition function in that successive terms of degree i in the polynomial are proportional to the amount of macromolecule with i ligands attached.

So now the binding of ligands to receptors can be defined for all possible cooperative binding schemes in terms of a binding polynomial $p(x)$ in the free ligand activity x, as follows

$$\begin{aligned} p(x) &= 1 + K_1 x + K_2 x^2 + \cdots + K_n x^n \\ &= 1 + A_1 x + A_1 A_2 x^2 + \cdots + \prod_{i=1}^{n} A_i x^n \\ &= 1 + \binom{n}{1} B_1 x + \binom{n}{2} B_1 B_2 x^2 + \cdots + \binom{n}{n} \prod_{i=1}^{n} B_i x^n, \end{aligned}$$

where the only difference between these alternative expressions concerns the meaning and interpretation of the binding constants. The fractional saturation is just the scaled derivative of the log of the polynomial with respect to $\log(x)$, and an important auxiliary function is $h(x)$, the Hessian of the binding polynomial defined as follows

$$\begin{aligned} y(x) &= \left(\frac{1}{n}\right) \frac{d \log p(x)}{d \log x} \\ &= \left(\frac{1}{n}\right) \frac{x p'(x)}{p(x)} \\ h(x) &= n p p'' - (n-1) p'^2. \end{aligned}$$

Definition of cooperativity

Given a binding polynomial of degree n there are $n - 1$ cooperativity coefficients c_i defined as

$$c_i = B_{i+1} - B_i \text{ for } i = 1, 2, \ldots, n-1,$$

or alternatively as $\log(B_{i+1}/B_i)$, and the interpretation of these is perfectly clear: in a situation where $c_i > 0$ the macromolecule has greater affinity for binding the $i + 1$th ligand after the ith ligand has been bound and it is perfectly reasonable to describe this as mechanistic positive cooperativity. Hence every binding situation for n ligands can be summarized by a succession of $n - 1$ signs and it might be thought that during the actual saturation of macromolecule with ligand there would be a succession of phases with possibly differing cooperativity. For instance, the sequence $+ - +$ might be supposed to give a saturation curve with positive, then negative, then positive cooperativity. Unfortunately the cooperativity coefficients cannot be interpreted in this way and they are not a unique indicator of the sign and magnitude of the type of cooperativity exhibited during the saturation process. The reason for this is simply that binding does not occur in a succession of isolated steps and at every stage for $0 < x < \infty$ every species that is possible is present, that is no ligands bound, one ligand bound, two ligands bound, etc. up to n ligands bound.

At every point in the range $0 < x < \infty$ there is a one site binding curve y_{app} with a uniquely defined apparent binding constant K_{app} according to the scheme

$$[\text{Free sites}] + X \rightleftharpoons [\text{Occupied sites}]$$

that is

$$y_{app}(x) = \frac{K_{app}x}{1 + K_{app}x}.$$

Surely all would agree that the sign and magnitude of cooperativity at that point in the saturation curve would depend on whether K_{app} is increasing or decreasing as a function of x. It turns out that

$$K_{app} = \frac{p'(x)}{np(x) - xp'(x)} \text{ and}$$
$$\frac{dK_{app}}{dx} = \frac{h(x)}{(np(x) - xp'(x))^2}$$

so that increasing affinity (i.e. positive cooperativity) requires $h(x) > 0$, decreasing affinity (i.e. negative cooperativity) requires $h(x) < 0$ while at a point where $h(x) = 0$ cooperativity changes sign. Bardsley and Wyman [4] emphasized that the magnitude of the Hill slope with respect to 1 is the unambiguous indicator of cooperativity which also depends on the sign of the Hessian as follows

$$\frac{d\log[y/(1-y)]}{d\log x} = 1 + \frac{xh(x)}{p'(x)(np(x) - xp'(x))}.$$

and Wood and Bardsley [5] proved that the Hessian can have at most $n - 2$ positive zeros.

Zeros of the binding polynomial

If the n zeros of the binding polynomial are α_i then the fractional saturation y can be expressed as

$$y = \left(\frac{x}{n}\right)\sum_{i=1}^{n}\frac{1}{x - \alpha_i},$$

but further discussion depends on the nature of the zeros.

First observe that, for a set of m groups of receptors, each with n_i independent binding sites and binding constant k_i, then the zeros are all real and

$$p(x) = \prod_{i=1}^{m}(1 + k_i x)^{n_i},$$
$$\text{and } y = \frac{1}{\sum_{i=1}^{m} n_i}\sum_{i=1}^{m}\frac{n_i k_i x}{1 + k_i x},$$

so y is just the sum of simple binding curves, giving concave down double reciprocal plots, etc.

Actually Bardsley et al [6] and [7] proved that, if a binding polynomial factorizes into m polynomials p_i with positive coefficients according to

$$p(x) = p_1(x)p_2(x)\ldots p_m(x)$$

then the Hill plot slope cannot exceed that of the Hill plot slope for any of the individual factors. As a binding polynomial can always be factorized into a product of linear factors with real negative zeros and complex conjugate pairs forming quadratic factors it might be supposed that the Hill slope can never exceed two. However, if a binding polynomial of degree > 2 has complex conjugate zeros, the Hill slope may exceed two and there may be evidence of strong positive cooperativity. That is why Hill plot slopes up to a maximum of the degree of the binding polynomial can be achieved if there are quadratic factors with negative coefficients, corresponding to a group of at least three linked binding sites.

For instance, the binding polynomial for a four site Monod-Wyman-Changeux model is

$$p(\alpha) = \frac{1}{1 + L}\left((1 + \alpha)^4 + L(1 + c\alpha)^n\right)$$

and this can factorize into the form

$$q(x) = (1 + a_1x + b_1x^2)(1 - a_2x + b_2x^2)$$

with $a_1 > 0, a_2 > 0, b_1 > 0, b_2 > 0$ under certain constraints so that the meaningless quadratic factor with a negative term allows Hill slopes greater than two.

Edelstein and Bardsley [8] subsequently explored the relationship between the Hill slope at half-saturation and the Hessian of the binding polynomial.

Statistical interpretation of saturation functions

The species fractional populations s_i which are defined for $i = 0, 1, \ldots, n$ as

$$s_i = \frac{K_i x^i}{K_0 + K_1x + K_2x^2 + \cdots + K_nx^n}$$

with $K_0 = 1$, are interpreted as the proportions of the receptors in the various states of ligation as a function of ligand activity. The species fractions defined as $y_i = is_i/n$ for $i = 1, 2, \ldots, n$ are the contributions of the species to the overall saturation. Note that

$$\sum_{i=0}^{n} s_i = 1, \text{, while}$$
$$\sum_{i=1}^{n} y_i = (1/n)d\log p/d\log x.$$

Such expressions are very useful when analyzing cooperative ligand binding data and they can be generated from the best fit binding polynomial after fitting binding curves with program **sffit**, or by interactive input of binding constants into program **simstat**. At the same time other important analytical results like factors of the Hessian and minimax Hill slope are also calculated.

The species fractional populations can be also used in a probability model to interpret ligand binding in several interesting ways. For this purpose, consider a random variable U representing the probability of a receptor existing in a state with i ligands bound. Then the the probability mass function, expected values and variance are

$$\begin{aligned}
P(U = i) &= s_i \ (i = 0, 1, 2, \ldots, n), \\
E(U) &= \sum_{i=0}^{n} is_i, \\
E(U^2) &= \sum_{i=0}^{n} i^2 s_i, \\
V(U) &= E(U^2) - [E(U)]^2 \\
&= x\left(\frac{p'(x) + xp''(x)}{p(x)}\right) - \left(\frac{xp'(x)}{p(x)}\right)^2 \\
&= n\frac{dy}{d\log x},
\end{aligned}$$

as fractional saturation y is $E(U)/n$. In other words, the slope of a semi-log plot of fractional saturation data indicates the variance of the number of occupied sites, namely; all unoccupied when $x = 0$, distribution with variance increasing as a function of x up to the maximum semi-log plot slope, then finally approaching all sites occupied as x tends to infinity. You can input binding constants into the statistical calculations procedure to see how they are mapped into all spaces, cooperativity coefficients are calculated, zeros of the binding

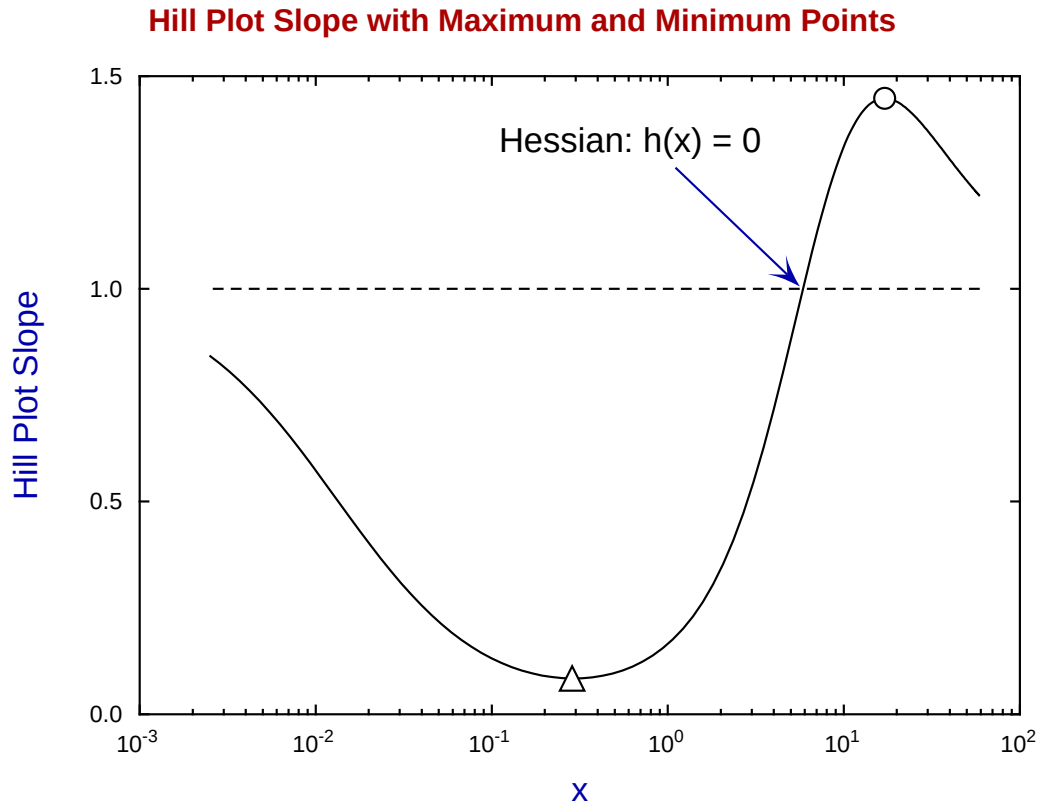
polynomial and Hessian are estimated, Hill slope is reported, and species fractions and binding isotherms are displayed, as is done automatically after every $n > 1$ fit by program **sffit**.

Cooperativity analysis

After fitting a model, program **sffit** outputs the binding constant estimates in all the conventions and, when $n > 2$ it also outputs the zeros of the best fit binding polynomial and those of the Hessian of the binding polynomial $h(x)$. The positive zeros of $h(x)$ indicate points where the theoretical one-site binding curve coinciding with the actual saturation curve at that x value has the same slope as the higher order saturation curve, which are therefore points of cooperativity change. The SimFIT cooperativity procedure allows users to input binding constant estimates retrospectively to calculate zeros of the binding polynomial and Hessian, and also to plot species population fractions.

For instance, for 4 sites with $K_1 = 100, K_2 = 10, K_3 = 1,$ and $K_4 = 0.1$, the Hessian has a positive zero at $x = 5.86139$, the minimum Hill slope in the range plotted is $0.0842,$ at $x = 0.28607$, the maximum is $1.44479,$ at $x = 17.059$, and the slope at half saturation is $1.0847,$ at $x = 6.5808$.

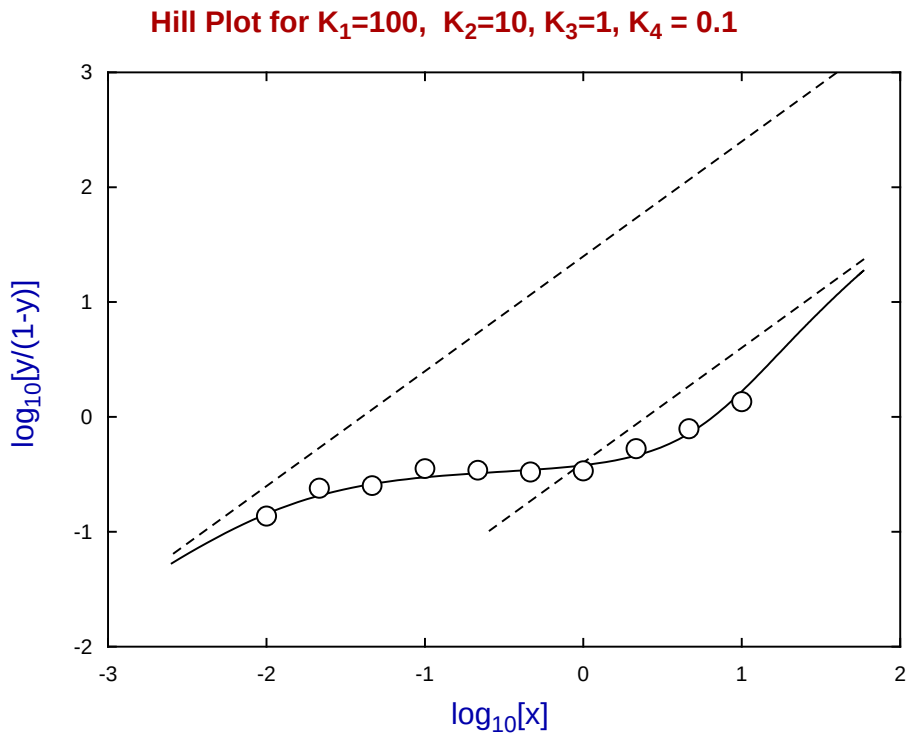
The next graph shows how the Hill plot slope varies with the maximum and minimum slopes indicated along with the point where the positive zero of the Hessian occurs.



The following graph shows the sort of complicated Hill plots that can be obtained when there are more than two cooperatively linked sites. The asymptotes are for the equation

$$y = \frac{kx}{1 + kx}$$

with $k = K_1/n$ as $x \to 0$ and $k = nK_n/K_{n-1}$ as $x \to \infty$.



References

[1] The possible effects of the aggregation of the molecules of haemoglobin on its dissociation curves.
Hill,A.V. (1910), *J.Physiol.* **40**, 4-7.

[2] The hemoglobin system. VI. The oxygen dissociation curve of hemoglobin.
Adair,G.S. (1925) *J.Biol. Chem.* **63**, 529-545.

[3] Allosteric Linkage.
Wyman,J. (1967), *J.Amer.Chem.Soc. 89.9* 2202-2218.

[4] Concerning the thermodynamic definition and graphical manifestations of positive and negative co-operativity.
Bardsley,W.G. & Wyman,J. (1978) *J. theor. Biol.* **72**, 373-376

[5] Critical points and sigmoidicity of positive rational functions.
Wood,R.M.W. & Bardsley,W.G. (1985) *Amer. Math. Month.* **92(1)**, 37-48

[6] Relationships between the magnitude of Hill plot slopes, apparent binding constants and factorability of binding polynomials and their Hessians.
Bardsley,W.G., Woolfson,R. & Mazat,J,-P. (1980) *J. theor. Biol.* **85**, 247-284

[7] Factorability of the Hessian of the binding polynomial. The central issue concerning statistical ratios between binding constants, Hill plot slope and positive and negative co-operativity.
Bardsley,W.G. & Waight,R.D. (1978) *J. theor. Biol.* **72**, 321-372

[8] Contributions of individual molecular species to the Hill coefficient for ligand binding by an oligomeric protein.
Edelstein,S.J & Bardsley,W.G. *J. Mol. Biol* (1997) **267**, 10–16

8.5.4 Fitting deviations from Michaelis-Menten kinetics

When accurate initial rate data are obtained over an extended range of substrate concentration then deviations from Michaelis-Menten kinetics such as substrate inhibition, substrate activation, sigmoidicity, and other types of cooperativity are often encountered. If artifacts such as aggregation, failing to correct for pH or ionic strength, etc. are not responsible, then the appropriate model would be positive rational function. The most important thing is to determine the degree of such a rational function, because this can be used to select a possible kinetic scheme and rate equation.

From the main SimFIT menu select [A/Z], then open program **rffit** and study the default test file `rffit.tf6` which has the following data set.

S	$v(S)$	$se(v)$
0.01000	0.02100	0.00157
0.01624	0.03409	0.00256
0.02637	0.05533	0.00415
0.04281	0.08975	0.00673
0.06952	0.14532	0.01090
0.11288	0.23421	0.01757
0.18330	0.37301	0.02798
0.29764	0.57659	0.04324
0.48329	0.83187	0.06239
0.78476	1.04968	0.07873
1.27427	1.09824	0.08237
2.06914	0.98829	0.07412
3.35982	0.87359	0.06552
5.45559	0.86246	0.06468
8.85867	0.96350	0.07226
14.38450	1.08641	0.08148
23.35721	1.07355	0.08052
37.92690	0.87789	0.06584
61.58482	0.62001	0.04650
100.00000	0.40461	0.03035

The columns contain data in the following format.

1. **Column 1**: the non–negative substrate concentration S which must be in non-decreasing order.
2. **Column 2**: the non–negative initial rate $v(S)$ presumed to be dependent on substrate in column 1.
3. **Column 3**: the positive sample standard deviation of the replicate rate measurements.
   This column can be omitted or set to 1 if unweighted regression is required.

SimFIT program **rffit** fits positive rational functions of the following form

$$v(S) = \frac{\alpha_0 + \alpha_1 S + \alpha_2 S^2 + \cdots + \alpha_n S^n}{\beta_0 + \beta_1 S + \beta_2 S^2 + \cdots + \beta_n S^n}$$

which will be referred to as a n:n function. This has $2n+1$ independent nonnegative parameters $\alpha_i \geq 0, \beta_i \geq 0$ so we define $\beta_0 = 1$. In addition, it is is usually the case that $\alpha_0 = 0$ so that $v(0) = 0$ and, if dead-end enzyme-substrate complexes are assumed, it will also be convenient to set $\alpha_n = 0$ to model the case where $v(\infty) = 0$.

Some observations on curve fitting n:n functions

Program **rffit** scales the data provided into internal coordinates and, by default, sets all parameters to one or to user-supplied values. For very difficult problems with widely spaced parameters, the program can estimate approximate starting parameters from the extremes of the data set, undertake a random search in an attempt to improve these, then perform a preliminary fit in the constrained L_1 to refine the starting estimates. It scales the model fitted so that the internal parameters will be of order unity at the start of the optimization, then performs weighted nonlinear regression by the quasi-Newton method with parameters constrained to be nonnegative. Even so, it must be realized that distinguishing order 2 from order 1 will usually be straightforward, distinguishing order 3 from order 2 will demand very accurate data over a wide range, and distinguishing order 4 from order 3 will usually only succeed if the data have special features.

The test files

Many of the test files distributed with the SimFIT package contain exact data. There are two reasons for this.

1. Exact data sets generated by program **makdat** have known parameters so, by fitting these it is possible to observe how accurately SimFIT can estimate the known parameters.
2. Such exact data sets can be corrupted by adding random error using program **adderr** to simulate experimental data.

The recommended procedure for analyzing your own data is to simulate using the suspected model and anticipated error, then fit to see how likely it will be for the fitting program to support your working hypothesis.

Example 1

In particular `rffit.tf6` contains exact data for a 4:4 rational function, and fitting 3:3 then 4:4 yields these parameter estimates.

Number	Parameter	Value	Std. Error	Lower95%cl	Upper95%cl	p	
1	α_0	1.0982E-05	6.0976E-06	-2.4385E-06	2.4403E-05	0.0991	*
2	α_1	2.0997E+00	3.8507E-04	2.0988E+00	2.1005E+00	0.0000	
3	α_2	2.4562E-04	1.4423E-03	-2.9289E-03	3.4201E-03	0.8679	*
4	α_3	1.0493E-01	7.4973E-05	1.0476E-01	1.0509E-01	0.0000	
5	α_4	1.0982E-13	9.3838E-08	-2.0654E-07	2.0654E-07	1.0000	*
6	β_1	4.0251E-04	7.3125E-04	-1.2070E-03	2.0120E-03	0.5930	*
7	β_2	1.0022E+00	1.5814E-03	9.9868E-01	1.0056E+00	0.0000	
8	β_3	1.0000E-11	1.2096E-05	-2.6623E-05	2.6623E-05	1.0000	*
9	β_4	2.4982E-03	1.9153E-06	2.4940E-03	2.5024E-03	0.0000	

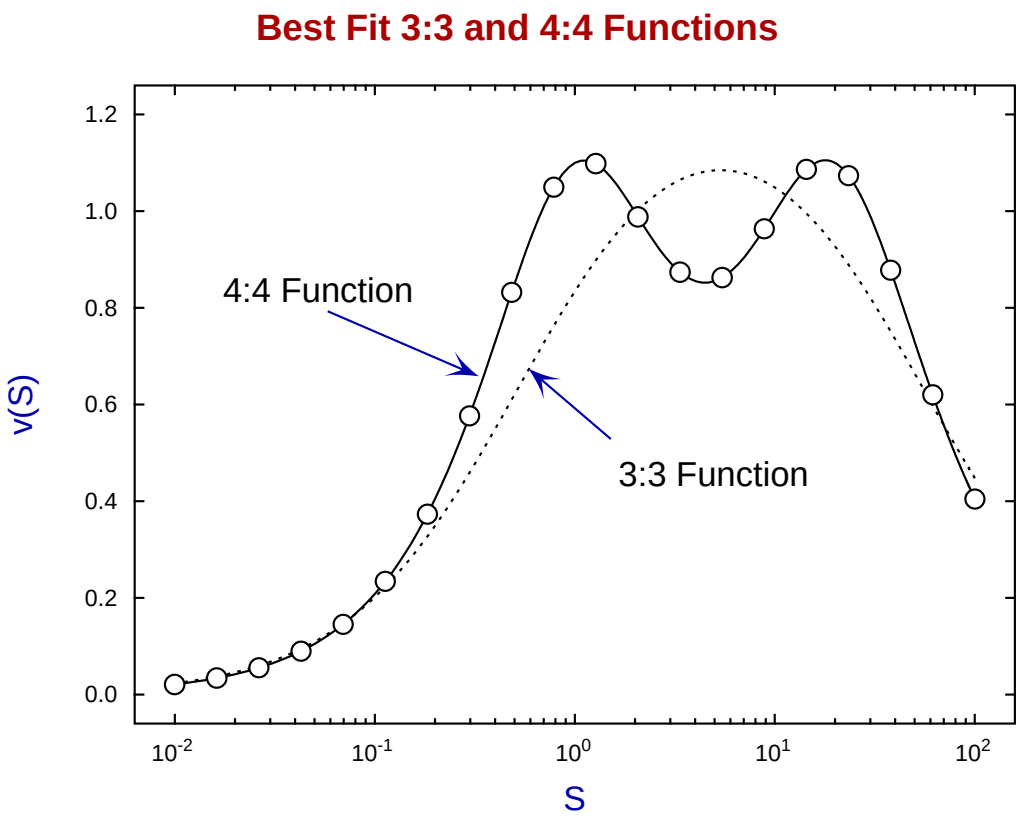
Now this test data was generated as the sum of two dead-end substrate inhibition models namely

$$v(S) = \frac{2S}{1+S^2} + \frac{0.1S}{1+0.0025S^2}$$
$$= \frac{2.1S + 0.105S^3}{1 + 1.0025S^2 + 0.0025S^4}$$

so, pointing out that numbers less than about 10^{-4} of the largest parameter estimate are effectively zero, the estimates are almost identical to the values used to generate the data, which supports the fact that program **rffit** is capable of determining the parameters of a 4:4 function given highly accurate and extensive data. The next table illustrates the strong support for the extra parameters in the 4:4 fit compared to the 3:3 fit, while the graph illustrates the reason: a positive rational function of order n:n can have at most $n-1$ positive turning points so the 3:3 function can never model the three positive turning points of the 4:4 function.

$WSSQ$-previous	7.8928E+01	
$WSSQ$-current	7.2902E-05	
Number of parameters-previous	7	
Number of parameters-current	9	
Number of S-values	20	
Akaike AIC-previous	4.1456E+01	
Akaike AIC-current	-2.3244E+02	ER = 2.9947E+59
Schwarz SC-previous	4.8426E+01	
Schwarz SC-current	-2.2348E+02	
Mallows C_p	1.1909E+07	C_p/M_1 = 1.7013E+06
Numerator degrees of freedom	2	
Denominator degrees of freedom	11	
F test statistic (FS)	5.9546E+06	
$P(F \geq FS)$	0.0000	
$P(F \leq FS)$	1.0000	
5% upper tail point	3.9823E+00	
1% upper tail point	7.2057E+00	

Conclusion based on F test
Reject previous model at 1% significance level
There is strong support for the extra parameters
Tentatively accept the current best fit model



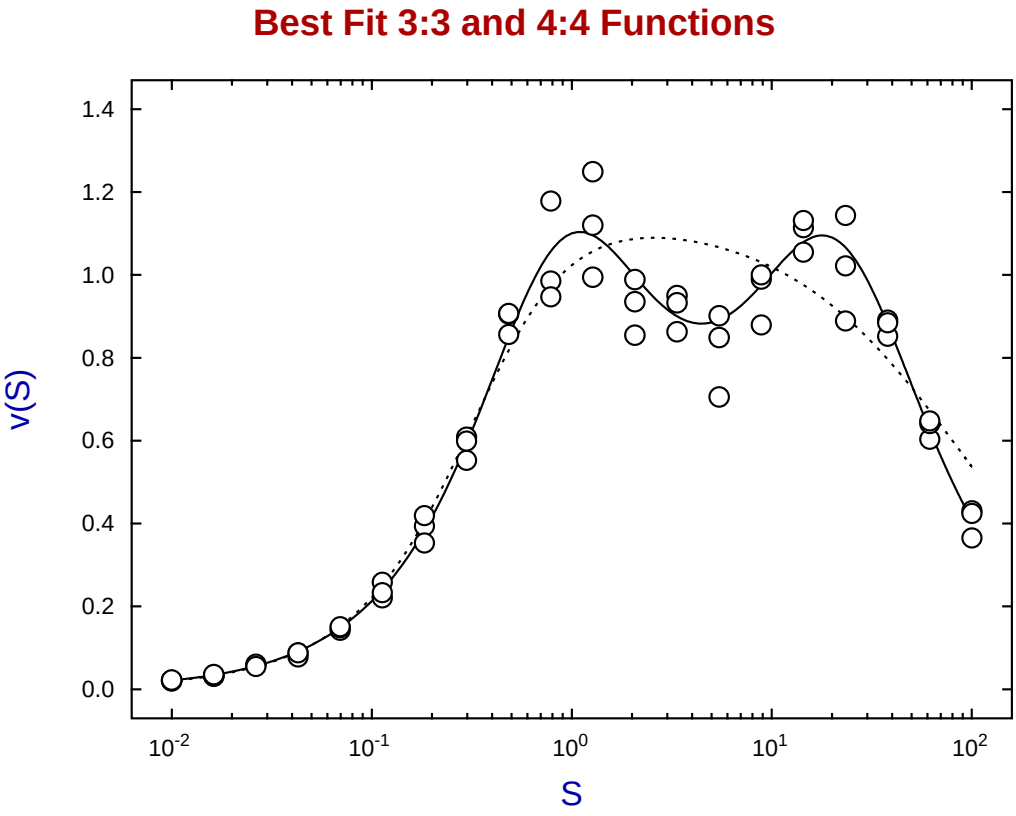
Note that the data are in a geometrical progression, that is, for k points x_i distributed equally on a log scale between end points A and B we would have $x_1 = A, x_2 = \lambda x_1, x_3 = \lambda x_2, \ldots, x_k = \lambda x_{k-1} = \lambda^{k-1} x_1 = B$

where $\lambda = (B/A)^{1/(k-1)}$. When fitting models like saturation functions, exponentials, or rational functions, a logarithmic scale is optimal for model discrimination and parameter determination [1].

Example 2

Test file `rffit.tf7` was obtained from `rffit.tf6` using **adderr** to generate triplicates with 7.5% relative error and standard errors calculated from replicates. Using starting estimates equal equal to one, but with α_0 and α_n suppressed gives reasonable parameter estimates as shown by the next table and graph.

Number	Parameter	Value	Std. Error	Lower95%cl	Upper95%cl	p	
2	α_1	2.1330E+00	2.7697E-02	2.0775E+00	2.1886E+00	0.0000	
3	α_2	2.1863E-01	3.1074E-01	-4.0464E-01	8.4189E-01	0.4848	*
4	α_3	1.0866E-01	2.0534E-02	6.7470E-02	1.4984E-01	0.0000	
6	β_1	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	1.0000	f
7	β_2	1.2320E+00	3.3648E-01	5.5711E-01	1.9069E+00	0.0006	
8	β_3	1.1784E-03	1.2893E-02	-2.4682E-02	2.7039E-02	0.9275	*
9	β_4	2.5835E-03	3.5474E-04	1.8719E-03	3.2950E-03	0.0000	



In general for n:n functions with $n > 2$ the following conclusions should be noted.

1. The data must be very accurate and span a large range of substrate concentration.
2. Data should be collected using a geometric progression between end points.
3. Program **rffit** must be run many times to compare the fit with random starting estimates with the fit from user-supplied starting estimates, or with all equal to one as with this example fitting `rffit.tf7`.
4. Model discrimination may work for $n \leq 3$ but parameter estimates may be poor for $n \geq 3$.

Theory

All kinetic schemes that can be devised for non-aggregating enzymes that have zero rate at zero substrate concentration lead to the following quasi steady state rate equation

$$v(S) = \frac{\alpha_1 S + \alpha_2 S^2 + \cdots + \alpha_n S^n}{\beta_0 + \beta_1 S + \beta_2 S^2 + \cdots + \beta_n S^n}$$

where $\alpha_i \geq 0$ and $\beta_i \geq 0$, and the only difference between models is the way that the coefficients α_i, β_i are expressed as functions of the rate constants. Some simple facts about this function are as follows.

- The order n can be as large as the subscript k in any enzyme substrate complexes ES_k.
- Given any order n with any nonnegative values for α_i, β_i it is possible to define a possible enzyme scheme with these values [2].
- The $v(S)$ curve is only sigmoidal if $\alpha_2\beta_0 > \alpha_1\beta_1$ [3].
- A order n positive rational function can have at most $n-1$ turning points and $2n-2$ points of inflexion in the first quadrant [4].
- If $\alpha_n = 0$ the $v(S)$ curve descends asymptotically to zero from a final turning point.
- If $\alpha_n > 0$ the $v(S)$ curve approaches a horizontal asymptote $v(\infty) = \alpha_n/\beta_n$.
- If $\alpha_n > 0$ the $v(S)$ curve descends from a final turning point if $\alpha_n\beta_{n-1} < \alpha_{n-1}\beta_n$.

As the main idea in fitting such a rational function to experimental data is to fix a minimum order n to suggest a possible enzyme mechanism, some facts about parameter estimation and discrimination should be noted.

- Any attempt to fit higher order models requires accurate data over a large range.
- Experimental points should have the S values in a geometric progression.
- Statistical techniques for model discrimination perform very well when distinguishing the case $n = 2$ from $n = 1$ but rapidly deteriorate for $n > 2$ [5],[6],[7].
- Fitting cases with $n > 2$ should be investigated by comparing the results with all starting estimates equal to one with those from several random searches.
- When the order n has been decided and good starting estimates have been located it is time to use the more advanced program **qnfit**.

Limiting cases

Before nonlinear regression became accepted as the only meaningful way to fit enzyme kinetic data, much use was made of fitting straight lines to extreme substrate concentrations in order to estimate parameter values. This has given rise to much confusion.

It might be thought that a satisfactory approximation to a n:n function at low substrate concentration would be the 1:1 Michaelis-Menten function

$$v(S) = \frac{\alpha_1 S}{\beta_0 + \beta_1 S}.$$

However this cannot be sigmoidal and $v(S)$ curves can be sigmoidal, so the best 1:1 function to approximate low substrate values is

$$v(S) = \frac{\alpha_1^2}{\alpha_1\beta_0 + (\alpha_1\beta_1 - \alpha_2\beta_0)S},$$

as this is the equation of the asymptotes approached at low substrate concentration in all the transformed plots such as double-reciprocal, Scatchard, etc. used in enzyme kinetics [8].

Irrespective of which technique is used to fit the data for low S the only parameters that can be estimated are the apparent kinetic constants

$$V_{max} = \alpha_1^2/(\alpha_1\beta_1 - \alpha_2\beta_0), \text{ and } K_m = \alpha_1\beta_0/(\alpha_1\beta_1 - \alpha_2\beta_0).$$

Again, it might be thought that a satisfactory approximation to a n:n function at high substrate concentration would be the n:n Hill equation

$$v(S) = \frac{\alpha_n S^n}{\beta_0 + \beta_n S^n}.$$

However $v(S)$ curves can have turning points and the Hill equation is monotonic, so the best 1:1 function to approximate high substrate values is

$$v(S) = \frac{\alpha_n^2}{(\alpha_n\beta_{n-1} - \alpha_{n-1}\beta_n) + \alpha_n\beta_n S},$$

as this is the equation of the asymptotes approached at high substrate concentration in all the transformed plots such as double-reciprocal, Scatchard, etc. used in enzyme kinetics [8].

Irrespective of which technique is used to fit the data for high S the only parameters that can be estimated are the apparent kinetic constants

$$V_{max} = \alpha_n/\beta_n, \text{ and } K_m = (\alpha_n\beta_{n-1} - \alpha_{n-1}\beta_n)/(\alpha_n\beta_n).$$

Another issue concerns reduction in degree by cancelation of common factors between numerator and denominator which can occur with some enzyme mechanisms containing cycles which, after using the principle of microscopic reversibility, lead to zero values for one or more of the Sylvester eliminants [9]. In the extreme case of factorization down to order 1:1, one possible expression for the reduced equation would be

$$v(S) = \frac{\alpha_1\alpha_n}{\alpha_n\beta_0 + \alpha_1\beta_n S}.$$

To the extent that fitting a Michaelis-Menten equation to a n:n function is justified because statistical evidence does not support $n > 1$, the two parameters that can be estimated are the apparent kinetic constants

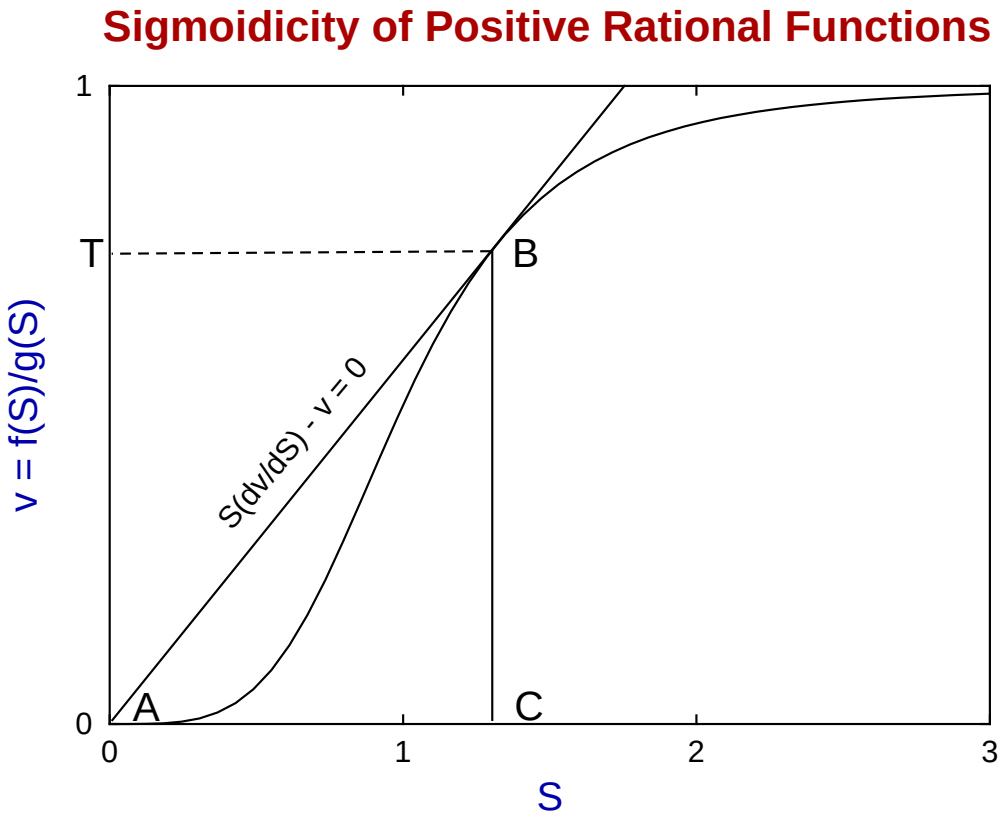
$$V_{max} = \alpha_n/\beta_n, \text{ and } K_m = \alpha_n\beta_0/(\alpha_1\beta_1).$$

Sigmoidicity

A positive rational function will be sigmoid if

$$\alpha_2\beta_0^2 - \alpha_1\beta_0\beta_1 - \alpha_0\beta_0\beta_2 + \alpha_0\beta_1^2 > 0,$$

but in the usual case $\alpha_0 = 0$ it is possible to define satisfactory measures of sigmoidicity, which can be explained by reference to the next figure [10].



The point labeled C is the first positive root of

$$S(dv/dS) - v = 0$$

and the point labeled $T = v(C)$ is the v coordinate where the tangent from the origin touches the curve. Consider then the expressions

$$\Delta_1 = \frac{T}{\max(v), \text{ for } S \geq 0}$$
$$\Delta_2 = \frac{Area(ABC)}{\int_0^C v(S)\, dS}$$

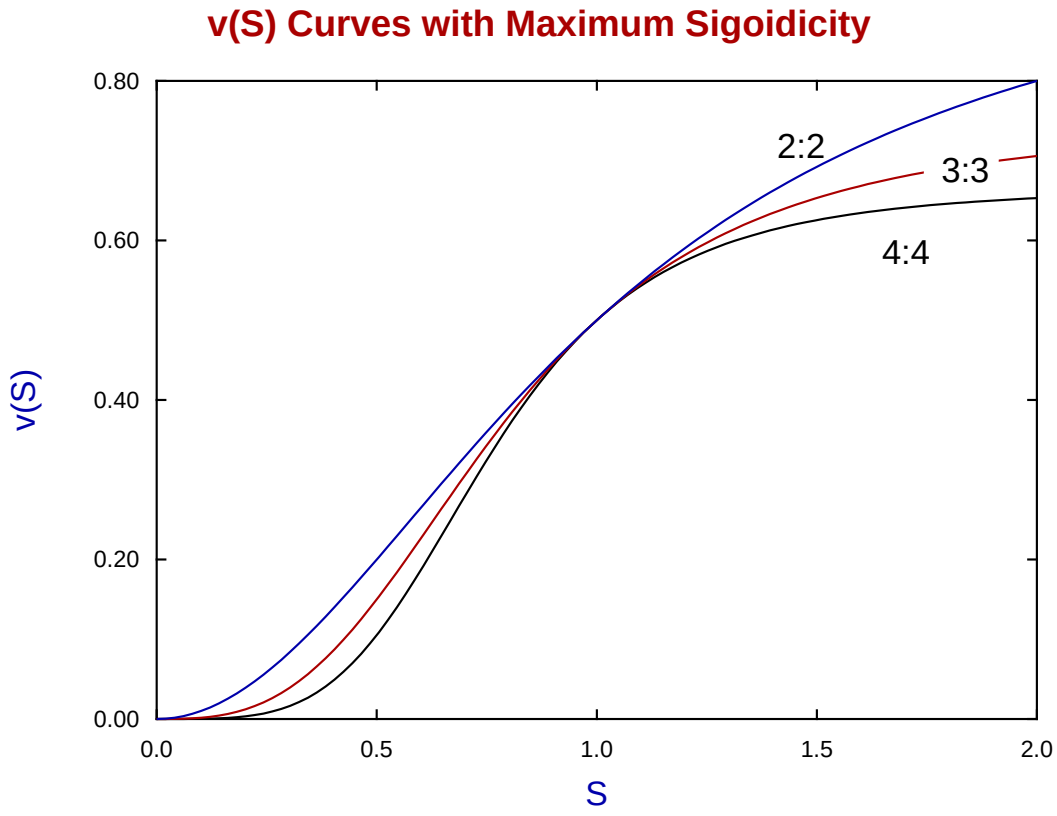
where Δ_1 and Δ_2 both increase as sigmoidicity increases. It can be shown that, for fractional saturation functions of order n, the following inequality applies

$$T \leq \frac{n-1}{n},$$

while, at least for the cases $n = 2, 3$, and $n = 4$ with some qualification, the positive rational function curve of maximum sigmoidicity is the equation

$$v = \frac{S^n}{1 + S^n}$$

shown in the next graph for the normalized functions $x^2/(1 + x^2)$, $(3/2)x^3/(1 + 2x^3)$, and $2x^4/(1 + 3x^4)$.



Substrate inhibition

Characterization of the shapes of rational functions in the first quadrant has been achieved for $n \leq 3$, and the most distinctive feature is the number of turning points [11]. All that can be said is that analysis depends on the signs and magnitudes of determinants such as

$$D_i = \begin{vmatrix} \alpha_i & \alpha_{i-1} \\ \beta_i & \beta_{i-1} \end{vmatrix}$$

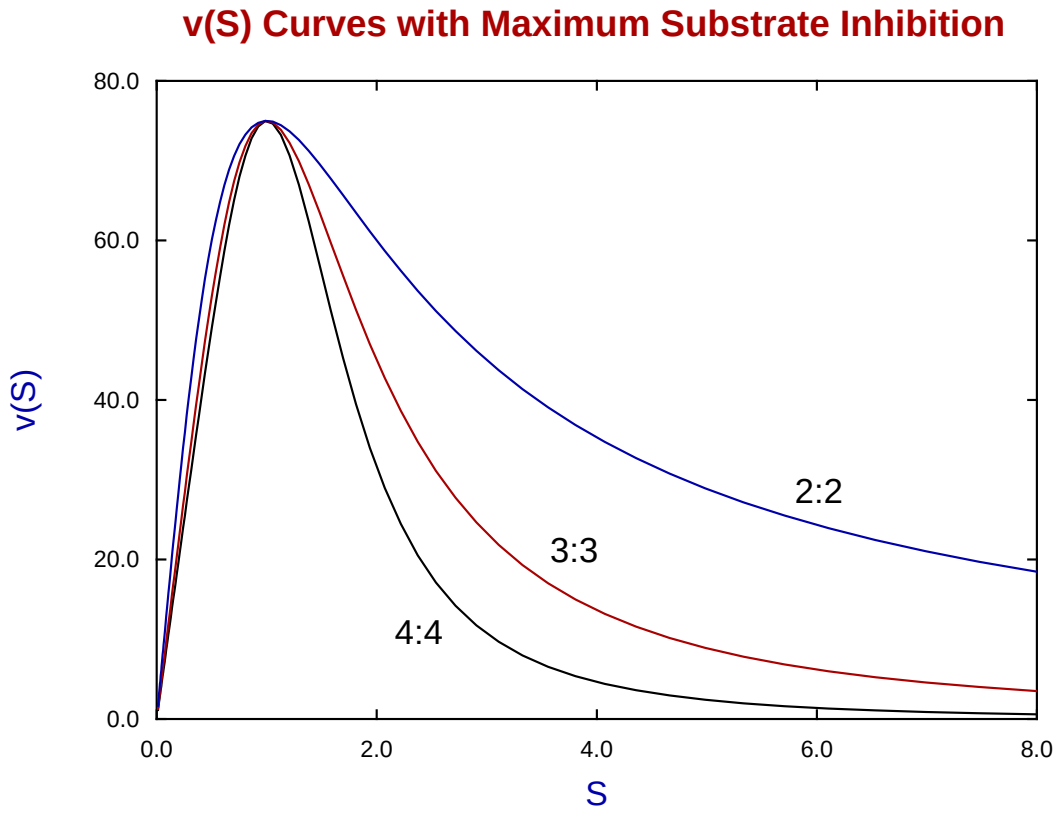
leading to the necessary and sufficient condition $D_n < 0$ for a final turning point, but only the necessary but not sufficient condition $D_i D_{i-1} < 0$ for the maximum number.

It is possible to establish a criterion for the maximum steepness of descent from a final maximum, which is the most commonly encountered type of substrate inhibition, and to present maximal examples for low degree cases. Unfortunately this is, like sigmoidicity, only possible for $n < 4$ and, with some reservations for $n = 4$. That is because, for $n > 4$ the number of possible curve shapes due to multiple turning points creates an intractable situation.

At least for the cases $n = 2, 3$ and $n = 4$ with some qualification, the positive rational function curve showing maximum substrate inhibition is the equation

$$v = \frac{S}{1 + S^n}$$

shown in the next graph for the normalized functions $150x/(1 + x^2)$, $225x/(2 + x^3)$, and $300x/(3 + x^4)$ [12].



References

[1] Optimal design for model discrimination using the F test with non-linear biochemical models. Criteria for choosing the number and spacing of experimental points.
Bardsley,W.G., McGinlay,P.B & Roig,M.G. (1989) *J. theor. Biol.* **139**, 85-102

[2] Simple enzyme kinetic mechanisms that can give all possible velocity profiles with chemically reasonable rate constant values.
Bardsley,W.G. (1983) *J. theor. Biol.* **104**, 485-491

[3] Sigmoid curves, non-linear double reciprocal plots and allosterism.
Bardsley,W.G. & Childs,R.E. (1975) *Biochem. J.* **149**, 313-328

[4] Critical points and sigmoidicity of positive rational functions.
Wood,R.M.W. & Bardsley,W.G. (1985) *Amer. Math. Month.* **92(1)**, 37-48

[5] The use of non-linear regression analysis and the F test for model discrimination with dose response curves and ligand binding data.
Bardsley,W.G. & McGinlay,P.B. (1987) *J. theor. Biol.* **126**, 183-201

[6] Conditions when statistical tests for model discrimination have high power. Some examples from pharmacokinetics, ligand binding, transient and steady-state enzyme kinetics.
Bardsley,W.G., McGinlay,P.B. & Roig,M.G. (1987) *Biophys. Chem.* **26**, 1-8

[7] Use of the F test for determining the degree of enzyme-kinetic and ligand-binding data. A Monte Carlo simulation study.
Burguillo,F.J., Wright,A.J. & Bardsley,W.G. (1983) *Biochem. J.* **211**, 23-34

[8] The determination of positive and negative co-operativity with allosteric enzymes and the interpretation of sigmoid curves and non-linear double reciprocal plots for the MWC and KNF models.
Bardsley,W.G. & Waight,R.D. (1978) *J. theor. Biol.* **70**, 135-156

[9] The reduction in degree of allosteric and other complex rate equations using Sylvester's dialytic method of elimination.
Bardsley,W.G. (1977) *J. theor. Biol.* **67**, 121-139

[10] A new approach to the measurement of sigmoid curves with enzyme kinetic and ligand binding data.
Bardsley,W.G. & Wright,A.J. (1983) *J. Mol. biol.* **165**, 163-182

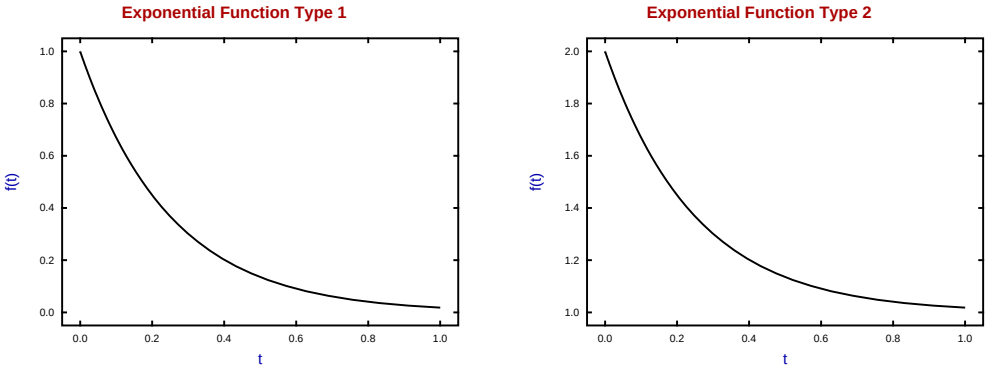
[11] The probability of obtaining complex kinetic curves for enzyme mechanisms with cubic terms in the pseudo-steady state rate equations.
Wardell,J.M., Bardsley,W.G., Kavanagh,J.P. & Wood,R.M.W. (1982) *J. theor. Biol.* **95**, 465-487

[12] Inhibition of enzymes by excess substrate. A theoretical and Monte Carlo study of turning points in $v(S)$ graphs.
Bardsley,W.G., Solano-Munoz,F., Wright,A.J. & McGinlay,P.B. (1983) *J. Mol. Biol.* **169**, 597-617

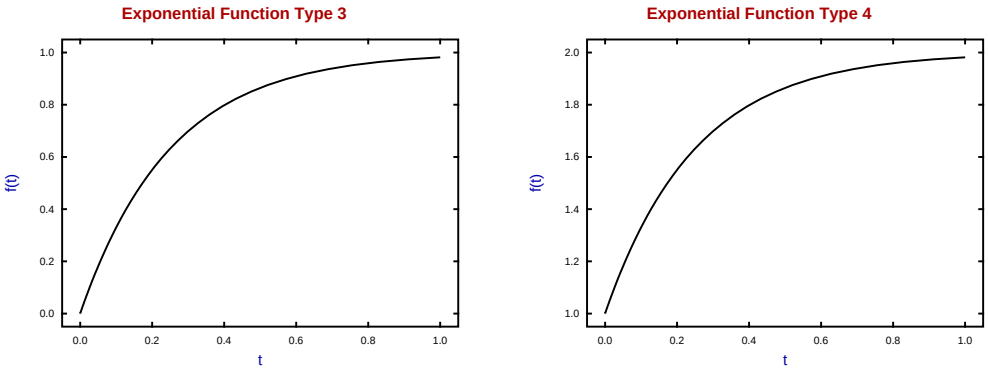
8.5.5 Fitting exponential functions

Exponential functions have wide applications in data analysis and the SimFIT package has a dedicated utility to fit six main categories of multi-exponential functions as illustrated below.

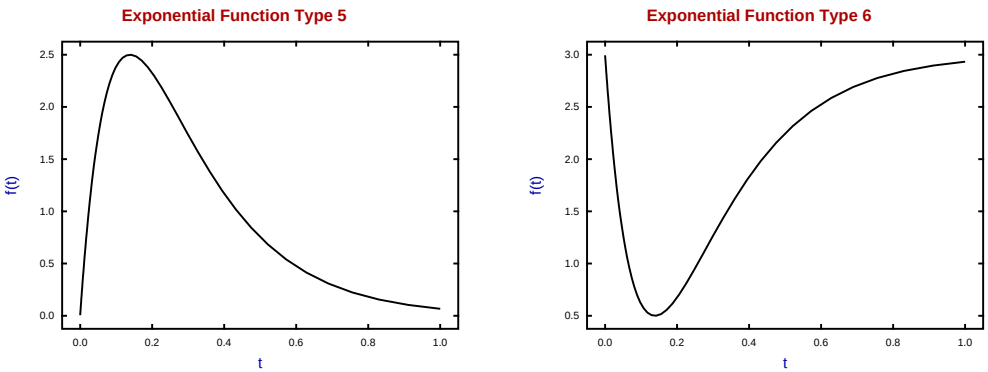
Simple exponential decline to zero (Type 1) or to a baseline (Type 2).



Monomolecular rise to a horizontal asymptote from zero (Type 3) or a baseline (Type 4).



Up-down (Type 5) or Down-up (Type 6) with at least two exponential terms.



In fact all of the exponential types are special cases of the following model

$$f(t) = A_1 \exp(-k_1 t) + A_2 \exp(-k_2 t) + \cdots + A_n \exp(-k_n t) + C$$

where the decisions facing the user are to define the category, i.e. the Type, and the number n of exponentials required.

There are three distinct SimFIT programs provided for fitting exponentials to data.

1. **exfit**
   This is a simple user-friendly interface to automatically scale the data, locate sensible starting estimates, and perform unconstrained weighted least squares fitting with goodness of fit analysis and model discrimination. It requires the data to be nonnegative, i.e. $f(t) \geq 0$ for $t \geq 0$, and also the time to start must be zero, i.e. $t = 0$, and t must be in nondecreasing order, as this is assumed when scaling and finding starting estimates. Additional parameters like the area under the curve (AUC) are calculated.

   Normally you should fit one then two exponentials, but in the case of Type 5 and Type 6 the lowest order to fit is two exponentials. After fitting Type 5 or Type 6 it is recommended to do a further relaxation fit to allow for variations in the starting and asymptotic values in order to fit data such as pharmacokinetics after a bolus ingestion. It is possible to fit models with more than two exponentials but this may require several random starts or user-defined starting values for success. In any case, model discrimination with more than two exponentials is somewhat unpredictable as explained in the following publication.

   The F test for model discrimination with exponential functions.
   Bardsley,W.G., McGinlay,P.B. & Wright,A.J. (1986) *Biometrika* **73**, 501-508

2. **qnfit**
   This is a quasi-Newton constrained nonlinear regression program for more experienced analysts which requires user-defined starting estimates and parameter limits. It has the advantage that fitting does not necessarily require $f(t) \geq 0, t \geq 0$ or t in nondecreasing order and provides many options for setting starting estimates and parameter limits as well as numerous fine tuning possibilities.

3. **deqsol**
   This simulates and fits systems of nonlinear differential equations.

Example 1: Simple exponential decay (Type 1)

This will illustrate fitting the most frequently used exponential model, namely exponential decay from a positive value at $t = 0$ to a final zero asymptote as $t \to \infty$.

From the main SimFIT menu choose [A/Z] then open program **exfit** and read the default test file provided which is `exfit.tf4`. If you choose to start with a model with one exponential the program will first fit the equation

$$f_1(t) = A_1 \exp(-k_1 t)$$

giving goodness of fit criteria, and if you select to fit up to order 2 the program will then fit the model

$$f_2(t) = A_1 \exp(-k_1 t) + A_2 \exp(-k_2 t)$$

giving information to compare these two fits, followed by the option for graphical deconvolution of $f_2(t)$ to illustrate the relative contribution of the two exponential terms.

The data, results tables, and graphs follow but note that the subscripts in equation $f_2(t)$ are arbitrary so, to preserve uniformity, parameters are rearranged if necessary after fitting so that the amplitudes A_i are in nondecreasing order.

t	$f(t)$	se
0.035983	1.7440	0.048730
0.035983	1.8367	0.048730
0.035983	1.8164	0.048730
0.054896	1.7028	0.033089
0.054896	1.6480	0.033089
0.054896	1.7075	0.033089
0.083750	1.6290	0.060314
0.083750	1.5359	0.060314
0.083750	1.6490	0.060314
0.12777	1.3919	0.013361
0.12777	1.3676	0.013361
0.12777	1.3702	0.013361
0.19493	1.1454	0.089321
0.19493	1.2240	0.089321
0.19493	1.3237	0.089321
0.29739	0.99897	0.043211
0.29739	0.94038	0.043211
0.29739	0.91466	0.043211
0.45370	0.80103	0.047423
0.45370	0.70902	0.047423
0.45370	0.73507	0.047423
0.69217	0.53660	0.041121
0.69217	0.50323	0.041121
0.69217	0.58501	0.041121
1.0560	3.8157	0.023248
1.0560	3.4769	0.023248
1.0560	3.9221	0.023248
1.6110	1.8573	0.011054
1.6110	1.9103	0.011054
1.6110	2.0697	0.011054

The columns contain data in the following format.

1. **Column 1**: the non–negative time t which must be in non-decreasing order.
2. **Column 2**: the non–negative response $f(t)$ presumed to be dependent on time in column 1.
3. **Column 3**: the positive sample standard deviation of the replicate response measurements. Note that column 3 can be omitted or set to 1 if unweighted regression is required.

The results from fitting two exponentials are as follows.

Parameter	Value	Std. error	Lower95%cl	Upper95%cl	p
A_1	0.85255	0.067715	0.71336	0.99174	0.0000
A_2	1.1764	0.074759	1.0228	1.3301	0.0000
k_1	6.7934	0.85439	5.0372	8.5496	0.0000
k_2	1.1121	0.051102	1.0070	1.2171	0.0000
AUC	1.1834	0.014710	1.1532	1.2136	0.0000

AUC is the area under the curve from $t = 0$ to $t = \infty$
Initial time point (A) = 0.035983
Final time point (B) = 1.6110
Area (from $t = A$ to $t = B$) = 0.93832
Average over range (A, B) = 0.59575

Parameter correlation matrix

1			
-0.8757	1		
-0.5963	0.8996	1	
-0.8479	0.9485	0.8199	1

Analysis of residuals: $WSSQ$	24.397
$P(\chi^2 \geq WSSQ)$	0.5533
R^2, cc(theory,data)^2	0.9934
Largest absolute relative residual	11.99%
Smallest absolute relative residual	0.52%
Average absolute relative residual	3.87%
Absolute relative residuals in range 0.1–0.2	3.33%
Absolute relative residuals in range 0.2–0.4	0.00%
Absolute relative residuals in range 0.4–0.8	0.00%
Absolute relative residuals > 0.8	0.00%
Number of negative residuals (n_1)	15
Number of positive residuals (n_2)	15
Number of runs observed (r)	16
P(runs $\leq r$: given n_1 and n_2)	0.5759
5% lower tail point	11
1% lower tail point	9
P(runs $\leq r$: given n_1 plus n_2)	0.6445
P(signs $\leq$ least number observed)	1.000
Durbin-Watson test statistic	1.8061
Shapiro-Wilks W statistic	0.9387
Significance level of W	0.0841
Akaike AIC (Schwarz SC) statistics	1.7979 (7.4027)

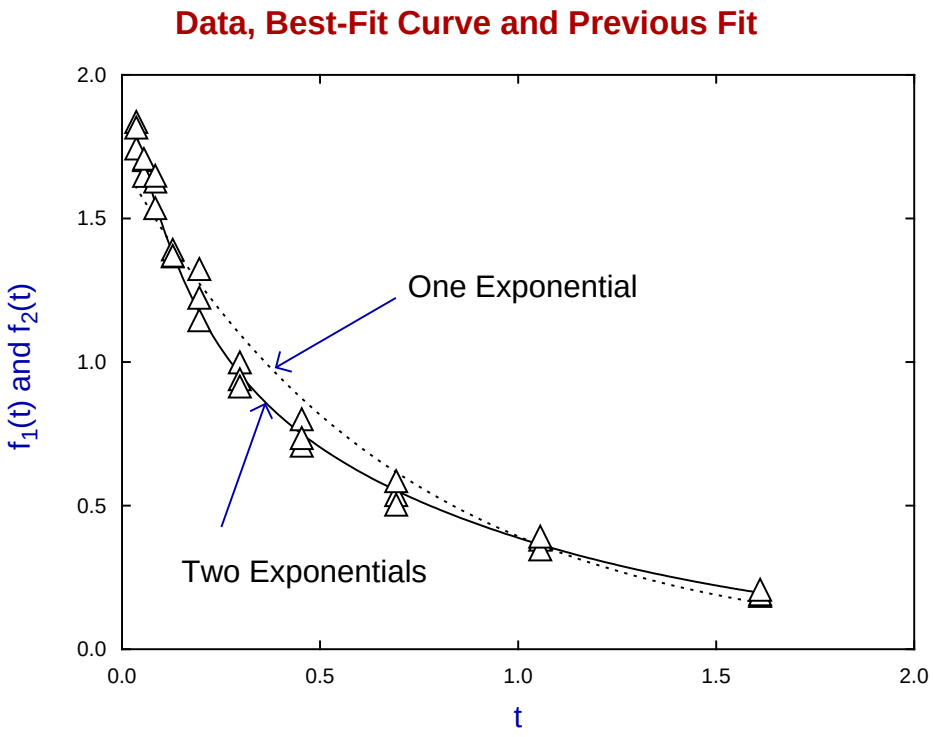
Verdict on goodness of fit: incredible

$WSSQ$-previous	224.9
$WSSQ$-current	24.4
Number of parameters-previous	2
Number of parameters-current	4
Number of x-values	30
Akaike AIC-previous	64.44
Akaike AIC-current	1.798, $ER = 3.998E + 13$
Schwarz SC-previous	67.24
Schwarz SC-current	7.403
Mallows C_p	213.7, $C_p/m_1 = 106.9$
Numerator degrees of freedom	2
Denominator degrees of freedom	26
F test statistic (FS)	106.9
$P(F \geq FS)$	0.0000
$P(F \leq FS)$	1.0000
5% upper tail point	3.369
1% upper tail point	5.526

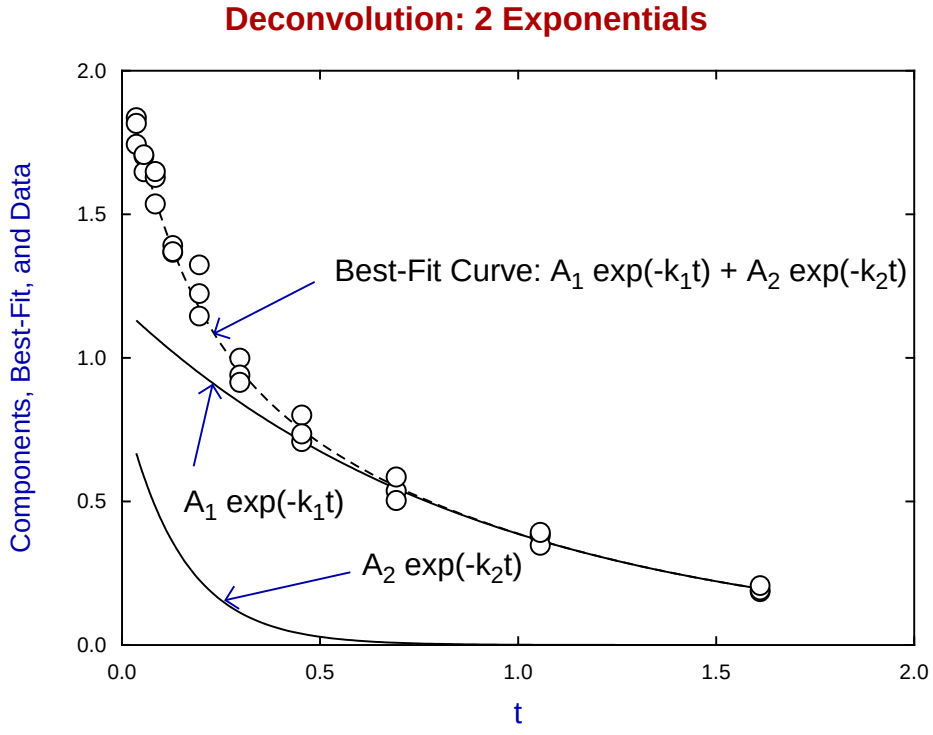
Conclusion based on the F test

Reject the previous model at 1% significance level
There is strong support for the extra parameters
Tentatively accept the current best fit model

The fit with two exponentials is clearly better than the fit with one exponential as show in the next graph.



From the next plot it is evident that both components make a significant contribution to the best fit double exponential curve.



Example 2: sequential kinetics (Type 5)

Another scheme that is frequently encountered is with an irreversible chemical reaction such as

$$X \xrightarrow{k_1} Y \xrightarrow{k_2} Z$$

with rate constants k_1 and k_2 and initial conditions $X(0) = X_0, Y(0) = Y_0, Z(0) = Z_0$ leading to

$$\begin{aligned}
X(t) &= X_0 \exp(-k_1 t) \\
Y(t) &= \frac{X_0 k_1}{k_2 - k_1} \exp(-k_1 t) + \left[Y_0 - \frac{X_0 k_1}{k_2 - k_1} \right] \exp(-k_2 t) \\
Z(t) &= X_0 + Y_0 + Z_0 - \frac{X_0 k_2}{k_2 - k_1} \exp(-k_1 t) - \left[Y_0 - \frac{X_0 k_1}{k_2 - k_1} \right] \exp(-k_2 t).
\end{aligned}$$

In the special case $X_0 > 0, Y_0 = 0, Z_0 = 0$ the expression for $Y(t)$ reduces to

$$\begin{aligned}
Y(t) &= \frac{X_0 k_1}{k_2 - k_1} [\exp(-k_1 t) - \exp(k_2 t)] \\
&= X_0 k t \exp(-k t) \text{ if } k = k_1 = k_2.
\end{aligned}$$

A similar expression is often encountered in pharmacokinetics, for instance if $Y(t)$ is the concentration of a substance in the blood after it ingested at $t = 0$, then absorbed from the stomach with rate constant k_1 but eliminated from the system with rate constant k_2. However several complications of this simple scheme are often encountered.

1. There may be insufficient data to characterize the early data points.
2. There may be insufficient time to record the final data points.
3. Data may arise from a repeated experiment with insufficient time for complete washout.
4. $Y(0)$ and/or $Z(0)$ may not be zero.
5. There may be additional steps requiring additional exponential terms.
6. Instead of rising then falling as in Type 5 the observed response may be a decrease followed by an increase as with Type 6.

Program **exfit** can attempt to deal with such issues by allowing additional exponential terms to be added and by allowing a relaxation phase to follow an initial fitting phase. For instance, this simplified three parameter model $g(t)$ can be fitted first

$$g(t) = A[\exp(-k_1 t) - \exp(k_2 t)]$$

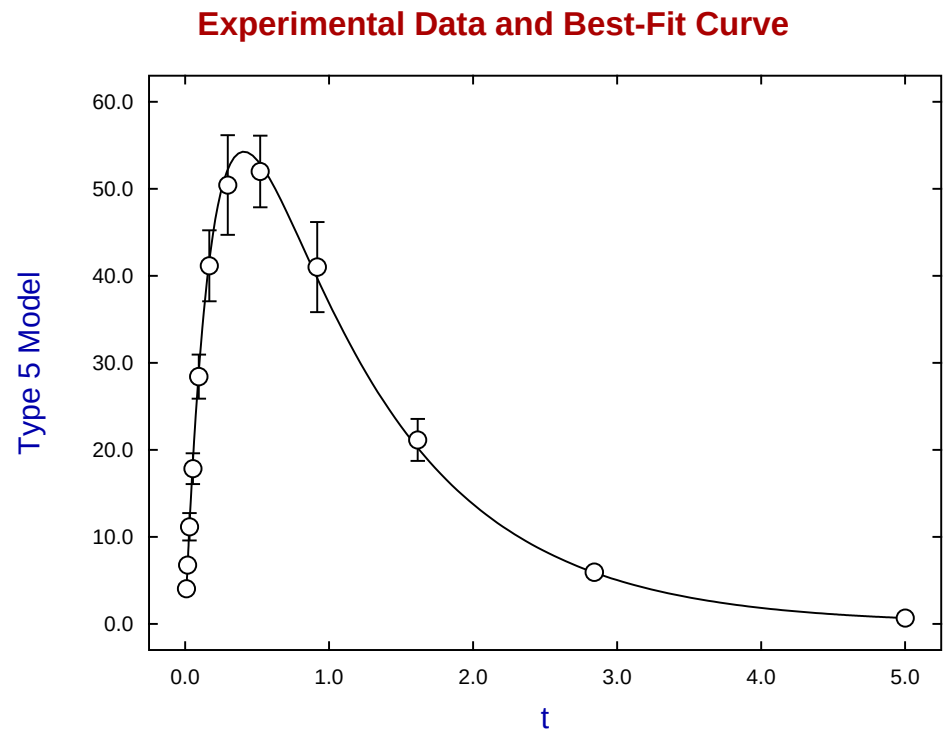
followed by using the parameter estimates from this fit as starting estimates to fit the richer four parameter model $h(t)$

$$h(t) = A_1 \exp(-k_1 t) + A_2 \exp(-k_2 t)$$

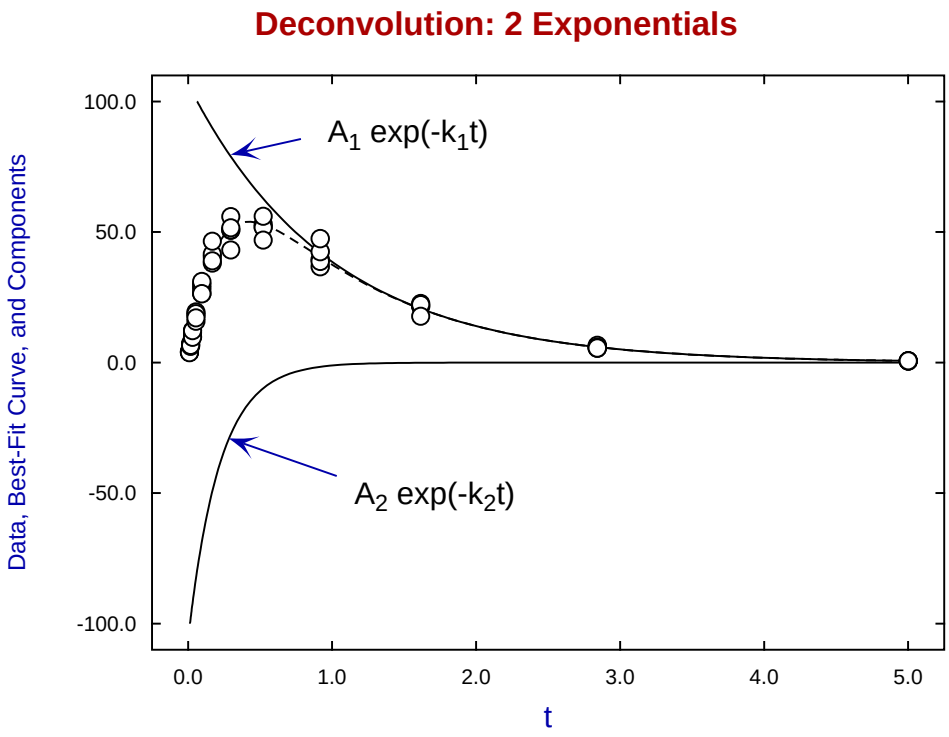
which will attempt to retain the up-down character supposed in the data. This relaxation method should be used when fitting Type 5 and Type 6 exponential models. However, in order to deal with models such as these it is best to use the greater versatility of program **qnfit**.

To illustrate fitting such a model the exact data in test file `exfit.tf5` was input into SIMFIT program **adderr** and five replicates per point were generated with 7.5% relative error with weights calculated from the sets of five replicates at each time point.

The simulated data and best-fit Type 5 curve from program **exfit** was as follows.



The graphical deconvolution of the best fit Type 5 model is shown next.



Theory

SimFIT program **exfit** will attempt to fit the multi-exponential model

$$f(t) = \sum_{i=1}^{n} A_i \exp(-k_i t) + C$$

by scaling the data, calculating starting estimates, then performing unconstrained nonlinear regression. It will only succeed if the data are extensive and accurate over a wide time range, the absolute values of the amplitudes A_i are similar, the rate constants k_i are sufficiently distinct, and the value of n modest, say $n \leq 3$. For more extreme conditions it may be necessary to input starting estimates interactively, or preferably use the advanced programs **qnfit** or **deqsol**.

The reason why users have to choose which of the six exponential types to fit is that the scaling, calculation of starting estimates, and parameterization of the model has considerable influence on the success of optimization and model discrimination. To appreciate this, consider a system where several irreversible first order processes are taking place as in this scheme.

$$\begin{aligned} X_1 &\xrightarrow{k_1} Y_1 \\ X_2 &\xrightarrow{k_2} Y_2 \\ &\dots \\ X_n &\xrightarrow{k_n} Y_n. \end{aligned}$$

For each independent component we would have the solutions

$$\begin{aligned} X_i &= X_{i0} \exp(-k_i t) \\ Y_i &= Y_{i0} + X_{i0}(1 - exp(-kt)) \end{aligned}$$

so that fitting a model to $\sum_{i=1}^{n} X_i(t)$ would require a Type 1 model, while fitting $\sum_{i=1}^{n} Y_i(t)$ would require a Type 4 model or a Type 3 model if $Y_{i0} = 0$.

As explained in Example 2, a consecutive scheme like

$$X_1 \xrightarrow{k_1} X_2 \xrightarrow{k_2} X_3 \dots \xrightarrow{k_n} X_n$$

would also lead to a different type of solution with n exponentials of Type 5 or Type 6.

The actual experimental situation could be further complicated for these reasons.

- Most processes fitted by exponential models are not irreversible but also involve backward flux.
- Reversible consecutive processes require $n \geq 2$ but here the exponential terms are not simple but involve calculating the eigenvalues for the system at each iteration.
- Cyclical consecutive processes can also lead to complex eigenvalues and oscillating solutions.

In such situations it is far better to model the situation as a set of differential equations and simulate then fit these using SimFIT program **deqsol**.

8.5.6 Fitting growth, decay, or survival models

Nonlinear growth, decay, and survival models are fitted to data in order to estimate parameters that can be used to compare the effects of treatments and/or different groups. The parameters that are usually estimated are the initial and final sizes and rates of change and meaningful numbers such as the half life and maximum rates of change.

Example 1: Growth data

From the main SIMFIT menu choose [A/Z], open program **gcfit**, select the option to fit growth curves then browse the default test file `gcfit.tf2` containing the following data.

Time	Size	Standard Error
0.0000	0.090501	0.0057406
0.0000	0.085148	0.0057406
0.0000	0.096621	0.0057406
1.0000	0.20400	0.0069302
1.0000	0.21300	0.0069302
1.0000	0.21763	0.0069302
2.0000	0.42858	0.017410
2.0000	0.45530	0.017410
2.0000	0.42261	0.017410
3.0000	0.71832	0.039573
3.0000	0.64283	0.039573
3.0000	0.70118	0.039573
4.0000	0.84408	0.041097
4.0000	0.76262	0.041097
4.0000	0.79382	0.041097
5.0000	0.91559	0.027506
5.0000	0.86060	0.027506
5.0000	0.88937	0.027506
6.0000	0.98545	0.016132
6.0000	o.95853	0.016132
6.0000	0.98738	0.016132
7.0000	1.0552	0.063158
7.0000	0.94115	0.063158
7.0000	1.0452	0.063158
8.0000	1.0433	0.040631
8.0000	0.96285	0.040631
8.0000	1.0130	0.040631
9.0000	0.99185	0.047010
9.0000	1.0452	0.047010
9.0000	1.0856	0.047010
10.000	1.0226	0.018664
10.000	0.98858	0.018664
10.000	0.99220	0.018664

Column 1 contains the time values t which must be nonnegative and in nondecreasing order.
Column 2 contains the size estimates $S(t)$ which must be nonnegative.
Column 3 contains the sample standard deviations for the triplicates to use for weighting, but this column can be set to one or omitted if weighting is not required.

Program **gcfit** can fit sequences of nonlinear growth, decay, or survival models giving statistics for goodness of fit and model discrimination but, before proceeding further, the definition of $S(t)$ must be explained.

If the data are for longitudinal measurements on the same individual or subjects they will be correlated so that fitting nonlinear models by weighted least squares will generate biased fits instead of maximum likelihood fits. One way to circumvent this is to fit flexible models such as polynomials or splines by techniques that attempt to estimate the autocorrelation. However it is only possible to estimate approximate correlations and polynomials cannot capture the shape of actual growth data or be used to estimate meaningful parameters to characterize growth profiles.

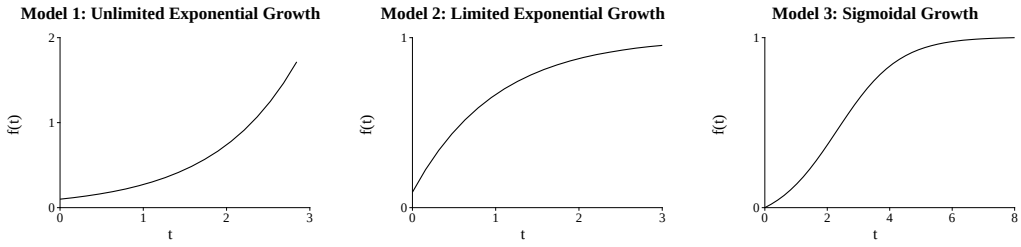
Ideally, **gcfit** should be used where observed $S(t)$ values are obtained in such a way as to make successive observations independent, e.g. sampling without replacement to estimate growth of bacterial colonies. Users will have to balance the usefulness of growth curve models with possible bias induced by fitting a deterministic model against the model-free data smoothing approach.

First of all note that most simple growth curve models are special cases of differential equations such as the Von Bertalannfy allometric equation

$$dS/dt = AS^{\alpha} - BS^{\beta}$$

which can be simulated and fitted using program **deqsol**, and it usual to explore the type of model required by fitting the first three models provided by program **gcfit** in a preliminary investigation. When a model has been selected there will be no further need to fit sequences of models.

Three typical growth curve shapes are shown in the next figure.



- **Model 1**
  This is exponential growth $S_1(t)$ which is only encountered in the early phase of development.

$$S_1(t) = A_1 \exp(k_1 t)$$

- **Model 2**
  This is limited exponential growth $S_2(t)$, concave down to an asymptote fitted by the monomolecular model

$$S_2(t) = A_2[1 - \exp(-k_2 t)]$$

- **Model 3**
  This is the logistic equation $S_3(t)$ which can fit sigmoidal profiles.

$$S_3(t) = \frac{A_3}{1 + B\exp(-k_3 t)}$$

Proceeding to fit these three models sequentially leads to the following conclusions and results table for model three, then a plot of data and all three best fit curves.

Model	$WSSQ/NDOF$	$P(\chi^2 \geq W)$	$P(R \leq r)$	$N > 10\%$	$N > 40\%$	$Av.r\%$	Verdict
1	152	0.000	0.000	29	17	40.03	Very bad
2	18.1	0.000	0.075	20	0	12.05	Very poor
3	1.32	0.113	0.500	0	0	3.83	Incredible

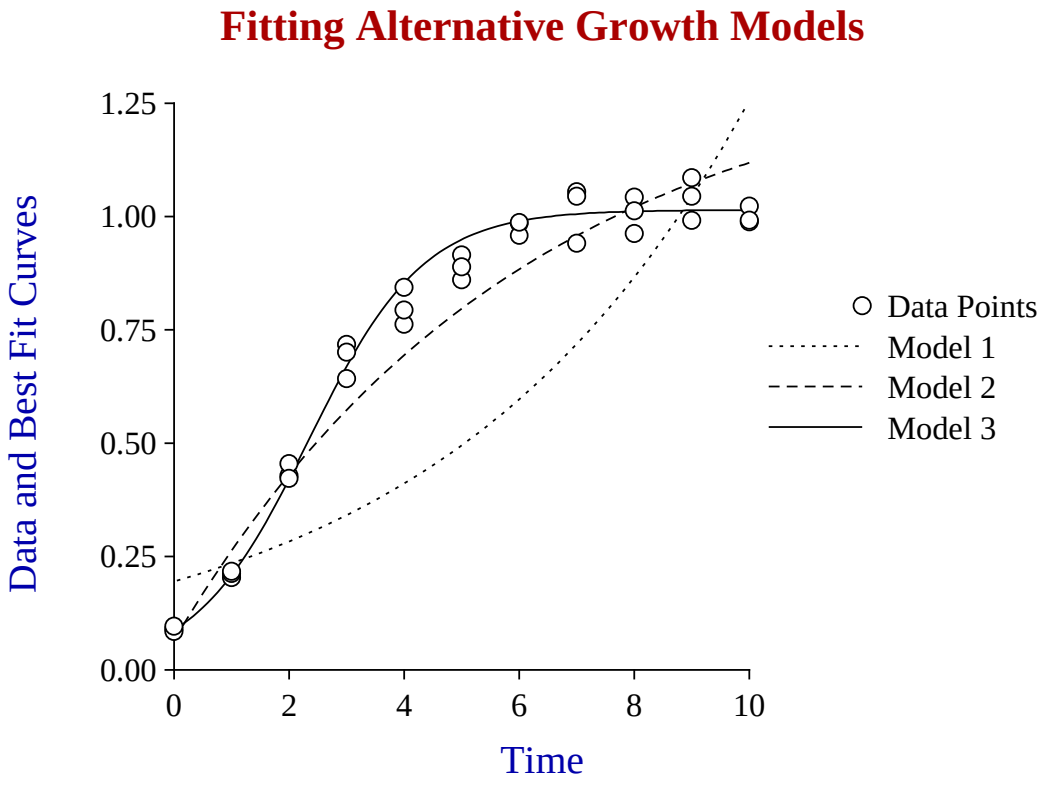
In this table $WSSQ/NDOF$ is the weighted sum of squares divided by degrees of freedom and $P(\chi^2 \geq W)$ is the significance level for this parameter in a chi-square test. $P(R \leq r)$ is the probability of runs less than or equal to the number observed given the number of positive and negative residuals, while $N > 10\%$ and $N > 40\%$ are the number of absolute residuals exceeding the stated percentage, and $Av.r\%$ is the average absolute residual. The conclusions in the last column are based on these results along with several other goodness of fit measures, and clearly model 3 is the preferred model with the estimated parameters shown next.

Results for model 3

Parameter	Value	Std.error	Lower95%cl	Upper95%cl	p
A	0.99891	0.0078551	0.9828 7	1.0150	0.0000
B	9.8901	0.33300	9.2100	10.570	0.0000
k	0.98814	0.026785	0.93344	1.0428	0.0000
$t_{1/2}$	2.3190	0.045070	2.2270	2.4111	0.0000

Parameter correlation matrix

1		
-0.0167	1	
-0.4388	0.7192	1



Example 2: Decay data

It is often useful to fit growth models to data that are decreasing as a function of time instead of increasing. For instance, the Gompertz model growth data in the next table are contained in test file `gcfit.tf5` while the same data are arranged into decay form in test file `gcfit.tf6`.

Growth data		Decay data	
t	$S(t)$	t	$S(t)$
0.0000	0.0048	0.0000	97.3685
1.1110	0.3044	1.1110	96.4062
2.2220	3.4696	2.2220	82.1162
3.3330	14.5225	3.3330	74.1991
4.4440	40.8277	4.4440	52.6928
5.5560	52.6928	5.5560	40.8277
6.6670	74.1991	6.6670	14.5225
7.7780	82.1162	7.7780	3.4696
8.8890	96.4062	8.8890	0.3044
10.0000	97.3685	10.0000	0.0048

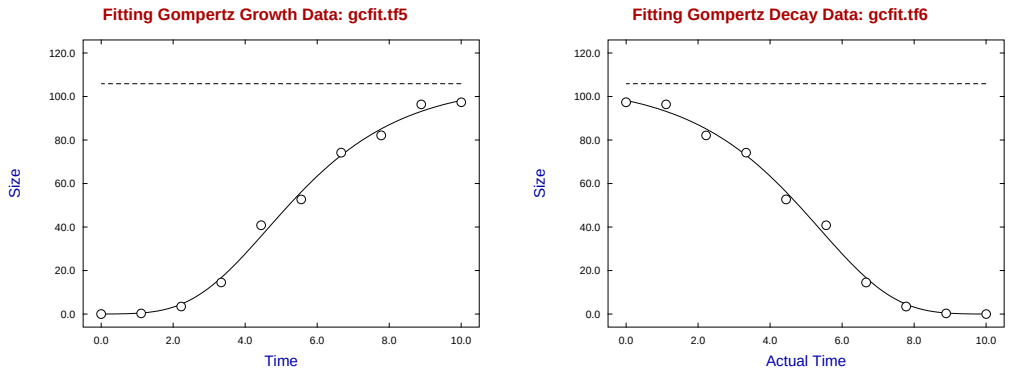
What happens in program **gcfit** when data in `gcfit.tf6` are analyzed is that the data are rearranged into the order of `gcfit.tf5` and then fitted by growth models as normal, except that some of the results and graphs are displayed in the original decay order with the original time scale.

First consider the parameters estimated for data in test file `gcfit.tf5`.

Parameter	Value	Std.error	Lower95%cl	Upper95%cl	p
A	105.87	4.6415	94.898	116.85	0.0000
B	9.1665	1.9138	4.6412	13.692	0.0020
k	0.48054	0.052253	0.35698	0.60410	0.0000
$t_{1/2}$	5.3733	0.20578	4.8867	5.8599	0.0000

Now consider the parameters estimated for data in test file `gcfit.tf6` and the best fit curves.

Parameter	Value	Std.error	Lower95%cl	Upper95%cl	p
A	105.87	4.6415	94.898	116.85	0.0000
B	9.1665	1.9138	4.6412	13.692	0.0020
k	0.48054	0.052253	0.35698	0.60410	0.0000
$t_{1/2}$	4.6267	0.20578	4.1401	5.1133	0.0000



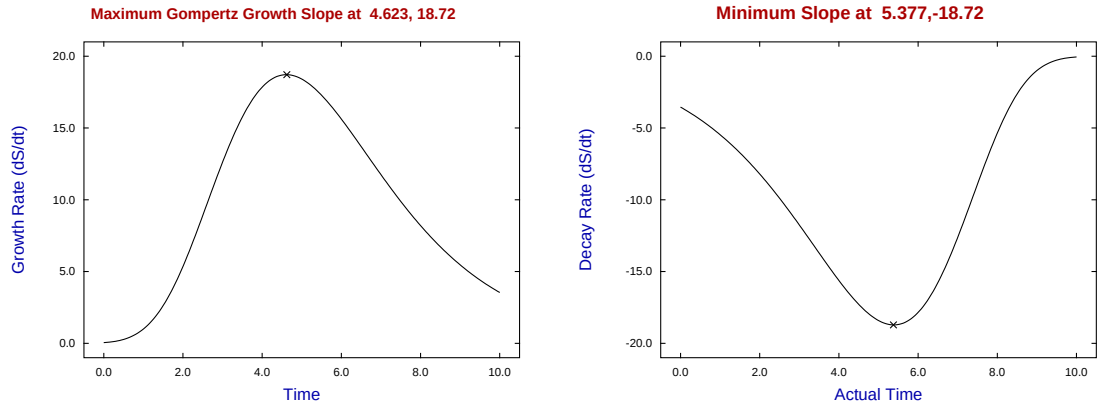
What has happened is that the following model was fitted to both of these data sets

$$S(t) = A\exp[-B\exp(-kt)]$$

but the only difference in the parameter estimates and graphs is that the data are presented in the original time scale for $t_{1/2}$ for the decay data and not using the transformed time $T = t_{max} + t_{min} - t$.

A similar situation is encountered when comparing the maximum and minimum slopes evaluated at the data points with the maximum and minimum values evaluated along the coordinates of the best fit curves. This is illustrated by the results displayed for extreme gradients in the extracts from the results log files and further clarified by the gradient plots.

Results for growth data			
Maximum observed growth rate	18.655	Best fit curve maximum	18.716
Time when max. rate observed	4.4440	Best fit curve time	4.6107
Minimum observed growth rate	0.048725	Best fit curve minimum	0.048725
Time when min. rate observed	0.0000	Best fit curve time	0.0000
Results for decay data			
Minimum observed growth rate	-18.655	Best fit curve minimum	-18.716
Time when min. rate observed	5.5560	Best fit curve time	5.3893
Maximum observed growth rate	-0.048725	Best fit curve maximum	-0.048725
Time when max. rate observed	10.000	Best fit curve time	10.000



The conclusion is simply that, if SimFIT program **gcfit** is supplied with decay data, the data will be reversed and fitted by the growth models, but the output tables and graphs will use the original decay coordinates.

Example 3: Nonlinear survival curves

In mode 2, **gcfit** fits a sequence of survival curves for data smoothing where it is assumed that the data are uncorrelated estimates of fractions surviving $0 \leq S(t) \leq 1$ as a function of time $t \geq 0$, e.g. such as would result from using independent samples for each time point. However, as normalizing data to $S(0) = 1$ can introduce bias, mode 2 allows an amplitude factor to be estimated.

It is important to realize that, if any censoring has taken place, the estimated fraction should be corrected for this. In other words, you start with a population of known size and, as time elapses, you estimate the fraction surviving by any sampling technique that gives estimates corrected to the original population at time zero.

The test files `weibull.tf1` and `gompertz.tf1` contain some exact data, which you can fit to see how mode 2 works. Then you can add error to simulate reality using program **adderr**. Note that you prepare your own data files for mode 2 using the same format as for program **makfil**, making sure that the fractions are between zero and one, and that only nonnegative times are allowed. It is probably best to do unweighted regression with this sort of data unless the variance of the sampling technique has been investigated independently.

In survival mode the time to half maximum response is estimated with 95% confidence limits and this can used to estimate $LD50$. The survivor function is $S(t) = 1 - F(t)$, the pdf is $f(t)$, i.e. $f(t) = -dS/dt$, the hazard function is $h(t) = f(t)/S(t)$, and the cumulative hazard is $H(t) = -\log(S(t))$. Plots are provided for $S(t), f(t), h(t), \log[h(t)]$ and, as in mode 1, a summary is given to help choose the best fit model from the models provided, all of which decrease monotonically from $S(0) = 1$ to $S(\infty) = 0$ with increasing time.

The test file `weibull.tf1` has the following data

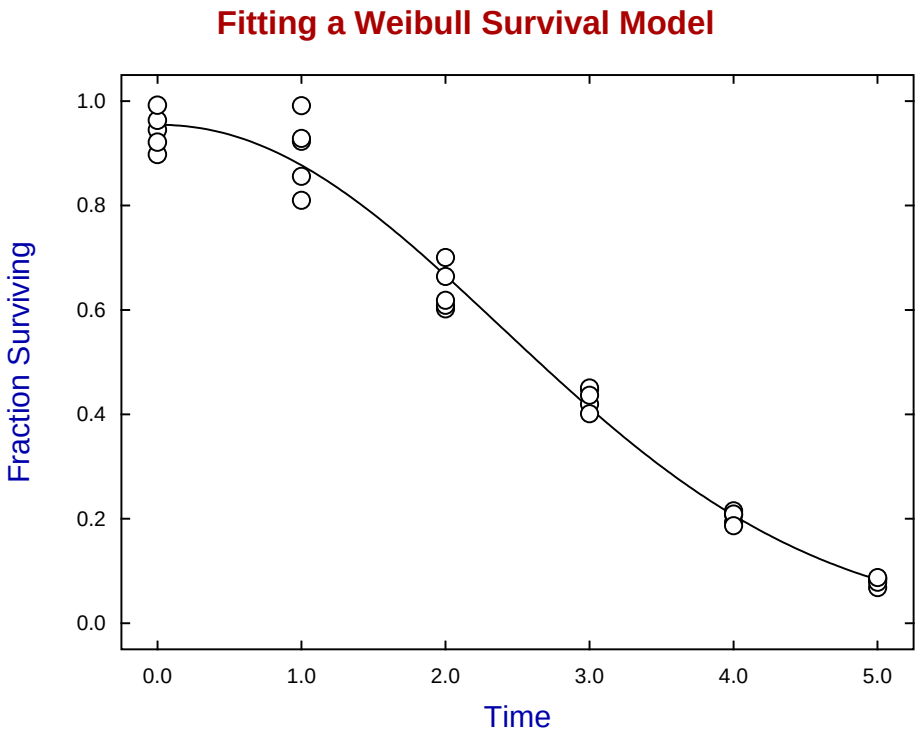
Time	Fraction	s.e.
0.000	1.000	1
1.000	0.9048	1
2.000	0.6703	1
3.000	0.4066	1
4.000	0.2019	1
5.000	0.08208	1

simulated by program **makdat** using the Weibull model

$$\begin{aligned} S(t) &= p_1 \exp[-p_2 t^{p_3}] \\ &= S_0 \exp[-(At)^B] \end{aligned}$$

for $p_1 = 1, p_2 = 0.1, p_3 = 2.0$ which, in the nomenclature used by **gcfit** is $S_0 = 1.0, A = 0.362, B = 2.0$. Then 7.5% relative error was added for five replicates using program **adderr** to generate test file `weibull.tf2` which was analyzed using the option to estimate S_0 giving the following table of parameter estimates and the best fit curve plot. Of course, if the starting fraction were known exactly, as in actual survival data, there would be no values for $t = 0$ since it would be assumed that $S_0 = 1$. However, allowing the $t = 0$ value to be estimated should perhaps always be used for data smoothing to avoid bias.

Parameter	Value	Std.error	Lower95%cl	Upper95%cl	p
A	0.30648	0.0056521	0.29489	0.31808	0.0000
B	2.0879	0.11926	1.8432	2.3326	0.0000
S_0	0.95465	0.015961	0.92190	0.98740	0.0000
$t_{1/2}$	2.7375	0.063837	2.6065	2.8685	0.0000



Growth and Survival Models available in program gcfit

1. Exponential model $dS/dt = kS$

$$S(t) = A\exp(kt), \text{ where } A = S_0$$

2. Monomolecular model $dS/dt = k(A - S)$

$$S(t) = A[1 - B\exp(-kt)], \text{ where } B = 1 - S_0/A$$

3. Logistic model $dS/dt = kS(A - S)/A$

$$S(t) = A/[1 + B\exp(-kt)], \text{ where B } = A/S_0 - 1$$

4. Gompertz model $dS/dt = kS[\log(A) - \log(S)]$

$$S(t) = A\exp[-B\exp(-kt)], \text{ where } B = log(A/S_0)$$

5. Von Bertalanffy 2/3 model $dS/dt = \eta S^{2/3} - \kappa S$

$$S(t) = [A^{1/3} - B\exp(-kt)]^3$$

$$\text{where } A^{1/3} = \eta/\kappa, B = \eta/\kappa - S_0^{1/3}, k = \kappa/3$$

6. Model 3 with constant $f(t) = S(t) - C$

$$df/dt = dS/dt = kf(t)(A - f(t))/A$$

$$S(t) = A/[1 + B\exp(-kt)] + C$$

7. Model 4 with constant $f(t) = S(t) - C$

$$df/dt = dS/dt = kf(t)[\log(A) - \log(f(t))]$$

$$S(t) = A\exp[-B\exp(-kt)] + C$$

8. Model 5 with constant $f(t) = S(t) - C$

$$df/dt = dS/dt = \eta f(t)^{2/3} - \kappa f(t)$$

$$S(t) = [A^{1/3} - B\exp(-kt)]^3 + C$$

9. Richards model $dS/dt = \eta S^m - \kappa S$

$$S(t) = [A^{1-m} - B\exp(-kt)]^{[1/(1-m)]}$$

$$\text{where } A^{1-m} = \eta/\kappa, B = \eta/\kappa - S_0^{1-m}, k = \kappa(1-m)$$

$$\text{if } m < 1 \text{ then } \eta, \kappa, A \text{ and } B \text{ are } > 0$$

$$\text{if } m > 1 \text{ then } A > 0 \text{ but } \eta, \kappa \text{ and } B \text{ are } < 0$$

10. Preece and Baines model $f(t) = \exp[k_0(t - \theta)] + \exp[k_1(t - \theta)]$

$$S(t) = h_1 - 2(h_1 - h_\theta)/f(t)$$

1. Exponential survival model $S(t) = \exp(-At)$

$$f(t) = AS(t)$$

$$h(t) = A$$

2. Weibull survival model $S(t) = \exp[-(At)^B]$

$$f(t) = AB[(At)^{B-1}]S(t)$$

$$h(t) = AB(At)^{B-1}$$

3. Gompertz survival model $S(t) = \exp[-(B/A)\{\exp(At) - 1\}]$

$$f(t) = B\exp(At)S(t)$$

$$h(t) = B\exp(At)$$

4. Log-logistic survival model $S(t) = 1/[1 + (At)^B]$

$$f(t) = AB(At)^{B-1}/[1 + (At)^B]^2$$

$$h(t) = AB(At)^{B-1}/[1 + (At)^B]$$

8.5.7 Fitting initial rates, half times, lag times and asymptotes

It frequently happens that measurements of a response as a function of time say, or concentration, etc., are made in order to measure an initial rate, a lag time, an asymptotic steady state rate, a half saturation point, or a horizontal asymptote. Examples could be the initial rate of an enzyme catalyzed reaction, the transport of labeled solute out of loaded erythrocytes, or extent of muscle contraction in response to an agonist.

The models required to perform these operations are available from the main SimFIT menu after first choosing [A/Z] then opening program **inrate**.

Theory

Stated in equations we have the responses given by a deterministic component plus a random error

$$y_i = f(t_i) + \epsilon_i,\ i = 1, 2, \ldots, m$$

and it is wished to measure the limiting values

$$\begin{aligned} \text{the initial rate} &= \frac{df}{dt} \text{ at } t = 0 \\ \text{the asymptotic slope} &= \frac{df}{dt} \text{ as } t \to \infty \\ \text{the half saturation point} &= t_{1/2} \text{ where } f(t_{1/2}) = f(0)/2 \\ \text{the final asymptote} &= f \text{ as } t \to \infty. \end{aligned}$$

There are numerous ways to make such estimates in SimFIT and the method adopted depends critically on the type of experiment. Choosing the wrong technique can lead to biased estimates, so you should be quite clear which is the correct method for your particular requirements. In particular, is $f(0) = C = 0$ required ?

The models used by program **inrate** are

$$\begin{aligned} f_1 &= Bt + C \\ f_2 &= At^2 + Bt + C \\ f_3 &= \alpha[1 - \exp(-\beta t)] + C \\ f_4 &= \frac{V_{max} t^n}{K_m^n + t^n} + C \\ f_5 &= Pt + Q[1 - \exp(-Rt)] + C \end{aligned}$$

and there are test files to illustrate each of these that can be selected from the SimFIT file selection control after using the [Demo] button. It is usual to assume that $f(t)$ is an increasing function of t with $f(0) = 0$, which is easily arranged by suitably transforming any initial rate data. For instance, if you have measured efflux of an isotope from vesicles you would analyze the rate of appearance in the external solute, that is, express your results as

$$f(t) = \text{initial counts - counts at time } t$$

so that $f(t)$ increase from zero at time $t = 0$. All you need to remember is that, for any constant K,

$$\frac{d}{dt}\{K - f(t)\} = -\frac{df}{dt}.$$

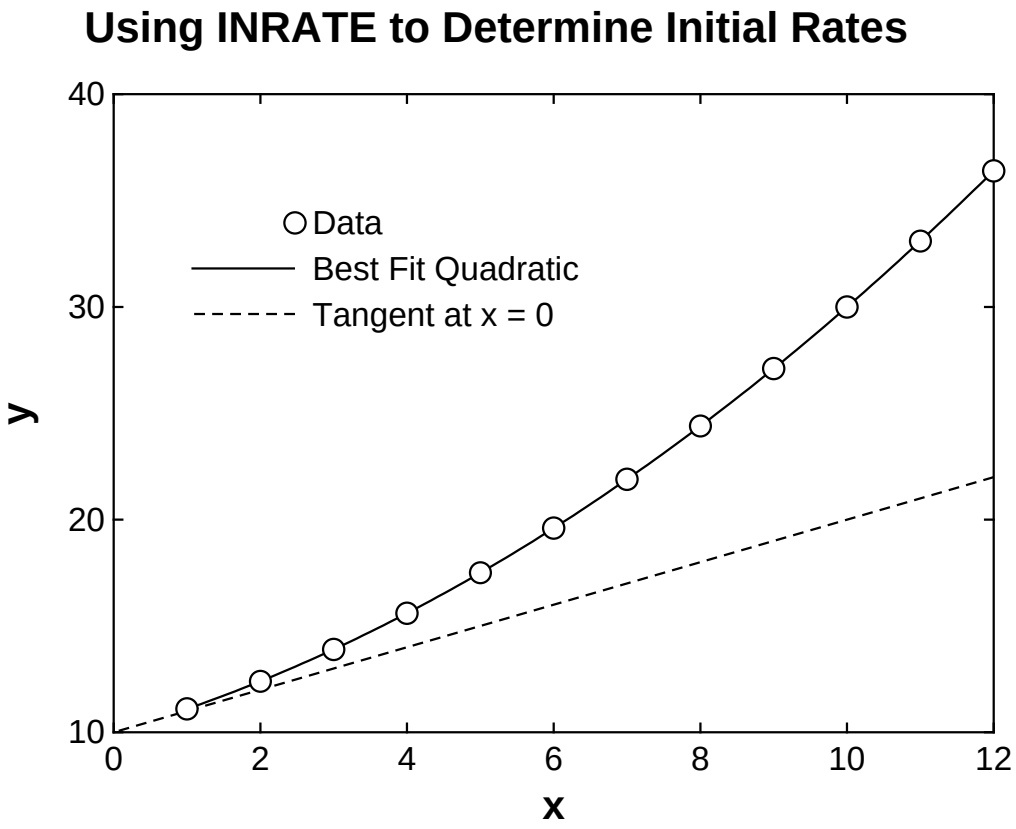
However it is sometimes difficult to know exactly when $t = 0$, e.g., if the experiment involves quenching, so there exists an option to force the best fit curve to pass through the origin with some of the models if this is essential. The models available will now be summarized.

1. f_1: This is used when the data are very close to a straight line and it can only measure initial rates.
2. f_2: This adds a quadratic correction and is used when the data suggest only a slight curvature. Like the previous model it can only estimate initial rates.
3. f_3: This model is used when the data rapidly bend to a horizontal asymptote in an exponential manner. It can be used to estimate initial rates, half-times ($\log(2)/\beta$), and final horizontal asymptotes.
4. f_4: This model can be used with n fixed (e.g., $n = 1$) for the Michaelis-Menten equation, or with n varied (the Hill equation). It is not used for initial rates but is sometimes better for estimating half saturation points (K_m), or final horizontal asymptotes than the previous model.
5. f_5: This is the progress curve equation used in transient enzyme kinetics. It is used when the data have an initial lag phase followed by an asymptotic final steady state. It is not used to estimate initial rates, final horizontal asymptotes, or AUC. However, it is very useful for experiments with cells or vesicles which require a certain time before attaining a steady state, and where it is wished to estimate both the length of lag phase and the final steady state rate.

To understand these issues, see what happens the test files. These are, models f_1 and f_2 with `inrate.tf1`, model f_3 with `inrate.tf2`, model f_4 with `inrate.tf3` and model f_5 using `inrate.tf4`.

Example 1: Using $f_2(t)$ to estimate initial rates

A useful method to estimate initial rates when the true deterministic equation is unknown is to fit the quadratic $f_2(t) = At^2 + Bt + C$, in order to avoid the bias that would inevitably result from fitting a line to nonlinear data. Use **inrate** to fit the test file `inrate.tf1`, and note that, when the model has been fitted, it also estimates the slope at the origin. The reason for displaying the tangent in this way, as in the figure, is to give you some idea of what is involved in extrapolating the best fit curve to the origin, so that you will not accept the estimated initial rate uncritically.



Example 2: Using $f_3(t)$ to estimate half times, initial rates, and horizontal asymptotes

This model,

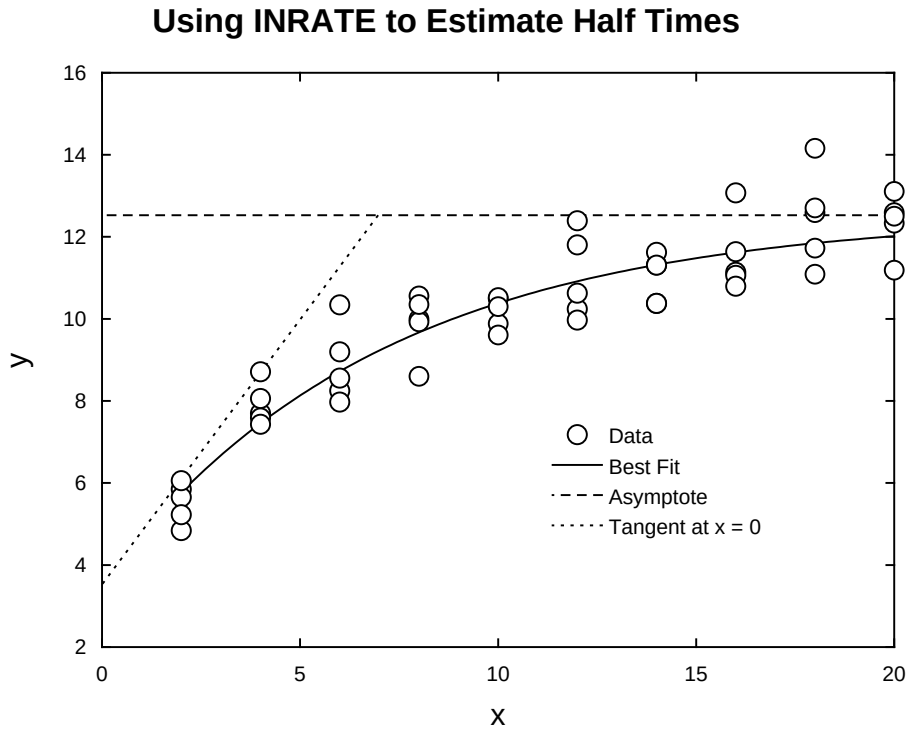
$$f_3(t) = \alpha[1 - \exp(-\beta t)] + C,$$

sometimes referred to as the monomolecular model, is useful for characterizing time-dependent processes that approach a horizontal asymptote.

For instance, program **adderr** was used to generate 5 replicates with 7.5% relative error added to the exact data in test file `inrate.tf2` to simulate experimental error with standard errors estimated from the replicates, and the monomolecular equation was fitted to yield the following results.

Parameter	Value	Std. error	Lower95%cl	Upper95%cl	p
α	9.0018	0.39481	8.2075	9.7960	0.0000
β	0.14346	0.023000	0.097186	0.18973	0.0000
C	3.5251	0.48988	2.5396	4.5106	0.0000

Estimated initial value i.e. $f(t = 0)$ = 3.5251
Estimated initial rate $df/dt(t = 0)$ = 1.2914
At t = -2.3035, $f = 0$ and df/dt = 1.7971 ... extrapolated value
Estimated final asymptote $f(\infty)$ = 12.527
Estimated half time (i.e. $\log(2)/\beta$) = 4.8318



Note that the estimate referred to as the half time is defined by

$$\hat{t}_{1/2} = \log(2)/\hat{\beta}$$

which will only be the time to reach half the maximal value ($\hat{\alpha} + \hat{C}$) when the model is fitted using the option to set $C = 0$. Often it will be sensible to use the option $f(0) = 0$, i.e. fixing $C = 0$ when using **inrate**.

Example 3: Using $f_4(t)$ to estimate half saturation points and horizontal asymptotes

The Hill equation given by

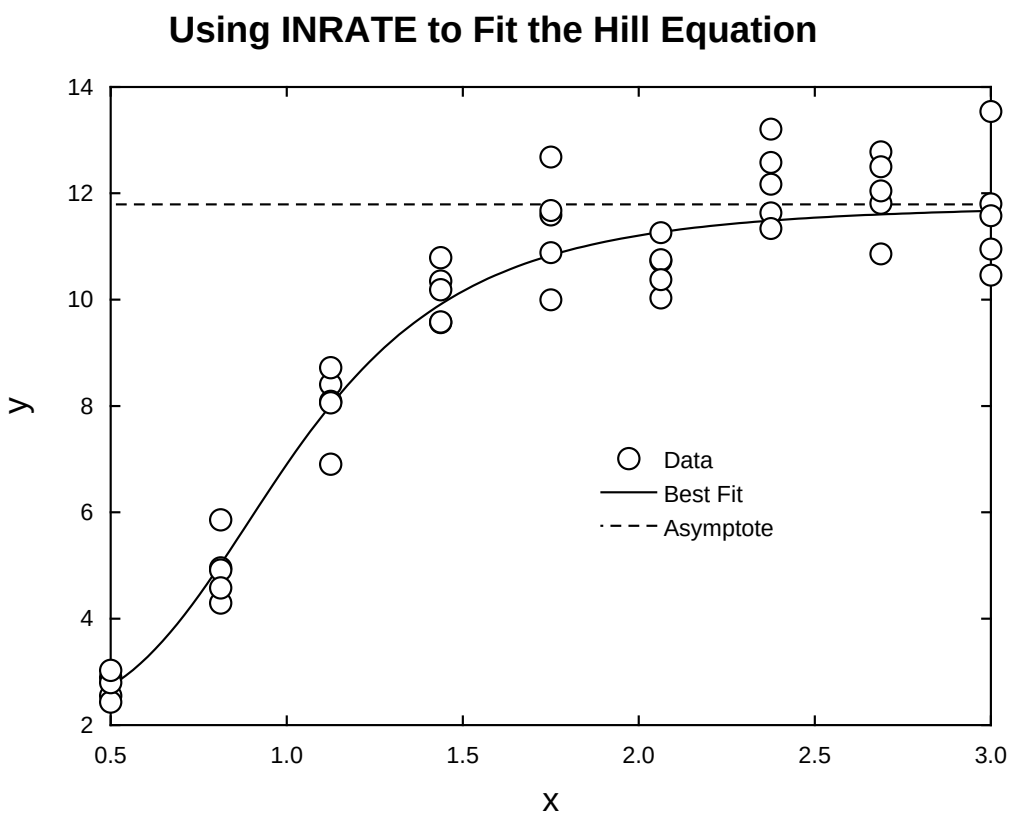
$$f_4(t) = \frac{V_{max}t^n}{K_m^n + t^n} + C$$

only models a meaningful process when $n = 1$, as the cases with $n < 1$ and n non-integer have no sensible interpretation, while those with integer $n > 1$ are only coarse approximations to receptor saturation. Nevertheless, this equation is widely used as an empirical model to fit data in order to estimate the origin C, half saturation point K_m, and horizontal asymptote $V_{max} + C$. Note that program **inrate** allows the exponent n to be fixed. which is preferred, or varied as a parameter, which is not usually recommended as it can lead to an ill-defined best-fit model.

For instance, program **adderr** was used to generate 5 replicates with 7.5% relative error added to the exact data in test file `inrate.tf3` to simulate experimental error with standard errors estimated from the replicates, and the Hill equation with $n = 4$ was fitted to yield the following results.

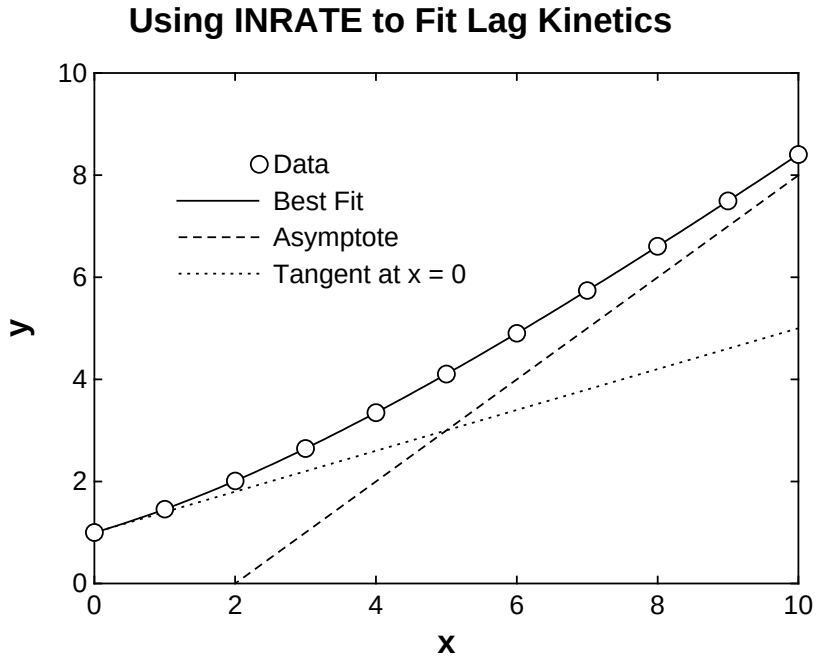
Parameter	Value	Std. error	Lower95%cl	Upper95%cl	p
V_{max}	9.6039	0.21213	9.1758	10.032	0.0000
K_m	1.0090	0.034222	0.93999	1.0781	0.0000
C	2.1875	0.16332	1.8579	2.5171	0.0000
n	4.0000	0.0000	4.0000	4.0000	Fixed

Estimated initial value i.e. $f(t = 0)$ = 2.1875
Estimated initial rate $df/dt(t = 0)$ = 0.0000
Estimated final asymptote $f(\infty)$ = 11.91
The exponent was fixed at the value n = 4

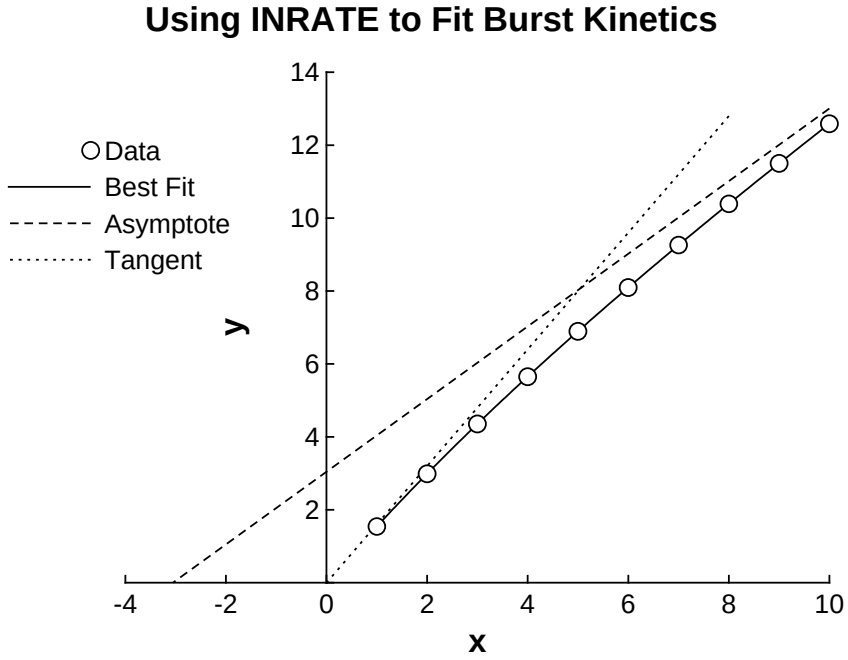


Example 4: Using $f_5(t)$ to estimate lag times and asymptotic steady states

Use **inrate** to fit $f_5(t) = Pt + Q[1 - \exp(-Rt)] + C$ to data in test file `inrate.tf4`, and observe that the asymptotic line is displayed in addition to the tangent at the origin, as in the next figure.



However, sometimes a burst phase is appropriate, rather than lag phase, as shown next.



8.6 Nonlinear regression: advanced

Tutorials and worked examples for simulation, curve fitting, statistical analysis, and plotting.
https://simfit.uk
https://simfit.org.uk
https://simfit.silverfrost.com

8.6.1 Introduction to constrained nonlinear regression

This involves minimizing an objective function such as a scaled weighted sum of squares

$$\frac{WSSQ}{NDOF} = \frac{1}{n-m}\sum_{i=1}^{n} w_i(y_i - f(x_i, \Theta))^2$$

with respect to a m dimensional parameter vector Θ, where there are n observations y_i, and a model $f(x, \Theta)$, along with parameter starting estimates and limits, and specified weights w_i. The aim is to use the parameter estimates and confidence limits to asses the value of model parameters that have a physical interpretation, such as diffusion constants, chemical reaction rate constants, or growth rates, etc. Of course the observations y, independent variables x, functions f, and weights w would often be vector quantities. The main SIMFIT programs to perform this type of optimization are **qnfit** providing quasi-Newton and other methods, and **deqsol** for systems of differential equations, and numerous considerations must be understood and several conditions must be satisfied before these iterative techniques can obtain sensible solution points. Some of these these are now summarized.

1. **The model**

   The model can be used from a default library of models, but it is normally anticipated that users would define their own specialized model, and create a model file using program **usermod**. It will be obvious that the model should be parsimonious, using only the minimum number of parameters, and where every parameter has a scientific interpretation. Often data can be normalized by users before curve-fitting if this reduces the number of parameters that need to be estimated. For instance, normalizing observations so that $f(0) = 0$, or $f(\infty) = 1$, or, especially in the case of differential equations, so that initial conditions do not need to be estimated.

2. **The data**

   The data must be extensive in the sense that $n >> m$, and with a high signal to noise ratio, but they should also cover a range where the effect of all parameters can be assessed. For instance, with exponential decay the range should extend beyond the longest half-life, and, where growth data or ligand binding data approach a horizontal asymptote, the experimental data should clearly be starting to look asymptotic.

3. **The weights**

   The variance of experimental observations almost always increases as the absolute value of the observations increase, and even though the expectation will often be zero, the distribution will have longer tails, more like a Cauchy than a Gaussian distribution. Now the theory required for the analysis of goodness of fit and calculation of statistics to estimate parameter reliability and perform model discrimination depends on the principle of maximum likelihood, which assumes a linear model with uncorrelated normally distributed error. This means that either $w_i = 1$ if the variance is constant, or $w_i = 1/s_i^2$ otherwise where the standard deviations s_i are known exactly. So several approaches are possible.

   - Assume constant variance and set all $s_i = 1$. This leads to fitting being dominated by large observations, and hence the parameters contributing to the large values will be estimated more accurately than those only contributing to small values.

- Assume constant relative error and assume that standard deviation is proportional to the absolute value of the observation. This can lead to the opposite effect to assuming constant variance, i.e. biasing the fit towards small values.
- Assume that s_i is a defined function of the observations or the best-fit function values. This requires an assumption about the functional dependency of variance and, if the best-fit model is used rather than observations, then weighting changes as iterations proceed.
- Estimate the error variance independently. This is undoubtedly the best method as long as at least five replicates are determined at each independent variable setting and, if possible, a smoothing technique is used to determine a reliable model for the change in variance as a function of the value of observations.

4. **The starting estimates and limits**

   Constrained nonlinear optimization is an iterative technique that attempts to find a local minimum given parameter starting estimates and parameter limits. So naturally it is important that the starting estimates are close to the true values and the limits are not so wide that parameters can stray into unlikely regions of parameter space. SimFIT programs **qnfit** and **deqsol** also use the starting estimates to normalize the internal parameters to order unity, as calculation of maxim descent vectors and augmented Lagrangians will be most accurate if all internal parameters are of order unity.

Success and Failure

If all the above criteria are met then convergence to a minimum should be achieved so that $WSSQ$ will be approximately chi-squared distributed with $NDOF = n - m$. Then the objective function should be of order unity with reasonable parameter estimates and satisfactory goodness of fit analysis.

On the other hand, if the conditions are not met then failure will occur with appropriate error messages. In such dubious cases you should switch on the options to evaluate the parameter covariance matrix, the condition number of the Hessian, and study tables of residuals and residuals plots. Note that, if the objective functions is too small or too large on entry due to an inappropriate model, poor quality data, incorrect weighting, or unrealistic starting estimates, then the routine will not be able to estimate the gradient vector and exit will occur without fitting.

In order to become familiar with program **qnfit** some very simple examples will be given next to illustrate the standard way to proceed. That is:

1. select the model type required, e.g. one function of one variable;
2. input a data set composed in the EXPERT mode with starting estimates and limits appended;
3. read in the model file, e.g. created using program **usermod**; then
4. proceed to fitting.

In the next fairly trivial examples note that the test data files and model files can be easily located using the [Demo] button on the file-open dialogue box.

Example 1: One function of one variable

Open SimFIT program **qnfit** then follow the next steps.

1. Choose to fit one function of one variable
2. From the file-open dialogue press [Demo] then view and open the test file `qnfit_data.tf1`

3. Choose to open an ASCII text model file then from the file-open dialogue press [Demo] then view and open the test file
   `qnfit_model.tf1`

4. Choose the EXPERT mode for starting estimates then fit

Note that this is simulated data for a quadratic and the EXPERT mode appended section is as follows.

```
begin{limits}
  -10   1   10
  -10   1   10
  -10   1   10
end{limits}
```

The model file defines a quadratic $f(x) = p_1x + p_2x^2 + p_3$ as follows.

```
%
Model for a polynomial of degree 2
f(x) = p(1)x + p(2)x^2 + p(3)
%
1 equation
1 variable
3 parameters
%
begin{expression}
f(1) = p(1)x + p(2)x^2 + p(3)
end{expression}
%
```

Now, to obtain a permanent copy of the outcome after fitting, extract the table of best-fit parameters using the [Results] then [Extract tables] options from the main SIMFIT menu to import the following table into your document.

Best-fit parameters for curve-fit 1 using LBFGSB

Number	Low-Limit	High-Limit	Value	Std.Error	Lower95%cl	Upper95%cl	p
1	-10.0	10.0	2.12035	0.0197309	2.07829	2.1624	0.0000
2	-10.0	10.0	-0.11565	0.0035714	-0.12326	-0.1080	0.0000
3	-10.0	10.0	0.10347	0.0032091	0.09663	0.1103	0.0000

For 50,90,95,99% confidence limits using [parameter value +/- $t(\alpha/2)$*std.err.]
$t(0.25) = 0.691$, $t(0.05) = 1.753$, $t(0.025) = 2.131$, $t(0.005) = 2.947$

Note that the $t_\nu(.)$ values are provided in case you want to calculate parameter confidence limits in addition to the default 95% values.

Example 2: One function of two variables

Proceeding exactly as for example one except that one function of two variables is chosen, the data file is `qnfit_data.tf2` and the model file is `qnfit_model.tf2` observe that now the data file has four columns $(x, y, g(x, y), s)$ for observations $g(x, y)$.

The appended EXPERT mode section defining the lower-limits, starting values, and upper limits for the three parameters follows

```
begin{limits}
-10 -2 10
-10  2 10
-10  4 10
end{limits
```

while the model has the next definition followed by the results from fitting.

```
%
Linear model with two variables
g(x,y) = p(1)x + p(2)y + p(3)
%
1 equation
2 variables
3 parameters
%
begin{expression}
f(1) = p(1)x + p(2)y + p(3)
end{expression}
%
```

Best-fit parameters for curve-fit 2 using LBFGSB

Number	Low-Limit	High-Limit	Value	Std.Error	Lower95%cl	Upper95%cl	p
1	-10.0	10.0	0.96311	0.0334131	0.895683	1.03054	0.0000
2	-10.0	10.0	0.95436	0.0323055	0.889165	1.01955	0.0000
3	-10.0	10.0	1.05694	0.0359344	0.984422	1.12946	0.0000

For 50,90,95,99% confidence limits using [parameter value +/- $t(\alpha/2)$*std.err.]
$t(0.25)$ = 0.680, $t(0.05)$ = 1.682, $t(0.025)$ = 2.018, $t(0.005)$ = 2.698

Example 3: One function of three variables

This time the data file is `qnfit_data.tf3`, while the model file is `qnfit_model.tf3` defining a function of three variables displayed next, followed by the results from curve fitting.

```
%
Linear model with three variables
h(x,y,z) = p(1)x + p(2)y + p(3)z + p(4)
%
1 equation
3 variables
4 parameters
%
begin{expression}
f(1) = p(1)x + p(2)y + p(3)z + p(4)
end{expression}
%
```

Best-fit parameters for curve-fit 3 using LBFGSB

Number	Low-Limit	High-Limit	Value	Std.Error	Lower95%cl	Upper95%cl	p
1	-10.0	10.0	1.00348	0.015231	0.97335	1.03361	0.0000
2	-10.0	10.0	0.99476	0.017071	0.96098	1.02853	0.0000
3	-10.0	10.0	0.98594	0.017159	0.95199	1.01988	0.0000
4	-10.0	10.0	-2.94533	0.055805	-3.05572	-2.83493	0.0000

For 50,90,95,99% confidence limits using [parameter value +/- $t(\alpha/2)$*std.err.]
$t(0.25)$ = 0.676, $t(0.05)$ = 1.657, $t(0.025)$ = 1.978, $t(0.005)$ = 2.614

8.6.2 Choosing parameter starting values and limits

Once a correct model has been chosen and accurate data consistent with this model and extending over a wide range have been obtained, the next most important item is to select parameter starting estimates close to the expected values, and also to provide sensible upper and lower parameter limits.

The simple SimFIT curve-fitting programs, such as **mmfit**, obtain starting estimates and limits as follows.

1. Scale the data to order unity.
2. Obtain slopes and/or horizontal asymptotes by linear regression to the early and late data.
3. Because data for simple curve-fitting programs must be in non-decreasing order, fits to extreme data points can usually be used to estimate parameters that become important at the extreme data regions.
4. As each of the simple curve-fitting programs is designed for a particular model, then starting estimates and limits can usually be guessed.
5. In addition, random searches around these estimates, but constrained by the parameter limits, can be used to refine such values.

However, as **qnfit** and **deqsol** are for expert users they must perform such tasks themselves. There is nothing to stop expert users from scaling the data before input to **qnfit** and submitting data in random order, etc., but the following facts cannot be avoided.

- The starting estimates must be reasonably close to the probable parameters.
- If this condition is not met then a meaningful solution is unlikely to be found as nonlinear regression only seeks to find a local minimum.
- The upper and lower limits must not be too wide or too narrow.
- If these conditions are not met then parameters will either not be able to move sufficiently freely to allow the gradient vector and Hessian to be estimated, or they will simply stick at boundaries.

Such effects will be immediately obvious by the following situations.

- Warnings about convergence problems and bad fitting will be displayed.
- Parameter standard errors will be large.
- The fit to data will be unsatisfactory.

The obvious action is to try new starting estimates and limits, possibly with a random search, until a good fit results. Then subsequent data files can have such values appended.

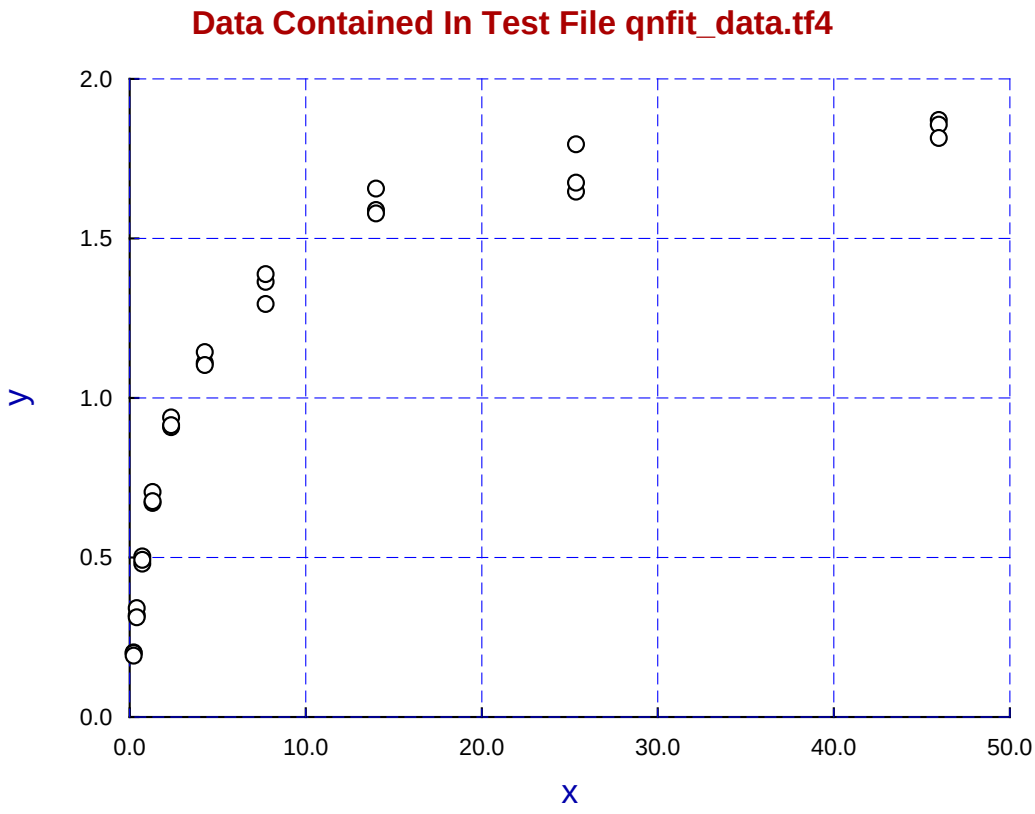
Some typical examples will be given under the assumption that users will supply their models as ASCII text files prepared using **usermod**, and will always append starting parameters and limits to the data files and use the EXPERT mode of running **qnfit**.

Example 4: Fitting the two-site Michaelis-Menten model

The initial steady-state for a mixture of two one-site enzymes or for two independent sites on one enzyme can be modeled by this expression

$$v(s) = \frac{Vmax_1 S}{Km_1 + S} + \frac{Vmax_2 S}{Km_2 + S}$$

where S is the (constant) substrate concentration. First of all use **simplot** to plot the tests data file `qnfit_data.tf4` to obtain the following graph.



From this graph we can see from the asymptotic behavior as $S \rightarrow \infty$ that $Vmax_1 + Vmax_2 \approx 2$, while from the half-saturation point Km_1 and Km_2 are between 1 and 10. It should also be observed that, because of symmetry in the model, the pairs $(Vmax_1, Km_1)$ and $(Vmax_2, Km_2)$ are arbitrary and can be interchanged.

To emphasize: the usual and recommended way to use **qnfit** is to read in the data as a SimFIT data file with starting estimates and limits appended, and then to supply the model required as a ASCII text file prepared using **usermod**. So, to demonstrate this procedure, the next sequence should be used.

- Open **qnfit** from the [A/Z] button on the main SimFIT menu, choose to input data for one function of one variable, and then, from the [Demo] button, select `qnfit_data.tf4` which has the following starting estimates and limits appended.

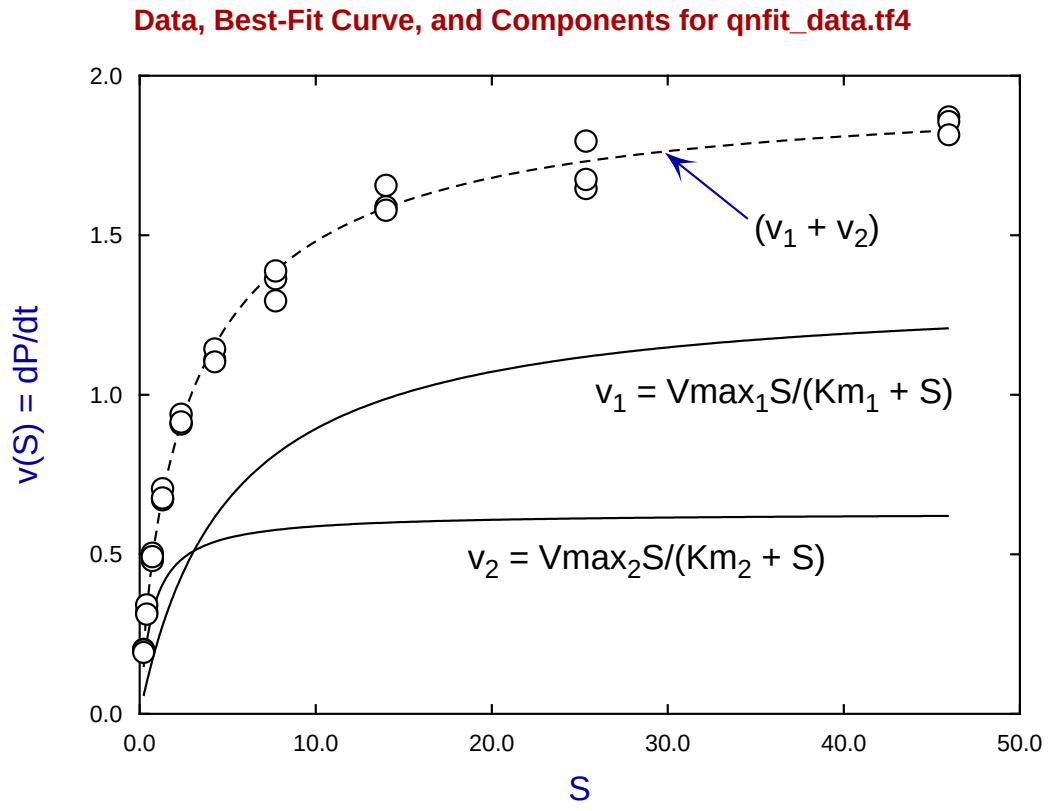
```
begin{limits}
 0.1   0.8   5.0
 0.1   1.2   5.0
 0.1   1.0  15.0
 0.1   8.0  15.0
end{limits}
```

- Then choose to input the corresponding ASCII text model file `qnfit_model.tf4` which defines the model in the following way, where using $f(1) = A + B$ shows how the model could easily be extended to three or more isoenzymes.

```
%
Sum of two independent Michaelis-Menten active sites
v(S) = Vmax_1*S/(Km_1 + S) + Vmax_2*S/(Km_2 + S) where
p(1) = Vmax_1, p(2) = Vmax_2 p(3) = Km_1, p(4) = Km_2
%
1 equation
1 variable
4 parameters
%
begin{expression}
A = p(1)x/[p(3) + x]
B = p(2)x/[p(4) + x]
f(1) = A + B
end{expression}
%
```

The parameter estimates and confidence limits from fitting are displayed in the next table followed by a plot of data, best-fit curve and sub-models contributing to the overall fit.

Number	Low-Limit	High-Limit	Value	Standard Error	Lower95%cl	Upper95%cl	*p*
1	0.1	5.0	1.02901	0.13352	0.7546	1.30346	0.0000
2	0.1	5.0	1.02542	0.10378	0.8121	1.23873	0.0000
3	0.1	15.0	1.04328	0.11698	0.8028	1.28372	0.0000
4	0.1	15.0	9.74605	2.86518	3.8566	15.6355	0.0022



The plot was constructed by re-fitting using the corresponding built-in **qnfit** library model which is recognized as a linear combination of sub-models, but this does not happen with user-defined model files so the separate components have to be plotted independently then overlayed upon the usual best-fit display.

Example 5: Fitting a double exponential model

This example involves fitting a model for B in the irreversible chemical kinetic scheme $A \xrightarrow{k_1} B \xrightarrow{k_2} C$ which, with various additional definitions, is used extensively elsewhere, for instance for absorption from the stomach and elimination from the blood in pharmacokinetics.

$$f(t) = \frac{k_1 A_0}{k_2 - k_1} \left[\exp(-k_1 t) - \exp(-k2t)\right], \quad k_1 \neq k_2$$

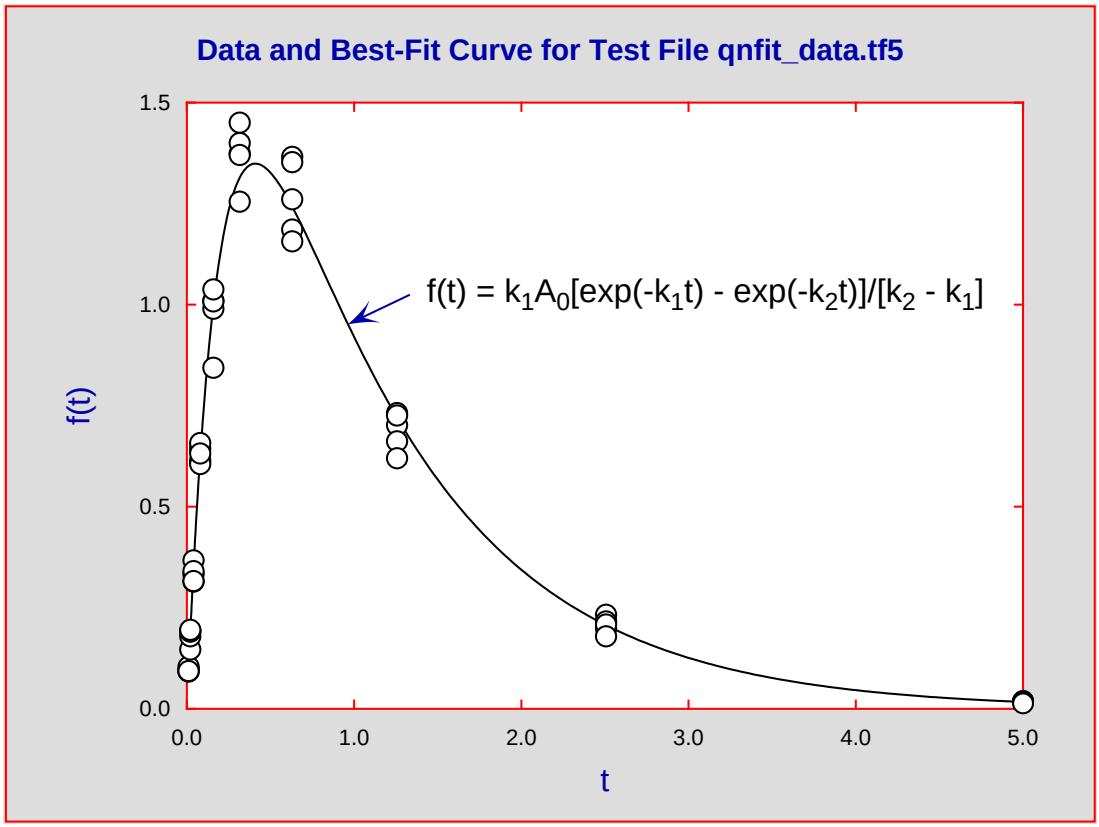
The test file is `qnfit_data.tf5` and, from viewing the data in **simplot** clearly k_1 is around unity, while $k_2 > k_1$, and $2 < A_0 < 30$ is less easy to guess, so the following starting estimates and limits are appended.

```
begin{limits}
0.0   0.5 10.0
0.0   6.0 10.0
0.0  12.0 30.0
end{limits}
```

The model file in `qnfit_model.tf5` is shown next, followed by the table of results and the plot.

```
A = p(1)p(3)/[p(2) - p(1)]
B = exp[-p(1)x] - exp[-p(2)x]
f(1) = A*B
```

Number	Low-Limit	High-Limit	Value	Standard Error	Lower95%cl	Upper95%cl	*p*
1	0.0	10.0	1.00458	0.015634	0.97313	1.03604	0.0000
2	0.0	10.0	4.85240	0.146363	4.55795	5.14684	0.0000
3	0.0	30.0	9.83258	0.224251	9.38144	10.2837	0.0000



8.6.3 Calculating with the best fit curve

When a theoretical model has been fitted to a data set it is sometimes required to employ the model evaluated using the m best-fit parameters θ_i for calculations. The SIMFIT program **qnfit** provides the following options for doing this using the best-fit model

$$\hat{f}(x) = f(x, \hat{\Theta}), \text{ where } \hat{\Theta} = (\hat{\theta}_1, \hat{\theta}_2, \ldots, \hat{\theta}_m)$$

after displaying the goodness of fit.

1. Plotting the best fit curve together with the data.
2. Plotting residuals in several ways.
3. Extrapolating the best-fit curve beyond the data range.
4. Displaying the contribution of individual components to the overall fit with those models that are sums of sub-models. This type of graphical deconvolution requires the use of those models in the SIMFIT library that are defined in this way. Actually the user-friendly curve-fitting programs also provide this feature where it is appropriate, e.g. **exfit** for sums of exponentials.
5. Evaluating the best-fit model at specific values of the independent variable.
6. Calculating the area under the best-fit curve between defined end points, i.e. the AUC.
7. Plotting the derivative of the best-fit curve over a data range in order to estimate the maximum and minimum slopes.
8. Evaluating the derivative at selected points.
9. Using the best-fit curve for inverse prediction, i.e. calibration.

Numerical techniques

Some explanation is required for those options in this list that require numerical techniques.

- **Calculating the AUC**

  This can only be done within **qnfit** using Simpson's rule. However the option is provided to vary the number of divisions so that this number can be increased if required until a stable result is obtained.

  Perhaps the most common reason for using this technique is to calculate the area under a curve where the range can be specified, e.g. in pharmacokinetics to estimate bioavailability such as drug plasma levels.

- **Calculating derivatives**

  Here the derivative is estimated at a specific point x using the parameter h in the approximation

$$\frac{dy}{dx} \approx \frac{\hat{f}(x+h) - \hat{f}(x)}{h}$$

  and the option is provided to alter h until a stable result is obtained.

  A very common use for this technique is to estimate the points where the maximum and minimum gradients occur in growth curve studies.

- **Calibration**

  For this procedure to reverse-predict x given it is necessary to solve the nonlinear equation

  $$y - \hat{f}(x) = 0$$

  for a specified y and the unknown x using an iterative technique. If the default starting values do not lead to a satisfactory solution the option is provided to choose new starting estimates A and B where

  $$(y - \hat{f}(A))(y - \hat{f}(B)) < 0.$$

  This functionality means that **qnfit** can be used as a general purpose calibration curve program in those instances where a non-typical calibration curve has to be constructed using a specific model rather than using the SIMFIT polynomial, GLM, or spline smoothing calibration curve programs.

As a very simple example to illustrative these calculations take the data file `qnfit_data.tf1` together with the model file `qnfit_model.tf1` and first fit the model to the data to obtain a best-fit curve. Then continue through the display of goodness of fit until the main **qnfit** menu is reached. The calculation options will then be seen.

This data-model pair is very convenient because the data are very accurate and the model is linear in parameters and very simple, being the quadratic

$$f(x) = \theta_1 x + \theta_2 x^2 + \theta_3,$$

using the usual SIMFIT scheme that constants in theoretical models come last. So, for this extreme case, we can perform the calculations analytically both for the best-fit parameters (2.12035, -0.115647, 0.103471) and the exact parameters (2, -0.1, 0.1) before random error was added using **adderr**, leading to the following results.

Procedure	QNFIT using $\hat{\Theta}$	Calculated using $\hat{\Theta}$	Exact using Θ
Area from 0 to 10	68.5031	68.5032	67.6667
Derivative for $x = 1$	1.88905	1.88906	1.8
Derivative for $x = 2$	1.65776	1.65776	1.6
Derivative for $x = 3$	1.42647	1.42647	1.4
x given $y = 1$	0.43305	0.43305	0.46068
x given $y = 2$	0.94294	0.94294	1.0
x given $y = 3$	1.48658	1.48660	1.57385
Function value for x = 1	2.10817	2.10817	2.0
Function value for x = 1	3.88158	3.88158	3.7
Function value for x = 1	5.42369	5.42370	5.2

Of course this is a ridiculously simple example which is just given to demonstrate how these numerical techniques could be used in more typical situations.

8.6.4 Fitting a mixture of two normal distributions

Often samples consist of a mixture of distributions. For instance a sample of heights of subjects drawn from a homogeneous population, i.e. of the same age and medical condition, could appear to be be approximately normally distributed but would actually consist of two sub-populations, male and female. In reality, special techniques exist for analyzing certain cases where populations cannot be physically separated into sub-groups but can be resolved into supposed sub-populations using the method of maximum likelihood. However, the curve fitting approach will be discussed in this tutorial because, in principle, it can be used for arbitrary mixtures of any any distributions, not just normal distributions.

For instance, to explain how to use SIMFIT program **qnfit** for this purpose, consider the simplest case of a sample arising from a mixture of two normally distributed sub populations, so that a sample partitioned into histogram bins could be approximately modeled by the expression

$$f(x) = \frac{t}{\sigma_1\sqrt{2\pi}}\exp\left(-\frac{1}{2}\left\{\frac{x-\mu_1}{\sigma_1}\right\}^2\right) + \frac{1-t}{\sigma_2\sqrt{2\pi}}\exp\left(-\frac{1}{2}\left\{\frac{x-\mu_2}{\sigma_2}\right\}^2\right)$$

where $0 \le t \le 1$ and the parameters $t, \mu_1, \mu_2, \sigma_1, \sigma_2$ must be estimated by fitting to the histogram bins. The serious problem with this approach is that the shape of the histogram, and therefore the best-fit parameters, will depend on the number of bins chosen. It should therefore be obvious that a very large sample will be necessary, and meaningful parameter estimates can only be expected when μ_1 and μ_2 are widely separate, σ_1 and σ_2 are similar and less than the difference between μ_1 and μ_2, and the partitioning parameter t must obey $t \approx 0.5$. Of course the constraints $\sigma_1 > 0, \sigma_2 > 0$ also must be imposed.

Example 6: Fitting histogram data

The data file `qnfit_data.tf6` can be selected from **qnfit** by clicking on the [Demo] button, and it is listed below after extracting as a table using the [Results] button on the main SIMFIT menu.

```
Data file qnfit_data.tf4
  10          3
-3.6     0.0375      1.0
-2.8     0.0625      1.0
-2.0     0.2000      1.0
-1.2     0.2000      1.0
-0.4     0.1000      1.0
 0.4     0.1250      1.0
 1.2     0.1250      1.0
 2.0     0.2500      1.0
 2.8     0.1250      1.0
 3.6     0.0250      1.0
begin{limits}
-5.0       -1.0       0.0
 0.1        0.8       5.0
 0.1        0.4       0.9
 0.0        1.0       5.0
 0.1        1.2       5.0
end{limits}
```

This file was created by reading a mixed sample of 50 $N(-1.5, 1)$ numbers and 50 $N(1.5, 1)$ numbers from program **rannum** into the exhaustive analysis of a vector routine available under [Data exploration] from the [Statistics] option on the main SIMFIT menu. This indicates that the mixed sample is not consistent with a single normal distribution and this step should be taken before fitting any data set because, if the sample is consistent with a single normal distribution, there is little point in trying to fit a sum of two non–identical distributions. This procedure also gives the option of plotting a histogram and then, having chosen the number

of bins required, it can create a curve fitting file either unweighted or weighted by the square root of the bin size. Unless a very large sample is under investigation and there are no empty bins an unweighted file should be created. Note in particular that, as the best-fit curve integrates to unity over the data range $(-\infty, \infty)$, the option to normalize the histogram to area 1 must also be chosen.

After reading in the data file `qnfit_data.tf6` the model file `qnfit_model.tf6` should be selected, and this contains the following definition for a sum of two normal distributions.

```
%
Sum of two normal pdfs
A = -(1/2)[(x - p(1))/p(2)]^2
B = -(1/2)[(x - p(4))/p(5)]^2
f(x) = {[1 - p(3)]exp(A)/p(2) + p(3)exp(B)/p(5)}/sqrt{2*pi)
%
1 equation
1 variable
5 parameters
%
begin{expression}
A = -0.5*[(x - p(1))/p(2)]^2
B = -0.5*[(x - p(4))/p(5)]^2
C = [1.0 - p(3)]*exp(a)/p(2) + p(3)*exp(b)/p(5)
f(1) = C/root2pi
end{expression}
%
```

The best fit results table follows.

Number	Low-Limit	High-Limit	Value	Std.Error	Lower95%cl	Upper95%cl	*p*
1	-5.0	0.0	-1.44100	0.172367	-1.88408	-0.99792	0.0004
2	0.1	5.0	1.05066	0.181518	0.58405	1.51726	0.0022
3	0.1	0.9	0.45929	0.061382	0.30150	0.61707	0.0007
4	0.0	5.0	1.96743	0.133634	1.62392	2.31095	0.0000
5	0.1	5.0	0.78877	0.135524	0.44039	1.13714	0.0021

It might be required to plot the best-fit curve superimposed on the sample histogram and the following steps are required to do this.

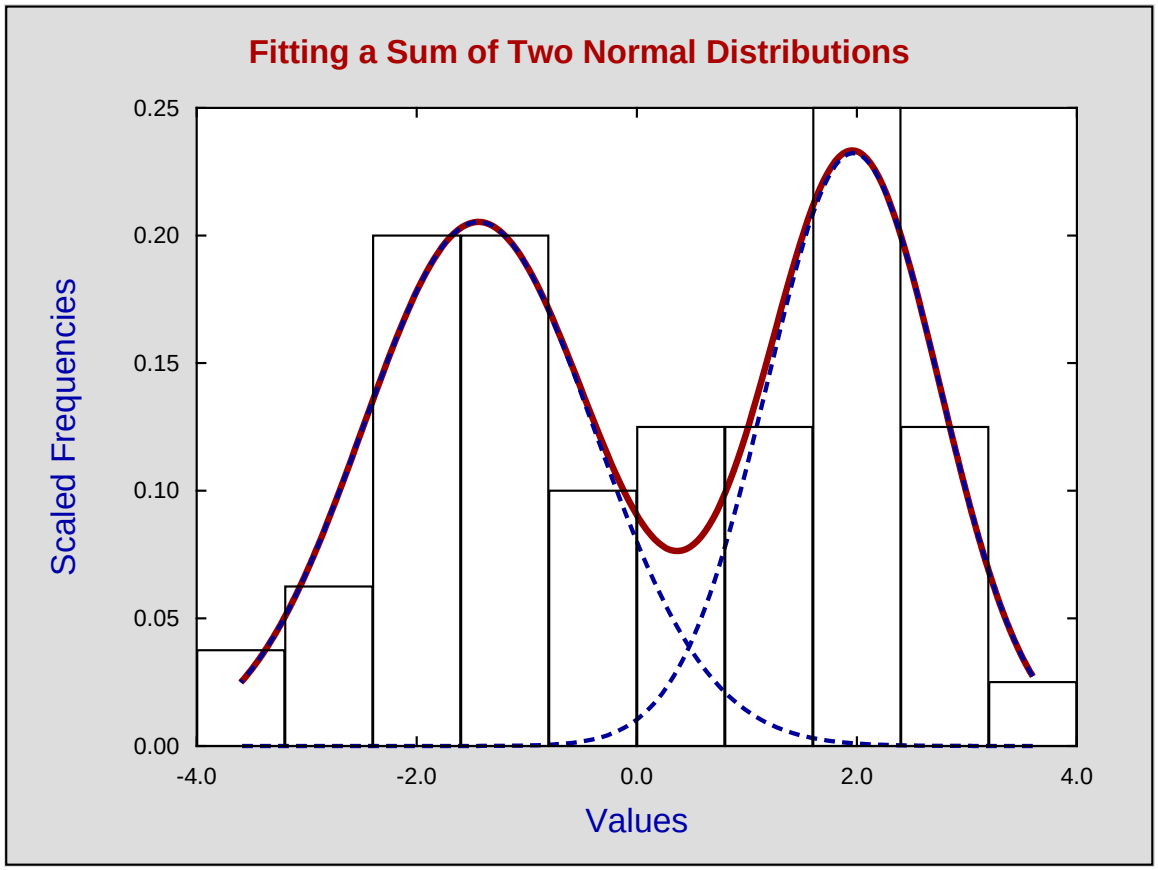
1. Request a plot in the usual way then choose [Advanced] and transfer to advanced editing.

2. The plot displayed will have symbols for the mid–points of the histogram which need to be changed.

3. From the [Data] options choose to plot bars instead of symbols.

4. The bar type, fill–style, color, and width can be altered if required.

A typical plot resulting from this editing is shown next and clearly shows that, with such dense and well–separated accurate data, a reasonable fit has been achieved. The profile of the two contributing sub–groups was obtained by using the SIMFIT library built–in equation instead of the model file and finally requesting graphical deconvolution



Example 7: Fitting a cumulative frequency

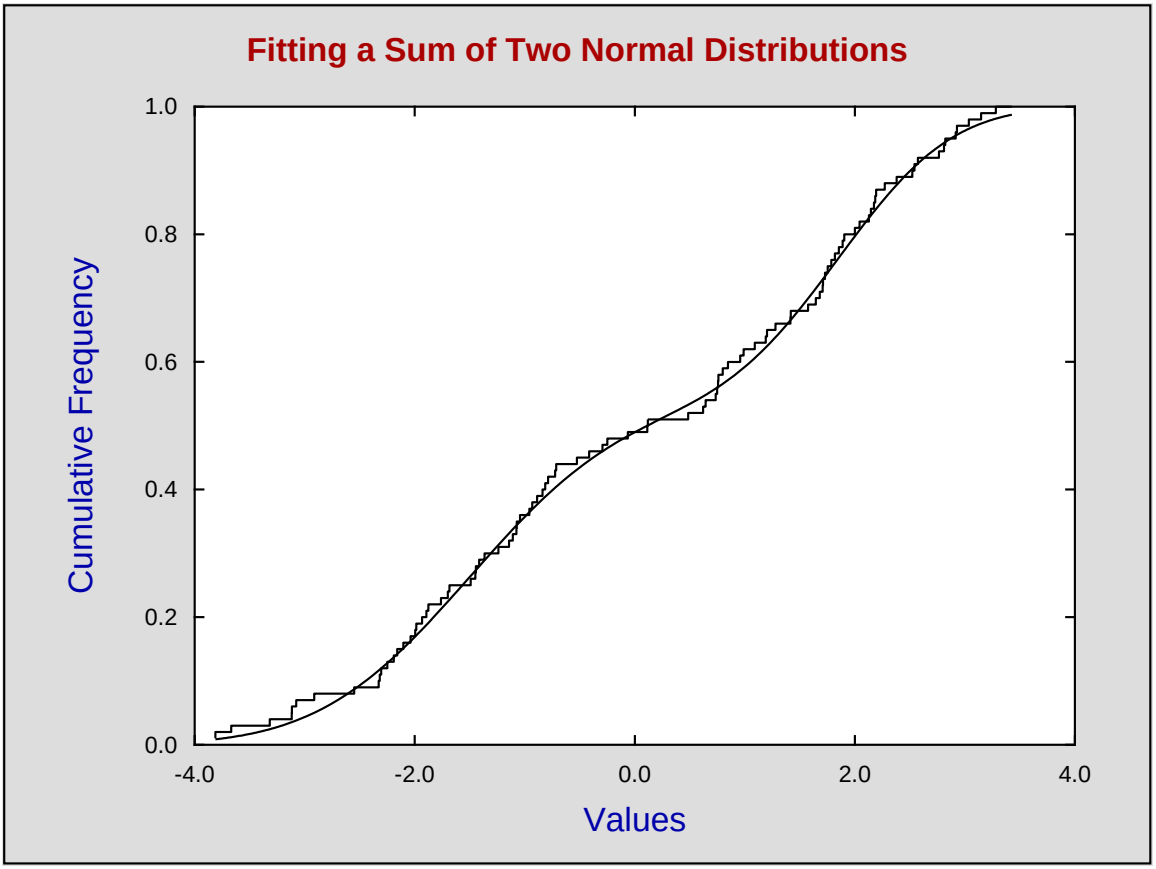
Often data are only available in partitioned form. For instance, counts from channels in flow cytometry are effectively in the form of histogram bins, so the analysis by fitting *pdfs* is all that is possible despite the fact that the results will depend on the number of bins. However, when a sample is available it is possible to fit a sum of two normal *cdfs* as discussed next, and this does not depend on partitioning into bins.

Read test file `normal.tf3` into the exhaustive analysis of a vector procedure exactly as with Example 6 but this time choose to export a *cdf* type curve fitting file. This test file is called `qnfit_data.tf7` and the model file `qnfit_model.tf7` created using SIMFIT **usermod** is as below.

```
%
Sum of two normal distributions
p(3)Phi((x - p(1))/p(2)) + (1 - p(3))Phi((x - p(4))/p(5))
%
1 equation
1 variable
5 parameters
%
begin{expression}
A = p(3)normalcdf((x - p(1))/p(2))
B = (1.0 - p(3))normalcdf((x - p(4))/p(5))
f(1) = A + B
end{expression}
%
```

The table of parameter estimates is displayed next followed by a plot of the data with best–fit curve.

Number	Low-Limit	High-Limit	Value	Std.Error	Lower95%cl	Upper95%cl	*p*
1	-5.0	0.0	-1.49012	0.034982	-1.55957	-1.42067	0.0000
2	0.1	5.0	1.08482	0.036551	1.01226	1.15738	0.0000
3	0.1	0.9	0.52956	0.010863	0.50799	0.55112	0.0000
4	0.0	5.0	1.85840	0.028793	1.80124	1.91556	0.0000
5	0.1	5.0	0.81238	0.030455	0.75192	0.87284	0.0000



To compare the results from fitting the *pdfs* and *cdfs* we can define the sums of squares SSQ between the parameter estimates $\hat{p}_i$ and the population parameters p_i as

$$SSQ = \sum_{i=1}^{5} (\hat{p}_i - p_i)^2$$

and note that

for the *pdfs*: $SSQ = 0.271$, and $\sqrt{SSQ} = 0.520$, while
for the *cdfs*: $SSQ = 0.171$, and $\sqrt{SSQ} = 0.414$

a slightly better result from fitting the *cdfs*.

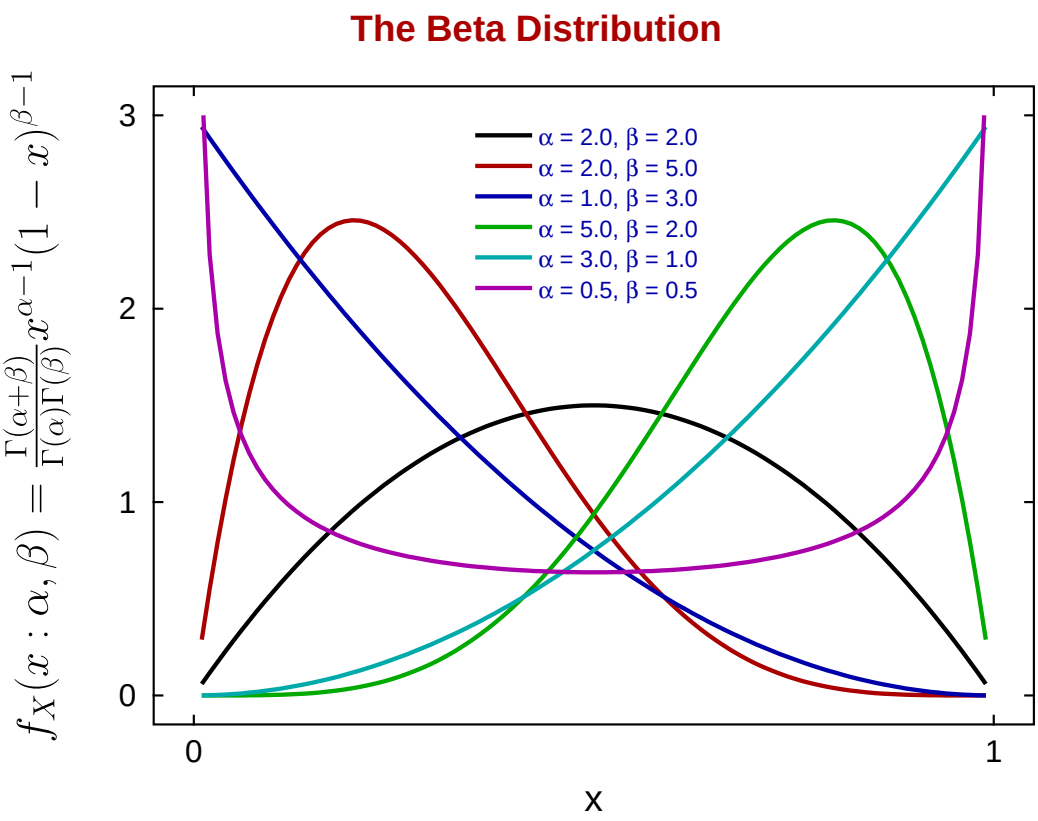
In order to succeed in estimating convincing parameter estimates there must be a very large sample with well–separated means, similar variances that do not cause too much overlap, and approximately equally sized sub–groups. Then fitting a *cdf* will give a unique set of parameter estimates as opposed to the way that fitting *pdfs* is dependent on the number of bins, but a visual display of the contribution by sub–groups is perhaps easier judged by superimposing a best fit curve on a histogram.

8.6.5 Fitting a beta distribution to a sample of observations

A random variable X $(0 \leq x \leq 1)$ with the following pdf $f_X(x : \alpha, \beta)$ and cdf $F_X(x : \alpha, \beta)$

$$f_X(x : \alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1}$$
$$F_X(x : \alpha, \beta) = \int_0^x f_X(t : \alpha, \beta)\, dt$$
$$= I_x(\alpha, \beta)$$

with parameters $\alpha > 0$ and $\beta > 0$, where $I_x(\alpha, \beta)$ is the regularized incomplete beta distribution, is referred to as a beta random variable. The widespread use of this distribution in data analysis arises not because many experimental observations do actually arise from a beta distribution, but because it is often a convenient unimodal distribution that serves well as an approximation in many situations, such as those involving the estimation of proportions. Some idea of the variation in the profile of a beta distribution as a function of the shape parameters α and β will be clear for the next figure.



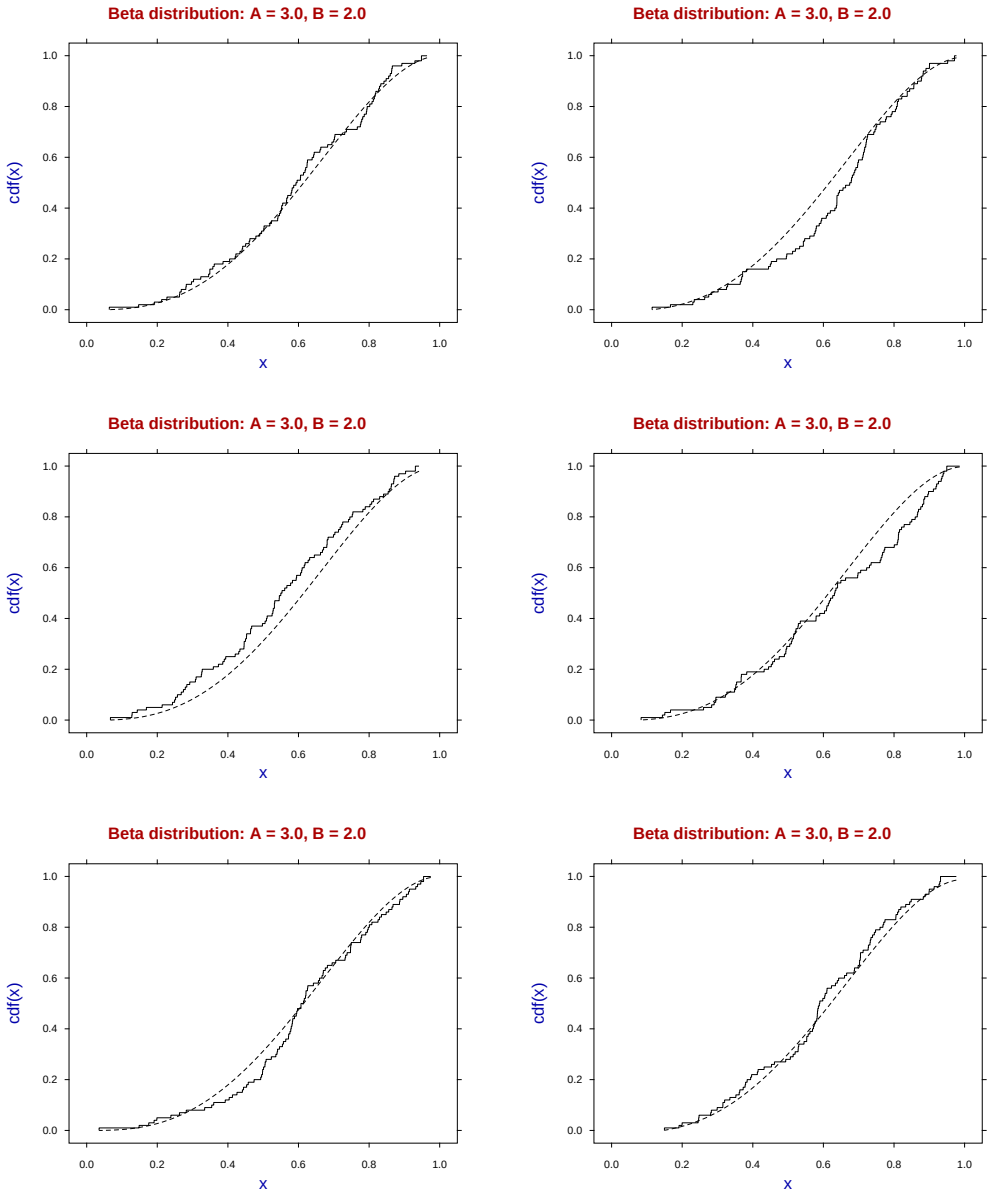
The wide variation in shape that is possible is what makes this a valuable empirical model for fitting arbitrary data that can be projected into the interval (0,1) in order to estimate and visualize skew and kurtosis. In addition the inversion of shape leading to poles at the extremes is often useful in some situations. This document explains how to use SimFIT to simulate pseudo–random beta variables and fit a beta distribution to observations using constrained weighted least squares in order to estimate goodness of fit.

Generating random samples

When fitting a specified probability distribution to a sample of observations it is valuable first to simulate random samples for the distribution, then observe how the values for random observations change as the

parameters vary. Random samples can then be plotted as histograms or cumulative distributions to get a feel as to how well your data can be modeled by the distribution and what are likely to be reasonable parameters.

From the main SimFIT menu choose [Simulate] then [Generate random numbers and walks] which opens up program **rannum**, and then select to generate sequences of random numbers for the stated distribution and parameters. As an example, consider the following six samples with 100 observations using a beta distribution with $\alpha = 3$ and $\beta = 2$ which demonstrates some rather surprising issues.



It will be seen that, even with a fairly large sample, it is possible to get seemingly large systematic deviations (from a Kolomogorov–Smirnov perspective) between the sample cumulative distribution (solid step curve) and the theoretical distribution (dotted curves) due to the unavoidable pseudo serial correlation in the sample cumulative. This must be kept in mind when assessing the goodness of fit by subjective graphical inspection of this type.

Of course things are no better with displaying the theoretical PDF overlayed on a histograms, as histogram shape depends on the number of bins chosen. At this point it should be noted that the data exploration option

in program **simstat** allows users to examine such PDF and CDF overlays for chosen distributions using any sample of observations.

Parameter estimation for statistical distributions

Before describing methods to estimate parameters for selected statistical distributions, such as the beta distribution from samples of observations, three points should be considered.

1. Experimental observations do not often follow statistical distributions exactly, rather distributions are assumed for convenience. For instance, the distribution of biological variables such as height, weight, blood pressure, etc., in populations are often analyzed as if the data followed a Gaussian distribution, which may appear reasonable in practise but is impossible mathematically, because the Gaussian distribution assumes $-\infty \leq X \leq \infty$.

2. Mathematical statistics is based on such precisely defined variables but everything that is measured experimentally has unavoidable observational error in addition to natural variation.

3. Many methods for parameter estimation depend on sample moments and it is well know that, apart from perhaps the first moment in some situations, higher moments are themselves parameter estimates with large variances, that is, are very innacurate.

For such reasons there is something to be said for estimating parameters by constrained nonlinear regression which offers the possibility of calculating parameter confidence limits and assessing goodness of fit by residuals analysis. That is, arranging a single sample of observations into a form suitable for fitting a statistical distribution as if it were a model for constrained weighted least squares fitting. Usually this means fitting a PDF to a histogram, or a CDF to a sample cumulative. If a very large sample is available, or observations are only available as already partitioned into bins, then fitting a histogram could be considered, as long as it is realized that the result will depend on the number of bins chosen.

Preparing samples of observations for curve fitting

Data must be available as vector, that is, a single column of values with no labels or missing values, and then this is input into the SIMFIT program for exhaustive analysis of a sample. This can be opened from the main SIMFIT menu by choosing [Data exploration]. There are then two options that will prove useful, and both are available using program **rannum**.

1. **Exhaustive analysis of an arbitrary vector**

   This allows you to create a PDF file for fitting by choosing the number of bins required then creating a PDF curve fitting file where the histogram area is scaled to one. Alternatively you can create a CDF curve fitting file. From this procedure you can also calculate the sample moments if these are needed to estimate starting estimates.

2. **Comparing data with a known distribution**

   A distribution is chosen then the parameters are varied until a reasonable fit is apparent when the data are displayed as a PDF–histogram or CDF–cumulative plot. The values chosen can then be used as starting estimates.

Another issue that is often considered is the minimum sample size that is required to begin to justify concluding that a specified distribution with the estimate parameters does reasonably represent the data. Rules of thumb such as . . . *at least ten times the number of parameters* . . . or similar are often suggested which would mean 20 for the beta distribution, but experience indicates a minimum sample size of about 100. So we now turn to a worked example using a beta distribution with a sample size of 100 and $\alpha = 3$ and $\beta = 2$, where the mode is shifted slightly to the right.

Example 1: Fitting a beta pdf

A random sample contained in the file `beta32_data.tf1` was generated by program **rannum** then transformed into a pdf–fitting histogram file with area one by the option to perform exhaustive analysis of a vector, leading to the curve–fitting data file `beta32_pdf.tf1` shown below.

```
beta pdf fitting file generated by RANNUM: A = 3, B = 2
10    3
1.8144613E-01    6.0634121E-01    1
2.6390795E-01    8.4887769E-01    1
3.4636977E-01    7.2760945E-01    1
4.2883159E-01    1.8190236E+00    1
5.1129342E-01    1.8190236E+00    1
5.9375524E-01    1.2126824E+00    1
6.7621706E-01    1.3339507E+00    1
7.5867888E-01    1.3339507E+00    1
8.4114070E-01    1.6977554E+00    1
9.2360252E-01    7.2760945E-01    1
begin{limits}
1    1    5
1    1    5
end{limits}
```

The first column contains the centers of the ten histogram bins, and the second column contains the scaled frequencies, while the third column (with weights equal to one) indicates that unweighted fitting is to be used.

The section starting with the token `begin{limits}` and ending with the token `end{limits}` gives the lower limits, the starting estimates, then the upper limits to be used by program **qnfit** for constrained nonlinear regression in the EXPERT mode, i.e. where such estimates are appended to the data file.

The results from fitting by the SIMFIT quasi–Newton constrained optimization technique are shown next.

Number	Low-Limit	High-Limit	Value	Std.Error	Lower95%cl	Upper95%cl	p
1	1	5	2.26957	0.454304	1.22195	3.31720	0.0011
2	1	5	1.70249	0.301995	1.00609	2.39889	0.0005
3	1	1	1.00000	0.000000	1.00000	1.00000	fixed

For 50,90,95,99% con. lim. using [parameter value +/- t(alpha/2)*std.err.]
t(.25) = 0.706, t(.05) = 1.860, t(.025) = 2.306, t(.005) = 3.355

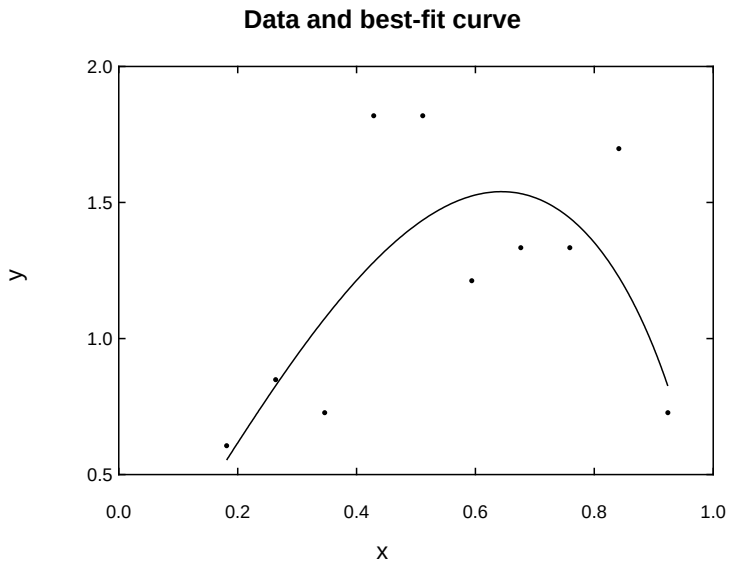
Note that the model used by SIMFIT has the following parameter definitions.

$$p(1) = \alpha$$
$$p(2) = \beta$$
$$p(3) = \Delta$$

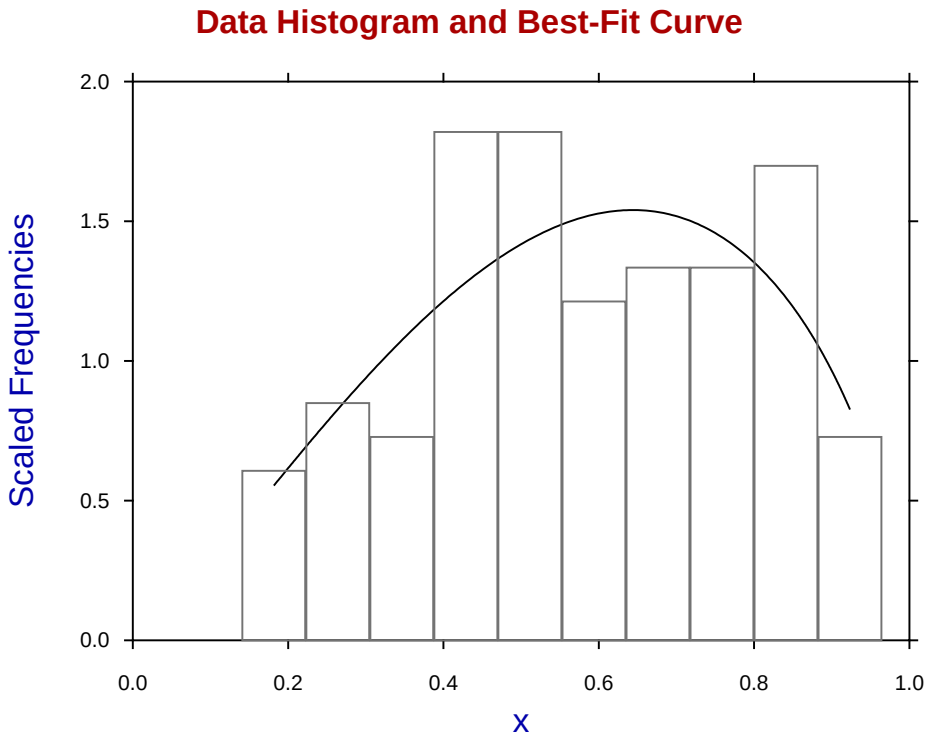
where Δ is a scaling factor than can be used if the area under the histogram is not one. For this fit $p(3)$ was not varied but was fixed, i.e., $\Delta = 1$.

After listing all the goodness of fit results program **qnfit** first shows a default graph where the tops of the histogram bins are shown as dots and the best–fit curve is displayed as a smooth curve ranging between the the centers of the first and last histogram bins.

This default graph from program **qnfit** is shown next.



Here is the default graph after editing to replace the dots by outline type histogram bars width 1.47, and other obvious changes, to give the next graph.



It is possible to create such graphs with many more possible options by saving the best–fit curve parameters, then reading the data into the Data Exploration option of program **simstat** to create the histogram overlayed by the pdf for a beta distribution with the best-fit parameters over the full range, etc.

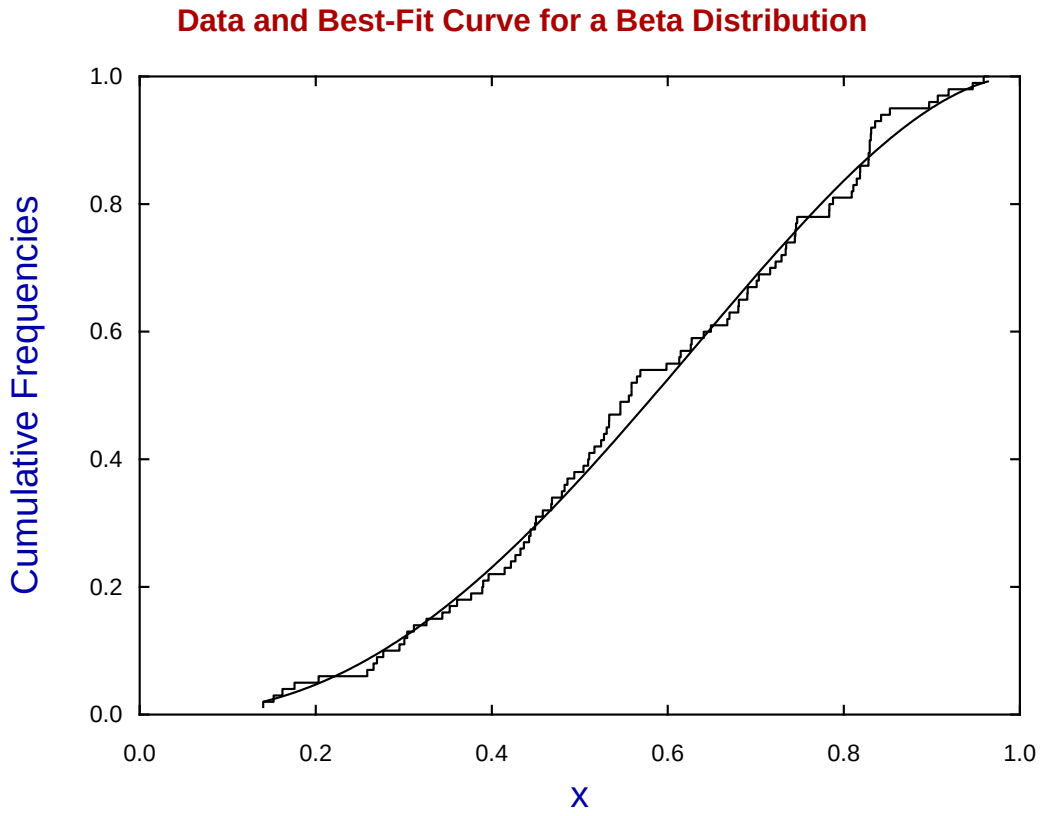
Example 2: Fitting a beta cdf

Proceeding as before then fitting a beta cdf to the data file `beta32_cdf.tf1` using program **qnfit** yields these parameter estimates.

Number	Low-Limit	High-Limit	Value	Std.Error	Lower95%cl	Upper95%cl	p
1	1	5	2.52779E+00	5.57879E-02	2.41708E+00	2.63850E+00	0.0000
2	1	5	1.88851E+00	4.03438E-02	1.80845E+00	1.96857E+00	0.0000
3	1	1	1.00000E+00	0.00000E+00	1.00000E+00	1.00000E+00	fixed

For 50,90,95,99% con. lim. using [parameter value +/- t(alpha/2)*std.err.]
t(.25) = 0.677, t(.05) = 1.661, t(.025) = 1.984, t(.005) = 2.627

The following best-fit curve was edited by simply replacing the default plotting symbols (dots with no lines) for the data by no symbols but a cdf-type step curve.



It is clear that fitting such simple models with just two varied parameters gives well–defined parameter estimates ($p = 0$) but fitting the cdf using all 100 points gives better estimates than fitting to a histogram which only fits ten points.

To quantify this observation, the procedure of data generation by program **rannum** followed by fitting using program **qnfit** was repeated, and the Euclidean distance D between the estimates $(\hat{\alpha}, \hat{\beta})$ and the actual parameter values (α, β) was calculated, where D is defined as follows,

$$D = \sqrt{(\alpha - \hat{\alpha})^2 + (\beta - \hat{\beta})^2}.$$

$\hat{\alpha}$	$\hat{\beta}$	D	Type	Best Fit
2.38518	1.84594	0.633828	pdf: $\alpha = 3, \beta = 2$	
2.52170	2.06080	0.482149	cdf: $\alpha = 3, \beta = 2$	cdf
2.60811	1.65327	0.523259	pdf: $\alpha = 3, \beta = 2$	
2.63019	1.76255	0.439479	cdf: $\alpha = 3, \beta = 2$	cdf
2.26957	1.70249	0.788695	pdf: $\alpha = 3, \beta = 2$	
2.52799	1.88851	0.484998	cdf: $\alpha = 3, \beta = 2$	cdf
1.94617	3.97530	0.059226	pdf: $\alpha = 2, \beta = 4$	
1.85620	3.86457	0.197534	cdf: $\alpha = 2, \beta = 4$	pdf
1.64910	3.79293	0.407442	pdf: $\alpha = 2, \beta = 4$	
1.67319	4.03689	0.328885	cdf: $\alpha = 2, \beta = 4$	cdf
2.45517	4.91705	1.023797	pdf: $\alpha = 2, \beta = 4$	
2.37186	4.82599	0.905836	cdf: $\alpha = 2, \beta = 4$	cdf
2.13494	7.54346	0.476065	pdf: $\alpha = 2, \beta = 8$	
2.12449	8.17192	0.212260	cdf: $\alpha = 2, \beta = 8$	cdf

From this table, where both the pdf and cdf were fitted to the same data set as both histograms (observations pooled into 10 bins) and cumulative frequencies (all 100 observations) for a total of seven separate simulations, a number of tentative conclusions can be drawn.

- The parameters were estimated rather better using the data in cumulative distribution format.
- There is a tendency to underestimate the parameters.

Although not shown, there is an improvement in parameter estimates when the additional normalizing parameter is allowed to vary, more so with histograms of course. However there are other ways to decide which technique to use.

Example 3: Plotting a combined graph

Often beta distributions are plotted simply to estimate the extent to which the mode is skewed away from the central position, and this is most convincingly seen in histograms as long as the number of bins is not too large.

So, as there are only two, or rarely three parameters to be fitted and the beta distribution is robust as an empirical model and easy to fit to a sample of observations, there seems no reason why both should not be fitted at the same time.

For a combined graph from such a fitting procedure there are two considerations.

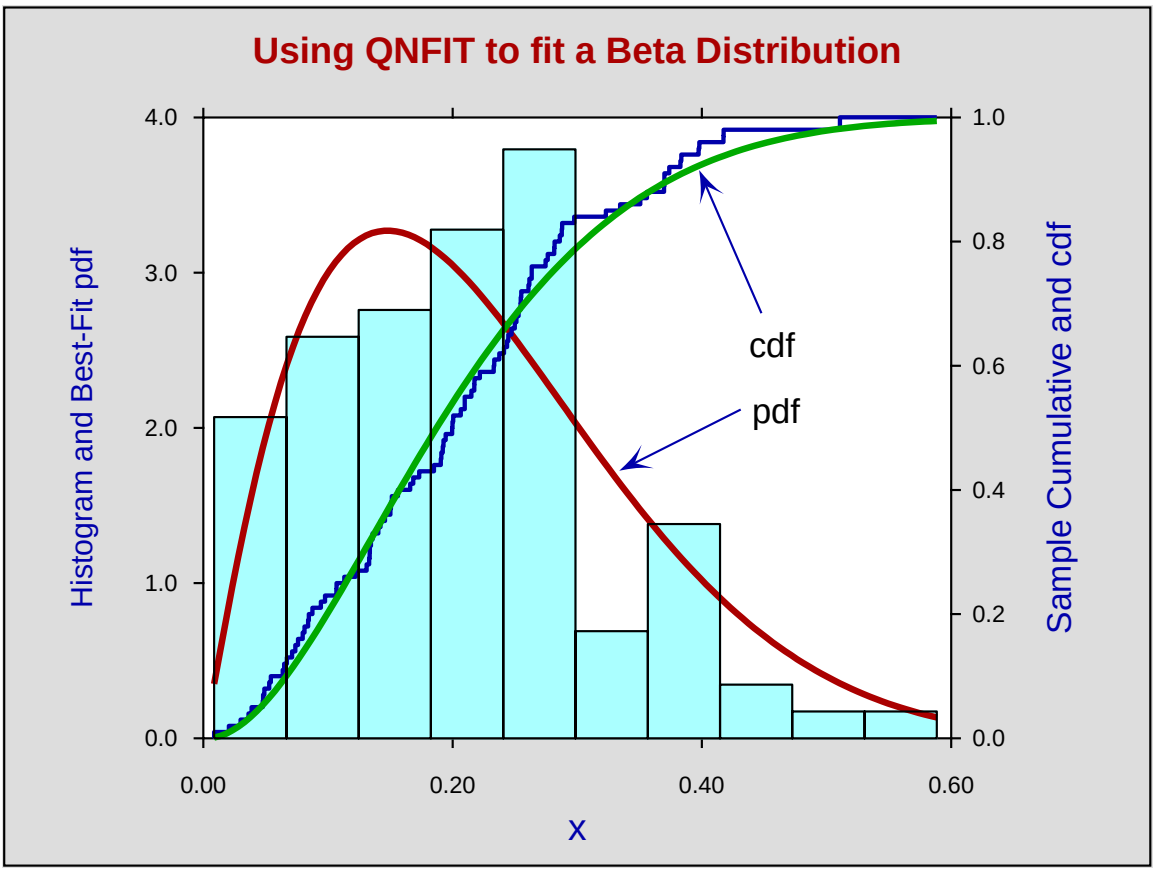
1. Four files of coordinates are required, that is:
   - File 1: coordinates for the histogram;
   - File 2: coordinates for the best–fit pdf;
   - File 3: coordinates for the sample cumulatives; and
   - File 4: coordinates for the best-fit cdf.
2. Two scales are required for the vertical axes, such as:
   - plotting the histogram and pdf using a left–hand scale; and
   - plotting the sample cumulative and cdf using a right–hand scale.

There are several methods by which this process can be done. Perhaps the most obvious is to save the coordinate files from the graphs of data and best fit graphs displayed by program **qnfit**, but this is not necessarily the best way.

Probably the best and easiest SIMFIT technique do this, for instance, for a beta distribution like the one from the previous table with $\alpha = 2, \beta = 8$, is as follows.

1. Open the option in the SIMFIT program **simstat** to compare a sample with an assumed distribution.
2. Read in the data and construct a histogram plotted against a beta distribution with best–fit parameters $\hat{\alpha} = 2.13494, \hat{\beta} = 7.54346$.
3. From this save the coordinates to File 1 and File 2.
4. Now construct a cumulative distribution stair-step type plot with added cdf with best–fit parameters $\hat{\alpha} = 2.12449, \hat{\beta} = 8.17192$.
5. From this save the coordinates to File 3 and File 4.
6. Open program **simplot** then choose two create a double axis plot and read in File 1 and File 2 to plot against the left–hand Y-axis, then File 3 and File 4 for the right–hand axis.

All that remains is fine tuning to create the following plot.



It should be noted that plotting symbols can be replaced by filled polygons, but if these are filled with color the histogram bin outlines will be lost. This can be overcome by using the same file (File 1) added interactively as an additional file (File 5) used to outline the resulting filled polygons. Alternatively, if this situation is anticipated, an additional copy of File 1 containing the histogram outlines can be added right from the start.

Practical issues

As the shape of the data will be evident before any computation of best–fit parameters, then visual inspection helps in the choice of starting estimates and limits.

Writing the beta probability density function, i.e. the PDF, in the following form

$$f_x(x : \alpha, \beta) = \frac{1}{B(\alpha, \beta)} x^{\alpha-1}(1-x)^{\beta-1}$$

emphasizes that, as the complete beta function itself, i.e., $B(\alpha, \beta)$ is a constant and not dependent on x, the graphical behaviour of this density function for $0 \leq x \leq 1$ depends only on the expression

$$x^{\alpha-1}(1-x)^{\beta-1}$$

so there is a single turning point for non-degenerate cases at the mode M where

$$M = \frac{\alpha - 1}{\alpha + \beta - 2}.$$

As $M = 0.5$ when $\alpha = \beta$, while $M > 0.5$ if $\alpha > \beta$, and $M < 0.5$ if $\alpha < \beta$, the displacement of the mode from 0.5 indicates the relative magnitude of α and β. Of course the degenerate case when $\alpha = \beta = 1$ corresponding to a uniform distribution, the complications due to vertical asymptotes when $x = 0$ for $\alpha < 1$ and $x = 1$ for $\beta < 1$, along with the general inversion of shape when $\alpha < 1$ and $\beta < 1$ must be considered. As the general shape would be indicated by the data then this means it is easy to decide on the lower limits of 1 when $\alpha > 1$ and $\beta > 1$ and upper limits of 1 when $\alpha < 1$ and $\beta < 1$.

There are two other practical issues to consider when fitting the beta distribution to observations.

1. **The range of x values**
   In the cases where $\alpha > 1$ and $\beta > 1$ then there is no restriction of range and observations can be anywhere between $x = 0$ and $x = 1$. However, if vertical asymptotes are anticipated, then values must be restricted near potential asymptotes so that computation does not lead to overflow.

2. **The parameter limits**
   As computation of best–fit parameters proceeds then, at every fixed value of x, the values of the internal estimates $\hat{\alpha}$ and $\hat{\beta}$ are perturbed by factors of the order of machine precision. So the upper and lower limits should normally be chosen such that singular cases are avoided.

   Another issue concerns the evaluation of the complete beta function for non–integer arguments. As α and β become larger then the time taken to evaluate the complete beta function increases very rapidly. Of course the computer code doing this is is optimized, but is still faced with such limitations. So it is recommended that the upper limits requested for parameter estimates should be selected conservatively with this in mind to avoid lengthy computations.

8.6.6 Graphical deconvolution

It is often necessary to fit models that are sums of sub–models, and some way of detecting the minimum number of sub–models required to explain the data is needed. Statistical tests like a sequential F test can help, but it is also useful to visualize the contribution of sub–models by plotting the sub–models at the same time as the best–fit curve. In SimFIT this technique is loosely referred to as graphical deconvolution. In short, given a model of the form

$$f(x,p) = \sum_{i=1}^{n} f_i(x,p)$$

where x is a vector of user–supplied independent variables and p is a vector of parameters to be estimated, then how should we attempt to calculate the contribution of sub–models f_i to the overall best–fit model, and thereby decide upon the minimum acceptable value for n, i.e., the minimum number of sub–models required.

As a typical example, consider the situation where a large sample is available and it is wished to fit a sequence of sums of normal distribution $cdfs$ as follows.

$$cdf_n(x) = \frac{p_1}{p_{2n+1}\sqrt{2\pi}} \int_{-\infty}^{x} \exp\left(-\frac{1}{2}\left\{\frac{u-p_{n+1}}{p_{2n+1}}\right\}^2\right) du + \frac{p_2}{p_{2n+2}\sqrt{2\pi}} \int_{-\infty}^{x} \exp\left(-\frac{1}{2}\left\{\frac{u-p_{n+2}}{p_{2n+2}}\right\}^2\right) du + \cdots$$
$$+ \frac{p_n}{p_{3n}\sqrt{2\pi}} \int_{-\infty}^{x} \exp\left(-\frac{1}{2}\left\{\frac{u-p_{2n}}{p_{3n}}\right\}^2\right) du + p_{3n+1}$$

However, particularly with machine–generated data, a histogram often has to be fitted using $pdfs$ in this form.

$$pdf_n(x) = \frac{p_1}{p_{2n+1}\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left\{\frac{x-p_{n+1}}{p_{2n+1}}\right\}^2\right) + \frac{p_2}{p_{2n+2}\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left\{\frac{x-p_{n+2}}{p_{2n+2}}\right\}^2\right) + \cdots$$
$$+ \frac{p_n}{p_{3n}\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left\{\frac{x-p_{2n}}{p_{3n}}\right\}^2\right) + p_{3n+1}$$

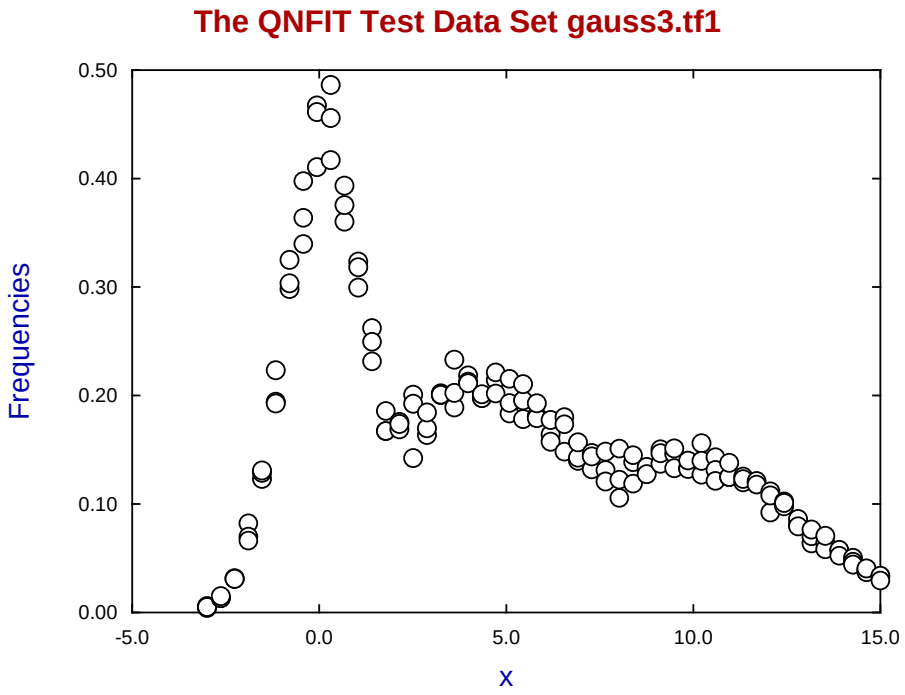
For a sum of n such sub–models then up to $3n+1$ parameters have to be estimated, namely

- $p_1, p_2, \ldots, p_n$ are positive partitioning fractions, which may be constrained, e.g., $\sum_{i=1}^{n} p_i = 1$
- $p_{n+1}, p_{n+2}, \ldots, p_{2n}$ are means of arbitrary sign,
- $p_{2n+1}, p_{2n+2}, \ldots, p_{3n}$ are positive standard deviations, and
- p_{3n+1} is an arbitrary background correction factor that is sometimes required.

Several points should be noted about fitting this model.

1. Fitting the cdf uses a data set that is unique up to order, but the shape is somewhat featureless making it difficult to guess parameter starting estimates and limits.
2. Fitting the pdf is not unique as it depends on the number of histogram bins but, when the peaks are well–separated, it is easier to guess parameters starting estimates and assess goodness of fit visually.
3. The numbering of sub–models is arbitrary as regards permutations, so the starting estimates and parameter limits must be chosen with considerable care to limit the search directions for minimizing the objective function in order to avoid ambiguity. This consideration also applies when fitting similar models like sums of saturation functions or exponentials.

However, models such as these are not always used for statistical analysis of a mixed sample but rather as simple empirical models in an attempt to resolve the contribution of individual signals to an overall profile. So, in order to illustrate this procedure in such an application, we shall consider the data set contained in the SimFIT test file `gauss3.tf1` which contains triplicates as illustrated next.



From inspecting this profile we see that there are at least three distributions involved. These appear to make similar contributions so we could guess that

p_1, p_2 and p_3 would be of order unity, while
$0 \approx p_4 << p_5 << p_6 << 12$,
$1 \approx p_7 << p_8 << p_9 << 5$, and it would be safe to fix the constant term so that
$p_{10} = 0$.

So attached to the end of `gauss3.tf1` will be found these limits and starting estimates, allowing this data set to be fitted in EXPERT mode where such estimates and limits are read from the data file, and which greatly facilitates fitting by SIMFIT program **qnfit**.

```
begin{limits}
  0,   0.5,   2
  0,   1.5,   2
  0,   0.5,   2
 -2,   0.0,   2
  2,   4.0,   6
  8,  10.0,  12
0.1,   1.0,   2
  1,   2.0,   3
  2,   3.0,   4
  0,   0.0,   0
end{limits}
```

To appreciate how such an analysis would be conducted proceed as follows.

How to plot a graphical deconvolution

1. Open SimFIT using either **w_simfit.exe** for the 32-bit version, or **x64_simfit.exe** for the 64-bit version.
2. From the main SimFIT menu press the [A/Z] option.
3. Scroll down the list displayed and open program **qnfit**.
4. Accept the default options and select the option to fit one function of one variable.
5. Read in the test file called `gauss3.tf1` by pressing the [Demo] button on the file opening dialogue and scrolling down the list provided.
6. Choose the option to fit Gaussian *pdfs* (spikes) with no constant term.
7. When asked how many terms are required input a 3, i.e., choose to fit a sum of three terms.
8. Select the option to run in the EXPERT mode which reads starting estimates and limits off the data file `gauss3.tf1`.
9. Proceed to fitting.

A summary of the optimization procedure and preliminary comments about the goodness of fit are given and then the parameter estimates from fitting are displayed and output as text to the results file as below.

```
 Best-fit parameters for curve-fit   1 using LBFGSB

No.  Low-Limit High-Limit    Value        Std.Error   Lower95%cl   Upper95%cl    p
  1  0.000E+00  2.000E+00  9.07541E-01  2.16240E-02  8.64792E-01  9.50291E-01  0.0000
  2  0.000E+00  2.000E+00  1.16433E+00  4.21732E-02  1.08096E+00  1.24770E+00  0.0000
  3  0.000E+00  2.000E+00  9.25185E-01  3.01303E-02  8.65619E-01  9.84750E-01  0.0000
  4 -2.000E+00  2.000E+00 -7.29763E-02  1.55718E-02 -1.03761E-01 -4.21918E-02  0.0000
  5  2.000E+00  6.000E+00  3.74510E+00  5.08157E-02  3.64464E+00  3.84556E+00  0.0000
  6  8.000E+00  1.200E+01  1.02774E+01  9.64127E-02  1.00868E+01  1.04680E+01  0.0000
  7  1.000E-01  2.000E+00  9.26404E-01  1.43311E-02  8.98073E-01  9.54736E-01  0.0000
  8  1.000E+00  3.000E+00  2.34330E+00  7.05668E-02  2.20380E+00  2.48281E+00  0.0000
  9  2.000E+00  4.000E+00  2.76906E+00  6.26372E-02  2.64523E+00  2.89289E+00  0.0000
     parameter(10) is the excluded constant term
 For 50,90,95,99% con. lim. using [parameter value +/- t(alpha/2)*std.err.]
 t(.25) = 0.676, t(.05) = 1.656, t(.025) = 1.977, t(.005) = 2.611
```

Later, after using the [Results] option from the main SimFIT menu, these results can be extracted as in the following table for inclusion in documents.

Best-fit parameters for curve-fit 1 using LBFGSB

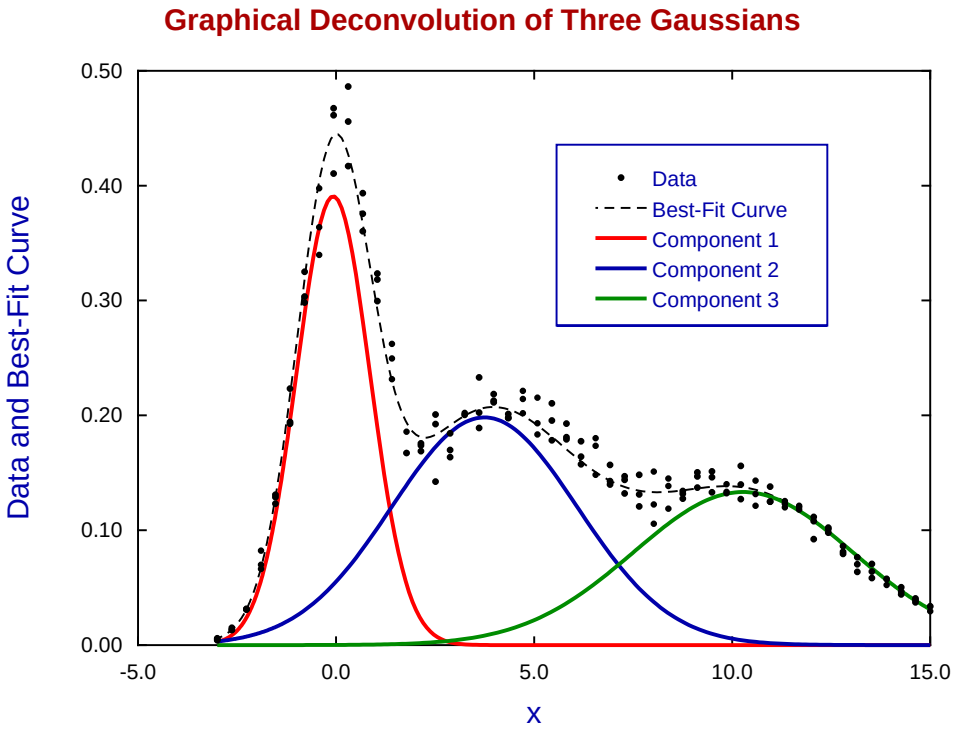
Number	Low-Limit	High-Limit	Value	Std.Error	Lower95%cl	Upper95%cl	*p*
1	0.0	2.0	0.90754	0.021624	0.86479	0.95029	0.0000
2	0.0	2.0	1.16433	0.042173	1.08096	1.24770	0.0000
3	0.0	2.0	0.92519	0.030130	0.86562	0.98475	0.0000
4	-2.0	2.0	-0.07298	0.015572	-0.10376	-0.04219	0.0000
5	2.0	6.0	3.74510	0.050816	3.64464	3.84556	0.0000
6	8.0	12	10.2774	0.096413	10.0868	10.4680	0.0000
7	0.1	2.0	0.92640	0.014331	0.89807	0.95474	0.0000
8	1.0	3.0	2.34330	0.070567	2.20380	2.48281	0.0000
9	2.0	4.0	2.76906	0.062637	2.64523	2.89289	0.0000

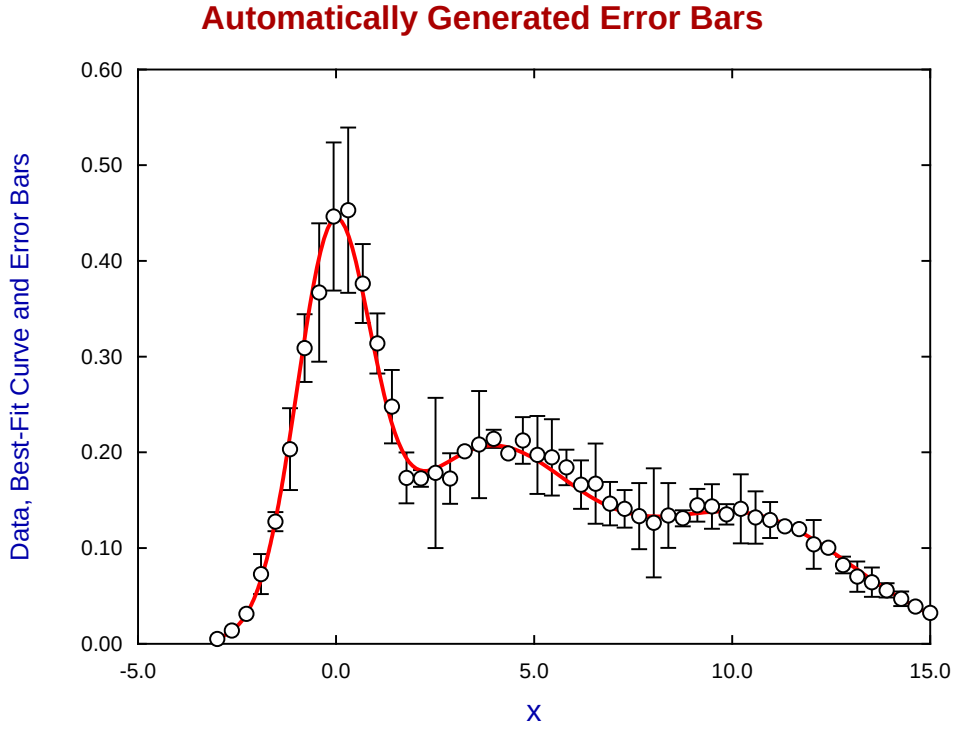
parameter(10) is the excluded constant term

For 50,90,95,99% con. lim. using [parameter value +/- t(α/2)*std.err.]

t(.25) = 0.676, t(.05) = 1.656, t(.025) = 1.977, t(.005) = 2.611

After proceeding through subsequent menus providing options for graphical display and goodness of fit analysis a final end–of–analysis menu is reached. From this the option to view the graphical deconvolution can be selected to create the next graph showing the data, best–fit curve, and individually contributing components, or the subsequent plot illustrating error bars.





8.6.7 Plotting contours of the objective function at solution points

Constrained nonlinear regression involves the attempt to locate a local minimum of some objective function which is a function of the variable parameters to be estimated given the fixed fixed data and weights supplied by the user.

In SIMFIT the following definitions are used

$$\begin{aligned} y_1, y_2, \ldots, y_n &: \text{ The observations} \\ x_1, x_2, \ldots, x_n &: \text{ Values of the independent variable} \\ w_1, w_2, \ldots, w_n &: \text{ The assumed weights} \\ \Theta = \theta_1, \theta_2, \ldots, \theta_m &: \text{ The parameters to be estimated} \\ g_1, g_2, \ldots, g_n &: \text{ The function values evaluted at } x, \Theta \end{aligned}$$

where it is assumed that, for errors ϵ,

$$y = g(x, \Theta) + \epsilon,$$

and the appropriate objective function to be minimized is the weighted sum of squared residuals divided by the number of degrees of freedom, that is

$$\begin{aligned} f(\Theta) &= \frac{WSSQ}{n-m} \\ &= \frac{1}{n-m} \sum_{i=1}^{n} w_i (y_i - g_i)^2. \end{aligned}$$

If the weights are reasonably accurate and the errors are normally distributed with zero means this should be an approximately chi-squared variable of order unity at the solution point and, in addition, if sensible starting estimates have been supplied the parameters should also be of order unity at the solution point.

Now the basis of constrained nonlinear regression is that the objective function should be approximately quadratic at the solution point, and that therefore a well-defined local minimum should have contours at the solution point that are approximately elliptical.

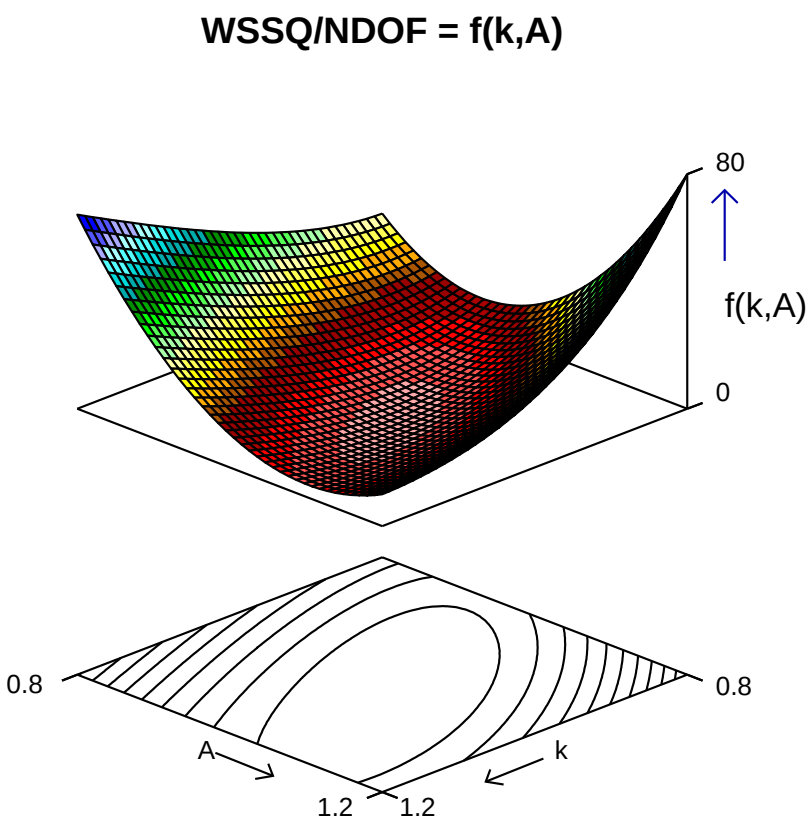
To demonstrate a case where this condition is realized then consider fitting the test data file `exfit.tf2` using a one-exponential model in the form

$$g(x, A, k) = A \exp(-kx)$$

using program **qnfit** and noting that starting estimates are indicated on this data file as follows

```
begin{limits}
0.0   0.5   3.0
0.0   1.5   3.0
0.0   0.0   0.0
end{limits}
```

so that the EXPERT mode can be used. After fitting, the option to view the objective function at the solution point can be chosen to obtain the following plot.



As a more complicated example consider fitting the data in test file `qnfit_data.tf4` using the model defined in `qnfit_model.tf4` which defines the double Michaelis-Menten model in the following way, first in parameter form then in the more familiar enzyme kinetic form

$$g(x,\Theta) = \frac{\theta_3 x}{\theta_1 + x} + \frac{\theta_4 x}{\theta_2 + x}$$
$$\equiv \frac{Vmax_1 S}{Km_1 + S} + \frac{Vmax_2 S}{Km_2 + S}.$$

Three important points should be noted.

1. **Complications with more than two parameters**

   There are now more than two parameters so the objective function and contours can be plotted in several ways to display the surface as a function of just two parameters.

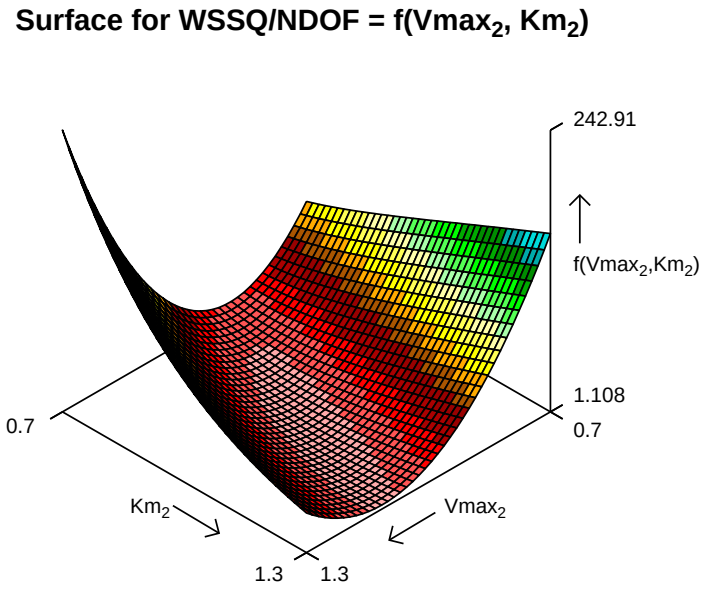
2. **Ambiguity with the model**

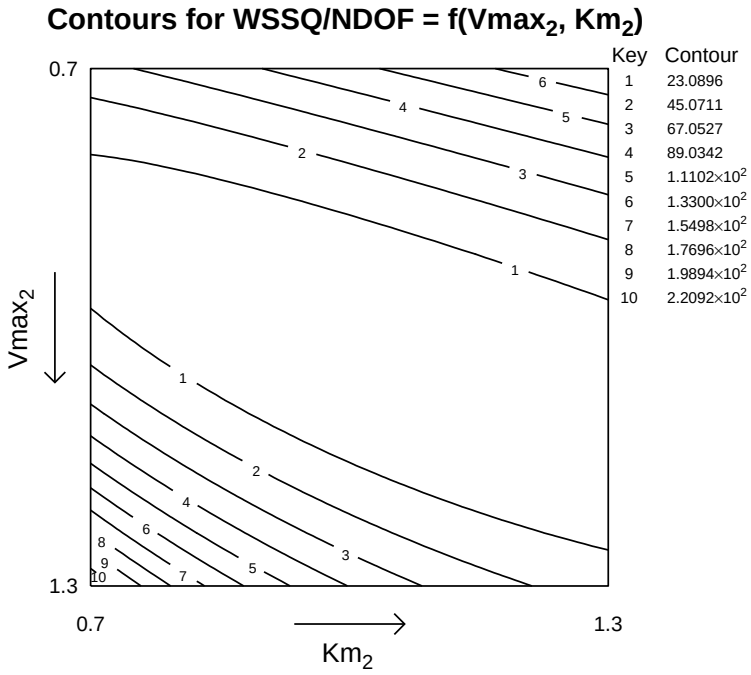
   The model is ambiguous in that the pairs $Vmax_1, Km_1$ and $Vmax_2, Km_2$ are not uniquely defined. In other words, the parameter estimates will refer to these pairs in arbitrary order, depending only on the starting estimates and limits set.

3. **Elongated valleys**

   The eccentricity of the elliptical contours at the solution point will be exaggerated if the order of magnitude of the parameters chosen and the range chosen to display the plots are not comparable.

The point is that the contours are stretched with the chosen pair of parameters as will be seen in the next plot, and this is what happens when more than two parameters are estimated namely, long valleys leading to the solution point makes convergence to the local minimum increasingly difficult.





This can also be confirmed by observing the eigenvalues and condition number of the internal Hessian matrix which, for this example, were as follows.

Eigenvalues of the internal Hessian matrix

5.07548E-01 1.45150E+01 1.88722E+02 3.44715E+03, Condition number = 6.79178E+03

The condition number is the ratio of largest to smallest eigenvalue, which reflects the ratio of the longest valley to the shortest. This value will clearly be distorted in cases where not all the estimated parameters are of comparable size in internal coordinates.

8.6.8 Contours with residuals and sections across a best fit surface

After fitting a function of two or more variables to a data set, the techniques for visualizing the best-fit model are more restricted than for cases with a single independent variable. To illustrate some possibilities, examples from fitting functions of two variables will be explained.

The first data set

For the first example, the data set contained in the SIMFIT test file `inhibit.tf1` will be used. This is for the case of enzyme kinetic data for a range of substrate concentrations at each of several fixed inhibitor concentrations. From the format of this data set it will be noted that the data are arranged for increasing sequences of S at each fixed inhibitor concentration. namely $I = 0, 1, 2, 3, 4$ and data should be arranged in such an ordered sequence to facilitate subsequent analysis.

The first model

This is taken from the library of enzyme kinetic models for two variables, namely the generalized inhibition model

$$f(x, y) = \frac{p_1 x}{p_2(1 + y/p_3) + x(1 + y/p_4)}$$

equivalent to the more familiar form

$$v(S, I) = \frac{VmaxS}{Km(1 + I/K_{is}) + S(1 + I/K_{ii})}.$$

The first technique: Plotting sections across a best-fit surface

Before the days of computers, nonlinear regression, and statistical analysis, this was cast in the form

$$\frac{1}{v} = \frac{Km(1 + I/K_{is})}{Vmax}\left(\frac{1}{S}\right) + \frac{(1 + I/K_{ii})}{Vmax}$$

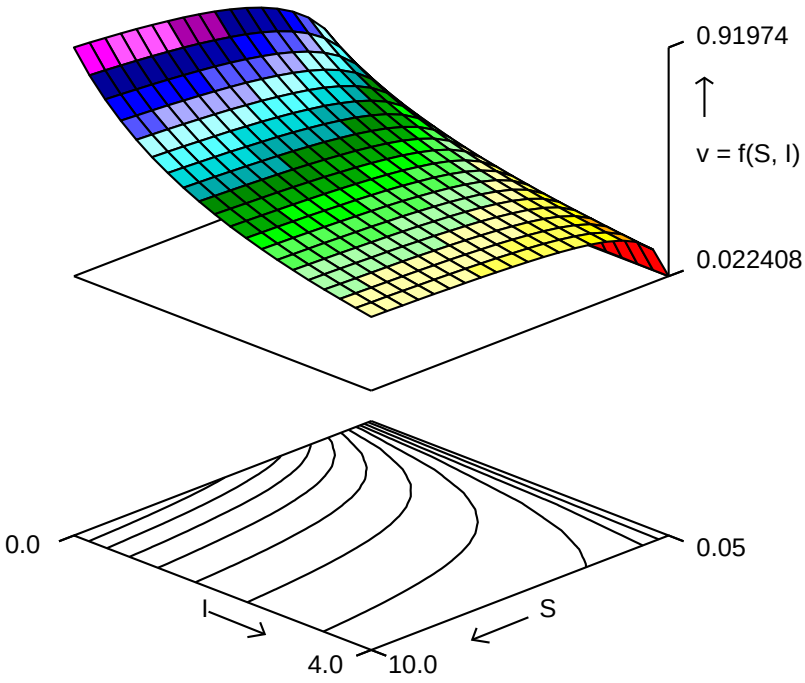
and analyzed by plotting double reciprocal plots then re–plotting the slopes and intercepts to estimate the inhibition constants from the observed slope and intercept effects.

To see the modern way to do this type of analysis, open program **qnfit** from the [A/Z] option on the main SIMFIT menu, choose to fit a function of two variables, read in the test file `inhibit.tf1` followed by the model just discussed from the library of statistical models for functions of two variables, then fit in the EXPERT mode. This reads in the following starting estimates and limits appended to the data file

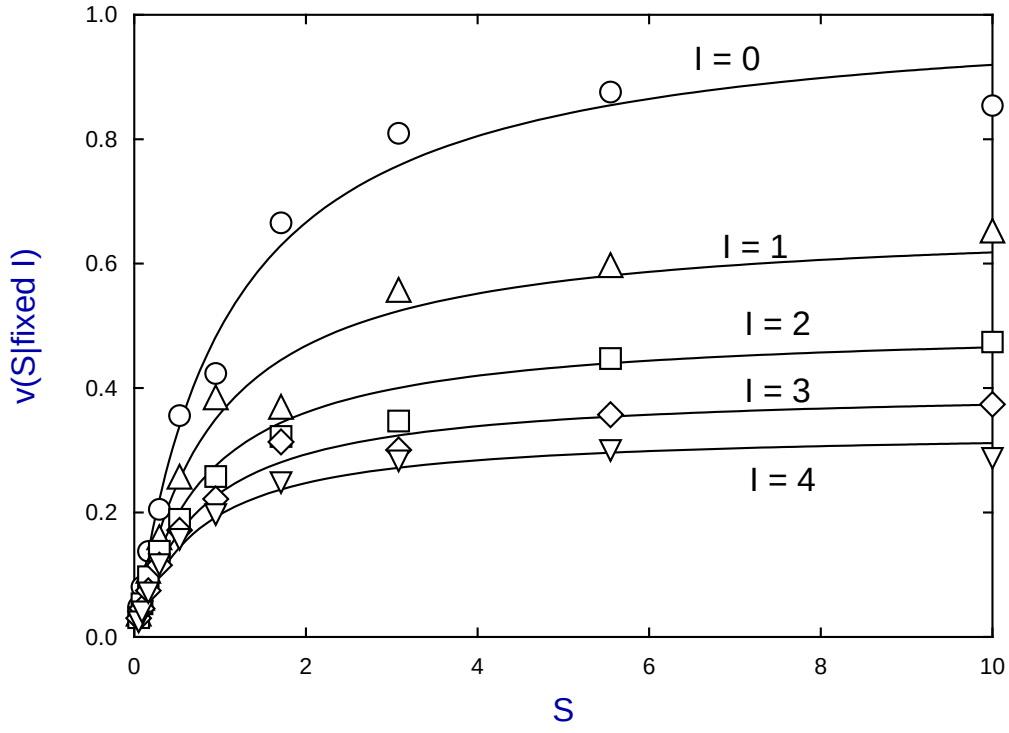
```
begin{limits}
0.0   0.5   5.0
0.0   1.5   5.0
0.0   5.0   9.0
0.0   1.0   5.0
end{limits}
```

and provides the options to display the best-fit surface and the sections across the best-fit surface for functions of S at fixed I, which simply requires accumulating the sections into your project archive in order to plot retrospectively or create a library file referencing the data and best-fit curves contributing to the five cross–sections.

Best-Fit Surface and Contours for: v = f(S,I)



Cross-Sections of the Surface v = f(S,I)



The second technique: Plotting the sign of residuals on contours

It is not always possible to create a good display of the data plotted with the corresponding best–fit surface because some data may lie below the surface and some above no matter how the surface is rotated.

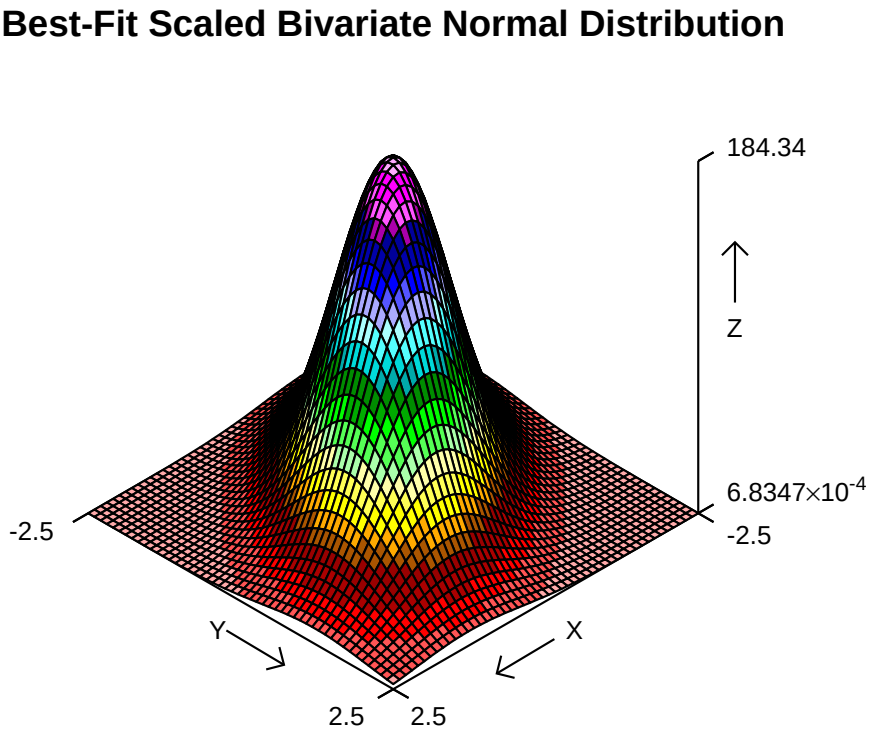
To illustrate this phenomenon and present an alternative way to view the residuals consider the second example using the following statistical model.

The bivariate normal pdf: $p_1 = \mu_x, p_2 = \sigma_x, p_3 = \mu_y, p_4 = \sigma_y, p_5 = \rho$

$$\frac{p_6}{2\pi\sigma_x\sigma_y\sqrt{1-\rho^2}} \exp\left\{\frac{-1}{2(1-\rho^2)}\left[\left(\frac{x-\mu_x}{\sigma_x}\right)^2 - 2\rho\left(\frac{x-\mu_x}{\sigma_x}\right)\left(\frac{y-\mu_y}{\sigma_y}\right) + \left(\frac{y-\mu_y}{\sigma_y}\right)^2\right]\right\} + p_7$$

This can be fitted as follows.

1. Open SIMFIT program **qnfit** then select to fit a function of two variables.
2. Read in the test file `bivariate.tf1`, which contains data generated using program **makdat** followed by adding random error using program **rannum**.
3. Fit in the EXPERT mode which reads the starting estimates from the data file.
4. Observe the goodness of fit results, view the residuals, and then plot the best–fit surface, as shown next.

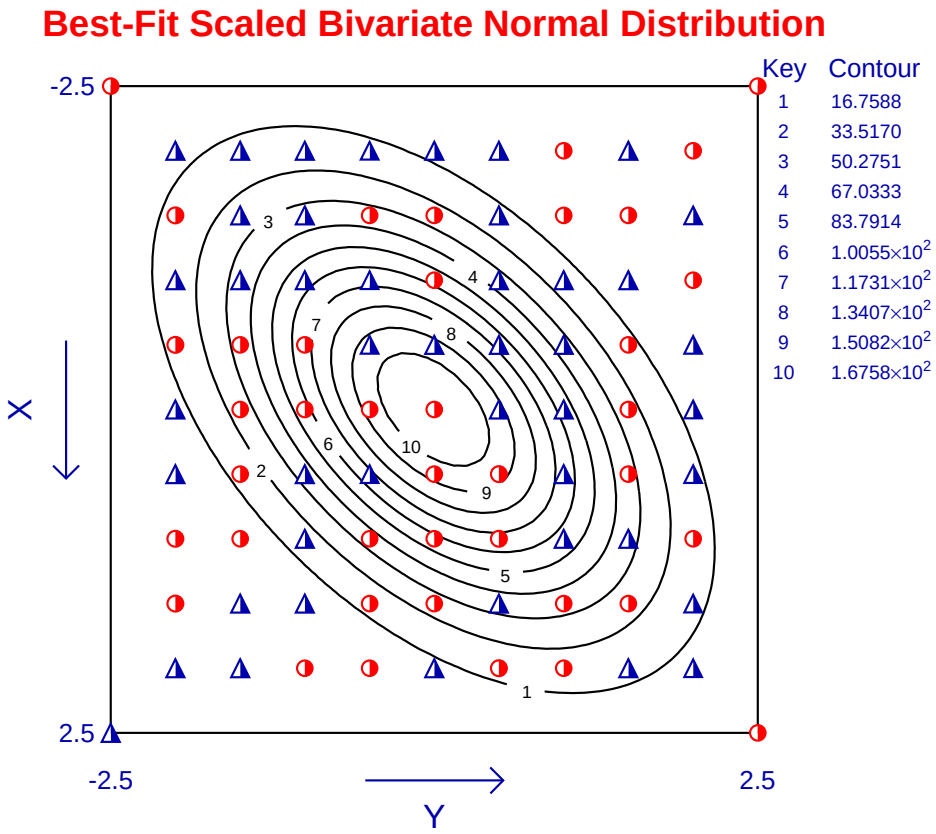


It is obvious from the table of residuals that there is a scatter of data points both above the best-fit surface (positive residuals) and below the best-fit surface (negative residuals) and a technique to display this scatter is presented next.

Note that each time a request is made to view the best-fit surface after fitting a function of two variables to a data set, program **qnfit** archives two temporary files containing the x, y coordinates of the data partitioned as follows.

- `qnfit_positive_residuals.txt` (contains coordinates for the positive residuals).
- `qnfit_negative_resdiduals.txt` (contains coordinates for the negative residuals).

These two temporary files are written to your `...\Documents\Simfit\res` folder, and it only remains to explain how these are to be overlayed as shown in the following contour diagram.



Each time the choice to display a best-fit surface is made there are several alternatives. In particular, it will be seen that on the main interface to the displays is an option to add overlays to the contour diagrams. This allows users to add arbitrary coordinate files to be plotted as lines or symbols in chosen colors as overlays on the contours.

For instance, in the above example the positive residuals are displayed as half-filled circles in red, while the negative residuals are shown as blue half-filled triangles. It is clear from the scatter of color over the evenly spaced grid of data points that there is no evidence for systematic lack of fit, and this can be corroborated from inspection of the best-fit parameters and tables of results together with using several other alternative methods for statistical analysis.

8.6.9 Simultaneous fitting of multiple equations in one variable

Often experimental data have been obtained for several responses as a function of the same independent variable and it is wished to fit a model consisting of several deterministic equations to the combined data set. For example, in a chemical experiment several components of a complex reaction can sometimes be measured simultaneously. There are two distinct situations.

- **The equations do not involve the same parameters**

  In this situation there is nothing to be gained from performing a joint determination of all parameters and it is more sensible to analyze the components separately.

- **The equations have common parameters**.

  Separate analysis of the components can be used to get some idea of possible starting estimates but, as the equations are linked, it will necessary to fit the comprehensive model to obtain meaningful parameter estimates.

This type of multiple curve fitting using SimFIT program **qnfit** requires the following steps.

1. **Choosing the correct option**

   From the [A/Z] option on the main SimFIT menu open program **qnfit** then choose to fit a model defining n functions of 1 variable.

2. **Providing data**

   The n data sets can be input in sequence by file-selection or from your curve fitting project archive, but by far the best method is to input a library file. This has the additional advantage that a percent sign can be used to imply a missing data set.

3. **Providing starting estimates**

   These can be input interactively but it is infinitely preferable to supply starting estimates using the `begin{limits} ... end{limits}` technique appended to the first data set because the program can then be run in EXPERT mode.

4. **Providing a model**

   The model must be prepared as an ASCII text model file using SimFIT program **usermod**.

To illustrate this procedure an extremely simple data set and model with three linear equations unlinked by common parameters will be used.

Example 8: case 1

From the [Demo] button on the file input control read in the library file `line3.tfl` which is as follows.

```
3 lines for line3.mod/qnfit
line1.data
line2.data
line3.data
```

The first line is the title of the library file and the next three lines are the names of the individual data files.

It is important to note that short names are used in this example because SimFIT recognizes that these three files are test files but, in your own examples, you must use the full path to your data files.

Now consider the first of these test data file, namely `line1.data` shown next.

```
data for line3.mod y = x + 1
5 3
1  1.1  1
2  2.0  1
3  2.9  1
4  4.2  1
5  4.8  1
begin{limits}
-10  0.5  10
-10  0.5  10
-10  1.0  10
-10  3.0  10
-10  2.0  10
-10  2.5  10
end{limits}
```

We see that, after the title, the file header dimension indicates that there 5 values in 3 columns, that is, $x = 1, 2, 3, 4, 5$ in column 1, $y = 1.1, 2.0, 2.9, 4.2, 4.8$ in column 2, and constant weights equal to 1 in column 3. These values are then followed by the limits and starting estimates.

The model

The model file is `line3.mod` which is now listed.

```
%
Three user supplied functions of 1 variable ... 3 straight lines
f(1) = p(1) + p(2)x: (line 1)
f(2) = p(3) + p(4)x: (line 2)
f(3) = p(5) + p(6)x: (line 3)
 %
 3 equations
 1 variable
 6 parameters
 %
 begin{expression}
 f(1) = p(1) + p(2)x
 f(2) = p(3) + p(4)x
 f(3) = p(5) + p(6)x
 end{expression}
 %
```

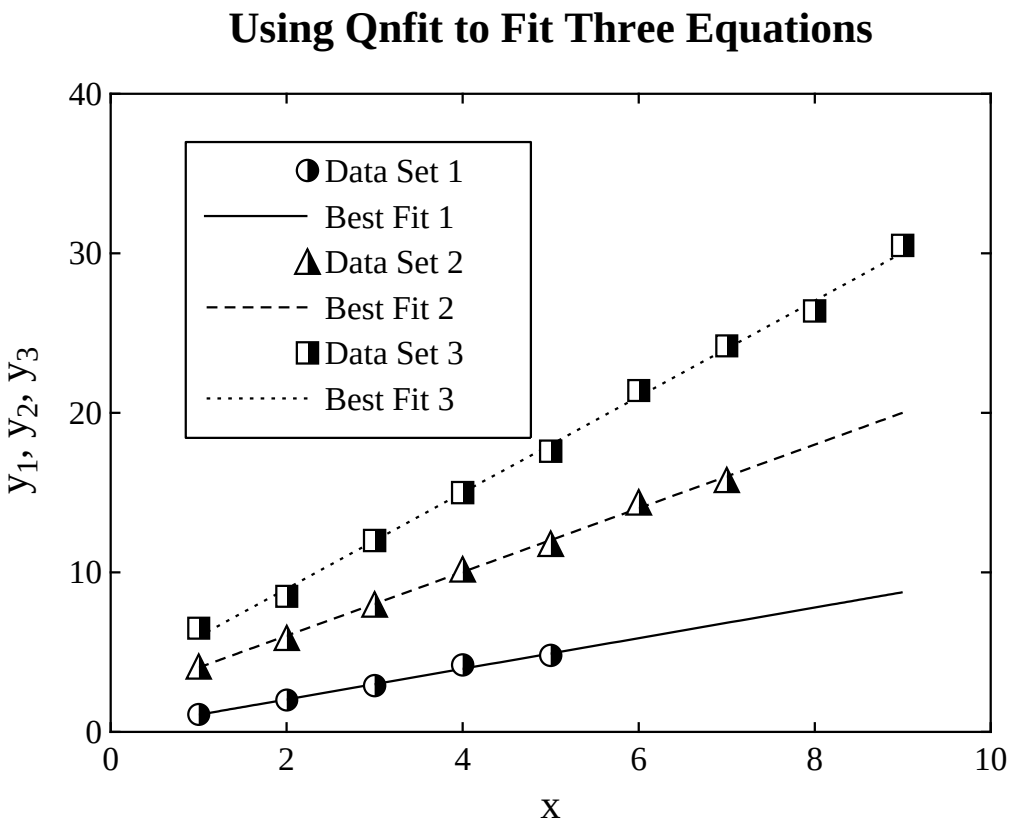
The parameter estimates

Upon proceeding to fit this model the following table of parameter estimates is obtained, indicating that, except for the intercept to line 1 all parameters were well–defined.

Number	Low-Limit	High-Limit	Value	Std.Error	Lower95%cl	Upper95%cl	p
1	-10.0	10.0	0.12001	0.367597	-0.66351	0.90352	0.7486 *
2	-10.0	10.0	0.96000	0.110835	0.72376	1.19623	0.0000
3	-10.0	10.0	2.04286	0.296218	1.41149	2.67423	0.0000
4	-10.0	10.0	1.99643	0.066236	1.85525	2.13761	0.0000
5	-10.0	10.0	2.96944	0.254625	2.42672	3.51216	0.0000
6	-10.0	10.0	3.00833	0.045248	2.91189	3.10478	0.0000

The best–fit curves

After fitting the best–fit curves can be displayed as below from the final menu by choosing the option to use the best fit model to plot/extrapolate/deconvolute.



Example 8: case 2

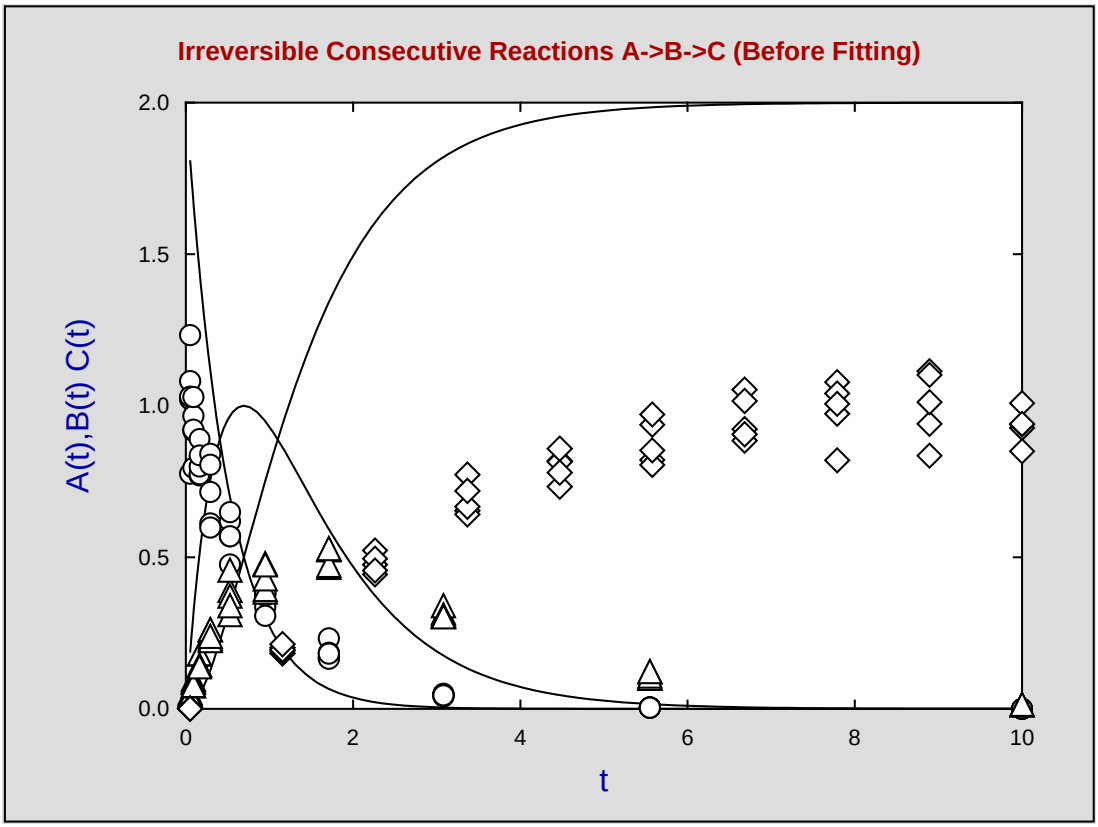
In cases linked by common parameters it is often necessary to fit a system of nonlinear differential equations using program **deqsol**, but the next example illustrates a case where an explicit solution can be obtained for the scheme

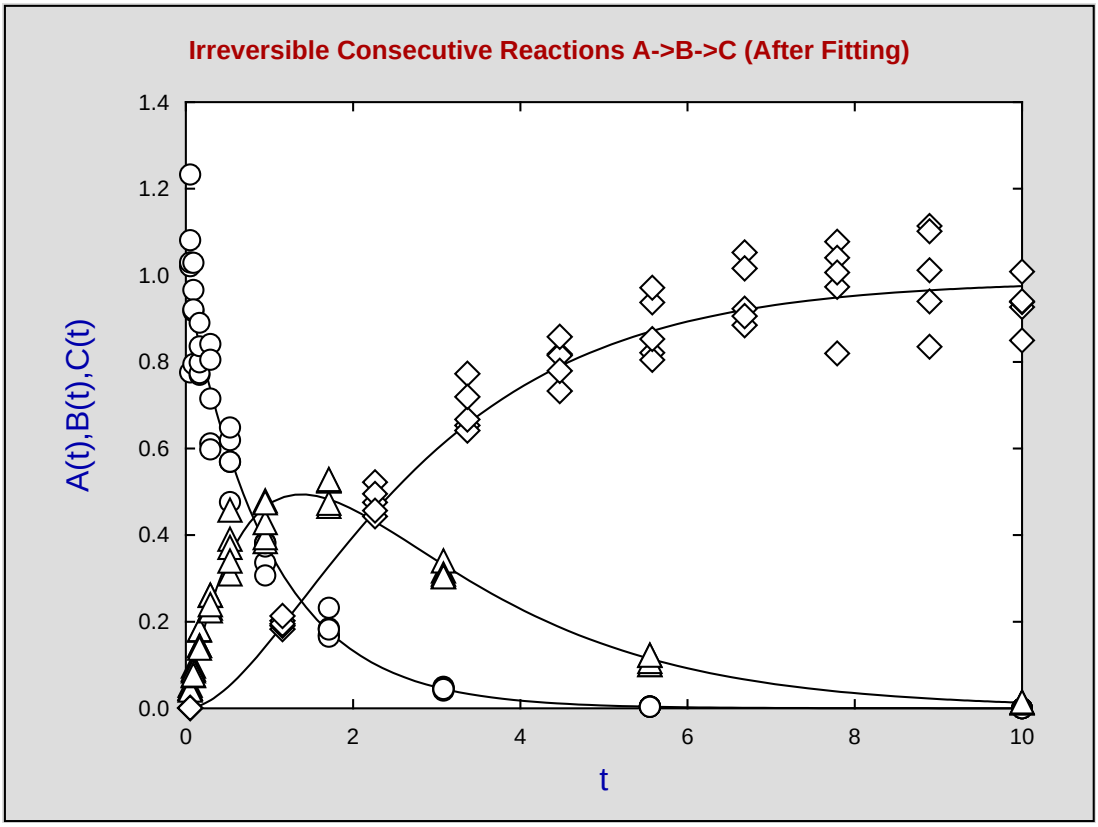
$$A \xrightarrow{k_1} B \xrightarrow{k_2} C$$

where $p_1 = k_1, p_2 = k_2, p_3 = A(0), p_4 = B(0), p(5) = C(0)$.

$$\begin{aligned}
A(t) &= p_3 \exp(-p_1 t) \\
B(t) &= \left\{\frac{p_1 p_3}{p_2 - p_1}\right\} \exp(-p_1 t) + \left\{p_4 - \frac{p_1 p_3}{p_2 - p_1}\right\} \exp(-p_2 t) \\
&= (p_1 p_3 t + p_4) \exp(-p_1 t), \text{ if } k_1 = k_2 \\
C(t) &= p_3 + p_4 + p_5 - \left\{\frac{p_2 p_3}{p_2 - p_1}\right\} \exp(-p_1 t) - \left\{p_4 - \frac{p_1 p_3}{p_2 - p_1}\right\} \exp(-p_2 t) \\
&= p_3 + p_4 + p_5 - A(t) - B(t), \text{ if } k_1 = k_2.
\end{aligned}$$

Using **qnfit** to analyze the data in library file `qnfit_data.tf8` with the model `qnfit_model.tf8`, which includes the alternative expression for the singular case $k_1 = k_2$, gives the results obtained before and after fitting shown next. Note that these two test files can be input using the [Demo] button from the **qnfit** file opening dialogue.


Irreversible Consecutive Reactions A->B->C (Before Fitting)
A(t),B(t) C(t)
2.0
1.5
1.0
0.5
0.0
0
2
4
6
8
10
t


Irreversible Consecutive Reactions A->B->C (After Fitting)
A(t),B(t),C(t)
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
2
4
6
8
10
t

8.6.10 Fitting a convolution integral

Fitting convolution integrals is often required when an output function is the time–dependent response resulting from an input function interacting with a target function. Parameter estimates and goodness of fit criteria can then be used to justify the assumed functions

The convolution integral required in such situations is defined as $f_3(x) = f_1(t) * f_2(t)$ resulting from the functions $f_1(t)$ and $f_2(t)$ according to

$$\begin{aligned} f_3(x) &= \int_0^x f_1(t) f_2(x-t)\, dt \\ &= f_1 * f_2 \\ &= f_2 * f_1 \end{aligned}$$

where it is understood that functions f_1 and f_2 vanish for negative arguments. This presents no difficulty if the functions are known and the integral can be evaluated formally, but a variety of special situations are encountered experimentally. The case that SimFIT program **qnfit** can analyze is where functions f_1 and f_2 are assumed deterministic functions but possibly with parameters that have to be estimated from observations on at least one of the functions f_1, f_2, or f_3, and where observations may have experimental error as well as arbitrary spacing and numbers of replicates. Because several situations can arise, for instance when $f_1(x)$ is known exactly, or where $f_2(x)$ cannot be determined independently, the method for supplying data must be sufficiently versatile to accommodate all possible cases and this will be explained first.

Example 9

In general there could be between 1 and 3 data sets and, for the example to be discussed, where the models are

$$\begin{aligned} f_1(t) &= \exp(-p_1 t) \\ f_2(t) &= p_2^2 t \exp(-p_2 t), \end{aligned}$$

the data will be accessed using the model test data set defined by the library file `qnfit_data.tf9`, and this file is as follows.

```
Convolution data
%
%
convolv3.data
```

Here the first line is the title of the library file and the next 3 lines identify the 3 individual data files for the three models f_1, f_2 and $f_3 = f_1 * f_2$. However, note that percentage signs at lines 2 and 3 indicate missing data so that only 1 data set is to be provided, i.e., for the output function. Users should note that when library files are created the files referenced must contain the fully qualified path and filename and not the short name. The reason a short name is used here is because SimFIT recognizes that `convolv3.data` is a known test file that is to be found in the installation `dem` folder, e.g., `c:\program files\simfit\dem`.

The data set referenced by test file `qnfit_data.tf9` is as as shown next.

```
  p(1) = 1, p(2) = 2, 7.5% relative error
    50     3
  1.0000E-01,  1.9472E-02,  2.0200E-03
  1.0000E-01,  1.6683E-02,  2.0200E-03
  1.0000E-01,  1.8557E-02,  2.0200E-03
  1.0000E-01,  1.5427E-02,  2.0200E-03
  1.0000E-01,  2.0353E-02,  2.0200E-03
  1.5440E-01,  4.1841E-02,  2.3773E-03
  1.5440E-01,  4.0062E-02,  2.3773E-03
  1.5440E-01,  3.8858E-02,  2.3773E-03
  1.5440E-01,  3.6756E-02,  2.3773E-03
  1.5440E-01,  3.6039E-02,  2.3773E-03
  2.3850E-01,  7.6279E-02,  6.5248E-03
  2.3850E-01,  6.7460E-02,  6.5248E-03
  2.3850E-01,  8.4294E-02,  6.5248E-03
  2.3850E-01,  8.0786E-02,  6.5248E-03
  2.3850E-01,  7.3433E-02,  6.5248E-03
  3.6840E-01,  1.5220E-01,  1.0492E-02
  3.6840E-01,  1.4586E-01,  1.0492E-02
  3.6840E-01,  1.2986E-01,  1.0492E-02
  3.6840E-01,  1.5788E-01,  1.0492E-02
  3.6840E-01,  1.4757E-01,  1.0492E-02
  5.6900E-01,  2.5783E-01,  5.3012E-03
  5.6900E-01,  2.5305E-01,  5.3012E-03
  5.6900E-01,  2.5728E-01,  5.3012E-03
  5.6900E-01,  2.4757E-01,  5.3012E-03
  5.6900E-01,  2.4645E-01,  5.3012E-03
  8.7880E-01,  3.5996E-01,  4.3440E-02
  8.7880E-01,  3.4269E-01,  4.3440E-02
  8.7880E-01,  3.0933E-01,  4.3440E-02
  8.7880E-01,  3.2765E-01,  4.3440E-02
  8.7880E-01,  4.2261E-01,  4.3440E-02
  1.3570E+00,  4.2238E-01,  2.4938E-02
  1.3570E+00,  4.1043E-01,  2.4938E-02
  1.3570E+00,  4.3381E-01,  2.4938E-02
  1.3570E+00,  3.7102E-01,  2.4938E-02
  1.3570E+00,  3.9175E-01,  2.4938E-02
  2.0960E+00,  2.8337E-01,  1.5549E-02
  2.0960E+00,  2.9260E-01,  1.5549E-02
  2.0960E+00,  2.9741E-01,  1.5549E-02
  2.0960E+00,  3.1985E-01,  1.5549E-02
  2.0960E+00,  3.1586E-01,  1.5549E-02
  3.2370E+00,  1.2023E-01,  1.2802E-02
  3.2370E+00,  1.1852E-01,  1.2802E-02
  3.2370E+00,  1.4342E-01,  1.2802E-02
  3.2370E+00,  1.1157E-01,  1.2802E-02
  3.2370E+00,  1.1337E-01,  1.2802E-02
  5.0000E+00,  2.3755E-02,  1.9893E-03
  5.0000E+00,  2.9036E-02,  1.9893E-03
  5.0000E+00,  2.7638E-02,  1.9893E-03
  5.0000E+00,  2.7016E-02,  1.9893E-03
  5.0000E+00,  2.7876E-02,  1.9893E-03
begin{limits}
0.001  0.5  5.0
0.001  1.0  5.0
end{limits}
```

Parameter starting estimates and limits have been appended to the data, not the library file, so that program **qnfit** can be used in the EXPERT mode, and the model file is `qnfit_model.tf9` shown below.

```
%
convolution integral: from 0 to x of f1(u)*f2(x - u) du, where
f1(t) = exp(-p(1)*t), f2(t) = [p(2)^2]*t*exp(-p(2)*t)
f3(x) = f1*f2
%
3 equations
1 variable
2 parameters
%
begin{expression}
A = p(1)
B = p(2)
C = p(2)*p(2)
end{expression}
1
x
user1(x,m)
f(1)
2
x
user1(x,m)
f(2)
0.0001
epsabs
0.001
epsrel
0
blim(1)
x
tlim(1)
convolute(1,2)
f(3)
%
begin{model(1)}
%
Example: exponential decay, exp(-p(1)*x)
%
1 equation
1 variable
0 parameter
%
begin{expression}
f(1) = exp(-A*x)
end{expression}
%
end{model(1)}
begin{model(2)}
%
Example: gamma density of order 2
%
1 equation
1 variable
0 parameters
%
begin{expression}
f(1) = C*x*exp(-B*x)
end{expression}
%
end{model(2)}
```

As usual, the model starts with a title section followed by a main section using some commands that require explanation, as all models for fitting convolution integrals must have these features.

1. **Values returned**

   For each value of the independent variable x the model returns the following results.

   - $f_1(x)$
   - $f_2(x)$
   - $f_3(x)$

   The value of $f_1(x)$ can be used to fit model 1 independently, similarly the value of $f_2(x)$ corresponds to the response measured independently of f_1. but not at the same time as $f(3)$ where the argument for f_2 in the integrand is $x - t$. The intention to be used demonstrates how $f_3(x)$ could be fitted at the same time as $f_1(x)$ if data were supplied for both $f_1(x)$ and $f_3(x)$ but $f_2(x)$ could only be fitted at the same time if the data corresponded to the response as a simple function. More usually $f_1(x)$ and $f_2(x)$ are only made available for plotting, as per text–book examples.

2. **Communicating parameters to sub–models**

```
begin{expression}
A = p(1)
B = p(2)
C = p(2)*p(2)
end{expression}
```

   This is a useful way to allow the parameters to be used by the sub–models without using the command `putpar` which is used for this purpose in the reverse Polish version `convolv3.mod`.

3. **Evaluating the sub–models**

```
1
x
user1(x,m)
f(1)
2
x
user1(x,m)
f(2)
```

   This code simply defines $f_1(x)$ and $f_2(x)$ at the current point x.

4. **Parameters controlling the integration**

```
0.0001
epsabs
0.001
epsrel
0
blim(1)
x
tlim(1)
convolute(1,2)
f(3)
```

This is how the relative and absolute tolerances are set then the convolution is performed.

5. **The sub–models**

   The sub–models are then defined as if they were standard models except that the number of parameters is set to zero in each model, since the parameters to be estimated have been declared globally using A, B, and C.

Example 9

The steps required to fit a convolution integral using data and models provided by SIMFIT are now listed with additional comments.

1. **Opening the curve fitting program data**

   From the main SIMFIT menu either choose [Fit] or [A/Z] and proceed to open program **qnfit**.

2. **Supplying data**

   Specify that you wish to fit n functions of 1 variable and then specify that n is to be 3. When asked to supply data press the [Demo] button on the file opening dialogue and select the library file `qnfit_data.tf9`.

3. **Supplying models**

   When asked for a model file use the [Demo] button on the file opening dialogue to choose the model file `qnfit_model.tf9`.

4. **Supplying starting estimates**

   The best way to supply parameter starting estimates and limits is to choose the EXPERT mode as that reads settings from the actual data set supplied which is `convolv3.data`, and not the library file `qnfit_data.tf9`.

5. **Performing a fit**

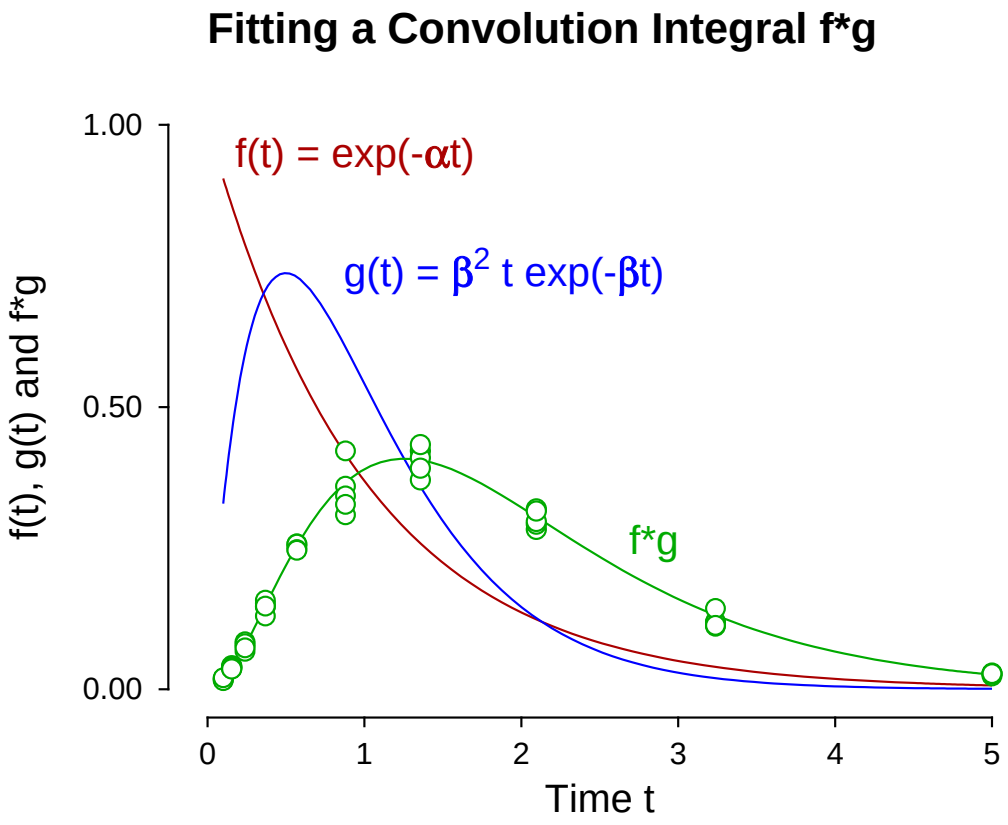
   At this stage you can preview the parameter starting estimates and limits, or view the fit of the model with starting estimates overlayed on the data before proceeding to fitting.

After fitting the following summary and a table of best-fit parameters will be displayed.

Results from curve-fit number 1	
Number of data points	50
Number of parameters	2 (0 currently fixed)
Degrees of freedom	48
WSSQ before entry	62318.7
IFAIL from LBFGSB	0
WSSQ from fitting	49.654
P($\chi^2 \geq$ WSSQ)	0.4072
Time taken to fit	0.194 (secs cpu time)

Best-fit parameters for curve-fit 1 using LBFGSB							
Number	Low-Limit	High-Limit	Value	Std.Error	Lower95%cl	Upper95%cl	p
1	0.001	5.0	0.99684	0.0089381	0.97887	1.01482	0.0000
2	0.001	5.0	2.00479	0.0115128	1.98164	2.02793	0.0000
For 50,90,95,99% parameter confidence limits using [parameter value +/- t(α/2)*std.err.] t(0.25) = 0.680, t(0.05) = 1.677, t(0.025) = 2.011, t(0.005) = 2.682							

Of course, now the model validity can be checked if the parameters are known independently by comparing the differences between the known and estimated parameters. Additional tests for goodness of fit can be displayed and eventually a plot showing the input function, response function (as a function of x and not shifted) and output convolution as illustrated next.



8.6.11 Fitting a single differential equation

Often observations are made that can be modeled by fitting a differential equation. This can be done using SimFiT program **deqsol** which is very useful for simulating and fitting, but program **qnfit** is recommended for fitting a single differential equations as there are many more options available for altering starting estimates interactively and recording goodness of fit criteria.

It is important to stress several issues that are important when fitting a differential equation.

1. **The Data**

   It is very important to only attempt fitting when data are very extensive and accurate and the model has been simulated to observe the behavior over the data range. Otherwise parameter estimates will be very unreliable.

2. **The initial condition**

   When advancing the solution of a differential equation from the chosen starting point it is clear that the whole trajectory is governed by this starting point. This raises several points.

   - If the initial condition is known with certainty then it is preferable to fit without the initial condition being estimated. However, an incorrectly specified fixed initial condition can result in a tail–wagging–the–dog situation leading to biased parameters estimates.
   - If the initial condition is estimated then the fitting can be dominated by the inial condition parameter having a different effect on convergence than the other parameters.
   - If the initial condition is to be estimated then this will probably be be known within a fairly narrow range, so the parameter limits on the initial condition could be fairly close. However, it should be pointed out that fixing parameter limits too close can prevent the convergence techniques from operating optimally.

3. **The starting estimates**

   Constrained nonlinear regression works best when the internal parameters are of order unity and, if possible, are parameterized in the formulation of the model so that sensitivity of the model to all the parameters to be estimated is similar. Where possible SimFiT scales parameters before optimization commences using the starting estimates. Evidently this cannot be done for very small starting estimates.

4. **The parameter limits**

   Models for differential equation must have n parameters involved in the model plus a further parameter that is the value of the initial condition. That is, for a differential equation of the form

   $$dy/dx = f(x, y, \Theta), y(0) = y_0$$

   involving parameters $\Theta = \theta_1, \theta_2, \ldots, \theta_{n+1}$, then parameter $\theta_{n+1} = y_0$. The parameter starting estimates and limits should be such that parameters to be varied should have a small range of variation (but not too small) and parameters to be fixed can be indicated by setting the limits equal to the starting estimates. In this way the inititial condition can be fixed.

5. **The simulation**

   A differential equation can easily be simulated using program **makdat**, or for more comprehensive options **deqsol** where, to initialize the simulation, the range of integration can be appended to the end of the model file.

6. **The methods**

   To allow for stiff equations a Jacobian can be added to the model file, but should also compared with simulation without an explicit Jacobian to ensure that the Jacobian has been coded correctly.

Case 1: Irreversible substrate depletion

This fits the irreversible Michaelis-Menten substrate depletion scheme

$$\frac{dS}{dt} = -\frac{V_{max}S}{K_m + S} = f$$
$$\frac{df}{dS} = -\frac{V_{max}K_m}{(K_m + S)^2} = J$$

which is also provided in the SimFIT model library both as a formal integrated equation as well as a built in differential equation model.

The data file `deqn_data.tf1` ends with the following limits

```
begin{limits}
0.5 0.75 1.5
0.5 1.25 2.5
0.5 0.8  1.5
end{limits}
```

so **qnfit** can be used in the EXPERT mode. The corresponding model file `deqn_model.tf1` is

```
%
  model: irreversible Michaelis-Menten substrate depletion curve
 differential equation: f(1) = dy(1)/dx
                             = -p(2)*y(1)/[p(1) + y(1)]
              jacobian: j(1) = df(1)/dy(1)
                             = -p(1)p(2)/[p(1) + y(1)]^2
    initial condition: y0(1) = p(3)
 Note: the last parameter must be y0(1) in a differential equation
       y(1) = S, p(1) = Km, p(2) = Vmax, y0(1) = S(0)
 %
 1 equation
 differential equation
 3 parameters
 %
 begin{expression}
 D = p(1) + y(1)
 f(1) = -p(2)y(1)/D
 end{expression}
 %
 begin{expression}
 j(1) = -p(1)p(2)/D^2
 end{expression}
  %
 begin{limits} ... low-limits, starting estimates, upper-limits
 0 1 3
 0 1 3
 0 1 3
 end{limits}
 begin{range} ... number of points, x_start and x_end for integration
 121
 0
 10
 end{range}
```

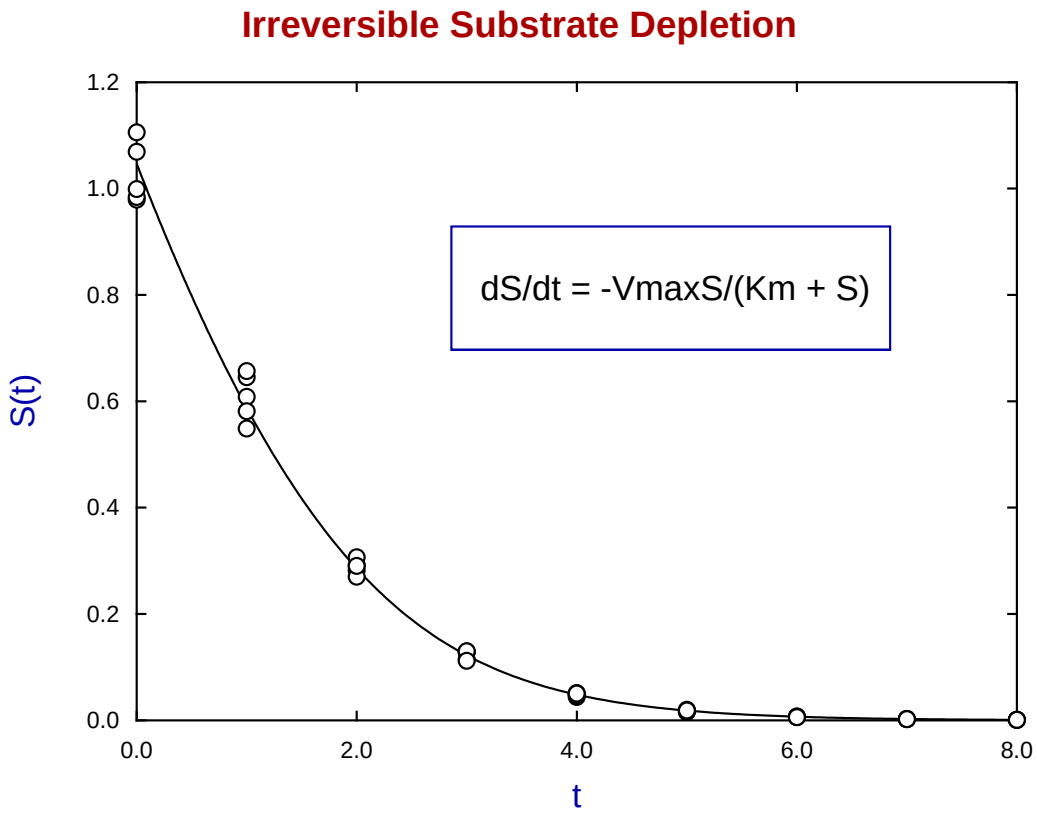
To appreciate how to supply your own data eventually it is suggested you read in the data file `deqn_data.tf1`, then the model file `qnfit_model.tf1`, and study the structure of the model file to make sure you understand the format.

SIMFIT program **usermod** can be used to create your own model if the model you require is not provided by the pre-compiled library of models for simulating and fitting.

A number of details need mentioning.

1. The `begin{limits} ... end{limits}` section appended to the data file is only provided to allow program **qnfit** to be used in the EXPERT mode. Program **qnfit** is opened, fitting a differential equation is selected, and `deqn_data.tf1` is read in using the [Demo] button on the file–open dialogue. Typically these limits would be edited interactively if a good fit cannot be obtained. After reading in the data, the option to open an ASCII text model file would be selected and `deqn_model.tf1` should be read in using the [Demo] button on the file–open dialogue.
2. The `begin{limits} ... end{limits}` section appended to the model file is only needed if fitting or simulation is to be undertaken using program **deqsol**.
3. The `begin{range} ... end{range}` section appended to the model file is only needed to fix the number of points and range of simulation if simulation is to be undertaken using program **deqsol**.

Here is the best–fit curve obtained using program **qnfit**.



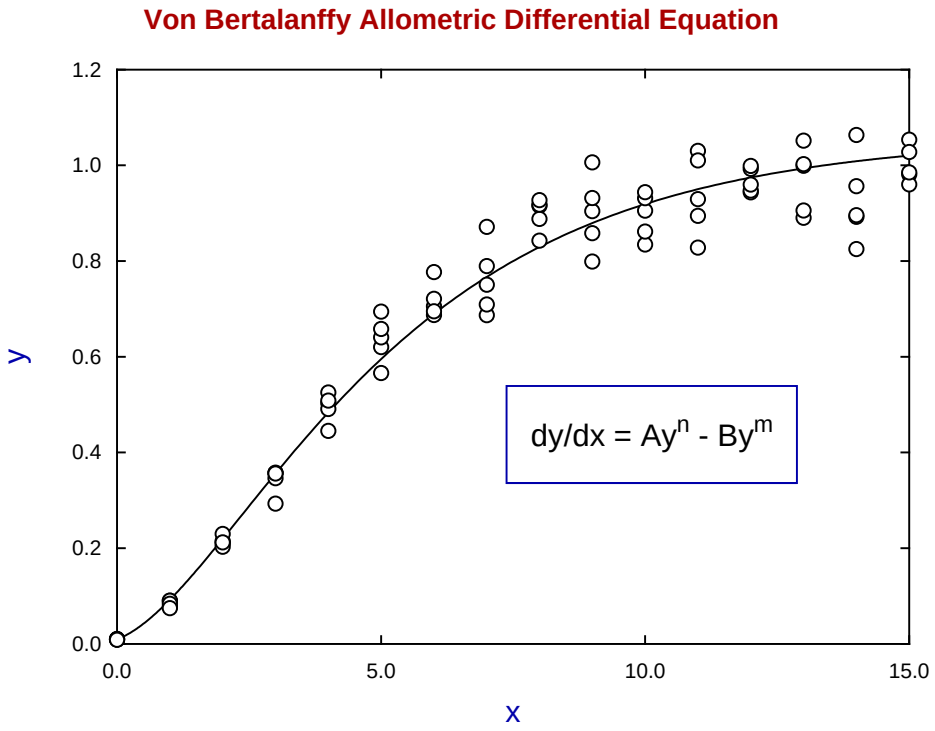
Case 2: The Von Bertalanffy Allometric Differential Equation

The appropriate default data file is `deqn_data.tf2` and the model file is `deqn_model.tf2` for

$$dy/dx = Ay^n - By^m$$

which can be integrated for some n and m to give the family of growth equations fitted by program **gcfit**.

```
%
 f(1) = p(1)*y(1)^p(2) - p(3)*y(1)^p(4), y(0) = p(5)
 j(1) = p(1)*p(2)*y(1)^(p(2) - 1.0) - p(3)*p(4) y(1)^(p(4) - 1.0)
 %
 1 equation
 differential equation
 5 parameters
 %
 begin{expression}
 f(1) = p(1)y(1)^p(2) - p(3)y(1)^p(4)
 end{expression}
 %
 begin{expression}
 A = p(2) - 1.0
 B = p(4) - 1.0
 j(1) = p(1)p(2)y(1)^A - p(3)p(4)y(1)^B
 end{expression}
 %
 begin{limits}
 0  1.0       3
 0  0.6666667 3
 0  1.0       3
 0  1.0       3
 0  0.01      1
 end{limits}
```



Case 3: Modified von Bertalanffy equation

The model dealt with in Case 1 is very easy to fit, but the model described in Case 2 is much more difficult to fit as it involves exponents. This means that the model fitting must be constrained to avoid raising negative numbers to non-integer powers, and to restrict the range within which exponents can vary.

However, the example to be described in Case 3 is chosen because it is an extreme example of a simple looking model that proves very difficult to fit, requiring both high quality data and careful choice of parameters and limits.

The motivation for this model should be explained first. The von Bertalanffy differential equation had a fanciful origin in proposing to balance the opposing effects of anabolism and catabolism by an appeal to the contrast between effects due to surface area and those due to volume. As it is an autonomous differential equation the derivative is defined unambiguously by the value of its argument and hence the integral can have no turning points. Now, as many growth situations have a phase of decline following a rise to maximum size, then clearly such a monotonic growth equation will not give a satisfactory fit, and parameter estimates will not be meaningful. To remedy this situation we can introduce a process of time–dependent deterioration affecting the surface area term more significantly than the volume term as follows, where C is a further parameter to be estimated.

$$dy/dx = \exp(-Cx)Ay^n - By^m$$
$$J = \exp(-Cx)Any^{n-1} - Bmy^{m-1}$$

The data set `deqn_data.tf3` was generated using program **makdat**, then pseudo–random error was added in triplicate using program **adderr** with constant relative error. The following model file (`deqn_data.tf3`) was composed using program **usermod**, noting than the last parameter is for the initial condition.

```
%
  model: modified von Bertalanffy growth model
 differential equation: f(1) = dy(1)/dx
                             = exp(-p(5)x)*p(1)*y(1)^p(2) - p(3)*y(1)^p(4)
              jacobian: j(1) = df(1)/dy(1)
                             = exp(-p(5)x)*p(1)*p(2)*y(1)^(p(2) - 1.0)
                               -p(3)*p(4)*y(1)^(p(4) - 1.0)
    initial condition: y0(1) = p(5)
 %
 1 equation
 differential equation
 6 parameters
 %
 begin{expression}
 C = p(1)*exp(-p(5)x)
 f(1) = C*y(1)^p(2) - p(3)y(1)^p(4)
 end{expression}
 %
 begin{expression}
 A = p(2) - 1.0
 B = p(4) - 1.0
 j(1) = C*p(2)y(1)^A - p(3)p(4)y(1)^B
 end{expression}
%
```

Note that this model has no protection against underflow or overflow, but the SimFIT library versions of this and the previous model are protected against such numerical instability.

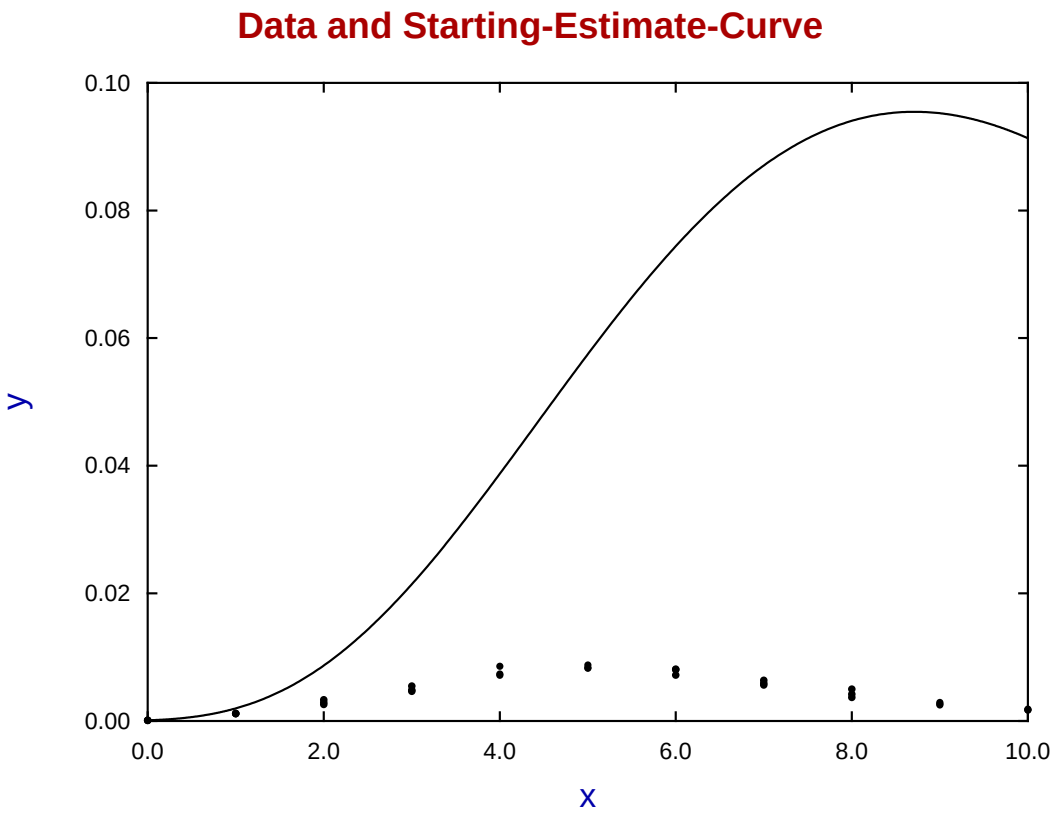
The parameter starting estimates and limits appended to the data set in `deqn_data.tf3` are shown next

```
begin{limits}
0.5     1.1    2.0
0.75    0.8    0.85
0.5     0.9    2.0
0.85    0.9    0.95
0.001   0.05    0.2
1.0e-5  1.0e-4 1.0e-3
end{limits}
```

which should be compared with the parameters used to generate the data.

```
 p(1) = 1.00000E+00
 p(2) = 8.00000E-01
 p(3) = 1.00000E+00
 p(4) = 9.00000E-01
 p(5) = 1.00000E-01
 p(6) = 1.00000E-04
```

After reading the data set `deqn_data.tf3` into program **qnfit** followed by the assumed model contained in the file `deqn_model.tf3`, the EXPERT mode was selected and, before commencing to fit, the starting-estimate best–fit curve was overlayed on the data as shown next.



This illustrates how, with differential equations containing parameters as exponents, a small variation in parameters can lead to large and somewhat unexpected changes in the integral, and at first sight it would seem improbable that this model could be made to fit the data.

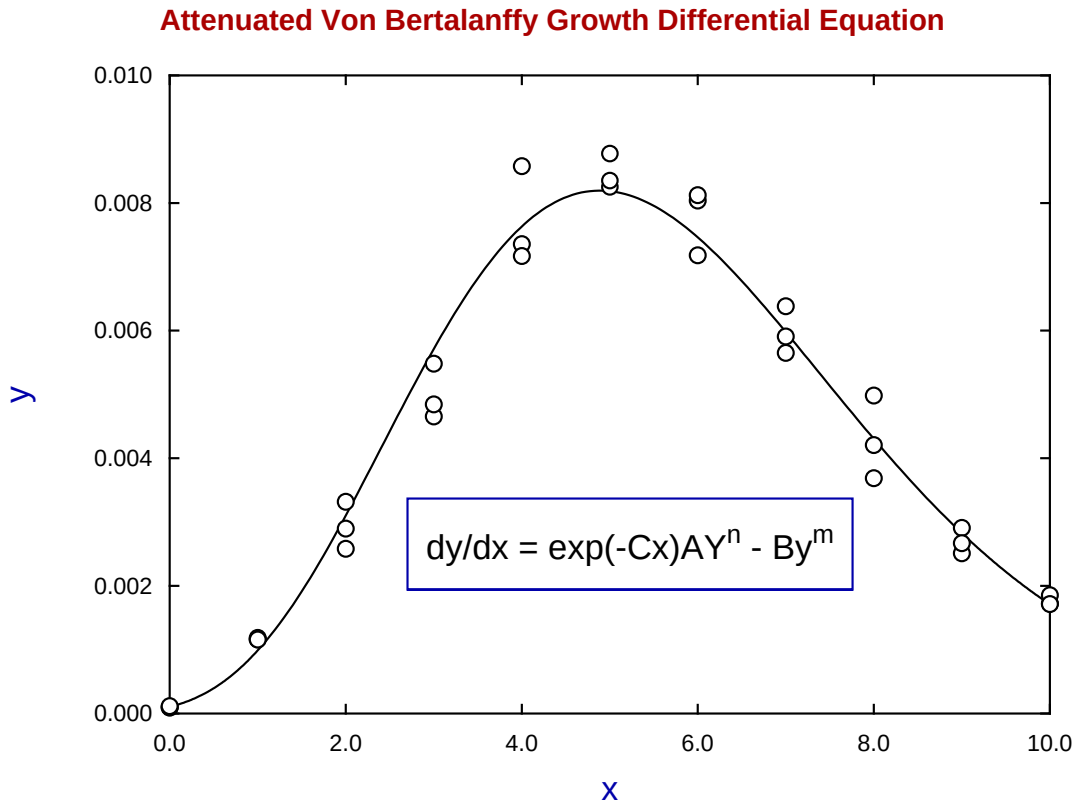
However, **qnfit** found a good solution with well–defined parameters and excellent goodness of fit criteria as shown next.

Results from curve-fit number 1
Number of data points = 33
Number of parameters = 6 (0 currently fixed)
Degrees of freedom = 27
WSSQ before entry = 1482310.0
IFAIL from LBFGSB = 0
WSSQ from fitting = 31.383
P($\chi^2 \geq$ WSSQ) = 0.2557
Time taken to fit = 2.5994 (secs cpu time)

Best-fit parameters for curve-fit 1 using LBFGSB/DVODE

Number	Low-Limit	High-Limit	Value	Std.Error	Lower95%cl	Upper95%cl	p
1	0.5	2.0	1.68365	0.00188	1.67979	1.68750	0.0000
2	0.75	0.85	0.84996	0.00625	0.83713	0.86279	0.0000 *
3	0.5	2.0	1.78207	0.03956	1.70091	1.86324	0.0000
4	0.85	0.95	0.92920	0.00661	0.91563	0.94278	0.0000
5	0.001	0.2	0.06624	0.00376	0.05853	0.07395	0.0000
6	1.000E-05	0.001	0.00010	4.04070E-06	9.64899E-05	0.00011	0.0000

For 50,90,95,99% confidence limits using [parameter value +/- t(α/2)*std.err.]
t(0.25) = 0.684, t(0.05) = 1.703, t(0.025) = 2.052, t(0.005) = 2.771



9 Data smoothing, calibration, and time series

Tutorials and worked examples for simulation, curve fitting, statistical analysis, and plotting.
https://simfit.uk
https://simfit.org.uk
https://simfit.silverfrost.com

9.1 Introduction

Data smoothing is used to fit arbitrary curves to observations when there is no sense in fitting deterministic models to estimate meaningful parameters, because all that is required is a smoothed representation of the data to identify trends, or perform calibration for instance. SimFIT provides the following procedures.

1. **Smoothing time series.**

   Given a succession of points measured at a constant interval of space, time, etc. then various filters, such as Hanning or T5432H, can be applied to generate a smoothed representation. This technique, available from program **simstat**, is widely used in time series analysis.

2. **Autocorrelation.**

   Program **simstat** can analyze an arbitrary succession of values for lags and autocorrelations.

3. **Moving average analysis.**

   Program **simstat** can also fit ARIMA models in order to predict future performance.

4. **Fitting a polynomial.**

   This can be done for a succession of polynomials of degree one to six with statistics given to find the highest degree that can be justified. The resulting best-fit polynomial fitted by program **polnom** can be used to estimate deviations from linearity, or to act as as a calibration curve.

5. **Fitting piecewise cubic splines.**

   Several types of splines are available.

   - Program **compare** can be used to fit splines with automatically calculated knots to two profiles in order to compare the similarity and differences between two sets of measurements over a similar range of independent variable. For instance, for nonparametric comparison of two growth profiles.
   - Program **calcurve** fits splines with knots fixed by users in order to generate a calibration curve that can be used for inverse prediction. That is, given measurements of a variable y as a function of some variable x, to predict x given y.
   - Program **spline** can fit splines with knots fixed by users, calculated automatically, or chosen by cross-validation in order to visualize trends.

Items 1, 2, and 3 in the above list require data in the form of a vector V, that is, a succession of n data points

$$V = (x_1, x_2, x_3, \ldots, x_n).$$

However items 4 and 5 require a n by 3 matrix M with independent variable x, observations y and standard error estimates se

$$M = \begin{pmatrix} x_1 & y_1 & se_1 \\ x_2 & y_2 & se_2 \\ \ldots & \ldots & \ldots \\ x_n & y_n & se_n \end{pmatrix}$$

where the third column would be standard deviations determined from replicates for weighting $w = 1/se^2$, or more usually set to 1, or even omitted altogether for unweighted fitting.

9.2 Spline smoothing

Tutorials and worked examples for simulation, curve fitting, statistical analysis, and plotting.
https://simfit.uk
https://simfit.org.uk
https://simfit.silverfrost.com

9.2.1 Fitting cubic splines

Given a set of observations there are several ways to fit a piecewise cubic spline curve in order to generate a best-fit smoothed approximation than can then be used to visualize trends in the data, to perform calculations such as estimating derivatives, or to use as a calibration curve. It is important to appreciate that the best-fit spline is very dependent on the technique used so this must be explained.

Definition of a piecewise spline curve

A spline curve consist of contiguous sections separated at junction points called called knots, where a distinct cubic polynomial is defined for each section. If the data points x, y are in nondecreasing order of x with only one observation y_i at each value x_i then such a piecewise cubic spline curve has $(k+4)/2$ knots in all, with four of these at the first x value, four at the last x value, and $(k-12)/2$ at interior x values. For each interval defined by the extreme x values and the interior knots there will be a cubic polynomial $p(x)$ given by

$$p(x) = p_0 + p_1 x + p_2 x^2 + p_3 x^3$$

so that the model equation is actually a set of cubic polynomials with one for each interval. This model is fitted to minimize some objective function using the model evaluated for x_i as the value of the corresponding cubic polynomial defined for the interval containing x_i. Cubic polynomials that are adjacent must have the same function value and derivative where they meet at a knot, but there are numerous ways to define or calculate knot positions, and to define the objective function to be minimized. Note that, as spline fitting procedures require only one observation y_i at each distinct x_i value, replicate observations in data sets supplied to SimFIT are transformed internally to data sets with means replacing replicates for fitting, and weights calculated from replicates if no weights were supplied.

Program **spline** which can be opened using the [A/Z] option from the SimFIT main menu offers two options.

- **Input a new data file for fitting**

  This can then be fitted to generate a set of spline knots and calculate spline coefficients which then become the defaults for all subsequent procedures.

- **Input a previously saved spline knots file**

  This must contain a set of knots and coefficients to define a default best-fit spline curves that can be used directly for all subsequent procedures.

In addition there are the following main fitting techniques available.

1. Knots defined by users
2. Knots calculated automatically given a smoothing factor F
3. Knots between data points defined by a smoothing parameter ρ
4. Knots between data points with ρ defined by cross-validation

There are no hard and fast rules but the SimFIT program **calcurve** uses method 1 which is the easiest to understand and gives users complete control over defining a calibration curve, program **compare** uses method 2 to define splines that can be used to compare profiles for similarity and differences, program **csafit** uses

method 1 to approximate flow cytometry profiles, while program **spline** is for those who wish to investigate several methods for estimating derivatives, arc lengths, curvature, displaying trends in data, or for calibration analysis. The SIMFIT test file that will be used to demonstrate spline fitting procedures is `compare.tf1` containing the following data.

x	*y*	*se*
0.25000	0.017267	1
0.25000	0.015585	1
0.25000	0.014268	1
0.25000	0.014136	1
0.48647	0.12861	1
0.48647	0.12536	1
0.48647	0.13339	1
0.48647	0.14230	1
0.67860	0.26261	1
0.67860	0.34277	1
0.67860	0.30364	1
0.67860	0.31373	1
0.94662	0.67252	1
0.94662	0.70382	1
0.94662	0.59192	1
0.94662	0.54850	1
1.3205	0.90417	1
1.3205	0.74158	1
1.3205	0.77353	1
1.3205	0.74208	1
1.8420	0.79030	1
1.8420	0.79384	1
1.8420	0.67971	1
1.8420	0.76176	1
2.5695	0.58575	1
2.5695	0.66178	1
2.5695	0.70023	1
2.5695	0.72772	1
3.5844	0.53286	1
3.5844	0.62744	1
3.5844	0.55484	1
3.5844	0.52923	1
4.2334	0.55003	1
4.2334	0.46641	1
4.2334	0.46840	1
4.2334	0.53647	1
5.0000	0.49920	1
5.0000	0.51847	1
5.0000	0.44355	1
5.0000	0.40895	1

The columns contain data in the following format.

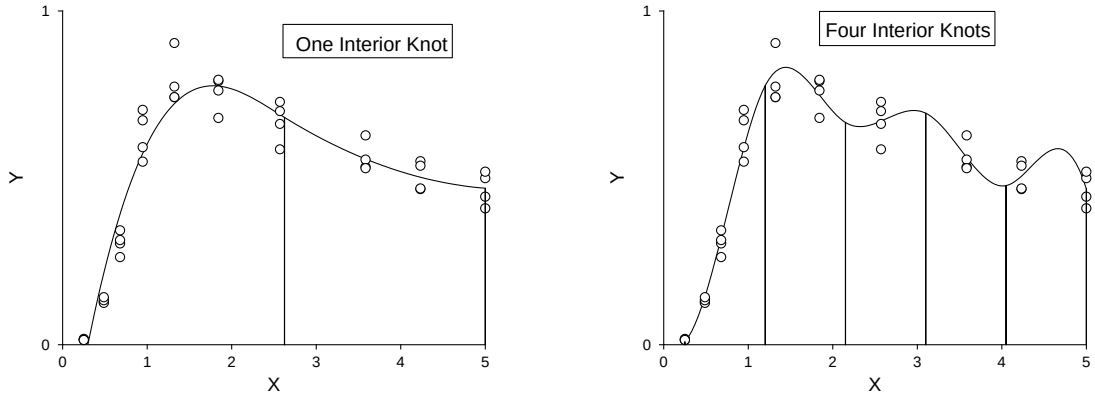
1. **Column 1**: the variable x which must be in non-decreasing order.

2. **Column 2**: the response y presumed to be dependent on x.

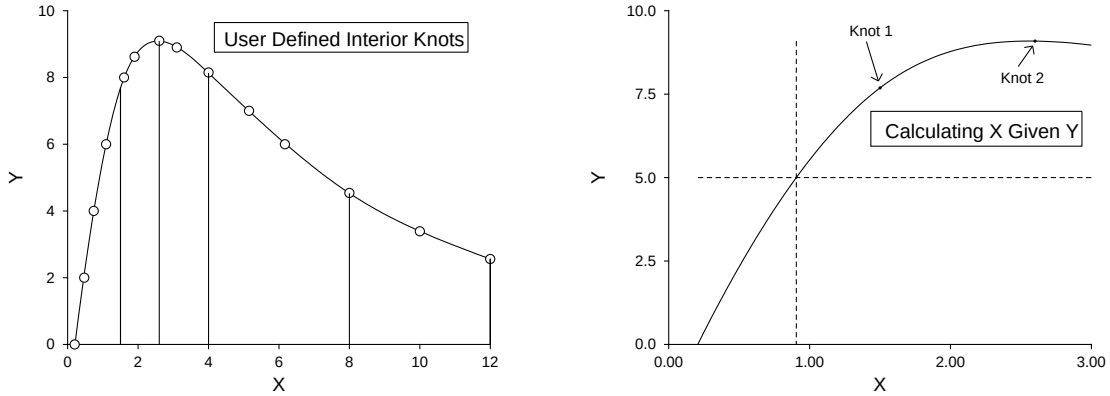
3. **Column 3**: the value of 1 indicates that the replicates will be used to calculate the sample standard deviations at each x-replicate value to be used for weighting.
   This column can be omitted or set to a positive value se if it is wished to supply weighting factors w directly which would then be used as $w = 1/se^2$.

Knots defined by user

Here the user must specify the number of interior knots and their spacing in such a way that genuine dips, spikes or asymptotes in the data can be modeled by clustering knots appropriately. Four knots are added automatically to correspond to the smallest x value, and four more are also added to equal the largest x value. If the data are monotonic and have no such spike features, then equal spacing can be resorted to, so users only need to specify the actual number of interior knots. The programs **calcurve** and **csafit** offer users both of these techniques, as knot values can be provided after the termination of the data values in the data file, while program **spline** provides the best interface for interactive spline fitting. Fixed knot splines have the advantage that the effect of the number of knots on the best fit curve is fully intuitive; too few knots lead to under-fit, while too many knots cause over-fit. The next figure illustrates the effect of changing the number



of equally spaced knots when fitting the data in `compare.tf1` by this technique. The vertical bars at the knot positions were generated by replacing the default symbols (dots) by narrow (size 0.05) solid bar-chart type bars. It is clear that the the fit with one interior knot is quite sufficient to account for the shape of the data, while using four gives a better fit at the expense of excessive undulation. To overcome this limitation of fixed knots SIMFIT provides the facility to provide knots that can be placed in specified patterns and, to illustrate this, the next figures display several aspects of the fit to `e02baf.tf1`.



The left hand figure shows the result when spline knots were input from the spline file `e02baf.tf2`, while the right hand figure shows how program **spline** can be used to predict X given values of Y. Users simply specify a range of X within the range set by the data, and a value of Y, whereupon the intersection of the dashed horizontal line at the the specified value of Y is calculated numerically, and projected down to the X value predicted by the vertical dashed line. Note that, after fitting `e02baf.tf1` using knots defined in `e02baf.tf2`, the best fit spline curve was saved to the file `spline.tf1` which can then always be input again into program **spline** to use as a deterministic equation between the limits set by the data in `e02baf.tf1`.

Automatically calculated knots

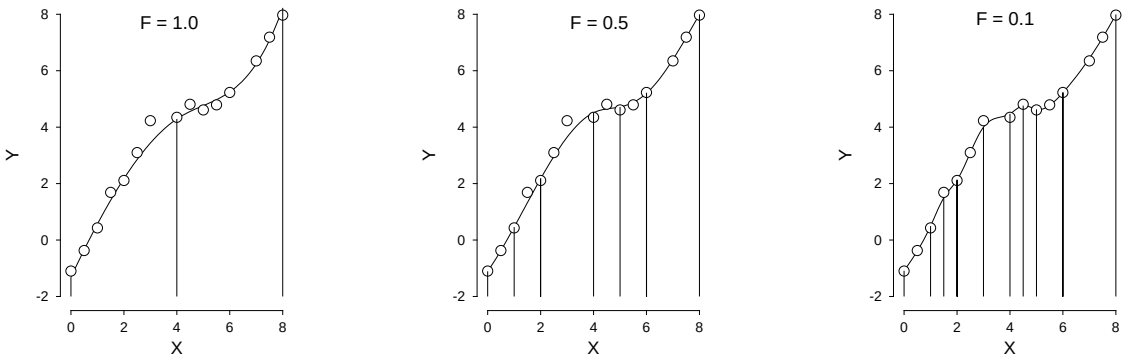
Here the knots are generated automatically and the spline is calculated to minimize

$$\eta = \sum_{i=5}^{m-5} \delta_i^2,$$

where δ_i is the discontinuity jump in the third derivative of the spline at the interior knot i, subject to the constraint

$$0 \leq WSSQ_N \leq F$$

where F is user-specified. If F is too large there will be under-fit and best fit curve will be unsatisfactory, but if F is too small there will be over-fit. For example, setting $F = 0$ will lead to an interpolating spline passing through every point, while choosing a large F value will produce a best-fit cubic polynomial with $\eta = 0$ and no internal knots. In weighted least squares fitting $WSSQ$ will often be approximately a chi-square variable with degrees of freedom equal to the number of experimental points minus the number of parameters fitted, so choosing a value for $F \approx N$ will often be a good place to start. The programs **compare** and **spline** provide extensive options for fitting splines of this type. The next figure, for example, illustrates the effect of fitting `e02bef.tf1` using smoothing factors of 1.0, 0.5, and 0.1.



In between knots: ρ input

Here there is one knot between each distinct x value and the spline $f(x)$ is calculated as that which minimizes

$$WSSQ_N + \rho \int_{-\infty}^{\infty} (f''(x))^2 \, dx.$$

As with the automatically generated knots, a large value of the smoothing parameter ρ gives under-fit while $\rho = 0$ generates an interpolating spline, so assigning ρ controls the overall fit and smoothness. As splines are linear in parameters then a matrix H can be found such that

$$\hat{y} = H\bar{y}$$

and the degrees of freedom ν can be defined in terms of the leverages h_{ii} in the usual way as

$$\begin{aligned} \nu &= \text{Trace}(I - H) \\ &= \sum_{i=1}^{N} (1 - h_{ii}). \end{aligned}$$

This leads to two ways to specify the spline coefficients which depend on ρ being fixed by the user or estimated in some way.

The spline can be fixed by specifying the value of ρ. To use this option, the value of ρ is input interactively, and the resulting fit inspected graphically until it is acceptable. This way users have complete control over the amount of smoothing required.

In between knots: ρ by generalized cross validation

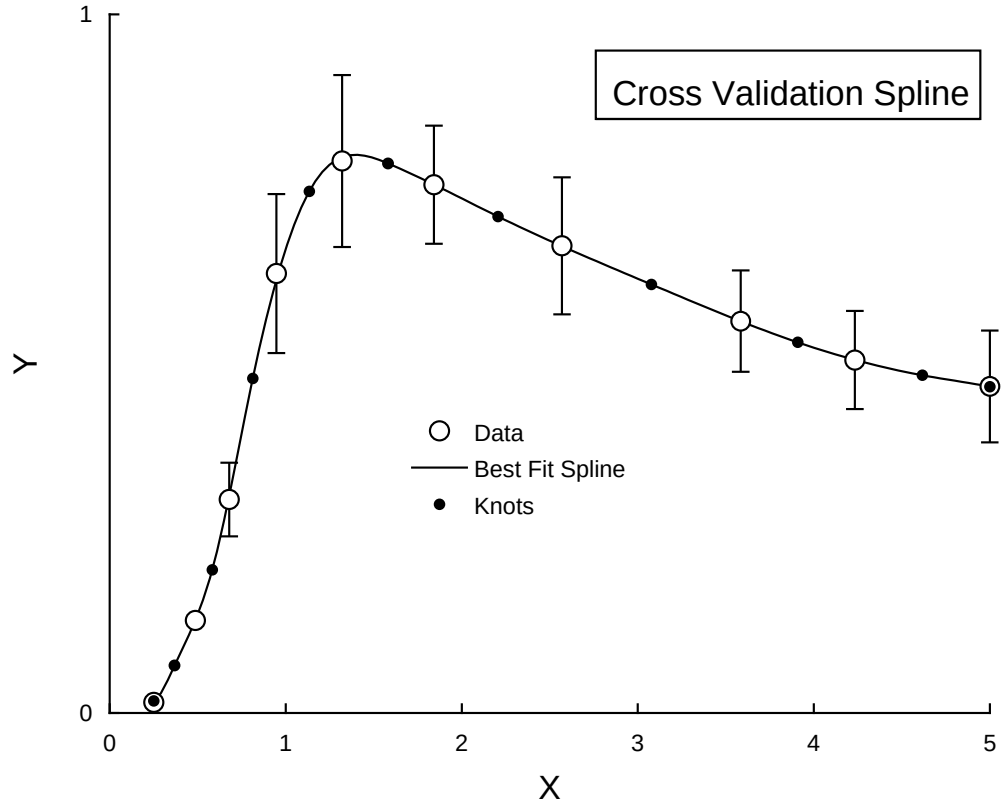
Alternatively, ρ can be estimated by minimizing the generalized cross validation GCV, where

$$GCV = N\left(\frac{\sum_{i=1}^{N} r_i^2}{(\sum_{i=1}^{N}(1-h_{ii}))^2}\right).$$

As this leads to a unique estimate for ρ, users have no control if the spline leads to either over-smoothing or over-fitting.

Here are the results for fitting `compare.tf1` which has 10 distinct data points using cross validation, and plotted with 95% confidence range error bars calculated from replicates, and with large dots to indicate the knot positions.

ρ	1.907E-03
DOF	3.489E-01
$WSSQ$	1.249E-04



9.2.2 Using cubic splines for calculations

Cubic splines that have been fitted to a data set can be used to calculate parameters that characterize a data profile, or to act as a standard curve for calibration, e.g. LD50 determination. Once a spline has been fitted to a data set it generates a set of knots and coefficients that can be save to a file in order to re-use the spline retrospectively.

From the main SimFIT menu choose [A/Z] and open program **spline** which has a default data set in test file `compare.tf1`, and a corresponding default spline file `spline.tf2`. Choosing to calculate using the current spline file offers the following choices.

1. X-Range: select $A = X_{min}, B = X_{max}$
2. Evaluate: spline and derivatives
3. Evaluate: Area-Length-Curvature
4. Calibrate: predict X given Y
5. Plot: spline
6. Plot: derivative
7. View: knots and coefficients

Choosing to view the knots and coefficients displays the following values.

Spline knots
0.25000
0.25000
0.25000
0.25000
0.67860
0.94662
1.32050
1.84200
5.00000
5.00000
5.00000
5.00000
Spline coefficients
0.01531456
0.05658395
0.20421635
0.71563087
0.82384236
0.63789726
0.50485453
0.46681088

There are always four knots at the lowest X value, and four at the highest X value, then between these are the locations of the interior knots. The coefficients are multipliers for the B splines. It should be noted that, after every fit of a piecewise spline to a data set has concluded, the splines from this fit become the current spline and can be saved to a file formatted like `spline.tf2`.

Evaluate a spline and derivatives

Supplying $x = 1$ for this calculation leads to these results.

X-value	Spline	First derivative	Second derivative	Third derivative
1	0.6678	0.7695	-3.216	7.129

Evaluate Area-Length-Curvature

Choosing this option leads to the following results for the range $A = X_{min}, B = X_{max}$.

A	B	Area	L = Arc-length	Integral$\lvert\kappa\rvert\, ds$	In degrees
0.25	5	2.720	5.053	1.738	99.61

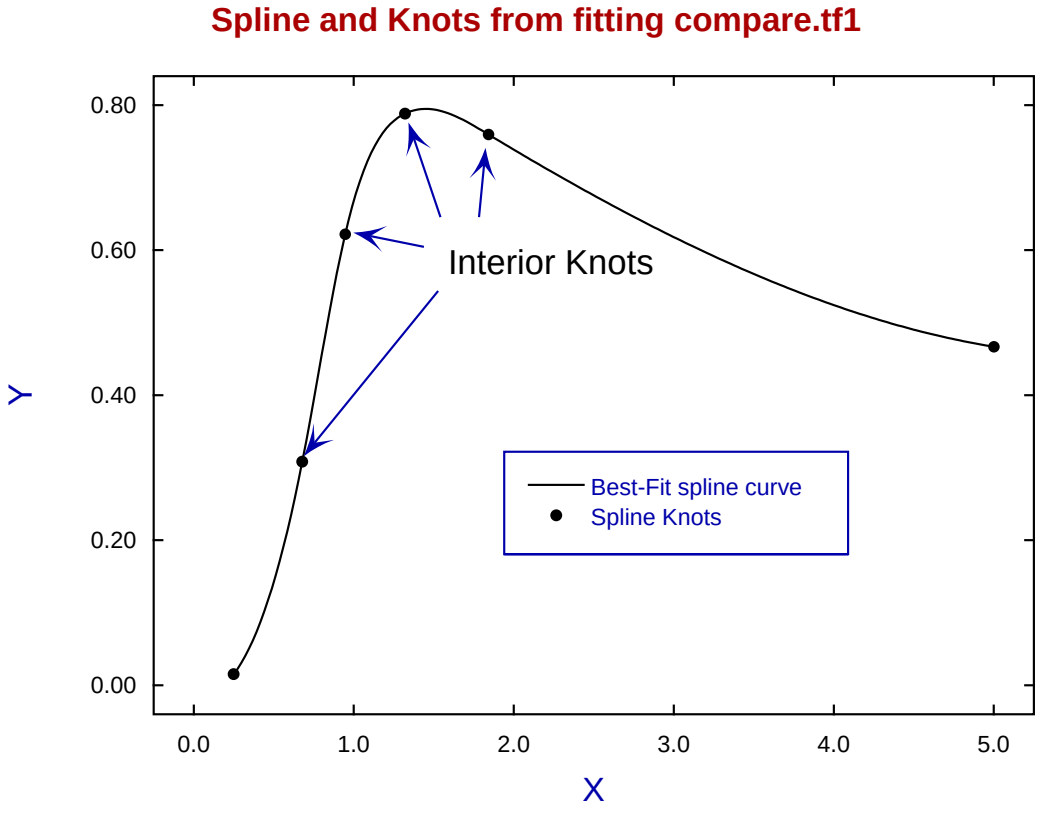
The area under the curve estimate is very valuable for calculating AUC in those pharmacology studies where exponentials cannot be fitted by program **exfit** to quantify drug availability.

The arc length estimate can be used in studies where response along a path is more meaningful than simply over the range of dependent or independent variable traversed.

The estimated absolute curvature integral is given in radial measure and degrees, and summarizes the amount of oscillation in the best-fit curve.

Plot a spline

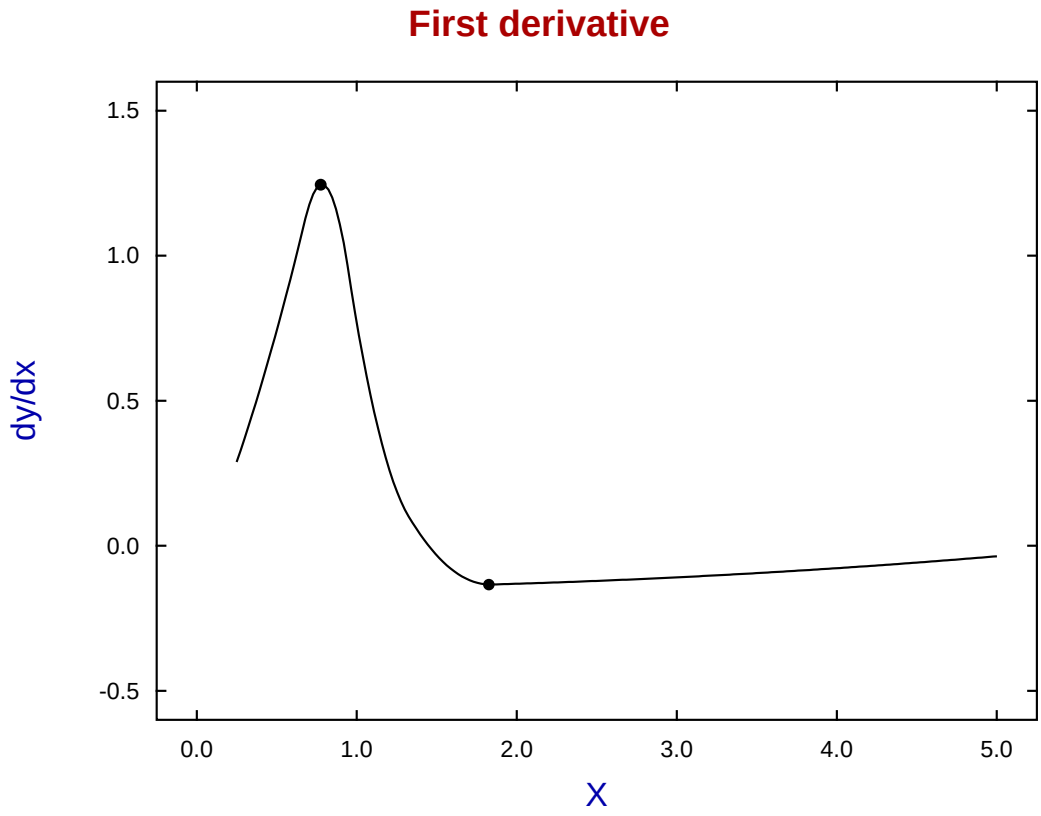
The best-fit spline curve and the position of the interior knots can be plotted as follows.



Plot the derivative

This estimates extreme values of dy/dx and corresponding x values as in the next table and graph.

Minimum independent variable	0.2500	Maximum independent variable	5.000
Minimum selected value (A)	0.2500	Maximum selected value (B)	5.000
Minimum first derivative	-0.1335	in range(A, B) at x-value	1.825
Maximum first derivative	1.245	in range(A, B) at x-value	0.7751



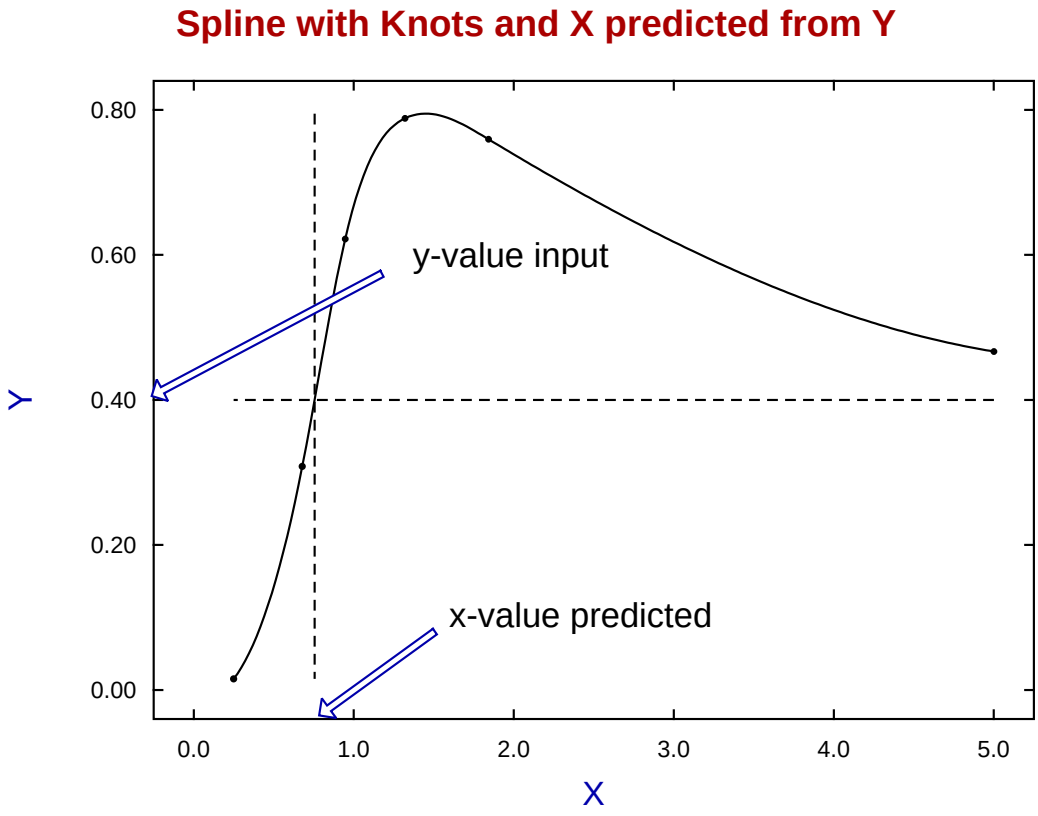
This procedure can be used to estimate the maximum and minimum growth rates with data sets are too noisy, or that cannot be analyzed for some other reason by program **gcfit**.

Calibrate by predicting x given y

Spline reference curves are useful with noisy data, or in situations where bias would result from using an ill-fitting deterministic equation, e.g. LD50 determination.

From requesting a value for x predicted given a value of $y = 0.4$ we get the next result and graph illustrating the values input and predicted as dotted lines.

For $Y = 0.4$, X predicted = 0.75545



X-Range: select $A = X_{min}$, $B = X_{max}$

The ability to vary the range of investigation has many applications because, if the values for A and B are such that A is larger than the lowest point where the spline is defined, or B is less than the highest point where the spline is defined, then calculations for areas, arc length, curvature, derivatives, and calibration will refer to the new restricted range of values $A \leq x \leq B$.

For instance, from the previous graph it will be clear that, because of the turning point, some attempts to predict x_0 given y_0 will fail because of ambiguity leading to two intersections of the line $y = y_0$ with the best-fit spline curve. This is because, for a spline defined as $f(x)$, SimFIT calculates $y_A = f(A) - y_0$ and $y_B = f(B) - y_0$ in order use these as starting estimates for solving the equation

$$f(x_0) - y_0 = 0$$

numerically, and iteration cannot commence when $y_A y_B > 0$.

To illustrate how this problem can be circumvented we calculate the two solutions for $y = 0.6$.

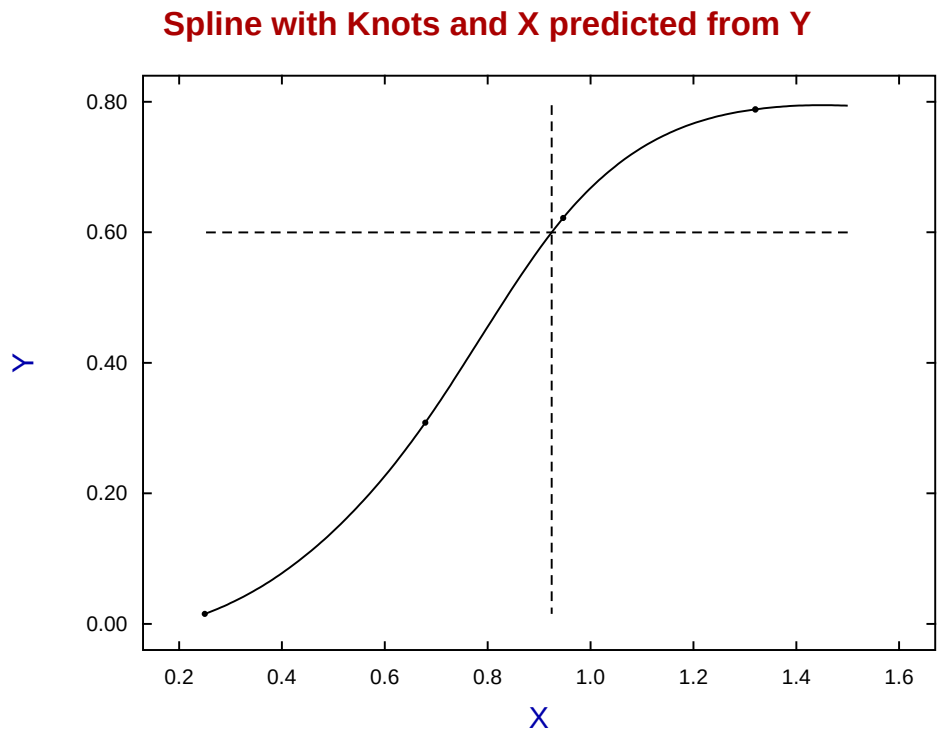
First we select the range $A = 0.25, B = 1.5$ which gives the lower solution.

For Y = 0.6, X predicted = 0.92438

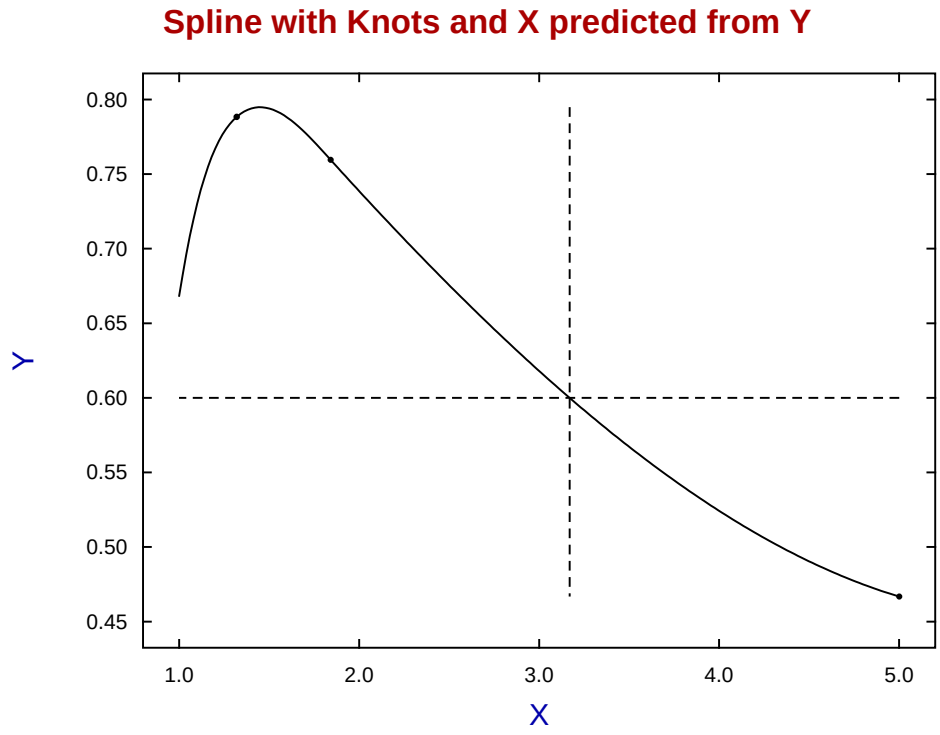
Then we re-define $A = 1.0, B = 5$ which gives the upper solution.

For Y = 0.6, X predicted = 3.1696

This is the graph for the low range solution



while this is the graph for the high range solution.



Theory

As splines are defined by knots and coefficients rather than equations, special techniques are required to re-use best fit spline functions. SimFIT provides procedures to save spline parameters to a file so that they can be re-used to restore previously fitted spline functions. This is particularly useful when a best-fit spline is to be re-used as reference function or standard curve, as in calibration.

Input a spline file such as `spline.tf1` into program **spline** to appreciate how to re-use a best fit spline stored from **spline**, **calcurve**, or **compare**, to estimate derivatives, areas, curvatures and arc lengths.

SimFIT spline files of length $k \geq 12$, such as `spline.tf1`, have $(k+4)/2$ knots, then $(k-4)/2$ coefficients as follows.

- There must be at least 8 nondecreasing knots
- The first 4 of these knots must all be equal to the lowest x value
- The next $(k-12)/2$ must be the non-decreasing interior knots
- The next 4 of these knots must all be equal to the highest x value
- Then there must be $(k-4)/2$ spline coefficients c_i

With $\bar{n}$ spline intervals (i.e. one greater than the number of interior knots), $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ are knots corresponding to the lowest x value, $\lambda_5, \lambda_6, \ldots, \lambda_{\bar{n}+3}$ are interior knots, while $\lambda_{\bar{n}+4}, \lambda_{\bar{n}+5}, \lambda_{\bar{n}+6}, \lambda_{\bar{n}+7}$ correspond to the largest x value. Then the best-fit spline $f(x)$ is

$$f(x) = \sum_{i=1}^{\bar{n}+3} c_i B_i(x).$$

where the c_i are the spline coefficients, and the $B_i(x)$ are normalized B-splines of degree 3 defined on the knots $\lambda_i, \lambda_{i+1}, \ldots, \lambda_{i+4}$.

When the knots and coefficients are defined in this way, the function $y = f(x)$ can be used as a model-free best fit curve to obtain point estimates for the derivatives y', y'', y''', as well as the area AUC, arc length L, or total absolute curvature κ over a range $\alpha \leq x \leq \beta$, defined as

$$\begin{aligned} AUC &= \int_\alpha^\beta y\,dx \\ L &= \int_\alpha^\beta \sqrt{1+y'^2}\,dx \\ \kappa &= \int_0^L \frac{|y''|}{(1+y'^2)^{\frac{3}{2}}}\,dl \\ &= \int_\alpha^\beta \frac{|y''|}{1+y'^2}\,dx \end{aligned}$$

which are valuable parameters to use when comparing data sets. For instance, the arc length L provides a valuable measure of the length of the fitted curve, while the total absolute curvature indicates the total angle turned by the tangent to the curve and indicates the amount of oscillatory behavior.

9.2.3 Using cubic splines to compare curves

Cubic splines can be used for nonparametric comparison of two data sets for similarities and differences. Splines under tension are first fitted to each data set, then the areas under each curve and the absolute area between them are estimated using the trapezoidal method, integration of the best-fit curves, and Simpson's method for the absolute differences, in order to express differences as percentages.

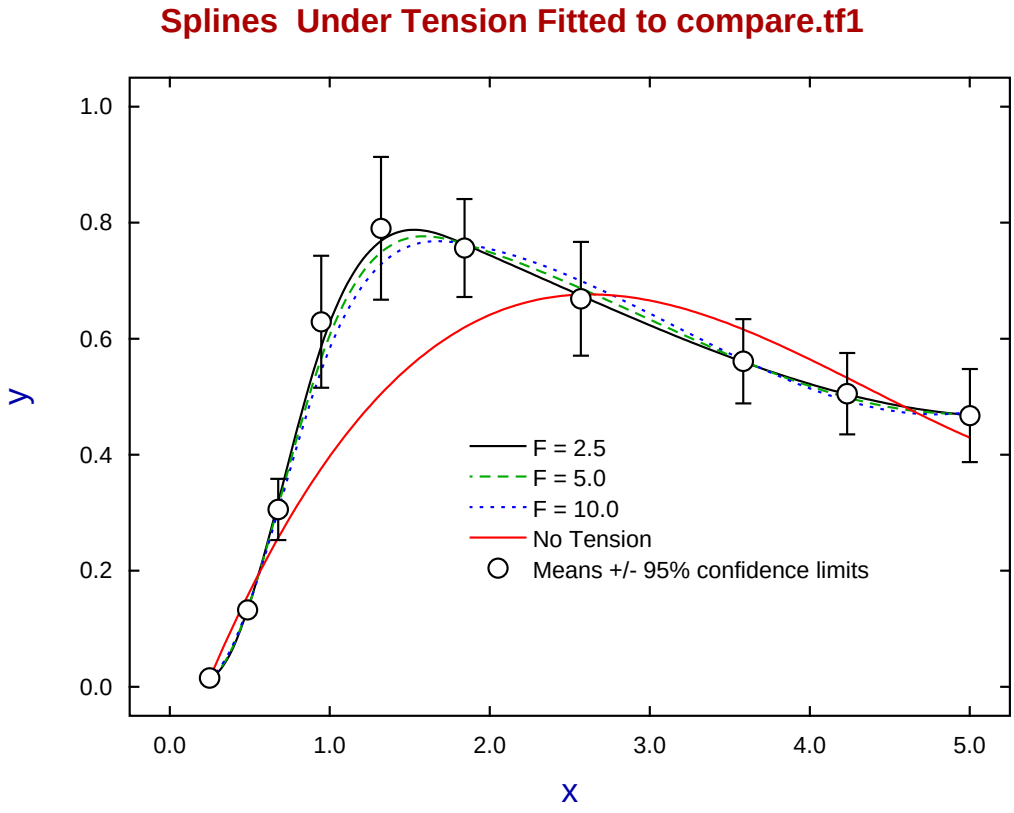
From the main SimFIT menu select [A/Z], open program **compare** then view the default test file `compare.tf1` containing the following data.

x	*y*	*se*
0.25000	0.017267	1
0.25000	0.015585	1
0.25000	0.014268	1
0.25000	0.014136	1
0.48647	0.12861	1
0.48647	0.12536	1
0.48647	0.13339	1
0.48647	0.14230	1
0.67860	0.26261	1
0.67860	0.34277	1
0.67860	0.30364	1
0.67860	0.31373	1
0.94662	0.67252	1
0.94662	0.70382	1
0.94662	0.59192	1
0.94662	0.54850	1
1.3205	0.90417	1
1.3205	0.74158	1
1.3205	0.77353	1
1.3205	0.74208	1
1.8420	0.79030	1
1.8420	0.79384	1
1.8420	0.67971	1
1.8420	0.76176	1
2.5695	0.58575	1
2.5695	0.66178	1
2.5695	0.70023	1
2.5695	0.72772	1
3.5844	0.53286	1
3.5844	0.62744	1
3.5844	0.55484	1
3.5844	0.52923	1
4.2334	0.55003	1
4.2334	0.46641	1
4.2334	0.46840	1
4.2334	0.53647	1
5.0000	0.49920	1
5.0000	0.51847	1
5.0000	0.44355	1
5.0000	0.40895	1

The columns contain data in the following format.

1. **Column 1**: the variable x which must be in non-decreasing order.
2. **Column 2**: the response y presumed to be dependent on x.
3. **Column 3**: the value of 1 indicates that the replicates will be used to calculate the sample standard deviations at each x-replicate value to be used for weighting.
   This column can be omitted or set to a positive value se if it is wished to supply weighting factors w directly which would then be used as $w = 1/se^2$.

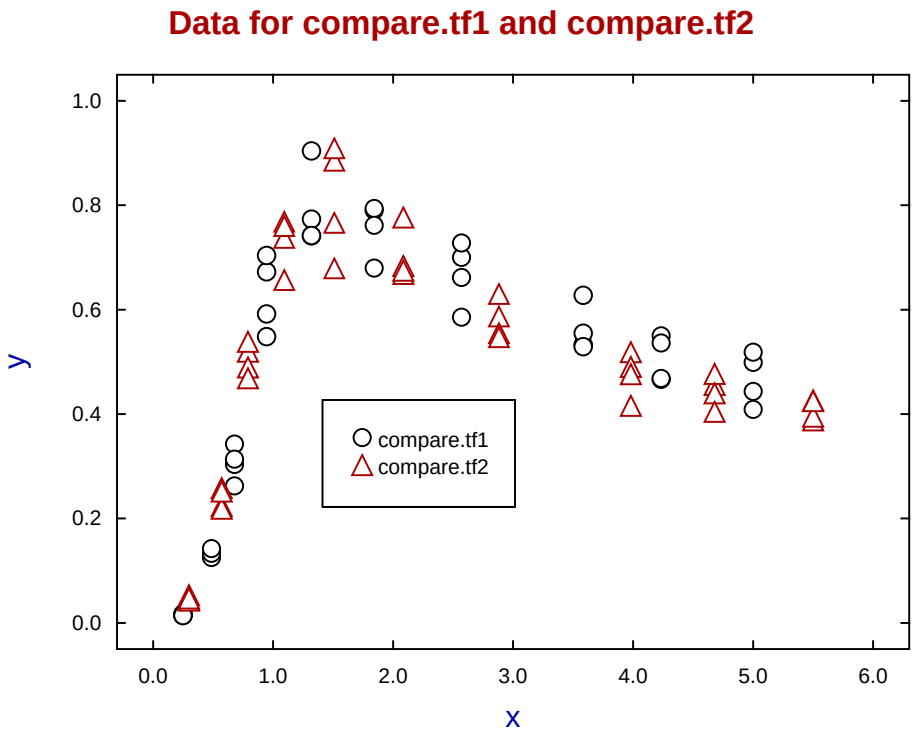
Splines were fitted with the default smoothing factor which simply fits a cubic with no internal knots, then the smoothing factor F was decreased to 10, which is the number of data points after replicates in the data were replaced by means, which gave a distinct improvement in fit. Then, as will be appreciated from the next diagram, increasing the tension by halving the smoothing factor to $F = 5$ then $F = 2.5$ gave very little subsequent improvement.

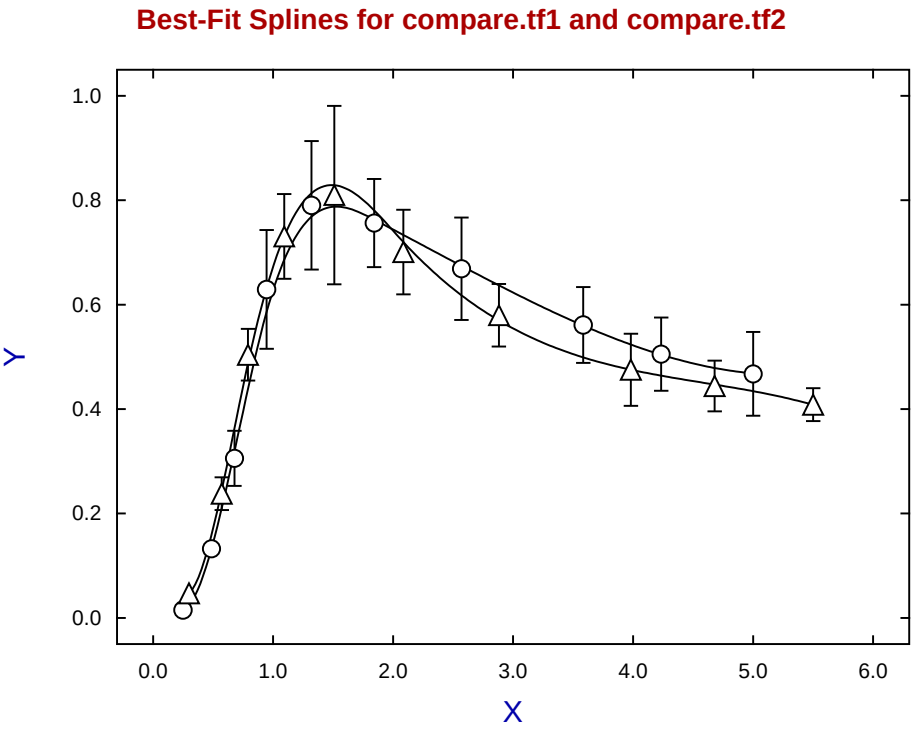


The following table summarizes the conclusion that, in this case, the trapezoidal estimate was very close to the area under the best-fit spline. Two figures are given for the percentage which depend on whether the absolute difference is scaled by the sum of areas or, perhaps more sensibly in some cases, by their average.

Area by trapezoidal rule (A)	2.7151
Area under best-fit spline (B)	2.7087
Absolute difference (C = |A - B|)	0.0063784
Fractional difference C/(A + B)	0.0012
Percent difference between A and B	0.1176% (denominator = sum)
Fractional difference C/[0.5(A + B)]	0.0024
Percent difference between A and B	0.2352% (denominator = average)

The next figures illustrate the comparison of data in test file `compare.tf1` and `compare.tf2`, first with original data, then with means, 95% confidence limits and best-fit splines.





It is clear that these data sets are very similar, and this is quantified by the next table.

Comparison of data sets and best-fit curves	
Area under curve 1 ($0.25 < x < 5.0$) (A_1)	2.7087
Area under curve 2 ($0.3 < x < 5.5$) (A_2)	2.8351
For window number 1: $0.3 < x < 5.0$, y_{min} = 0	
For window number 2: $0.3 < x < 5.0$, y_{min} = 0.024346	
Area under curve 1 inside window 1 (B_1)	2.7077
Area under curve 2 inside window 1 (B_2)	2.6241
Integral of |curve 1 - curve 2| for the x-overlap (A_0)	0.20507
Area under curve 1 inside window 2 (C_1)	2.5933
Area under curve 2 inside window 2 (C_2)	2.5096

Estimated percentage differences between the curves	
Over total range of x values: $100\|A_1 - A_2\|/(A_1 + A_2)$	2.2808%
In window 1 (with a zero baseline): $100(A_0)/(B_1 + B_2)$	3.8462%
In window 2 (with y_{min} baseline): $100(A_0)/(C_1 + C_2)$	4.0187%
Over total range of x values: $200\|A_1 - A_2\|/(A_1 + A_2)$	4.5616%
In window 1 (with a zero baseline): $200(A_0)/(B_1 + B_2)$	7.6924%
In window 2 (with y_{min} baseline): $200(A_0)/(C_1 + C_2)$	8.0374%
Conclusion: *Comparison of curves is good (likely to be identical)*	

Note that corrections may have to be made if the ranges of x are not identical for both data sets, if negative y values are encountered, or if the smallest y value is not zero, so these estimates have the following meanings.

1. **Areas under curves** A_1, A_2
   These are calculated for the best-fit spline curves over the ranges indicated without any corrections.

2. **Windows**
   These are defined as rectangles where the x ranges overlap and, if no y value is zero, or if any y values are negative, these are corrected by the parameter y_{min}. In window 1 $y_{min} = 0$, but in window 2 y_{min} is the smallest value that must be used to correct the y values to make sure the minimum y value is zero, and that all areas are for integration of nonnegative curves.

3. **Area under curves inside windows** B_1, B_2, C_1, C_2
   The values C_1, C_2 are corrected, if required, depending on y_{min}.

4. **Integral of absolute difference**
   This is evaluated by using Simpson's rule with the integrand defined as the absolute value of the difference between curves, but only over the range of x-overlap.

5. **100 or 200 in percentage calculations**
   100 is used to refer to the sum of areas in the denominator so that the value cannot exceed 100%, whereas 200, which refers to the average of areas in the denominator, can cause confusion as it can be as large as 200%.

6. **Conclusion**
   This is an arbitrary qualitative decision based on considering all the values in the above table.

The most useful values from this table are the last two giving the percentage difference between the curves with a zero baseline and when using a baseline shift, and also using the average of the two areas in the denominator when calculating the ratios, which is preferred when $B_1 \approx B_2$ and $C_1 \approx C_2$.

Archiving spline files

When a best-fit spline has been calculated the knots and coefficients can be saved to a spline file. This can be used during the running of program **compare** or can be input into program **spline** for retrospective analysis, such as using as a standard curve for calibration.

9.3 Smooth interpolation of discrete data

Smooth interpolation of a sparse set of exact (x_i, y_i) points consists of filling in gaps where function values are changing rapidly, by adding extra points in order to create a better curve when the data are plotted or printed as a smooth curve.

There are two cases.

1. The x, y data points were generated from a function $y = f(x)$ where y is a single–valued function of x.

   This situation is encountered when a best fit curve has been obtained by evaluating a model equation at equally spaced points and has a peak or valley where closer spaced points would have been better able to capture the turning points.

2. The x, y data points were generated as the solutions to an implicit function $g(x, y) = 0$, for instance parametrically as $x(t)$, $y(t)$.

   This may be necessary when orbits of a system of differential equations have been plotted and there may be multiple values of y given x.

Of course the problem can usually be rectified by simply repeating the calculation using more function values, but it is sometimes required to be wise after the event, e.g., when preparing results for publication when parameter values are no longer available. More trivially, when preparing a graph for publication the ability to arbitrarily smooth chosen curves specified by known mathematical functions can be much appreciated.

It must be admitted that we cannot get something for nothing and that any interpolation technique must be understood and controlled otherwise spurious excursions and undulations could be introduced.

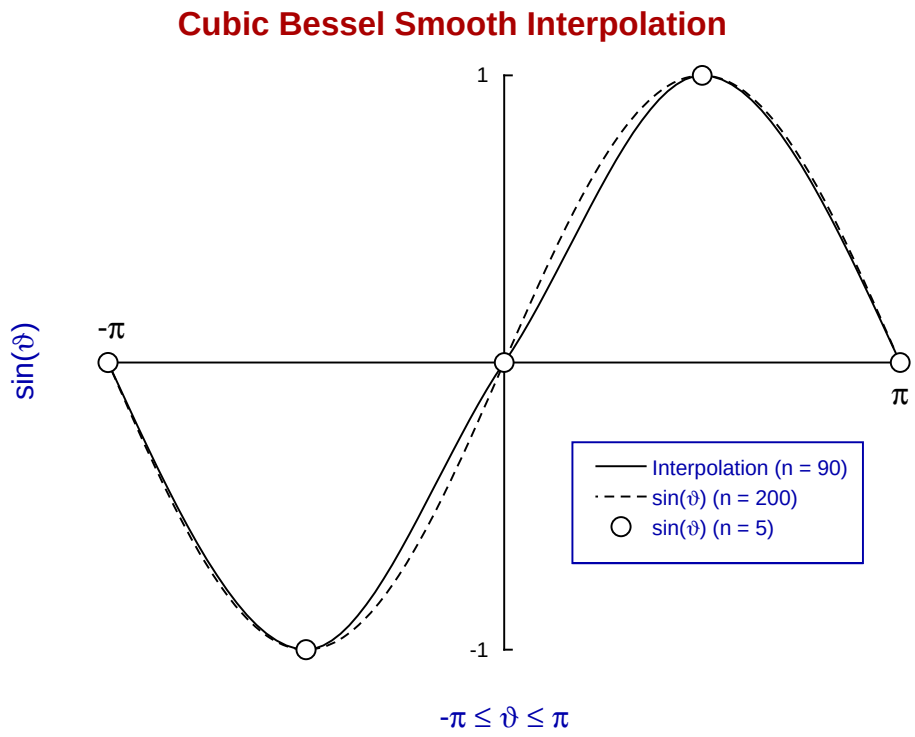
Potential problems with interpolation

Most methods for smooth interpolation use some sort of cubic polynomial smoothing and the methods provided by SimFIT are based on using cubics which are specified by treating neighboring points to generate local cubics in much the the same way as knots are used in global spline fitting.

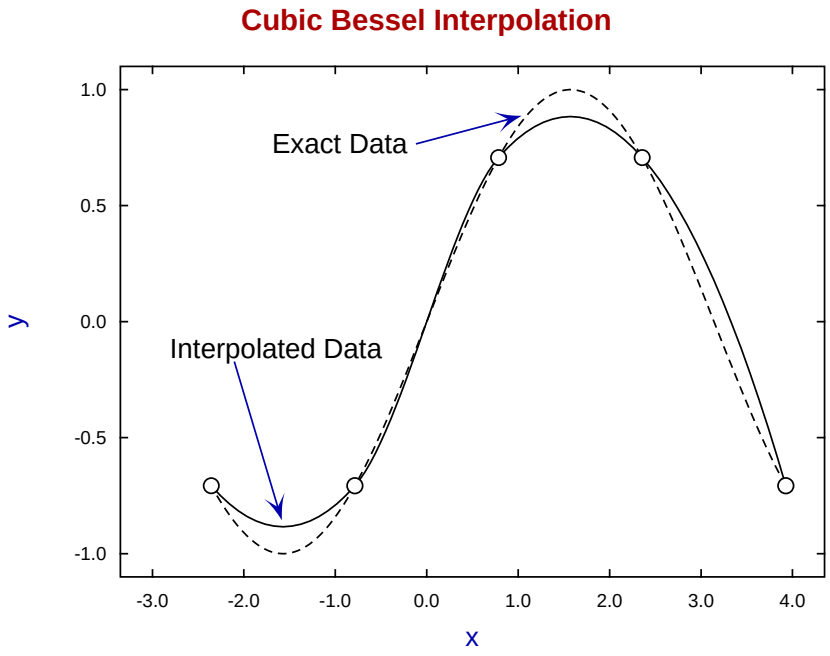
Evidentally, placing knots strategically is vital in order to avoid unwanted oscillations and we commence by using the example of a simple sine curve with the following data, which is to be smoothed using the cubic Bessel method.

x	y
-3.1415927	0.0
-1.5707963	-1.0
0.0000000	0.0
1.5707963	1.0
3.1415927	0.0

Now all computer generated curves are actually constructed by joining defined pixels by straight lines, and in the next graph we can see how the interpolated curve with 90 points generated from the original 5 five points is remarkably close to the exact curve with 200 points.

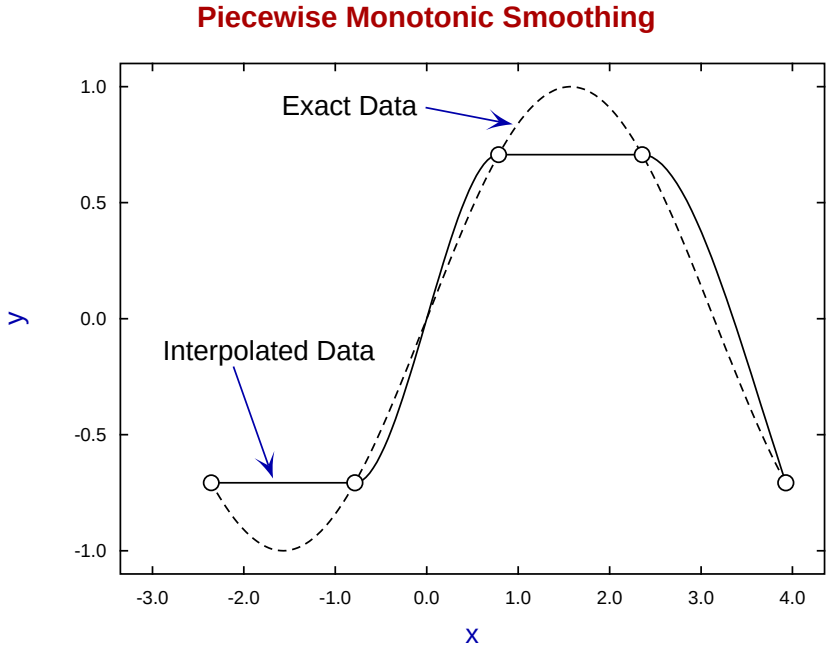


To illustrate the effect of displacing the original data from the extreme positions, consider next the result when the given x data are simply incremented by $\pi/4$. Because of this shift from the critical coordinates the next graph indicates a very much worse fit because the original data were not situated at the turning points leading to undershoot. This is one of the problems when interpolating sparse data with turning points when undershoot or overshoot can occur with the cubic Bessel technique.



As undershoot or overshoot can give a misleading impression of false turning points when observing the

closeness of data to a best–fit curve or when plotting the profile from numerical solution of differential equations, it is necessary to impose constraints by using the piecewise monotonic method which preserves the sign of $f'(x)$ between successive x values as shown next.



As this method does not allow the interpolated data to exceed obvious turning points by setting $f'(x) = 0$ at extreme y_i points it avoids overshoot and undershoot, but at the expense of straight line sections.

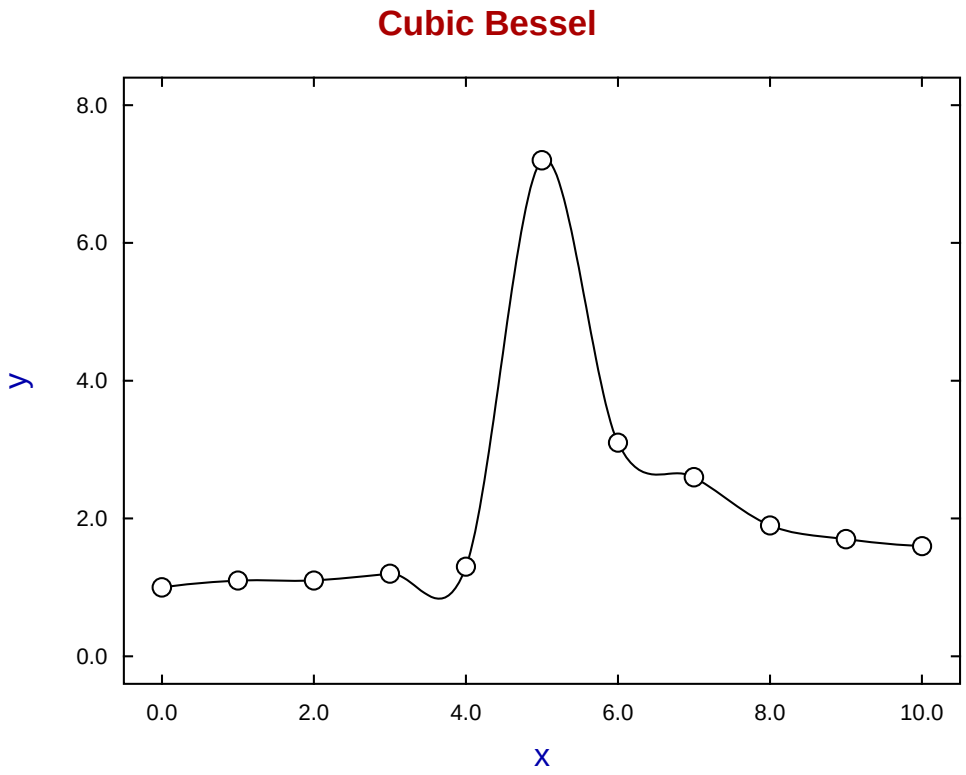
Example 1: smoothing a single valued function

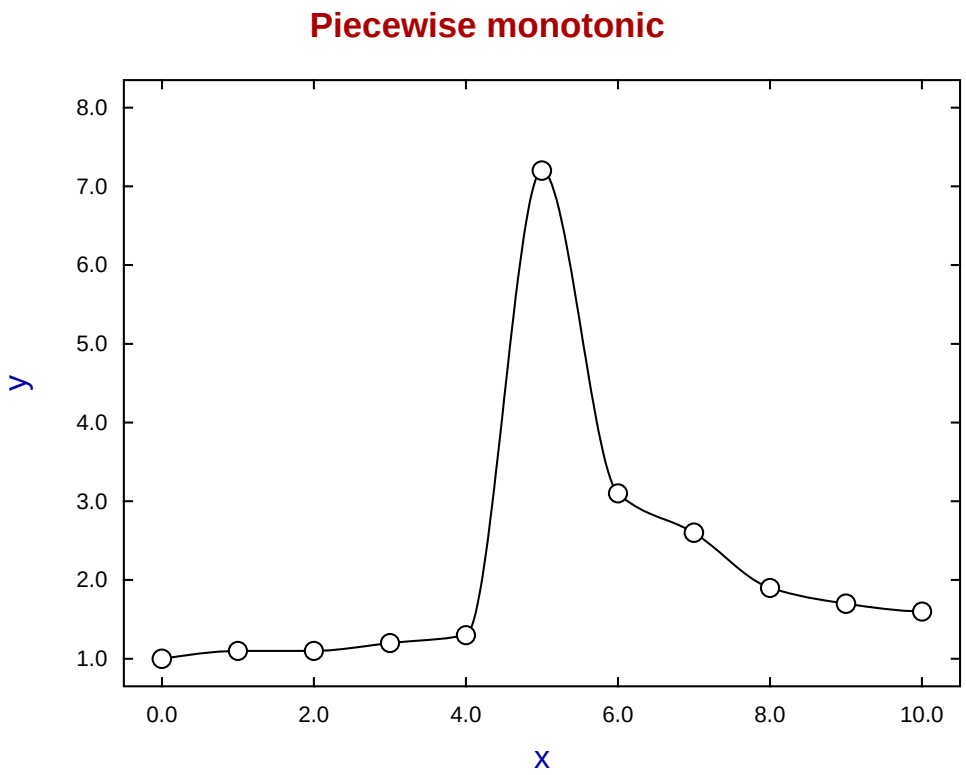
From the main SIMFIT menu choose the [Statistics] option then the [Data smoothing] option and select smooth interpolation of discrete data for $y = f(x)$, which uses the following data contained in the test file `j07caf.tf1`.

x	y
0.0	1.0
1.0	1.1
2.0	1.1
3.0	1.2
4.0	1.3
5.0	7.2
6.0	3.1
7.0	2.6
8.0	1.9
9.0	1.7
10.0	1.6

Note that you can decide whether to use the cubic Bessel or piecewise monotonic method, and also you can specify the tolerance factor. This would usually be satisfactory with the default value of 2000. Note that decreasing this factor will result in less smooth curves but with fewer extra interpolated points.

The difference between the possibly over flexible cubic Bessel result and rather stiffer piecewise monotonic curve will be clear from the next two graphs.


Cubic Bessel
8.0
6.0
4.0
2.0
0.0
y
0.0
2.0
4.0
6.0
8.0
10.0
x

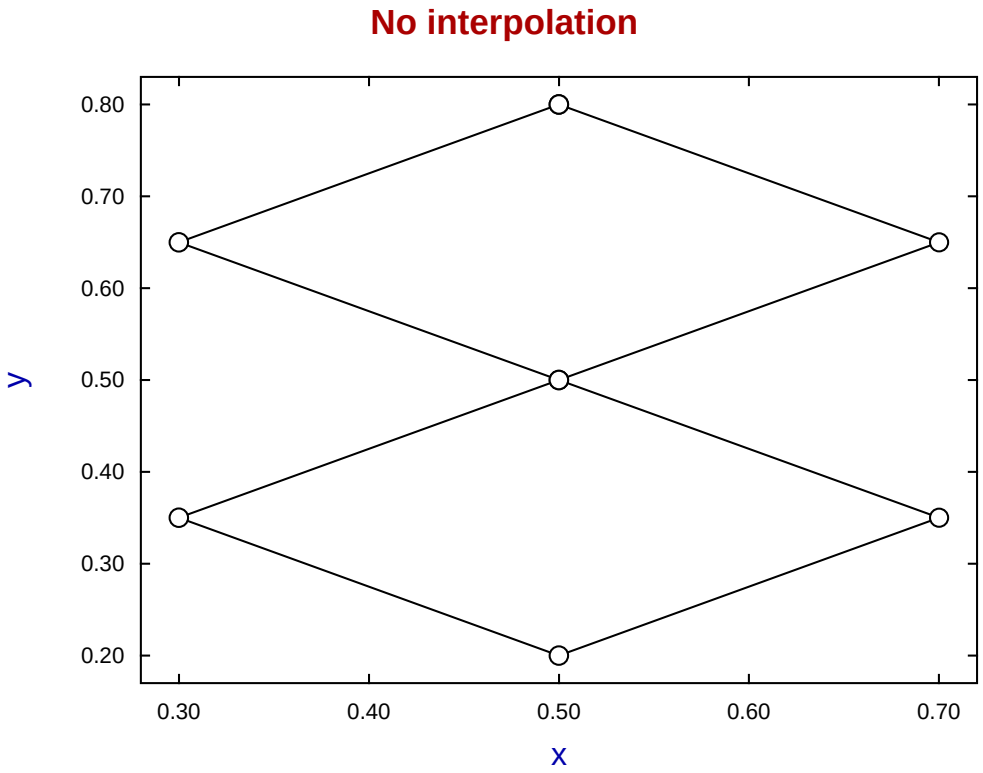

Piecewise monotonic
8.0
7.0
6.0
5.0
4.0
3.0
2.0
1.0
y
0.0
2.0
4.0
6.0
8.0
10.0
x

Example 2: Smoothing a parametric curve

From the main SIMFIT menu choose the [Statistics] option then the [Data smoothing] option and select smooth interpolation of discrete data for $x(t), y(t)$) which uses the following data contained in the test file `j07ccf.tf1`.

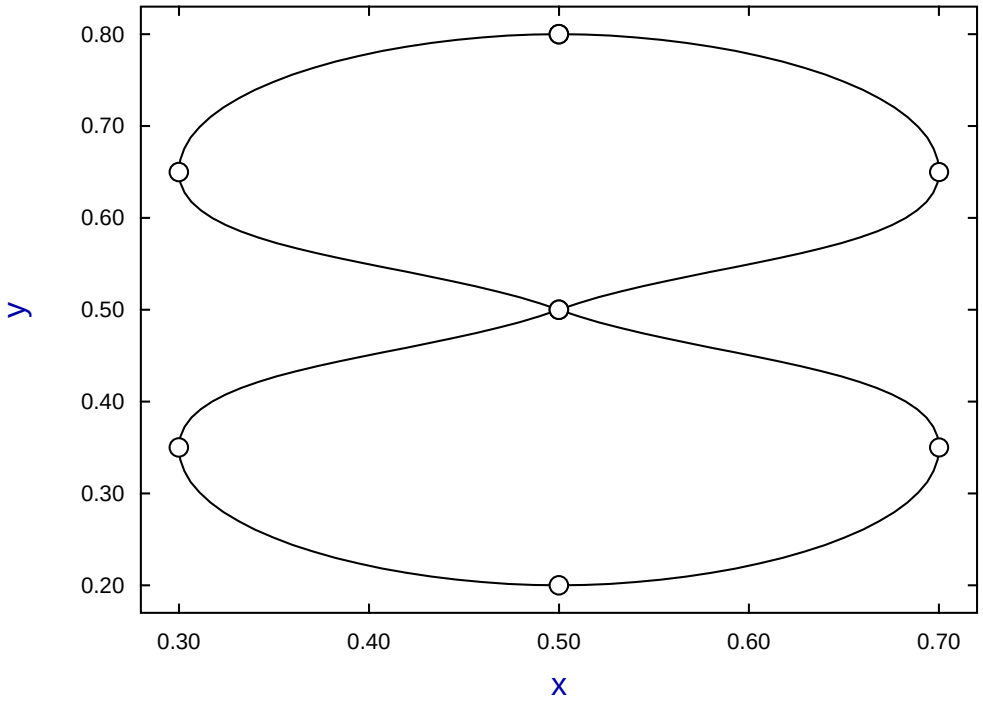
x	y
0.5	0.80
0.7	0.65
0.5	0.50
0.3	0.35
0.5	0.20
0.7	0.35
0.5	0.50
0.3	0.65
0.5	0.80

Note that these data have the last pair of coordinates equal to the first pair, as this technique is required to indicate a closed loop. Also observe that the parameter t does not occur in the data file, only the (x_i, y_i) coordinates. For instance, a circle defined parametrically by $x = r\cos\theta$, $y = r\sin\theta$ can also be expressed in the implicit form $x^2 + y^2 = r^2$ without reference to the parameter θ. First consider simply joining up these coordinates with no smoothing.

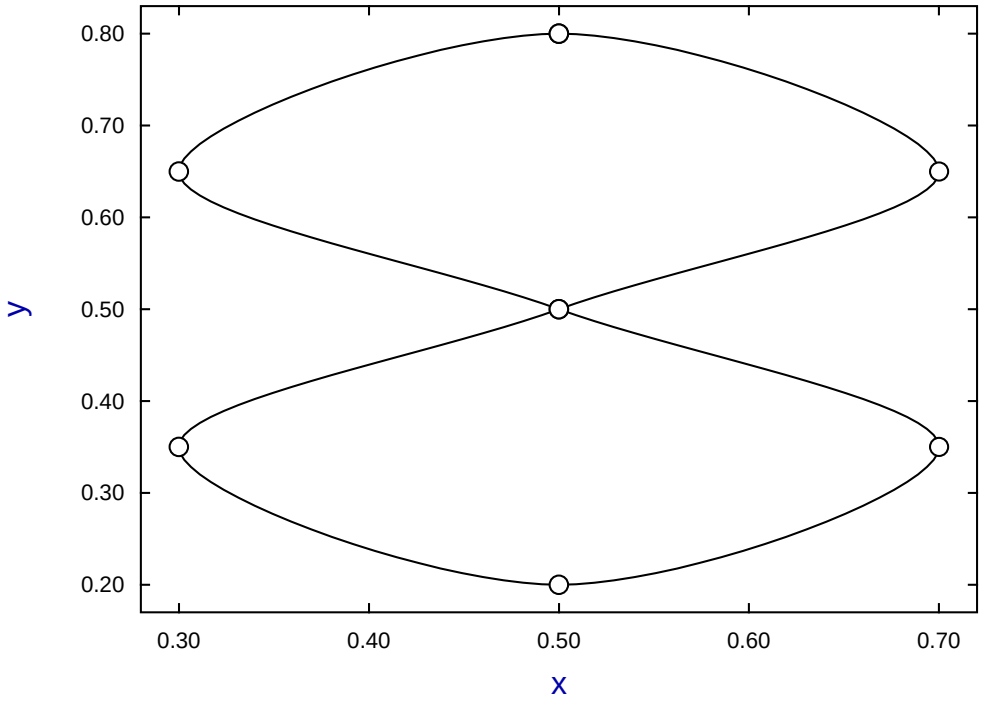


Now the McConalogue method which is analogous to the cubic Bessel providing a flexible curve and the Butland method which is similar to the piecewise monotonic method and yields a stiffer curve.

McConalogue method



Butland method



Theory

The SIMFIT program **spline** can be used to make a global spline interpolation for a single valued function but the user must then specify the additional interpolation points. The reason to use these local methods is that they can be used interactively to achieve satisfactory visual smoothness from within the SIMFIT graphics utilities. They are perfectly adequate for this purpose as will be explained elsewhere for advanced curve fitting using program **qnfit**, differential equation simulation and fitting using program **deqsol**, and advanced graph plotting and contouring using program **simplot**.

The code used is based on development of the two routines `j06caf` and `j06ccf` from the discontinued NAG graphics library which differ from the original NAG routines by supplying the discrete data on input but then returning the enlarged and interpolated data set. They can be called by programmers who use the Simdem package with the following details.

- For single valued functions

```
subroutine smooth$(n, nmax, x, y)
integer,          intent (in)    :: nmax
integer,          intent (inout) :: n
double precision, intent (inout) :: x(nmax), y(nmax)
```

Here we must have `x` monotonically increasing (or decreasing). The value of `n` returned is increased and the coordinates returned contain the interpolated values, so the dimension `nmax` for `x` and `y` in the calling program must be large enough to contain the enlarged data set.

For configuration use

```
subroutine j06cfg_1(isend, itolf, method)
integer, intent (in)    :: isend
integer, intent (inout) :: itolf, method
```

where `isend = 1` sets the parameters directly otherwise an interactive interface is provided, while `method = 1` uses the piecewise monotonic method and `method = 2` uses the cubic Bessel technique.

- For parametric functions

```
subroutine contr1$(n, nmax, x, y)
integer,          intent (in)    :: nmax
integer,          intent (inout) :: n
double precision, intent (inout) :: x(nmax), y(nmax)
```

To use the end points for a closed polygon the first and last coordinate pairs must be equal.

For configuration use

```
subroutine j06cfg(isend, itolf, method)
integer, intent (in)    :: isend
integer, intent (inout) :: itolf, method
```

where `isend = 1` sets the parameters directly otherwise an interactive interface is provided, while `method = 1` uses the Butland method and `method = 2` uses the McConalogue technique.

The tolerance parameter `ITOLF` controls the number of additional interpolated points required before interpolation is satisfactory, and the default value is 2000. Increasing `ITOLF` beyond this value is not likely to have much effect, while decreasing `ITOLF` down to about 200 generates fewer additional interpolation points and progressively decreases the smoothness.

This will be clear from the details as follows.

For `smooth$` the cubic $c(x)$ and piecewise linear polynomial $p(x)$ must satisfy

$$|c(x) - p(x)| \leq \frac{|YMAX - YMIN|}{ITOLF}.$$

For `contr1$` the piecewise cubic $(x(t), y(t))$ and piecewise linear polynomial $(p_x(t), p_y(t))$ must satisfy

$$|x(t) - p_x(t)| \leq \frac{|XMAX - XMIN|}{ITOLF}$$
$$\text{and } |y(t) - p_y(t)| \leq \frac{|YMAX - YMIN|}{ITOLF}.$$

9.4 Calibration

Tutorials and worked examples for simulation, curve fitting, statistical analysis, and plotting.
https://simfit.uk
https://simfit.org.uk
https://simfit.silverfrost.com

9.4.1 Introduction to calibration and bioassay

It is convenient to deal separately with univariate calibration and bioassay, as multivariate calibration is performed in SIMFIT using the partial least squares (PLS) technique.

Calibration

This requires fitting a curve $y = f(x)$ to a (x, y) training data set with x known exactly and y measured with limited error, so that the best fit model $\hat{f}(x)$ can then be used to predict x_i given arbitrary y_i. Usually the model is of no significance and steps are taken to use a data range over which the model is approximately linear, or at worst a shallow smooth curve. It is assumed that experimental errors arising when constructing the best fit curve are uncorrelated and normally distributed with zero mean, so that the standard curve is a good approximation to the maximum likelihood estimate.

- **Calibration curves**

  Creating and using a standard calibration curve involves:

  1. Measuring responses y_i at fixed values of x_i, and using replicates to estimate s_i, the sample standard deviation of y_i if possible.
  2. Preparing a curve fitting type file with x, y, and s using program **makfil**, and using **makmat** to prepare a vector type data file with x_i values to predict y_i.
  3. Finding a best fit curve $y = f(x)$ to minimize $WSSQ$, the sum of weighted squared residuals.
  4. Supplying y_i values and predicting x_i together with 95% confidence limits, i.e. inverse-prediction of $x_i = \hat{f}^{-1}(y_i)$. Sometimes you may also need to evaluate $y_i = \hat{f}(x_i)$.

  It may be that the s_i are known independently, but often they are supposed constant and unweighted regression, i.e. all $s_i = 1$, is unjustifiably used. Any deterministic model can be used for $f(x)$, e.g., a sum of logistics or Michaelis-Menten functions using program **qnfit**, but this could be unwise. Calibration curves arise from the operation of numerous effects and cannot usually be described by one simple equation. Use of such equations can lead to biased predictions and is not always recommended. Polynomials are useful for gentle curves as long as the degree is reasonably low (≤ 3 ?) but, for many purposes, a weighted least squares data smoothing cubic spline is the best choice. Unfortunately polynomials and splines are too flexible and follow outliers, leading to oscillating curves, rather than the data smoothing that is really required. Also they cannot fit horizontal asymptotes. You can help in several ways.

  a) Get good data with more distinct x-values rather than extra replicates.

  b) If the data approach horizontal asymptotes, either leave some data out as they are no use for prediction anyway, or try using $\log(x)$ rather than x, which can be done automatically by program **calcurve**.

  c) Experiment with the weighting schemes, polynomial degrees, spline knots or constraints to find the optimum combinations for your problem.

  d) Remember that predicted confidence limits also depend on the s values you supply, so either get the weighting scheme right, or set all all $s_i = 1$.

- **Turning points in calibration curves**

  Some programs will warn you if $f(x)$ has a turning point, since this can make inverse prediction ambiguous. You can then re-fit to get a new curve, eliminate bad data points, get new data, etc., or carry on if the feature seems to be harmless. You will be given the option of searching upwards or downwards for prediction in such ambiguous cases. It should be obvious from the graph, nature of the mathematical function fitted, or position of the turning point in which direction the search should proceed.

- **Calibration using polynomials**

  For linear or almost linear data you can use program **linfit** which just fits straight lines of the form

  $$f(x) = p_0 + p_1 x.$$

  However, for smooth gentle curves, program **polnom** is preferred because it can also fit a polynomial

  $$f(x) = p_0 + p_1 x + p_2 x^2 + \cdots + p_n x^n,$$

  where the degree n is chosen according to statistical principles. What happens is that **polnom** fits all polynomials from degree 0 up to degree 6 and gives statistics necessary to choose the statistically justified best fit n. However, in the case of calibration curves, it is not advisable to use a value of n greater than 2 or at most 3, and warnings are issued if the best fit standard curve has any turning points that could make inverse prediction non-unique.

- **Calibration using cubic splines**

  If a polynomial of degree 2 or at most 3 is not adequate, a cubic spline calibration curve could be considered. It does not matter how nonlinear your data are, **calcurve** can fit them with splines with several types of knots and tension. The best-fit spline curve from programs such as **calcurve**, **compare**, and **spline** can be archived for repeated initialization of a reference standard curve to use for calibration.

- **Advanced calibration using special models**

  Sometimes you would want to use a specific mathematical model for calibration such as a straight line through the origin, or a quadratic with no linear term, but other models might be more appropriate. For instance, a mixture of two High/Low affinity binding sites or a cooperative binding model might be required for a saturation curve, or a mixture of two logistics might adequately fit growth data. If you know an appropriate model for the standard curve, use **qnfit** for inverse prediction because, after fitting, the best-fit curve can be used for calibration, or for estimating derivatives or areas under curves (AUC) if appropriate.

Bioassay

This is a special type of calibration, where the data are obtained over as wide a range as possible, nonlinearity is accepted (e.g. a sigmoid curve), and specific parameters of the underlying response, such as the time to half-maximum response, final size, maximum rate, area AUC, $EC50$, $LD50$, or $IC50$ are to be estimated. With bioassay, a known deterministic model may be required, and assuming normally distributed errors may sometimes be a reasonable assumption, but alternatively the data may consist of proportions in one of two categories (e.g. alive or dead) as a function of some treatment, so that binomial error is more appropriate and probit analysis, or similar, is called for.

A special type of inverse prediction is required when equations are fitted to dose response data in order to estimate some characteristic parameter, such as the half time $t_{1/2}$, the area under the curve AUC, or median effective dose in bioassay (e.g. $ED50$, $EC50$, $IC50$, $LD50$, etc.), along with standard errors and 95% confidence limits. The model equations used in this sort of analysis are not supposed to be exact models constructed according to scientific laws, rather they are empirical equations, selected to have a shape that is

close to the shape expected of such data sets. So, while it is is pedantic to insist on using a model based on scientific model building, it is important to select a model that fits closely over a wide variety of conditions.

Older techniques, such as using data subjected to a logarithmic transform in order to fit a linear model, are no longer called for as they are very unreliable, leading to biased parameter estimates. Hence, in what follows, it is assumed that data are to be analyzed in standard, not logarithmically transformed coordinates, but there is nothing to prevent data being plotted in transformed space after analysis, as is frequently done when the independent variable is a concentration, i.e., it is desired to have an the independent variable proportional to chemical potential. The type of analysis called for depends very much on the nature of the data, the error distribution involved, and the goodness of fit of the assumed model. It is essential that data are obtained over a wide range, and that the best fit curves are plotted and seen to be free from bias which could seriously degrade routine estimates of percentiles, say. The only way to decide which of the following procedures should be selected for your data, is to analyze the data using those candidate models that are possibilities, and then to adopt the model that seems to perform best, i.e., gives the closest best fit curves and most sensible inverse predictions.

- **Exponential models**

  If the data are in the form of a simple or multiphasic exponential decline from a finite value at $t = 0$ to zero as $t \to \infty$, and half times $t_{1/2}$, or areas AUC are required, use **exfit** to fit one or a sum of two exponentials with no constant term.

- **Trapezoidal estimation**

  If no deterministic model can be used for the AUC it is usual to prefer the trapezoidal method with no data smoothing, where replicates are simply replaced by means values that are then joined up sequentially by sectional straight lines. The program **average** is well suited to this sort of analysis.

- **The Hill equation**

  This empirical equation is

$$f(x) = \frac{Ax^n}{B^n + x^n},$$

  which can be fitted using program **inrate**, with either n estimated or n fixed, and it is often used in sigmoidal form (i.e. $n > 1$) to estimate the maximum value A and half saturation point B, with sigmoidal data (not data that are only sigmoidal when x-semilog transformed, as all binding isotherms are sigmoidal in x-semilog space).

- **Ligand binding and enzyme kinetic models**

  There are three cases:
  a) data are increasing as a function of an effector, i.e., ligand or substrate, and the median effective ligand concentration $ED50$ or apparent $K_m = EC50 = ED50$ is required,
  b) data are a decreasing function of an inhibitor $[I]$ at fixed substrate concentration $[S]$ and $IC50$, the concentration of inhibitor giving half maximal inhibition, is required, or
  c) the flux of labeled substrate $[Hot]$, say, is measured as a decreasing function of unlabeled isotope $[Cold]$, say, with $[Hot]$ held fixed.
  If the data are for an increasing saturation curve and ligand binding models are required, then **hlfit** or, if cooperative effects are present, **sffit** can be used to fit one or two binding site models.

  More often, however, an enzyme kinetic model, such as the Michaelis-Menten equation will be used as now described. To estimate the maximum rate and apparent K_m, i.e., $EC50$ the equation fitted by **mmfit** in substrate mode would be

$$v([S]) = \frac{V_{max}[S]}{K_m + [S]}$$

  while the interpretation of $IC50$ for a reversible inhibitor at concentration $[i]$ with substrate fixed at concentration S would depend on the model assumed as follows.

$$\text{Competitive inhibition } v([I]) = \frac{V_{max}[S]}{K_m(1 + I/K_i) + [S]}$$
$$IC50 = \frac{K_i(K_m + [S])}{K_m}$$
$$\text{Uncompetitive inhibition } v([I]) = \frac{V_{max}[S]}{K_m + [S](1 + [I]/K_i]}$$
$$IC50 = \frac{K_i(K_m + [S])}{[S]}$$
$$\text{Noncompetitive inhibition } v([I]) = \frac{V_{max}[S]}{(1 + [I]/K_i)(K_m + [S])}$$
$$IC50 = K_i$$
$$\text{Mixed inhibition } v([I]) = \frac{V_{max}[S]}{K(1 + [I]/K_{i1}) + [S](1 + [I]/K_{i2})}$$
$$IC50 = \frac{K_{i1}K_{i2}(K_m + [S])}{(K_mK_{i2} + [S]K_{i1})}$$
$$\text{Isotope displacement } v([Cold]) = \frac{V_{max}[Hot]}{K_m + [Hot] + [Cold]}$$
$$IC50 = K_m + [Hot]$$

Of course, only two independent parameters can be estimated with these models, and, if higher order models are required and justified by statistics and graphical deconvolution, the apparent V_{max} and apparent K_m are then estimated numerically.

- **Growth curves**

  If the data are in the form of sigmoidal increase, and maximum size, maximum growth rate, minimum growth rate, $t_{1/2}$ time to half maximum size, etc. are required, then use **gcfit** in growth curve mode. For instance, with the logistic model

$$f(t) = \frac{A}{1 + B\exp(-kt)}$$
$$t_{1/2} = \frac{\log(B)}{k}$$

  the maximum size A and time to reach half maximal size $t_{/2}$ are estimated.

- **Survival curves**

  If the data are independent estimates of fractions remaining as a function of time or some effector, i.e. sigmoidally decreasing profiles fitted by **gcfit** in mode 2, and $t_{1/2}$ is required, then normalize the data to proportions of time zero values and use **gcfit** in survival curve mode 2. The Weibull model

$$S(t) = 1 - \exp(-(At)^B)$$
$$t_{1/2} = \frac{\log(2)}{AB}.$$

  is very useful.

- **Survival time models**

  If the data are in the form of times to failure, possibly censored, then **gcfit** should be used in survival time mode 3. With survival time models the median survival time $t_{1/2}$ is estimated, where

$$\int_0^{t_{1/2}} f_T(t)\,dt = \frac{1}{2},$$

  and $f_T(t)$ is the survival probability density function.

- **Models for proportions**

  If the data are in the form of numbers of successes (or failures) in groups of known size as a function of some control variable and you wish to estimate percentiles, e.g., $EC50$, $IC50$, or maybe $LD50$ (the median dose for survival in toxicity tests), use **gcfit** in GLM dose response mode. This is because the error distribution is binomial, so generalized linear models should be used. You should

95% confidence regions in inverse prediction

polnom estimates non-symmetrical confidence limits assuming that the N values of y for inverse prediction and weights supplied for weighting are exact, and that the model fitted has n parameters that are justified statistically. **calcurve** uses the weights supplied, or the estimated coefficient of variation, to fit confidence envelope splines either side of the best fit spline, by employing an empirical technique developed by simulation studies. Root finding is employed to locate the intersection of the y_i supplied with the envelopes. The AUC, LD50, half-saturation, asymptote and other inverse predictions in SIMFIT use a t distribution with $N - n$ degrees of freedom, and the variance-covariance matrix estimated from the regression. That is, assuming a prediction parameter defined by $p = f(\theta_1, \theta_2, \ldots, \theta_n)$, a central 95% confidence region is constructed using the prediction parameter variance estimated by the propagation of errors formula

$$\hat{V}(p) = \sum_{i=1}^{n} \left(\frac{\partial f}{\partial \theta_i}\right)^2 \hat{V}(\theta_i) + 2 \sum_{i=2}^{n} \sum_{j=1}^{i-1} \frac{\partial f}{\partial \theta_i} \frac{\partial f}{\partial \theta_j} \hat{CV}(\theta_i, \theta_j).$$

Note that this formula for the propagation of errors can be used to calculate parameter standard errors for parameters that are calculated as functions of parameters that have been estimated by fitting, such as apparent maximal velocity when fitting sums of Michaelis-Menten functions. However, such estimated standard errors will only be very approximate.

9.4.2 Using straight line standard curves

When an experimental data set is approximately linear it is possible to fit a straight line and use this as a standard curve for calibration, that is, predicting x with 95% confidence limits given y.

From the main SIMFIT menu choose [A/Z], open program **linfit**, then select simple calibration using a straight line and view `line.tf1`, the test file provided, containing the following data.

x	x	se
1.0000	4.4644	1.7613
1.0000	3.1709	1.7613
1.0000	3.7496	1.7613
1.0000	4.2803	1.7613
2.0000	3.2552	1.6302
2.0000	2.6066	1.6302
2.0000	1.3453	1.6302
2.0000	4.0994	1.6302
3.0000	4.1542	1.0229
3.0000	3.9500	1.0229
3.0000	4.5843	1.0229
3.0000	2.2531	1.0229
4.0000	2.7339	0.88866
4.0000	3.7064	0.88866
4.0000	4.8613	0.88866
4.0000	4.1335	0.88866
5.0000	6.2512	0.83435
5.0000	5.8701	0.83435
5.0000	7.6285	0.83435
5.0000	5.8742	0.83435
6.0000	4.7595	0.43208
6.0000	5.0619	0.43208
6.0000	5.5940	0.43208
6.0000	4.6181	0.43208
7.0000	5.2595	1.2461
7.0000	7.8164	1.2461
7.0000	5.9849	1.2461
7.0000	5.0962	1.2461
8.0000	5.1014	2.4624
8.0000	10.259	2.4624
8.0000	6.8319	2.4624
8.0000	9.8217	2.4624
9.0000	10.568	1.2809
9.0000	9.9537	1.2809
9.0000	7.6227	1.2809
9.0000	9.0264	1.2809
10.000	13.560	1.5105
10.000	10.573	1.5105
10.000	10.966	1.5105
10.000	10.249	1.5105

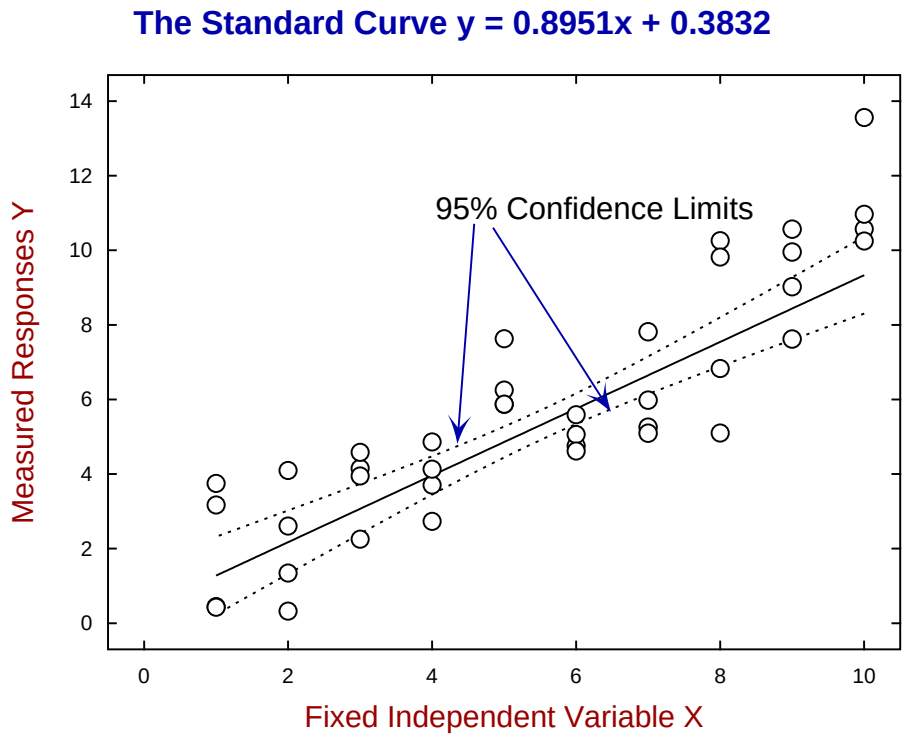
The columns contain data in the following format.

1. **Column 1**: the independent variable x.
2. **Column 2**: the response y presumed to be linearly dependent on the value in column 1.
3. **Column 3**: the positive sample standard deviation of the replicate response measurements. This column can be omitted or set to 1 if unweighted regression is required.

Here are the parameter estimates for the best-fit straight line followed by a graph with confidence limits, which are the branches of a hyperbola for the true values of y at the corresponding fixed values of x.

Parameter	Value	Standard error	Lower95%cl	Upper95%cl	p	
Intercept	0.38322	0.61376	-0.85926	1.6257	0.5361	*
Slope	0.89513	0.10508	0.68241	1.1078	0.0000	

$R^2 = 0.7862, R = 0.8867, p = 0.0000$



The following table gives the prediction of x with 95% confidence limits given y, which are confidence limits for the true value of x given the corresponding true, i.e. fixed exact, values of y. Note that, if a prediction leads to a confidence limit outside of the range of x values, a warning message is displayed, e.g., for the case $y = 2$ and $y = 9$.

y-measured	x-predicted	Lower95%cl	Upper95%cl	
2.0	1.8062	5.3878	2.6274	*Limit out of range*
3.0	2.9234	1.9662	3.5681	
4.0	4.0405	3.3589	4.5434	
5.0	5.1577	4.6680	5.6024	
6.0	6.2748	5.8250	6.8134	
7.0	7.3920	6.8487	8.1578	
8.0	8.5091	7.8092	9.5653	
9.0	9.6263	8.7435	10.999	*Limit out of range*

9.4.3 Using polynomial standard curves

When observations are not linear but suggest that a gentle curve would give a better fit than a straight line, then polynomials can be used to generate a standard curve for calibration analysis. For most applications piecewise cubic splines would probably be better, especially if there is statistical evidence that a polynomial of degree greater than two is required, e.g., a cubic rather than a quadratic.

From the main SimFIT menu choose the [A/Z] option, open program **polnom**, then browse the default test file `polnom.tf1` which contains the following data set.

x	y	s
0.0	0.098421	0.0056072
0.0	0.10950	0.0056072
0.0	0.10248	0.0056072
2.0	3.8448	0.052139
2.0	3.8647	0.052139
2.0	3.9434	0.052139
4.0	6.8490	0.38867
4.0	6.1469	0.38867
4.0	6.2091	0.38867
6.0	8.5864	0.22982
6.0	9.0156	0.22982
6.0	8.6585	0.22982
8.0	9.8616	0.45524
8.0	9.8748	0.45524
8.0	9.0798	0.45524
10.0	9.5218	0.51790
10.0	9.3098	0.51790
10.0	10.294	0.51790

The columns are for data simulated by SimFIT according to $y = 0.1 + 2.0x + 0.1x^2$ and have the following meanings.

1. The first column contains the independent variable x_i in triplicate.
2. The second column contains the dependent variable y_i arising from evaluating the model equation using SimFIT program **makdat**, then adding 5% relative error using SimFIT program **adderr** to simulate experimental error.
3. The third column are the sample standard deviations s_i calculated by SimFIT program **adderr** to use for weights $w_i = 1/s_i^2$. In the absence of replicates to calculate sample standard deviations for y_i at fixed x_i, the third column could be replaced by $s_i = 1$, or simply omitted, whereupon a default value of $s_i = 1$ would be used for unweighted regression.

Program **polnom** will then proceed to fit polynomials of degree m according to

$$f(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \cdots + \theta_6 x^6$$

for $m = 0, 1, 2, \ldots, k$ where $k \leq 6$ depends on the number of distinct values of x. That is, $m = 0$ for a constant term, $m = 1$ for a straight line, $m = 2$ for a quadratic, $m = 3$ for a cubic, and so on. After fitting each degree, several statistics are output to assess goodness of fit and determine the highest degree that can be justified.

The idea of this systematic procedure is to determine if there is statistical evidence to justify a trend line or progressive curvature in noisy data, or to select a model equation to use as a calibration curve for inverse prediction. To appreciate this aspect consider the following results tables when the data are analyzed.

Table 1: Degree fitted and Chebyshev coefficients

m	A_0	A_1	A_2	A_3	A_4	A_5
0	0.31113					
1	16.034	7.9080				
2	12.737	4.8194	-1.4456			
3	12.735	4.8132	-1.4591	-0.0083774		
4	12.762	4.8342	-1.4387	-0.055083	-0.059600	
5	12.654	4.6602	-1.3858	-0.087456	-0.035275	0.22979

Another table of statistics required to determine the degree of the polynomial required is also displayed as follows.

Table 2: Statistics to determine degree of the fitted polynomial

m	σ	%change	$WSSQ$	%change	$P(\chi^2 \geq WSSQ)$	5%	FV	$P(F \geq FV)$	5%
0	36.703		22901		0.0000	no			
1	8.0833	77.98	1045.4	95.44	0.0000	no	334.50	0.0000	yes
2	0.9914	87.73	14.744	98.59	0.4700	yes	1048.6	0.0000	yes
3	1.0253	3.42	14.718	0.18	0.3977	yes	0.0249	0.8769	no
4	1.0511	2.52	14.363	2.41	0.3488	yes	0.3213	0.5805	no
5	1.0000	4.87	11.999	16.46	0.4457	yes	2.3639	0.1501	no

Here m is the degree fitted, $\sigma = \sqrt{WSSQ/NDOF}$, and FV is the F value for assessing the significance of variance reduction by adding higher degree terms.

There are many results displayed in Tables 1 and 2 in order to suggest the highest degree that can be justified statistically. The qualitative conclusions do not use a Bonferroni correction, but the actual significance levels are also provided for purists. At this point SIMFIT program **polnom** outputs the next table to aid decision.

Table 3: information to help you select a best-fit polynomial

Lowest degree where < 10% change in σ	2
Lowest degree where < 10% change in $WSSQ$	2
Lowest degree by chi-sq. at 5% significance level	2
Lowest degree by chi-sq. at 1% significance level	2
Lowest degree by F test at 5% significance level	2
Lowest degree by F test at 1% significance level	2

Accepting the recommendations of Table 3 leads to Table 4 for the best-fit quadratic.

Table 4: Results for weighted fitting $(w = 1/s^2)$

Parameter	Value	Std. error	Lower95%cl	Upper95%cl	p
θ_0	0.10347	0.0032091	0.096630	0.11031	0.0000
θ_1	2.1203	0.019731	2.0783	2.1624	0.0000
θ_2	-0.11565	0.0035714	-0.12326	-0.10803	0.0000

Correlation matrix

1		
-0.0960	1	
0.0516	-0.8432	1

If you selected to predict x from y the following warning is issued.

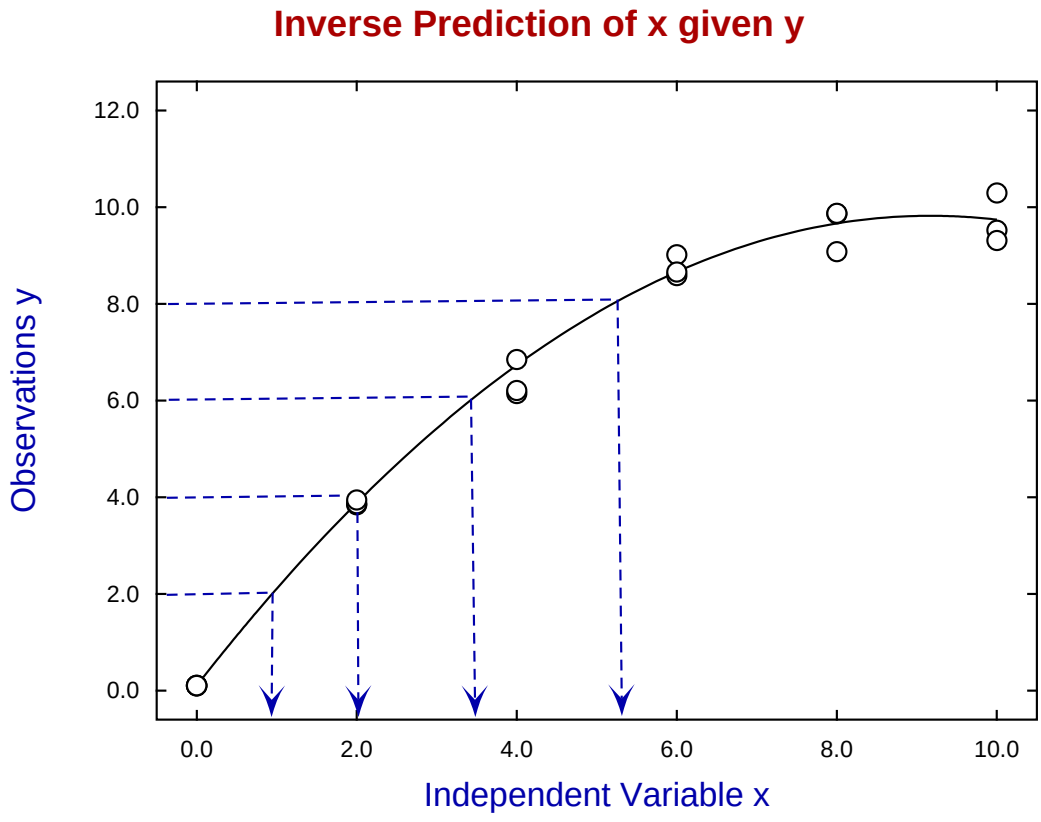
> You must be very careful if you wish to use this best-fit curve as a calibration curve for predicting x given y since there are turning points for $X_{min} \leq x \leq X_{max}$ as follows:
>
> | x-value | y-value |
> |---|---|
> | 9.1673 | 9.8224 |

This is because the quadratic has a turning point within the range of the data, and so predicting x from y could be misleading if a horizontal line for $y = y_0$ for some y_0 intersected the best fit curve twice. So you have to choose whether to search upwards or downwards along the x axis for the prediction required. If a spurious prediction results you have to change the search order. For degrees greater than two there may be multiple turning points, so using degrees greater than two is not normally recommended for inverse prediction. Table 5 results from choosing to predict x from y along with 95% confidence ranges using the data supplied in test files `polnom.tf2` and `polnom.tf3` or typed in interactively.

Table 5: using a best-fit polynomial to predict x given y
Inverse prediction data for program **polnom** : y = 2, 4, 6, 8

y-measured	x-predicted	95% confidence limits
2.0	0.9429	0.9253, 0.9612
4.0	2.0718	2.0347, 2.1100
6.0	3.4182	3.3566, 3.4819
8.0	5.1976	5.0739, 5.3342

This next graph shows the data and best-fit quadratic along with arrows indicating the prediction of x given y from Table 5. Confidence limits for the prediction are calculated by an extension of the method for unweighted linear regression to the case of weighted polynomial regression, based on the presumption that the weights are accurate, and that the y values used to predict x are exact, not means of replicate observations.



9.4.4 Using cubic spline standard curves

Cubic splines can be used to create standard curves for calibration when the data are nonlinear and more complex than a simple quadratic or cubic. Under some circumstances it is also possible to generate approximate 95% confidence limits for plotting and predicting x from y, or y from x.

From the main SIMFIT menu choose [A/Z], open program **calcurve** and scan the default test file `calcurve.tf2` containing the following data.

x	y
0.50	0.26885
0.50	0.30026
0.50	0.27048
0.50	0.28205
0.50	0.27368
1.00	0.81924
1.00	0.79264
1.00	0.80419
1.00	0.86795
1.00	0.80573
1.50	1.5215
1.50	1.6423
1.50	1.6953
1.50	1.7242
1.50	1.4159
2.00	2.5742
2.00	2.3165
2.00	2.5198
2.00	2.6637
2.00	2.5150
4.00	6.7101
4.00	6.4626
4.00	6.4935
4.00	5.9591
4.00	6.3087
6.00	7.9744
6.00	7.9506
6.00	8.2786
6.00	8.4860
6.00	8.1895
8.00	9.6610
8.00	9.6185
8.00	9.5581
8.00	9.3566
8.00	8.0443
10.0	10.713
10.0	10.619
10.0	9.7203
10.0	9.7127
10.0	9.4258

1. Column 1 contains the fixed independent variable x
2. Column 2 contains the observations y
3. The absence of a third column of weighting factors s is equivalent to a column of $s = 1$ indicating unweighted regression, but if accurate estimates for s, the sample standard deviations of replicates, are available they can be added as a third column.

Example 1

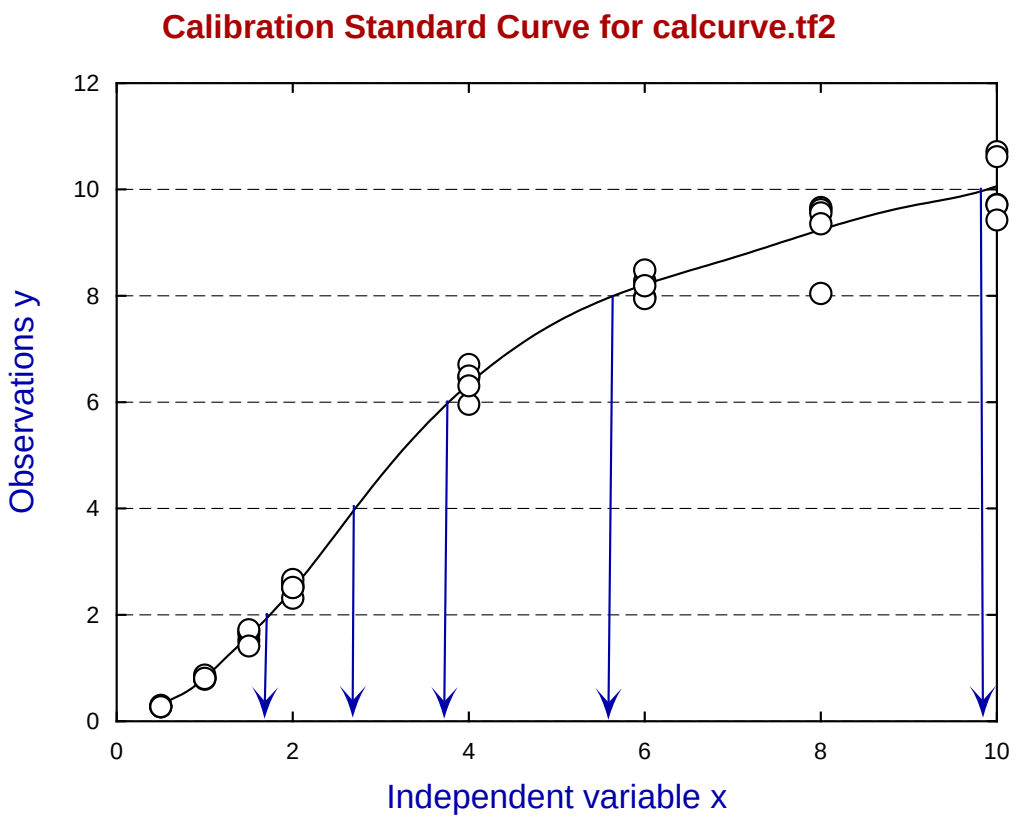
Choosing the option to create a standard calibration curve using the **calcurve** defaults then using the option to predict x given y equal to 2, 4, 6, 8, 10 leads to the following table.

y-measured	x-predicted
2.00	1.7410
4.00	2.7105
6.00	3.7716
8.00	5.6437
10.0	9.8853

To appreciate how this table results, consider the following graph where it will be clear that, for any given value of $y = y_0$, program **calcurve** solves the equation

$$y_0 - f(x) = 0$$

involving the best-fit spline function $f(x)$, using numerical techniques to locate the x-values located at the arrow heads.



Example 2

The two most often required changes to the default configurations in program **calcurve** are

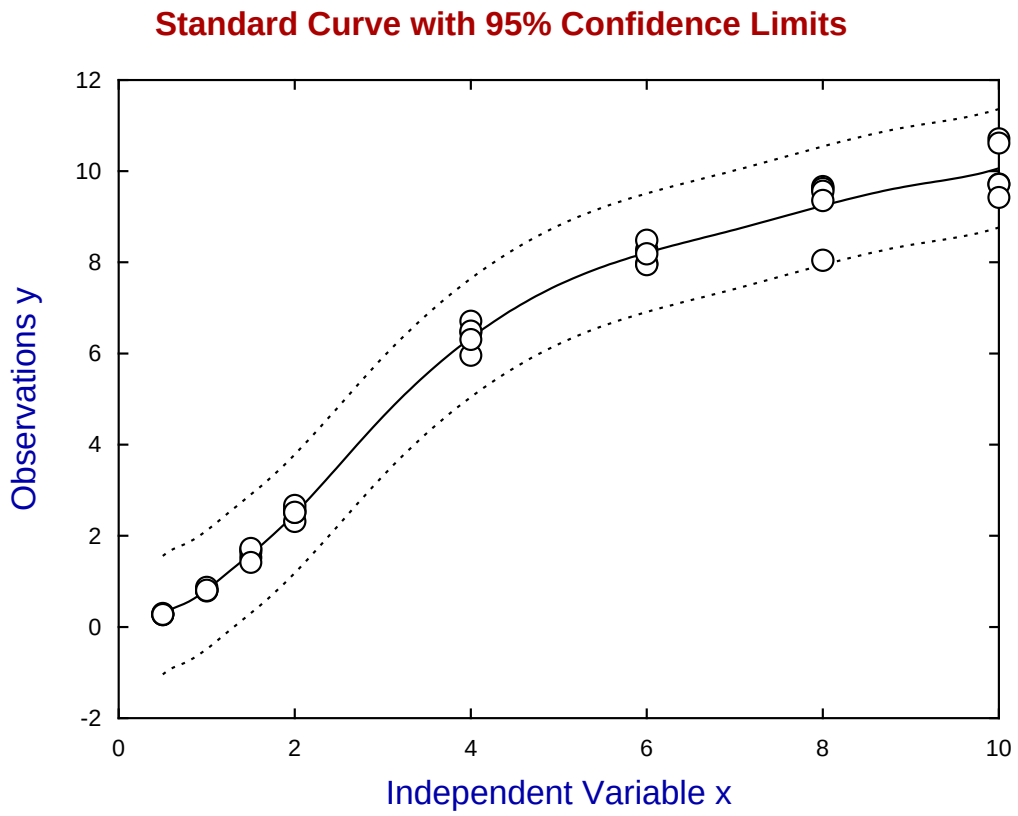
1. swap from using data-files/clipboard for standard curve data to typing in data interactively; and
2. estimate 95% confidence limits for plotting or prediction.

Typing in data interactively instead of using files or the clipboard is merely a convenience choice, but estimating confidence limits is more controversial. Unlike the situation when an appropriate deterministic equation is fitted and the best-fit parameters can be used for this purpose, the cubic spline is an empirical model which can be made to fit arbitrarily closely to data by choosing the knot placements and tension settings. Hence the weighted sum of squares at the solution point cannot be used to create a variance estimate if the default choice of unweighted fitting with cross-validation splines is employed.

Nevertheless **calcurve** will attempt to generate confidence limits in this default case leading to the following outcome when attempting to predict x from y.

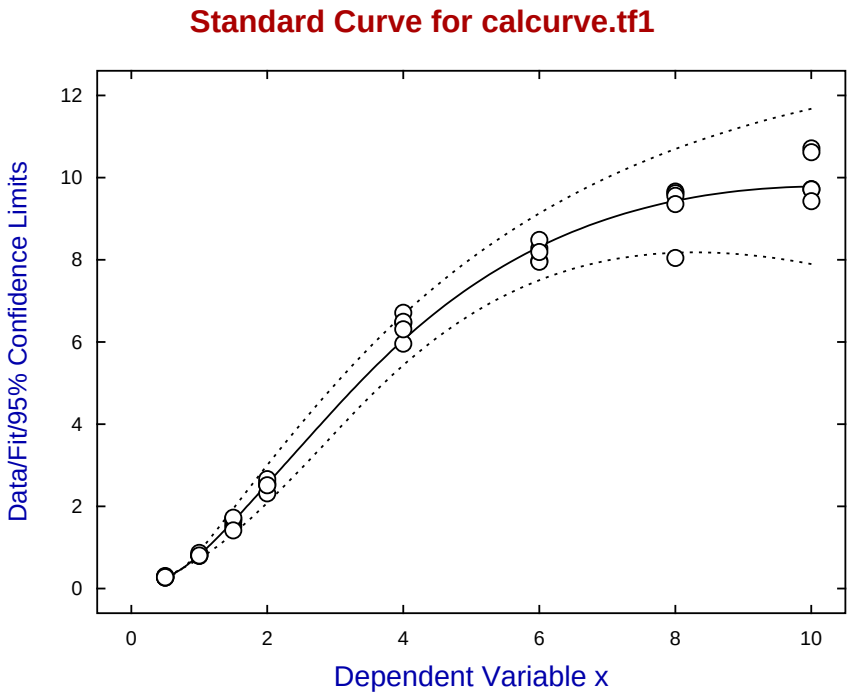
y-measured	x-predicted	Lower95%cl	Upper95%cl	
2.00	1.7410	0.78716	2.4583	
4.00	2.7105	1.9565	3.5280	
6.00	3.7716	2.8267	5.2894	
8.00	5.6437	3.8358	9.6669	
10.0	9.8853	5.5667	10.000	** Discard

The intersection of horizontal lines displayed in the previous graph is used to find the intersections with the confidence limit curves and, as will be obvious from the next graph, this fails to locate the intersection of $y = 10$ with the lower confidence limit curve. A fuller discussion of this subject will be presented later.

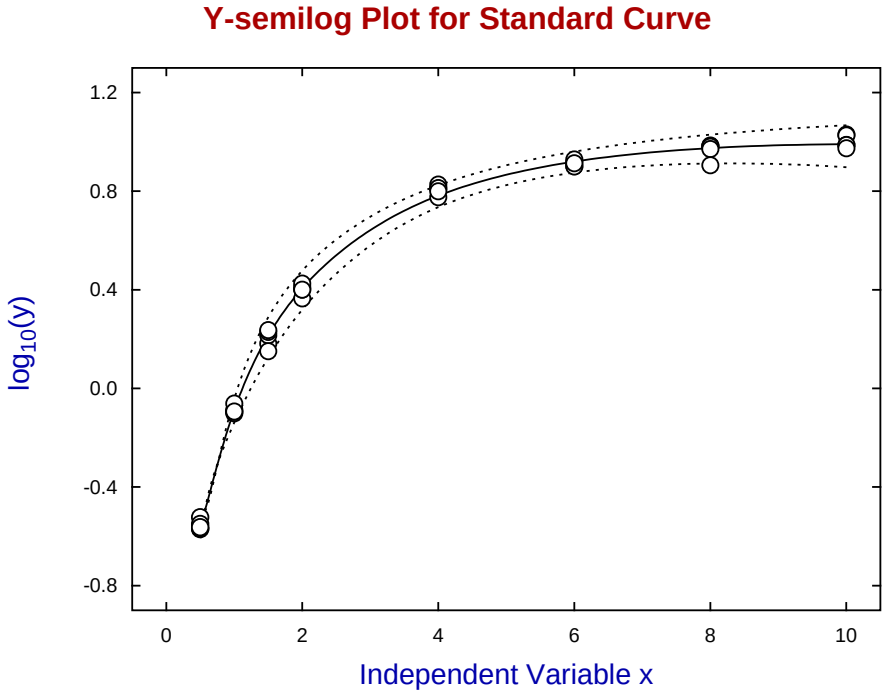


Example 3

Configuration options can be added to the data set to over-ride defaults, as illustrated by fitting `calcurve.tf1` with standard deviations from replicates added as a third column.

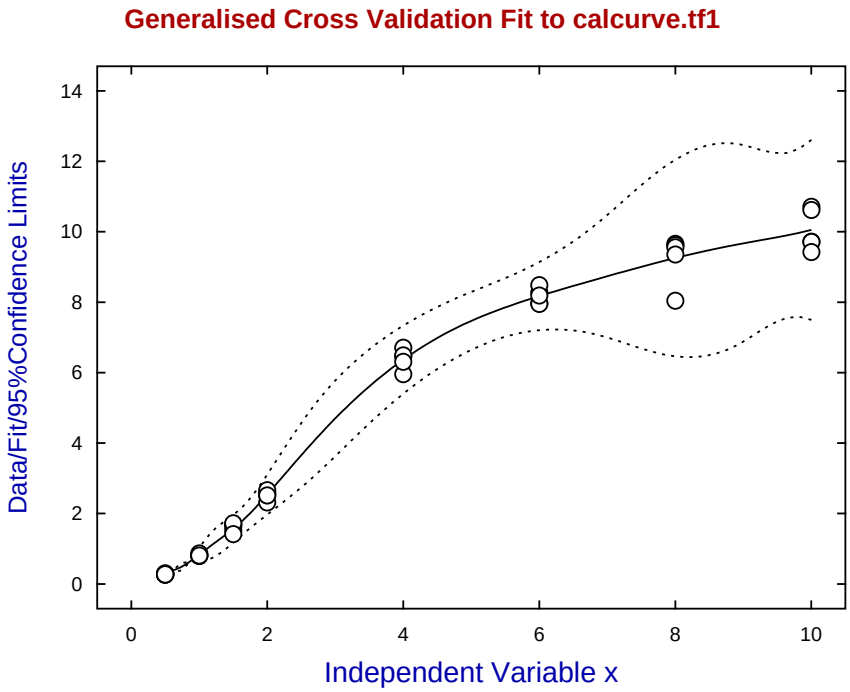


The next graph illustrates how the actual situation of constant relative error appears like constant variance under *y*-semilog transformation.



The best way to use program **calcurve** is to experiment with the configuration options until a satisfactory standard curve is obtained. The details of such a configuration can then be added to the data file to temporarily over-ride the defaults. The great advantage of this expert mode is that data sets for creating standard curves will always give rise to the same standard curve. For instance, fitting `calcurve.tf1` in expert mode reads 8 integers followed by a floating point number that informs **calcurve** to use weights $w = 1/s^2$, $\log x$ instead of x as independent variable, and weighted least squares splines instead of cross-validation splines.

To illustrate this point, the next graph shows the fit obtained for `calcurve.tf1` using the expert mode parameters except that a cross-validation spline curve was used. The previous graphs show how weighted least squares fitting smooths out noisy data, while the next graph illustrates how using too many knots, as with cross validation, can lead to excessive undulations in response to local variations in noise. A full explanation of such configuration issues will now be given.



Expert mode configuration options

It will be seen on inspection that `calcurve.tf1` has an integer after the data indicating the number of additional lines of text appended to the data. The first line appended has the following eight integers followed by a floating point number.

2, 2, 2, 1, 2, 3, 2, 2, 5.0

Program **calcurve** will recognize that this second line after the end of the data is a list of instructions indicating that the default configuration options are temporarily to be replaced by these new options as long as the data file is the current data file.

When a new data file is opened, the last set of active settings derived from the defaults will be re-instated unless the new data set has another set of such temporary re-configuration instructions.

There follows a full description of all the **calcurve** options

Configuration options

- **Option 1**

  1. Use a file (or equivalently the clipboard) for all data input
  2. Use a file (or equivalently the clipboard) for observations but type in prediction values
  3. Type in both observations and prediction values from the terminal

  The value of 3 is not valid in expert mode which must use file/clipboard input mode for observations. If sub-option 3 is selected, then data typed in will be written to a temporary file that can be saved from the SIMFIT user results folder for re-use. It is always best to use a file for input of observations, especially if the number of observations is large, and this is also true if a large number of prediction values are required

- **Option 2**

  1. Use the independent variable as supplied
  2. Transform the independent variable internally into the logarithm for fitting

  The second sub-option can only be used if the independent variable supplied is positive, and is valuable for calibration data that approaches a horizontal asymptote in order to prevent an undulating standard curve. All transformation occurs internally, and all communication with the user is in the coordinates supplied for the original observations.
  If either of these sub-options are changed interactively the standard curve will have to be re-fitted.

- **Option 3**

  1. Sparse knots ($K = N/12$)
  2. Medium knots ($K = N/6$ but $K \geq 1$)
  3. Dense knots ($K = N/3$ but $K \geq 2$)
  4. Solid knots ($K = N - 1$ but cross validation)

  Four knots are always placed at the first and last distinct points, but the number of equally spaced interior knots K depends on the number of distinct data points N. Solid knots places a knot between each distinct data point then uses generalized cross validation to estimate a smoothing factor. As explained, the type of knots used will strongly effect the shape of the standard curve. For instance, sparse knots will usually fit a simple cubic and may give rise to turning points, while solid knots may give too much undulation.
  If either of these sub-options are changed interactively the standard curve will have to be re-fitted.

- **Option 4**

  This option is linked to option 6.

  1. Weights given by $w = 1/s^2$ where s values are supplied
  2. Weights given by $w = 1/(cv\%|y|)^2$ where $cv\%$ is the percentage coefficient of variation assumed
  3. Weights given by $w = 1$ i.e., unweighted regression

  It is only sensible to supply weights s as sample standard deviations if the sample sizes are sufficiently large to be meaningful (i.e. ≥ 5), but using unweighted regression assumes constant variance which is usually incorrect. If constant relative error rather than constant variance can be established, it is best to input the estimated percentage coefficient of variation and use $w = 1/(cv\%|y|)^2$ for smoother weighting.
  If either of these sub-options are changed interactively the standard curve will have to be re-fitted.

- **Option 5**

  1. Plot y as a function of x
  2. Also plot approximate 95% confidence limits using the settings of options 4 and 6.

  Confidence limits should only be plotted if the settings of options 4 and 6 are sensible.

- **Option 6**

  This option is linked to option 4.

  1. No confidence limits on prediction
  2. Slack confidence limits on prediction (using $4s$)
  3. Medium confidence limits on prediction (using $3s$)
  4. Tight confidence limits on prediction (using $2s$)

  Confidence limits are calculated as the intersection of either y_0 or x_0 as appropriate with the upper and lower confidence limit envelopes described for option 4. They are very approximate and must be interpreted with restraint. When sub-option 3 of option 4 is selected it is not sensible to use $WSSW/NDOF$ as a variance estimated as the degrees of freedom depend on the knots and, using too many knots with no replicates can lead to $WSSQ = 0$. In this case the number of standard deviations s used to space the confidence limits above and below the best-fit standard curve are calculated using the percentage coefficient of variation assumed and the average absolute value of the observations.
  If either of these sub-options are changed interactively the standard curve will have to be re-fitted.

- **Option 7**

  Reserved for future use.

- **Option 8**

  Reserved for future use.

- **Option 9**

  If sub-options 2 or 3 of option 4 are selected the estimated percentage coefficient of variation must be provided.
  If this option is changed interactively the standard curve will have to be re-fitted.

Some examples for Expert mode settings

So, for instance, the expert line to restore defaults is

2, 1, 4, 3, 1, 1, 2, 2, 7.5

while changing to input of observations and prediction values from the terminal would require

3, 1, 4, 3, 1, 1, 2, 2, 7.5

Also changing from cross-validation splines to medium density weighted least squares would need

3, 1, 2, 3, 1, 1, 2, 2, 7.5

while a further change to invoke weighting $w = 1/s^2$, medium prediction confidence limits and addition of 95% confidence limits for graphs and predictions would result from

3, 1, 2, 1, 2, 3, 2, 2, 7.5

9.4.5 Dose-response curves, bioassay, percentiles and LD50

Dose-response curves and related bioassay techniques are frequently used to estimate percentiles such as LD50, the median concentration causing 50% mortality. When the experiment consists of estimating proportions in disjoint groups as a function of some effector, the proportions are estimates of binomial probabilities so it may be reasonable to fit a generalized linear model (GLM) that assumes binomial distribution of error.

Example 1

From the main SimFIT menu choose [A/Z] then proceed as follows.

- Open program **gcfit**
- Select the option to perform bioassay (percentiles/EC50/LD50)
- Read in the default test file `ld50.tf1` using the [Demo] button on the SimFIT file selection control

Test file `ld50.tf1` contains the following data.

y	N	x
1	10	1
4	20	2
4	10	3
5	10	4
15	30	5
7	10	6
9	10	7
12	15	8
9	10	9
8	10	10

The results were for the number of animals dying in ten separate groups after dosing with toxin as follows.

1. **Column 1**
   The number of animals dying (y_i) within a fixed time at a dose x_i
2. **Column 2**
   The number of animals in the corresponding group (N_i)
3. **Column 3**
   The concentration of toxin (x_i)

Here, for $i = 1, 2, \ldots, 10$ there is a binomial distribution with probabilities p_i and estimates $\hat{p}_i$ as follows

$$\hat{p}_i = \frac{y_i}{N_i}.$$

Further, as the groups are independent, we can calculate exact non-symmetrical confidence limits for such estimates. However, recognizing the distribution of the proportions has no further value unless it is possible to construct a model for the dependence of such proportions on the effector substance at concentration x_i.

In the event that a deterministic model cannot be constructed in the usual way it is possible to explore generalized linear models to see if they can give an adequate fit and yield meaningful estimates of the percentiles with confidence limits. For instance, here are the results from analyzing the test data using the logistic, probit, and complementary log-log models in order to estimate the 50% point. From the results it is clear that, in this case, all three models give comparable estimates for the 50% point.

Method: GLM with binomial errors, Link: Logistic
Number of samples = 10, Deviance = 4.2461

Parameter	Value	std. error	Lower95%cl	Upper95%cl	*p*
Constant	-2.0986	0.47329	-3.1900	-1.0072	0.0022
Slope	0.4507	0.08725	0.2495	0.6519	0.0009
50% point	4.6564	0.44415	3.6322	5.6806	0.0000

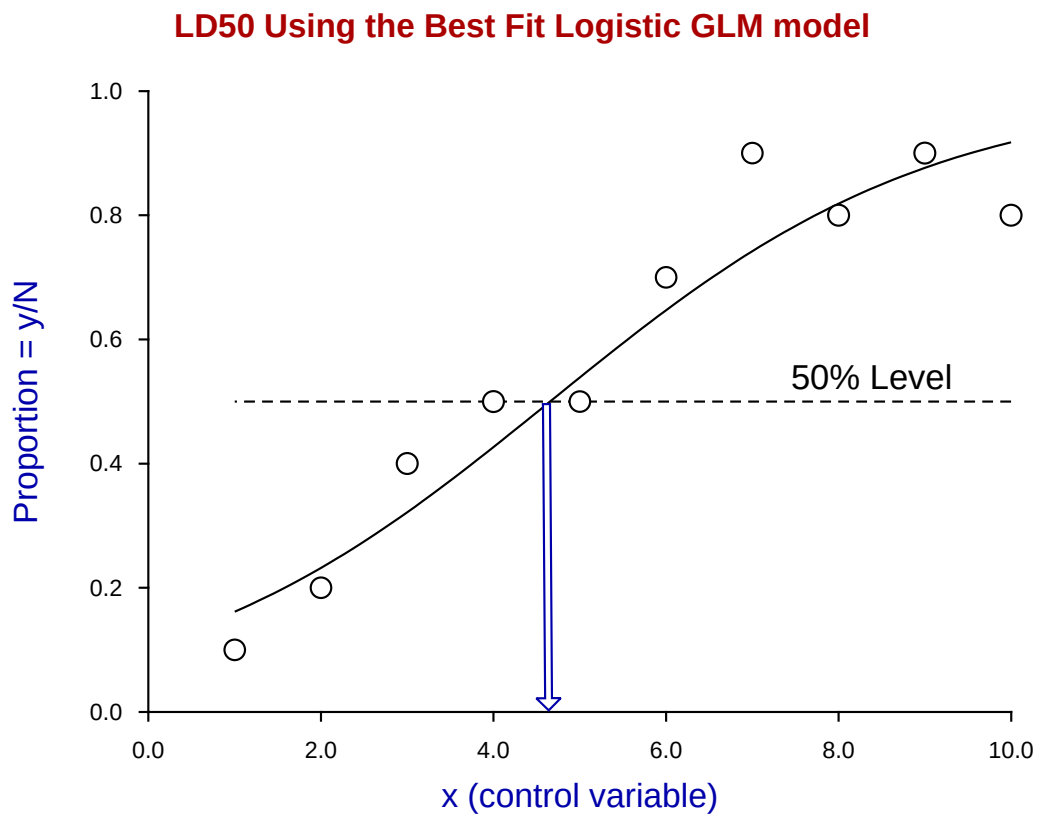
Method: GLM with binomial errors, Link: Probit
Number of samples = 10, Deviance = 4.5642

Parameter	Value	std. error	Lower95%cl	Upper95%cl	*p*
Constant	-1.2513	0.27078	-1.8757	-0.6269	0.0017
Slope	0.2668	0.04855	0.1548	0.3787	0.0006
50% point	4.6902	0.44633	3.6610	5.7194	0.0000

Method: GLM with binomial errors, Link: Complementary log-log
Number of samples = 10, Deviance = 6.6004

Parameter	Value	std. error	Lower95%cl	Upper95%cl	*p*
Constant	-1.6696	0.32951	-2.4294	-0.9097	0.0010
Slope	0.2664	0.05079	0.1492	0.3835	0.0008
50% point	4.8922	0.51820	3.6972	6.0872	0.0000

The next graph shows the best-fit curve obtained with the logistic model and indicates how the intersection of $\hat{p} = 0.5$ with the curve leads to the estimation of the 50% point. Actually other percentiles can also be estimated if required.



Example 2

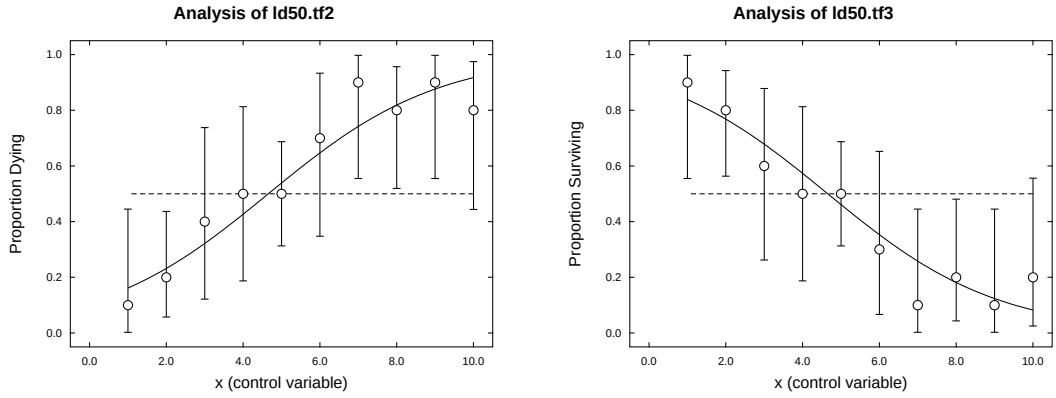
As GLM techniques are to be used it should be mentioned that the analysis just described can also be carried out with data in the standard GLM format. For instance, the column order for binomial GLM fitting is covariates first, successes, trials, then finally weighting factors. Generally, as in this case, the weighting factors would be 1, but they must be included so that the number of covariates is counted correctly.

Another point is that the same estimates for percentiles can also be carried out with the data in complementary format, where the percent surviving is estimated and not the percent dying, as long as it is remembered to reflect values other than the 50% point. One major advantage of the GLM data format is that it can be extended to include other covariates such as time, or allowing for heterogeneity in the groups.

Here, for example, are the data contained in test files `ld50.tf2` and `ld50.tf3`.

ld50.tf2				ld50.tf3			
x	y	N	s	x	N-y	N	s
1	1	10	1	1	9	10	1
2	4	20	1	2	16	20	1
3	4	10	1	3	6	10	1
4	5	10	1	4	5	10	1
5	15	30	1	5	15	30	1
6	7	10	1	6	3	10	1
7	9	10	1	7	1	10	1
8	12	15	1	8	3	15	1
9	9	10	1	9	1	10	1
10	8	10	1	10	2	10	1

The next graphs illustrate the difference between analyzing data as proportion surviving instead of proportion dying, and from these it will be clear that both data sets give the same 50% point but that any other percentiles would have to be treated carefully. For instance, 10% dying would be equivalent to 90% surviving.



Now the standard SIMFIT analysis of proportions routines only involve an indicator variable x to identify samples or to plot the variation in estimates with confidence limits as functions of x. In the estimation of LD50 we have to make further assumptions as to how the binomial parameter depends on x, not as an indicator variable but as an independent variable or, as it is usually referred to in generalized linear models, a covariate. Such models are not based on biochemical theories as to how a toxin acts, but are empirical models that are only useful to the extent that they fit the dose-response curve adequately.

Theory

Given N trials at fixed x with probability p for success in each trial then the probability of y successes is

$$P(y) = \binom{N}{y} p^y (1-p)^{N-y}, \text{ for } y = 0, 1, \ldots, N$$

and the best estimate for p is

$$\hat{p} = \frac{y}{N}$$

where an unsymmetrical $100(1-\alpha)\%$ confidence range (p_l, p_u) may be obtained by solving the nonlinear equations

$$\sum_{t=y}^{N} \binom{N}{t} p_l^t (1-p_l)^{N-t} = \alpha/2$$

$$\sum_{t=0}^{y} \binom{N}{t} p_u^t (1-p_u)^{N-t} = \alpha/2$$

for p_l and p_u.

The generalized linear model for binomial errors supposes that the expectation of Y is to be estimated, i.e.,

$$E(Y) = \mu.$$

given the distribution

$$f_Y = \binom{N}{y} p^y (1-p)^{N-y}$$

and the assumption that a predictor function η exists, which is a linear function of the m covariates, i.e., independent explanatory variables, as in

$$\eta = \sum_{j=1}^{m} \beta_j x_j.$$

Finally, it is assumed that a link function $g(\mu)$ exists between the expected value of Y and the linear predictor. The choices for

$$g(\mu) = \eta$$

with the binomial distribution, where y successes have been observed in N trials, are the logistic, probit, or complementary log-log link functions

$$\text{logistic: } \eta = \log\left(\frac{\mu}{N-\mu}\right)$$

$$\text{probit: } \eta = \Phi^{-1}\left(\frac{\mu}{N}\right)$$

$$\text{complementary log-log: } \eta = \log\left(-\log\left(1-\frac{\mu}{N}\right)\right).$$

9.5 Time series

Tutorials and worked examples for simulation, curve fitting, statistical analysis, and plotting.
https://simfit.uk
https://simfit.org.uk
https://simfit.silverfrost.com

9.5.1 Running medians, moving averages and the Tukey-Hanning 4253H twice smoother

Given a sequence of observations at equal increments of time or space, etc. a smoothed curve can be fitted in several alternative ways using running medians and/or moving averages. From the main SIMFIT menu choose [Statistics], [Time series], [Data smoothing], then study the test file provided which is `g10caf.tf1`.

569
416
422
565
484
520
573
518
501
505
468
382
310
334
359
372
439
446
349
395
461
511
583
590
620
578
534
631
600
438
516
534
467
457
392
467
500
493
410
412
416
403
422
459
467

512
534
552
545

These are measurements of coal production in millions of tons per year in the USA from 1920 to 1968, and SIMFIT provides the following smoothing options.

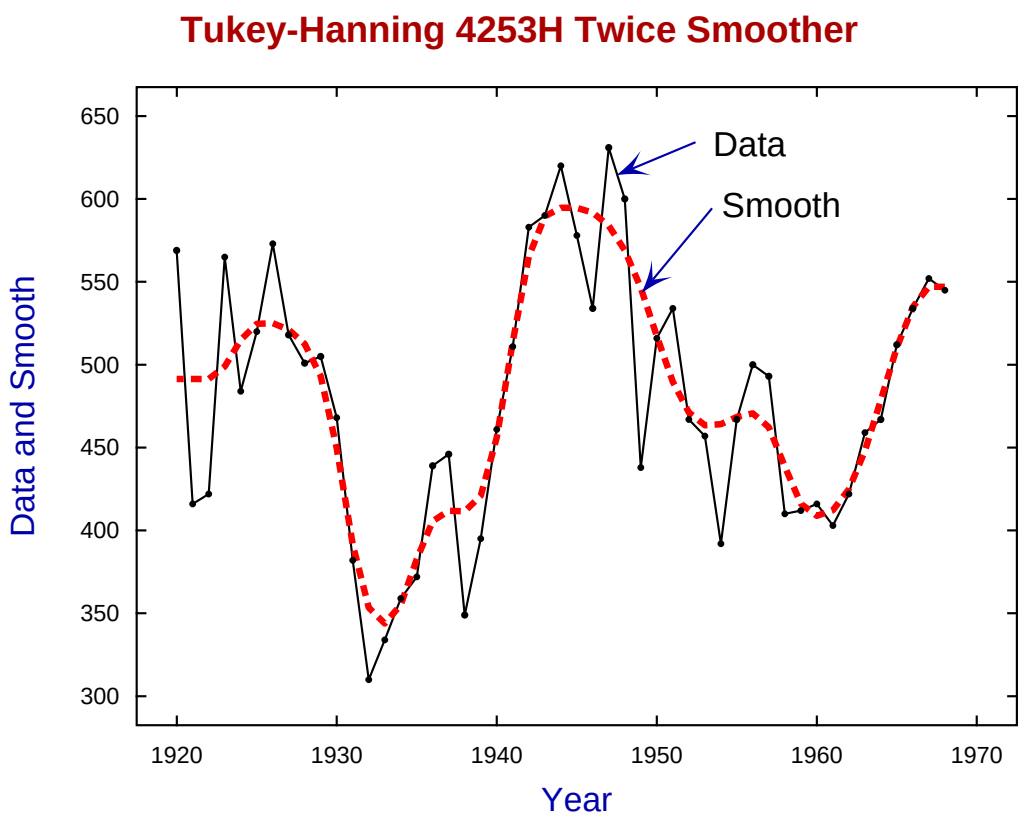
1. Running medians with span 4 then 2
2. Running medians with span 5
3. Running medians with span 3
4. Moving averages with span 3 using Hanning
5. The Tukey-Hanning 4253H twice smoother

The result of smoothing is to create a smoothed fit together with residuals, known as rough, that is

Data = Smooth + Rough

and the usual options are available to analyze and plot the smoothed fit and residuals.

The first four options may be investigated, especially with sparse data, but for most purposes the Tukey-Hanning 4253H twice smoother would be preferred as illustrated in the next graph.



Theory

1. **Running medians of even span (e.g. 4 followed by 2)**

   These are applied in pairs as there are complications due to lack of symmetry. For instance, a span of four generates n + 1 smooths from a sample size n as follows.

$$\begin{aligned} z_1 &= y_1 \\ z_2 &= (y_1 + y_2)/2 \\ z_3 &= \text{median}(y_1, y_2, y_3, y_4) \\ &\dots \\ z_{n-1} &= (y_{n-2} + y_{n-1})/2 \\ z_n &= (y_{n-1} + y_n)/2 \\ z_{n+1} &= y_n \end{aligned}$$

   Following this by a span of two restores the dimension of smooths to n in the following way.

$$\begin{aligned} z_1* &= (z_1 + z_2)/2 \\ z_2* &= (z_2 + z_3)/2 \\ &\dots \\ z_{n-1}* &= (z_{n-1} + z_{n-2})/2 \\ z_n* &= (z_n + z_{n+1})/2 \end{aligned}$$

2. **Running medians of odd span**

   For instance a span of five proceeds as follows

$$\begin{aligned} z_1 &= y_1 \\ z_2 &= \text{median}(y_1, y_2, y_3) \\ z_3 &= \text{median}(y_1, y_2, y_3, y_4, y_5) \\ &\dots \\ z_n &= y_n \end{aligned}$$

   while a span of three is simply

$$\begin{aligned} z_1 &= y_1 \\ z_2 &= \text{median}(y_1, y_2, y_3) \\ z_3 &= \text{median}(y_2, y_3, y_4) \\ &\dots \\ z_n &= y_n \end{aligned}$$

3. **Moving averages**

   This usually involves a weighting scheme as with the Hanning of span three with binomial weights.

$$\begin{aligned} z_1 &= y_1 \\ z_2 &= 0.25y_1 + 0.5y_2 + 0.25y_3 \\ &\dots \\ z_n &= y_n \end{aligned}$$

4. **Endpoint rules**

   To correct for singular behavior at the extreme points SIMFIT uses the correction

$$z_1 = \text{median}(3z_2 - 2z_3, z_1, z_2)$$
$$z_n = \text{median}(3z_{n-2} - 2z_{n-1}, z_n, z_{n-1})$$

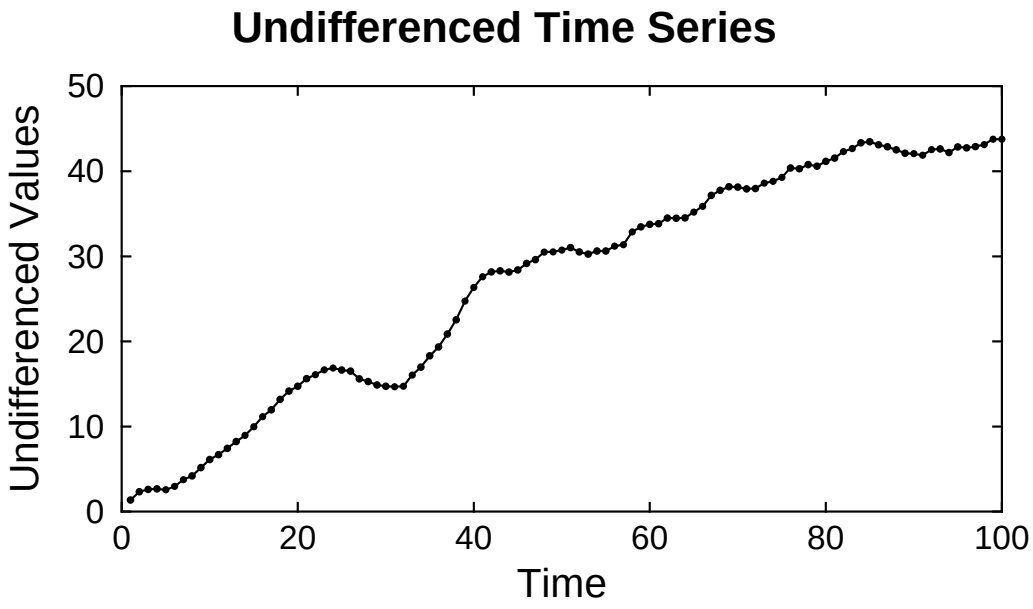
5. **The 4253H twice smoother**

   The previous simple smoothers may be of some use with sparse data, but the most widely recommended smoother is the Tukey-Hanning 4253H twice smoother. This applies running medians of four, then two, then five, then 3, followed by a Hanning filter of span 3. The rough (i.e. residuals) are then calculated but are then also subjected to the same sequence before being added to the smooth from the first pass, followed by calculating new rough.

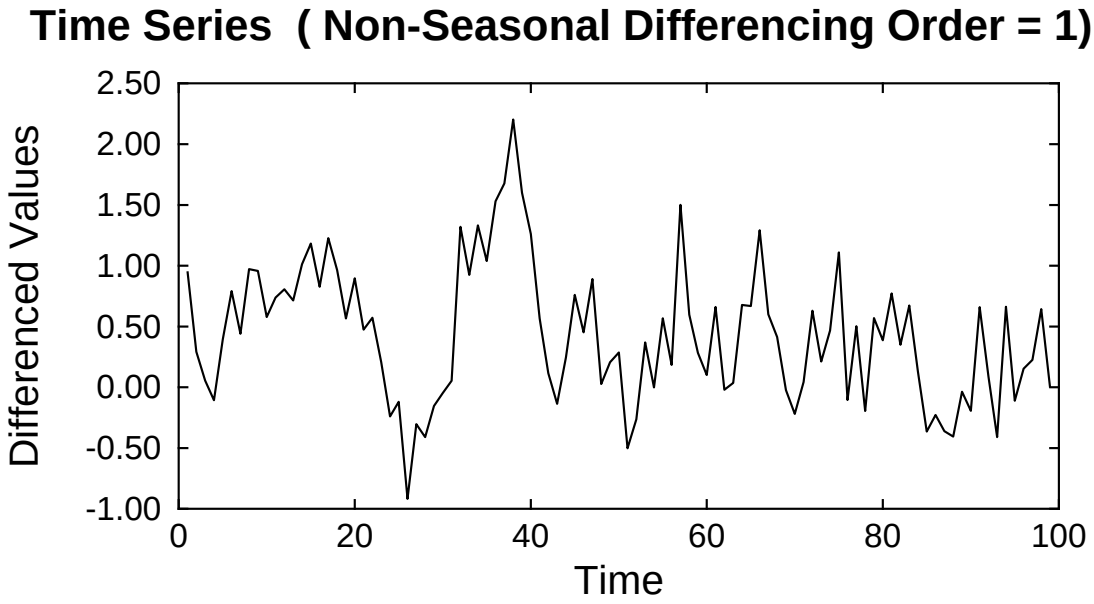
9.5.2 Lags, auto-correlations, and partial auto-correlation functions

It is usual to analyze a time series for auto-correlations before proceeding to ARIMA.

From the main SimFIT menus choose [Statistics], [Time series] then [Lags and auto-correlations], analyze the test file `times.tf1`, then plot the undifferenced time series to obtain the following graph.



As there is clearly an increasing trend with these data it would be sensible to try a non-seasonal differencing of order 1 to remove a linear trend leading to the following differenced series.



The next table presents the results from an analysis with such a non-seasonable differencing of one and calculating for ten lags.

Current data title is: Test file times.tf1: time series data (J06SBF)		
Original dimension (NX)	100	
After differencing (NXD)	99	
Non-seasonal order (ND)	1	
Seasonal order (NDS)	0	
Seasonality (NS)	0	
Number of lags (NK)	10	
Number of PACF (NVL)	10	
X-mean (differenced)	0.42833	
X-variance (differenced)	0.31521	
Statistic (S)	83.1349	
$P(\chi^2 \geq S)$	0.0000	*Reject H_0 at 1% significance level*

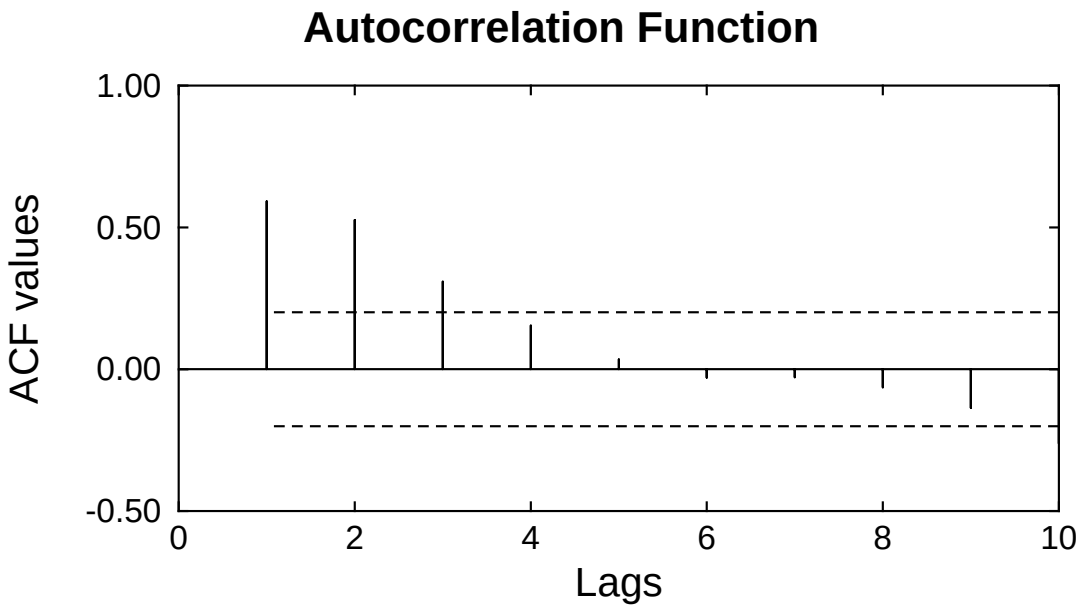
Lag	R	PACF	VR	ARP
1	0.5917	0.591740	0.649844	0.391570
2	0.5258	0.270263	0.602378	0.398778
3	0.3087	-0.129867	0.592219	0.001602
4	0.1536	-0.143970	0.579944	-0.143950
5	0.0345	-0.054313	0.578233	-0.136536
6	-0.0297	0.011048	0.578162	-0.045279
7	-0.0284	0.071092	0.575240	0.147351
8	-0.0642	-0.044918	0.574080	0.130623
9	-0.1366	-0.175865	0.556324	-0.067066
10	-0.2619	-0.249824	0.521603	-0.249824

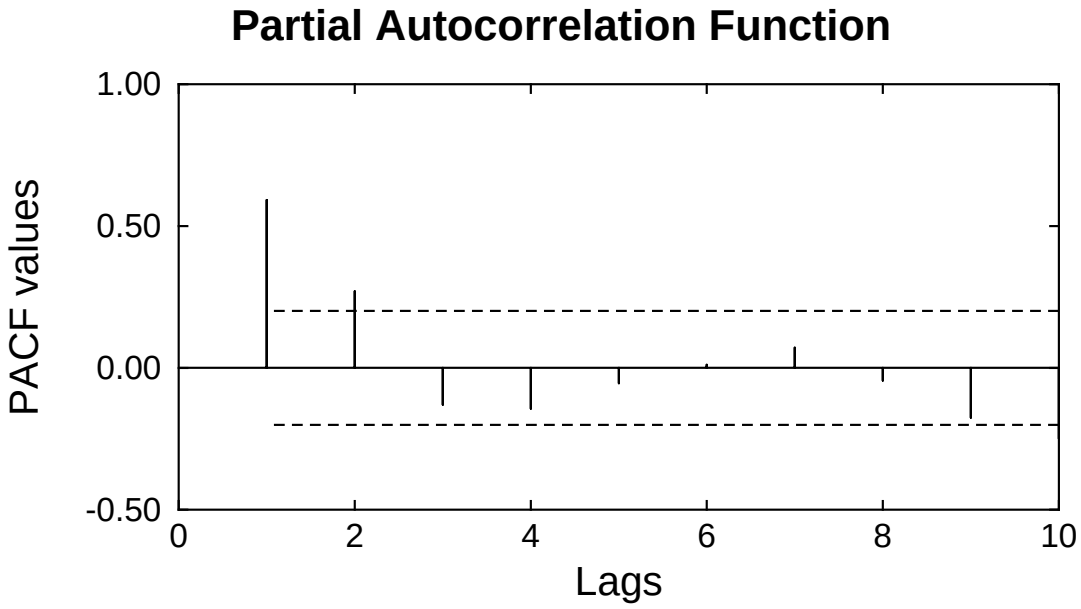
The abbreviations used in this table are defined as follows.

Abbreviation	Meaning
X	the sample (i.e. vector of data with no missing values)
NX	dimension of X
XD	transformed vector derived from X by differencing
NXD	dimension of XD (i.e. NXD = NX - ND - NS*NDS)
ND	order of non-seasonal differencing
NDS	order of seasonal differencing
NS	seasonality of seasonal differencing
NK	number of lags requested
NL	number of partial correlations requested
NVL	number of valid partial correlations
R	calculated auto-correlation coefficients
PACF	calculated partial auto-correlation coefficients
VR	calculated predictor error variance ratios
ARP	calculated auto-regressive parameters of maximum order
S	statistic for testing H_0: all autocorrelations zero
Note	if ND = NDS = NS = 0, the original sample X is analyzed

From these results the χ^2 test clearly indicates that not all correlations are zero, so it is useful to proceed to comparing the strengths of autocorrelations and partial autocorrelations as functions of the lags.

Note that, in the next figures, the approximate 95% confidence limits are estimated as $2/\sqrt{n}$, where n is the sample size after differencing (if any).





These figures illustrate what is summarized in the previous table, that only the correlations with lags of one and two are highly significant.

Theory

A time series is a vector $x(t)$ of $n > 1$ observations x_i obtained at a sequence of points t_i, e.g., times, distances, etc., at fixed intervals Δ, i.e.

$$\Delta = t_{i+1} - t_i, \text{ for } i = 1, 2, \ldots, n-1,$$

and it is assumed that there is some seasonal variation, or other type of autocorrelation to be estimated.

A linear trend can be removed by first order differencing

$$\nabla x_t = x_t - x_{t-1},$$

while seasonal patterns of seasonality s can be eliminated by first order seasonal differencing

$$\nabla_s x_t = x_t - x_{t-s}.$$

Note that differencing of orders $d = ND$, $D = NDS$, and seasonality $s = NS$ may be applied repeatedly to a series so that

$$w_t = \nabla^d \nabla_s^D x_t$$

will be shorter, of length $NXD = n - d - D \times s$, and will extend for $t = 1 + d + D \times s, \ldots, NX$.

Non-seasonal differencing up to order d is calculated sequentially using

$$\begin{array}{lcll} \nabla^1 x_i & = & x_{i+1} - x_i & \text{for } i = 1, 2, \ldots, n-1 \\ \nabla^2 x_i & = & \nabla^1 x_{i+1} - \nabla^1 x_i & \text{for } i = 1, 2, \ldots, n-2 \\ \ldots & & & \\ \nabla^d x_i & = & \nabla^{d-1} x_{i+1} - \nabla^{d-1} x_i & \text{for } i = 1, 2, \ldots, n-d \end{array}$$

while seasonal differencing up to order D is calculated by the sequence

$$\begin{array}{lcll} \nabla^d \nabla_s^1 x_i & = & \nabla^d x_{i+s} - \nabla^d x_i & \text{for } i = 1, 2, \ldots, n-d-s \\ \nabla^d \nabla_s^2 x_i & = & \nabla^d \nabla_s^1 x_{i+s} - \nabla^d \nabla_s^1 x_i & \text{for } i = 1, 2, \ldots \; n-d-2s \\ \ldots & & & \\ \nabla^d \nabla_s^D x_i & = & \nabla^d \nabla_s^{D+1} x_{i+s} - \nabla^d \nabla_s^{D+1} x_i & \text{for } i = 1, 2, \ldots, n-d-D \times s. \end{array}$$

Note that, as indicated in the previous tables, either the original sample X of length NX, or a differenced series XD of length NXD, can be analyzed interactively, by simply adjusting ND, NDS, or NS. Also the maximum number of autocorrelations $NK < NXD$ and maximum number of partial autocorrelations $L \leq NK$, can be controlled, although the maximum number of valid partial autocorrelations NVL may turn out to be less than L. Now, defining either $x = X$, and $n = NX$, or else $x = XD$ and $n = NXD$ as appropriate, and using $K = NK$, the mean and variance are recorded, plus the autocorrelation function R, comprising the autocorrelation coefficients of lag k according to

$$r_k = \sum_{i=1}^{n-k} (x_i - \bar{x})(x_{i+k} - \bar{x}) \Big/ \sum_{i=1}^{n} (x_i - \bar{x})^2.$$

If n is large and much larger than K, then the S statistic

$$S = n \sum_{k=1}^{K} r_k^2$$

has a chi-square distribution with K degrees of freedom under the hypothesis of zero autocorrelation, and so it can be used to test that all correlations are zero.

The partial autocorrelation function $PACF$ has coefficients at lag k corresponding to $p_{k,k}$ in the autoregression

$$x_t = c_k + p_{k,1}x_{t-1} + p_{k,2}x_{t-2} + \cdots + p_{k,l}x_{t-k} + e_{k,t}$$

where $e_{k,t}$ is the predictor error, and the $p_{k,k}$ estimate the correlation between x_t and x_{t+k} conditional upon the intermediate values $x_{t+1}, x_{t+2}, \ldots, x_{t+k-1}$. Note that the parameters change as k increases, and so $k = 1$ is used for $p_{1,1}$, $k = 2$ is used for $p_{2,2}$, and so on.

These parameters are determined from the Yule-Walker equations

$$r_i = p_{k,1}r_{i-1} + p_{k,2}r_{i-2} + \cdots + p_{k,k}r_{i-k},\ i = 1, 2, \ldots, k$$

where $r_j = r_{|j|}$ when $j < 0$, and $r_0 = 1$.

An iterative technique is used and it may not always be possible to solve for all the partial autocorrelations requested. This is because the predictor error variance ratios VR are defined as

$$\begin{aligned} v_k &= Var(e_{k,t})/Var(x_t) \\ &= 1 - p_{k,1}r_1 - p_{k,2}r_2 - \cdots - p_{k,k}r_k, \end{aligned}$$

unless $|p_{k,k}| \geq 1$ is encountered at some $k = L_0$, when the iteration terminates, with $NVL = L_0 - 1$.

The Autoregressive parameters of maximum order ARP are the final parameters $p_{L,j}$ for $j = 1, 2, \ldots, NVL$ where NVL is the number of valid partial autocorrelation values, and L is the maximum number of partial autocorrelation coefficients requested, or else $L = L_0 - 1$ as before in the event of premature termination of the algorithm.

9.5.3 Auto-correlation and cross-correlation matrices

This technique is used when there are two corresponding time series, or in fact any series of signals recorded at a sequence of fixed discrete intervals of time or space etc., and a comparison of the two series is required.

From the main SIMFIT menu choose [Statistics], [Time series], then [Auto- and cross-correlation matrices] and examine the test file `g13dmf.tf1` provided which contains the following data.

X	*Y*
-1.490	7.340
-1.620	6.350
5.200	6.960
6.230	8.540
6.210	6.620
5.860	4.970
4.090	4.550
3.180	4.810
2.620	4.750
1.490	4.760
1.170	10.880
0.850	10.010
-0.350	11.620
0.240	10.360
2.440	6.400
2.580	6.240
2.040	7.930
0.400	4.040
2.260	3.730
3.340	5.600
5.090	5.350
5.000	6.810
4.780	8.270
4.110	7.680
3.450	6.650
1.650	6.080
1.290	10.250
4.090	9.140
6.320	17.750
7.500	13.300
3.890	9.630
1.580	6.800
5.210	4.080
5.250	5.060
4.930	4.940
7.380	6.650
5.870	7.940
5.810	10.760
9.680	11.890
9.070	5.850
7.290	9.010
7.840	7.500
7.550	10.020
7.320	10.380
7.970	8.150
7.760	8.370
7.000	10.730
8.350	12.140

Column 1 contains time series X and column 2 contains the corresponding time series Y. As multivariate time series with more than two variables can be difficult to analyze it is necessary to select any two variables for pairwise analysis using this technique.

The next table illustrates the results from analyzing test file `g13dmf.tf1` for the first ten lags using the SIMFIT cross-correlation matrices time series options.

Auto- and cross-correlation matrices

Sample size $n = 48$
Approximate standard deviation = 0.1443
Mean of $X = 4.37021$
Mean of $Y = 7.86750$
For lag $m = 0$: sample X, Y Correlation coefficient $r = 0.2493$

m = 1	0.7366(***)	0.1743
	0.2114	0.5541(***)
m = 2	0.4562(**)	0.0764
	0.0693	0.2602
m = 3	0.3795(**)	0.0138
	0.0260	-0.0381
m = 4	0.3227(*)	0.1100
	0.0933	-0.2357
m = 5	0.3414(*)	0.2694
	0.0872	-0.2499
m = 6	0.3634(*)	0.3436(*)
	0.1323	-0.2263
m = 7	0.2802	0.4254(**)
	0.2069	-0.1283
m = 8	0.2482	0.5217(***)
	0.1970	-0.0845
m = 9	0.2400	0.2664
	0.2537	0.0745
m = 10	0.1621	-0.0197
	0.2667	0.0047

Indicators: $p < 0.005$(***), $p < 0.01$(**), $p < 0.05$(*)
Maximum off-diagonal, $m = 8$, $|C(1,2)| = 0.5217$

In the above two by two matrices r_{ij} the positions have the following meanings at the lags indicated.

$r(1,1)$: auto-correlation for X
$r(2,2)$: auto-correlation for Y
$r(1,2)$: cross-correlation for X with Y (lags in Y)
$r(2,1)$: cross-correlation for Y with X (lags in X)

The significance levels are indicated if $p \leq 0.05$. These indicate significant auto-correlation for X at lags 1 to 6 but for Y only at lag 1, and also significant cross-correlation for r_{12} at lags 6 to 8.

Theory

It is assumed that the data are from a multivariate time series or similar set of observations of several variables at fixed intervals, and it is wished to make pairwise analysis of such observations.

The data must be supplied as two vectors, say X and Y of length n for instance, with X as column 1 of a n by 2 matrix, and Y as column 2.

The routine first calculates the sample means $\bar{x}$ and $\bar{y}$, the sample variances V_x and V_y, and sample correlation coefficient r. Then, for a selected number of lags $m = 1, 2, \ldots, k$, the auto-correlations and cross-correlations are output as a sequence of 2 by 2 matrices.

Since $1/\sqrt{n}$ is a rough approximation to the standard errors of these estimates, the approximate significance for the sample cross-correlations is indicated as in the table using the following labeling scheme.

$$|r(i,j)| > 3.29/\sqrt{n} : ***$$
$$|r(i,j)| > 2.58/\sqrt{n} : **$$
$$|r(i,j)| > 1.96/\sqrt{n} : *.$$

Finally, the off-diagonal i.e., cross-correlation, coefficient with largest absolute value is indicated. If this value is close to unity it indicates that the series are closely similar, and the value of m at which this occurs indicates the extent to which the series have to be slid past each other to obtain maximum similarity of profiles. Usually, the largest value of m selected for analysis would be for $k \leq n/4$.

Defining the denominator D as follows

$$D = \sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

then the auto-correlations $r(1,1)$ and $r(2,2)$, and the cross-correlations $r(1,2)$ and $r(2,1)$ as functions of m are given by

$$r(1,1) = \frac{1}{D}\sum_{i=1}^{n-m}(x_i - \bar{x})(x_{i+m} - \bar{x})$$
$$r(1,2) = \frac{1}{D}\sum_{i=1}^{n-m}(x_i - \bar{x})(y_{i+m} - \bar{y})$$
$$r(2,1) = \frac{1}{D}\sum_{i=1}^{n-m}(x_{i+m} - \bar{x})(y_i - \bar{y})$$
$$r(2,2) = \frac{1}{D}\sum_{i=1}^{n-m}(y_i - \bar{y})(y_{i+m} - \bar{y})$$

for $m = 1, 2, \ldots, k$.

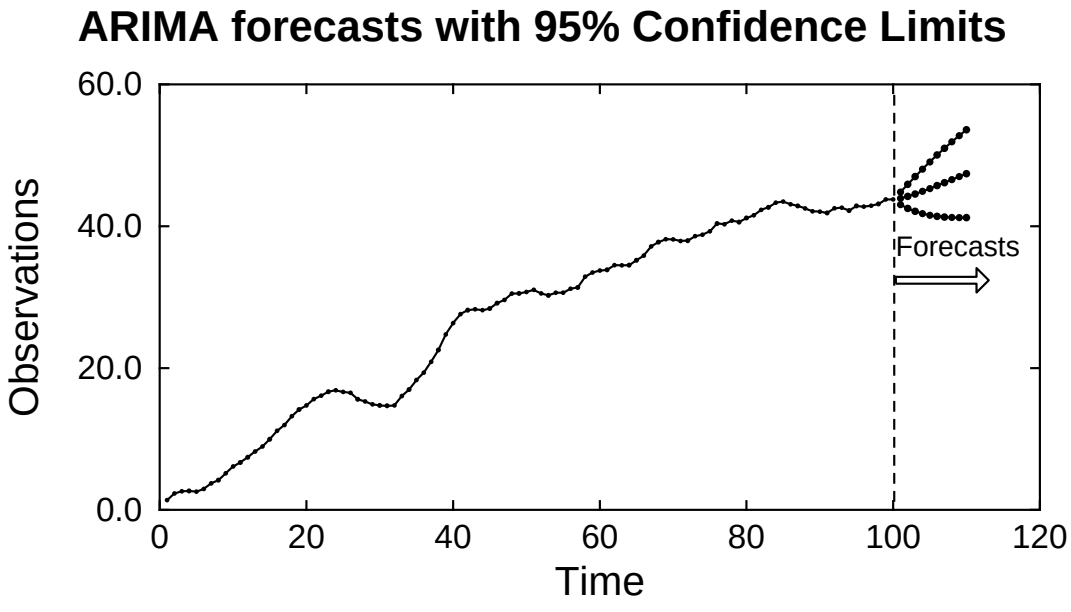
9.5.4 ARIMA with forecasts

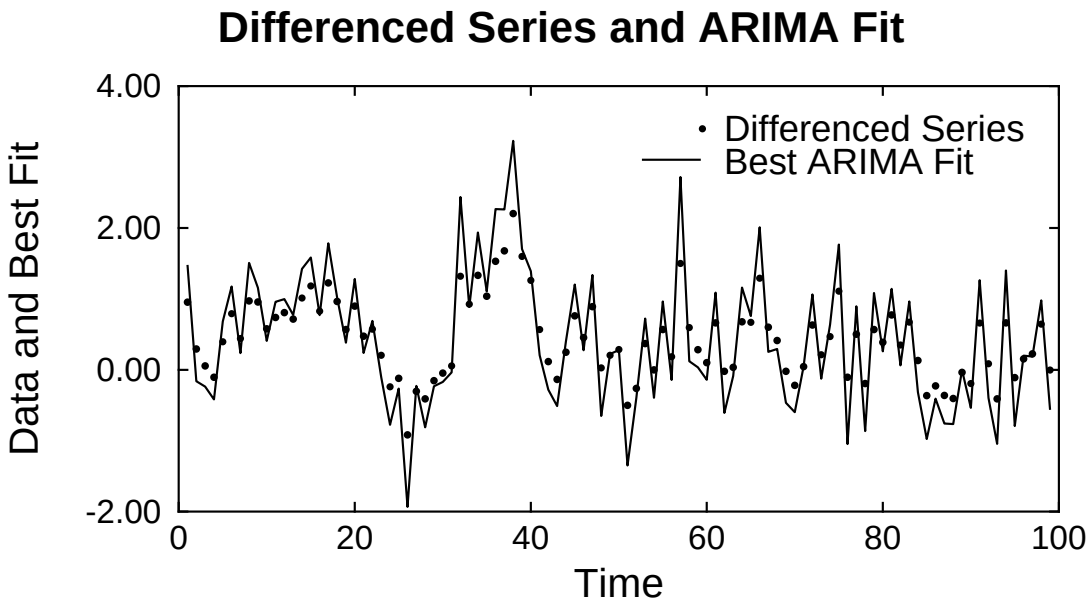
When a time series has been analyzed for non-seasonal and seasonal differencing it is possible to fit an autoregressive integrated moving average model (ARIMA) and obtain forecasts for future values.

From the main SIMFIT menu choose [Statistics], [Time series], then the [ARIMA with forecasts] option and fit the test file `time.tf1` using the default settings to obtain the following results, and plots.

ARIMA with forecasts	
Current data title is:	
Test file times.tf1: time series data (J06SBF)	
Original dimension (NX)	100
After differencing (NXD)	99
Non-seasonal order (ND)	1
Seasonal order (NDS)	0
Seasonality (NS)	0
Number of forecasts (NF)	3
Number of parameters (NP)	1
Number of iterations (ITC)	2
Sum of squares (SSQ)	19.9092

Parameter	Value	Std. err.	Type
$\phi(1)$	0.60081	0.08215	Autoregressive
$C(0)$	0.42960	0.11239	Constant term
prediction(1)	43.9351	0.45305	Forecast
prediction(2)	44.2082	0.85512	Forecast
prediction(3)	44.5437	1.23335	Forecast





Theory

It must be stressed that fitting an ARIMA model is a very specialized iterative technique that does not yield unique solutions. So, before using this procedure, you must have a definite idea, by using the auto-correlation and partial auto-correlation options or by knowing the special features of the data, exactly what differencing scheme to adopt and which parameters to fit. Users can select the way that starting estimates are estimated, they can monitor the optimization, and they can alter the tolerances controlling the convergence, but only expert users should alter the default settings.

It is assumed that the time series data $x_1, x_2, \ldots, x_n$ follow an ARIMA model so that a differenced series given by

$$w_t = \nabla^d \nabla_s^D x_i - c$$

can be fitted, where c is a constant, d is the order of non-seasonal differencing, D is the order of seasonal differencing and s is the seasonality. The method estimates the expected value c of the differenced series in terms of an uncorrelated series a_t and an intermediate series e_t using parameters $\phi, \theta, \Phi, \Theta$ as follows. The seasonal structure is described by

$$w_t = \Phi_1 w_{t-s} + \Phi_2 w_{t-2\times s} + \cdots + \Phi_P w_{t-P\times s} + e_t - \Theta_1 e_{t-s} - \Theta_2 e_{t-2\times s} - \cdots - \Theta_Q e_{t-Q\times s}$$

while the non-seasonal structure is assumed to be

$$e_t = \phi_1 e_{t-1} + \phi_2 e_{t-2} + \cdots + \phi_p e_{t-p} + a_t - \theta_1 a_{t-1} - \theta_2 a_{t-2} - \cdots - \theta_q a_{t-q}.$$

The model parameters $\phi_1, \phi_2, \ldots, \phi_p, \theta_1, \theta_2, \ldots, \theta_q$ and $\Phi_1, \Phi_2, \ldots, \Phi_P, \Theta_1\Theta_2, \ldots, \Theta_Q$ are estimated by nonlinear optimization, the success of which is heavily dependent on choosing an appropriate differencing scheme, starting estimates and convergence criteria. After fitting an ARIMA model, forecasts can be estimated along with 95% confidence limits.

10 Simulation

Tutorials and worked examples for simulation, curve fitting, statistical analysis, and plotting.
https://simfit.uk
https://simfit.org.uk
https://simfit.silverfrost.com

10.1 Introduction

Simulation is used to predict the outcome of experiments where a deterministic mathematical model has been formulated, so that the effect of varying parameters and independent variables can be investigated. Once a profile for exact data has been generated it can be perturbed to resemble various types of uncertainty, for instance that resulting from stochastic processes or experimental error.

SIMFIT provides the following procedures.

- Program **random**

  This allows the creation of random matrices or vectors from specified distributions and also provides facilities for scrambling a vector, generating random Latin squares for use in experimental design, and testing the Uniform(0,1) generator. As computers cannot calculate genuine random numbers, they first accumulate pseudo-random numbers from a Uniform(0,1) distribution using techniques that have a large cycle and minimal autocorrelation. Alternative distributions can then be simulated from such Uniform(0,1) numbers, and the ability to test the generator is provided so users can appreciate sample sizes required (often hundreds) before the histograms generated resemble the simulated distributions.

- Program **makdat**

  This program allows the creation of exact data sets in just two cases.

  1. The model is available from one of the SIMFIT library of models
  2. The model is supplied from a user-defined script

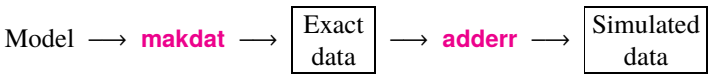
  Users simply supply the model parameters and the range, size, and number of data points to be simulated. The simulated data sets are then written to coordinate files.

- Program **deqsol**

  This is used if the deterministic model is set of nonlinear differential equations.

- Program **adderr**

  This accepts an exact data set then adds random noise to simulate uncertainty or experimental error as summarized below.



- Program **makcsa**

  This creates random histograms, for example, to simulate flow cytometry investigation of cell surface antigens.

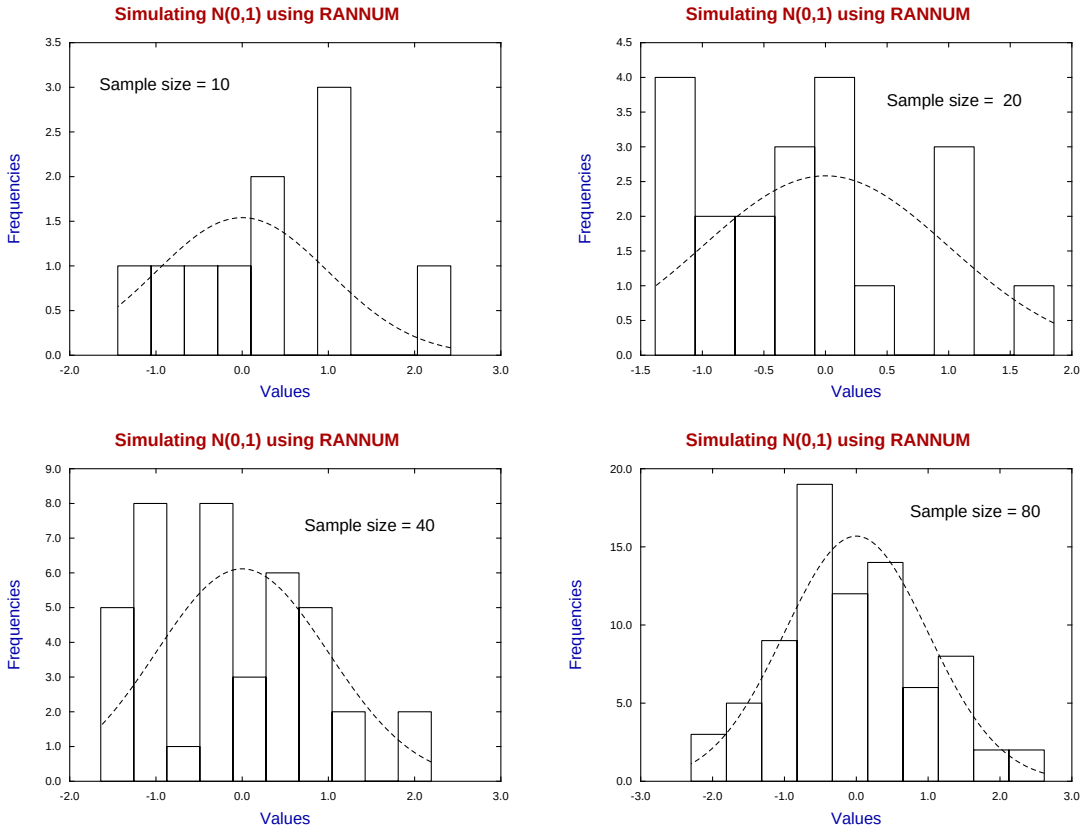
Simulation should be used where a model has been fitted to some data and best-fit parameters have been estimated, because then the model can be simulated in order to get some idea of how to interpret the goodness of fit statistics, and also what reliance to place on the point estimates for the parameter values. This is because the procedures used to fit nonlinear models are iterative, and the objective function solution point will depend on the starting estimates and may not be unique. Further, the statistical techniques used to calculate parameter standard error estimates and make decisions about goodness of fit are only approximations, and usually only close to the assumed distributions with very large numbers of observations.

10.2 Simulation: random numbers

Tutorials and worked examples for simulation, curve fitting, statistical analysis, and plotting.
https://simfit.uk
https://simfit.org.uk
https://simfit.silverfrost.com

10.2.1 Generating a random vector

A sample of pseudo–random numbers can be generated using program **rannum** after selecting the distribution, parameters required, and sample size. For instance, selecting [Simulate] from the main SIMFIT menu, then choosing to simulate a normal distribution with $\mu = 0$ and $\sigma^2 = 1$ for sample sizes of 10, 20, 40, and 80 created four vector files. These were then analyzed by program **normal** leading to the histograms shown below.



In these histograms we can see that a sample size of ten generates a sample that bears little comparison to the theoretical distribution shown as a dotted curve. Even a sample size of twenty does not look convincingly like the theoretical distribution and a sample size of eighty is required before the histogram does really suggest the theoretical distribution. Of course this is merely a warning not to the use the shape of a histogram to suggest the parent distribution unless the sample size is sufficiently large, and in any case the shape of histograms depend on the number of bins used and are therefore to a degree arbitrary. Of course program **normal** provides many graphical and statistical techniques to determine if a sample can be regarded as coming from a normal distribution.

There follows a list of distributions available using program **rannum**.

Distributions and their density/mass functions

Cauchy: parameters A and B, where $B > 0$

$$f(x) = \frac{1}{B\pi[1 + ((x - A)/B)^2]}$$

Chi-square: N degrees of freedom, where $N > 0, x > 0$

$$f(x) = \frac{x^{N/2-1} \exp(-x/2)}{2^{N/2}(N/2 - 1)!}$$

Negative exponential: mean A, where $A > 0, x > 0$

$$f(x) = (1/A) \exp(-x/A)$$

Gamma: parameters A and B, where $A > 0, B > 0, x > 0$

$$f(x) = \frac{x^{A-1} \exp(-x/B)}{B^A \Gamma(A)}$$

Logistic: mean A, spread B, where $B > 0$

$$f(x) = \frac{\exp[(x - A)/B]}{B(1 + \exp[(x - A)/B])^2}$$

Lognormal: parameters A and B, where $B > 0, x > 0$

$$f(x) = \frac{1}{Bx\sqrt{2\pi}} \exp -\frac{1}{2}\left(\frac{\log(x) - A}{B}\right)^2$$

Normal: mean A, standard deviation B, where $B > 0$

$$f(x) = \frac{1}{B\sqrt{2\pi}} \exp -\frac{1}{2}\left(\frac{x - A}{B}\right)^2$$

Uniform: real parameters A and B, where $B > A$

$$f(x) = \frac{1}{B - A}$$

Weibull: parameters A and B, where $A > 0, B > 0, x > 0$

$$f(x) = \frac{Ax^{A-1}}{B} \exp -\left(\frac{x^A}{B}\right)$$

F: M (numerator), N (denominator) degrees of freedom, where $x > 0$

$$f(x) = \frac{\Gamma((M + N)/2)(M/N)^{M/2} x^{M/2-1}}{\Gamma(M/2)\Gamma(N/2)[1 + (M/N)x]^{(M+N)/2}}$$

t: N degrees of freedom

$$f(x) = \frac{\Gamma((N + 1)/2)}{\Gamma(N/2)\sqrt{N\pi}} \left(1 + \frac{x^2}{N}\right)^{-(N+1)/2}$$

Binomial: parameters N and p, where $N \geq 1, 0 \leq p \leq 1$

$$f(x) = \binom{N}{x} p^x (1 - p)^{N-x}$$

Poisson: mean T, where $T > 0$

$$f(x) = \frac{T^x \exp(-T)}{x!}$$

Uniform: integer parameters M and N, where $N > M$

$$f(x) = \frac{1}{N - M + 1}$$

10.2.2 Generating a random matrix

A data matrix of pseudo–random numbers can be generated using program **rannum** after selecting the distribution, parameters required, number of rows, and number of columns.

This can be very useful when exploring the variation to be expected in typical experiments. So there are default choices for the distributions of the columns (or rows by transposition using **editmt**) but SIMFIT also allows user to simulate random matrices with a very large choice for the column distributions by joining m random columns of length n together using program **editmt** to form a n by m matrix.

Example 1: Simulating analysis of proportions

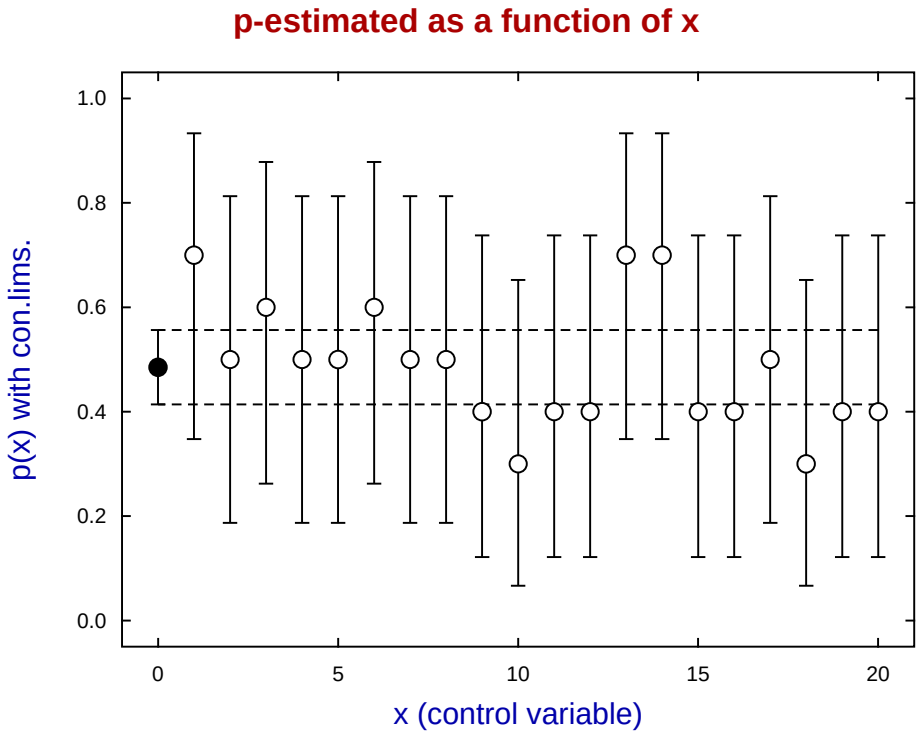
For instance, from the main SIMFIT menu, choosing [Simulate] then running program **rannum** was used to simulate a binomial distribution with $N = 10$ and $p = 0.5$. The resulting random matrix was written to a data file that was then read into program **simstat** for analysis of proportions leading to the following results.

y	N	lower 95%	$\hat{p}$	upper 95%
7	10	0.34755	0.70000	0.93326
5	10	0.18709	0.50000	0.81291
6	10	0.26238	0.60000	0.87845
5	10	0.18709	0.50000	0.81291
5	10	0.18709	0.50000	0.81291
6	10	0.26238	0.60000	0.87845
5	10	0.18709	0.50000	0.81291
5	10	0.18709	0.50000	0.81291
4	10	0.12155	0.40000	0.73762
3	10	0.06674	0.30000	0.65245
4	10	0.12155	0.40000	0.73762
4	10	0.12155	0.40000	0.73762
7	10	0.34755	0.70000	0.93326
7	10	0.34755	0.70000	0.93326
4	10	0.12155	0.40000	0.73762
4	10	0.12155	0.40000	0.73762
5	10	0.18709	0.50000	0.81291
3	10	0.06674	0.30000	0.65245
4	10	0.12155	0.40000	0.73762
4	10	0.12155	0.40000	0.73762

Successes and number of trials for binomial $N, p = 10, 0.5000$

To test H_0: equal binomial p-values	
Sample-size (rows)	20
Overall sum of Y	97
Overall sum of N	200
Overall estimate of p	0.4850
Lower 95% confidence limit	0.4139
Upper 95% confidence limit	0.5565
$-2\log\lambda$ $(-2LL)$	11.67, $NDOF = 19$
$P(\chi_2 \geq -2LL)$	0.8991
χ^2 test statistic (C)	11.43, $NDOF = 19$
$P(\chi^2 \geq C)$	0.9085

The next figure illustrates the wide 95% confidence limits for the p estimates for the individual samples of size 10 (open circles) compared to the overall estimate for the pooled sample (filled circle).



Example 2: simulating analysis of variance

Program **rannum** provides several options for random matrices, but it is possible to control the distributions for the columns by generating sample vectors, i.e. columns, then joining them up to form a matrix. For instance, the distribution of errors in many experiments are more like a Cauchy than a normal distribution. So three columns of Cauchy random numbers with $A = 0, B = 1$ were generated then joined together by program **editmt** to form this 10 by 3 data matrix, and from the results it seems that ANOVA was not sensitive to the outliers (indicated by *) in this particular simulation.

0.7553767	0.0114847	0.1967812
*15.782768	-10.974171	0.0389082
0.1525358	0.0614315	-0.6627294
0.3088897	-0.4970650	-2.0314559
-0.4559783	0.2725820	-0.7291483
2.8106498	*8.2014348	0.9221295
-0.9132832	-1.3773182	-1.5086666
-0.7499996	1.5989911	-1.5862460
-1.1157865	-0.8385944	-3.8728595
1.3796620	-0.6206551	*23.120279

1-Way Analysis of Variance: Grand Mean 9.227E-01
Transformation:- x (untransformed data)

Source	*SSQ*	*NDOF*	*MSQ*	*F*	*p*
Between Groups	27.72	2	13.86	0.3885	0.6818
Residual	963.1	27	35.67		
Total	990.8	29			

Kruskal–Wallis Nonparametric One Way Analysis of Variance

Test statistic	*NDOF*	*p*
2.114	2	0.3476

10.2.3 Generating a randomly shuffled list

In experimental design it is often required to take a list of numbers, names, or similar, and jumble them up to generate a randomly shuffled list.

From the main SIMFIT menu select [A/Z], open program **rannum** and choose to permute a list. As all lists of n items can be regarded as in one to one correspondence to the successive integers $1, 2, \ldots, n$, all that is required is a technique to select one of the $n!$ possible lists, where every permutation is equally likely.

Here is the starting set followed by such a set of ten shuffled lists, each with ten items but note that, for $n \leq 26$, the corresponding alphabetical characters are also displayed as here, which some may find convenient.

Starting list

1	2	3	4	5	6	7	8	9	10
A	B	C	D	E	F	G	H	I	J

10 out of the 10! possible shuffled lists

7	4	6	1	5	3	9	10	2	8
G	D	F	A	E	C	I	J	B	H
6	3	10	7	1	5	4	8	2	9
F	C	J	G	A	E	D	H	B	I
8	2	3	7	9	1	6	5	10	4
H	B	C	G	I	A	F	E	J	D
9	10	6	8	4	5	2	3	7	1
I	J	F	H	D	E	B	C	G	A
9	6	3	4	2	1	7	5	8	10
I	F	C	D	B	A	G	E	H	J
7	6	2	5	3	1	9	8	4	10
G	F	B	E	C	A	I	H	D	J
4	6	9	3	10	8	2	1	5	7
D	F	I	C	J	H	B	A	E	G
6	8	3	7	10	9	2	5	1	4
F	H	C	G	J	I	B	E	A	D
2	7	1	10	9	5	4	8	6	3
B	G	A	J	I	E	D	H	F	C
10	9	5	8	2	1	7	6	3	4
J	I	E	H	B	A	G	F	C	D

After scrambling such a list, a chosen permutation can be copied to the clipboard or written to a SIMFIT type data file for retrospective use, with alphabetical equivalents appended for $n \leq 26$. Note that program **rannum** will generate a new permutation at every operation unless the option to input a fixed seed to regenerate an identical list is chosen, and the fixed seed is input at the start of the permutations.

10.2.4 Generating a random Latin square

In experimental design, especially with Latin square ANOVA, it is often required to use a randomly generated Latin square.

To make this clear, consider the SIMFIT test file `ANOVA3.TF1` (shown in full and colored blue below) by choosing [View] from the SIMFIT main menu and browsing the test files.

```
Latin Square ANOVA Data ... see NAG routine G04ADF
   10    5
    5    4    1    3    2
    2    5    4    1    3
    3    2    5    4    1
    1    3    2    5    4
    4    1    3    2    5
 6.67 7.15 8.29 8.95 9.62
 5.40 4.77 5.40 7.54 6.93
 7.32 8.53 8.50 9.99 9.68
 4.92 5.00 7.29 7.85 7.08
 4.88 6.16 7.83 5.38 8.51
    7
```

Line 1	title for this data set
Line 2	number of rows then number of columns
Line 3	first row of keys (using 1,2,3,... instead of A,B,C...)
Line 7	last row of keys (using 1,2,3,... instead of A,B,C...)
Line 8	first row of data values(corresponding to first row of keys)
Line 12	last row of data values(corresponding to last row of keys)
Line 13	number of rows of comments appended to data set

First of all note that there are two ways to represent Latin squares.

1. Using integers as in

$$\begin{matrix} 1 & 2 & 3 \\ 2 & 3 & 1 \\ 3 & 1 & 2. \end{matrix}$$

2. Using characters as in

$$\begin{matrix} A & B & C \\ B & C & A \\ C & A & B. \end{matrix}$$

To create a SIMFIT data file for n by n Latin square ANOVA you first generate a n by n Latin square and save it to file. Then take the corresponding n by n data file, and copy and paste the Latin square (in integer format) into the data just before the first line of data. Finally you must change the dimension header from $n \quad n$ into $2n \quad n$. For example, in the above test file the original line number two would have had $5 \quad 5$ but, after pasting in the 5 by 5 Latin square, it would contain $10 \quad 5$ to account for the doubling of the number of rows of data.

Theory

A Latin square is a n by n array of integers with the following properties.

1. Each row must contain the complete set of positive integers $1, 2, \ldots, n$ in some order.
2. Each column must contain the complete set of positive integers $1, 2, \ldots, n$ in some order.
3. No two rows can be identical and no two columns can be identical.

Sometimes characters or symbols are used, for instance alphabetical characters if $n \leq 26$ as in the following 4 by 4 standard cases.

A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D
B	A	D	C	B	C	D	A	B	D	A	C	B	A	D	C
C	D	B	A	C	D	A	B	C	A	D	C	C	D	A	B
D	C	A	B	D	A	B	C	D	C	B	A	D	C	B	A

The way that SIMFIT generates a random Latin square is to start by generating the n by n formulation

$$\begin{pmatrix} 1 & 2 & 3 & 4 & \ldots & n \\ n & 1 & 2 & 3 & \ldots & n-1 \\ n-1 & n & 1 & \ldots & \ldots & n-2 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 2 & 3 & 4 & \ldots & \ldots & 1 \end{pmatrix}$$

with the indicated shifting. Then the rows and columns are shuffled and the option is provide to view or save the randomly generated Latin square. For $n \leq 26$ the alphabetical representation is also generated and appended to any examples that are written to files.

Rather surprisingly, there is no simple way to calculate the total number of permutations given n, but the following table will give some idea of the numbers involved.

n	Number
3	12
4	576
5	161250
6	812851200
7	61479419904000

10.2.5 Generating a random walk

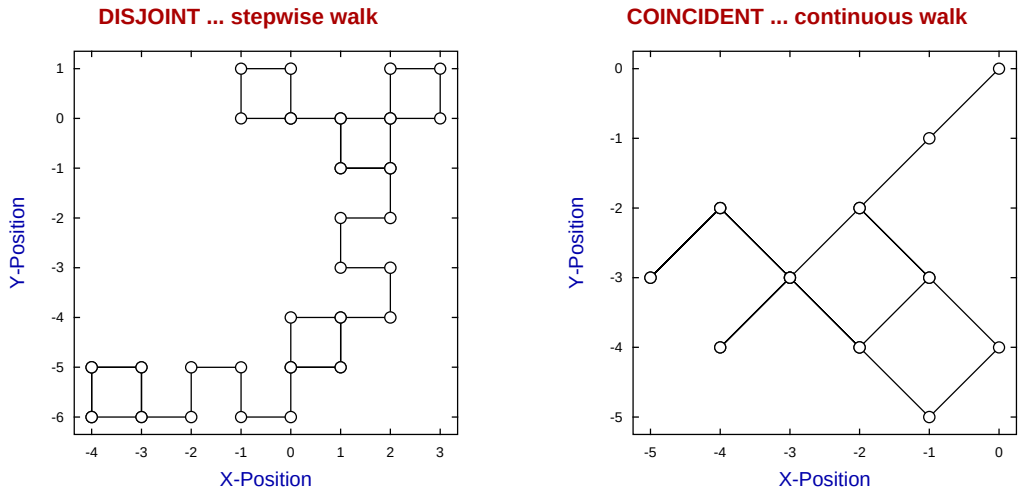
Random walks can be used to model numerous natural processes such as diffusion, or the migration of motile cells in a permeable gel matrix

The SIMFIT program **rannum** provides numerous ways to generate random walks with the following options.

1. There can be 1, 2, or 3 dimensions starting from user-defined initial coordinates.
2. The distributions used for the variables can be chosen independently for each dimension.
3. Simulated walks can be displayed, tabulated, or written to files.
4. The walks can be disjoint where each coordinate is varied in sequence, or coincident when all variables change at the same step.

Example 1: Disjoint or coincident walks

To illustrate the distinction between disjoint and coincident walks consider the next two examples.



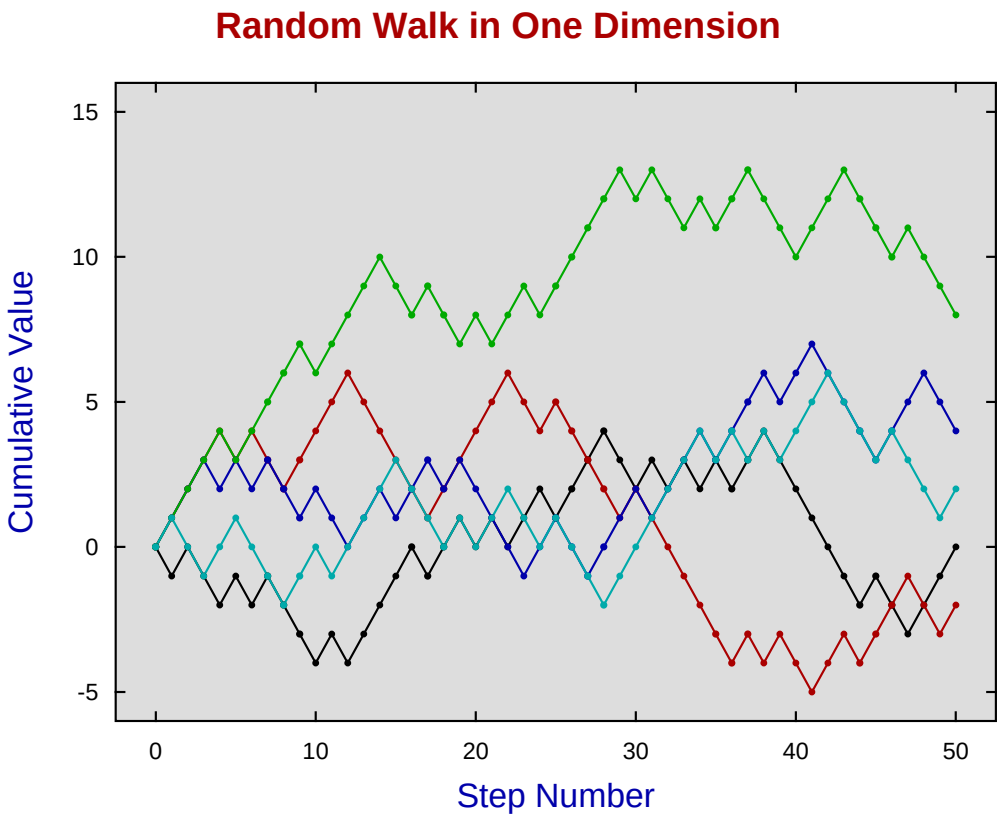
The distribution chosen for the two variables X and Y was a random integer with just two possible values, that is

$$P(X = -1) = p$$
$$P(X = +1) = 1 - p$$
$$P(Y = -1) = p$$
$$P(Y = +1) = 1 - p$$
$$p = \frac{1}{2}.$$

In the disjoint mode the walk takes place first along the X coordinate then along the Y coordinate, so that all movements can clearly be seen to be either horizontal or vertical, whereas in the coincident mode the walk on the grid occurs in diagonal jumps from node to node. Tables and data written to file are in the more economical coincident mode.

Example 2: Plotting collections of walks

It is often useful to plot several walks starting from the same position as it is often surprising how such repeated walks diverge. The examples below illustrate a common phenomenon where walks may tend to persist in the same direction even with equiprobable positive and negative steps, and which can appear somewhat contrary to intuition. The examples illustrated all had the probability of a step length of 1 equal to the probability of a step length of -1.



There are two ways to create such plots depending on the technique used to archive the coordinates.

- Using the [Advanced] option
  When it is a wished to archive the coordinates of a walk then the [Advanced] button should be used to provide the option to save the coordinates to a file. After the file has been saved you will have the additional option to store the file in your graphics project archive. This is simply a very convenient technique to store graphics files for subsequent re-plotting. There are then three ways to create a graph with the collected walks using program **simfit**.

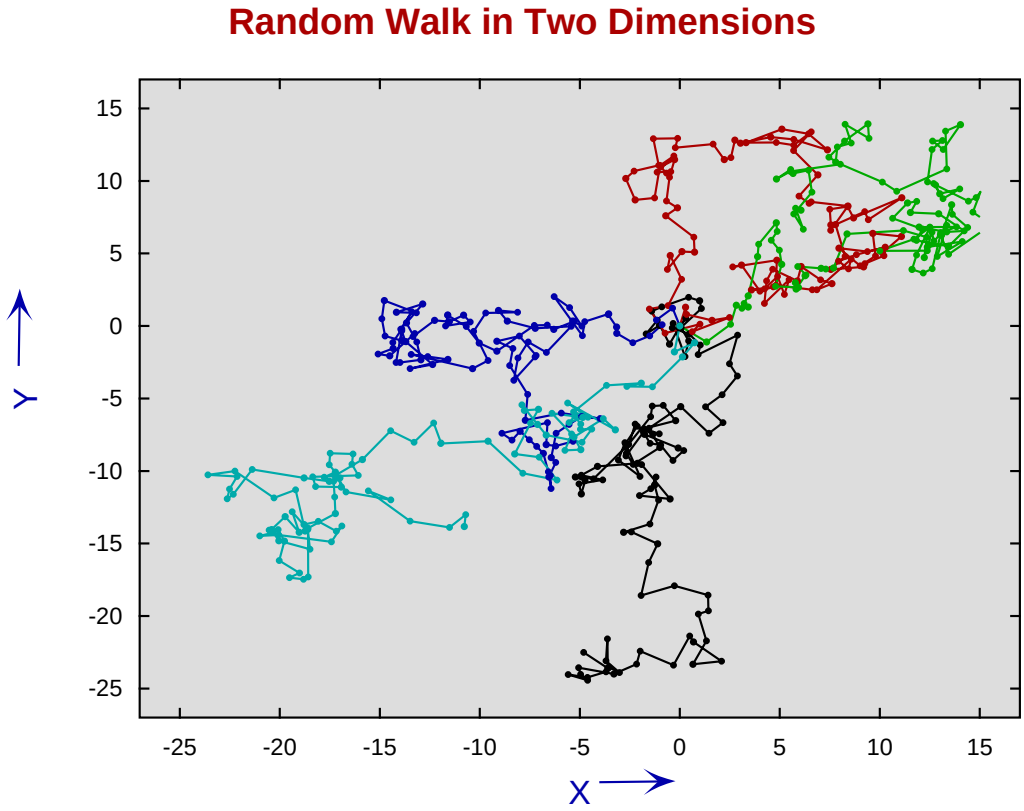
  1. Select the files individually.
  2. Select the required files as a set from a library file. created by program **maklib**.
  3. Select the required files by multiple file selection from your project archive.

- Using coordinates written to a file
  Files of random walks saved from program **rannum** can be used directly for retrospective plotting as, with the special case of just one dimension, the step number is added as a first column to the file so it can be used as the X coordinate.

Example 3: Plotting two dimensional random walks

The previous examples used steps of 1 or -1 with equal probability but other distributions are often much more useful to represent natural processes.

For example, the next example illustrates five random walks all starting at $x = 0$ and $y = 0$ to illustrate the large variation in walks possible where the steps were all taken from a standard normal distribution.



In random walks where equally probable steps of -1 or 1 are not used then the walk is no longer taking place between the integer intersection points on rectangular grids, instead jumps can be taken between arbitrary coordinates as allowed by the distributions selected. In the previous graph the distribution used for the steps x_i, y_i has density function

$$f(t) = \frac{1}{\sigma\sqrt{2\pi}} \exp -\frac{1}{2}\left(\frac{t-\mu}{\sigma}\right)^2$$

with $\mu = 0$ and $\sigma = 1$, and so jumps larger than 3 or smaller than -3 are not very likely.

To summarize: after n random steps x_i, y_i the coordinates X_n, Y_n plotted would be

$$X_n = X_{start} + \sum_{i=1}^{n} x_i$$

$$Y_n = Y_{start} + \sum_{i=1}^{n} y_i.$$

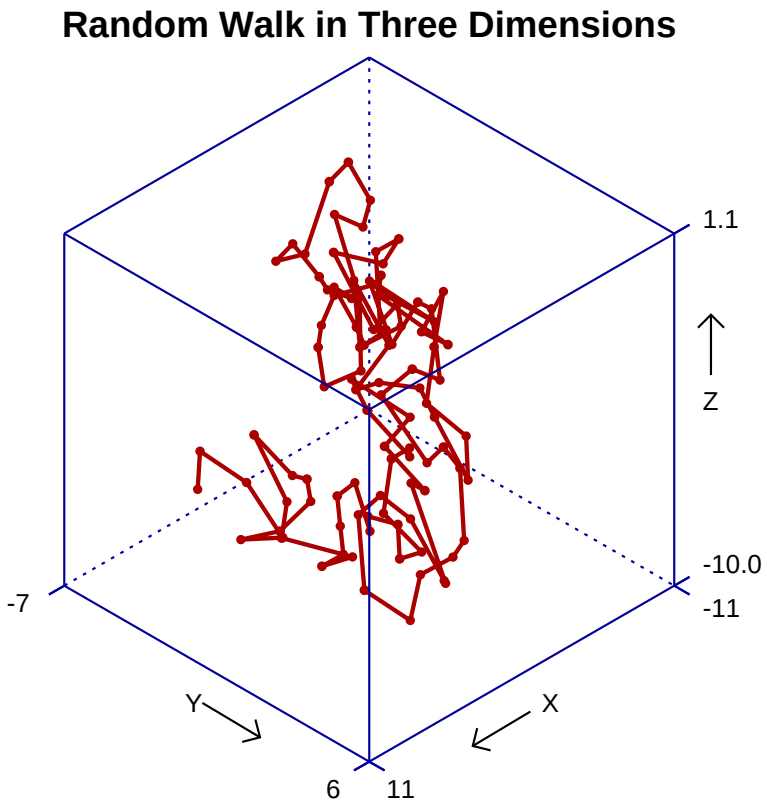
Example 4: Plotting three dimensional random walks

Three dimensional random walks are no different than two dimensional walks except that the procedure for plotting collections of walks differs somewhat.

For instance, the next graph shows a typical three dimensional random walk from the starting point

$$x = 0,\ y = 0,\ z = 0$$

where a standard normal distribution was used for all three variables.



To plot collections of such walks, the walks are first saved from program **rannum** to coordinate files. Then program **simplot** is opened and the coordinate files are supplied using the option for plotting space curves. This anticipates data in parametric form. In other words, it is assumed that the three columns in the coordinate files represent the triples

$$x = f(t)$$
$$y = g(t)$$
$$z = h(t)$$

for arbitrary functions $f(.), g(.),$ and $h(.)$ evaluated at equal increments of parameter t.

Note that this option should not be confused with the option to plot three dimensional surfaces.

10.3 Simulation: User-selected models

Tutorials and worked examples for simulation, curve fitting, statistical analysis, and plotting.
https://simfit.uk
https://simfit.org.uk
https://simfit.silverfrost.com

10.3.1 Simulating a function of one variable

Simulating a data set that is exact to computer precision is the first step in simulation, so that random error can be added retrospectively to simulate experimental results.

Choosing the [A/Z] option from the SimFIT main menu is used to open the program **makdat**, which then allows you to create such almost-exact data from a library of models, or from a user-supplied model.

Before attempting a simulation it is essential to understand these issues.

1. Choosing the correct model
2. Choosing sensible parameters
3. Choosing a reasonable range for the independent variable(s)
4. Choosing a meaningful technique for the distribution of evaluation points
5. Viewing the current simulation
6. Saving the simulated data to a file

Program **makdat** has access to the particular library that is contained in either `w_models.dll` for 32-bit applications, or `x64_models.dll` for 64-bit applications, and there are numerous versions of these where the standard library has been augmented for special user requirements. The standard version has models for one, two, and three variables as well as for single differential equations. For systems of differential equations program **deqsol** must be used.

1. Choosing the correct model

It is essential that users should have a good idea of what would be an appropriate mathematical model.

For instance, it is assumed that biochemists would know which of the following ligand binding models should be used for simulating data for two binding sites.

$$f(x) = \frac{A_1K_1x}{1+K_1x} + \frac{A_2K_2x}{1+K_2x}$$
$$g(x) = \frac{\beta}{2}\left\{\frac{\phi_1x + 2\phi_1\phi_2x^2}{1+\phi_1x+\phi_1\phi_2x^2}\right\}$$

Again, chemists would be expected to know which of the following double exponential models to use for reactants in the two linked irreversible chemical reaction scheme.

$$F(t) = \left\{\frac{p_1p_3}{p_2-p_1}\right\}\exp(-p_1t) + \left\{p_4 - \frac{p_1p_3}{p_2-p_1}\right\}\exp(-p_2t)$$
$$G(t) = p_3+p_4+p_5 - \left\{\frac{p_2p_3}{p_2-p_1}\right\}\exp(-p_1t) - \left\{p_4 - \frac{p_1p_3}{p_2-p_1}\right\}\exp(-p_2t)$$

So the first step in simulation is to choose the correct model and appreciate the relationship between the library parameters, which are all in the form $p_1, p_2, \ldots, p_n$, and the more usual dedicated nomenclature such as $K_m, V_{max}, k_{app}, A_0, B_0$ and so on.

Note that the models contained in the basic library are defined in the SimFIT reference manual `w_manual.pdf`.

2. Choosing sensible parameters

Most users will have a good idea of the parameter values required, normally because these are values reported from curve-fitting and it is wished to check the fit to data based on data simulated using the best-fit parameters.

Often some idea of parameter values can be obtained by simply inspecting experimental data. For instance, all the parameters in models $f(x), g(x), F(t)$, and $G(t)$ must be nonnegative. The final asymptotic level reached by $f(x)$ and $g(x)$ as $x \to \infty$ are of course $A_1 + A_2$ and β respectively, while the binding constants will tend to be around the value of x at the half saturation point. Again, model $F(t)$ tends to zero while $G(t)$ tends to $q_3 + q_4 + q_5$ as $t \to \infty$ and the exponential parameters will be of the order of the half life since $F(t)$ is the intermediate species $B(t)$ while $G(t)$ is the final product $C(t)$ in the irreversible consecutive reaction

$$A \overset{k_1}{\to} B \overset{k_2}{\to} C$$

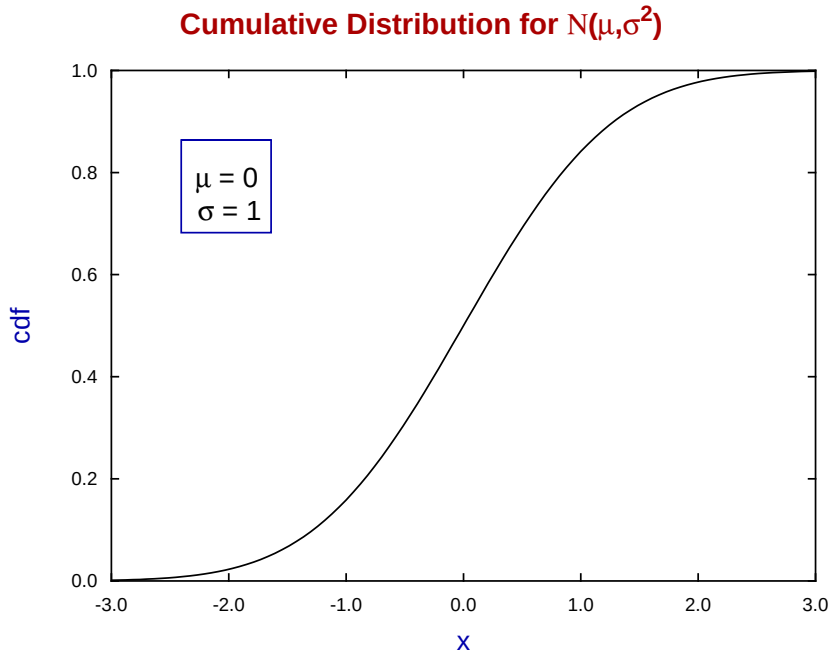
so that $p_1 = k_1, p_2 = k_2, p_3 = A(0), p_4 = B(0)$, and $p_5 = C(0)$.

3. Choosing a reasonable range for the independent variable(s)

In the case of functions of one variable there are four distinct ways to choose starting points and end points that define the range for the independent variable.

- User inputs starting and ending X values manually.
- User inputs starting and ending Y values and corresponding X values are calculated numerically.
- User reads a set of values from a file like `vector.tf1`.
- User edits (or simply accepts) the current data.

For an example of how to set the range manually, run the program **makdat**, select functions of one variable, pick statistical distributions, choose the normal *cdf*, decide to have a zero constant term, set the mean $p(1) = 0$, fix the standard deviation $p(2) = 1$, input the scaling factor $p(3) = 1$ and then choose $X_{\text{start}} = -3$ and $X_{\text{stop}} = 3$ to generate the next figure (after adding the maths).



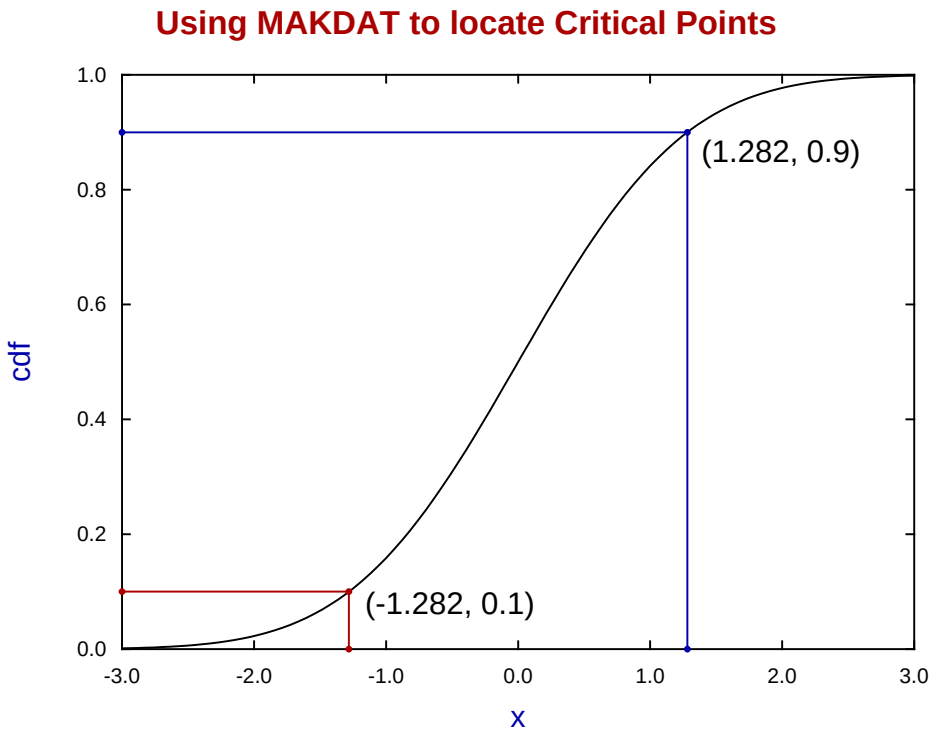
The process of finding a range of x for simulation when this depends on fixed values of y, that is to find $x = x(y)$ when there is no simple explicit expression for $x(y)$, is frequently required. For instance, finding $x = x(y)$ when $y(x)$ is the normal distribution function

$$y(x) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{x} \exp -\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2 \, dx$$

where $\sigma > 0$. After setting parameters $\mu = 0, \sigma = 1$, it is clear from the previous graph that, to find the point x_1 where $y(x_1) = 0.1$ it would be reasonable to guess $-3 \le x_1 \le -1$, while to locate the point x_2 where $y(x_2) = 0.9$ it would be sensible to guess that $1 \le x_2 \le 3$. Using these initial estimates program **makdat** refined them to give the following results

$$X_{\text{start}} = -3,\ X_{\text{stop}} = -1: \quad y_1 = 0.1,\ x_1 = -1.282,$$
$$X_{\text{start}} = 1,\ X_{\text{stop}} = 3: \quad y_2 = 0.9,\ x_2 = 1.282.$$

These critical points are illustrated by the next figure.



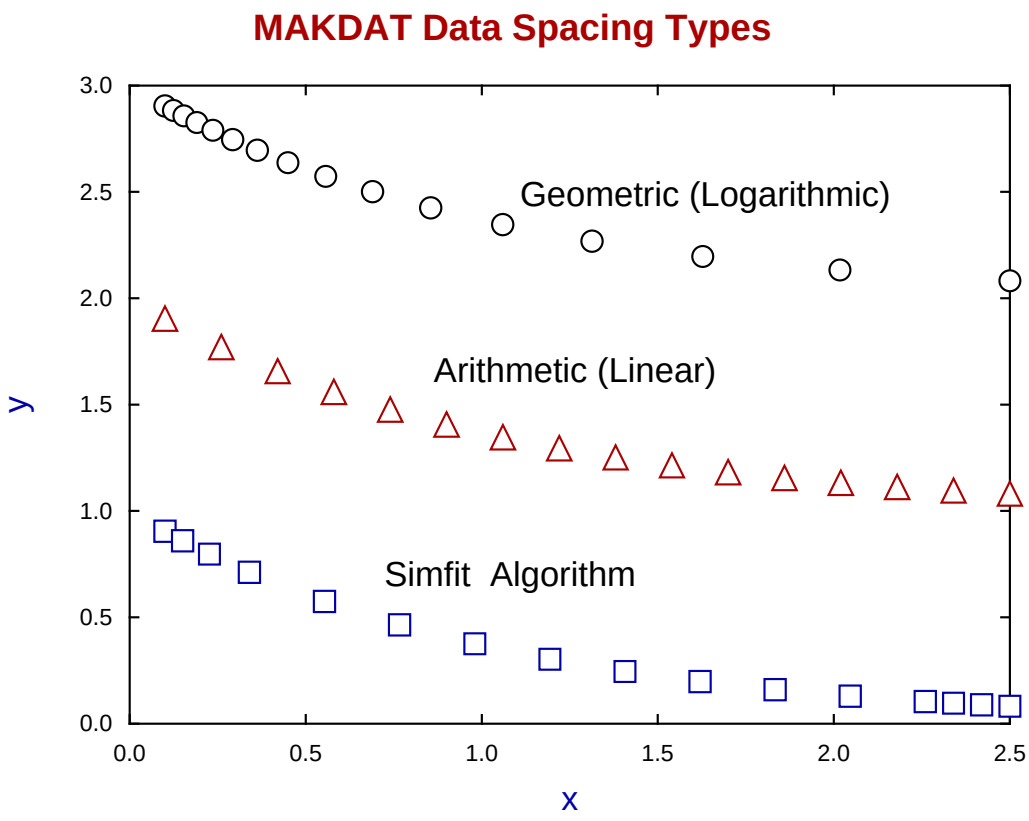
Note that, when attempting such root-finding calculations, **makdat** will attempt to alter the starting estimates by decreasing X_{start} and increasing X_{stop} if a root cannot be located, but it will not change the sign of these starting estimates. In the event of problems locating roots, there is no substitute for plotting the function to get some idea of the position of the roots, as shown in the previous figures.

4. Choosing a meaningful technique for the distribution of evaluation points

For functions of one variable program **makdat** provides three ways to space points.

- Points in a geometric progression (logarithmic spacing)
- Points in an arithmetic progression (linear spacing)
- Points following a SIMFIT spacing algorithm

To illustrate these data spacing options consider the next figure.



Geometric (Logarithmic) spacing

This spacing leads to consecutive points in a geometric progression, i.e. with increasing separation between consecutive points, and requires positive coordinates. From the point of view of optimum design for parameter estimation and model discrimination with models like the exponentials and rational functions so often encountered in experiments, this is the best choice as discussed in the following publication.

Optimal design for model discrimination using the F test with non-linear biochemical models. Criteria for choosing the number and spacing of experimental points.
Bardsley,W.G., McGinlay,P.B & Roig,M.G. (1989) *J. theor. Biol.* **139**, 85-102

Arithmetic (Linear) spacing

This spacing leads to consecutive points in an arithmetic progression, i.e. with a constant separation between consecutive points, and is the design most used, probably out of of convenience. However it is a bad choice for many situations where it is best to place more points where model equations change most rapidly, i.e. near the origin.

Simfit Algorithm

The SIMFIT spacing is used by SIMFIT when a best-fit curve is calculated for plotting against experimental data. This spacing is a compromise as it attempts to cover the cases where data and best-fit curves are plotted in other transformations than the logarithmic, like a Scatchard plot or double reciprocal plot for instance. It approximates to geometric spacing for early points and reverse geometric spacing for late points.

5. Viewing the current simulation

Once a data set has been simulated it is always possible to display a table of the independent variables and simulated values. In addition, for one independent variable a plot can be created, while for two independent variables a surface can be plotted together with contours if required. It is not possible to plot a function of three independent variables.

6. Saving the simulated data to a file

Files saved will have the independent variable(s) in the first column(s), then the simulated values in the penultimate column, followed by a final column of weights.

There are three reasons for saving the simulated data to a file.

- **Using the saved file for fitting**

  It is assumed that normally data are simulated in order to be fitted. So a final column of weights (usually 5% of the calculated function value) will be added to the saved file to make it a curve fitting file. The last column must be left in for weighting, or else replaced by a column of 1 if unweighted fitting is required. This is easily done using program **editmt**.

- **Using the saved file for plotting**

  As the final column of weights is not necessary for plotting it would be usual to delete the last column of weights using program **editmt**. However, for plotting functions of one variable using program **simplot** there is no need to delete the last column (i.e. column 3). On the other hand, for a functions of two or three variables for plotting, the last column of weights must always be deleted.

- **Adding random error**

  As the data written to file will be almost exact, they can be used to confirm that a model simulated with no added error can be fitted to return the correct best-fit parameters. If it is wished to add random error to simulate experimental data, the files can be input into program **adderr**, which can also overwrite the default weights by alternative weighting schemes if required.

10.3.2 Simulating a function of two variables

Simulating a data set for a function of two variables that is exact to computer precision is the first step in simulation, so that random error can be added retrospectively to simulate experimental results.

Choosing the [A/Z] option from the SimFIT main menu is used to open the program **makdat**, which then allows you to create such almost-exact data from a library of models, or from a user-supplied model.

Before attempting to simulate a function of two variables it is essential to understand these issues.

1. Choosing the correct model
2. Choosing sensible parameters
3. Choosing a reasonable range for the independent variable(s)
4. Choosing a sensible number of divisions of the X and Y axes
5. Viewing the current simulation
6. Saving the simulated data to a file

Program **makdat** has access to the particular library that is contained in either `w_models.dll` for 32-bit applications, or `x64_models.dll` for 64-bit applications, and there are numerous versions of these where the standard library has been augmented for special user requirements. The standard version has models for one, two, and three variables as well as for single differential equations. For systems of differential equations program **deqsol** must be used.

1. Choosing a model

From the interface to the SimFIT library choose a polynomial of degree two which has the following form.

$$f(x, y) = p_1x + p_2y + p_3x^2 + p_4xy + p_5y^2 + p_6$$

This can be simulated using the default parameters but the result is not very interesting.

2. Choosing sensible parameters

A more illuminating example would be the function

$$f(x, y) = x^2 - y^2$$

which can easily be simulated by editing the default parameters as follows.

$$\begin{aligned} p_1 &= 0 \\ p_2 &= 0 \\ p_3 &= 1 \\ p_4 &= 0 \\ p_5 &= 1 \\ p_6 &= 0 \end{aligned}$$

3. Choosing a reasonable range for the independent variable(s)

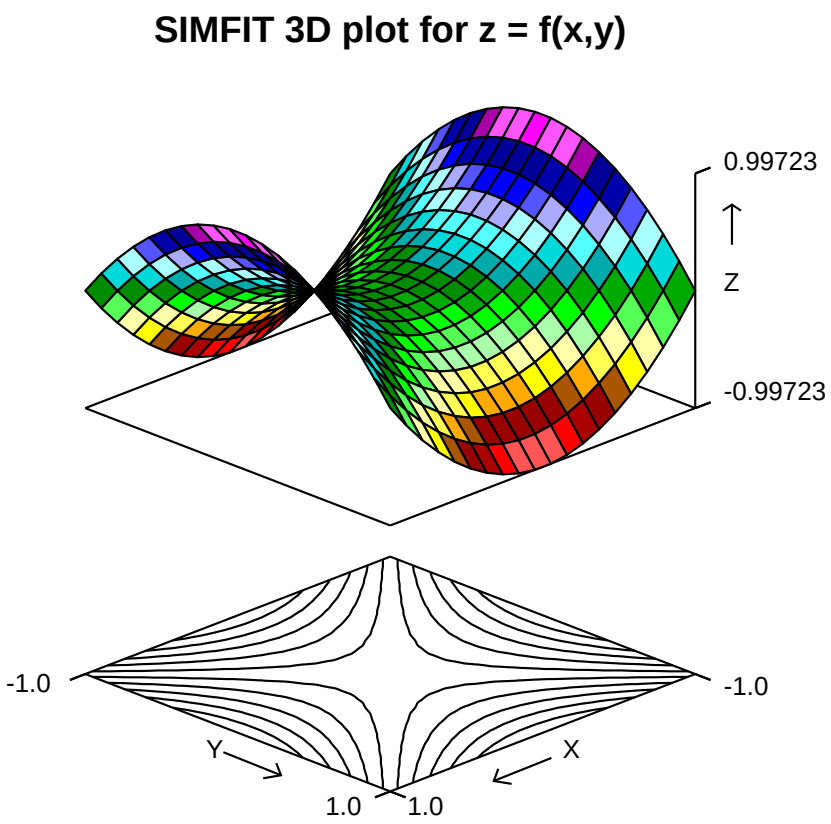
In this case it is suggested that you use the default values which are

$$-1 \leq x \leq 1$$
$$-1 \leq y \leq 1.$$

Choosing the number of divisions should be considered carefully. For instance, accepting the default values of 20 divisions along the axes requires the simulation of 400 data points. With this example the surface can be viewed without problems but, for other models, the range and number of points will depend on the model selected and parameters chosen.

4. Viewing the current simulation

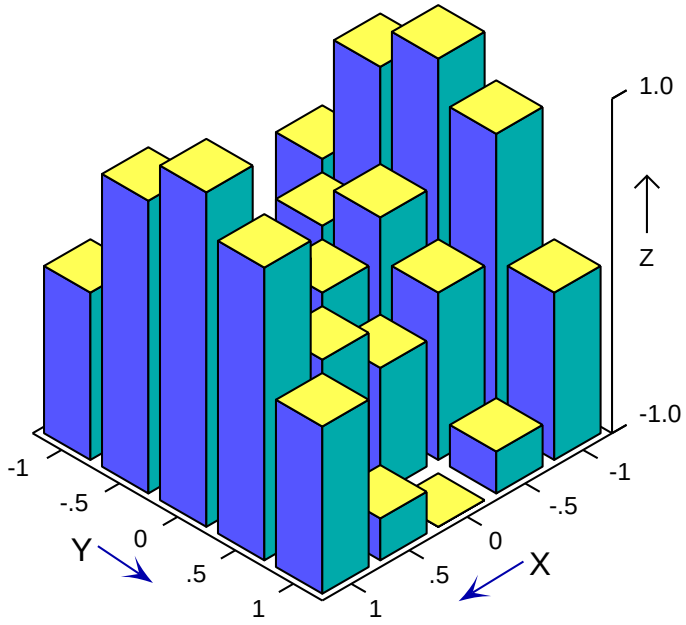
Once a data set has been simulated it is always possible to display a table of the independent variables and simulated values or a plot as follows.



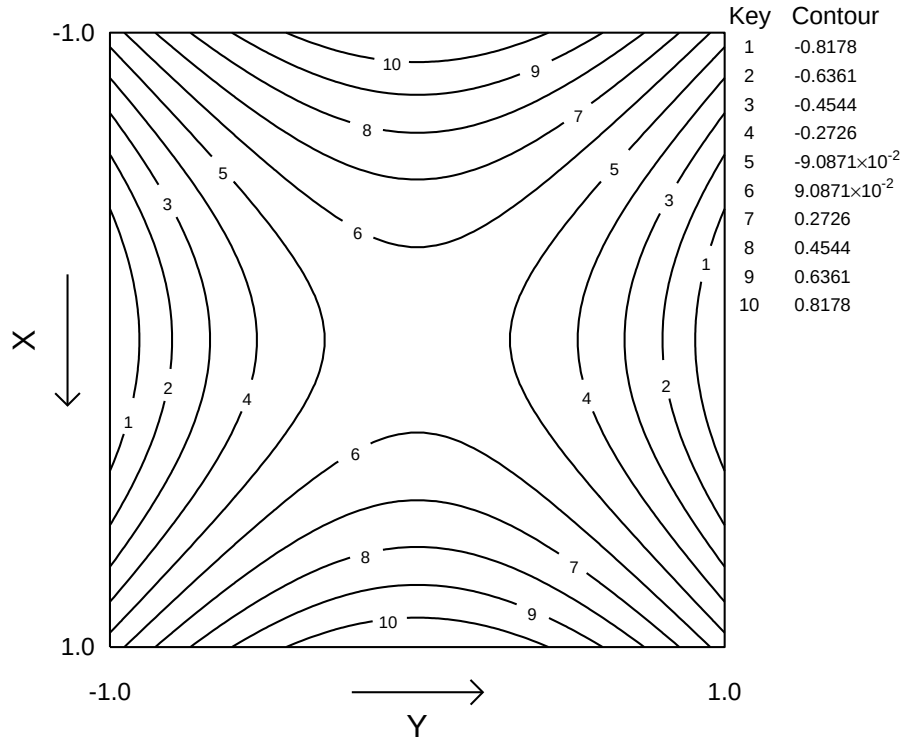
This example highlights the surface with colored rectangles (facets) and includes the corresponding contours, but there are a great many other ways to view the surface, including skyscraper and cylinder plots with or without error bars if the number of divisions is suitably small.

Note that, in order to generate a convincing contour diagram, it may be necessary to increase the number of divisions up to the maximum of 100, which would then involve the simulation of 10000 function values. A skyscraper plot with 5 divisions and a contour plot with 50 divisions for the same function and parameters are illustrated next.

SIMFIT 3D skyscraper plot for $z = x^2 - y^2$



SIMFIT 2D contour plot for $z = x^2 - y^2$



5. Saving the simulated data to a file

Curve-fitting files saved will have the independent variables in the first two columns, then the simulated values in the penultimate column, followed by a final column of weights. However such files cannot be used for plotting as will be explained below.

There are three reasons for saving the simulated data to a file.

- **Using the saved file for fitting**

  It is assumed that normally data are simulated in order to be fitted. So a final column of weights (usually 5% of the calculated function value) will be added to the saved file to make it a curve fitting file. The last column must be left in for weighting, or else replaced by a column of 1 if unweighted fitting is required. This is easily done using program **editmt**.

- **Using the saved file for plotting**

  In order to plot a surface SIMFIT requires plotting files in a special format. For n_x divisions of the x axis and n_y divisions of the y axis a menu option from the plot concerned allows the creation of a vector plotting file with a column of $N = n_x n_y + 6$ values. After the usual title and row dimension (N) and column dimension (1), the first six values in such a SIMFIT vector file are as follows.

  1. Line 1: n_x the number of x divisions
  2. Line 2: n_y the number of y divisions
  3. Line 3: initial x value
  4. Line 4: final x value
  5. Line 5: initial y value
  6. Line 6: final y value

  After these are the N function values in column-major format with priority x.

- **Using the saved file for adding random error**

  As the data written to file will be almost exact, they can be used to confirm that a model simulated with no added error can be fitted to return the correct best-fit parameters. If it is wished to add random error to simulate experimental data, the files can be input into program **adderr**, which can also overwrite the default weights by alternative weighting schemes if required.

10.3.3 Simulating a differential equation

Simulating a data set that is exact to computer precision for differential equations is the first step in simulation, so that random error can be added retrospectively to simulate experimental results.

Choosing the [A/Z] option from the SimFIT main menu is used to open the program **makdat**, which then allows you to create such almost-exact data from a library of models, or from a user-supplied model.

Before attempting to simulate a differential equation it is essential to understand these issues.

1. Choosing the correct model
2. Choosing sensible parameters (including initial conditions)
3. Choosing a reasonable range for the independent variable
4. Choosing a sensible number of divisions of the X axis
5. Viewing the current simulation
6. Saving the simulated data to a file

Program **makdat** has access to the particular library that is contained in either `w_models.dll` for 32-bit applications, or `x64_models.dll` for 64-bit applications, and there are numerous versions of these where the standard library has been augmented for special user requirements.

The standard version has models for a single differential equation, but for systems of n differential equations program **deqsol** must be used.

1. Choosing a model

From the interface to the SimFIT library choose the Michaelis-Menten substrate depletion model which has the following form

$$\frac{dy}{dx} = \frac{-p_2 y}{p_1 + y}.$$

Of course this model can actually be integrated and would usually have the following interpretation

$$\frac{dS}{dt} = \frac{-V_{max} S}{K_m + S}.$$

2. Choosing sensible parameters

There is an additional complication with differential equations as the the initial conditions are also parameters, and the SimFIT convention is that with a system of n equations the last n parameters must be the initial conditions. Note that the default parameters for this model are as follows.

$$\begin{aligned} p_1 &= 1 \\ p_2 &= 1 \\ p_3 &= 1 \\ &= S(0) \end{aligned}$$

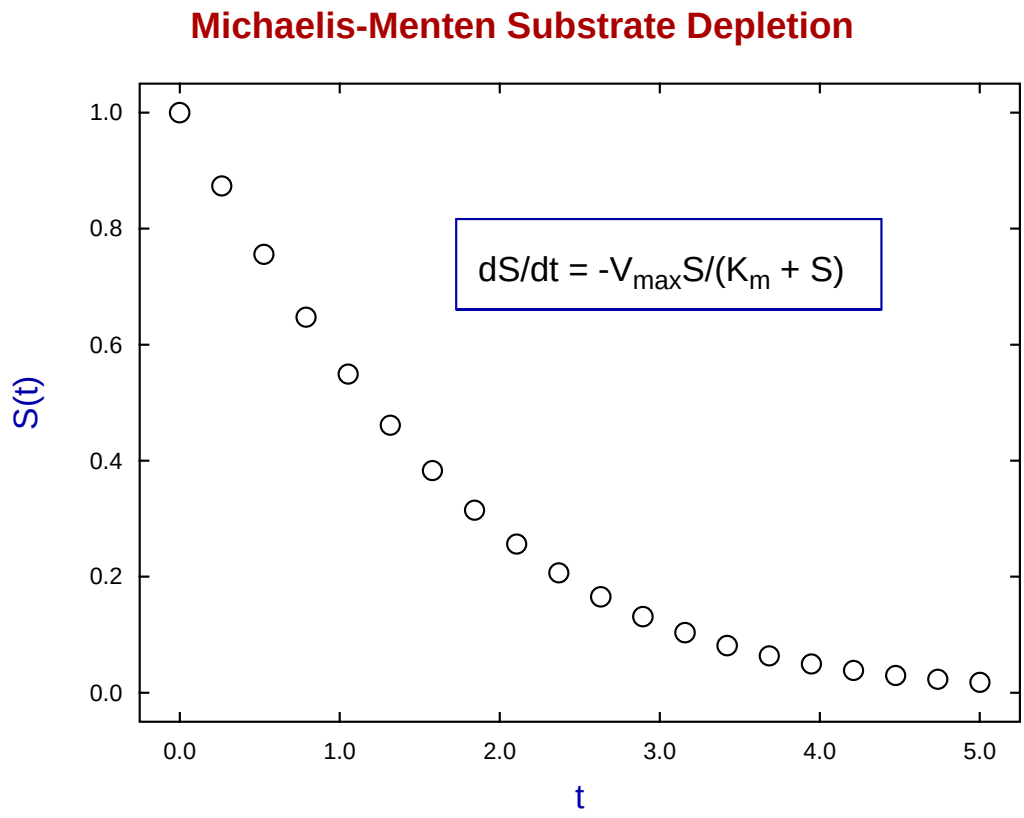
so that, with this model, parameter $p_3 > 0$ is the substrate concentration at time $t = 0$.

3. Choosing a reasonable range for the independent variable(s)

It is usual to advance the solution for differential equations from time zero, which means that program **makdat** will not allow starting values less that zero.

4. Viewing the current simulation

Once a data set has been simulated it is always possible to display a table of the independent variables and simulated values, or a plot as follows.



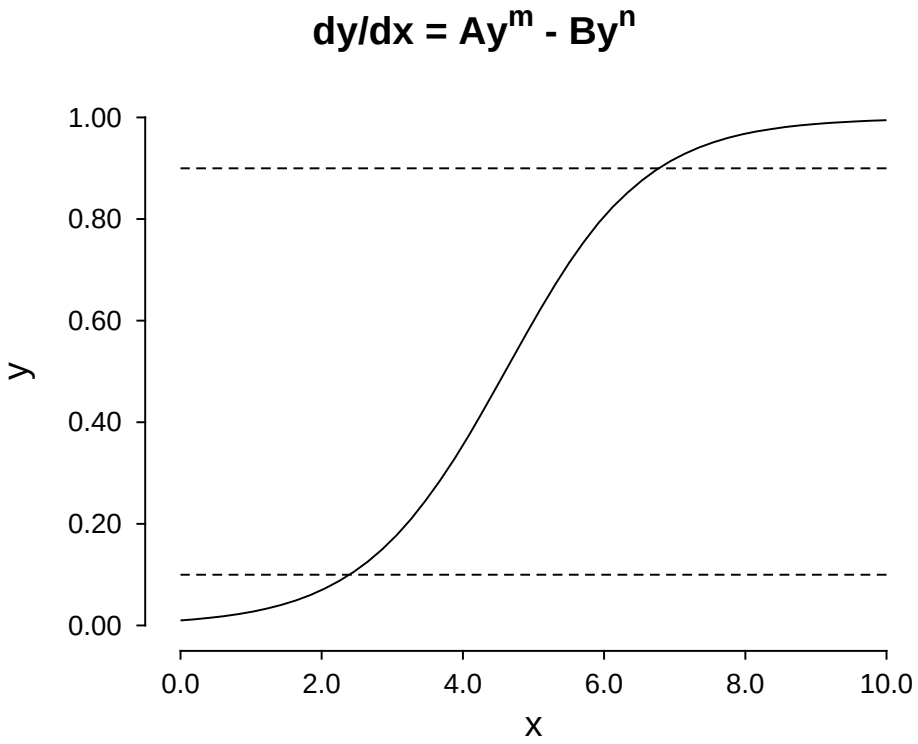
To illustrate the process of finding a range of x for simulation when this depends on fixed values of y, that is to find $x = x(y)$ when there is no simple explicit expression for $x(y)$, consider the Von Bertalanffy growth differential equation

$$\frac{dy}{dx} = Ay^m - By^n,$$

where $A > 0, B > 0$ and $n > m$. After setting the parameters $A = B = m = 1$, $n = 2$, and initial condition $y_0 = 0.001$, for instance, program **makdat** estimated the following results:

$X_{\text{start}} = 0,\ X_{\text{stop}} = 5,\ y_1 = 0.1,\ x_1 = 2.3919,$
$X_{\text{start}} = 0,\ X_{\text{stop}} = 9,\ y_2 = 0.9,\ x_2 = 6.7679,$

providing the roots required to simulate this equation between the limits $y_1 = 0.1$ and $y_2 = 0.9$.



Note that, when attempting such root-finding calculations, **makdat** will attempt to alter the starting estimates by decreasing X_{start} and increasing X_{stop} if a root cannot be located, but it will not change the sign of these starting estimates. In the event of problems locating roots, there is no substitute for plotting the function to get some idea of the position of the roots, as shown in the previous figure.

5. Saving the simulated data to a file

Curve-fitting files saved will have the independent variables in the first column, then the simulated values in the penultimate column, followed by a final column of weights.

There are three reasons for saving the simulated data to a file.

- **Using the saved file for fitting**
  It is assumed that normally data are simulated in order to be fitted. So a final column of weights (usually 5% of the calculated function value) will be added to the saved file to make it a curve fitting file. The last column must be left in for weighting, or else replaced by a column of 1 if unweighted fitting is required. This is easily done using program **editmt**.
- **Using the saved file for plotting**
  The simulated data can be used retrospectively for plotting by program **simplot** which ignores the final column of weights.
- **Using the saved file for adding random error**
  As the data written to file will be almost exact, they can be used to confirm that a model simulated with no added error can be fitted to return the correct best-fit parameters. If it is wished to add random error to simulate experimental data, the files can be input into program **adderr**, which can also overwrite the default weights by alternative weighting schemes if required.

10.3.4 Simulating a user-defined model: creating a model-file

In the unlikely event that the model you want to plot, simulate or fit is not in the SimFIT library of compiled models you will have to create a user–defined–model file. Actually, once the format for user–defined model equations is understood, it is very simple to create model files using any text editor. However, program **usermod** can be opened using the [A/Z] option on the SimFIT main menu, and this is an easy way to create models and understand how to use them. It provides these options.

- Open a file from the selection of demonstration examples provided using the [Demo] button on the File-Open dialogue.
- View the model files supplied to examine the way these test examples formulate the various models and sets of models.
- Use the options provided to check the models or use them to plot, integrate, locate zeros, or perform constrained optimization.
- Invoke the facility to open a user-defined file or a default template in a text editor to create a model file, then save it to a temporary file and read it back into the main program to check, etc. until the syntax is correct before finally archiving the model.

From SimFIT version 7.1.6 onwards there are two procedures that can be used to define models.

1. **Reverse Polish**
   This is the most versatile technique, but it is very verbose and may be found difficult to use by non-programmers.
2. **Standard mathematical expressions**
   This is far easier to understand by scientists, and is recommended when developing new models since it uses normal mathematical formulas but then SimFIT automatically transforms them into reverse Polish at run time.

The most versatile manner is to use standard mathematical expressions for simple one-line equations, but to resort to a mixture of standard expressions linked by permitted SimFIT reverse Polish stack operations for complicated models.

It should be pointed out that models in the compiled library are protected against critical events such as taking logs or square roots of negative numbers, or dividing by zero, but it is your responsibility when fitting your own user–defined models that the parameter values and independent variables are constrained to avoid such singularities.

Reverse Polish

Reverse Polish (i.e. last–in–first–out, or post–fix) is an excellent way to prepare user–defined models for a number of reasons.

1. It can be used to express any mathematical model unambiguously.
2. It is very similar to the way that computers perform calculations.
3. It is the way that the PostScript language and programmable calculators work.
4. There many model files distributed with SimFIT to demonstrate the technique and documentation to explain it.
5. **usermod** can be used to view these then develop your own models, check them for correct syntax, then use them for plotting, calculating integrals, finding zeros of functions, fitting, or optimization.

As a simple example to introduce this technique, consider how to formulate damped simple harmonic motion, namely

$$f(x) = \delta \exp(-\gamma t)\cos(\alpha t - \beta)$$

with

$$\begin{aligned} \text{Frequency } \alpha &= p(1) \\ \text{Offset } \beta &= p(2) \\ \text{Decay constant } \gamma &= p(3) \\ \text{Amplitude } \delta &= p(4), \text{ and} \\ \text{Independent variable } t &= x. \end{aligned}$$

Here, for example, is the test file `usermod1.tf9` to simulate damped simple harmonic motion with these definitions

```
%
f(x) = p(4)*exp[-p(3)*x]*cos[p(1)*x - p(2)]
%
1 equation
1 variable
4 parameters
%
p(1)
x
multiply
p(2)
subtract
cosine
p(3)
x
multiply
negative
exponential
multiply
p(4)
multiply
f(1)
%
```

It consists of three distinct sections, separated by percentage signs as follows.

Section 1 Up to 24 lines containing any information with no formatting restrictions.

Section 2 The number of equations, variables (or else the phrase differential equation), and parameters.

Section 3 The reverse Polish commands.

In addition to browsing the library of models available using the [View] option from the main SIMFIT menu, and examining the tutorial documents on the SIMFIT website, such as `user_defined_models.html`, there are several additional sources of help.

- The readme files `w_readme.f4`, `w_readme.f5`, ... `w_readme.f10` also available from the [View] option on the main SIMFIT menu.
- The SIMFIT reference manual `w_manual.pdf` available from the [Manual] option on the main SIMFIT menu.
- The help sections provided by program **usermod** which can be opened from the [A/Z] option on the main SIMFIT menu.

It is recommended that the first few user-defined model files, i.e. `usermod1.tf1`, `usermod1.tf2`, and `usermod1.tf3` be examined carefully as they contain appended sections containing an analysis of the way the reverse Polish model definition scheme works.

Standard expressions

The usual mathematical notation can be used at any point in a model file provided that it is included in a `begin{expression}... end{expression}` structure like the following code for defining a cubic.

```
begin{expression}
f(1) = p(1) + p(2)x + p(3)x^2 + p(4)x^4
end{expression}
```

The essential rules for using expressions are now listed.

- Parameters must be indexed as `p(i)` where the index `i` is consistent with the header defining the number of parameters.
- The independent variables must be consistent with these rules.

  1 independent variable: x

  2 independent variables: x, y

  3 independent variables: x, y, z

  n independent variables: $y(1), y(2), \ldots, y(n)$
- The only other symbols that can be used are dummy variables (such as $A = x^2 + y^2$, etc. to avoid repetition as discussed below) and those listed for for use with SimFIT reverse Polish, e.g.
  `cos, sin, tan, exp, log, log10, pi, sqrt`, etc.
  with the constants and special functions described in `commands.txt` and `w_manual.pdf`.
- Successive functions must be indexed as `f(j)` where the index `j` is consistent with the header defining the number of equations.
- Make liberal use of the `*` sign for multiplication, and employ brackets `{.}, [.], (.)` to avoid ambiguities, for example:

  Use `x*log(x)` instead of `xlogx`

  Use `1/(2pi)` instead of `1/2pi`

  Use `exp(-kt)` instead of `e^-kt`
- It is possible to spread equations over several lines in an expression construct but blank lines should not be used anywhere in the model defining section of a user–defined model file.
- When the expression has been evaluated the result is added to the top of the stack so it can be left there, duplicated, stored, or used immediately to define a function value, which would pop the value off the stack and return the function value for use by the calling procedure.

It should be pointed out that when SimFIT reads a user-defined model file it creates a temporary copy called `f$parser.tmp` in your temporary file folder where equations defined in expression constructs are parsed and written out in reverse Polish. For this reason it is important to realize the need to examine this temporary file if the model does not compute properly, or crashes with error messages due to unrecognizable elements in expressions which have been written unchanged into the reverse Polish code.

Here is `usermod1_e.tf9` defining the previous model using a standard mathematical expression. Note the use of `_e` to indicate a model using expressions. It should be noted how the command `f(1)` is used to define the function.

```
%
f(x) = p(4)exp[-p(3)x]cos[p(1)x - p(2)]
%
1 equation
1 variable
4 parameters
%
begin{expression}
f(1) = p(4)exp(-p(3)x)cos(p(1)x - p(2))
end{expression}
%
```

Here is another example of the model file `d01faf_e.mod` that is much more succinct using the expression technique rather than reverse Polish

```
%
f(y) = {4y(1)y(3)^2[exp(2y(1)y(3))]}/{1 + y(2) + y(4)}^2
%
1 equation
4 variables
0 parameters
%
begin{expression}
f(1) = 4y(1)y(3)^2[exp(2y(1)y(3))]/[1.0 + y(2) + y(4)]^2
end{expression}
%
```

Here is an example with nine functions of nine variables `c05nbf_e.mod` which is much easier to understand than the reverse Polish version.

```
%
f(1)=(3-2x(1))x(1)-2x(2)+1, ..., f9=-x(8)+(3-2x(9))x(9)+1
%
9 equations
9 variables
0 parameters
%
begin{expression}
f(1) = (3 - 2y(1))y(1) + 1 - 2y(2)
f(2) = (3 - 2y(2))y(2) + 1 - y(1) - 2y(3)
f(3) = (3 - 2y(3))y(3) + 1 - y(2) - 2y(4)
f(4) = (3 - 2y(4))y(4) + 1 - y(3) - 2y(5)
f(5) = (3 - 2y(5))y(5) + 1 - y(4) - 2y(6)
f(6) = (3 - 2y(6))y(6) + 1 - y(5) - 2y(7)
f(7) = (3 - 2y(7))y(7) + 1 - y(6) - 2y(8)
f(8) = (3 - 2y(8))y(8) + 1 - y(7) - 2y(9)
f(9) = (3 - 2y(9))y(9) + 1 - y(8)
end{expression}
%
```

Using dummy variables in standard mathematical expressions

A common feature required when creating models with long or complicated expressions is to define dummy intermediate values to improve readability as in this example for a rational function

```
begin{expression}
alpha = p(1)x + p(2)x^2
beta = 1 + p(3)x + p(4)x^2
f(1) = alpha/beta
end{expression}
```

or, more usually, to save intermediate values for repeated re-use as in this example for `optimum_e.mod`.

```
%
Rosenbrock's 2-dimensional function for optimisation
f(1) = 100(y - x^2)^2 + (1 - x)^2
f(2) = d(f(1))/dx = -400x(y - x^2) - 2(1 - x)
f(3) = d(f(1))/dy = 200(y - x^2)
%
3 equations
2 variables
0 parameters
%
begin{expression}
A = y - x^2
```

```
B = 1 - x
f(1) = 100A^2 + B^2
f(2) = -400A - 2B
f(3) = 200A
end{expression}
```

Note the use of **A** and **B** dummy variables to avoid re-calculations. During model evaluation this method implements the SimFIT `get(.)` and `put(.)` reverse Polish commands to archive and retrieve dummy variables.

This technique is especially useful when formulating Jacobians for systems of differential equations, as there are frequently common expressions between the family of differential equations and their Jacobians.

10.3.5 Simulating a user-defined model: simple examples

Simulating a data set that is exact to computer precision for a user-defined model is often the first step in simulation, so that random error can be added retrospectively to simulate experimental results.

SimFIT can simulate a very large number of models from user-supplied model files and allows models to include the following procedures.

1. All the standard mathematical operations such powers, logarithms, trigonometric and hyperbolic functions, etc.
2. A very substantial library of special functions including erf, erfc, and Bessel functions, etc.
3. A large number of numerical techniques such as quadrature, root finding, eigenvalues, matrix operations, and convolution of two functions, etc.
4. Systems of differential equations with Jacobians either supplied or approximated numerically.
5. Logical tests for branching.
6. Models that include calls to sub-models.

Such models can also be used for nonlinear regression, and details will be found in the following publication

Using ASCII text files in post-fix notation (reverse Polish) to define mathematical models and systems of differential equations for simulation and nonlinear regression.
Bardsley,W.G. & Prasad,N *Computers and Chemistry* (1997) **21**, 71–82,

but also in the SimFIT reference manual `w_manual.pdf`, or in `user_defined_models.html`.

However, it is also possible to use standard mathematical notation in order to define a model as long as the model is contained within a `begin{expression}` and a `end{expression}` section as will be demonstrated for the models to be used here. For many of the demonstration model files supplied with SimFIT there are two versions, one in reverse Polish, e.g. `usermod1.tf9`, and one for the same model but with subscript `_e`, e.g. `usermod1_e.tf9`, using standard mathematical expressions.

Choosing the [A/Z] option from the SimFIT main menu is used to open the program **makdat**, which then allows you to create such almost-exact data from a a user-supplied model.

Example 1. Damped simple harmonic motion

For a simple example we will investigate how to simulate damped simple harmonic motion given by

$$f(x) = \exp(-x/2)\cos 5x.$$

For instance, the test file `usermod1_e.tf9` has the following model for damped simple harmonic motion

$$f(x) = p_4 \exp(-p_3 x)\cos(p_1 x - p_2)$$

which simply requires choosing the parameters

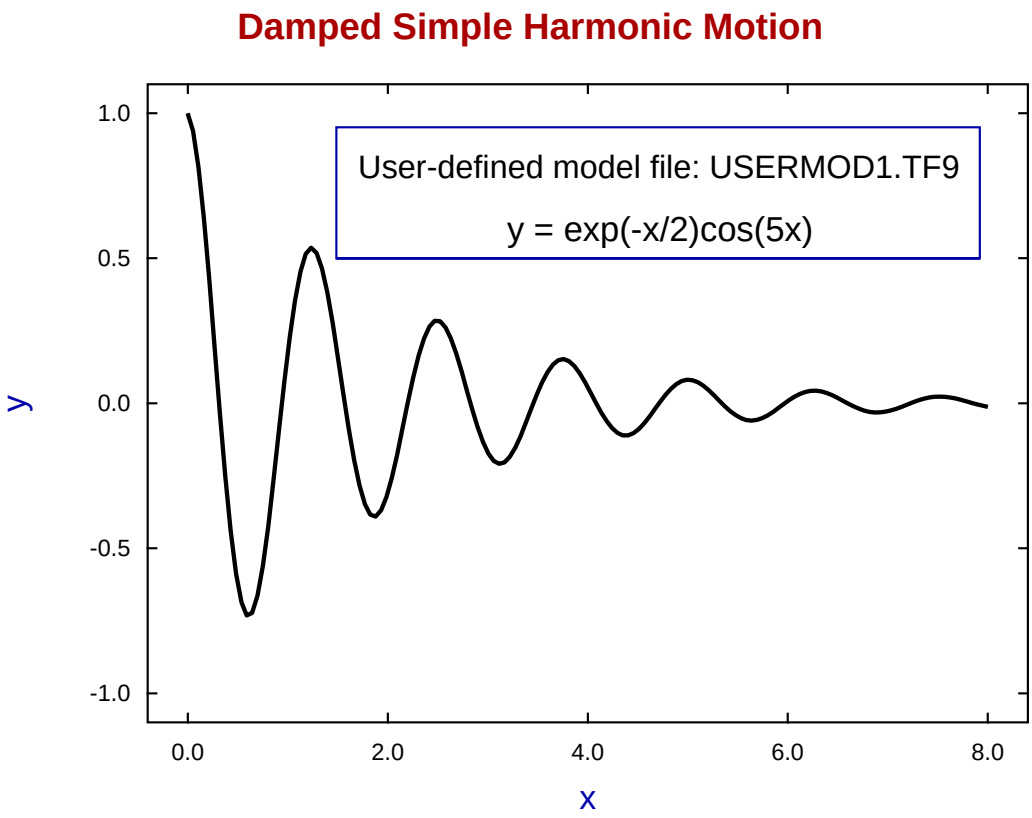
$$\begin{aligned} p_1 &= 5 \\ p_2 &= 0 \\ p_3 &= 0.5 \\ p_4 &= 1. \end{aligned}$$

Model files are simple ASCII text files with a title, details of the number of equations, then the model in reverse Polish, (i.e. last-in first-out which is how computers work and some languages like PostScript), with standard mathematical notation included if required.

Note that the percentage sign(%) is used as a special character to denote change in content, and that there is a SIMFIT program **usermod** to help users develop such models.

The model file and simulated data are shown next.

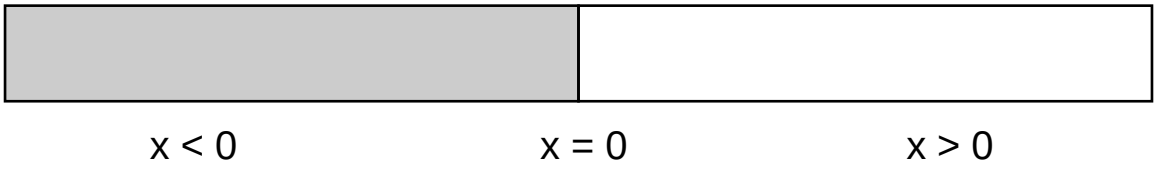
```
%
Example: user supplied function of 1 variable ... damped SHM
         Damped simple harmonic motion in the form
         f(x) = p(4)*exp[-p(3)*x]*cos[p(1)*x - p(2)]
         where p(i) >= 0
 %
 1 equation
 1 variable
 4 parameters
 %
 begin{expression}
 f(1) = p(4)exp(-p(3)x)cos(p(1)x - p(2))
 end{expression}
 %
```



Example 2: Diffusion in a long tube

This example illustrates how to use special functions such as erfc in a user-defined model.

Consider a very long tube of constant cross section filled with solvent but with a barrier separating the tube into two sections with half the tube filled with solute at concentration C_0 as illustrated next, using a coordinate system with the barrier at position $x = 0$.



In an idealized situation where the barrier is removed instantaneously at time $t = 0$ with no mixing effect there would be an initial situation as follows

$$C(x,t) = C_0, \text{ for } x < 0$$
$$C(x,t) = \tfrac{1}{2}C_0, \text{ at } x = 0$$
$$C(x,t) = 0, \text{ for } x > 0.$$

If there is no effect operating except for simple diffusion then the concentration subsequently would obey the diffusion equation

$$\frac{\partial C}{\partial t} = D\frac{\partial^2 C}{\partial x^2}$$

with solution

$$C(x,t) = \tfrac{1}{2}C_0 \operatorname{erfc} \frac{x}{2\sqrt{(Dt)}}.$$

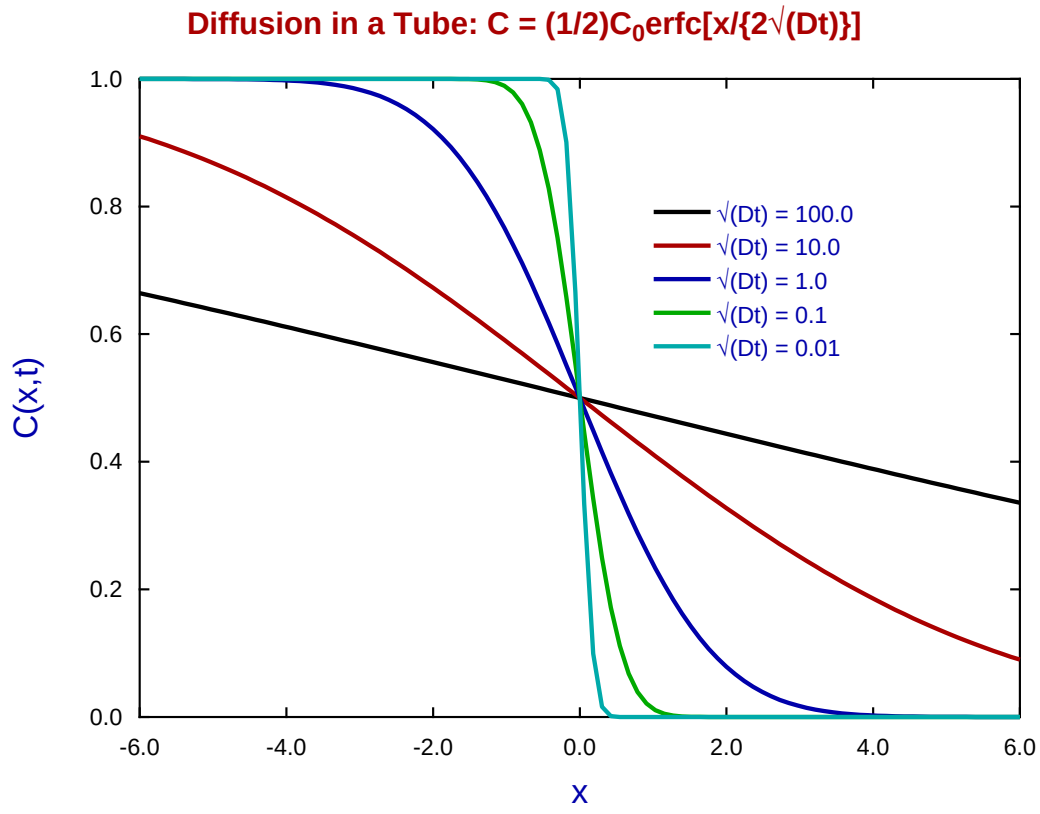
and the SIMFIT test file `usermod1_e.tf8` shown below contains the text required to simulate this equation.

```
 %
Example: user supplied function of 1 variable ... capillary diffusion

f(x) = p(1)*erfc[x/(2*sqrt(p(2))]

 %
 1 equation
 1 variable
 2 parameters
 %
 begin{expression}
 f(1) = p(1)erfc(x/(2sqrt(p(2))))
 end{expression}
 %
```

With $p_1 = 0.5$ then $p_2 = \sqrt{(Dt)}$ varied the following graph was simulated to illustrate the result of diffusion as a function of distance and time with $p_2 = 0.01, 0.1, 1.0, 10.0, 100.0$, and fixed diffusion constant $D = 1$. Evidently diffusion continues until the concentration everywhere is $\frac{1}{2}C_0$.



To generate this graph, program **makdat** was used with the range of x fixed at $-6 \le x \le 6$, and after each plot the [Advanced] option was used to save each profile as an ASCII text file that was recorded in the graphics project archive. This was then opened by program **simplot** to draw the composite graph above.

Theory

The SIMFIT procedures for user defined models include many one-line commands to evaluate special functions, and the case of the complementary error function used in mode file `usermode1_e.tf8` is typical.

Note that the error function erf and the complementary error function erfc are defined as

$$\begin{aligned}
\mathrm{erf}(x) &= \frac{2}{\sqrt{\pi}} \int_0^x \exp(-t^2)\, dt \\
\mathrm{erf}(-x) &= -\mathrm{erf}(x) \\
\mathrm{erfc}(x) &= \frac{2}{\sqrt{\pi}} \int_x^\infty \exp(-t^2)\, dt \\
\mathrm{erf}(x) + \mathrm{erfc}(x) &= 1
\end{aligned}$$

so that $-1 \le \mathrm{erf}(x) \le 1$ and $0 \le \mathrm{erfc}(x) \le 2$. As with all special functions listed in the documents `w_manual.pdf` and `user-defined-models.html`, these can only be evaluated using numerical techniques.

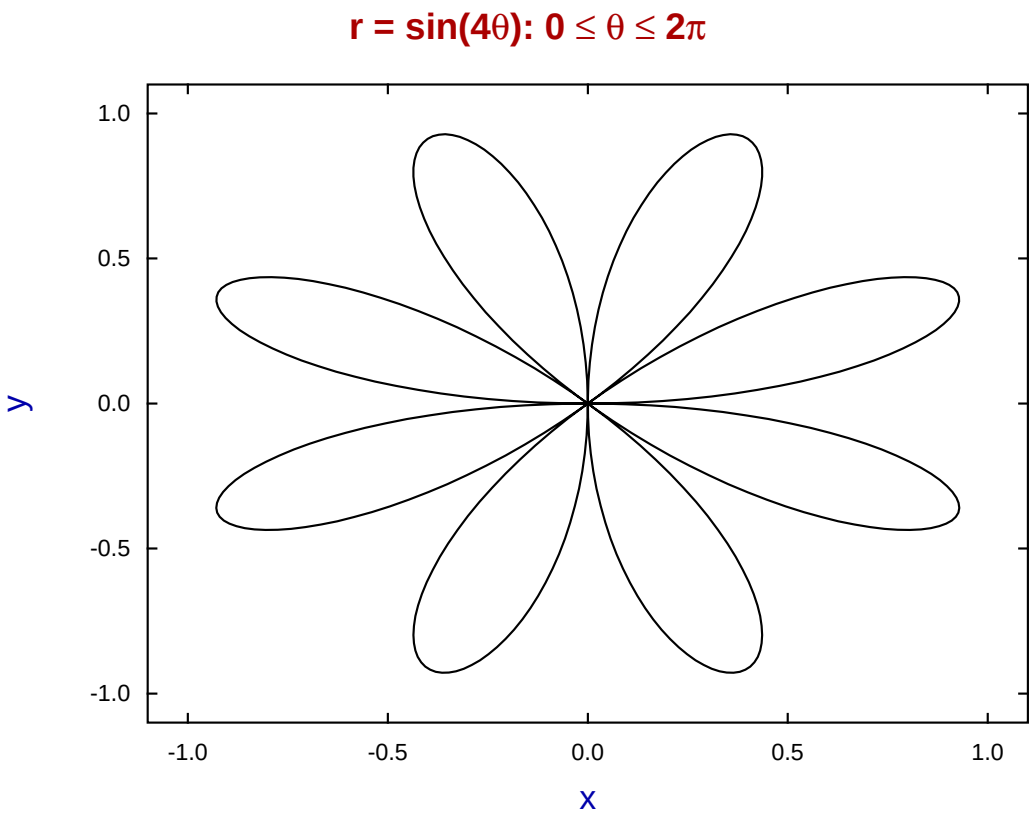
10.3.6 Simulating a user-defined model: parametric curves

It is frequently required to plot a curve where the model has to be defined parametrically as $x(t), y(t)$ instead of $y = f(x)$, or sometimes a parametric space curve $x(t), y(t), z(t)$ may be required.

Example 1: The curve $r = A\sin(4\theta)$

This is achieved by opening **makdat** then reading in the parametric model file `rose_e.mod` shown below to create the plot illustrated for the parameter values $p_1 = 1$, and $0 \le \theta \le 2\pi$.

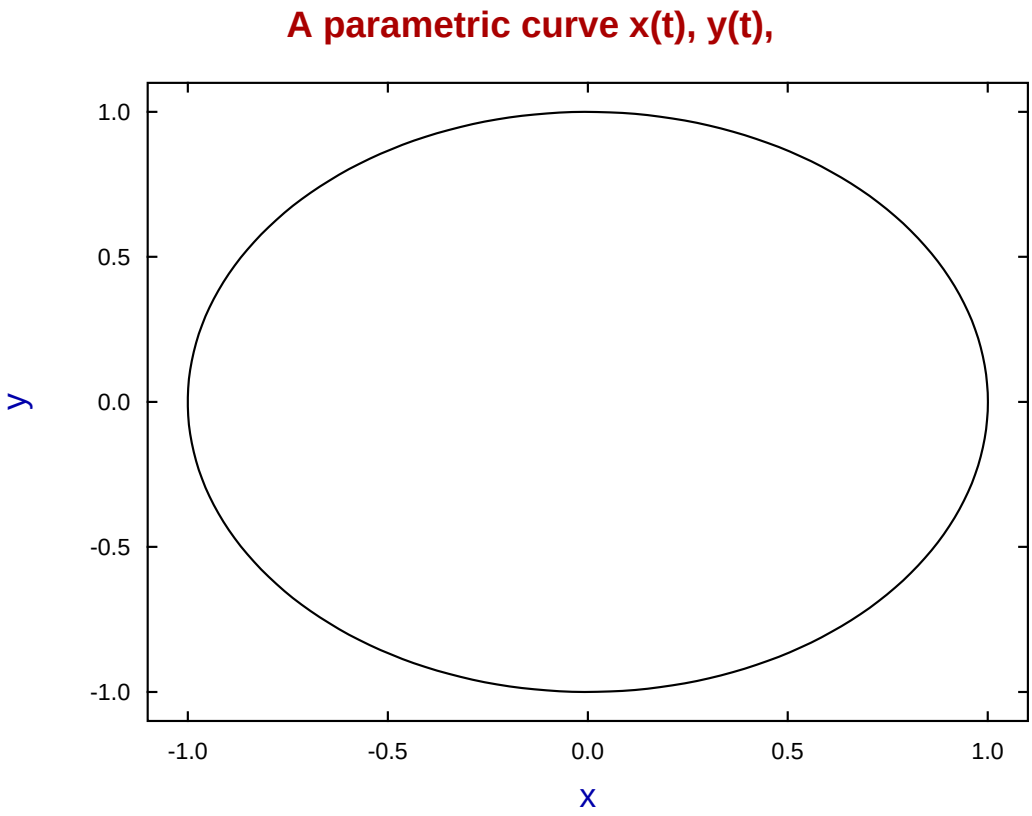
```
%
Example: Eight leaved rose
..............
r = A*sin(4*theta): where theta = x, r = f(1) and A = p(1)
..............
 %
 1 equation
 1 variable
 1 parameter
 %
 begin{expression}
 f(1) = p(1)sin(4x)
 end{expression}
 %
```



Example 2: The ellipse $x = A\cos(t), y = B\sin(t)$.

The following model file `ellipse_e.mod` and graph show how a SimFIT model file can be used to define multiple models.

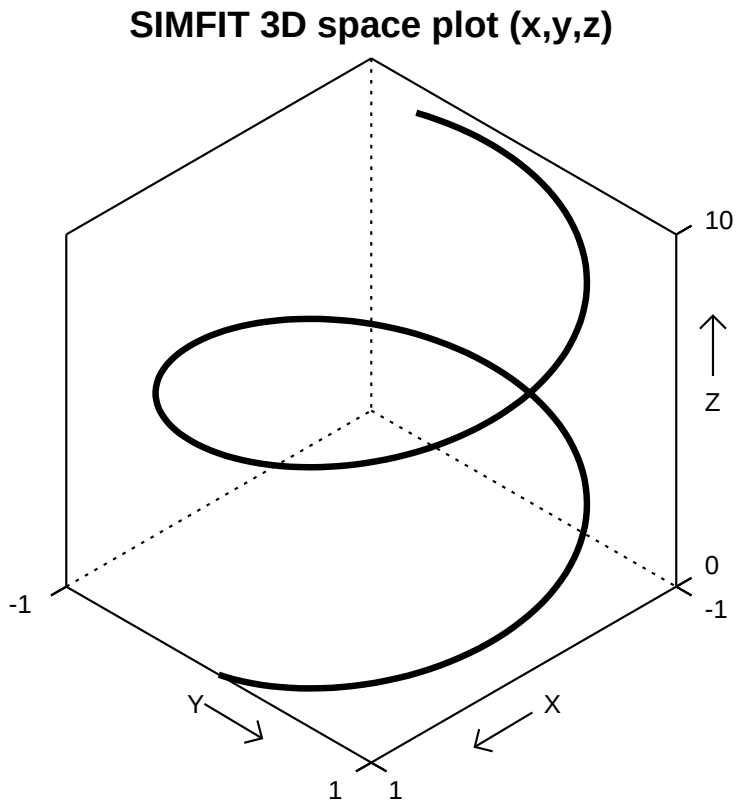
```
 %
Example: the ellipse
         X = A*cos(t), Y = B*sin(t)
         where: t = x, A = p(1), B = p(2)
                and X(t) = f(1), Y(t) = f(2)
 %
 2 equations
 1 variable
 2 parameters
 %
 begin{expression}
 f(1) = p(1)cos(x)
 f(2) = p(2)sin(x)
 end{expression}
 %
```



Example 3: The helix $x = A\cos(t), y = B\sin(t), z = Ct$

Model file `helix_e.mod` illustrates how the three coordinates for a space curve can be defined and plotted.

```
 %
Example: the helix
         X = A*cos(t), Y = B*sin(t), Z = C*t
         where: t = x, A = p(1), B = p(2), C = p(3)
                and X(t) = f(1), Y(t) = f(2), Z(t) = f(3)
 %
 3 equations
 1 variable
 3 parameters
 %
 begin{expression}
 f(1) = p(1)cos(x)
 f(2) = p(2)sin(x)
 f(3) = p(3)x
 end{expression}
 %
```



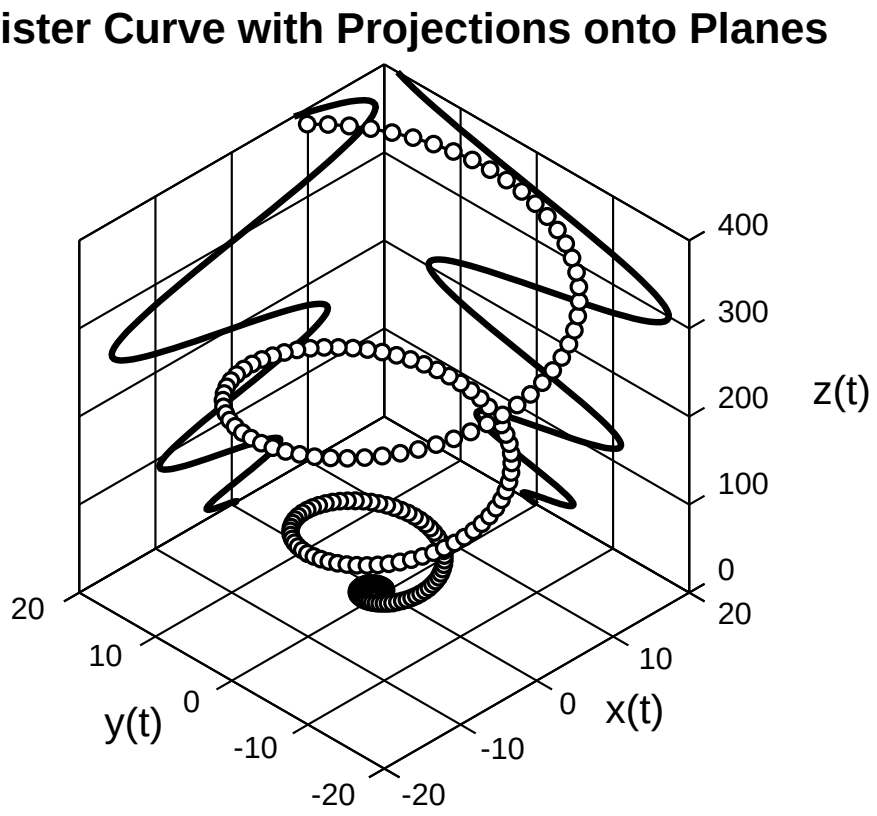
Example 4: Projecting space curves onto planes

Sometimes it is useful to project space curves onto planes for purposes of illustration. The next figure shows a simulation using **usermod** with the model file `twister.mod`.

The parametric equations are

$$x = t\cos t, y = t\sin t, z = t^2$$

and projections are created by fixing one the variables to a constant value.



Note the following about the model file `twister.mod`.

- There are 3 curves so there are 9 functions of 1 variable
- The value of x supplied is used as the parameter t
- Functions $f(1), f(4), f(7)$ are the $x(t)$ profiles
- Functions $f(2), f(5), f(8)$ are the $y(t)$ profiles
- Functions $f(3), f(6), f(9)$ are the $z(t)$ profiles

Also observe that the model parameters fix the values of the projection planes just outside the data range, at

$$p(1) = 20, p(2) = 20.$$

Example 5: Two dimensional families of curves

Users may need to plot families of curves indexed by parameters. For instance, diffusion of a substance from an instantaneous plane source is described by the equation

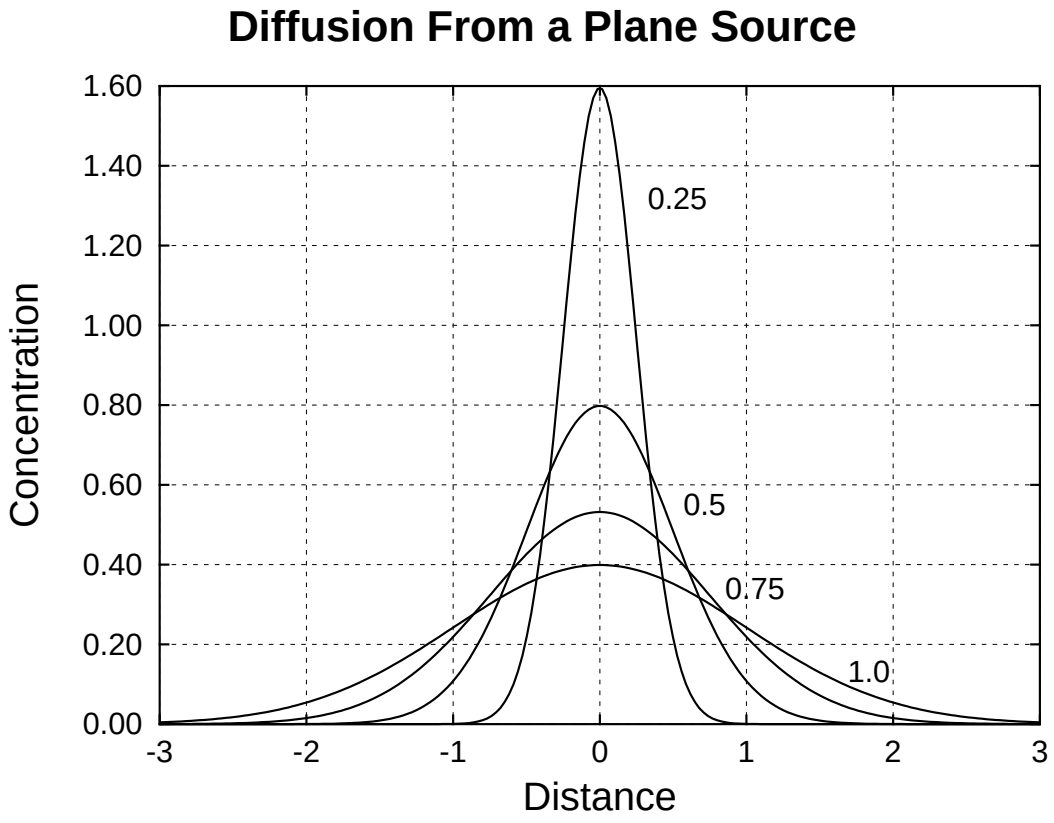
$$f(x) = \frac{1}{2\sqrt{\pi Dt}} \exp\left(\frac{-x^2}{4Dt}\right)$$

which is, of course, a normal distribution with $\mu = 0$ and $\sigma^2 = 2Dt$, where D is the diffusion constant and t is time, so that $2Dt$ is the mean square distance diffused by molecules in time t. Now it is easy to plot the concentration $f(x)$ predicted by this equation as a function of distance x and time t given a diffusion constant D, by simulating the equation using **makdat** saving the curves to a library file or project archive, then plotting the collected curves. However, there is a much better way using program **usermod** which has the important advantage that families of curves indexed by parameters can be plotted interactively. This is a more powerful technique which provides numerous advantages and convenient options when simulating systems to observe the behavior of the profiles as the indexing parameters vary.

The next figure shows the above equation plotted (in arbitrary units) using the model parameters

$$p_i = 2Dt_i, \text{ for } i = 1, 2, 3, 4$$

to display the diffusion profiles as a function of time. The plot was created using the model file `family2d.mod`, which simply defines four identical equations corresponding to the diffusion equation but with four different parameters p_i. Program **usermod** was then used to read in the model, simulate it for the parameter values indicated, then plot the curves simultaneously.

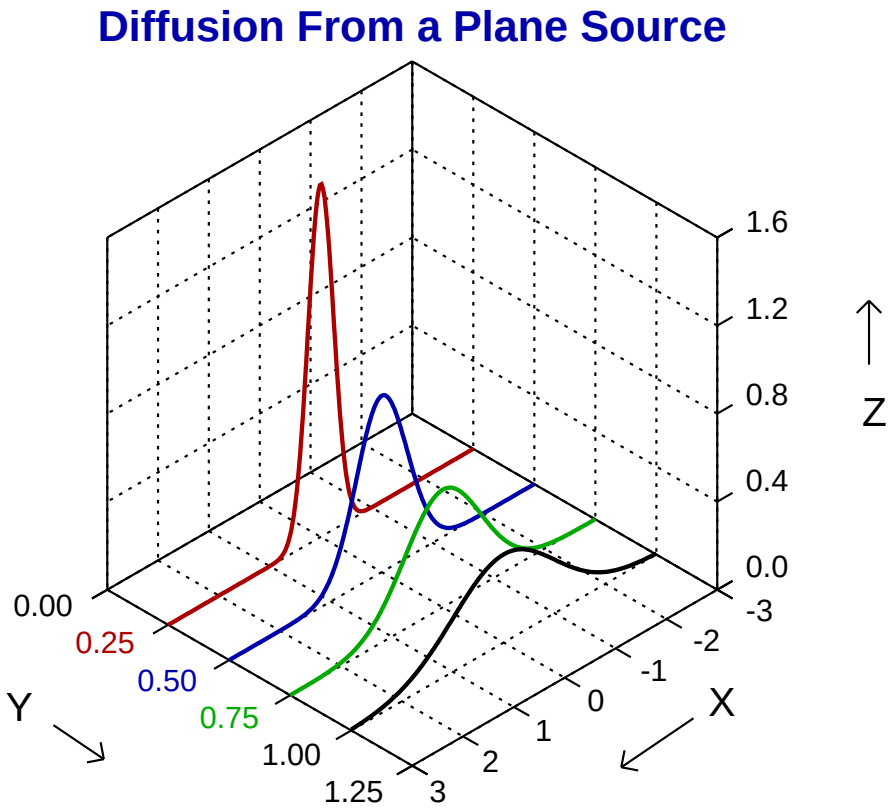


Example 6: Three dimensional families of curves

Users may need to plot families of curves indexed by parameters in three dimensions. To show how this is done, the diffusion equation dealt with previously in Example 5 is reformulated, using $y = \sqrt{2Dt}$, as

$$z(x, y) = \frac{1}{y\sqrt{2\pi}} \exp\left\{-\frac{1}{2}\left(\frac{x}{y}\right)^2\right\}$$

and is plotted in the next figure for the same parameter values used before, but now as sections through the surface of a function of two variables.



This is, of course, a case of a family of parametric space curves projected onto the fixed values of y. Now the model file `family3d.mod` was used by program **usermod** to create this figure, using the option to plot n sets of parametric space curves, but you should observe a number of important facts about this model file before attempting to plot your own families of space curves.

- There are 4 curves so there are 12 functions of 1 variable
- Functions $f(1), f(4), f(7), f(10)$ are the parameter t, i.e. x
- Functions $f(2), f(5), f(8), f(11)$ are the y values, i.e. $\sqrt{2Dt}$
- Functions $f(3), f(6), f(9), f(12)$ are the z values, i.e. the concentration profiles

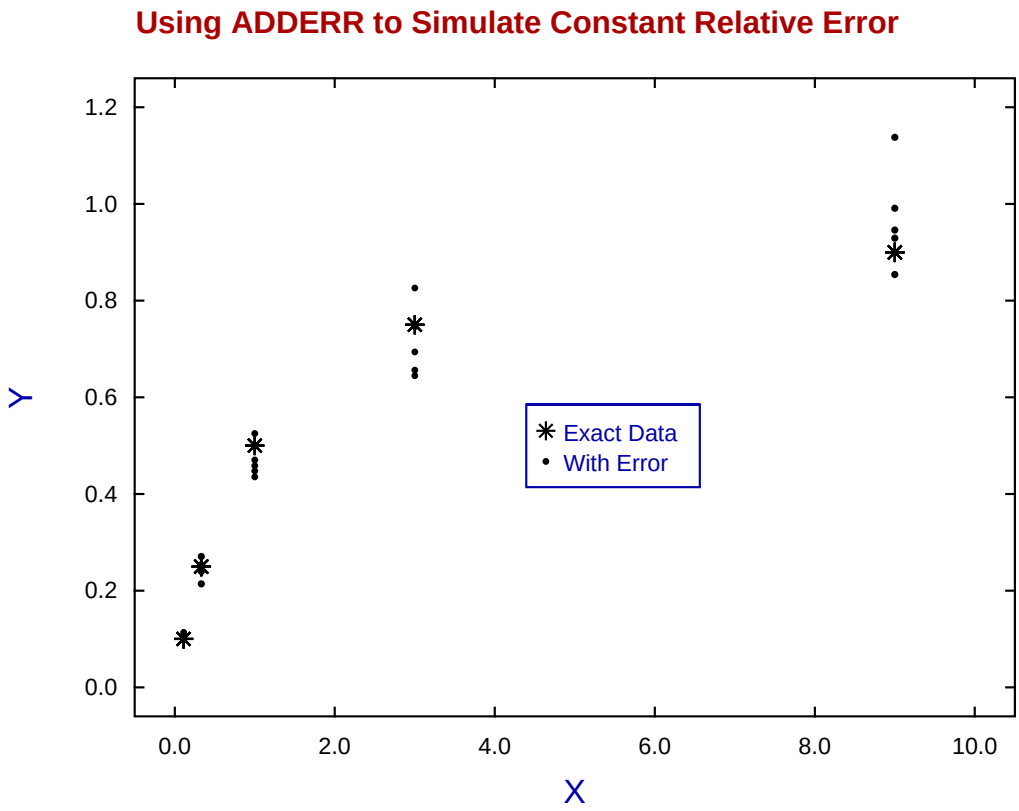
Finally, it is clear that n space curves require a model file that specifies $3n$ equations, but you should also realize that space curves cannot be plotted if there is insufficient variation in any of the independent variables, e.g. if all $y = k$, for some fixed parameter k.

10.3.7 Simulating a user-defined model: adding experimental error

To check the conclusions from model fitting it is advisable to simulate the model of interest, say $y = f(x)$, then study the robustness of parameter estimation and model discrimination with data simulated according to the assumed experimental design and variance model $V(y)$.

Example 1: Constant relative error, $V(y) = (\%y)^2$

From the main SIMFIT menu choose [A/Z], open program **adderr**, then read in the test file `mmfit.tf1`. This has exact data for the Michaelis-Menten model generated by program **makdat**. After choosing the defaults to add five replicates with 7.5% constant relative error the following plot resulted.



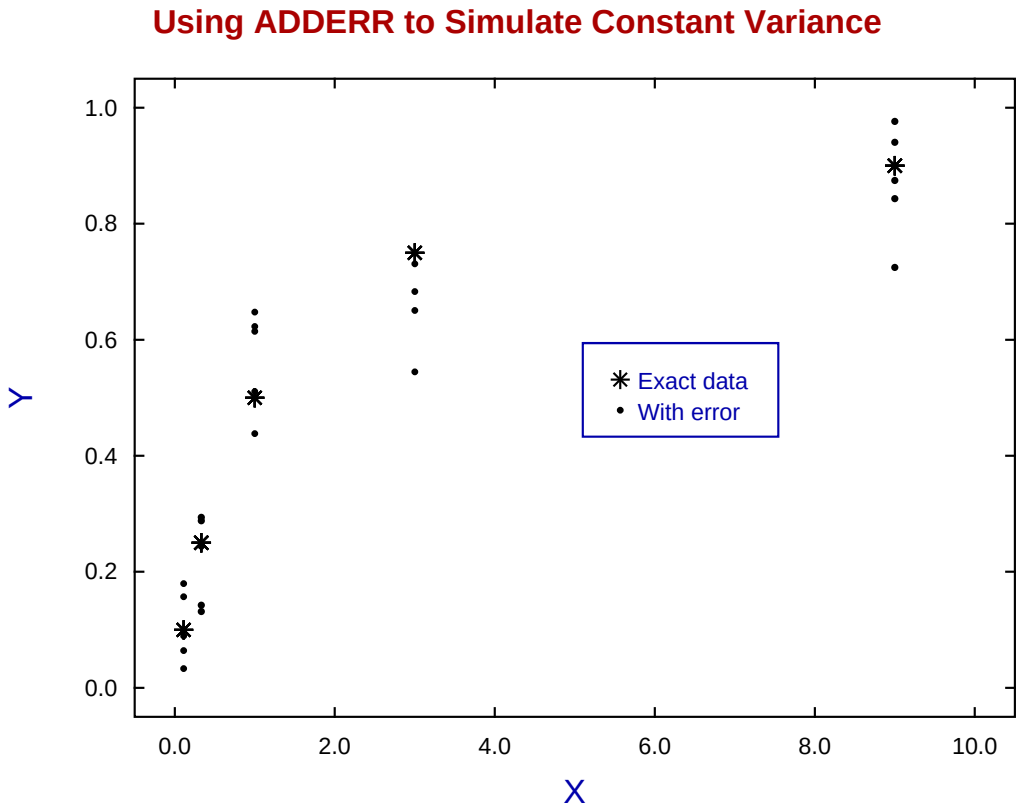
Constant relative error assumes that the observations O_i have experimental error which is a normal variate E_i with mean zero, and standard deviation equal to a percentage of the absolute true value $T_i = y(x_i)$, i.e.

$$\begin{aligned} O_i &= T_i + E_i \\ E_i &\approx N(\mu, \sigma_i^2) \\ \mu &= 0 \\ \sigma_i &= \%|T_i| \end{aligned}$$

Although this choice attempts to represent the reality that experimental error is proportional to response, it does tend to underestimate the error at low response so that the fitting may be dominated by small observations.

Example 2: Constant variance, $V(y) = k^2$

Note that program **adderr** can write a simulated data set to file containing the replicates along with weighting factors appropriate for the error type assumed. Actually most curve fitting is unweighted, that is with all weighting factors equal to one, which corresponds to assuming constant variance. The next plot illustrates adding error of the constant variance type to the exact data.



Here the observations O_i are assumed to result from adding a normal variate E_i with mean equal to zero and constant variance to the true value $T_i = y(x_i)$, i.e.

$$
\begin{aligned}
O_i &= T_i + E_i \\
E_i &\approx N(\mu, \sigma^2) \\
\mu &= 0 \\
\sigma &= k, \text{ a fixed positive constant}
\end{aligned}
$$

It will be seen from this plot that assuming constant variance presumes that all data points are of equal importance, which tends to bias towards a fit dominated by the largest observations.

In order to fit a model to data it is necessary to understand the type of weighting required, and this can only be known by performing replicate observations in order to discover the best statistical model for the error variance. Program **adderr** provides many options for simulating experimental error as will now be summarized.

Theory

Weighting curve fitting data

Curve fitting is undertaken when n observations y_i have been made in the belief that the true model is known to be a function of independent variable(s) x and some unknown parameters Θ, but contaminated by experimental or observational error ϵ_i. The intention is to obtain meaningful estimates for the model parameters in order to interpret the observations in the light of some basic scientific principles.

A common approach is to appeal to the principle of maximum likelihood and assume that the observations are the sum of a deterministic model plus random error which is normally distributed with mean zero and standard deviation σ_i as in

$$y_i = f(x_i, \Theta) + \epsilon_i$$
$$\epsilon_i \approx N(0, \sigma_i^2).$$

In this case the maximum likelihood estimate for the parameters Θ is obtained by minimizing the weighted sum of squares $WSSQ$ given by

$$WSSQ = \sum_{i=1}^{n} \left(\frac{y_i - f(x_i, \Theta)}{\sigma_i} \right)^2.$$

This poses a serious problem since the true standard deviations σ_i can never be known and must be replaced by some estimates. The aim of simulating experimental error is to examine the way that the accuracy with which the parameters Θ can be estimated depends on the choice of approximations s_i to the σ_i.

There are essentially four ways to choose s_i values.

1. Set all $s_i = 1$
   This is reasonable if the data are very noisy, or the range of values of the observations is relatively small.

2. Estimate s_i using replicates
   This reasonable if the number of replicates is sufficiently large (say at least 5) to make a sensible estimate for σ_i

3. Estimate $s_i = g(y_i)$
   This results in weights not being constant at fixed x.

4. Estimate $s_i = g(\hat{f}(x_i, \hat{\Theta}))$
   This results in weights not being constant at fixed x but has the added complication that the weights change at each iteration.

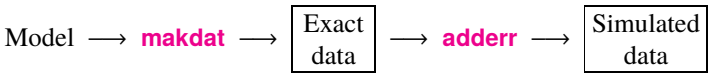
Method 1 assumes constant variance and uses the value of $WSSQ/NDOF$ at the solution point to estimate the variance, where $NDOF$ is the number of degrees of freedom, i.e. number of observations minus the number of parameters estimated. The other methods assume that the s_i supplied are proportional to the true σ_i, then use the value of $WSSQ/NDOF$ at the solution point to estimate the square of the proportionality factor. The choice of function $g(.)$ is often a variant of

$$g(t)^2 = A + Bt^{\lambda}$$

where A, B, and λ are either fixed or estimated from the data, and t can be taken as equal to the replicates or the best-fit function value.

Simulating experimental error

The output files from program **makdat** contain exact data for $y = f(x)$, which are then used to add random error to simulate experimental error. To do this, the output file then becomes an input file for program **adderr**. After adding random error, the input file is left unchanged and a new output file is produced as in this scheme.



There are numerous ways to use program **adderr**, including generating replicates. If in doubt, pick 7.5% constant relative error with 5 replicates, as this mimics many situations. Note: constant relative error cannot be used where $y = 0$ (which invokes a default value).

The options available with program **adderr** are as follows.

1. Single measurements: constant relative error
2. Single measurements: fixed constant variance
3. Single measurements: mixed power law error
4. Generate replicates: constant relative error
5. Generate replicates: fixed constant variance
6. Generate replicates: mixed power law error
7. Choose from selection of error distributions
8. Just add outliers to the data set supplied

In options 3 and 6 the model for the variance of observations is the mixed power law

$$V(y) = \sigma_0^2 + \sigma_1^2 y^2,$$

so that the error resembles constant variance white noise at low response levels with a transition to constant relative error at high response levels. Constant variance ($\sigma_1 = 0$) fails to account for the way variance always increases as the signal increases, while constant relative error ($\sigma_0 = 0$) exaggerates the importance of small response values. Note that using program **adderr** you can also simulate the effect of outliers or use a variety of error generating probability density functions, such as the Cauchy distribution which is a often a better model for experimental error.

When calculating variance $V(y)$ as a function of y it is possible to use the true y as the argument (option 1), or to use the mean of the simulated observations (option 2) in program **adderr**, but there are further points to be made.

- There are some experiments where the observations can never be negative, e.g. the size of a population or weight of a crop yield. Program **adderr** draws attention to such events and gives options that always result in positive simulated observations.
- Outliers are often encountered, that is observations that are not typical observations, and program **adderr** also allows this to be simulated. However many would argue that the best way to deal with outliers is for the experimentalists to view the scatter of observations then de-select extreme values as required.

11 Numerical analysis

Tutorials and worked examples for simulation, curve fitting, statistical analysis, and plotting.
https://simfit.uk
https://simfit.org.uk
https://simfit.silverfrost.com

11.1 Introduction

There are numerous occasions in data analysis when it is useful to be able to examine data sets using numerical techniques in order to perform data transformations, or make calculations relevant to a particular problem. Typical examples would be to perform a singular value decomposition on a data matrix to estimate the rank, to locate the positive zeros of the Hessian of a binding polynomial to identify ligand concentrations separating regions of positive and negative cooperativity, or to calculate the zeros of the derivative of a polynomial used for data smoothing or calibration so as to locate turning points.

The following techniques are available from the numerical calculations option available from the SimFIT program **simstat**.

Item	Action
Polynomial	Calculate zeros
Matrix	Determinant/eigenvalues/inverse
Matrix	Singular value decomposition
Matrix	Pseudo inverse and rank
Matrix	LU factorization/norms/condition number
Matrix	QR factorization
Matrix	Cholesky factorization of a positive definite matrix
Solve	$Ax = b$ (A nonsingular)
Solve	$Ax = b$ (L_1 norm over-determined)
Solve	$Ax = b$ (L_2 norm over-determined)
Solve	$Ax = b$ (L_∞ norm over-determined)
Calculate	$y^T Ay$, $y^T A^{-1} y$
Calculate	AB, $A^T B$, AB^T, $A^T B^T$
Calculate	$Ax = \lambda Bx$ (B positive definite)
Rotation	Orthomax
Rotation	Procrustes

In order to use one of these techniques the following procedure is recommended.

1. Open the [Statistics] option from the main SimFIT menu, or alternatively use the [A/Z] option and open program **simstat**
2. Select the [Numerical analysis] option
3. Choose the procedure required
4. Study the default test file(s) provided to appreciate the data format required
5. Perform the calculations and view the results
6. Now try with your own data

Note that many SimFIT test files have further information following on from the data section to help you understand the nature of the data set required, or to add information to assist in the interpretation of results.

11.2 Zeros of a polynomial

A polynomial $f(x)$ of degree $n-1$ has n coefficients, either p_i defined in ascending order of powers of x, or A_j defined in descending order of powers of x. It also has $n-1$ zeros, i.e. roots α_i satisfying $f(\alpha_i) = 0$, so it can be expressed in terms of monomials x^k for $k = 1, 2, \ldots, n-1$, or factored into linear terms $x - \alpha_i$ as follows

$$\begin{aligned} f(x) &= p_0 + p_1 x + p_2 x^2 + \cdots + p_{n-1} x^{n-1} \\ &= A_1 x^{n-1} + A_2 x^{n-2} + A_3 x^{n-3} + \cdots + A_n \\ &= p_{n-1}(x-\alpha_1)(x-\alpha_2)\ldots(x-\alpha_{n-1}). \end{aligned}$$

Numerical estimation of the zeros α_i given the coefficients can be done interactively by choosing [Statistics] from the main SimFIT menu, then [Numerical analysis] followed by selecting the options to estimate the zeros of a polynomial.

The default parameters are $A_i = i$ for $i = 1, 2, \ldots\ 5$ leading to the following results.

Zeros of $f(x) = A_1 x^{n-1} + A_2 x^{n-2} + \cdots + A_5$		
Coefficients	Real Part	Imaginary Part
A_1 = 1.0	-1.49179800	
A_2 = 2.0	-0.80578647	-1.2229047
A_3 = 3.0	-0.80578647	1.2229047
A_4 = 4.0	0.55168546	-1.2533489
A_5 = 5.0	0.55168546	1.2533489
A_6 = 6.0		

One of the problems in estimating the roots of polynomials is when either the real or the imaginary parts are close to zero. For instance with the difficult case

$$\begin{aligned} f(x) &= (x-1)^2(x^2+1) \\ &= x^4 - 2x^3 + 2x^2 - 2x + 1 \\ &= (x-1)(x-1)(x+i)(x-i) \end{aligned}$$

leading to the following results.

Zeros of $f(x) = A_1 x^{n-1} + A_2 x^{n-2} + \cdots + A_5$		
Coefficients	Real Part	Imaginary Part
A_1 = 1.0	1.0	
A_2 = -2.0		-1.0
A_3 = 2.0		1.0
A_4 = -2.0	1.0	
A_5 = 1.0		

This example illustrates how the output from the SimFIT routine suppresses values for very small real or imaginary components to emphasize cases with pure real roots and pure nonreal roots. However, for completeness, full details of the estimated values for all real and imaginary parts are written to the results file.

11.3 Determinant, inverse, eigenvalues, and eigenvectors of a matrix

It is often useful to examine the determinant, inverse, eigenvalues, and eigenvectors of a nonsingular square matrix. To do this from the main SIMFIT menu choose [Statistics] then [Numerical analysis] and select the option to calculate the determinant, etc.

The default test file `matrix.tf1` contains an arbitrary 5 by 5 matrix A as follows.

Matrix A

1.20	4.50	6.10	7.20	8.00
3.00	5.60	3.70	9.10	12.5
17.1	23.4	5.50	9.20	3.30
7.15	5.87	9.94	8.82	10.8
12.4	4.30	7.70	8.95	1.60

Analysis of matrix A then yielded the following results for the determinant and inverse.

The estimated determinant = 4.4833699E+04

The estimated inverse matrix A^{-1}

-2.4109851E-01	6.2912193E-02	4.4392110E-04	1.0122985E-01	2.9773962E-02
8.5852748E-02	-4.4069117E-02	5.2547695E-02	-1.9963429E-02	-5.8600243E-02
1.1817858E-01	-1.7354437E-01	-5.5370008E-03	1.1957082E-01	-3.0760472E-02
2.2291097E-01	6.7828265E-02	-1.9731089E-02	-2.5804239E-01	1.3801828E-01
-1.7785844E-01	8.6634195E-02	-7.6447234E-03	1.3711034E-01	-7.2265052E-02

The eigenvalues and eigenvectors of matrix A were also estimated as shown next.

Eigenvalues	Real Part	Imaginary Part
	3.8861300E+01	0.0000000E+00
	-8.3436467E+00	0.0000000E+00
	-2.7508195E+00	7.2563904E+00
	-2.7508195E+00	-7.2563904E+00
	-2.2960146E+00	0.0000000E+00

Eigenvector columns (real parts only)

-3.1942365E-01	-3.4409440E-01	-1.3613072E-01	-1.3613072E-01	-3.5397608E-01
-3.7703253E-01	-7.1957517E-02	-5.0496235E-02	-5.0496235E-02	6.2281655E-02
-6.0200219E-01	7.8212108E-01	8.0288388E-01	8.0288388E-01	-1.3074000E-01
-4.8975712E-01	-4.4618971E-01	-2.6270288E-01	-2.6270288E-01	7.8507480E-01
-3.9184988E-01	2.5616894E-01	-2.1155646E-01	-2.1155646E-01	-4.8722330E-01

Eigenvector columns (imaginary parts only)

0.0000000E+00	0.0000000E+00	-7.5604855E-02	7.5604855E-02	0.0000000E+00
0.0000000E+00	0.0000000E+00	3.9888409E-01	-3.9888409E-01	0.0000000E+00
0.0000000E+00	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.0000000E+00
0.0000000E+00	0.0000000E+00	-1.9106206E-01	1.9106206E-01	0.0000000E+00
0.0000000E+00	0.0000000E+00	-1.3855601E-01	1.3855601E-01	0.0000000E+00

Error messages are output if the matrix supplied for analysis is not square or appears to be singular.

11.4 Singular value decomposition of a matrix (SVD)

The singular value decomposition of a matrix (SVD) is one of the most useful techniques employed in the statistical analysis of data matrices, and should always be used to calculate the rank when data sets appear difficult to analyze unambiguously due to singularity.

A m by n matrix A can always be factored as

$$A = U\Sigma V^T,$$

where U and V are orthogonal, and $\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_r)$, $r = \min(m, n)$, with $\sigma_1 \geq \cdots \geq \sigma_r \geq 0$. The σ_i are the singular values, the first r columns of V are the right singular vectors, and the first r columns of U are the left singular vectors.

Example 1: m > n

From the main SIMFIT menu choose [Statistics], [Numerical analysis], then the singular value decomposition and analyze data contained in the default test file `f08kff.tf1` to obtain the following results for a case with $m > n$.

Current matrix: SVD data for `f08kff.tf1` with rank = 4

-5.7000000E-01	-1.2800000E+00	-3.9000000E-01	2.5000000E-01
-1.9300000E+00	1.0800000E+00	-3.1000000E-01	-2.1400000E+00
2.3000000E+00	2.4000000E-01	4.0000000E-01	-3.5000000E-01
-1.9300000E+00	6.4000000E-01	-6.6000000E-01	8.0000000E-02
1.5000000E-01	3.0000000E-01	1.5000000E-01	-2.1300000E+00
-2.0000000E-02	1.0300000E+00	-1.4300000E+00	5.0000000E-01

Index	σ_i	Fraction	Cumulative	σ_i^2	Fraction	Cumulative
1	3.9987197E+00	0.4000	0.4000	1.59898E+01	0.5334	0.5334
2	3.0005164E+00	0.3002	0.7002	9.00310E+00	0.3003	0.8337
3	1.9967125E+00	0.1998	0.9000	3.98686E+00	0.1330	0.9666
4	9.9994082E-01	0.1000	1.0000	9.99882E-01	0.0334	1.0000

Right singular vectors by row (V-transpose)

8.2514556E-01	-2.7935888E-01	2.0479917E-01	4.4626307E-01
-4.5304486E-01	-2.1212912E-01	-2.6220881E-01	8.2522611E-01
-2.8285271E-01	-7.9609569E-01	4.9515860E-01	-2.0259308E-01
1.8406389E-01	-4.9314451E-01	-8.0257199E-01	-2.8072616E-01

Left singular vectors by column (U)

-2.0271367E-02	2.7939484E-01	4.6900514E-01	7.6917561E-01
-7.2841546E-01	-3.4641438E-01	-1.6941649E-02	-3.8290338E-02
4.3926966E-01	-4.9545699E-01	-2.8679802E-01	8.2222482E-02
-4.6784650E-01	3.2584052E-01	-1.5355623E-01	-1.6362605E-01
-2.2003450E-01	-6.4277549E-01	1.1245506E-01	3.5724829E-01
-9.3523390E-02	1.9268002E-01	-8.1318411E-01	4.9572409E-01

Example 2: m < n

The test file `f08kff.tf2` has $m < n$ and yields the following results.

Current matrix: SVD data for `f08kff.tf2` with rank = 4

-5.4200000E+00	3.2800000E+00	-3.6800000E+00	2.7000000E-01	2.0600000E+00	4.6000000E-01
-1.6500000E+00	-3.4000000E+00	-3.2000000E+00	-1.0300000E+00	-4.0600000E+00	-1.0000000E-02
-3.7000000E-01	2.3500000E+00	1.9000000E+00	4.3100000E+00	-1.7600000E+00	1.1300000E+00
-3.1500000E+00	-1.1000000E-01	1.9900000E+00	-2.7000000E+00	2.6000000E-01	4.5000000E+00

Index	σ_i	Fraction	Cumulative	σ_i^2	Fraction	Cumulative
1	7.9987291E+00	0.3077	0.3077	6.39797E+01	0.3677	0.3677
2	7.0059360E+00	0.2695	0.5771	4.90831E+01	0.2821	0.6498
3	5.9952457E+00	0.2306	0.8077	3.59430E+01	0.2066	0.8564
4	4.9988922E+00	0.1923	1.0000	2.49889E+01	0.1436	1.0000

Right singular vectors by row (V-transpose)

-7.9334138E-01	3.1632760E-01	-3.3417094E-01	-1.5140353E-01	2.1421955E-01	3.0010505E-01
1.0023978E-01	6.4422154E-01	4.3706509E-01	4.8903426E-01	3.7714369E-01	5.0128185E-02
1.1079521E-02	1.7244335E-01	-6.3666698E-01	4.3538482E-01	-4.3019579E-02	-6.1105243E-01
2.3606094E-01	2.1558447E-02	-1.0247828E-01	-5.2856734E-01	7.4604456E-01	-3.1199799E-01

Left singular vectors by column (U)

8.8835057E-01	1.2751497E-01	4.3306768E-01	8.3818775E-02
7.3269096E-02	-8.2640879E-01	1.9433222E-01	-5.2336903E-01
-3.6065168E-02	5.4351983E-01	7.5594729E-02	-8.3520712E-01
4.5184534E-01	-7.3311896E-02	-8.7691095E-01	-1.4658902E-01

Fractions and Cumulatives

For an explanation of the tables of fractions and cumulatives displayed and plotted by SIMFIT the following expanded expression for the singular value decomposition of a m by n matrix A with $m > n$ should be noted

$$A = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \cdots + \sigma_n u_n v_n^T,$$

where u_i are the columns of U, v_i are the columns of V and n is the rank of A. If a m by n matrix B of rank $k < n$ is required which best approximates A in the least squares sense we need to minimize the sum of squares $S = (A - B)(A - B)^T$, i.e.

$$S = \sum_{i=1}^{m} \sum_{j=1}^{n} (a_{ij} - b_{ij})^2,$$

and this is satisfied by

$$B = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \cdots + \sigma_k u_k v_k^T.$$

It follows from this result that, if A is a mean-centered m by n matrix of rank n with $m > n$, then the expression

$$f_k = \frac{\sigma_1^2 + \cdots + \sigma_k^2}{\sigma_1^2 + \cdots + \sigma_n^2}$$

is the fraction of the total variance of A accounted for by approximating A by a matrix B of rank k formed by evaluating the usual expression for the SVD for matrix A but with all $\sigma_i = 0$ for $i > k$.

11.5 Pseudo-inverse (or generalized inverse) of a matrix

The pseudo-inverse (or generalized inverse) of a matrix satisfies many of the properties of the usual inverse in situations when the matrix itself is not square and of full rank. It is encountered in the solution of many rank-deficient data analysis situations, and is best calculated using the singular value decomposition (SVD).

From the main SIMFIT menu choose [Statistics], then [Numerical analysis], and select the pseudo-inverse option which provides the default test file `f01blf.tf1` containing the following matrix.

7	-2	4	9	1.8
3	8	-4	6	1.3
9	6	1	5	2.1
-8	7	5	2	0.6
4	-1	2	8	1.3
1	6	3	-5	0.5

Using the tolerance factor set at $TOL = 1.0E^{-7}$, the rank of this matrix was estimated to be 4 and the following matrix was calculated as the pseudo-inverse.

1.7807132E-02	-1.1826257E-02	4.7156796E-02	-5.6636340E-02	-3.6741218E-03	3.8408070E-02
-2.1564769E-02	4.3417257E-02	2.9445712E-02	2.9132145E-02	-1.3781035E-02	3.4256117E-02
5.2028568E-02	-8.1265321E-02	1.3926152E-02	4.7441829E-02	1.6646584E-02	5.7593532E-02
2.3686052E-02	3.5716849E-02	-1.3808338E-02	3.0477616E-02	3.5665495E-02	-5.7134309E-02
7.1956983E-03	-1.3957472E-03	7.6720321E-03	5.0415250E-03	3.4856923E-03	7.3123409E-03

The estimation of rank is based on the TOL value. The singular values are calculated in descending order and when a value, say the k'th, becomes less than or equal to TOL multiplied by the first (i.e. largest) singular value, than all subsequent singular values are set to zero and the rank is recorded as $k-1$.

Given the singular value decomposition of a m by n matrix A of rank r

$$A = U\Sigma V^T$$

where $m \geq n \geq r$ the pseudo-inverse is the n by m matrix A^+ satisfying these properties.

1. $AA^+A = A$
2. $A^+AA^+ = A^+$
3. $(A^+A)^T = A^+A$
4. $(AA^+)^T = AA^+$
5. $A^+ = V\Lambda U^T$

If the singular values are $\sigma_1 \geq \ldots \geq \sigma_r > 0$ and $\sigma_i = 0$ for $i > r$, then Λ is found to be

$$\Lambda = \begin{pmatrix} 1/\sigma_1 & 0 & 0 & \ldots & \ldots & \ldots & 0 \\ 0 & 1/\sigma_2 & 0 & \ldots & \ldots & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \ldots & \ldots & \ldots & \ldots & 1/\sigma_r & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & \ldots & \ldots & 0 \end{pmatrix}.$$

11.6 LU factorization, norms, and condition numbers of a matrix

The LU factorization of a matrix, the matrix norms, and condition numbers of a matrix are of considerable value in solving non-singular linear systems.

From the main SIMFIT menu choose [Statistics], then [Numerical analysis], and select the option to calculate the LU decomposition of a matrix, noting that the following arbitrary 5 by 5 matrix A is contained in the default test file `matrix.tf1`.

Matrix A

1.20	4.50	6.10	7.20	8.00
3.00	5.60	3.70	9.10	12.5
17.1	23.4	5.50	9.20	3.30
7.15	5.87	9.94	8.82	10.8
12.4	4.30	7.70	8.95	1.60

Proceeding to the analysis of this matrix yields the following results.

LU factorization of matrix A

Matrix 1-norm = 43.67, Condition number = 3.6940420E+01
Matrix ∞-norm = 58.50, Condition number = 2.6184088E+01

Lower triangular/trapezoidal L where $A = PLU$

1				
7.2514620E-01	1			
7.0175439E-02	-2.2559202E-01	1		
1.7543860E-01	-1.1798920E-01	4.8433085E-01	1	
4.1812865E-01	3.0897383E-01	9.9116389E-01	-6.3185748E-01	1

Upper triangular/trapezoidal U where $A = PLU$

1.7100000E+01	2.3400000E+01	5.5000000E+00	9.2000000E+00	3.3000000E+00
	-1.2668421E+01	3.7116959E+00	2.2786550E+00	-7.9298246E-01
		6.5513641E+00	7.0684324E+00	7.5895305E+00
			4.3313617E+00	8.1516455E+00
				7.2933958E+00

This calculation produces the L and U factors after the standard pivoting operations, familiar in Gaussian elimination, so that the matrix equation is not simply $A = LU$ but

$$A = PLU$$

where P is a permutation matrix. To appreciate this fact, note that multiplying L and U together does not give matrix A directly, but yields the following matrix instead with rows of A permuted.

Matrix LU

17.1	23.4	5.50	9.20	3.30
12.4	4.30	7.70	8.95	1.60
1.20	4.50	6.10	7.20	8.00
3.00	5.60	3.70	9.10	12.5
7.15	5.87	9.94	8.82	10.8

So it is convenient to consider the sequence of pivots applied to matrix A which are contained in the vector

$$IPIV = (3, 5, 3, 5, 5).$$

This is the sequence of row exchanges applied to matrix A in order to rearrange it as the calculation proceeds. That is, row i of a m by n matrix is interchanged with row $IPIV(i)$ for $i = 1, 2, \ldots, \min(m, n)$. Clearly matrix A was transformed into LU by these pivots, which can therefore be used to calculate the permutation matrix P needed to represent A directly in terms of L and U if required.

As the *LU* representation is of interest in the solution of linear equations, this procedure also calculates the matrix norms and condition numbers needed to assess the sensitivity of the solutions to perturbations when the matrix is square. Given a vector norm $\|.\|$, a matrix A, and the set of vectors x where $\|x\| = 1$, the matrix norm subordinate to the vector norm is

$$\|A\| = \max_{\|x\|=1} \|Ax\|.$$

For a m by n matrix A, the three most important norms are

$$\begin{aligned}
\|A\|_1 &= \max_{1\le j\le n} \left(\sum_{i=1}^{m} |a_{ij}|\right) \\
\|A\|_2 &= (\lambda_{\max}|A^T A|)^{\frac{1}{2}} \\
\|A\|_\infty &= \max_{1\le i\le m} \left(\sum_{j=1}^{n} |a_{ij}|\right),
\end{aligned}$$

so that the 1-norm is the maximum absolute column sum, the 2-norm is the square root of the largest eigenvalue of $A^T A$, and the infinity norm is the maximum absolute row sum. The condition numbers estimated are

$$\begin{aligned}
\kappa_1(A) &= \|A\|_1 \|A^{-1}\|_1 \\
\kappa_\infty(A) &= \|A\|_\infty \|A^{-1}\|_\infty \\
&= \kappa_1(A^T)
\end{aligned}$$

which satisfy $\kappa_1 \ge 1$, and $\kappa_\infty \ge 1$ and they are included in the tabulated output unless A is in singular, when they are infinite. For a perturbation δb to the right hand side of a linear system with $m = n$ we have

$$\begin{aligned}
Ax &= b \\
A(x + \delta x) &= b + \delta b \\
\frac{\|\delta x\|}{\|x\|} &\le \kappa(A) \frac{\|\delta b\|}{\|b\|},
\end{aligned}$$

while a perturbation δA to the matrix A leads to

$$\begin{aligned}
(A + \delta A)(x + \delta x) &= b \\
\frac{\|\delta x\|}{\|x + \delta x\|} &\le \kappa(A) \frac{\|\delta A\|}{\|A\|},
\end{aligned}$$

and, for complete generality,

$$\begin{aligned}
(A + \delta A)(x + \delta x) &= b + \delta b \\
\frac{\|\delta x\|}{\|x\|} &\le \frac{\kappa(A)}{1 - \kappa(A)\|\delta A\|/\|A\|} \left(\frac{\|\delta A\|}{\|A\|} + \frac{\|\delta b\|}{\|b\|}\right)
\end{aligned}$$

provided $\kappa(A)\|\delta A\|/\|A\| < 1$. These inequalities estimate bounds for the relative error in computed solutions of linear equations, so that a small condition number indicates a well-conditioned problem, a large condition number indicates an ill-conditioned problem, while an infinite condition number indicates a singular matrix and no solution. To a rough approximation; if the condition number is 10^k and computation involves n-digit precision, then the computed solution will have about $(n - k)$-digit precision.

11.7 QR factorization of a matrix

The QR factorization of a matrix is a widely used technique in data analysis, for instance when solving linear least squares problems.

From the main SIMFIT menu choose [Statistics], then [Numerical analysis], and select the option to calculate the QR decomposition of a matrix, noting that the following arbitrary 7 by 5 matrix A is contained in the default test file `matrix.tf2`.

Matrix A

1.20	3.60	1.90	8.50	3.20
4.70	8.85	9.91	2.50	8.06
6.34	8.12	5.56	3.45	7.76
3.65	7.78	3.48	1.15	6.67
3.32	8.83	4.46	7.82	4.49
3.61	7.82	1.08	5.22	6.38
6.12	5.51	8.03	5.61	4.43

Proceeding to the analysis of this matrix yields the following results.

QR factorization of matrix A

The orthogonal matrix Q_1

-1.0194525E-01	-2.5040780E-01	7.4979981E-02	7.3028426E-01	5.5733913E-01
-3.9928558E-01	-2.1648714E-01	7.2954175E-01	-2.9595704E-01	2.7393825E-01
-5.3861076E-01	2.6379623E-01	-3.2944854E-01	-1.4891526E-01	1.8269264E-01
-3.1008348E-01	-3.0017948E-01	-1.5219943E-01	-4.2044142E-01	6.5037721E-02
-2.8204853E-01	-5.2829344E-01	7.5783253E-02	2.7828157E-01	-6.9912752E-01
-3.0668530E-01	-3.1512480E-01	-5.5196151E-01	3.2818257E-02	1.4906187E-01
-5.1992079E-01	5.9357854E-01	1.4156702E-01	3.1879447E-01	-2.5637022E-01

The lower triangular/trapezoidal matrix R

-1.1771024E+01	-1.8440180E+01	-1.3988503E+01	-1.0802824E+01	-1.5318642E+01
	-6.8692395E+00	-1.2916776E-01	-4.5510241E+00	-4.2543105E+00
		5.8894908E+00	-3.4486568E-01	-5.7542482E-03
			8.6061685E+00	-1.1373065E+00
				2.5191361E+00

Note that the factorizing of a m by n matrix depends upon m and n as in

$$\begin{aligned} A &= QR \text{ when } m = n \\ &= Q_1 Q_2 \begin{pmatrix} R \\ 0 \end{pmatrix} \text{ when } m > n \\ &= Q(R_1 R_2) \text{ when } m < n, \end{aligned}$$

where Q is a m by m orthogonal matrix and R is either upper triangular or upper trapezoidal.
When $m \geq n$ then $A = Q_1 R$ and R is upper triangular.
When $m < n$ then R_1 is upper triangular and R_2 is rectangular.

You can display or write to file the matrices Q, Q_1, R, or R_1.

11.8 Cholesky factorization of a matrix

The Cholesky factorization of a positive definite symmetric matrix is widely used in data analysis for evaluation of quadratic forms and other calculations involving covariance matrices.

From the main SIMFIT menu choose [Statistics] followed by [Numerical analysis], and then open the Cholesky factorization procedure. The default test file is `matrix.tf3` and analysis yields the following results.

The current matrix A

4.16	-3.12	0.561	-0.10
-3.12	5.03	-0.83	1.09
0.56	-0.83	0.76	0.34
-0.10	1.09	0.34	1.180

Lower triangular L where $A = LL^T$

2.0396078E+00			
-1.5297059E+00	1.6401219E+00		
2.7456259E-01	-2.4998141E-01	7.8874881E-01	
-4.9029034E-02	6.1885642E-01	6.4426613E-01	6.1606334E-01

Upper triangular U where $A = U^T U$

2.0396078E+00	-1.5297059E+00	2.7456259E-01	-4.9029034E-02
	1.6401219E+00	-2.4998141E-01	6.1885642E-01
		7.8874881E-01	6.4426613E-01
			6.1606334E-01

Note that an error message will be issued if the matrix supplied is not square, or positive definite to within a tolerance factor.

Also note that there are two conventions used to define the Cholesky factors for a matrix A, i.e.

$$\begin{aligned} A &= LL^T \\ &= U^T U. \end{aligned}$$

You can display or write to file the matrices A, L, or U.

11.9 Matrix multiplication

Given two matrices A and B, it is frequently necessary to form the product, or the product of the transposes, as an m by n matrix C, where $m \geq 1$ and $n \geq 1$. The options are

$$C = AB, \text{ where } A \text{ is } m \times k, \text{ and } B \text{ is } k \times n,$$
$$C = A^T B, \text{ where } A \text{ is } k \times m, \text{ and } B \text{ is } k \times n,$$
$$C = AB^T, \text{ where } A \text{ is } m \times k, \text{ and } B \text{ is } n \times k,$$
$$C = A^T B^T, \text{ where } A \text{ is } k \times m, \text{ and } B \text{ is } n \times k,$$

as long as $k \geq 1$ and the dimensions of A and B are appropriate to form the product, as indicated.

From the main SIMFIT menu choose [Statistics] followed by [Numerical analysis], and then open the Cholesky factorization procedure and save the lower triangular matrix L to a file. Then use the matrix multiplication procedure from the SIMFIT numerical analysis options to form the product LL^T as shown below.

The current matrix A

4.16	-3.12	0.561	-0.10
-3.12	5.03	-0.83	1.09
0.56	-0.83	0.76	0.34
-0.10	1.09	0.34	1.180

Estimated lower triangular $\hat{L}$ where $A = LL^T$

2.0396078E+00			
-1.5297059E+00	1.6401219E+00		
2.7456259E-01	-2.4998141E-01	7.8874881E-01	
-4.9029034E-02	6.1885642E-01	6.4426613E-01	6.1606334E-01

Estimated product $\hat{A} = \hat{L}\hat{L}^T$

4.1600000E+00	-3.1200001E+00	5.6000000E-01	-1.0000000E-01
-3.1200001E+00	5.0300000E+00	-8.3000000E-01	1.0900000E+00
5.6000000E-01	-8.3000000E-01	7.6000001E-01	3.4000000E-01
-1.0000000E-01	1.0900000E+00	3.4000000E-01	1.1800000E+00

Another example using the singular value decomposition routine, followed by multiplying the calculated U, Σ, and V^T matrices for the simple 4 by 3 matrix indicated shows that, while for exact factors

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \end{pmatrix} = \begin{pmatrix} -1/\sqrt{6} & 0 & 1/\sqrt{2} \\ 0 & 1 & 0 \\ -1/\sqrt{6} & 0 & -1/\sqrt{2} \\ -2/\sqrt{6} & 0 & 0 \end{pmatrix} \begin{pmatrix} \sqrt{3} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} -1/\sqrt{2} & 0 & -1/\sqrt{2} \\ 0 & 1 & 0 \\ 1/\sqrt{2} & 0 & -1/\sqrt{2} \end{pmatrix},$$

singular value decomposition to yield the calculated factors $\hat{U}$, $\hat{\Sigma}$ and $\hat{V}^T$ followed by matrix multiplication leads to the following matrix.

The product matrix $\hat{A} = \hat{U}\hat{\Sigma}\hat{V}^T$

1.0000000E+00	-1.5700925E-16	1.2789252E-08
0.0000000E+00	1.0000000E+00	0.0000000E+00
1.2789252E-08	2.2204460E-16	1.0000000E+00
1.0000000E+00	1.5700925E-16	1.0000000E+00

Numbers colored red in the above results tables can be regarded as correct since any digits less than 10^{-7} are due to rounding error and can be taken as zero compared to 1.

11.10 Evaluation of quadratic forms

Quadratic forms often need to be evaluated in data analysis given a n by 1 vector x and a n by n matrix A. Frequently the inverse A^{-1} is required and it is convenient to be able to estimate this interactively.

For instance, in nonlinear optimization or multivariate statistics the following expressions for Q_1 and/or Q_2 are frequently required

$$Q_1 = x^T Ax$$
$$Q_2 = x^T A^{-1} x.$$

To evaluate such quadratic forms interactively, open [Statistics] then [Numerical analysis] from the main SIMFIT menu and select the option to evaluate quadratic forms which provides the default test files `matrix.tf3` defining matrix A and `vector.tf3` defining x as follows

$$A = \begin{pmatrix} 4.16 & -3.12 & 0.56 & -0.10 \\ -3.12 & 5.03 & -0.83 & 1.09 \\ 0.56 & -0.83 & 0.76 & 0.34 \\ -0.10 & 1.09 & 0.34 & 1.18 \end{pmatrix}$$

$$x = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4. \end{pmatrix}$$

The following table illustrates the output from running the SIMFIT procedure with these default test files.

Title of matrix A Test file matrix.tf3: 4 by 4 positive-definite symmetric matrix Title of vector x Test file vector.tf3: vector with components 1, 2, 3, 4
$x^T Ax$ = 55.72 $x^T A^{-1} x$ = 20.635258

Using the SIMFIT procedure to evaluate quadratic forms allows the matrix A and vector x to be changed but two facts must be clear.

1. The dimensions of A and x must be consistent, i.e. identical.
2. Calculation of Q_2 requires that A is nonsingular.

Of course, in many applications, as when estimating a Mahalanobis distance in multivariate statistics, it is also vital that the matrix A to be used is a symmetric positive definite matrix (e.g. a covariance matrix) and the vector x has a defined meaning (e.g. a difference vector) if the scalar results from such quadratic forms are to be interpreted correctly.

11.11 Solving exact linear equations Ax = b

Linear equations of the form $Ax = b$ can be solved uniquely only if the matrix is square and nonsingular.

Under such circumstances SIMFIT provides a procedure to solve the following system

$$Ax = b$$
$$x = A^{-1}b$$

to high accuracy, i.e. given a n by n full rank matrix A and a n by 1 vector b, to calculate an n by 1 vector x satisfying the above equations.

From the main SIMFIT menu choose [Statistics] then [Numerical analysis] and open the procedure to solve a full rank linear system. The default test files provided to demonstrate the procedure are `matrix.tf1` and `vector.tf1` containing the following data.

$$A = \begin{pmatrix} 1.20 & 4.50 & 6.10 & 7.20 & 8.00 \\ 3.00 & 5.60 & 3.70 & 9.10 & 12.5 \\ 17.1 & 23.4 & 5.50 & 9.20 & 3.30 \\ 7.15 & 5.87 & 9.94 & 8.82 & 10.8 \\ 12.4 & 4.30 & 7.70 & 8.95 & 1.60 \end{pmatrix}$$

$$b = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{pmatrix}$$

The following table will be output giving the results.

Solution to $Ax = b$ where the square matrix A is:
Test file `matrix.tf1`: arbitrary 5 by 5 matrix
and the vector b is:
Test file `vector.tf1`: Vector with components 1, 2, 3, 4, 5

RHS vector b	Solution x
1.0000000E+00	4.3984686E-01
2.0000000E+00	-2.1749733E-01
3.0000000E+00	7.8959766E-02
4.0000000E+00	-4.2703888E-02
5.0000000E+00	1.5959190E-01

The data set consisting of A and b can be varied interactively but error messages will be output under the following conditions.

1. A and b have inconsistent dimensions
2. A is singular
3. The LU factorization failed
4. The system is very ill-conditioned

Under these circumstances a meaningful unique solution cannot be obtained although various other approaches using the pseudo inverse or other techniques may be used to obtain more insight.

11.12 Solving overdetermined linear equations Ax = b

Overdetermined linear equations of the form $Ax = b$, where the number of rows of matrix A exceeds the number of columns, can often be solved by optimization techniques, although solutions may not be unique.

Such a linear system consisting of a m by n matrix A where $m > n$, and a m by 1 vector b as in $Ax = b$ cannot be solved uniquely, but often solutions can be found by minimizing some L_p norm of the residuals r_i such as

$$L_p = \left(\sum_{i=1}^{m} |r_i|^p \right)^{1/p}$$

where typically p can be 1, 2, or ∞. In some cases starting estimates will be required.

From the main SIMFIT menu choose [Statistics] then [Numerical analysis] and run the three options for p using the default test files `matrix.tf2` defining the 7 by 5 matrix A and `vector.tf2` containing the 7 by 1 vector $b^T = (1, 2, 3, 4, 5, 6, 7)$ as follows.

$$A = \begin{pmatrix} 1.20 & 3.60 & 1.90 & 8.50 & 3.20 \\ 4.70 & 8.85 & 9.91 & 2.50 & 8.06 \\ 6.34 & 8.12 & 5.56 & 3.45 & 7.76 \\ 3.65 & 7.78 & 3.48 & 1.15 & 6.67 \\ 3.32 & 8.83 & 4.46 & 7.82 & 4.49 \\ 3.61 & 7.82 & 1.08 & 5.22 & 6.38 \\ 6.12 & 5.51 & 8.03 & 5.61 & 4.43 \end{pmatrix}$$

L_1 norm solution to $Ax = b$
1.9514418E+00
4.2111129E-01
-5.6336298E-01
4.3037848E-02
-6.7286341E-01
objective function = 4.9251750E+00

L_2 norm solution to $Ax = b$
1.2955430E+00
7.7602676E-01
-3.3656942E-01
8.2383926E-02
-9.8542254E-01
The rank of A (from SVD) = 5
objective function = 1.0961673E+01

L_∞ norm solution to $Ax = b$
1.0529866E+00
7.4896175E-01
-2.7683128E-01
2.6138630E-01
-9.7904715E-01
objective function = 1.5226995E+00

11.13 Solving symmetric eigenvalue problems

Symmetric eigenvalue problems of the form $Ax = \lambda Bx$ can be solved uniquely if A and B are symmetric and B is positive definite, as long as appropriate scaling conventions are understood.

From the SIMFIT main menu choose [Statistics] then [Numerical analysis] and open the procedure to solve symmetric eigenvalue problems. From this control you are given the options to solve any of the following three problems.

$$Ax = \lambda Bx$$
$$ABx = \lambda x$$
$$BAx = \lambda x$$

The SIMFIT default test files are `matrix.tf4` containing matrix A, and `matrix.tf3` containing matrix B as now displayed.

Matrix A

0.24	0.39	0.42	-0.16
0.39	-0.11	0.79	0.63
0.42	0.79	-0.25	0.48
-0.16	0.63	0.48	-0.03

Matrix B

4.16	-3.12	0.56	-0.10
-3.12	5.03	-0.83	1.09
0.56	-0.83	0.76	0.34
-0.10	1.09	0.34	1.18

The results from analyzing the standard problem $Ax = \lambda Bx$ are then as follows.

Eigenvalues...Case: $Ax = \lambda Bx$

-2.2254476E+00
-4.5475588E-01
1.0007648E-01
1.1270387E+00

Eigenvectors by column ...Case: $Ax = \lambda Bx$

-6.9005765E-02	3.0795498E-01	-4.4694499E-01	-5.5278790E-01
-5.7401486E-01	5.3285741E-01	-3.7084023E-02	-6.7660179E-01
-1.5427579E+00	-3.4964452E-01	5.0476980E-02	-9.2759211E-01
1.4004070E+00	-6.2110938E-01	4.7425180E-01	2.5095480E-01

It should be noted that the eigenvectors are the columns of a matrix X that is normalized so that

$$X^T BX = I, \text{ for } Ax = \lambda Bx, \text{ and } ABx = \lambda x,$$
$$X^T B^{-1} X = I, \text{ for } BAx = \lambda x.$$

where I is the identity matrix.

Warnings will be issued if there is a clash of dimensions, or A and B are not symmetric, or B is not positive definite.

11.14 Trapezoidal estimate of area under a curve

Trapezoidal estimates for the area under a curve (AUC) or average function value for a set of data points over a range can be used when the data are noisy, or the spacing is too sparse or irregular to fit a deterministic equation or to use data smoothing techniques.

From the SIMFIT main menu select [A/Z], open program **average** and inspect the default test file `average.tf1` which contains this data set.

x	y
1	1
2	2
3	3
4	4
5	5
6	5
7	4
8	3
9	4
10	5
11	8
12	10
13	5
14	9
15	4

Note that program **average** creates a continuous model for the data by joining adjacent points by straight line segments then using the intersection of this model with the Y_{crit} threshold to determine the range of X values where the function lies above or below this threshold.

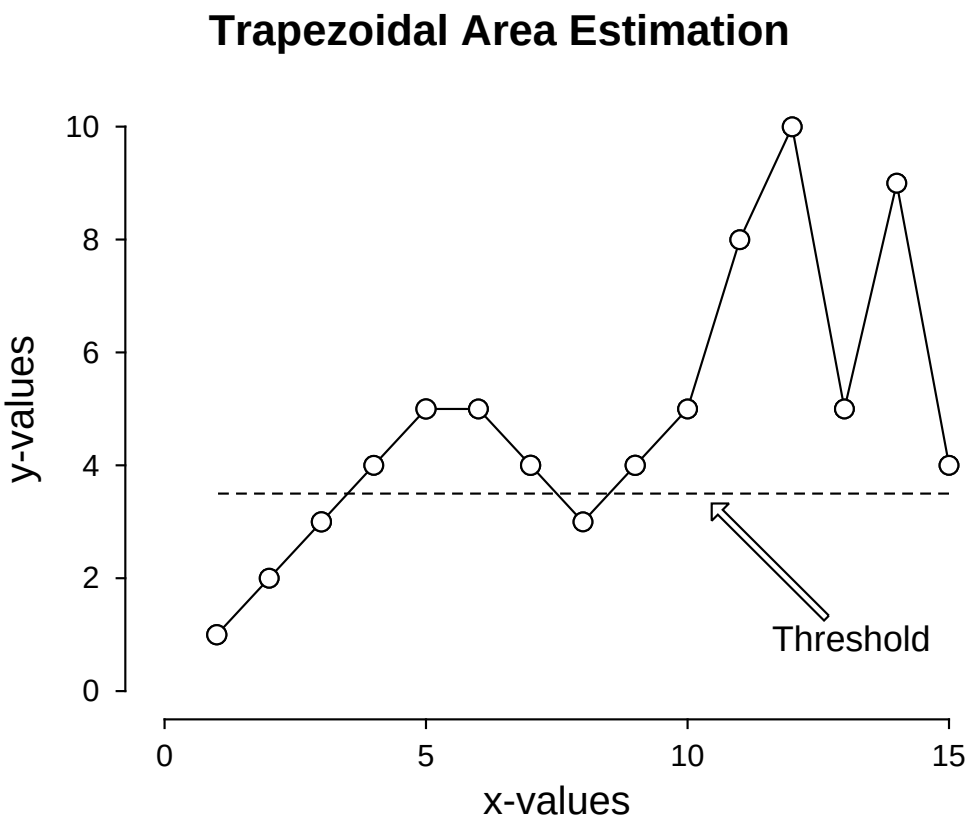
So, proceeding to analyze these data leads to the following results for the default state and the changes resulting from altering the Y threshold from $Y_{test} = 1$ to $Y_{test} = 3.5$.

X_{min}	X_{max}	Y_{min}	Y_{max}					
1	15	1	10					
$Begin = x_1$	$End = x_2$	Above	(as %)	Below	(as %)	Y_{test}	Y_{area}	Y_{mean}
1	15	14	100	0	0	1.0	69.5	4.96
1	15	10.5	75	3.5	25	3.5	69.5	4.96

The results displayed in this table have the following interpretations.

1. The minimum and maximum data values $X_{min}, X_{max}, Y_{min}, Y_{max}$.
2. The range x_1, x_2 of x selected where $X_{min} \le x_1 \le x \le x_2 \le X_{max}$.
3. The trapezoidal area estimate Y_{area} over the range x_1, x_2.
4. The average function value $Y_{mean} = Y_{area}/(x_2 - x_1)$ over the range x_1, x_2.
5. The critical threshold y value Y_{test} where $Y_{min} \le Y_{test} \le Y_{max}$.
6. The percentage of the range x_1, x_2 where the piecewise linear model lies above the critical threshold.
7. The percentage of the range x_1, x_2 where the piecewise linear model lies below the critical threshold.

To understand the effect of changing Y_{test} as are emphasized in red in the above table, consider the next graph.



Further details are now provided to compare and contrast the use of program **average** with some alternative SIMFIT programs.

Estimating AUC using deterministic equations

Observations y_i are often made at settings of a variable x_i as for a regression, but where the main aim is to determine the area under a best fit theoretical curve AUC rather than any best fit parameters. Frequently also $y_i > 0$, which is the case we now consider, so that there can be no ambiguity concerning the definition of the area under the curve. One example would be to determine the average value f_{average} of a function $f(x)$ for $\alpha \le x \le \beta$ defined as

$$f_{\text{average}} = \frac{1}{\beta - \alpha} \int_{\alpha}^{\beta} f(u)\, du.$$

Another example is motivated by the practise of fitting an exponential curve in order to determine an elimination constant k by extrapolation, since

$$\int_0^{\infty} \exp(-kt)\, dt = \frac{1}{k}.$$

Yet again, given any arbitrary function $g(x)$, where $g(x) \ge 0$ for $\alpha \le x \le \beta$, a probability density function f_T can always be constructed for a random variable T using

$$f_T(t) = \frac{g(t)}{\displaystyle\int_{\alpha}^{\beta} g(u)\, du}$$

which can then be used to model residence times, etc.

If the data do have a known form, then fitting an appropriate equation is probably the best way to estimate slopes and areas.

For instance, in pharmacokinetics you can use program **exfit** to fit sums of exponentials and also estimate areas over the data range and AUC by extrapolation from zero to infinity since

$$\int_0^\infty \sum_{i=1}^{n} A_i \exp(-k_i t)\, dt = \sum_{i=1}^{n} \frac{A_i}{k_i}$$

which is calculated as a derived parameter with associated standard error and confidence limits. Other deterministic equations can be fitted using program **qnfit** since, after this program has fitted the requested equation from the library or your own user-supplied model, you have the option to estimate slopes and areas using the current best-fit curve.

Estimating AUC using splines

The SIMFIT spline fitting programs **compare** and **spline** can also be used if the data are extensive enough to fit a meaningful cubic spline reference curve.

Estimating AUC using program average

The main objection to using a deterministic equation to estimate the AUC stems from the fact that, if a badly fitting model is fitted, biased estimates for the areas will result. For this reason, it is frequently better to consider the observations y_i, or the average value of the observations if there are replicates, as knots with coordinates x_i, y_i defining a linear piecewise function. This can then be used to calculate the area for any sub range x_1, x_2 where $X_{min} \le x_1 \le x \le x_2 \le X_{max}$.

Another use for the trapezoidal technique is to calculate areas above or below a baseline, or fractions of the x range above and below a threshold, for example, to record the fraction of a certain time interval that a patients blood pressure was above a baseline value.

Note that, in the previous figure, the base line was set at $y = 3.5$, and program **average** calculates the points of intersection of the horizontal threshold with the linear spline in order to work out fractions of the x range above and below the baseline threshold. For further versatility, you can select the end points of interest, but of course it is not possible to extrapolate beyond the data range to estimate AUC from zero to infinity.

Another feature of program **average** is that the parameters x_1, x_2, Y_{test} that are set for the first data set can be preserved instead of being re-set for each data set. This is intended for a scheme where the first set is intended to be a reference data set. Of course, if these parameters are not consistent with subsequent data sets, warning messages will be issued, and the parameters will be re-set to the normal defaults.

$$\begin{aligned} x_1 &= X_{min} \\ x_2 &= X_{max} \\ Y_{test} &= Y_{min} \end{aligned}$$

11.15 Zeros of 1 function of 1 variable

Given a function of one variable it is possible to estimate zeros as long as two values of the independent variable are provided where the function has opposite signs.

The steps involved are as now detailed.

1. Select a model $f(\Theta, x)$, i.e. a function of one variable, and possibly some parameters Θ.
2. Adjust the parameters Θ if required.
3. Input a constant term C to define the function $g(x) = f(\Theta, x) - C$.
4. Just set $C = 0$ if a root of $g(x) = f(\Theta, x) = 0$ is required.
5. Plot the function $f(\Theta, x)$ to locate two values A, B such that $g(A)g(B) < 0$.
6. Input the proposed limits A, B then solve $g(x) = 0$.
7. If this fails, adjust the limits A, B or tolerance factor $epsrel$ and try again.

As an example, the procedure required to input a user-defined model will be described.

From the main SIMFIT menu select [A/Z], open program **usermod** then read in the SIMFIT model-defining test file `usermods_e.tf1` which has the following form.

```
%
phi(x) = normal cdf
%
1 equation
1 variable
0 parameters
%
begin{expression}
f(1) = phi(x)
end{expression}
%
```

This is a model file for special functions for which the normal cdf function $\Phi(x)$ has been selected, that is

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} \exp\left(-\frac{t^2}{2}\right) dt$$

and it will now be available to program **usermod** for calculations such as root finding or plotting.

The steps necessary to find a root of the equation $g(x) = 0$ where $g(x)$ is defined as

$$g(x) = \Phi(x) - C$$

for some constant C will now be described.

Users must supply a relative error accuracy factor $epsrel$, two values A and B, and a constant C such that, for $g(x) = \Phi(x) - C$, then $g(A)g(B) < 0$.

If the values supplied are such that $g(A)g(B) > 0$, the program will attempt to enlarge the interval in order to bracket a zero, but it will not change the sign of A or B.

Users must do this if necessary by editing the starting estimates A and B.

The program returns the root as X if successful, where X and Y have been located such that

$$|X - Y| \leq 2.0 \times epsrel \times |Z|$$

and $|g(Z)|$ is the smallest known function value, as described for NAG routine C05AZF.

For instance, input $C = 0.975$ so the routine is required to estimate x such that

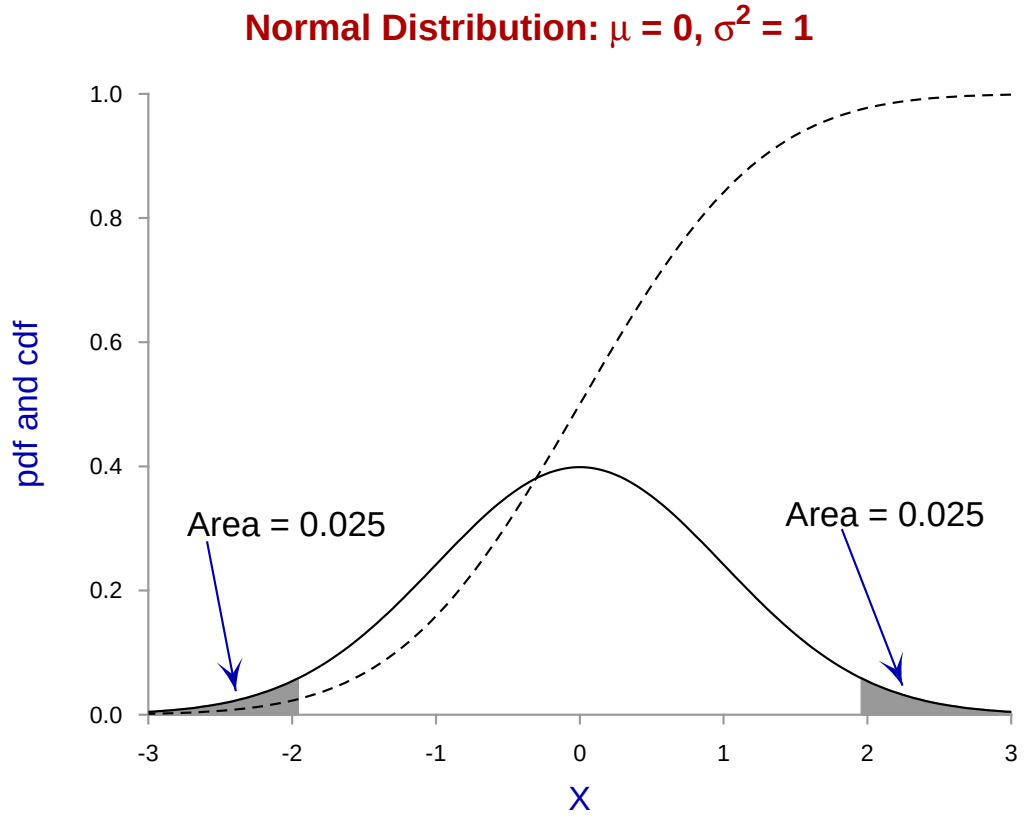
$$0.975 = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} \exp\left(-\frac{t^2}{2}\right) dt$$

and, after setting some reasonable starting estimates, e.g., the defaults (-1,1), the following message will be printed

```
Success : Root =  1.96 (EPSREL = 0.001)
```

giving the root estimated by the SIMFIT implementation of C05AZF.

In other words, this calculation has located the upper 2.5% point of the standard normal distribution as illustrated by the following plot.



11.16 Zeros of n functions of n variables

Given n functions of n variables in a user-defined model file it is sometimes possible to estimate zeros as long as good starting estimates are input, and a reasonable tolerance factor is provided.

The model file must define a system of n equations f_i in n variables x_i and the SIMFIT program **usermod** will attempt to locate $x_1, x_2, \ldots, x_n$ such that

$$f_i(x_1, x_2, \ldots, x_n) = 0, \text{ for } i = 1, 2, \ldots, n.$$

Users must supply good starting estimates by editing the default starting estimates $y_1, y_2, \ldots, y_n$, or installing a new y vector from a file, and the accuracy can be controlled by varying $xtol$, since the program attempts to ensure that

$$||x - \hat{x}|| \leq xtol \times ||\hat{x}||,$$

where $\hat{x}$ is the true solution, as described for NAG routine C05NBF. Failure to converge will lead to nonzero $IFAIL$ values, requiring a re-run with new starting estimates.

From the main SIMFIT menu choose [A/Z] then open program **usermod** and input the test file `usermodn.tf4` which defines 9 equations in 9 variables for the following tridiagonal system.

$$\begin{aligned}(3 - 2x_1)x_1 - 2x_2 + 1 &= 0\\ -x_{i-1} + (3 - 2x_i)x_i - 2x_{i+1} + 1 &= 0, \; i = 2, 3, \ldots, 8\\ -x_8 + (3 - 2x_9)x_9 + 1 &= 0.\end{aligned}$$

After setting the starting estimates $y(i) = 0$ for $i = 1, 2, \ldots, 9$ proceed to locate zeros of n equations in n variables when the following table will result.

$IFAIL$ = 0, $FNORM$ = 7.448E-10, $xtol$ = 1.000E-03

Variable	Value	Function	Value
$x(1)$	-5.7065289E-01	$fvec(1)$	2.5267933E-06
$x(2)$	-6.8162532E-01	$fvec(2)$	1.5688139E-05
$x(3)$	-7.0173246E-01	$fvec(3)$	2.8357029E-07
$x(4)$	-7.0421463E-01	$fvec(4)$	-1.3083878E-05
$x(5)$	-7.0136741E-01	$fvec(5)$	9.8768418E-06
$x(6)$	-6.9186497E-01	$fvec(6)$	6.5557114E-06
$x(7)$	-6.6579418E-01	$fvec(7)$	-1.3053615E-05
$x(8)$	-5.9603414E-01	$fvec(8)$	1.1777047E-06
$x(9)$	-4.1641142E-01	$fvec(9)$	2.9510981E-06

The values displayed at the solution point are as follows.

$IFAIL$	IFAIL = 0, otherwise re-run.
$FNORM$	final 2-norm of the residuals.
$xtol$	Tolerance factor.
$x(i)$	Estimates for x_i.
$fvec(i)$	$f_i(x)$ values.

As values less than about 10^{-7} are effectively zero compared to 1, this represents a satisfactory outcome since $f_i(x) \approx 0$ for $i = 1, 2, \ldots, n$ at the solution point.

For reference, the model is as follows.

```
%
Example: 9 functions of 9 variables as in NAG C05NBF
         set y(1) to y(9) = -1 or 0 for good starting estimates
f(1)=(3-2x(1))x(1)-2x(2)+1, &, f9=-x(8)+(3-2x(9))x(9)+1
 %
 9 equations
 9 variables
 0 parameters
 %
 begin{expression}
 f(1) = (3 - 2y(1))y(1) + 1 - 2y(2)
 f(2) = (3 - 2y(2))y(2) + 1 - y(1) - 2y(3)
 f(3) = (3 - 2y(3))y(3) + 1 - y(2) - 2y(4)
 f(4) = (3 - 2y(4))y(4) + 1 - y(3) - 2y(5)
 f(5) = (3 - 2y(5))y(5) + 1 - y(4) - 2y(6)
 f(6) = (3 - 2y(6))y(6) + 1 - y(5) - 2y(7)
 f(7) = (3 - 2y(7))y(7) + 1 - y(6) - 2y(8)
 f(8) = (3 - 2y(8))y(8) + 1 - y(7) - 2y(9)
 f(9) = (3 - 2y(9))y(9) + 1 - y(8)
end{expression}
```

11.17 Integration of 1 function of 1 variable

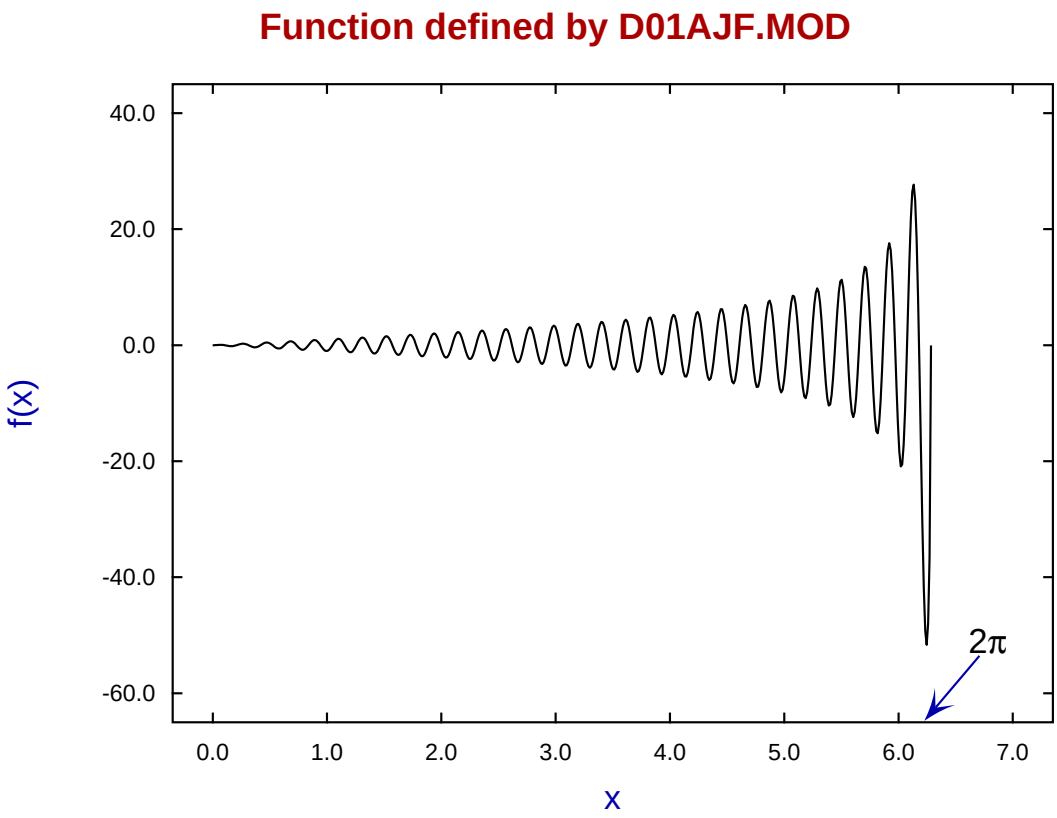
The Simpson method for estimating an area between two end points is satisfactory for smooth well-behaved functions. However, for complicated functions, adaptive numerical quadrature is required where the method used takes account of the rate of change of the function.

So, to integrate a function, it is necessary to define the function along with the range of integration and parameters to control the number of Simpson rule divisions as well as a tolerance factor for numerical quadrature, as demonstrated in the next worked example.

From the main SimFIT menu select [A/Z] then open the SimFIT program **usermod**, choose to integrate a function of one variable and read in the test file `d01ajf.mod` which defines the following function.

$$f(x) = \frac{x \sin(30x)}{\sqrt{\left(1 - \left(\frac{x}{2\pi}\right)^2\right)}}$$

The next plot for this function over the range 0 to 6.283153 (i.e. $< 2\pi$) indicates that this will be a very difficult function integrate and that adaptive quadrature will be required. Note that the range of plotting and integration must not actually include the pole at $x = 2\pi$.



Integration by the Simpson rule	
Number of Simpson divisions	100
Area by the Simpson rule	-2.2143991

Integration by adaptive quadrature	
IFAIL (from D01AJF)	0
EPSABS	0.000001
EPSREL	0.001
ABSERR	0.001926
Area by adaptive integration	-2.5432599

The definition of the function contained in test file `d01ajf.mod` now follows.

```
%
Example: function for d01ajf
.................................
x*sin(30*x)/sqrt{1 - (x/2pi)^2}
Note: -2pi < x < 2pi to avoid poles
.................................
Usage as follows
Select simulation and open program usermod
Select 1 function of 1 variable then read in this file
Set tolerances and limits (0 to just less than 2pi)
Select integrate 1 function of 1 variable and integrate
NAG reports -2.54326 for A=0,B=6.2832,epsabs=0,epsrel=1.e-4
Simfit agrees but with B=6.28318 to avoid the discontinuity
%
1 equation
1 variable
0 parameters
%
begin{expression}
f(1) = x*sin(30x)/sqrt[1.0 - (x/{2pi})^2]
end{expression}
%
```

11.18 Integration of 1 function of m variables

Given a model defining one or more equations in one or more variables, the integral(s) can be estimated over a hyper-rectangular region defined by fixed limits.

The following procedure is required for just one function of $m \geq 1$ variables, although $n \geq 1$ functions of $m \geq 1$ variables can be analyzed in exactly the same way.

1. Create a file defining the function of m variables to be integrated.
2. Open program **usermod** and input the file defining one function of m variables.
3. It is necessary to explicitly indicate that one function is required and m must be input correctly.
4. Program **usermod** then checks that the function is defined correctly.
5. The range of integration required must be defined by editing the vectors $BLIM$ and $TLIM$ to specify the m lower and upper limits for the corresponding variables.
6. The absolute error $EPSABS$ and relative error $EPSREL$ parameters required must be set.
7. Integration can then be requested but the result should only accepted if $IFAIL = 0$ on completion.
8. If $|IFAIL| > 0$ some of the previous parameters will have to be adjusted.

From the main SimFIT menu, choose [A/Z], open program **usermod**, then read in the test file `d01fcf.mod` which defines the the integrand used to evaluate the following integral

$$I = \int_0^1 \int_0^1 \int_0^1 \int_0^1 \frac{4u_1u_3^2 \exp(2u_1u_3)}{(1+u_2+u_4)^2}\, du_4\, du_3\, du_2\, du_1$$

and the results are listed in the next table.

$IFAIL$	0 (from $D01EAF$)
$EPSABS$	1.000E-06
$EPSREL$	1.000E-03

Number	$BLIM$	$TLIM$
1	0.0	1.0
2	0.0	1.0
3	0.0	1.0
4	0.0	1.0

Number	$INTEGRAL$	$ABSEST$
1	0.57533267	1.0782E-04

Note that in order to perform the integration it may be necessary to re-define the limits, absolute, and relative tolerances, which can be done interactively.

Exit with IFAIL = 0 indicates that the absolute error estimate $ABSEST$ satisfies

$$|ABSEST| \leq \max(EPSABS, ABSREL \times |INTEGRAL|)$$

as defined for NAG routine D01EAF.

The model equation file `d01fcf_e.mod` is as follows.

```
%
f(y) = {4y(1)y(3)^2[exp(2y(1)y(3))]}/{1 + y(2) + y(4)}^2
%
1 equation
4 variables
0 parameters
%
begin{expression}
f(1) = 4y(1)y(3)^2[exp(2y(1)y(3))]/[1.0 + y(2) + y(4)]^2
end{expression}
%
```

11.19 Integration of n functions of m variables

Given a model defining several equations in one or more variables, the integrals can be estimated over a hyper-rectangular region defined by fixed limits.

The following procedure is required for $n \geq 1$ functions of $m \geq 1$ variables.

1. Create a file defining the n functions of m variables to be integrated.
2. Open program **usermod** and input the file defining n function of m variables.
3. It is necessary to explicitly indicate that n functions of m variables are required and the values for n and m must be specified correctly.
4. Program **usermod** then checks that the function is defined correctly.
5. The range of integration required must be defined by editing the vectors $BLIM$ and $TLIM$ to specify the m lower and upper limits for the corresponding variables.
6. The absolute error $EPSABS$ and relative error $EPSREL$ parameters required must be set.
7. Integration can then be requested but the result should only accepted if $IFAIL = 0$ on completion.
8. If $IFAIL = 1$ on exit, then re-entry for continued iterations will be offered, otherwise some of the previous parameters will have to be adjusted and the integration repeated.

From the main SimFIT menu, choose [A/Z], open program **usermod**, then read in the test file `d01eaf.mod` which defines the the integrand used to evaluate the following integral The program accepts a user defined model for n functions of m variables and estimates the n integrals

$$I_i = \int_{A_1}^{B_1} \int_{A_2}^{B_2} \ldots \int_{A_m}^{B_m} f_i(x_1, x_2, \ldots, x_m)\, dx_m \ \ldots \ dx_2\, dx_1$$

for $i = 1, 2, \ldots, n$, where the limits are taken from the arrays $A_i = blim(i)$ and $B_i = tlim(i)$. The procedure only returns IFAIL = 0 when

$$\max_i(ABSEST(i)) \leq \max(EPSABS, EPSREL \times \max_i |FINEST(i)|),$$

where $ABSEST(i)$ is the estimated absolute error in $FINEST(i)$, the final estimate for integral i, as described for NAG routine D01EAF.

The n functions defined by SimFIT test file `d01eaf_e.mod` are

$$f_j = \log(x_1 + 2x_2 + 3x_3 + 4x_4)\sin(j + x_1 + 2x_2 + 3x_3 + 4x_4) \text{ for } j = 1, 2, \ldots, 10$$

while the results from integration are listed in the following tables.

Results from the integration of `d01eaf_e.mod`

$IFAIL$	0 (from D01EAF)
$EPSABS$	1.000E-06
$EPSREL$	1.000E-03
$MINCLS$	459 (Function evaluations)
$TESTER$	4.417E-04 (Error threshold: * where exceeded)

Variable	*BLIM*	*TLIM*
1	0.0	1.0
2	0.0	1.0
3	0.0	1.0
4	0.0	1.0

Function	*INTEGRAL*	*ABSEST*
1	3.8352146E-02	1.8779E-04
2	4.0118447E-01	2.3766E-04
3	3.9516964E-01	1.6379E-04
4	2.5837668E-02	1.7314E-04
5	-3.6724934E-01	2.3574E-04
6	-4.2268900E-01	1.5493E-04
7	-8.9510341E-02	1.5503E-04
8	3.2596371E-01	2.2910E-04
9	4.4174823E-01	4.5854E-03 *
10	1.5139146E-01	5.1370E-04 *

The other parameters in these tables that have not already been defined have the following meanings.

MINCLS	Number of calls to the subroutine for function evaluations.
TESTER	Maximum error estimate acceptable so that items larger than this (if any) are indicated by the symbol * in the listing (as for functions 9 and 10). There can be a few * symbols and still have IFAIL = 0 on exit as a slightly weaker test than this is performed by the numerical integrator.
INTEGRAL	Integral for listed function.
ABSEST	Error estimate for listed function.

The SIMFIT test file defining these 10 functions of 4 variables is now listed.

```
%
...
model for the 10 functions in 4 variables required to demonstrate D01EAF
f_j = log(x_1 + 2x_2 + 3x_3 + 4x_4)(sin(j + x_1 + 2x_2 + 3x_3 + 4x_4)
      for j = 1, 2, ..., 10
...
%
10 equations
4 variables
0 parameters
%
begin{expression}
A = y(1) + 2y(2) + 3y(3) + 4y(4)
B = log(A)
 f(1) = B*sin(1 + A)
 f(2) = B*sin(2 + A)
 f(3) = B*sin(3 + A)
 f(4) = B*sin(4 + A)
 f(5) = B*sin(5 + A)
 f(6) = B*sin(6 + A)
 f(7) = B*sin(7 + A)
 f(8) = B*sin(8 + A)
```

```
 f(9) = B*sin(9 + A)
f(10) = B*sin(10 + A)
end{expression}
%
```

Note the use of A and B dummy variable commands to avoid re-calculations.

11.20 Bound-constrained quasi-Newton optimization

Bound-constrained quasi-Newton optimization by LBFGSB can be used to minimize a user-defined model, but for this procedure it requires starting estimates and the partial derivatives to be provided as well as the function to be minimized.

So the user supplied model must define $n + 1$ functions of n variables as follows

$$\begin{aligned} f(1) &= F(x_1, x_2, \ldots, x_n) \\ f(2) &= \partial F/\partial x_1 \\ f(3) &= \partial F/\partial x_2 \\ &\ldots \\ f(n+1) &= \partial F/\partial x_n. \end{aligned}$$

The limited memory quasi-Newton optimization procedure also requires several other parameters, as now listed.

- $MHESS$ is the number of limited memory corrections to the Hessian that are stored. The value of 5 is recommended but, for difficult problems, this can be varied in the range 4 to 17.

- $FACTR$ should be about 1.0e+12 for low precision, 1.0e+07 for medium precision, and 1.0e+01 for high precision. Convergence is controlled by $FACTR$ and $PGTOL$ and will be accepted if

  $$|F_k - F_{k+1}|/\max(|F_k|, |F_{k+1}|, 1) \leq FACTR * EPSMCH$$

  at iteration $k + 1$, where $EPSMCH$ is machine precision, or if

  $$\max_i(\text{Projected Gradient}(i)) \leq PGTOL.$$

- Starting estimates and bounds on the variables can be set by editing the defaults or by installing from a data file.

- The parameter $IPRINT$ allows intermediate output every $IPRINT$ iterations, and the final gradient vector can also be printed if required.

- The program opens two files at the start of each optimization session, `w_usermod.err` stores intermediate output every $IPRINT$ iterations plus any error messages, while `iterate.dat` stores all iteration details, as for SIMFIT programs **qnfit** and **deqsol** when they use the LBFGSB suite for optimization.

- Note that, when $IPRINT > 100$ full output, including intermediate coordinates, is written to `w_usermod.err` at each iteration.

As an example, input the model file `optimum.mod` defining Rosenbruck's two dimensional test function

$$F(x, y) = 100(y - x^2)^2 + (1 - x)^2$$

which has a unique minimum at $x = 1, y = 1$ and is represented as follows.

```
 %
f(1) = 100(y - x^2)^2 + (1 - x)^2
f(2) = -400x(y - x^2) - 2(1 - x)
f(3) =  200(y - x^2)
%
 3 equations
 2 variables
 0 parameters
 %
 begin{expression}
 A = y - x^2
 B = 1 - x
 f(1) = 100*A^2 + B^2
 f(2) = -400A*x - 2B
 f(3) = 200A
 end{expression}
 %
```

In order to locate a minimum of this function it is necessary to specify the starting estimates along with parameters controlling the optimization and the output.

For example, the iteration using the default optimization parameters and good starting estimates, in particular

$$-10 \le x \le 10, \quad x_{start} = 0$$
$$-10 \le y \le 10, \quad y_{start} = 0$$

with IPRINT = 5 for output at every fifth iteration proceeds as in the next table.

Iterate	$F(x)$	$\|prj.grd.\|$	$Task$
1	6.9219E-01	5.0534E+00	NEW_X
6	2.1146E-01	3.1782E+00	NEW_X
11	1.7938E-02	3.5920E-01	NEW_X
16	1.7768E-04	4.4729E-02	NEW_X
20	5.5951E-13	7.2120E-06	$CONVERGENCE$: $NORM\ OF\ PROJECTED\ GRADIENT \le PGTOL$
Solution	Derivatives		
$x = 1$	$\partial F/\partial x$	7.21198E-06	
$y = 1$	$\partial F/\partial y$	-2.87189E-06	

The parameter $Task$ informs users of the action required after each intermediate iteration, then finally it records the reason for termination of the optimization.

As this type of minimization can only locate a local minimum and success depends critically on scaling the problems and choosing good starting estimates, it will usually be necessary to experiment with alternative settings until a reliable convergence has been achieved.

Actually program **usermod** opens two files `iterate.dat` and `w_usermod.err` in the user folder which can be consulted retrospectively to follow the iterations.

In particular, choosing the maximum value of IPRINT = 101 causes output to `w_usermod.err` for every iteration as well as error reports, and these data can be used retrospectively to create contour maps with the optimization trajectory overlaid. This will be discussed in another tutorial.

11.21 Optimization contours with trajectory

It is often useful to be able to plot contours for a function of two variables around the minimum of an objective function, with the other variables fixed if there are more two, and then to overlay the optimization trajectory.

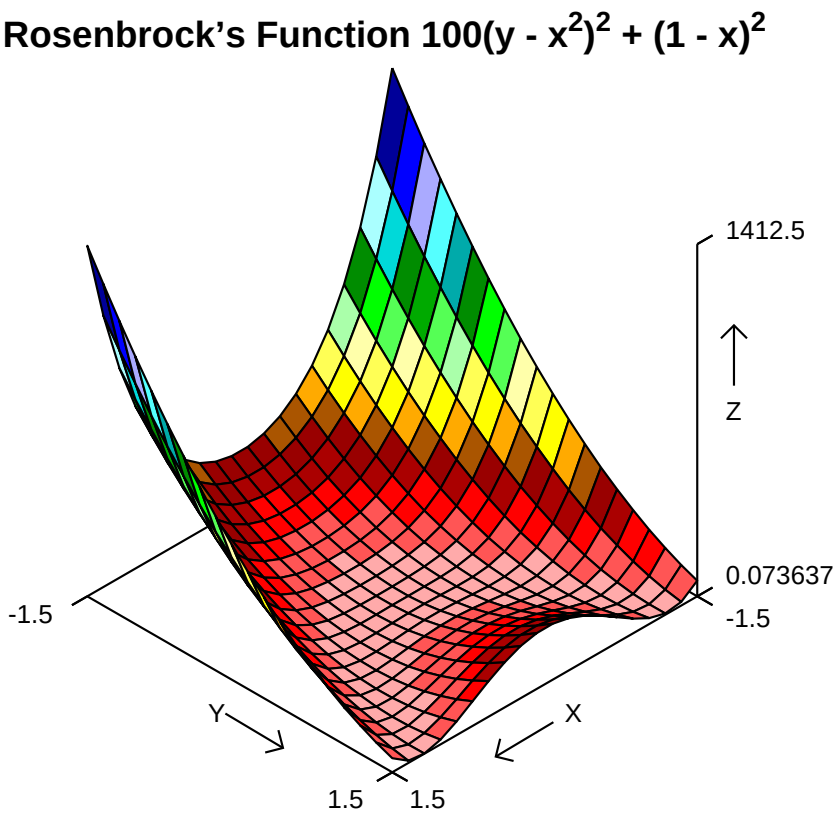
Plotting the 3D surface

As an example, from the main SimFIT menu choose [A/Z], open program **usermod** and read in the model file `optimum.mod` defining Rosenbrock's two dimensional test function

$$f(x, y) = 100(y - x^2)^2 + (1 - x)^2$$

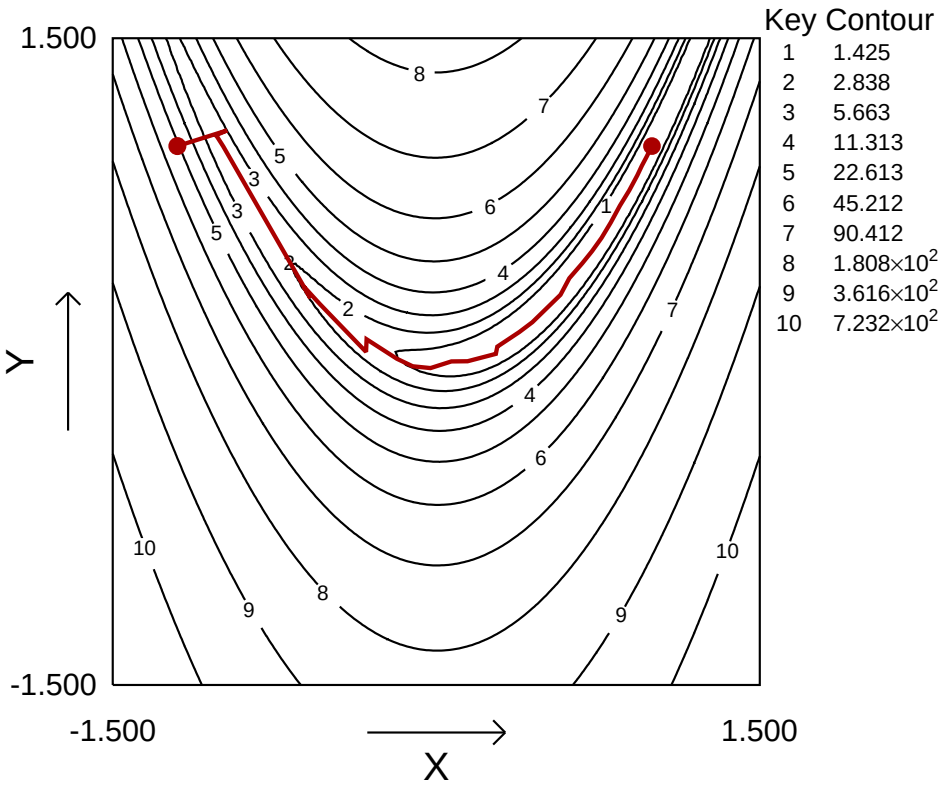
which has a unique minimum at $x = 1, y = 1$. Note that this model file also defines the two partial derivatives that are required for optimization.

To begin with we can simply plot the 3D surface as follows.



Overlaying a trajectory on the contours

This necessitates obtaining a set of coordinates for the progress of optimization from a starting point and several other actions. The next plot is followed by details of how it was constructed.



1. **The optimization step**
   An optimization was performed, starting at $x = -1.2, y = 1$ with $IPRINT = 101$, which created the file `w_usermod.err` in the results folder containing optimization details.

2. **Creating an iteration file**
   The coordinates for the iterations from the file `w_usermod.err` in the results folder were copied into a file, `rosenbruck_iterations.txt`, with x in column 1 and y in column 2 which were eventually plotted as the red polyline.

3. **Creating a user-defined spacing file**
   A file `rosenbruck_proportions.txt` with a column of 10 proportions 1/2, 1/4, 1/8, etc. was created to space the contour values sensibly as proportions of the maximum function value.

4. **Plotting the contours**
   The Rosenbruck model was plotted over the range $-1.5 \geq x \leq 1.5$, $-1.5 \geq y \leq 1.5$ using 100 divisions for the x and y axes in order to create smooth contours.

5. **Spacing the contours**
   To complete the diagram as illustrated above, the file `rosenbruck_proportions.txt` was installed to display the contours in geometric progression up to half the maximum functions value.

6. **Adding the trajectory and end points**
   A file was created with the starting coordinates $-1.2, 1$ and final coordinates $1, 1$ that are plotted as red solid circles, then the contours were overlayed using the coordinate file `rosenbruck_iterations.txt`. Alternatively, the start and end coordinates could have been added as graphical objects.

11.22 Evaluation of convolution integrals for plotting or fitting

The evaluation of convolution integrals is frequently required in simulation or data analysis, e.g., where such integrals occur as output functions from the response of some device to an input function.

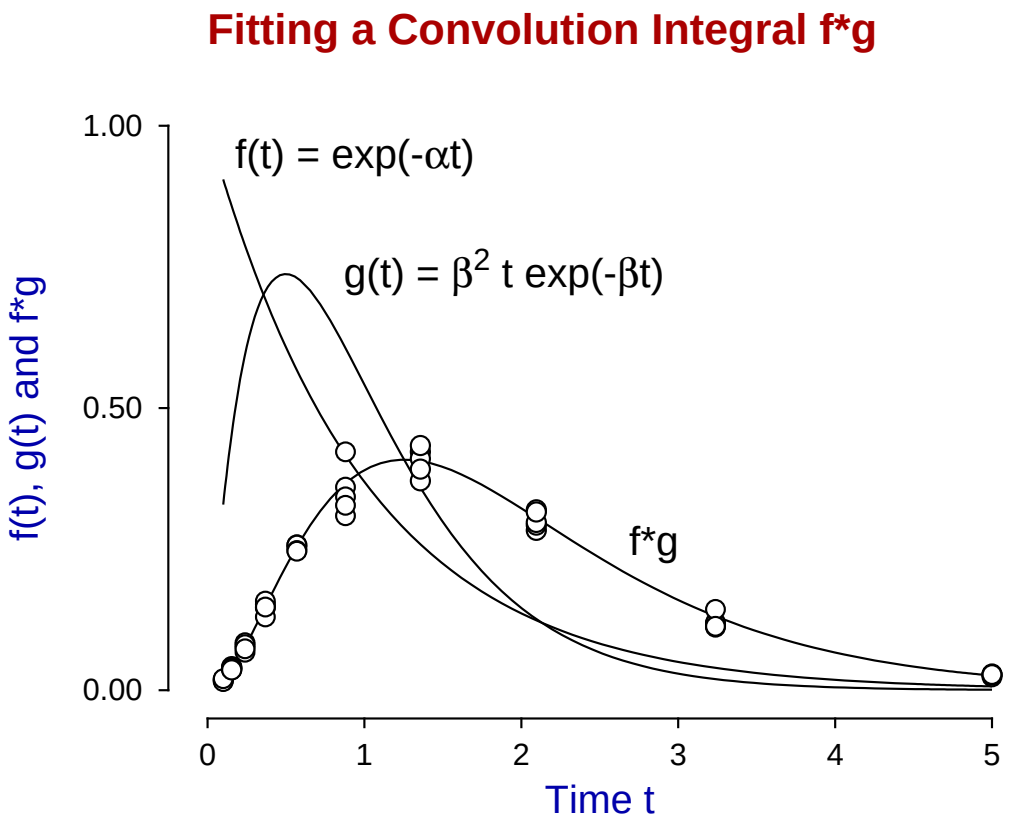
For instance, fitting convolution integrals involves parameter estimation in model functions $f(t)$, $g(t)$, and their convolution $(f * g)(t)$, where

$$(f * g)(t) = \int_0^t f(u)g(t-u)\,du.$$

Sometimes the input function can be controlled independently so that, from sampling the output function, parameters of the response function can be estimated, and frequently the functions may be normalized, e.g. the response function may be modeled as a function integrating to unity as a result of a unit impulse at zero time. However, any one, two or even all three of the functions may have to be fitted. The next figure illustrates fitting data in `convolv.tfl` using the model `convolv3.mod`, i.e.

$$f(t) = \exp(-\alpha t)$$
$$g(t) = \beta^2 t \exp(-\beta t)$$
$$1 = \int_0^\infty g(t)\,dt$$

with SIMFIT program **qnfit**.



In some cases a convolution integral can be performed explicitly, say by Laplace transforms, but SIMFIT provides a method to evaluate two arbitrary functions $f(\Theta, x)$, $g(\Theta, x)$ and the convolution integral $f * g$ at the same time so that plotting and estimation of parameters Θ can be performed on any combination of the two functions and the convolution integral. The next example shows how the test file `convolv3.mod` is formatted. Note the following special commands.

- *putpar*
  This ensures that parameters are global, i.e., available to all models.
- *user*1(x, m)
  This reads m then x off the stack and adds submodel m evaluated at argument x.
- *begin model*(m), and *end model*(m)
  These define the start and completion of code to define model m.

In the results displayed above, the library file for the data was `convolv3.tfl` as follows.

```
Convolution data
%
%
convolv3.data
```

Percentage signs indicate that only output data was supplied in data file `convolv3.data` so only the convolution was fitted to the data, and plots for the input and transformation functions were for information.

```
 %
This demonstrates how to define 2 equations as sub-models, using
the command putpar to communicate parameters to the sub-models,
and the command convolute(1,2) to integrate sub-models 1 and 2
(by adaptive quadrature) from blim(1) = 0 to t = tlim(1) = x.
Precision of D01AJF quadrature is controlled by epsabs and epsrel
and blim(1) and tlim(1) must be used for the convolution limits
which, in this case are 0 to x, where x > 0 by assumption.
.............................................................
convolution integral: from 0 to x of f(u)*g(x - u) du, where
f1(t) = f(t) = exp(-p(1)*t)
f2(t) = g(t) = [p(2)^2]*t*exp(-p(2)*t)
f3(t) = f*g  = f1*f2
.............................................................
Note that usually extra parameters must be supplied if it wished
to normalise so that the integral of f or g or f*g is specified
(e.g.,equals 1) over the total range of possible integration.
This must often be done, e.g., if g(.) is a density function.
The gamma distribution normalising  factor p(2)**2 is stored in
this example to avoid unnecessary re-calculation.
 %
 3 equations
 1 variable
 2 parameters
 %
 putpar
 p(2)
 p(2)
 multiply
 put(1)
 1
 x
```

```
 user1(x,m)
 f(1)
 2
 x
 user1(x,m)
 f(2)
 0.0001
 epsabs
 0.001
 epsrel
 0
 blim(1)
 x
 tlim(1)
 convolute(1,2)
 f(3)
 %
begin{model(1)}
 %
Example: exponential decay, exp(-p(1)*x)
 %
 1 equation
 1 variable
 1 parameter
 %
 p(1)
 x
 multiply
 negative
 exponential
 f(1)
 %
end{model(1)}
begin{model(2)}
 %
Example: gamma density of order 2
 %
 1 equation
 1 variable
 2 parameters
 %
 p(2)
 x
 multiply
 negative
 exponential
 x
 multiply
 get(1)
 multiply
 f(1)
 %
end{model(2)}
```

12 Simulating and fitting differential equations

Tutorials and worked examples for simulation, curve fitting, statistical analysis, and plotting.
https://simfit.uk
https://simfit.org.uk
https://simfit.silverfrost.com

12.1 introduction

To simulate theoretical models or to fit deterministic models to experimental data it is frequently necessary to solve an ordinary nonlinear differential equation, or a system of ordinary nonlinear differential equations. This usually cannot be done by formal integration but is often possible using numerical methods under the following conditions.

1. The independent variable, say x, is nonnegative so that the system is parameterized for $x \geq 0$.
2. The model can be expressed naturally as a system of n nonlinear equations. Alternatively, there is one equation of order n where the highest order derivative can be expressed as a function of the independent variable, the dependent variable, and all the other derivatives, so that this can be transformed into a system of n equations.
3. The initial conditions at $x = 0$ are known, i.e., it is an initial value problem (IVP).
4. The range of integration and the number of intermediate independent variable points are defined.
5. An appropriate method is selected.

Definitions

The system of n equations must be expressed in the following form

$$\begin{aligned} y_1' &= f_1(x, y_1, y_2, \ldots, y_n) \\ y_2' &= f_2(x, y_1, y_2, \ldots, y_n) \\ &\ldots \\ y_n' &= f_n(x, y_1, y_2, \ldots, y_n) \end{aligned}$$

where $y_i' = dy_i/dx$ for components $y_1, y_2, \ldots, y_n$.

For example, consider the Lotka-Volterra predator-prey differential equations where y_1 are numbers of prey, say rabbits, while y_2 are numbers of predators, say foxes, so that

$$\begin{aligned} \frac{dy_1}{dx} &= p_1 y_1 - p_2 y_1 y_2 \\ \frac{dy_2}{dx} &= -p_3 y_2 + p_4 y_1 y_2. \end{aligned}$$

Here the model is expressed, using four fixed or variable parameters $p_1, p_2, p_3,$ and p_4, as a system of two autonomous differential equations, i.e., x does not appear on the right hand sides of the defining equations.

However, the Van der Pol oscillator is the second order differential equation

$$\frac{d^2z}{dx^2} + \mu(z^2 - 1)\frac{dz}{dx} + z = 0$$

which can be expressed as the following autonomous set of two equations using $y_1 = z$, $y_2 = dy_1/dx = dz/dx$, and $dy_2/dx = d^2z/dx^2$

$$\frac{dy_1}{dx} = y_2$$
$$\frac{dy_2}{dx} = -y_1 - \mu(y_1^2 - 1)y_2.$$

Of course the differential equations must be supplied along with the range of independent variable, initial conditions, and tolerance settings, but also the numerical method must be chosen subject to these considerations.

- For non-stiff problems where all components have similar relaxation times, the standard methods due to Runge-Kutta, or Adams etc. should be used.
- For stiff problems, where components have widely differing relaxation times, i.e., the condition number of the Jacobian matrix is large, it will be necessary to use the backward differentiation method (BDF) due to Gear. In this case the Jacobian $J_{ij} = \partial f_i/\partial y_j$ is used and there are two possibilities.
    1. The Jacobian matrix can be supplied along with the differential equations. This is the best technique to use, but has the great disadvantage that if the Jacobian matrix is not coded accurately the convergence may be slow or lead to an erroneous solution.
    2. To avoid this happening the Jacobian can be approximated at run time by finite differences. This is not so good as supplying an accurate explicit Jacobian, but will avoid the problem of having to supply an accurate explicit Jacobian.
- The recommended procedure is to experiment with the methods and tolerances interactively until a satisfactory outcome has been achieved.
- Usually a system of n equations will involve $k \geq 0$ parameters p_i that are to be fixed at known values or estimated by curve fitting a given data set. The rule used by SimFIT is that there must be $k + n$ parameters, as the last n parameters i.e., $p_{k+1}, p_{k+2}, \ldots, p_{k+n}$ are the initial conditions.

Simulation

A single differential equation can be simulated using SimFIT program **makdat**, which is dedicated to generating data intended to be used by program **adderr** in order to add random error to mimic experimental data.

Systems of differential equations can be simulated by program **deqsol** which offers several options as follows.

- Use equations from a supplied library or user-defined model file.
- Change the method and tolerance factor in order to identify optimum conditions.
- Plot or tabulate selected components.
- Store orbits to be collected together for plotting.
- Plot phase diagrams for autonomous systems.
- Change the equations, parameter values, initial conditions, or range of integration interactively.

Simulation would normally be used to explore the outcome of changing parameters or initial conditions, especially to estimate starting parameters for subsequent curve-fitting.

Curve fitting differential equations

The SimFIT program **qnfit** can be used to fit a single differential equation to data, and it is preferred for this purpose as it offers very extensive options for goodness of fit, parameter reliability, and model discrimination.

However, program **deqsol** has additional options making it more suitable for fitting systems of equations. This program provides options for overlaying multiple trajectories on data points before and after fitting in order to visualize the progress of optimization to a solution point.

It should be emphasized that curve fitting differential equations to experimental data is an extremely difficult subject requiring considerable expertise, and the following facts should be considered.

- The data should be supplied using the library file approach.
- The data points must be supplied in nondecreasing order with respect to the independent variable. This is to facilitate the ability to integrate methodically between fixed levels of the independent variable, but also to ensure that only the first integral for the model is calculated for groups of replicates.
- The system should be simulated before fitting in order to ensure that starting estimates and parameter limits are reasonable.
- There are two options available concerning initial conditions, i.e., the last n parameters.

  1. The initial conditions can be set at known fixed values.
  2. Some or all of the initial conditions can be estimated.

  Wherever possible the first of these methods should be used because many serious problems can be encountered when attempting to estimate initial conditions along with other model parameters.

To summarize: fitting differential equations to data will only be satisfactory when these conditions are met.

1. The model proposed can simulate the data and makes sense theoretically.
2. The data are accurate with a high signal to noise ratio.
3. The range of independent variable is sufficiently extensive to expose the influence of all the parameters to be estimated.
4. The starting estimates are close to the parameters expected at the eventual solution point, and parameter limits prevent excessive movement of parameters away from those expected at the solution point.

Note that programs **qnfit** and **deqsol** are designed to terminate with fixed values for system components as a warning if these conditions are not met.

12.2 The Von Bertalanffy allometric differential equation

The Von Bertalanffy differential equation arose in allometric studies in order to represent growth as the result of a balance between anabolic and catabolic processes.

For instance, a spherical bacterium of radius r has a rate of absorbtion of nutrients proportional to the surface area $4\pi r^2$ but a rate of metabolism proportional to the volume $4\pi r^3/3$, so the ratio of surface area to bulk decreases as $3/r$ and it would be anticipated that the rate of change in size y as a function of time x, say dy/dx, would increase rapidly for small y but slow down as y increases. This leads to the expression

$$\frac{dy}{dx} = p_1 y^{p_2} - p_3 y^{p_4}, \quad y(0) = p_5$$

for some exponents p_2 and p_4 with $p_4 > p_2 \geq 0$, where to model a growth process y and the parameters p_i would be nonnegative. For certain special parameter values formal integral is possible and leads to some of the classical growth equations as follows.

$$\begin{aligned}
\text{Exponential model } dS/dt &= kS \\
S(t) &= A\exp(kt), \text{ where } A = S_0 \\
\text{Monomolecular model } dS/dt &= k(A - S) \\
S(t) &= A[1 - B\exp(-kt)], \text{ where } B = 1 - S_0/A \\
\text{Logistic model } dS/dt &= kS(A - S)/A \\
S(t) &= A/[1 + B\exp(-kt)], \text{ where B } = A/S_0 - 1 \\
\text{Von Bertalanffy 2/3 model } dS/dt &= \eta S^{2/3} - \kappa S \\
S(t) &= [A^{1/3} - B\exp(-kt)]^3, \\
&\quad \text{where } A^{1/3} = \eta/\kappa, B = \eta/\kappa - S_0^{1/3}, k = \kappa/3
\end{aligned}$$

However the Von Bertalanffy differential equation will be used to illustrate the methods available using SimFIT to study the properties of a differential equation, followed by simulation and curve fitting.

To examine this differential equation requires choosing the fixed parameters and the initial condition so the following values will be used for this purpose.

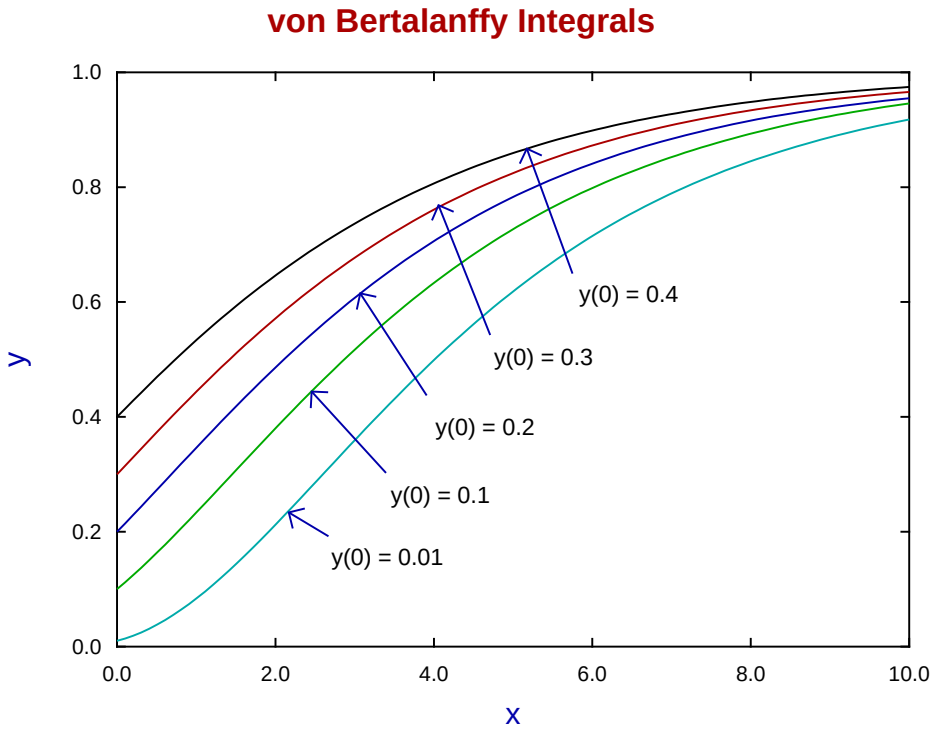
$$\begin{aligned}
p_1 &= 1 \\
p_2 &= 2/3 \\
p_3 &= 1 \\
p_4 &= 1 \\
p_5 &= 0.01
\end{aligned}$$

Of course this is the original case where a formal integral exists and it is an interesting exercise to consider the alternative $y(x)$ profiles that result as these parameters are varied. However, with these parameter values the following observations can be made.

1. The slope is zero when y is zero and when y is 1
2. As x does not appear on the right hand side the slope is fixed given any y for $0 \leq y \leq 1$
3. The $y(x)$ curve is monotonically increasing for $x > 0$

Initial conditions

Here, for example, is a series of plots illustrating how increasing the initial condition $y(0)$ from $p_5 = 0.01$ to $p_5 = 0.4$ simply slides the profiles to the left.



To make composite plots of $y(x)$ profiles such as this for any differential equations use the following protocol.

1. Open program **deqsol** from the [A/Z] option on the main SIMFIT menu and select the differential equation required from the library of pre-compiled models, or by reading in a user-defined model.
2. Edit the parameters as required, which in this example using the library model is just p_5.
3. Plot the $y(x)$ data between the end points selected (in this case 100 points for $0 \leq x \leq 10$) and use the [Advanced] option to save the coordinates to file. Note that, after doing this, the files saved can be added to your graph plotting archive for retrospective plotting as part of a group.
4. Input the group of coordinate files into program **simplot** either individually, from your project archive, or best of all from a library file created using program **maklib**, then edit as required.

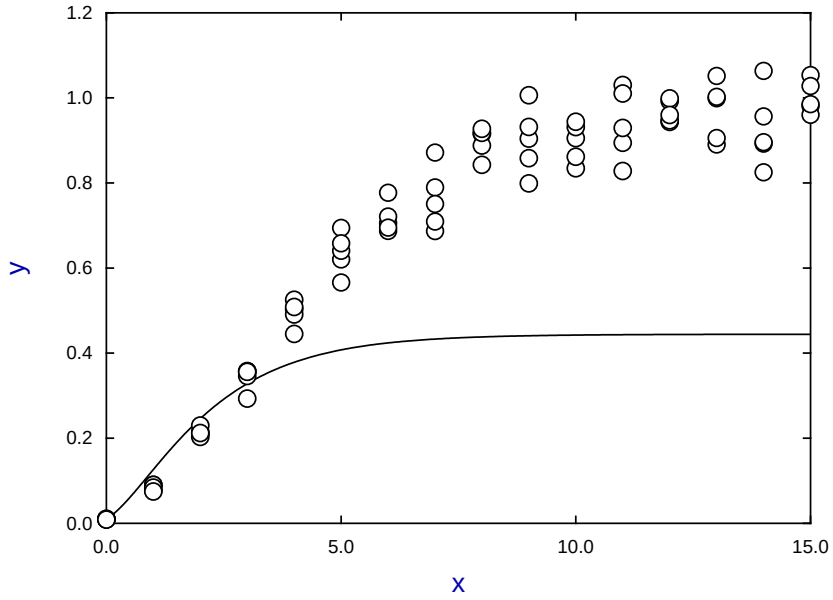
Simulating and fitting

As we are dealing with a single differential equation, it is best to proceed as follows

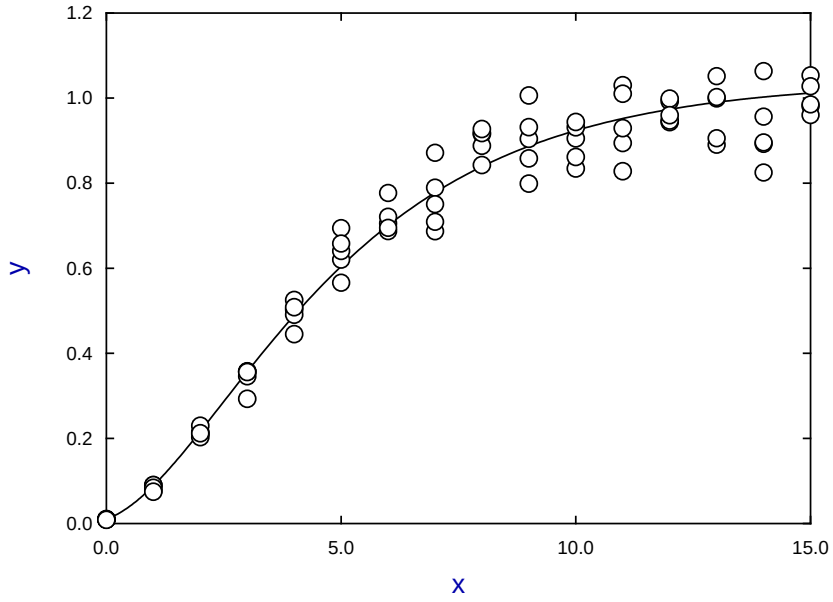
- Use program **makdat** to simulate a $y(x)$ profile.
- Use program **adderr** to add random error.
- Finally fit using program **qnfit**.

That is because these programs provide more options than are available with program **deqsol**, which is really designed to work with systems of differential equations. Using program **qnfit** to fit the default data set `qnfit_ode.tf2` results in the following outcome. First the overlay before fitting, then the overlay after fitting, and finally the results.

Data and Starting-Estimate-Curve



Data and best-fit curve



Best-fit parameters for curve-fit 1 using LBFGSB/DVODE

Number	Low-Limit	High-Limit	Value	Std. Error	Lower95%cl	Upper95%cl	*p*
1	0.50	1.50	0.887160	0.0029641	0.88126	0.89306	0.0000
2	0.25	0.75	0.616683	0.0009505	0.61479	0.61858	0.0000
3	0.50	1.50	0.875717	0.0021387	0.87146	0.87998	0.0000
4	0.75	1.25	0.943496	0.0026182	0.93828	0.94871	0.0000
5	0.0001	0.05	0.010133	0.0000739	0.00999	0.00103	0.0000

Of course this is an artificial data set over an extended range using a model defined in the SIMFIT library, and the fitting included the initial condition for purposes of illustration, which should be avoided at all costs with actual data.

The von Bertalannffy user defined model

For information, the corresponding default user-defined model file is `deqmod1_e.tf6` which is now listed.

```
%
  model: von Bertalannfy growth model

 differential equation: f(1) = dy(1)/dx
                             = p(1)*y(1)^p(2) - p(3)*y(1)^p(4)

              Jacobian: j(1) = df(1)/dy(1)
                             = p(1)*p(2)*y(1)^(p(2) - 1.0)
                               -p(3)*p(4) y(1)^(p(4) - 1.0)

    initial condition: y0(1) = p(5)
 %
 1 equation
 differential equation
 5 parameters
 %
 begin{expression}
 f(1) = p(1)y(1)^p(2) - p(3)y(1)^p(4)
 end{expression}
 %
 begin{expression}
 A = p(2) - 1.0
 B = p(4) - 1.0
 j(1) = p(1)p(2)y(1)^A - p(3)p(4)y(1)^B
 end{expression}
 %

 begin{limits}
 0  1.0       3
 0  0.6666667 3
 0  1.0       3
 0  1.0       3
 0  0.01      1
 end{limits}

 begin{range}
 121
 0
 10
 end{range}
```

12.3 The Lotka-Volterra predator-prey equations

The Lotka-Volterra predator-prey system of two differential equations is justly famous. Given a prey species y_1, e.g. rabbits, and a predator species y_2, e.g. foxes, as functions of time, x say, then a plausible simple system is the following.

$$\frac{dy_1}{dx} = p_1 y_1 - p_2 y_1 y_2$$
$$\frac{dy_2}{dx} = -p_3 y_2 + p_4 y_1 y_2$$

Starting in a situation where the population of foxes is at a low ebb then the rabbit population will grow almost exponentially until they provide ample food for the foxes to multiply, and so on.

It is assumed that the parameters $p_1, p_2, p_3,$ and p_4 and also the populations y_1 and y_2 are nonnegative so there are clearly the two stable states

$$y_1 = 0$$
$$y_2 = 0, \text{ and}$$
$$y_1 = p_3/p_4$$
$$y_2 = p_1/p_2$$

of which only the second is of interest.

As the system is autonomous, i.e., with no independent variable on the right hand sides then, apart for the meaningful singular stable state referred to, it is possible to eliminate x and consider the derivative

$$\frac{dy_1}{dy_2} = \frac{p_1 y_1 - p_2 y_1 y_2}{-p_3 y_2 + p_4 y_1 y_2}.$$

In other words, given an initial point $y_1(0) = \alpha$ and $y_2(0) = \beta$ in y_1, y_2 phase space, then the shape of the plot of $f(x, y_1, y_2|\alpha, \beta) = 0$ is determined uniquely as a pseudo-elliptical type of orbit around the singularity at the center $(p_3/p_4, p_1/p_2)$.

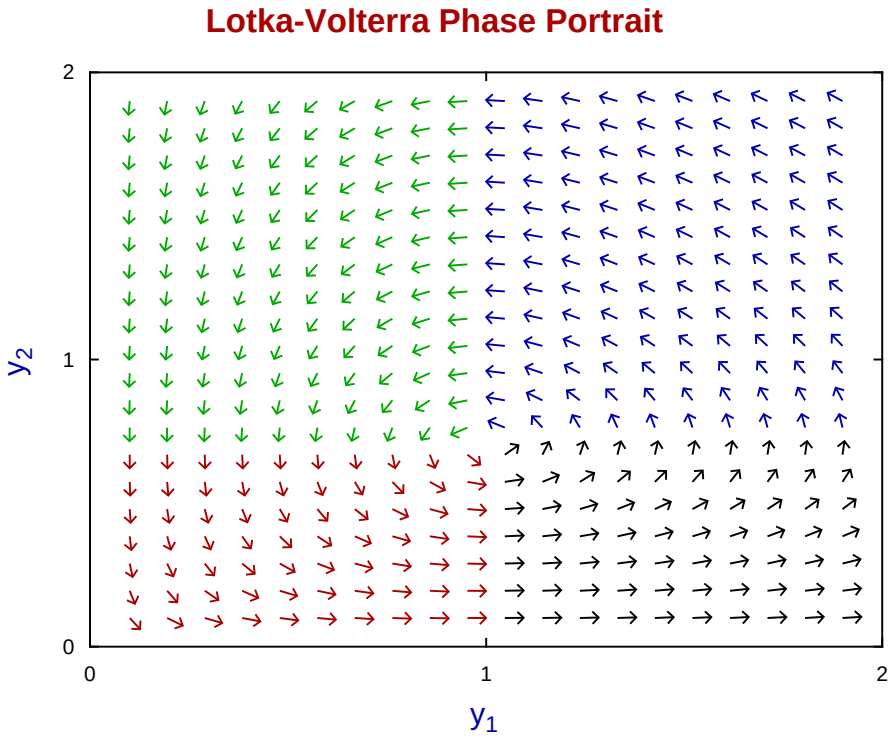
This provides two ways to study the qualitative behavior of this system of equations.

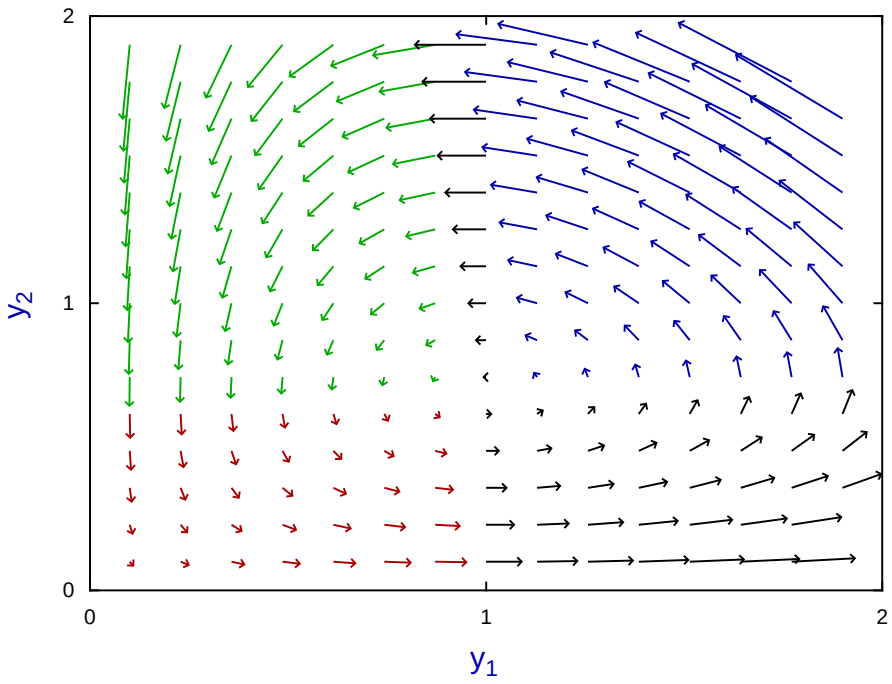
1. **A vector field**
   A grid of points is selected and at each point an arrow is drawn to indicate the direction of dy_2/dy_1. Such a vector field is easy to create, and can be further enhanced by coloring the arrows to indicate direction, as well as altering the size of the stem of the arrows in proportion to the size of the derivatives.

2. **A collection of orbits**
   A set of initial conditions is chosen then the orbits are plotted and subsequently collected together to form a composite graph. This can be more tedious to do, but perhaps the result is easier to understand in terms of the way that the populations fluctuate depending on the initial conditions.

Vector field representation of the phase plane

The next graph is the default phase portrait diagram using colour to emphasize changes in direction while the following one demonstrates the option to increase arrow length in proportion to the absolute size of the derivative.

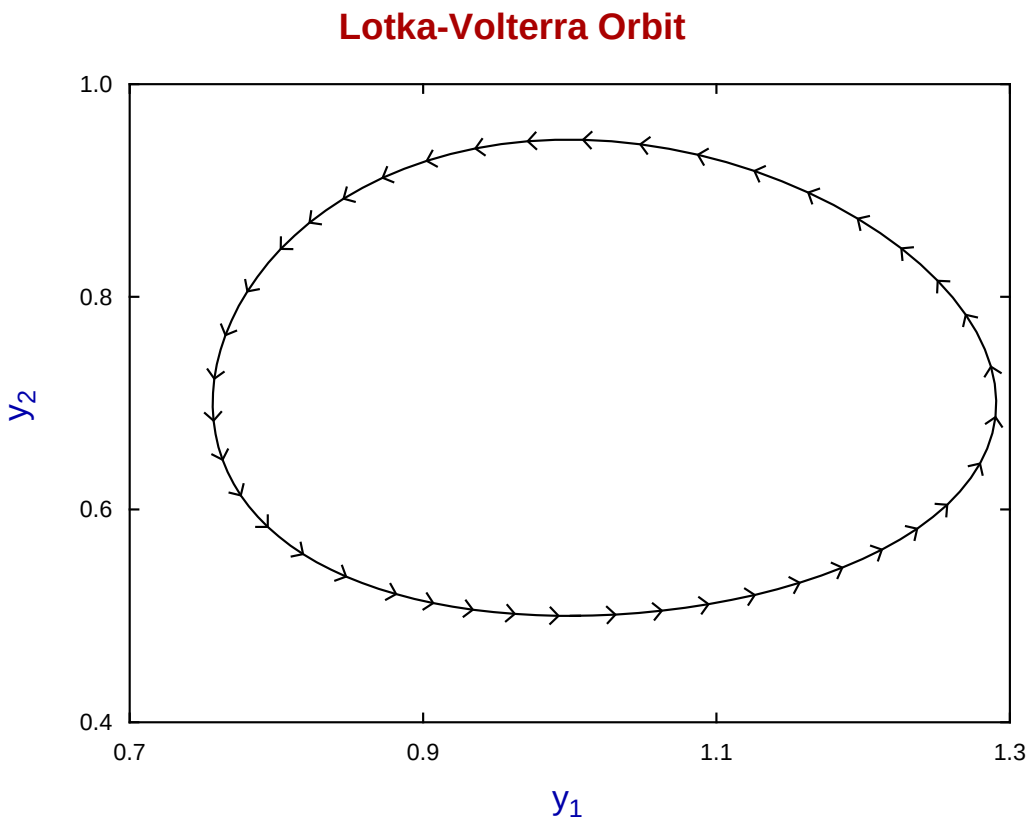




Note that the base of the arrow is at coordinates y_1, y_2 but, because the tips of the arrows will exceed the range set by these coordinates it is easiest to use ranges that will allow for sensible labels for the axes. In this case the range selected for the portrait was $0.1 \leq y_1 \leq 1.9$ and $0.1 \leq y_2 \leq 1.9$ resulting in a plot with integer labels. Also the density of arrows must be considered when plotting arrows with sizes proportional to the length of the derivative vectors as well as the proportionality factor which, in this case, was 0.1.

Plotting orbits

Once a model has been selected and parameters defined then clearly the trajectory will depend on the initial conditions. In the case of the Lotka-Volterra equations there is a point of stable equilibrium, and the equations are autonomous so the orbits will surround this critical point as shown in the next plot.



In order to compose a graph with several orbits as an alternative to the simple vector-field phase portrait the following actions are required.

1. Fix the parameter values that do not represent initial conditions. In this case these would be $p_1, p_2, p_3,$ and p_4.

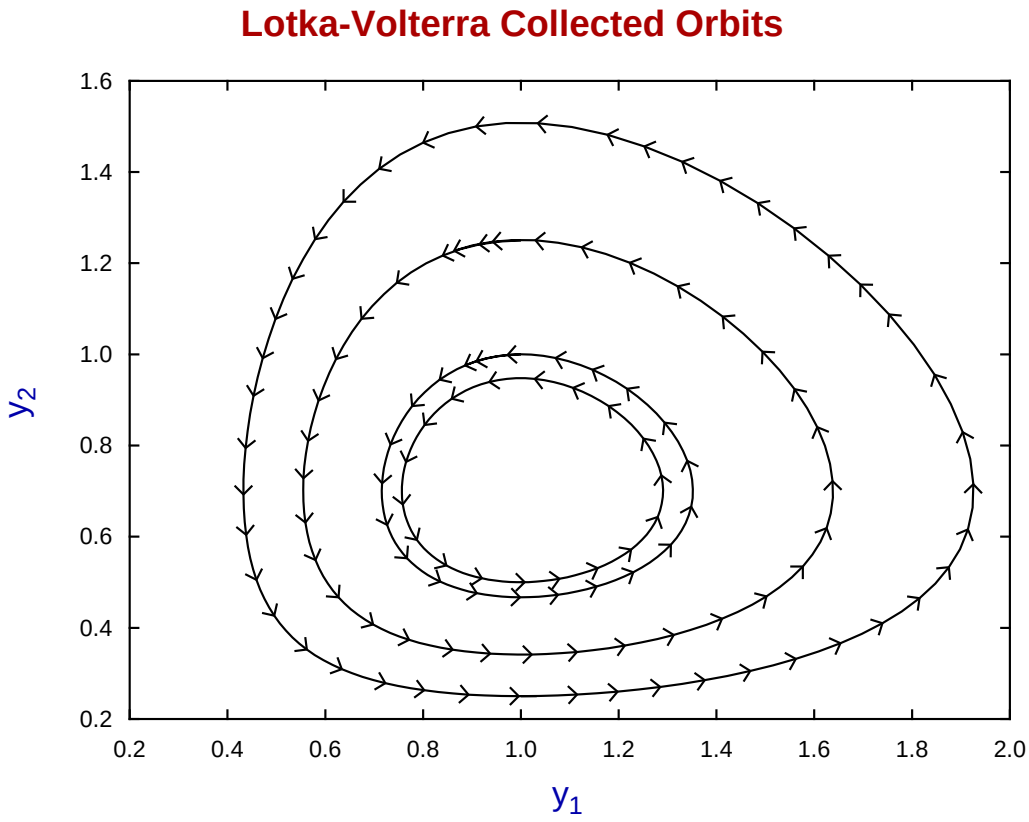
2. Alter the initial conditions, that is parameters p_5 and p_6.

3. When a satisfactory orbit has been constructed where a complete circuit of the equilibrium point has been completed the orbit can be archived.

4. These orbit files will be `f$orbits.001, f$orbits.002`, etc. created in the folder `\User\Documents\simfit\res`

5. These can be retrieved as required to create a composite graph.

To illustrate this a set of orbits was created by maintaining $y_1(0) = 1.0$ then varying $y_2(0)$ over the range 0.25, 0.5, 1.0, 1.25 archiving the orbits each time then using the facility to import stored orbits. The resulting collection of archived orbits is displayed in the next figure.



Curve fitting

In order to fit data sets using systems of differential equations the following steps are required.

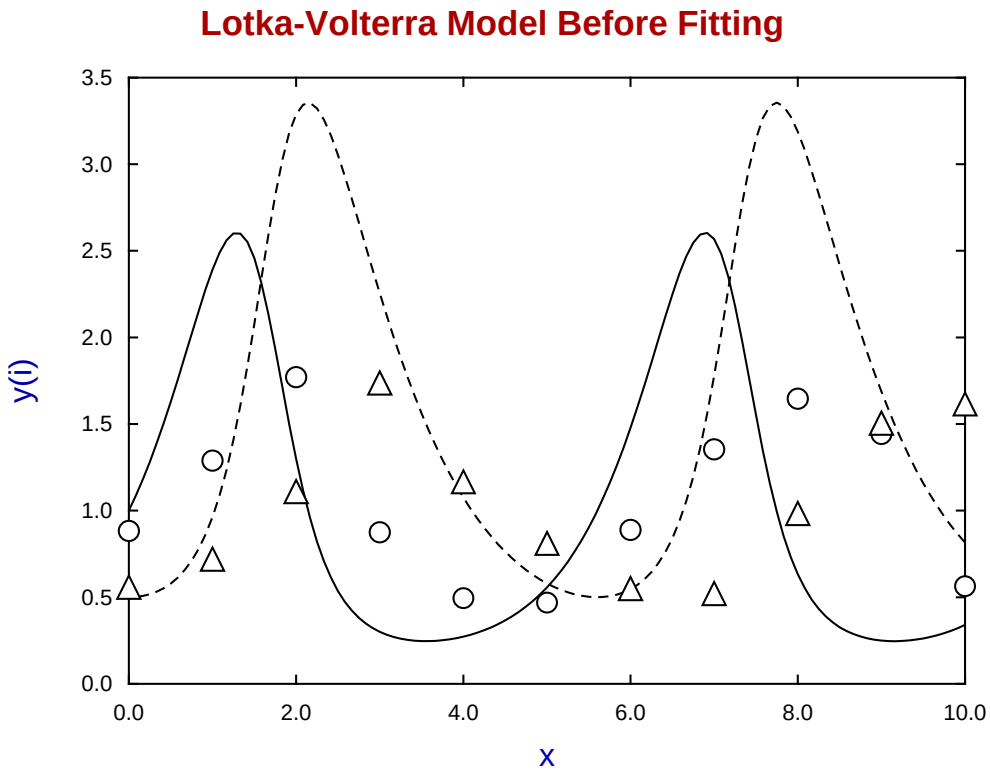
1. Data sets must be prepared in one of two formats, that is
   Column 1 = $x(i)$, Column 2 = $y_j(i)$, or
   Column 1 = $x(i)$, Column 2 = $y_j(i)$, Column 3 = $s_j(i), i = 1, 2, \ldots, n_j$ for $j = 1, 2$
2. Each data set must have $x(i)$ in nondecreasing order for $i = 1, 2, \ldots, n_j$ and standard errors $s_j(i)$ must be accurate estimates if weighting is to be used.
3. Ideally the collected data file should be referenced by a library file if possible where it is understood that if data sets are missing for a component this will be indicated by a percentage sign %.
4. Starting estimates and limits should be appended to the first data file.

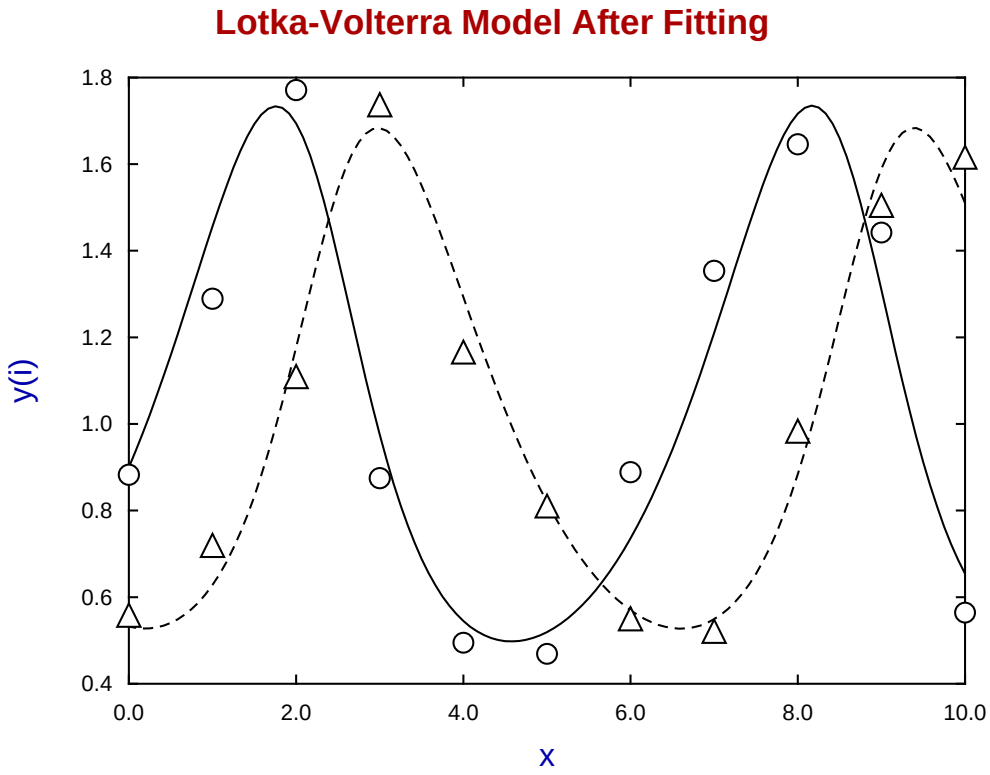
Here, for example is the default library file `deqsol.tfl` for the SimFIT program **deqsol**.

```
deqsol (library file) to fit Lotka-Volterra eqns. (2-eqns/menu item 2)
lv1.tf1
lv2.tf1
```

Note that, in general, the two filenames supplied in such a library file must be fully qualified filenames. This exception is because SimFIT recognizes the two files `lv1.tf1` and `lv2.tf1` as default test files.

The next two graphs illustrate first overlaying the starting curve on the data, and then the best-fit curves resulting from fitting the system of differential equations, followed by the parameter estimates, then goodness of fit analysis.


Lotka-Volterra Model Before Fitting
y(i)
x
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0.0
2.0
4.0
6.0
8.0
10.0


Lotka-Volterra Model After Fitting
y(i)
x
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.0
2.0
4.0
6.0
8.0
10.0

Best-fit parameters for Lotka-Volterra model

Number	Low-Limit	High-Limit	Parameter	Std. Error	Lower95%cl	Upper95%cl	p
1	0.0	2.0	1.08447	0.129892	0.809116	1.35983	0.0000
2	0.0	2.0	1.08901	0.123310	0.827608	1.35042	0.0000
3	0.0	2.0	0.93975	0.115146	0.695649	1.18384	0.0000
4	0.0	2.0	0.94854	0.108451	0.718635	1.17845	0.0000
5	0.0	2.0	0.89979	0.055012	0.783168	1.01641	0.0000
6	0.0	2.0	0.53189	0.043215	0.440273	0.62350	0.0000

Analysis of residuals: SSQ	0.18589	
Average % coefficient of variation	10.33%	
R^2, i.e. correlation coefficient$(theory, data)^2$	0.9547	
Largest Absolute relative residual	18.81%	
Smallest Absolute relative residual	1.10%	
Average Absolute relative residual	8.11%	
Absolute relative residual in range 0.1-0.2	40.91%	
Absolute relative residual in range 0.2-0.4	0.00%	
Absolute relative residual in range 0.4-0.8	0.00%	
Absolute relative residual > 0.8	0.00%	
Number of negative residuals (m)	13	
Number of positive residuals (n)	9	
Number of runs observed (r)	14	
P(runs $\leq r$: given m and n)	0.9031	
5% lower tail point	7	
1% lower tail point	6	
P(runs $\leq r$: given m plus n)	0.9054	
P(signs $\leq$ least number observed)	0.5235	
Durbin-Watson test statistic	2.6043	>2.5, -ve serial correlation?
Shapiro-Wilks W statistic	0.9515	
Significance level of W	0.3385	
Akaike AIC statistic	-93.020	
Schwarz SC statistic	-86.474	
Verdict on goodness of fit	very good	

User-defined models

Model files have three mandatory sections separated by the escape character % followed an optional section which is very important when constructing models for differential equations, summarized as follows.

%

Title section

Arbitrary text used to describe the equations but must not exceed 24 lines

%

2) Summary section

The number of equations, type of equations, and number of parameters

%

3) Equation section

The equations and Jacobian if required

%

4) Optional additional information

The SimFiT default test file `deqmod2_e.tf2` for the Lotka-Volterra scheme is displayed next.

```
 %
 Example of a user supplied pair of differential equations
 file: deqmod2_e.tf2 (with appended parameter limits data)
 model: Lotka-Volterra predator-prey equations
 differential equations: f(1) = dy(1)/dx
                              = p(1)*y(1) - p(2)*y(1)*y(2)
                         f(2) = dy(2)/dx
                              = -p(3)*y(2) + p(4)*y(1)*y(2)
              jacobian: j(1) = df(1)/dy(1)
                              = p(1) - p(2)*y(2)
                        j(2) = df(2)/dy(1)
                              = p(4)*y(2)
                        j(3) = df(1)/dy(2)
                              = -p(2)*y(1)
                        j(4) = df(2)/dy(2)
                              = -p(3) + p(4)*y(1)
    initial condition: y0(1) = p(5), y0(2) = p(6)
 Note: the last parameters must be y0(i) in differential equations
 %
 2 equations
 differential equation
 6 parameters
 %
begin{expression}
f(1) = p(1)y(1) - p(2)y(1)y(2)
f(2) = -p(3)y(2) + p(4)y(1)y(2)
end{expression}
%
begin{expression}
j(1) = p(1) - p(2)y(2)
j(2) = p(4)y(2)
j(3) = -p(2)y(1)
j(4) = -p(3) + p(4)y(1)
end{expression}
%

 begin{limits}
 0  1.0  3
 0  1.0  3
 0  1.0  3
 0  1.0  3
 0  1.0  3
 0  0.5  3
 end{limits}

 begin{range}
 121
 0
 10
 end{range}
```

12.4 The epidemic differential equations

The standard epidemic differential equation model considers the interaction between susceptible, infected, and resistant individuals as a development of a Lotka-Volterra type of interaction.

For $y_1(x)$ susceptible, $y_2(x)$ infected, and $y_3(x)$ resistant individuals in the population as functions of time x the scheme is as follows

$$\begin{aligned} f_1 &= \frac{dy_1}{dx} = -p_1 y_1 y_2 \\ f_2 &= \frac{dy_2}{dx} = p_1 y_1 y_2 - p_2 y_2 \\ f_3 &= \frac{dy_3}{dx} = p_2 y_2 \end{aligned}$$

where p_1 and p_2 are positive parameters, and the additional parameters $p_3 = y_1(0)$, $p_4 = y_2(0)$ and $p_5 = y_3(0)$ are the initial conditions. Here the number of susceptible individuals declines as a result of contact between themselves and those already infected, the number of infected individuals increases as a result of such contacts but declines resulting from the development of resistance, while resistance increases in proportion to the number of those infected. Note that the overall population $Y(x) = y_1(x) + y_2(x) + y_3(x)$ remains static as will be obvious from the conservation equation

$$\begin{aligned} \frac{dY}{dx} &= \frac{dy_1}{dx} + \frac{dy_2}{dx} + \frac{dy_2}{dx} \\ &= 0 \end{aligned}$$

while the Jacobian matrix required for stiff systems is defined as follows.

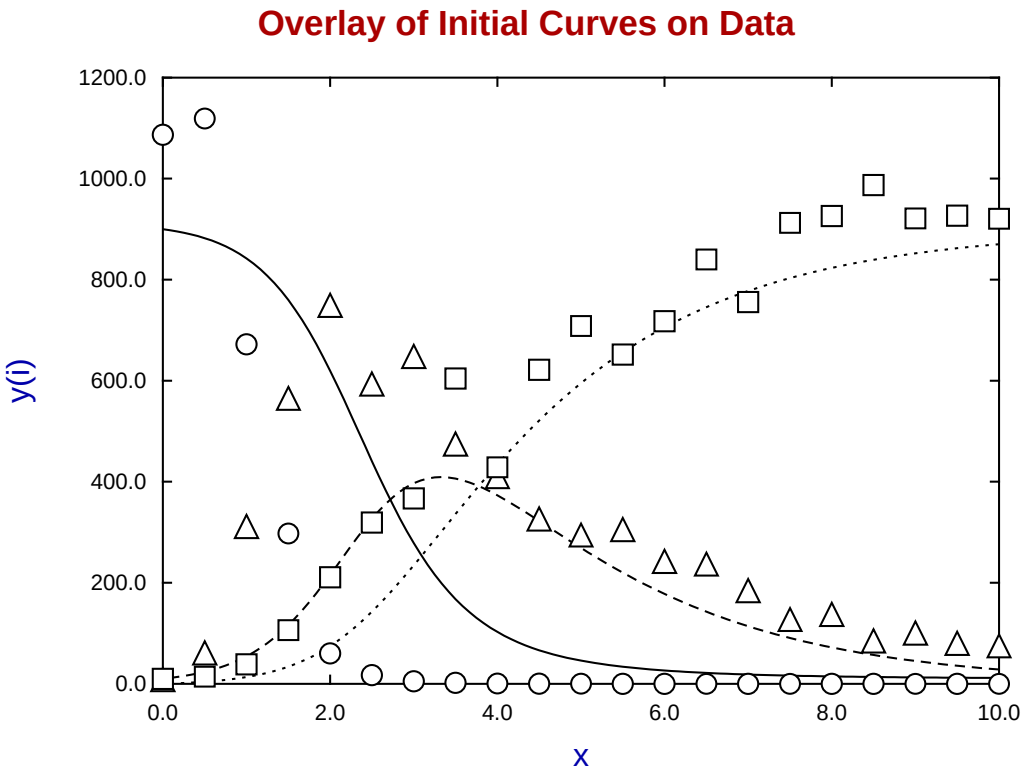
$$\frac{\partial f_i}{\partial y_j} = \begin{pmatrix} -p_1 y_2 & -p_1 y_1 & 0 \\ p_1 y_2 & p_1 y_1 - p_2 & 0 \\ 0 & p_2 & 0 \end{pmatrix}$$

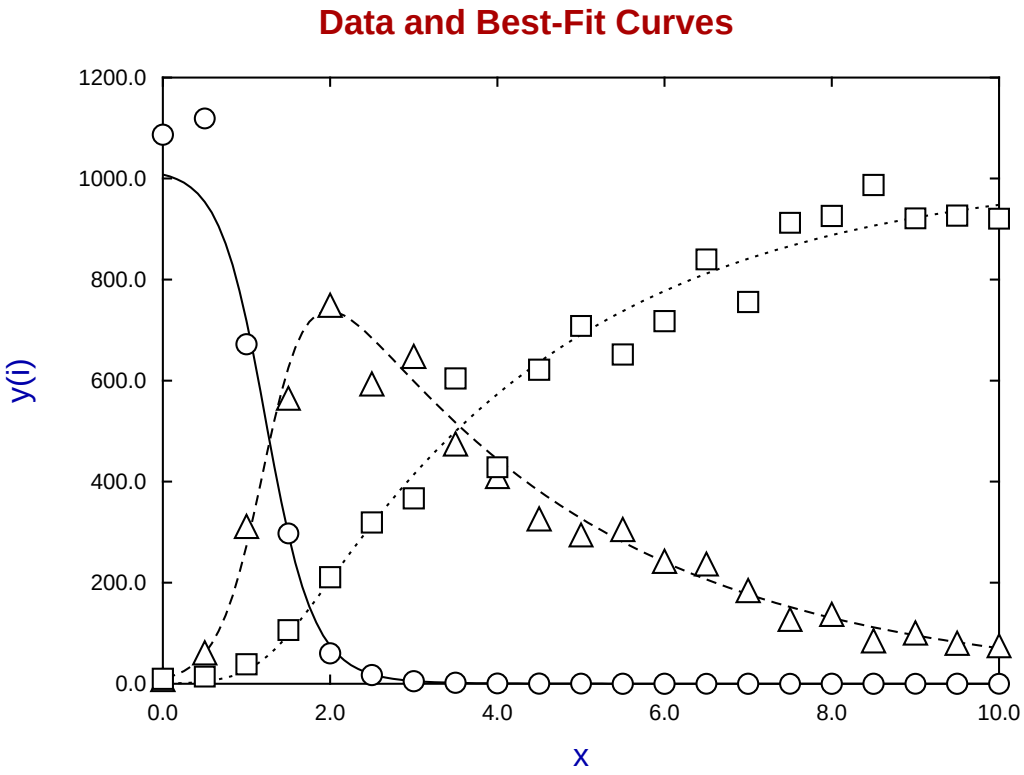
Using SIMFIT program **deqsol** to fit the data contained in the default library file `epidemic.tfl` leads to the following results.

Best-fit parameters for the epidemic model

Number	Low-Limit	High-Limit	Parameter	Std. Error	Lower95%cl	Upper95%cl	p	
1	0	1	0.00389	0.00036	0.00355	0.00460	0.0000	
2	0	2	0.30718	0.00919	0.28800	0.32558	0.0000	
3	400	1400	1008.14	12.9747	982.169	1034.11	0.0000	
4	0	50	10.2413	4.48755	1.25851	19.2241	0.0262	
5	0	50	0.09828	0.09758	-0.09705	0.29361	0.3180	*

Note that the initial conditions were also fitted, and the nonzero p values for the corresponding estimates $\hat{p}_4$ and $\hat{p}_5$ indicates the unreliable nature of the fit to these parameters. The overlay before fitting, and then the final best-fit curve are displayed next, followed by the code for this model contained in the model file `deqmod3.tf1`.


Overlay of Initial Curves on Data
1200.0
1000.0
800.0
600.0
400.0
200.0
0.0
y(i)
0.0
2.0
4.0
6.0
8.0
10.0
x


Data and Best-Fit Curves
1200.0
1000.0
800.0
600.0
400.0
200.0
0.0
y(i)
0.0
2.0
4.0
6.0
8.0
10.0
x

```
%
differential equations: f(1) = dy(1)/dx = -p(1)y(1)y(2)
                        f(2) = dy(2)/dx =  p(1)y(1)y(2) - p(2)y(2)
                        f(3) = dy(3)/dx =  p(2)y(2)
y(1) = Susceptible, y(2) = Infected, y(3) = Resistant
Jacobian: j(1) = df(1)/dy(1) = -p(1)y(2)
          j(2) = df(2)/dy(1) =  p(1)y(2)
          j(3) = df(3)/dy(1) =  0
          j(4) = df(1)/dy(2) = -p(1)y(1)
          j(5) = df(2)/dy(2) =  p(1)y(1) - p(2)
          j(6) = df(3)/dy(2) =  p(2)
          j(7) = df(1)/dy(3) =  0
          j(8) = df(2)/dy(3) =  0
          j(9) = df(3)/dy(3) =  0
initial condition: y0(1) = p(3), y0(2) = p(4), y0(3) = p(5)
%
3 equations
differential equation
5 parameters
%
begin{expression}
A = p(1)y(1)y(2)
B = p(2)y(2)
f(1) = -A
f(2) = A - B
f(3) = B
end{expression}
%
begin{expression}
C = p(1)y(2)
D = p(1)y(1)
j(1) = -C
j(2) = C
j(3) = 0
j(4) = -D
j(5) = D - p(2)
j(6) = p(2)
j(7) = 0
j(8) = 0
j(9) = 0
end{expression}
%
begin{limits}
  0     0.0025    1
  0     0.5       2
400   900      1400
  0    10        50
  0     0.1      50
end{limits}
begin{range}
121
0
10
end{range}
```

12.5 The recurrent epidemic differential equations

The standard SIR epidemic differential equation model considers the interaction between susceptible, infected, and resistant individuals as a development of a Lotka-Volterra type of interaction. This system of equations can be extended in many ways to include censoring by birth, death, immigration, or emigration as well as including additional covariates and other complicating factors such as seasonal effects and vaccination.

For instance, a simple extension to include incomplete resistance after infection can be included by similar mass action reasoning but, in order to preserve a constant conservation equation, the simple addition of a third parameter p_3 can model births adding to the susceptible sub–population and loss by death from the resistant sub–population according to the following scheme.

For $y_1(x)$ susceptible, $y_2(x)$ infected, and $y_3(x)$ resistant individuals in the population as functions of time x we would then have

$$\begin{aligned} f_1 &= \frac{dy_1}{dx} = -p_1 y_1 y_2 + p_3 \\ f_2 &= \frac{dy_2}{dx} = p_1 y_1 y_2 - p_2 y_2 \\ f_3 &= \frac{dy_3}{dx} = p_2 y_2 - p_3 \end{aligned}$$

where p_1, p_2 and p_3 are positive parameters, and the additional parameters $p_4 = y_1(0), p_5 = y_2(0)$ and $p_6 = y_3(0)$ are the initial conditions. Here the number of susceptible individuals declines as a result of contact between themselves and those already infected, the number of infected individuals increases as a result of such contacts but declines resulting from the development of resistance, while resistance increases in proportion to the number of those infected. Note that the overall population $Y(x) = y_1(x) + y_2(x) + y_3(x)$ remains static as will be obvious from the derivative of the conservation equation

$$\begin{aligned} \frac{dY}{dx} &= \frac{dy_1}{dx} + \frac{dy_2}{dx} + \frac{dy_2}{dx} \\ &= 0 \end{aligned}$$

while the Jacobian matrix required for stiff systems is defined as follows.

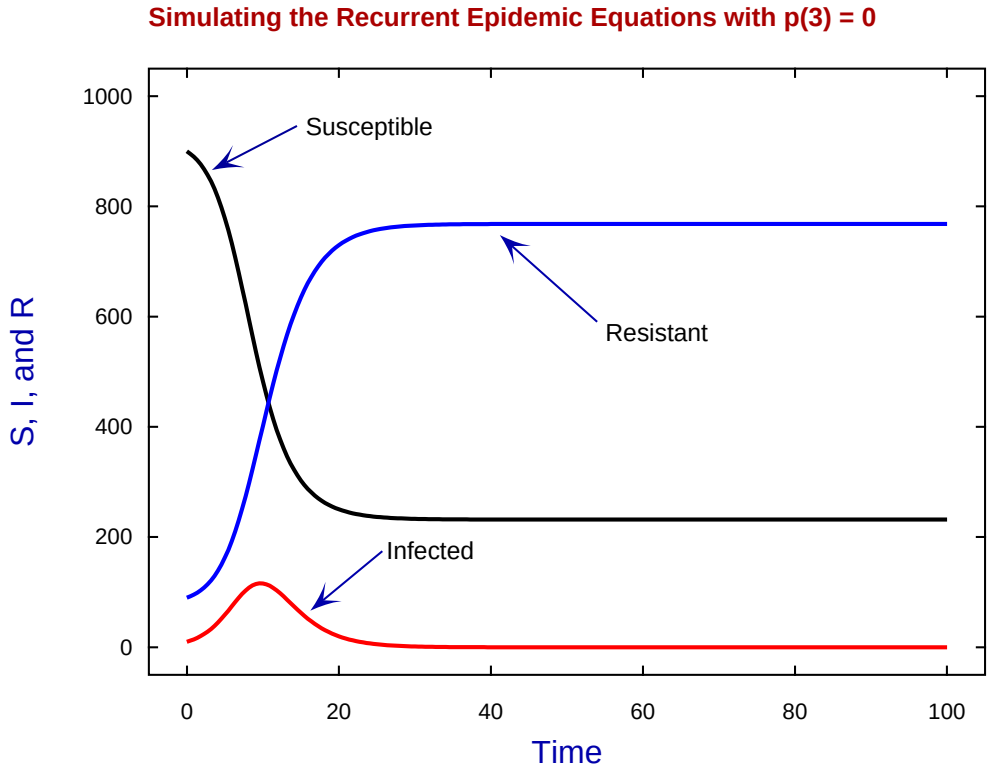
$$\frac{\partial f_i}{\partial y_j} = \begin{pmatrix} -p_1 y_2 & -p_1 y_1 & 0 \\ p_1 y_2 & p_1 y_1 - p_2 & 0 \\ 0 & p_2 & 0 \end{pmatrix}$$

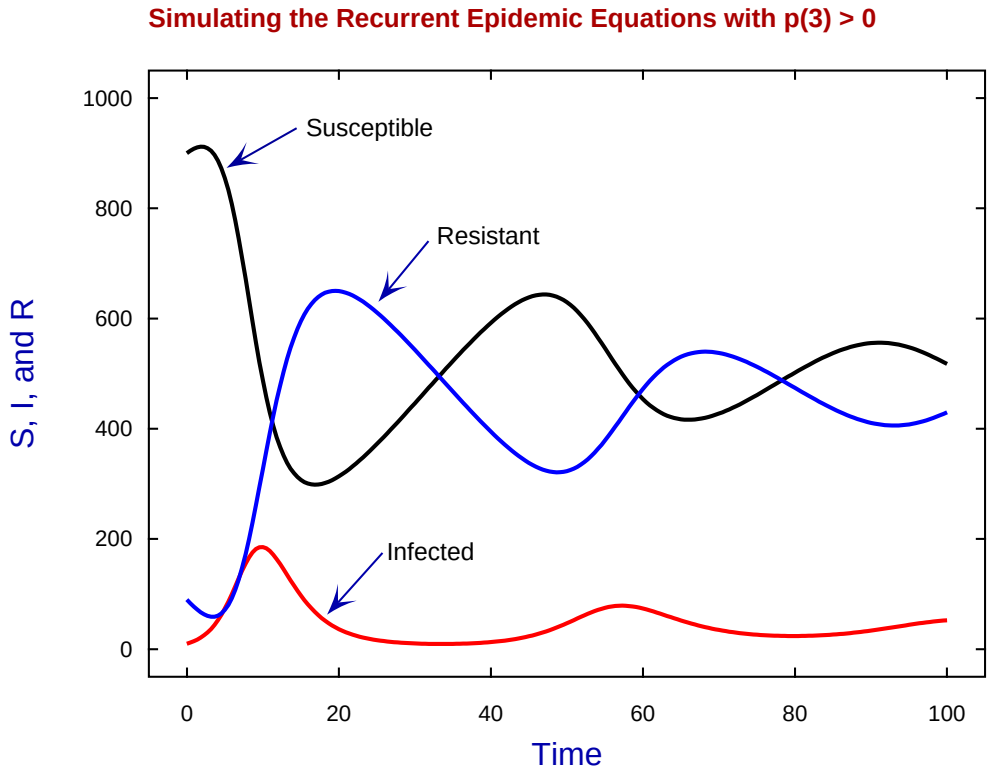
When $p_3 > 0$ it can be shown that if $p_1 p_3 / p_2^2 < 4$ there is light damping to an equilibrium points with the system converging as follows

$$\begin{aligned} y_1 &\approx \frac{p_2}{p_1} \\ y_2 &\approx \frac{p_3}{p_2} \end{aligned}$$

at large time values, i.e., the endemic state of partial herd immunity.

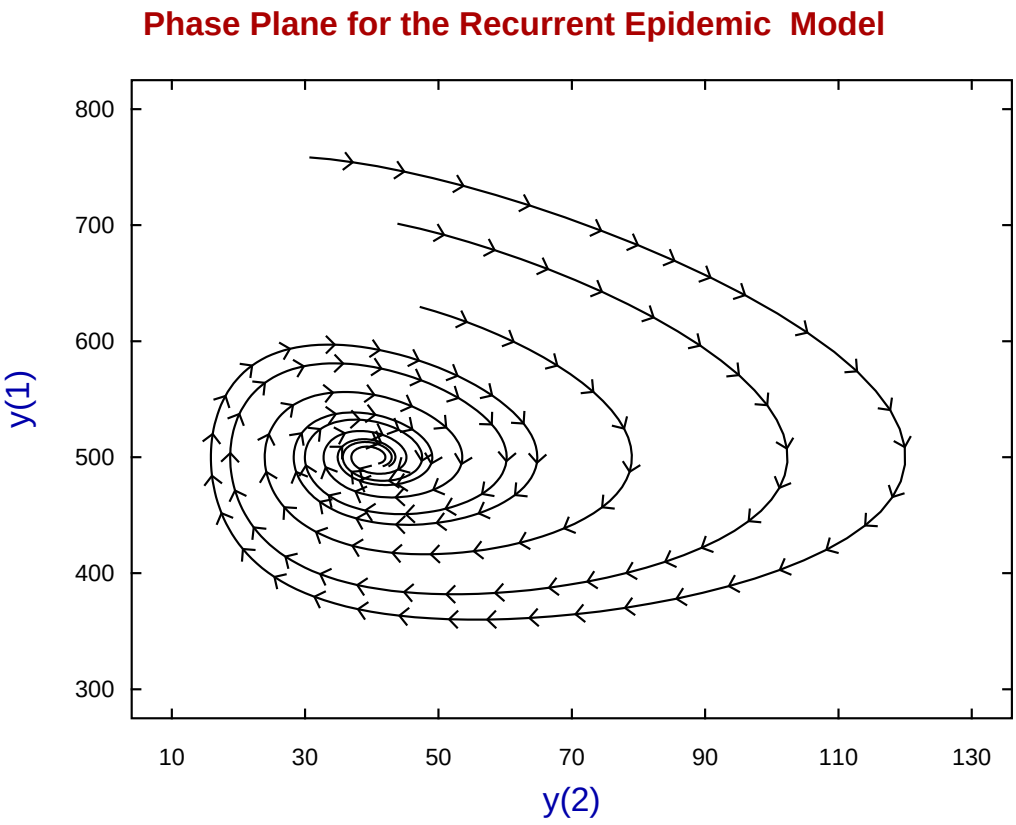
The next two figures illustrate the non–oscillating behaviour when $p_3 = 0$ compared to the damped oscillations when $p_3 > 0$ and so establishing this set of equations being a model for a recurrent epidemic, i.e. where acquired resistance does not preclude subsequent re–infection.





These figures were generated using SIMFIT program **deqsol** with the built–in defaults and initial conditions for $p(i)$ where $i = 1, 6$.

A convenient way to display this oscillatory property of the recurrent epidemic model is to plot the phase plane as shown next as it is more convincing when visualized such a plot than staring at mere algebra.

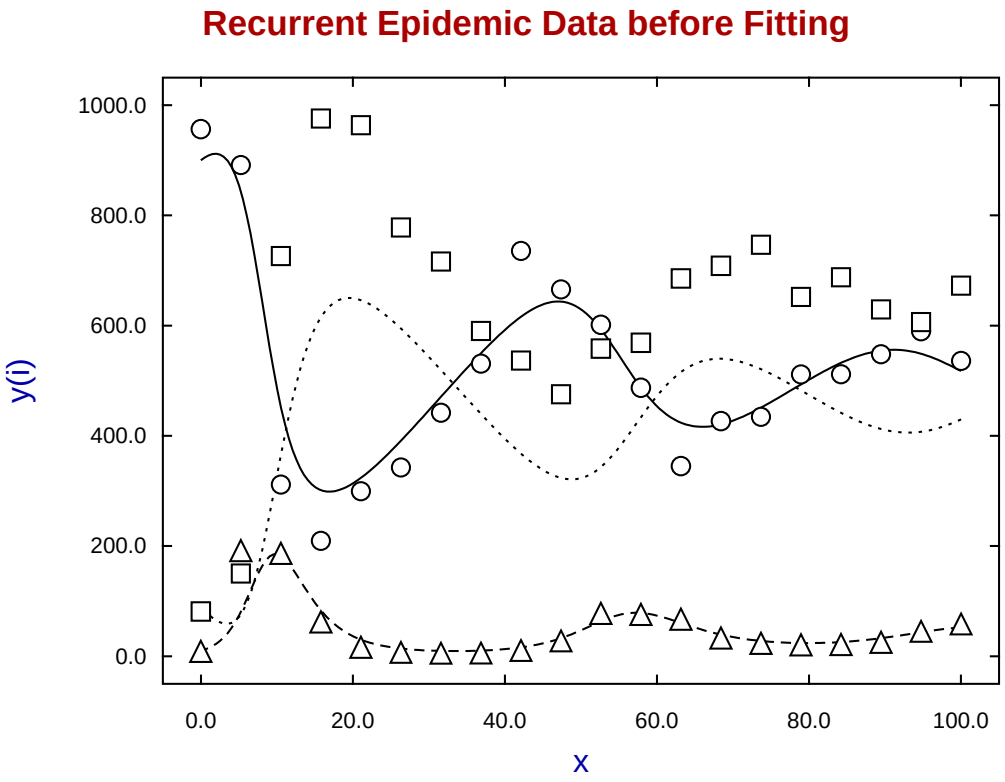


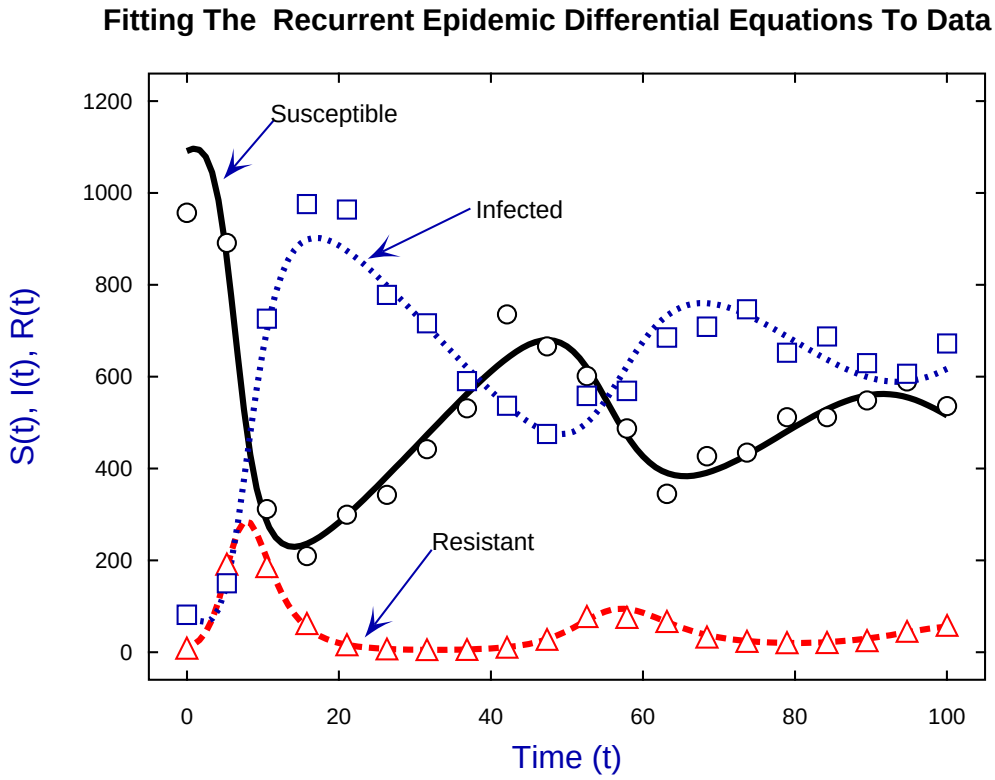
Such phase plane diagrams are easily constructed using the SIMFIT program **deqsol** using the following steps.

1. Fix parameters $p_i, i = 1, 6$ and also the starting and ending points for the integration and number of points required then integrate.
2. Plot the orbit then choose the option to store the orbit.
3. Repeat the process keeping the parameters $p_i, i = 1, 3$ fixed but varying the initial conditions $p_i, i = 4, 6$ then storing the orbits for each choice of initial conditions.
4. Choose the option to plot the archived orbits that will have been temporarily stored then select the archived orbits required to form a composite phase plane diagram like the above figure.
5. Proceed to sculpture the graph then save

Now we consider fitting the recurrent epidemic data. Here the next two figures illustrate that before fitting was attempted the starting estimates gave profiles far away from the data. However, after constrained nonlinear optimization, a good fit was eventually located.

Actually, after several abortive attempts from numerous starting estimates for the six parameters made no progress, the curve fitting had to make extensive use of the SIMFIT procedures to vary the starting estimates randomly over a range specified by the upper and lower parameter limits until eventually convergence was achieved. It must be emphasized that this use of several random starts using a uniform or normal distribution to sequentially perturb the starting estimates within the limits of constraint is frequently required when the final solution is far removed from the starting estimates and the process of optimization is stalled so that the WSSQ does not change appreciably until the perturbation generates a sensible descent vector.


Recurrent Epidemic Data before Fitting
y(i)
x
1000.0
800.0
600.0
400.0
200.0
0.0
0.0
20.0
40.0
60.0
80.0
100.0


Fitting The Recurrent Epidemic Differential Equations To Data
S(t), I(t), R(t)
Time (t)
Susceptible
Infected
Resistant
1200
1000
800
600
400
200
0
0
20
40
60
80
100

The next table lists the best fit parameters and confidence limits and it will be clear from the p values that the parameters are well determined.

Number	Parameter	Std.Error	Lower95%cl	Upper95%cl	p
1	0.00101	0.00012	0.00099	0.00103	0.0000
2	0.49581	0.00615	0.48348	0.50815	0.0000
3	19.7314	0.17848	19.3736	20.0893	0.0000
4	1089.28	6.91486	1075.42	1103.15	0.0000
5	9.85414	0.19370	9.46580	10.2425	0.0000
6	90.7113	1.08120	88.5436	92.8790	0.0000

Some technical details follow giving more information to assist users who want to develop their own equations for simulating and fitting.

1. To state the obvious: the model to be fitted must be appropriate, the data must be extensive and reasonably accurate, the starting estimates and parameter limits must be sensible, and attention must be paid to any weighting that may be required. The copious goodness of fit tables, residuals analysis, and advice output by SimFIT must be read and appreciated because simulating and fitting nonlinear models is extremely difficult. Usually several runs using randomised starting estimates will be required.

2. SimFIT uses the Open Source programs DVODE to simulate the differential equations using the BDF method by default and the constrained nonlinear optimisation quasi Newton routine LBFGSB. These are bundled as part of the SimFIT package which uses a built–in reverse communication procedure during the optimisation in an attempt to maintain the parameters of order unity at the start each iteration.

3. For those who have a license to access the numerical algorithms group routines (NAG), SimFIT has an interface to the NAG library that can be used instead of the built in SimFIT routines. This is extremely valuable as being able to switch at will between different simulation and particularly optimisation codes can often improve the success of data fitting.

4. The SimFIT built–in interface to the NAG library supports the following tried and tested routines.

   D02CJF, D02EJF for solving systems of nonlinear differential equations and E04KZF, E04YJF, E04UEF, E04UFF for constrained nonlinear optimisation

   However there are documents available on the website showing how SimFIT can be programmed to call any NAG library routine.

5. It must be emphasised that to simulate and fit differential equations users do need to configure the methods used by specifying values to be used to to control step length and convergence criteria.

6. The SimFIT test library file `recurrent.TFL` contains the data described in this document and the model required to simulate and fit the recurrent epidemic is built into program **deqsol**. However it should be pointed out that if a method to integrate stiff systems is required instead of the Runge–Kutta or Adams methods a user–defined Jacobian can be supplied if possible or a Jacobian can be estimated. Note that an incorrect explicit user–supplied Jacobean is much worse than one estimated.

7. SimFIT users can write their own equations for simulation and fitting by just using a text editor. This involves a method whereby the equations can be written using standard mathematical expressions because there is a built–in routine to transform them into reverse Polish. To facilitate the models being used in iterative procedures SimFIT scans the ASCII text file just once then creates a temporary internal stack so the program does not have to re–read the model file again.

8. The file `deqmod3_e.tf3` used to simulate and fit the recurrent epidemic equations as described in this document is listed next.

```
%
Example of a user supplied set of 3 differential equations
file: deqmod3_e.tf1
model: coupled equations for a recurrent epidemic
differential equations: f(1) = dy(1)/dx = -p(1)y(1)y(2) + p(3)
                        f(2) = dy(2)/dx =  p(1)y(1)y(2) - p(2)y(2)
                        f(3) = dy(3)/dx =  p(2)y(2) - p(3)
y(1) = Susceptible, y(2) = Infected, y(3) = Resistant
 jacobian: j(1) = df(1)/dy(1) = -p(1)y(2)
           j(2) = df(2)/dy(1) =  p(1)y(2)
           j(3) = df(3)/dy(1) =  0
           j(4) = df(1)/dy(2) = -p(1)y(1)
           j(5) = df(2)/dy(2) =  p(1)y(1) - p(2)
           j(6) = df(3)/dy(2) =  p(2)
           j(7) = df(1)/dy(3) =  0
           j(8) = df(2)/dy(3) =  0
           j(9) = df(3)/dy(3) =  0
   initial condition: y0(1) = p(3), y0(2) = p(4), y0(3) = p(5)
Note: the last parameters must be y0(i) in differential equations
%
3 equations
differential equation
6 parameters
%
begin{expression}
A = p(1)y(1)y(2)
B = p(2)y(2)
f(1) = -A + p(3)
f(2) = A - B
f(3) = B - p(3)
end{expression}
%
begin{expression}
C = p(1)y(2)
D = p(1)y(1)
j(1) = -C
j(2) = C
j(3) = 0
j(4) = -D
j(5) = D - p(2)
j(6) = p(2)
j(7) = 0
j(8) = 0
j(9) = 0
end{expression}
%
begin{limits}
  0.0      0.001     1.0
  0.0      0.5       2.0
  0.0     30.0     100.0
400.0    900.0    1400.0
  0.0     10.0      50.0
  0.0     90.0     150.0
end{limits}
begin{range}
121
0
200
end{range}
```

13 Graph plotting

Tutorials and worked examples for simulation, curve fitting, statistical analysis, and plotting.
https://simfit.uk
https://simfit.org.uk
https://simfit.silverfrost.com

13.1 Introduction

To create plots you must supply SIMFIT graphical routines with coordinates in one of the following ways.

Method 1
Data are transferred directly by the program you are using.

Method 2
Data are pasted in from the clipboard, or input as a file with a table of coordinates.

Method 3
Data are input as multiple coordinate files by a library file, from your graphics project archive, or by multiple file selection.

Method 4
Data are input as a previously generated metafile to resume editing.

Note that SIMFIT program **simplot** provides test data files to demonstrate most of the available plotting options as well as worked examples.

After editing to change symbols, line-types, titles, legends, colours, etc., you can proceed as follows.

Print directly
Print the current display as a high resolution bit map.

Save Windows quality hardcopy
Choose either *.svg, *.png, *.jpg, or *.emf, but not *.tif, or *.bmp.

Save PostScript quality hardcopy
Archive encapsulated PostScript files (*.eps) then, when required, print from your PS-viewer, or generate *.pdf,*.png, etc., retrospectively. Note that this is the superior way to use SIMFIT graphics, especially for LaTeX users, and it offers numerous additional options. However it requires you to have GhostScript and (optionally) a PS-viewer such as GSview installed.

There are actually four distinct levels of SIMFIT graphics.

1. **Simple graphics**
   This is when data are displayed as a simple plot with limited editing options.
2. **Advanced graphics**
   This allows very extensive editing and numerous additional features such as adding graphical objects, e.g. as extra text, arrows, rectangles, or plotting symbols. In addition metafiles can be created which can be entered retrospectively into program **simplot** to resume editing.
3. **3D graphics**
   Space curves can be plotted and surfaces as wire-nets, skyscrapers, cylinders, or contours.
4. **PostScript graphics**
   This is for experienced users who require the highest quality industry standard graphics as well as additional features such as retrospective editing in a text editor, adding whitespace to stretch graphs without changing the aspect ratio of symbols and text, overlaying graphs, or creating collages.

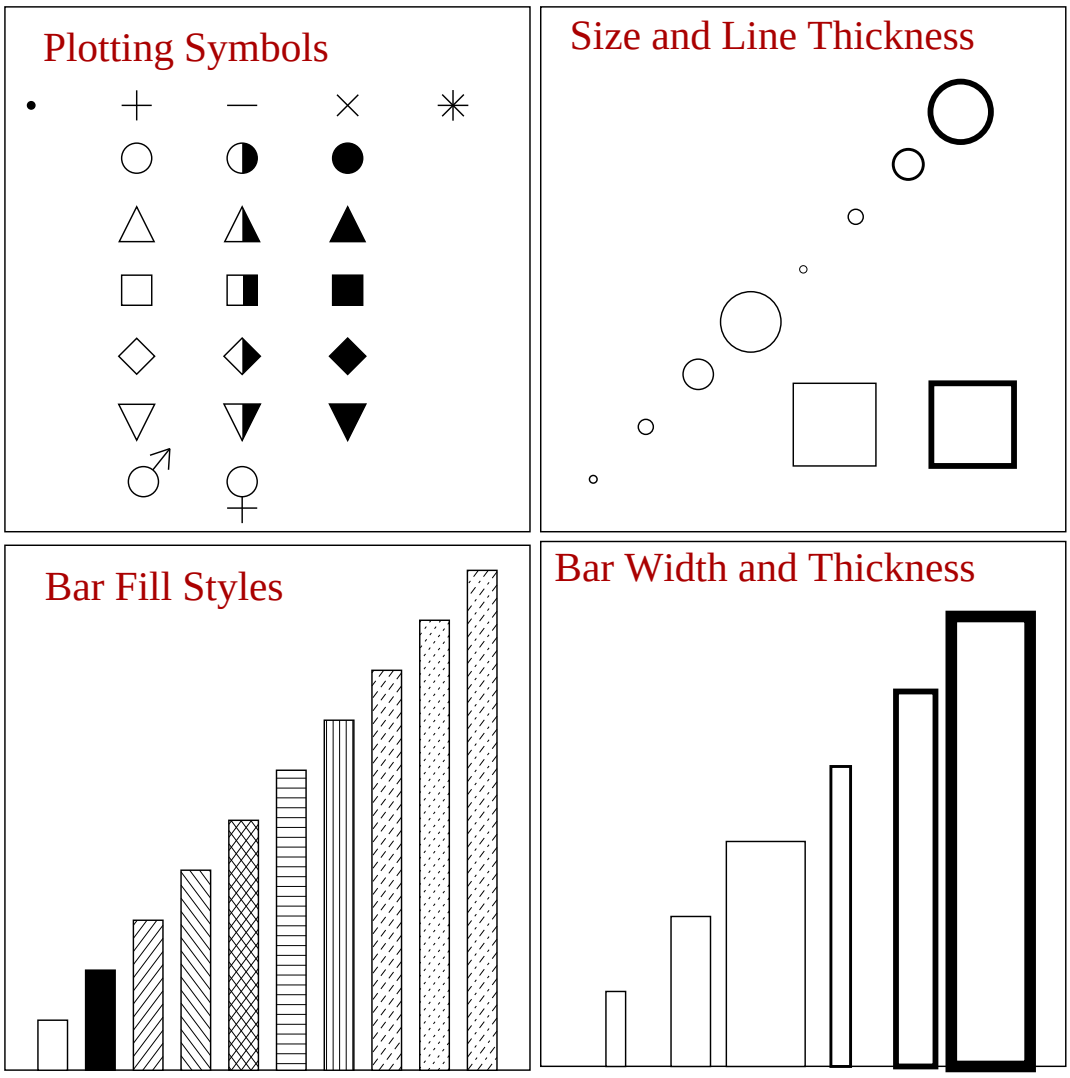
13.2 Simple graphs

Tutorials and worked examples for simulation, curve fitting, statistical analysis, and plotting.
https://simfit.uk
https://simfit.org.uk
https://simfit.silverfrost.com

13.2.1 Lines, symbols, and text

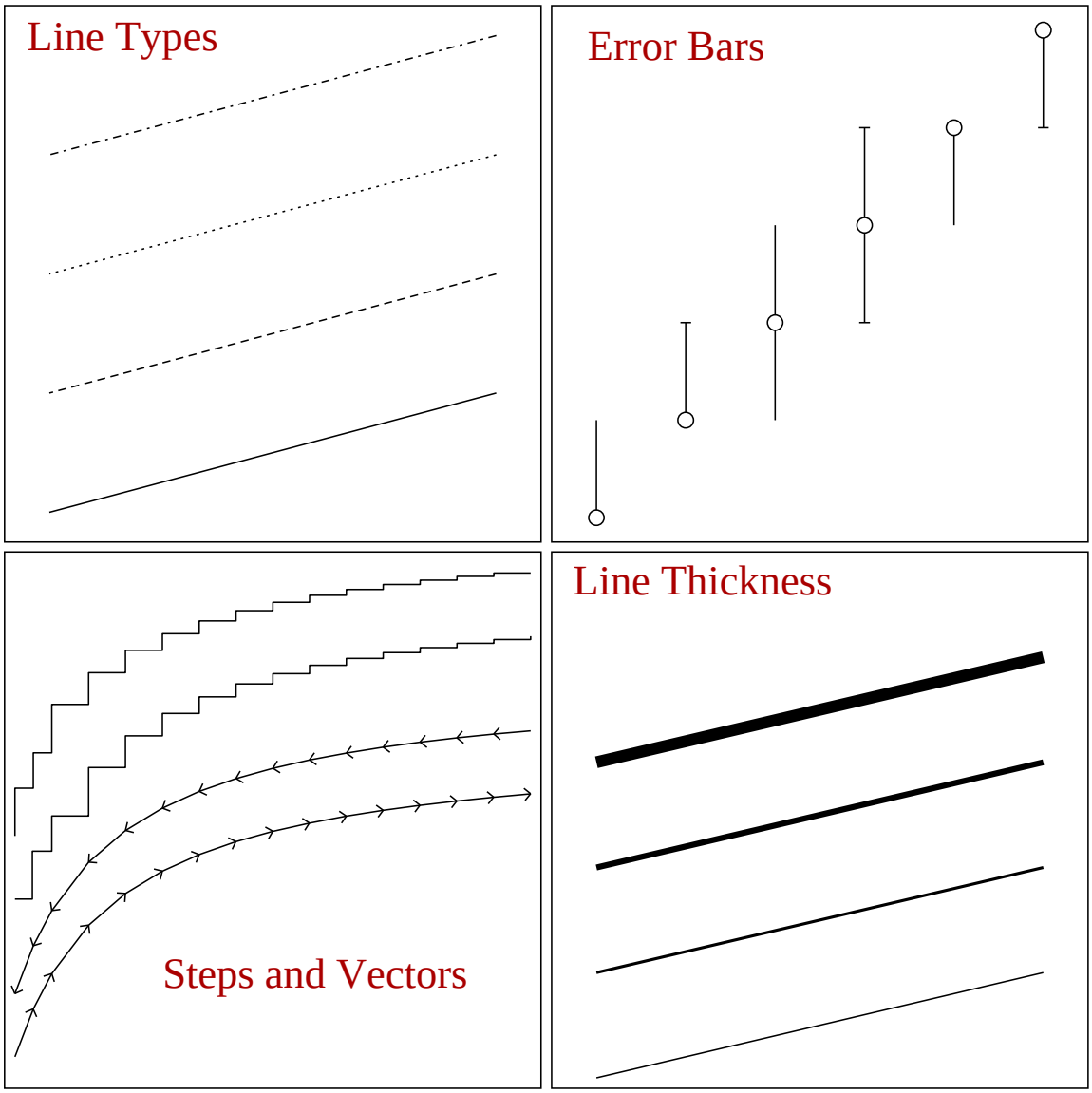
Symbols

The collage below illustrates the following features of SIMFIT graphics.
a) There are 22 plotting symbols plus some outline-only symbols.
b) The size and line thickness used to plot symbols can be varied.
c) There are 10 alternative fill styles for bars in addition to colors.
d) The bar width and line thickness used to plot bars can be varied.



Lines: standard types

There are four standard SIMFIT line types, normal, dashed, dotted and dot-dashed, and error bars can terminate with or without end caps if required, as shown in the collage below.

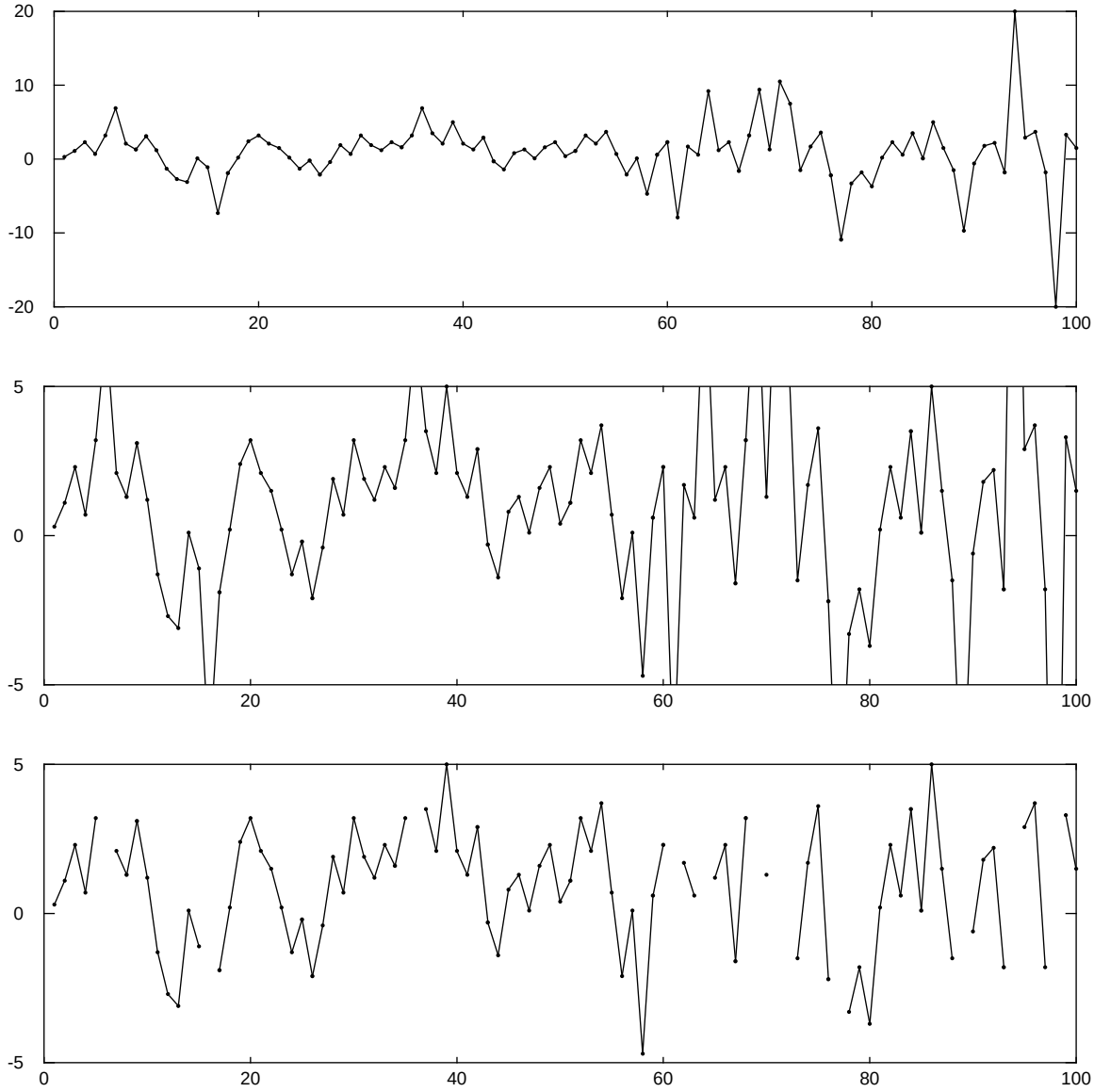


Special effects can be created using stair step lines, which can be used to plot *cdfs* for statistical distributions, or survival curves from survivor functions, and vector type lines, which can be used to plot orbits of differential equations. Note that steps can be first y then x, or first x then y, while vector arrows can point in the direction of increasing or decreasing t, and lines can have variable thickness.

Program **simplot** reads in default options for the sequence of line types, symbol types, colors, barchart styles, piechart styles and labels which will then correspond to the sequence of data files. Changes can be made interactively and stored as graphics configuration templates if required. However, to make permanent changes to the defaults, you configure the defaults from the main SIMFIT configuration option, or from program **simplot**.

Lines: extending to boundaries

The collage below illustrates the alternative techniques available in SIMFIT when the data to be plotted are clipped to boundaries so as to eliminate points that are identified by symbols and also joined by lines.

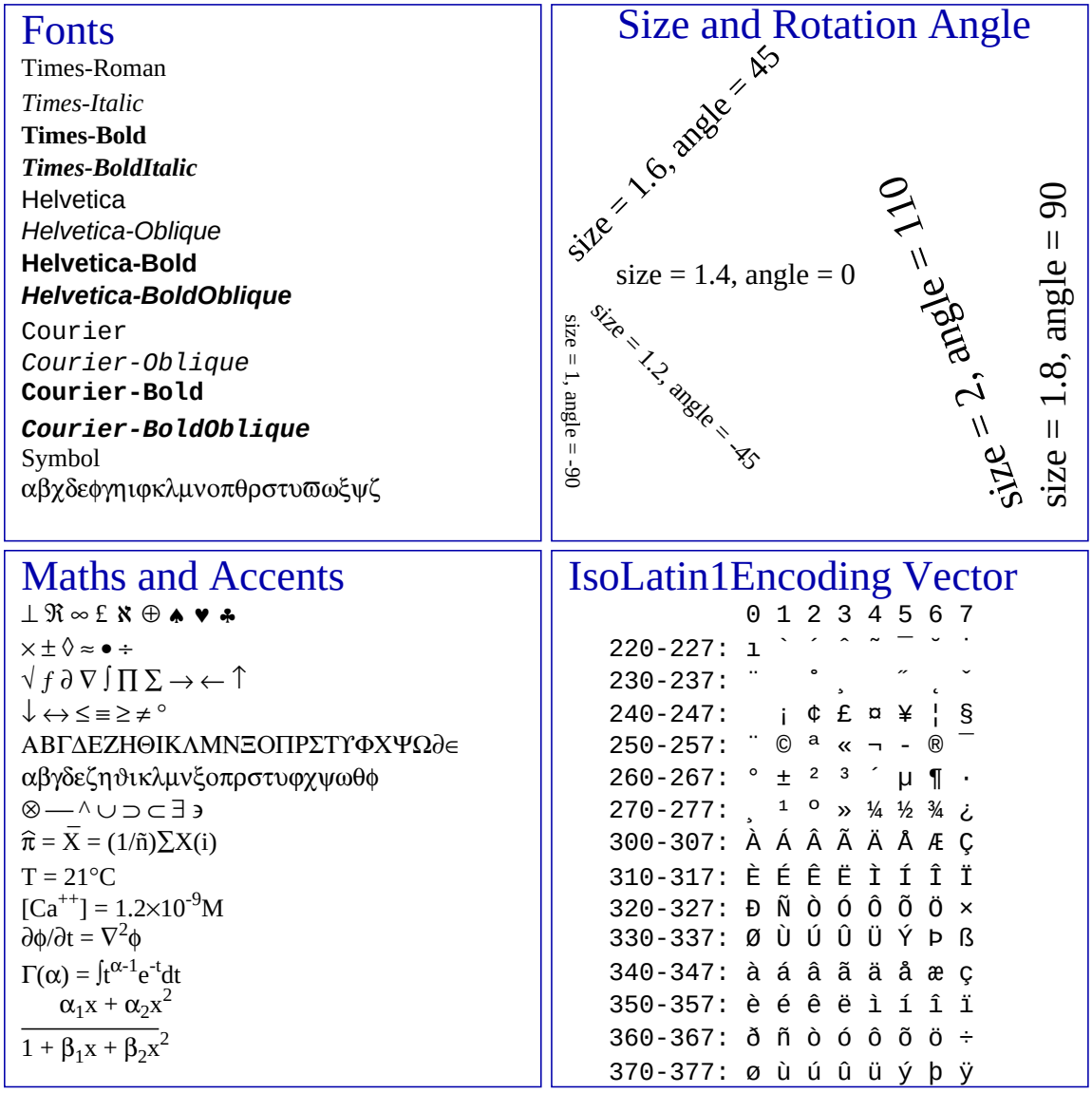


The first graph shows what happens when the test file `zigzag.tf1` was plotted with dots for symbols and lines connecting the points, but with all the data within the boundaries. The second graph illustrates how the lines can be extended to the clipping boundary to indicate the direction in which the next undisplayed symbol is located, while the third figure shows what happens when the facility to extend lines to boundaries is suppressed.

Note that these plots were first generated as .ps files using the flat-shape plotting option, then a PostScript x stretch factor of 2 was selected, followed by the use of GSview to transform to .eps and so recalculate the BoundingBox parameters.

Text

The next collage shows how fonts can be used in any size or rotation and with many nonstandard accents, e.g., $\hat{\theta}$.



Special effects can be created using graphics fonts such as ZapfDingbats, or user-supplied dedicated special effect functions, as described elsewhere. Scientific symbols and simple mathematical equations can be generated, but the best way to get complicated equations, chemical formulas, photographs or other bitmaps into SimFIT graphs is to use PSfrag or **editps**.

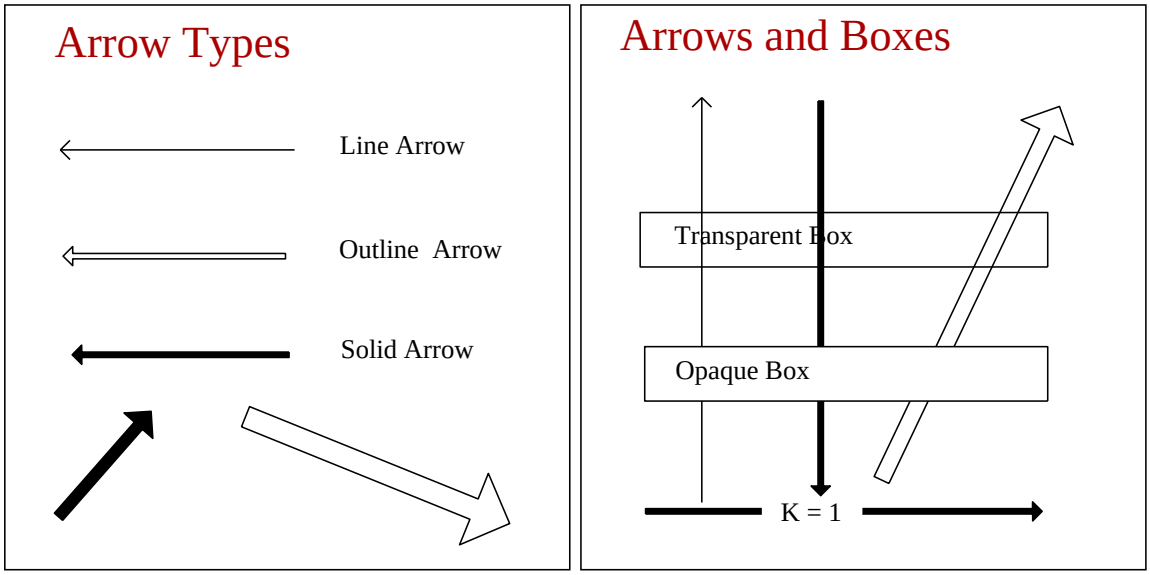
The collage also demonstrates several possibilities for displaying mathematical formulae directly in SimFIT graphs, and it also lists the octal codes for some commonly required characters from the IsoLatin1 encoding. Actually, octal codes can be typed in directly (e.g., `\361` instead of ñ), but note that text strings in SimFIT plots can be edited at two levels: at the simple level only standard characters can be typed in, but at the advanced level nonstandard symbols and maths characters can be selected from a font table. Note that, while accents can be added individually to any standard character, they will not be placed so accurately as when using the corresponding hard-wired characters e.g., from the IsoLatin1 encoding.

Fonts, character sizes and line thicknesses

The fonts, letter sizes, and line thicknesses used in SimFIT graphics are those chosen from the PostScript menu, so, whenever a font or line thickness is changed, the new details are written to the PostScript configuration file `w_ps.cfg`. If the size or thickness selected is not too extreme, it will then be stored as the default to be used next time. However, it should be noted that, when the default sizes are changed, the titles, legends, labels, etc. may not be positioned correctly. You can, of course, always make a title, legend, or label fit correctly by moving it about, but, if this is necessary, you may find that the defaults are restored next time you use SimFIT graphics. If you insist on using an extremely small or extremely large font size or line thickness and SimFIT keeps restoring the defaults, then you can overcome this by editing the PostScript configuration file `w_ps.cfg` and making it read-only. Users who know PostScript will prefer to use the advanced PostScript option, whereby the users own header file can be automatically added to the PostScript file after the SimFIT dictionary has been defined, in order to re-define the fonts, line thicknesses or introduce new definitions, logos plotting symbols, etc.

Arrows

The next figures show that arrows can be of three types: line, hollow or solid and these can be of any size.



However use can be made of headless arrows to create special effects. From this point of view a headless line arrow is simply a line which can be solid, dashed, dotted or dash-dotted. These are useful for adding arbitrary lines. A headless outline arrow is essentially a box which can be of two types: transparent or opaque. Note that the order of priority in plotting is

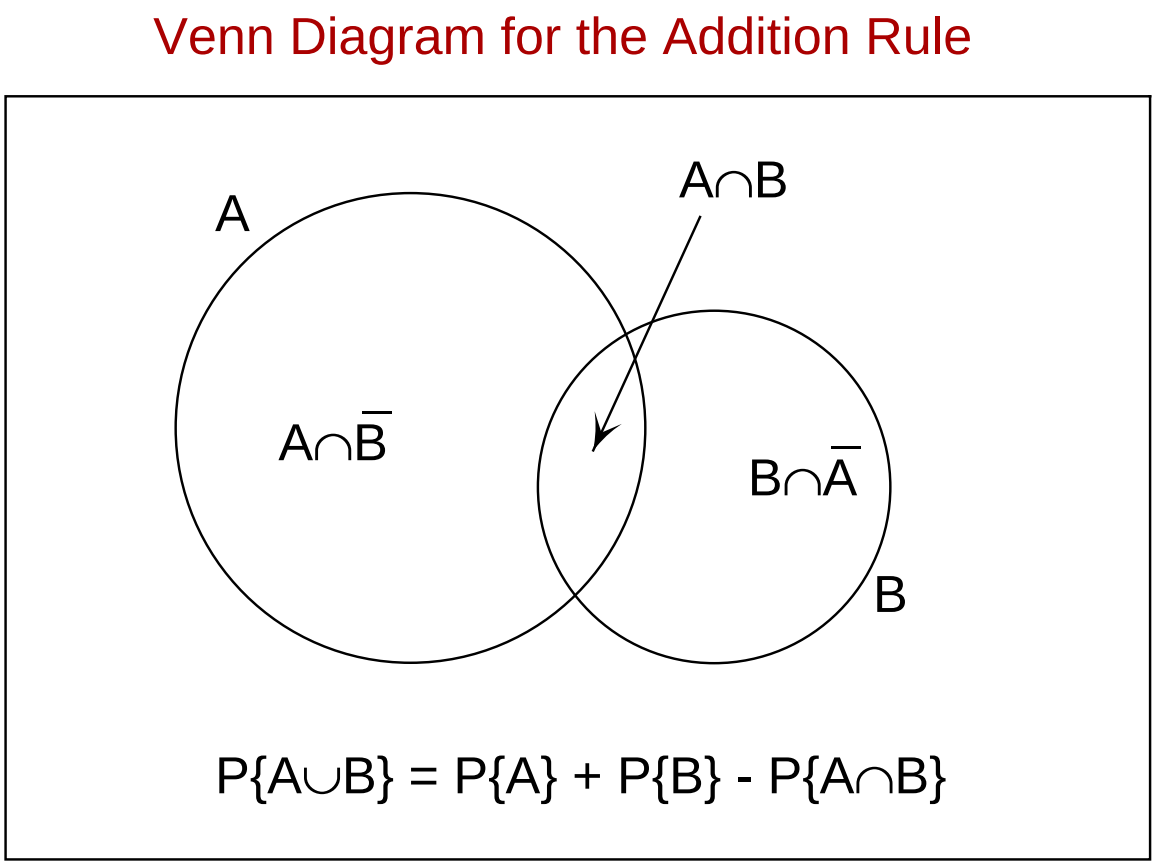
Extra Text > Graphical Objects > Data plotted, titles and legends

and this allows boxes to be used to simply obliterate plotted data or to surround extra text allowing the background to show through.

Transparent boxes, are useful for surrounding information panels, opaque boxes are required for chemical formulae or mathematical equations, while background colored solid boxes can be used to blank out features as shown in the above figures. To surround a text string by a rectangular box for emphasis, position the string, generate a transparent rectangular box, then drag the opposing corners to the required coordinates.

Example of plotting without data: Venn diagram

It is possible to use program **simplot** as a generalized diagram drawing program without any data points, as illustrated in the next figure.

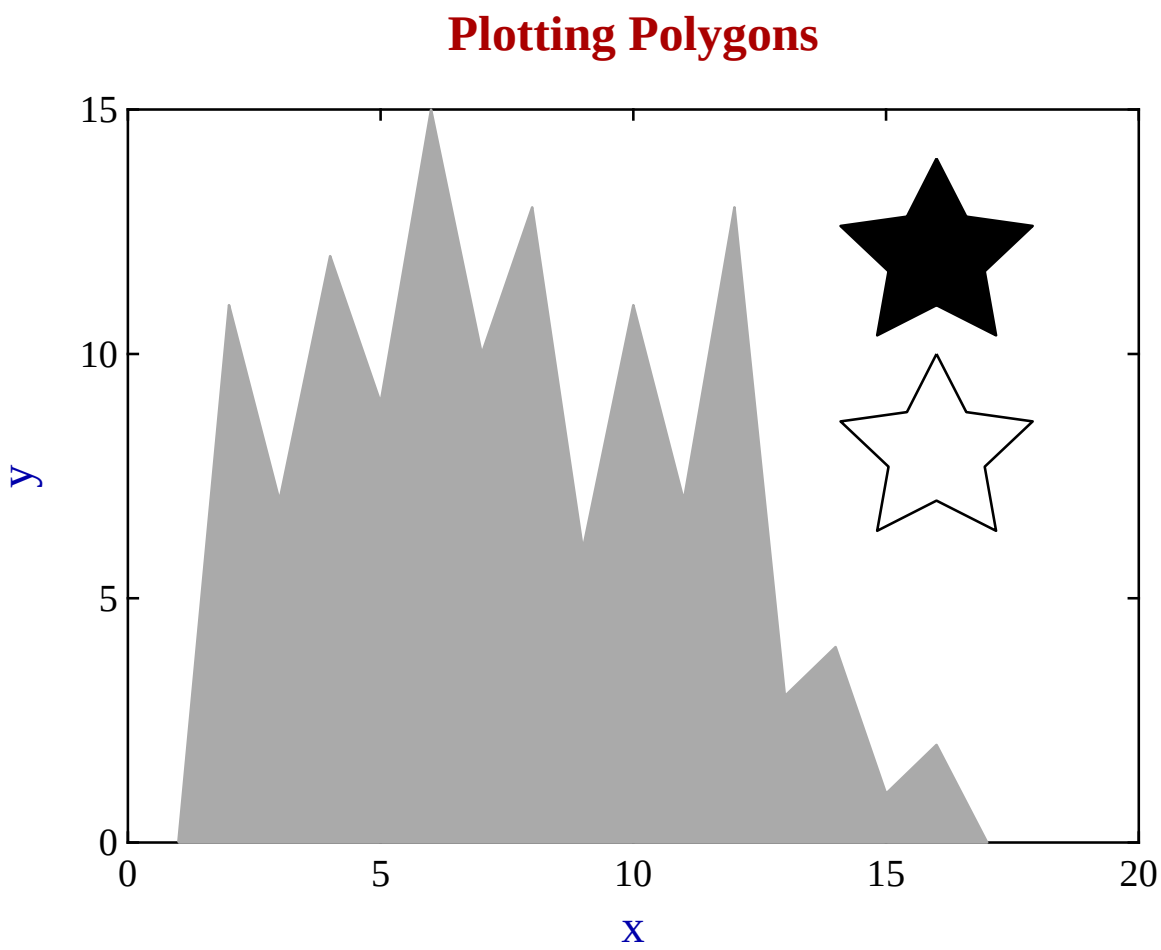


The procedure used to create such graphs using any of the SIMFIT graphical objects will be clear from the details now given for this particular Venn diagram.

- Program **simplot** was opened using an arbitrary dummy graphics coordinate file.
- The [Data] option was selected and it was made sure that the dummy data would not be plotted as lines or symbols, by suppressing the lines and symbols for this dummy data set.
- This transformed program **simplot** into an arbitrary diagram creation mode and the display became completely blank, with no title, legends, or axes.
- The circles were chosen (as objects) to be outline circle symbols, the box was selected (as an arrow-line-box) to be a horizontal transparent box, the text strings were composed (as text objects), and finally the arrow was chosen (as an arrow-line-box) to be a solid script arrow.
- The diagram was then completed by editing the text strings (in the expert mode) to introduce the mathematical symbols.

Polygons

Program **simplot** allows filled polygons as an optional linetype. So this means that any set of n coordinates (x_i, y_i) can be joined up sequentially to form a polygon, which can be empty if a normal line is selected, or filled with a chosen color if the filled polygon option is selected. If the last (x_n, y_n) coordinate pair is not the same as the first (x_1, y_1), the polygon will be closed automatically. This technique allows the creation of arbitrary plotting objects of any shape, as will be evident from the sawtooth plot and stars in the next figure.



The sawtooth graph above was generated from a set of (x, y) points in the usual way, by suppressing the plotting symbol but then requesting a filled polygon linetype, colored light gray. The open star was generated from coordinates that formed a closed set, but then suppressing the plotting symbol and requesting a normal, i.e. solid linetype. The filled star was created from a similar set, but selecting a filled polygon linetype, colored black.

If you create a set of ASCII text plotting coordinates files containing arbitrary polygons, such as logos or special plotting symbols, these can be added to any graph. However, since the files will simply be sets of coordinates, the position and aspect ratio of the resulting objects plotted on your graph will be determined by the ranges you have chosen for the x and y axes, and the aspect ratio chosen for the plot. Clearly, objects created in this way cannot be dragged and dropped or re-scaled interactively. The general rule is that the axes, title, plot legends, and displayed data exist in a space that is determined by the range of data selected for the coordinate axes. However, extra text, symbols, arrows, information panels, etc. occupy a fixed space that does not depend on the magnitude of data plotted. So, selecting an interactive data transformation will alter the position of data dependent structures, but will not move any extra text, lines, or symbols.

13.2.2 Basic plotting styles

SIMFIT offers a large choice of options to present data and results from model fitting and, to illustrate the basic plotting styles, an example from fitting three epidemic differential equations to data using program **deqsol** will be presented in the next collage.

Figure 1: Standard Axes

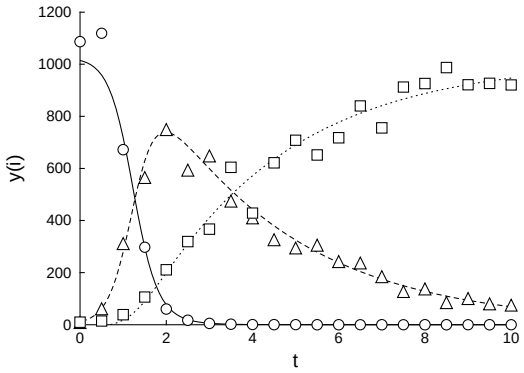


Figure 2: Offset Axes

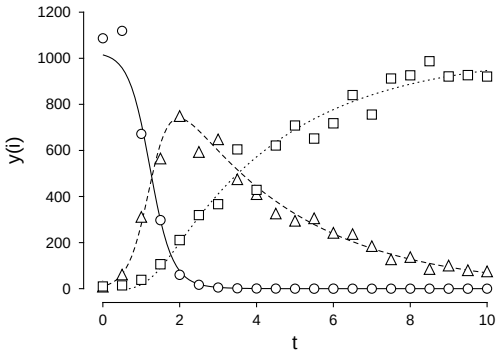


Figure 3: Offset and Box

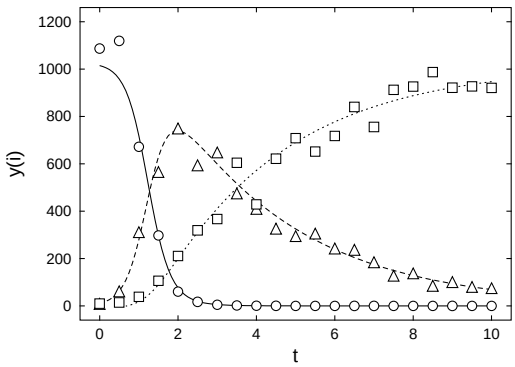


Figure 4: Grid Lines

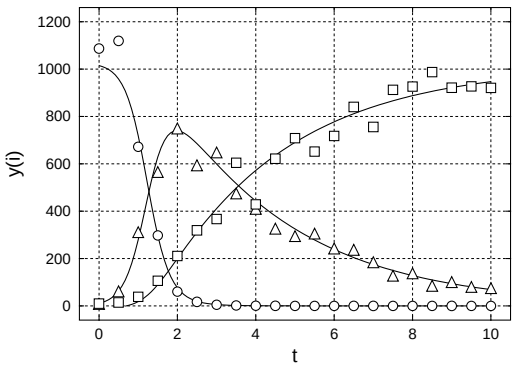


Figure 1 Perhaps the usual style for scientific graphics, i.e., to simply display the data and axes with tick marks pointing inwards and no additional features as in the Standard Axes plot in Figure 1 above.

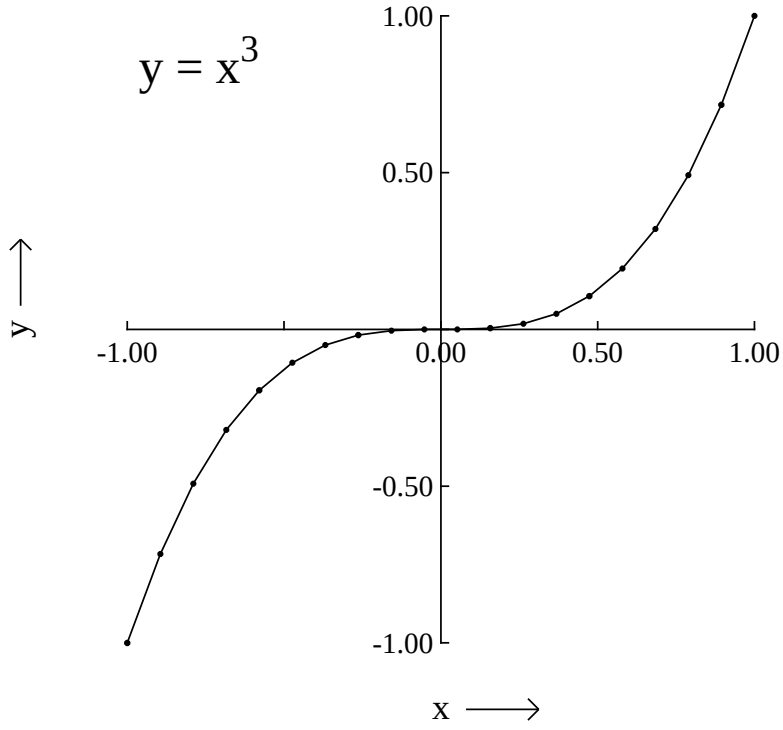
Figure 2 However, the tick marks pointing inwards coupled with overlaying the data and fitted curve on top of the *X* axis suggests that offset axes with tick marks pointing outwards, as in the Offset Axes plot of Figure 2 above, could be an improvement.

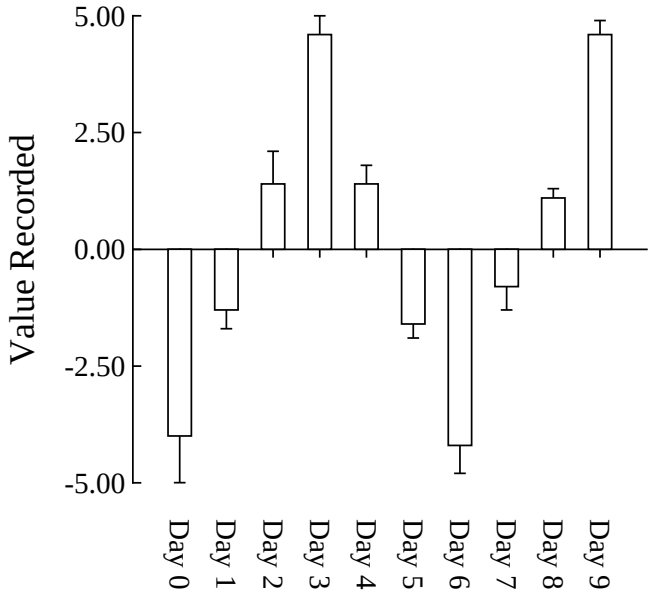
Figure 3 Again, some regard a box round the data plotted, as in the Offset and Box plot of Figure 3 above, to be visually pleasing.

Figure 4 Often grid lines, as in the Grid Lines example of Figure 4 above, help to establish coordinates, especially in calibration curves.

Alternative axes and labels

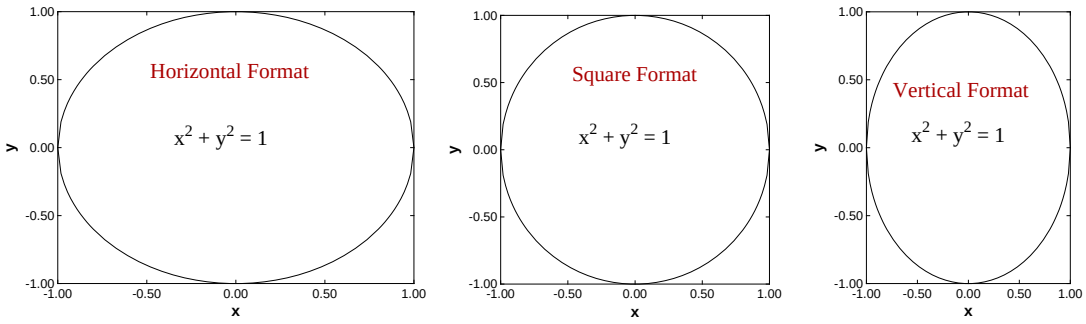
It is useful to move axes to make plots more meaningful, and it is sometimes necessary to hide labels, as with the plot of $y = x^3$ in the next figure, where the second x and third y label are suppressed. The figure also illustrates moving an axis in barcharts with bars above and below a baseline.





Alternative sizes, shapes, and clipping

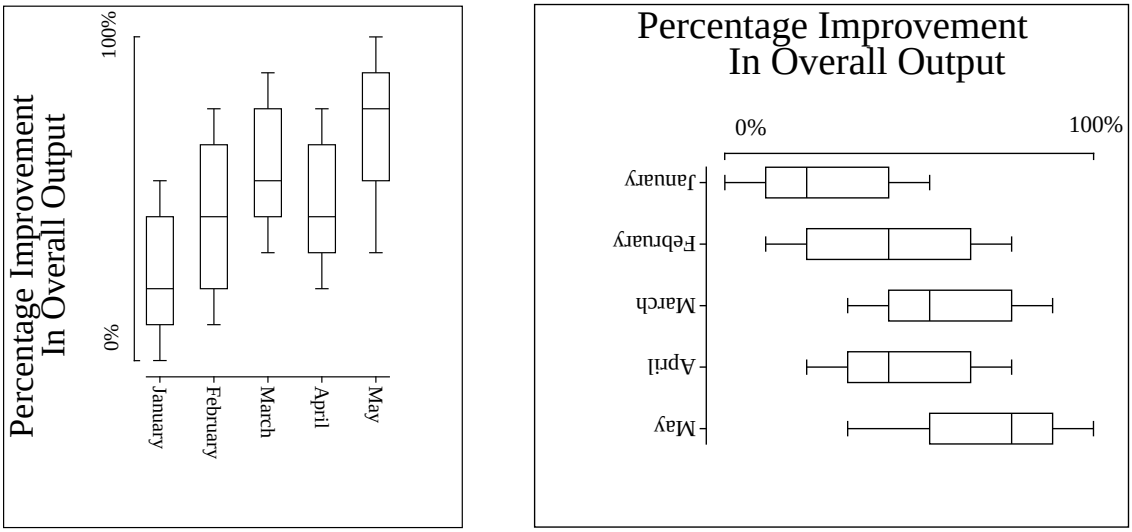
Plots can have horizontal, square or vertical format as in the next figure, and user-defined clipping schemes can be used. After clipping, SimFIT adds a standard BoundingBox so all plots with the same clipping scheme will have the same absolute size but, when GSview/Ghostscript transforms ps into eps, it clips individual files to the boundary of white space and the desirable property of equal dimensions will be lost.



There is also a stretched format for long horizontal ribbon type graphs and a PostScript option to stretch white space without altering symbols and fonts, which is very useful with dense data sets such as dendrograms.

Rotated and re-scaled graphs

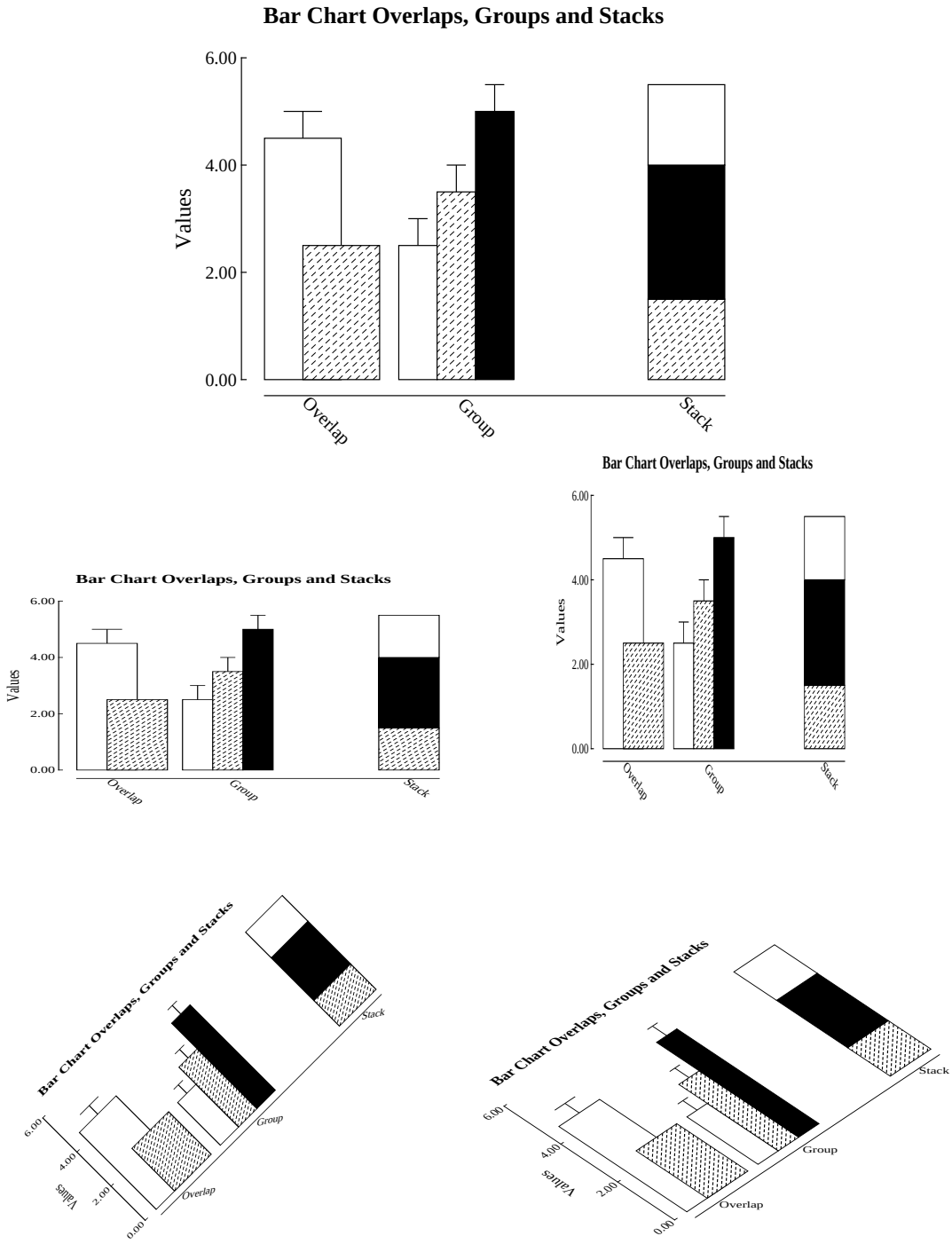
PostScript files can be read into **editps** which has options for re-sizing, re-scaling, editing, rotating, making collages, etc. In the next figure the box and whisker plot was turned on its side to generate a side-on barchart. To do this sort of thing you should learn how to browse a SimFIT PostScript file in the SimFIT viewer to read BoundingBox coordinates, in PostScript units of 72 to one inch, and calculate how much to translate, scale, rotate, etc.



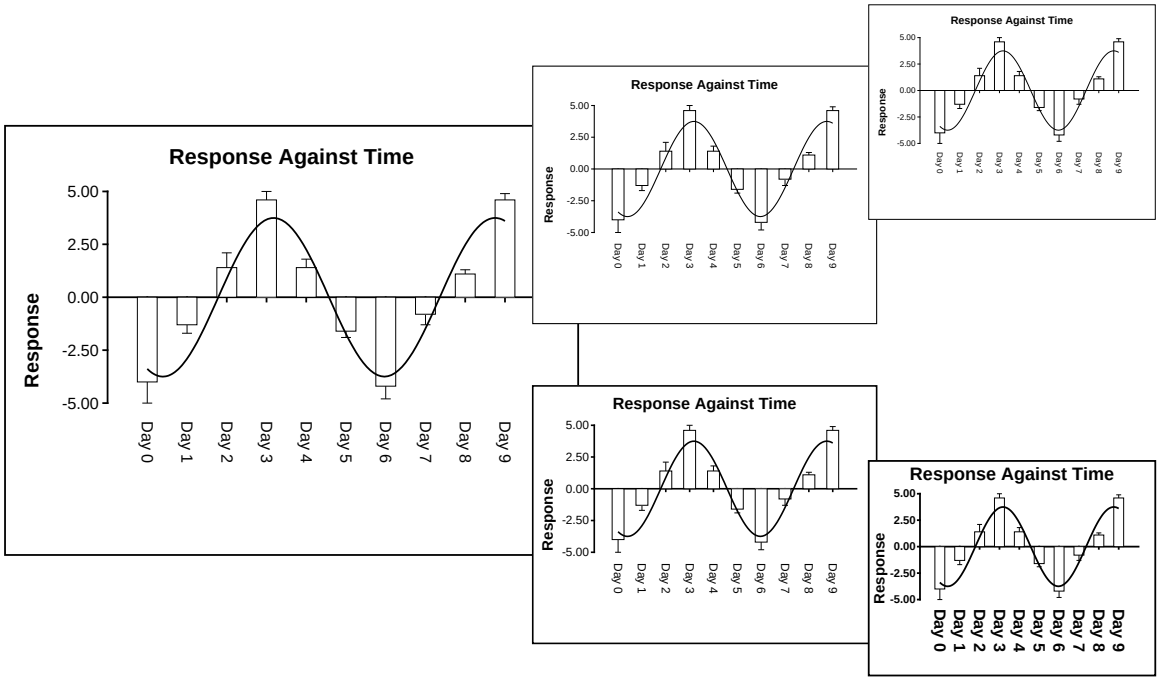
PostScript users should be warned that the special structure of SimFIT PostScript files that allows extensive retrospective editing using **editps**, or more easily if you know how using a simple text editor like **notepad**, is lost if you read such graphs into a graphics editor program like Adobe Illustrator. Such programs start off by redrawing vector graphics files into their own conventions which are only machine readable.

Changed aspect ratios and shear transformations

The barchart in the figure below was scaled to make the X-axis longer than the Y-axis and vice-versa, but note how this type of differential scaling changes the aspect ratio as illustrated. Since rotation and scaling do not commute, the effect created depends on the order of concatenation of the transformation matrices. For instance, scaling then rotation cause shearing which can be used to generate 3-dimensional perspective effects as in the last sub-figure.



Reduced or enlarged graphs



It is always valuable to be able to edit a graph retrospectively, to change line or symbol types, eliminate unwanted data, suppress error bars, change the title, and so on. SIMFIT PostScript files are designed for just this sort of thing, and a typical example would be altering line widths and font sizes as a figure is re-sized.

In the figure above the upper sub-figures are derived from the large figure by reduction, so the text becomes progressively more difficult to read as the figures scale down.

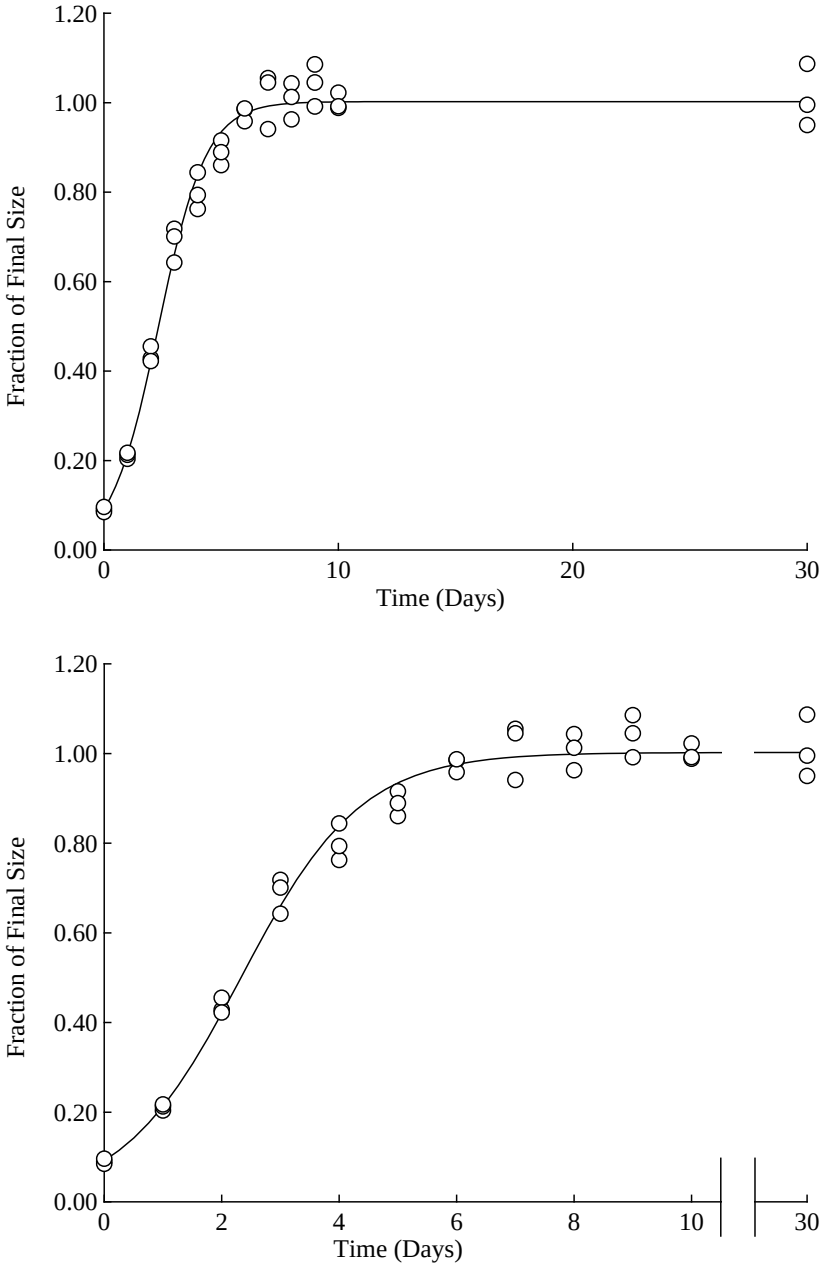
In the lower sub-figures, however, line thicknesses and font sizes have been increased as the figure is reduced, maintaining legibility. Such editing can be done interactively, but SIMFIT PostScript files are designed to make such retrospective editing easy as described in the `w_readme.*` files and now summarized.

- Line thickness: Changing 11.00 setlinewidth to 22 setlinewidth doubles, while, e.g. 5.5 setlinewidth halves all line thicknesses, etc. Relative thicknesses are set by **simplot**.
- Fonts: Times-Roman, Times-Bold, Helvetica, Helvetica-Bold (set by **simplot**), or, in fact, any of the fonts installed on your printer.
- Texts: ti(title), xl(*x* legend), yl(*y* legend), tc(centered for *x* axis numbers), tl(left to right), tr(right to left), td(rotated down), ty(centered for *y* axis numbers).
- Lines: pl(polyline), li(line), da(dashed line), do(dotted line), dd(dashed dotted).
- Symbols: ce(i.e. circle-empty), ch(circle-half- filled), cf(circle-filled), and similarly for triangles(te, th, tf), squares(se, sh, sf) and diamonds(de, dh, df). Coordinates and sizes are next to the abbreviations to move, enlarge, etc.

If files do not print after editing you have probably added text to a string without padding out the key. Find the fault using the **GSview/Ghostscript** package then try again.

Split axes

Sometimes split axes can show data in a more illuminating manner as in the figures below. The options are to delete the zero time point and use a log scale to compress the sparse asymptotic section, or to cut out the uninformative part of the best fit curve between 10 and 30 days.

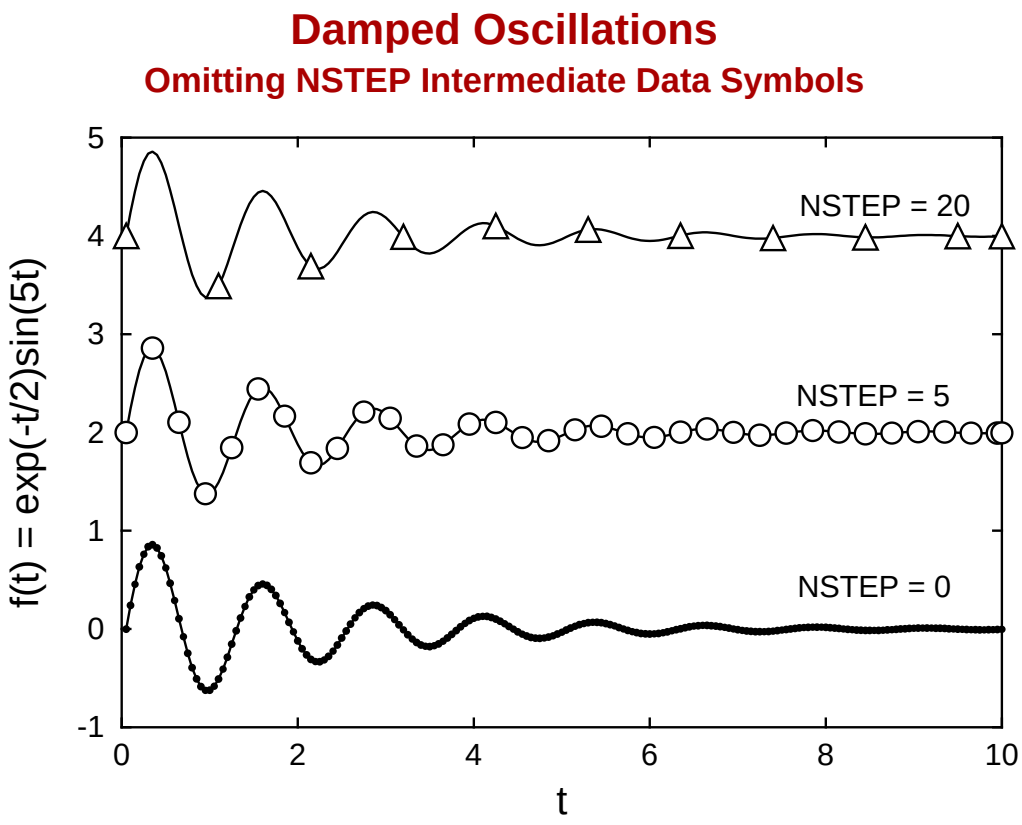


Windows users can do such things with enhanced metafiles (`*.emf`), but there is a particularly powerful way for PostScript users to split SimFIT graphs in this way. When the SimFIT PostScript file is being created there is a menu selectable shape option that allows users to chop out, re-scale, and clip arbitrary pieces of graphs, but in such a way that the absolute position, aspect ratio, and size of text strings does not change. In this way a master graph can be decomposed into any number of appropriately scaled slave sub-graphs. Then **editps** can be used to compose a graph consisting of the sub-graphs rearranged, repositioned, and resized in any configuration. The lower figure was created in this way after first adding the extra lines shown at the splitting point.

Stepping over intermediate data points

Sometimes it is advantageous to step over intermediate data points and, for clarity, only plot points at intervals through the data set. For instance, there may be a large number of machine generated data points, far more than is required to define a curve by the usual technique of joining joining up successive points by straight lines. Plotting all the points in such cases would slow down the graphics display and, more importantly, could lead to very large PostScript output files. Again, there would be times when a continuous curve is required to illustrate a function defined by a reasonable number of closely spaced data points, but symbols at all points would obscure the curve, or might only be required for labeling purposes.

The next figure illustrates a case in point.



Here even small symbols, like dots in the bottom figure where no points are omitted, would obscure the plot and the middle plot with intermediate groups of five suppressed, could also be regarded as too crowded, while the upper plot has sufficient symbols even with twenty points stepped over to identify the plot, say in an information panel.

Of course such graphs can easily be created by pre-processing the data files before submitting for plotting. However, SimFIT has a special facility to do this interactively. The technique required to create plots like the above figures is to submit two files with identical data, one to be plotted as a line joining up data to create the impression of a continuous curve, the other to be plotted as symbols. Then, from the [Data] menu when the plot is displayed, a parameter $NSTEP$ can be defined, where $NSTEP$ is the number of intermediate points in each group to be stepped over.

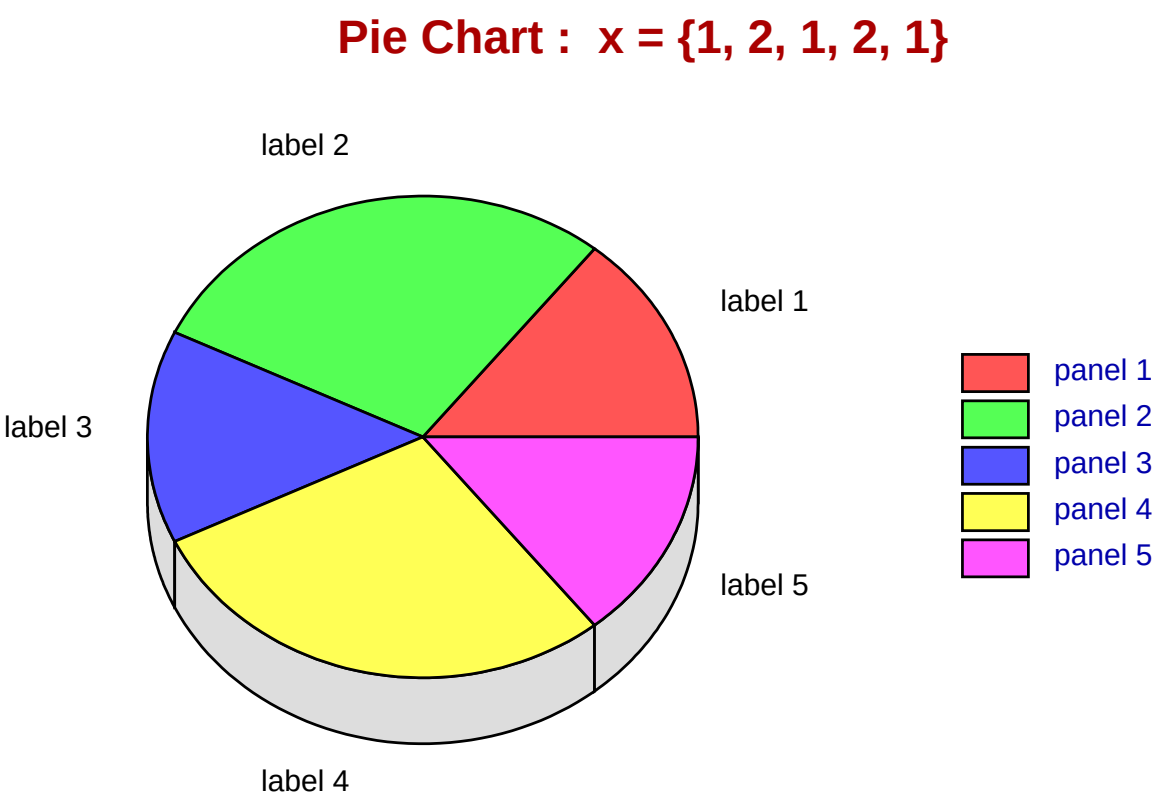
Observe that, when using this technique, the first and last data points are always plotted to avoid misunderstanding.

13.2.3 Pie charts

The simplest way to create a pie chart is to input a vector of positive numbers into SIMFIT program **simplot**. For instance the vector

1
2
1
2
1

generates the following pie chart with default fill styles, colours, labels, and panel labels, and where the volume of segments is proportional to the number in the vector.

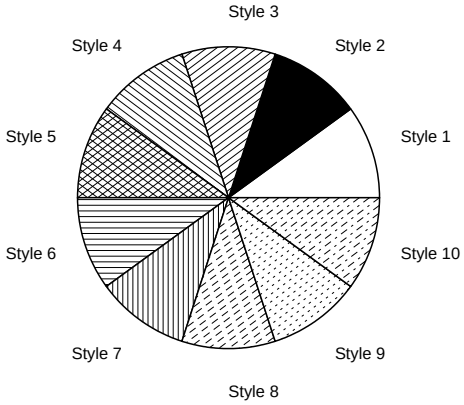


As it can be tedious to input a vector and then have to edit the title, segment details, panel labels, etc. There are two ways to simplify this process.

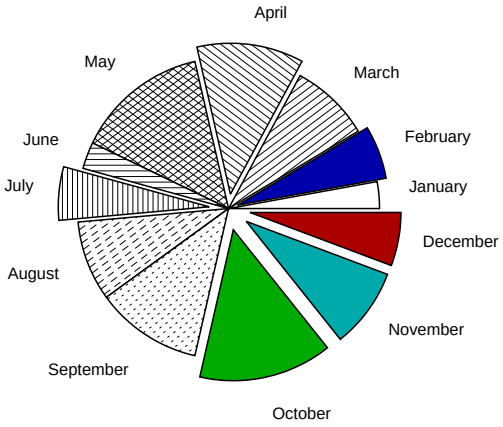
1. Save a configuration file from program **simplot** and read it in after supplying the numeric vector to use special defaults.
2. Prepare a special file like `piechart.tf1` containing 4 columns to input the data along with all the details for colors, segment displacements, and labels. The format can be appreciated by examining this file in a text editor.

Then next three plots illustrate piecharts created using `piechart.tf1`, `piechart.tf2`, and `piechart.tf31`, followed by two further examples illustrating special features.

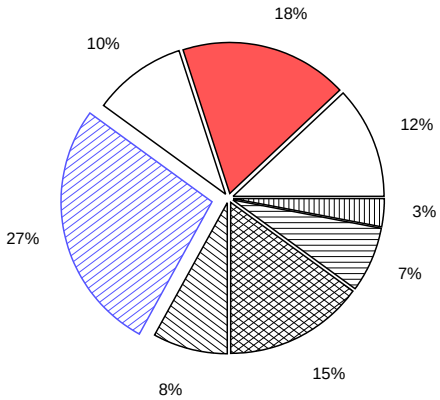
File piechart.tf1: fill styles

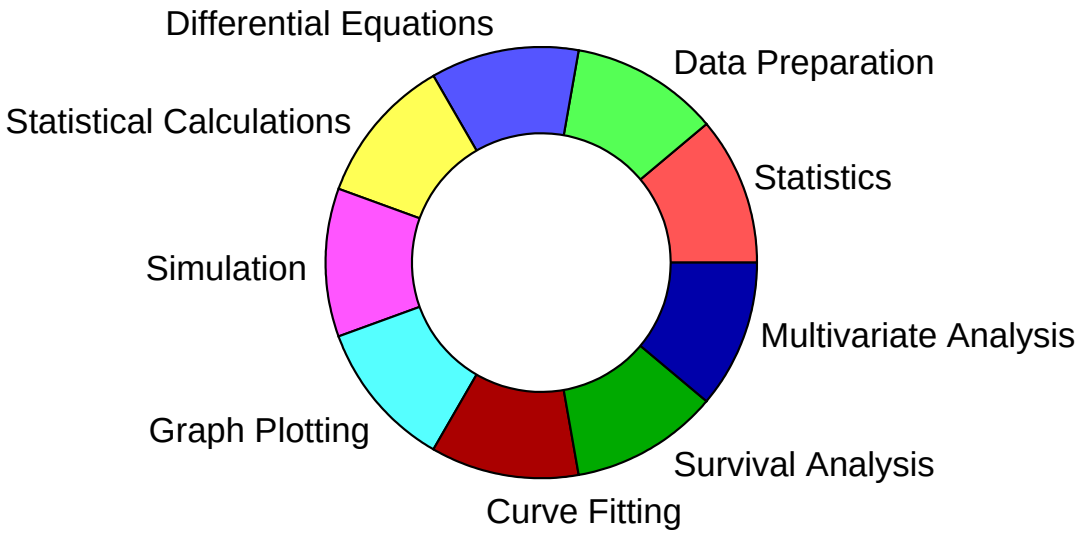


File piechart.tf2: displacements



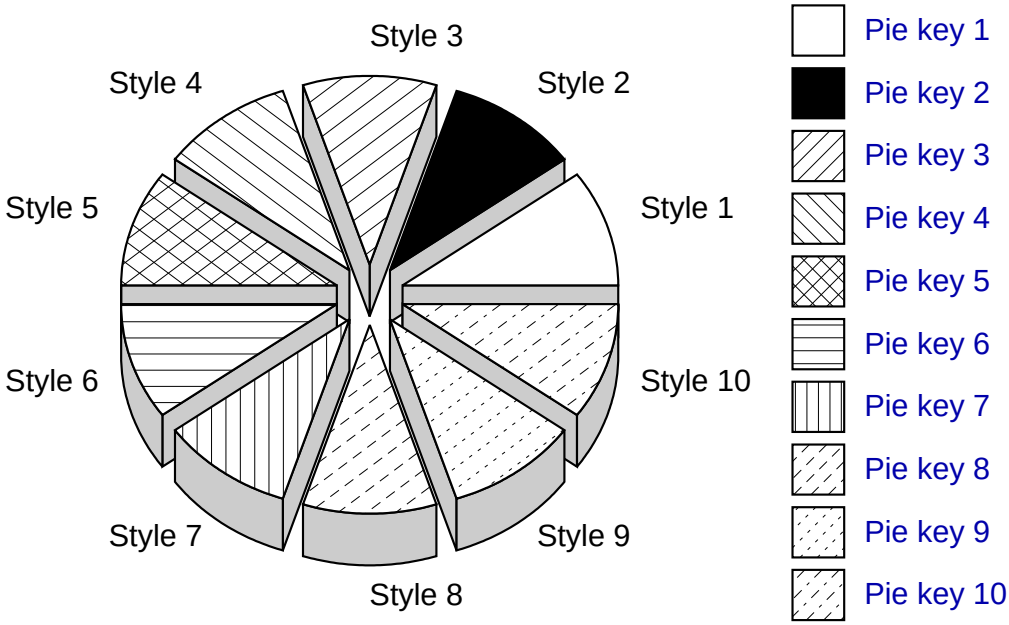
File piechart.tf3: features





SIMFIT

Pie Chart Fill Styles

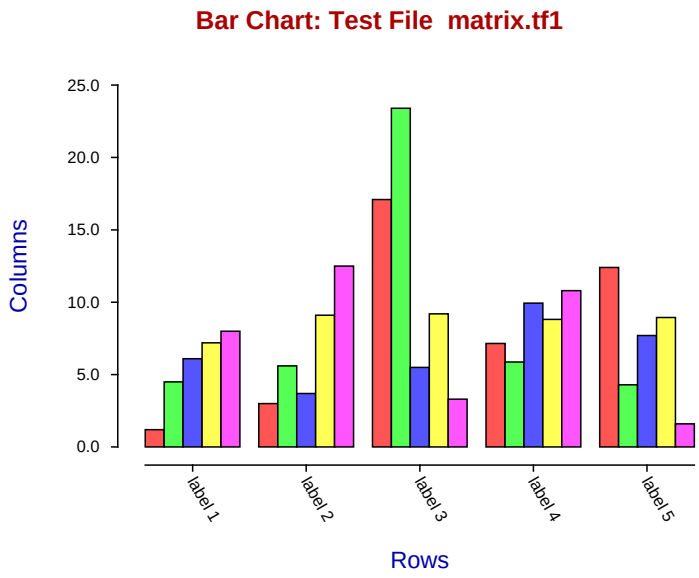


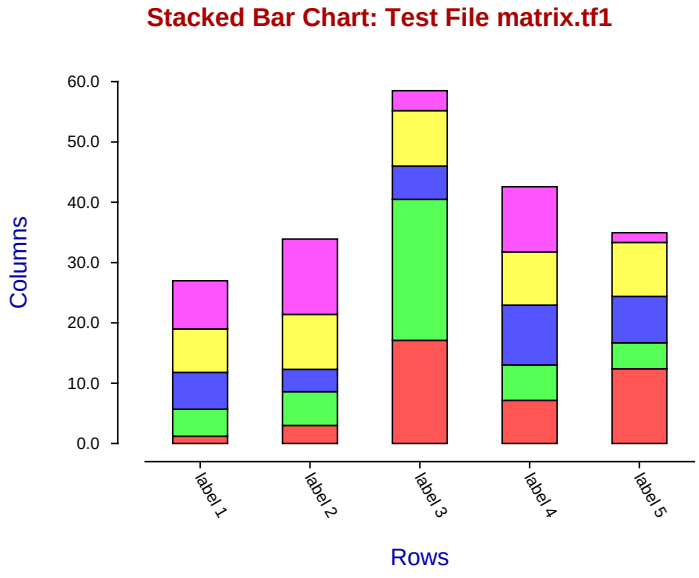
13.2.4 Bar charts

The simplest way to plot a bar chart is to prepare a data table with rows representing groups and columns for bars within groups. For example the test file `matrix.tf1` has the following values

$$A = \begin{pmatrix} 1.20 & 4.50 & 6.10 & 7.20 & 8.00 \\ 3.00 & 5.60 & 3.70 & 9.10 & 12.5 \\ 17.1 & 23.4 & 5.50 & 9.20 & 3.30 \\ 7.15 & 5.87 & 9.94 & 8.82 & 10.8 \\ 12.4 & 4.30 & 7.70 & 8.95 & 1.60 \end{pmatrix}$$

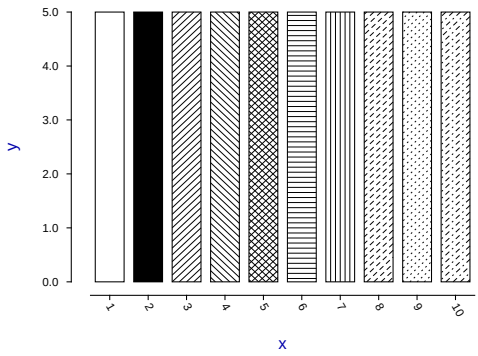
and from the exhaustive analysis procedure in program **simstat** these plots can be displayed.



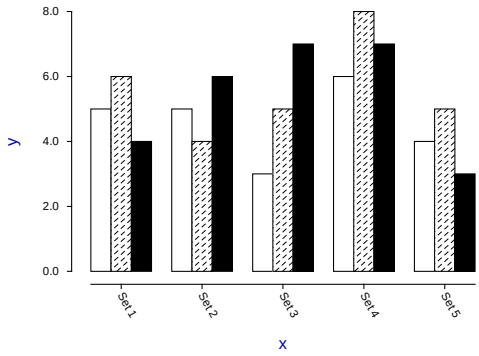


The next plots illustrate the effects created by several test files using program **simplot**.

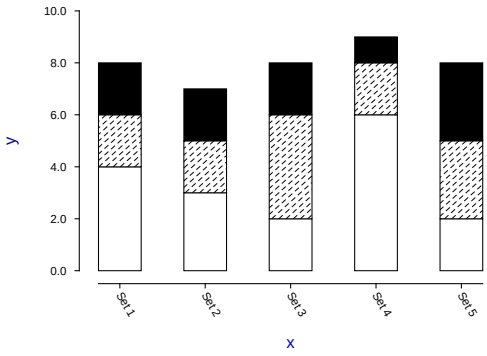
Test File barchart.tf2: 10 Fill Styles



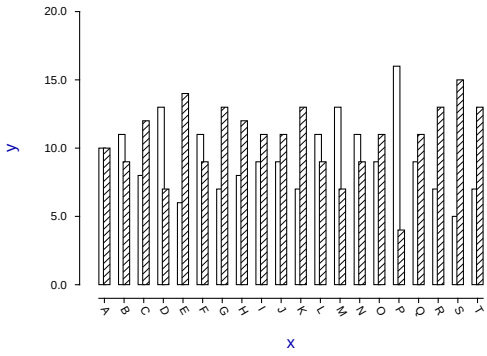
Test File barchart.tf3: Adjacent Groups



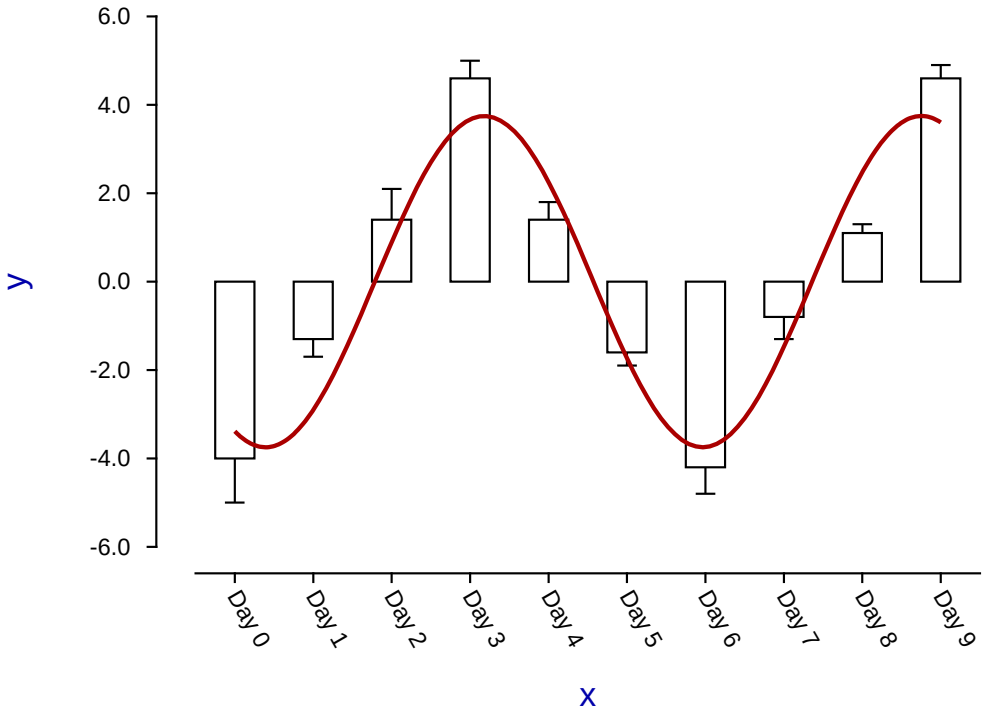
Test File barchart.tf4: Stacked Bars



Test File barchart.tf5: Overlayed Bars



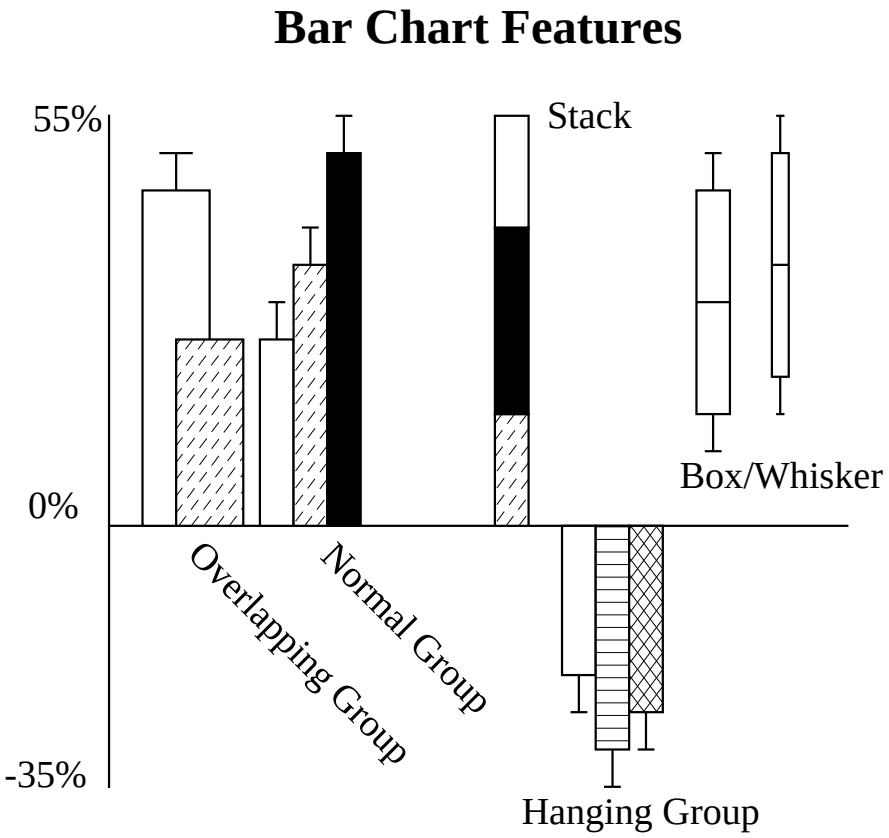
Test Files barchart.tf6 and barchart.tf7



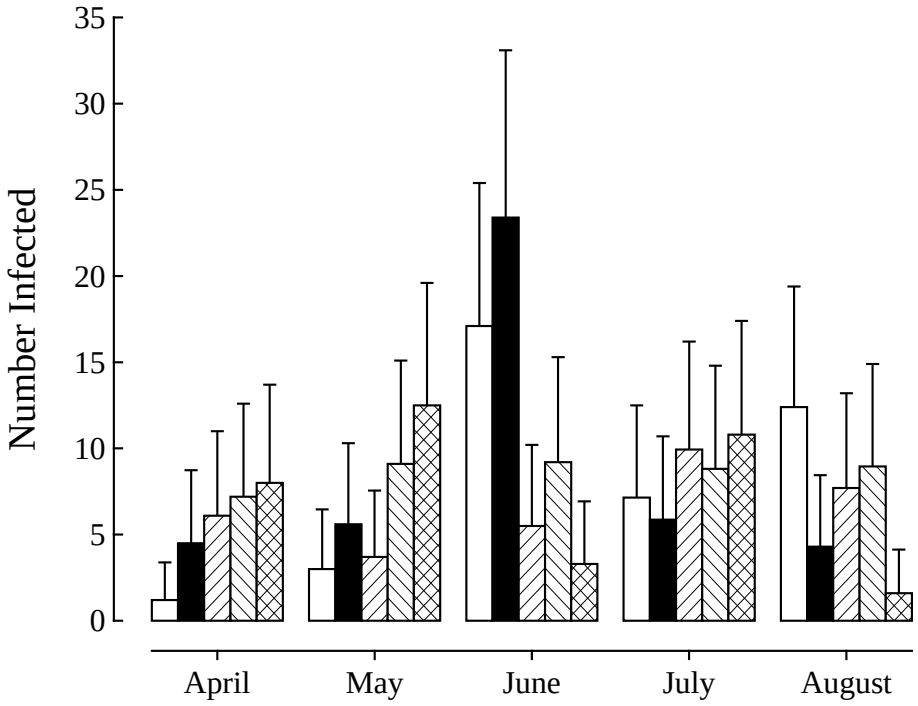
The way that SIMFIT creates bar charts is as follows.

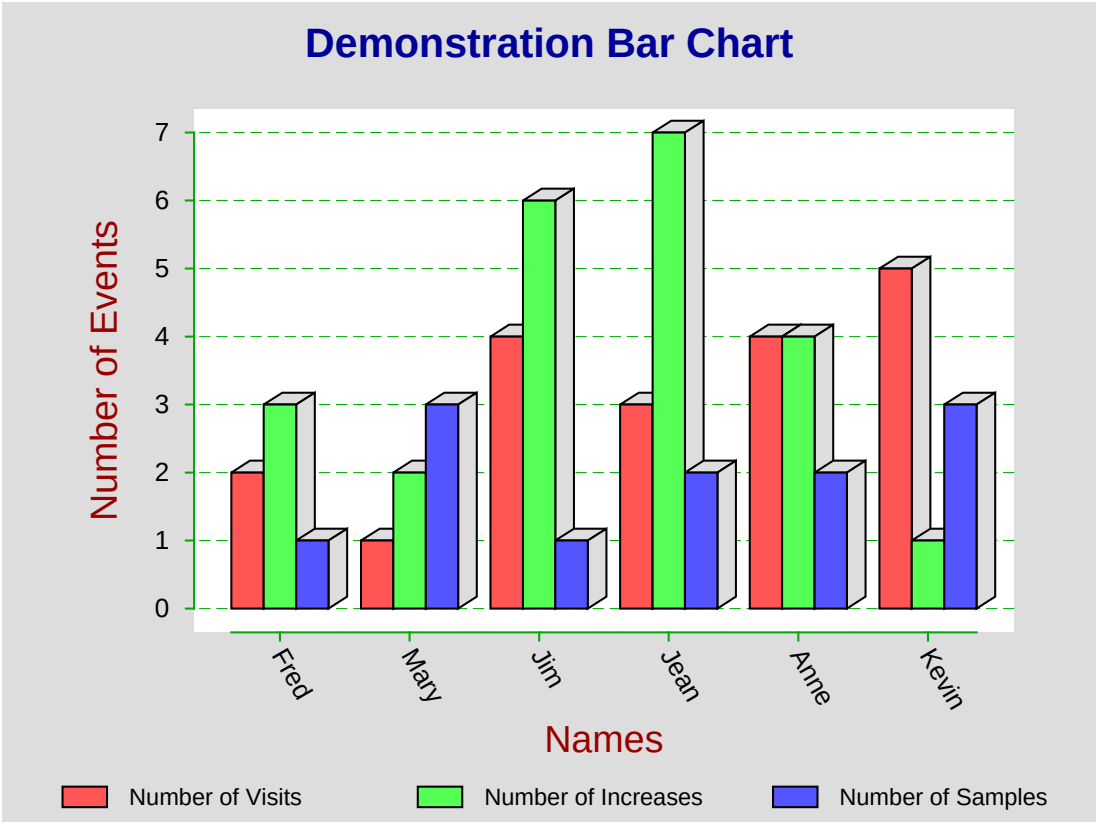
1. If the data matrix input does not have precisely 9 columns then a special advanced barchart file is created with exactly 9 columns, where the value in each column represents a graphical feature.
2. In case you want to plot a data matrix with 9 columns, simply add an extra column 10 containing all values equal to 0 and **simplot** will display the barchart correctly.
3. The format for such files can be seen from examining the barchart test files then checking the features as displayed previously.
4. It will be appreciated from the explanations added after the data sections in the test files that this method allows tremendous possibilities to sculpture barcharts with almost every possible variation concerning upper and lower error bars, fill styles, colors, labels, sizes and positions.
5. Once a matrix or advanced barchart file has been input, program **simplot** allows other data files to be input, for instance to represent error bars, or another curve to be overlayed as in the previous plot where the curve from `barchart.tf7` is overlayed on the barchart from `barchart.tf6`.

Some features that are possible with advanced barcharts are illustrated by the following figure.



Adding error bars to the barchart from `matrix.tf1`, and further editing that is possible after program **simplot** has displayed the default barchart are demonstrated in the next plots.


Number Infected
35
30
25
20
15
10
5
0
April
May
June
July
August


Demonstration Bar Chart
Number of Events
7
6
5
4
3
2
1
0
Fred
Mary
Jim
Jean
Anne
Kevin
Names
Number of Visits
Number of Increases
Number of Samples

13.2.5 Box and whisker plots

For a box and whisker plot you can use the exhaustive analysis of a matrix or 1-way ANOVA options from program **simstat**, or the simple and statistical plotting using program **simplot**. The case to be considered is where there are 5 groups of sizes 5, 8, 6, 8, and 8 using the library file `anova1.tfl`.

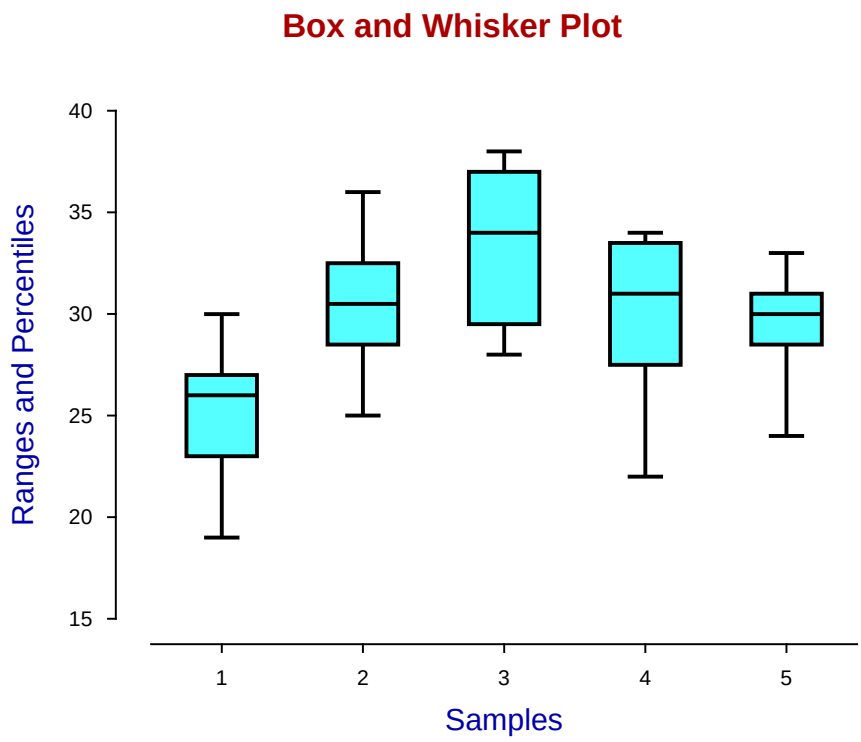
23	29	38	30	31
27	25	31	27	33
26	33	28	28	31
19	36	35	22	28
30	32	33	33	30
	28	36	34	24
	30		34	29
	31		32	30

If sample sizes differ, data can be entered as an incomplete matrix with missing values, e.g., `incomplete.tf1`, individual column vectors, from a project archive, or as a library file referencing data files for each of the columns. Results from 1-way ANOVA are as follows.

1-Way Analysis of Variance: Grand Mean 29.89
Transformation: x (untransformed data)

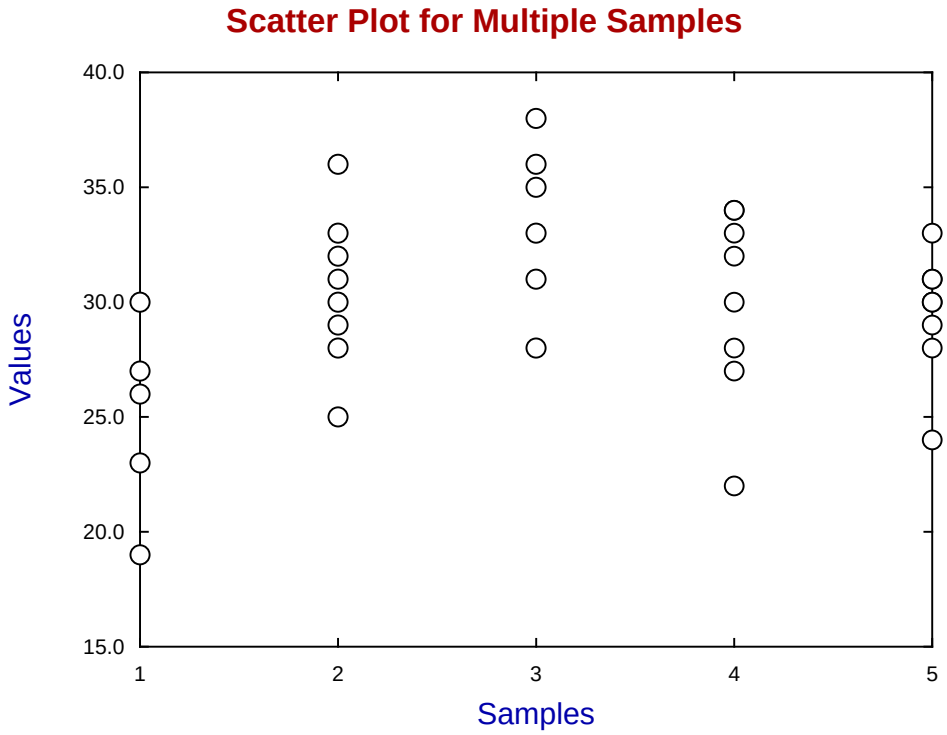
Source	SSQ	NDOF	MSQ	F	p
Between Groups	202.0	4	50.51	3.931	0.0111
Residual	385.5	30	12.85		
Total	587.5	34			

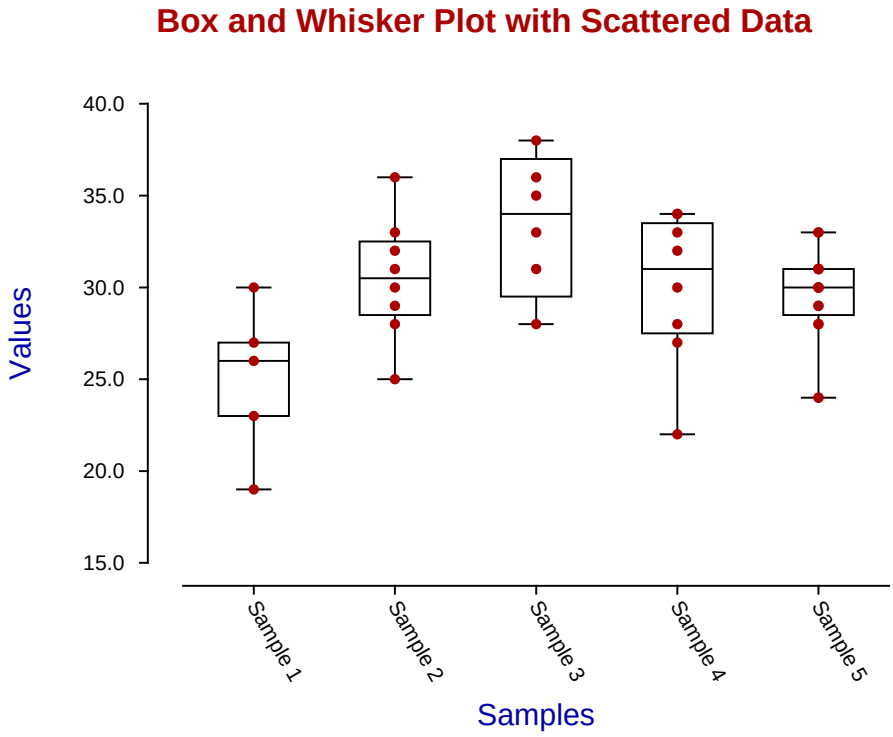
The the next plot shows sample ranges with medians and quartiles as a box and whisker plot.



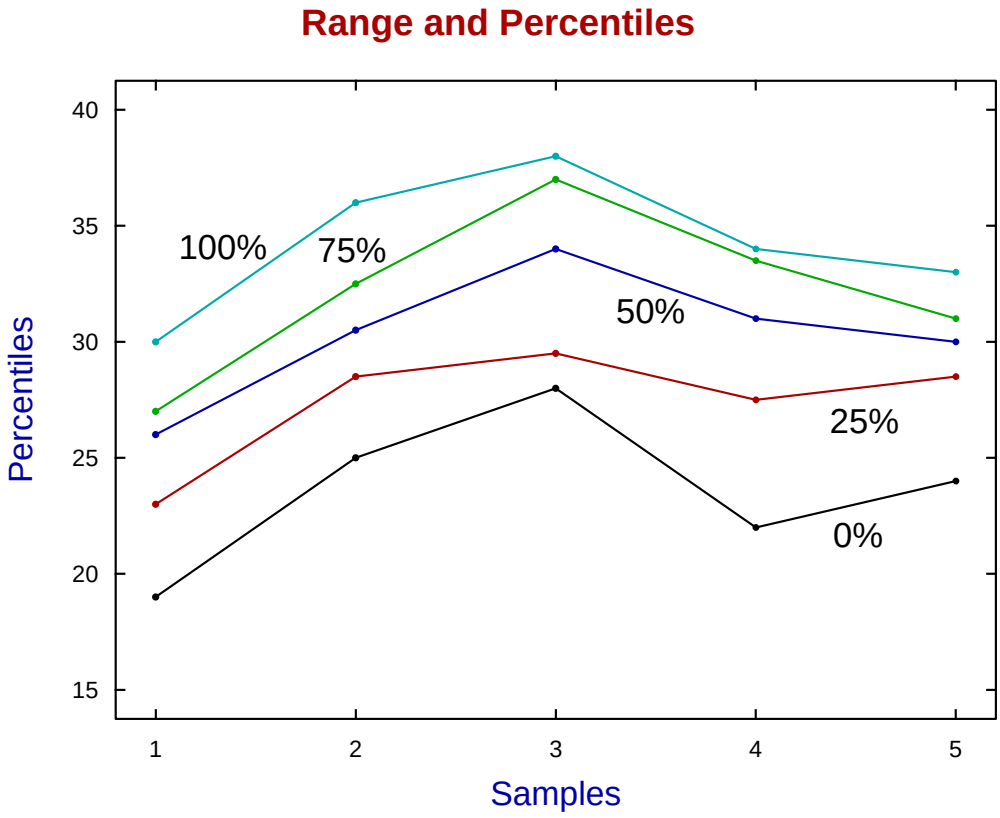
The results suggest rejecting the null hypothesis of equal means at the 5% significance level as $p < 0.05$, but not the 1% significance level as $p > 0.01$.

The data can also be shown with error bars or as scattered points or even points scattered on the box and whisker plot as shown below.





Another way to display the data is a range and percentiles plot as illustrated next. Here the lowest line segments join the lowest sample value for the corresponding groups, the upper line segments join the largest sample values, while between them the line segments join the points corresponding to the 25%, 50%, and 75% levels.



Actually box and whisker plots can be created using advanced bar chart files like `barchart.tf1` which has the following format, derived from the data in `anova.tfl`.

```
Box and whisker plot
5    9
1    -2    -1    0    2    3    1    1    15
3    -1     0    2    4    5    1    1    15
5     1     2    3    5    6    1    1    15
7     0     1    2    4    5    1    1    15
9     1     3    5    6    7    1    1    15
7
begin{labels}
January
February
March
April
May
end{labels}
```

Note that, by constructing such advanced bar chart files, it is possible to create a great number of specialized plots, for instance: plots with overlapping bars, bars with error bars, mixed bars and box and whiskers, hanging bars, etc.

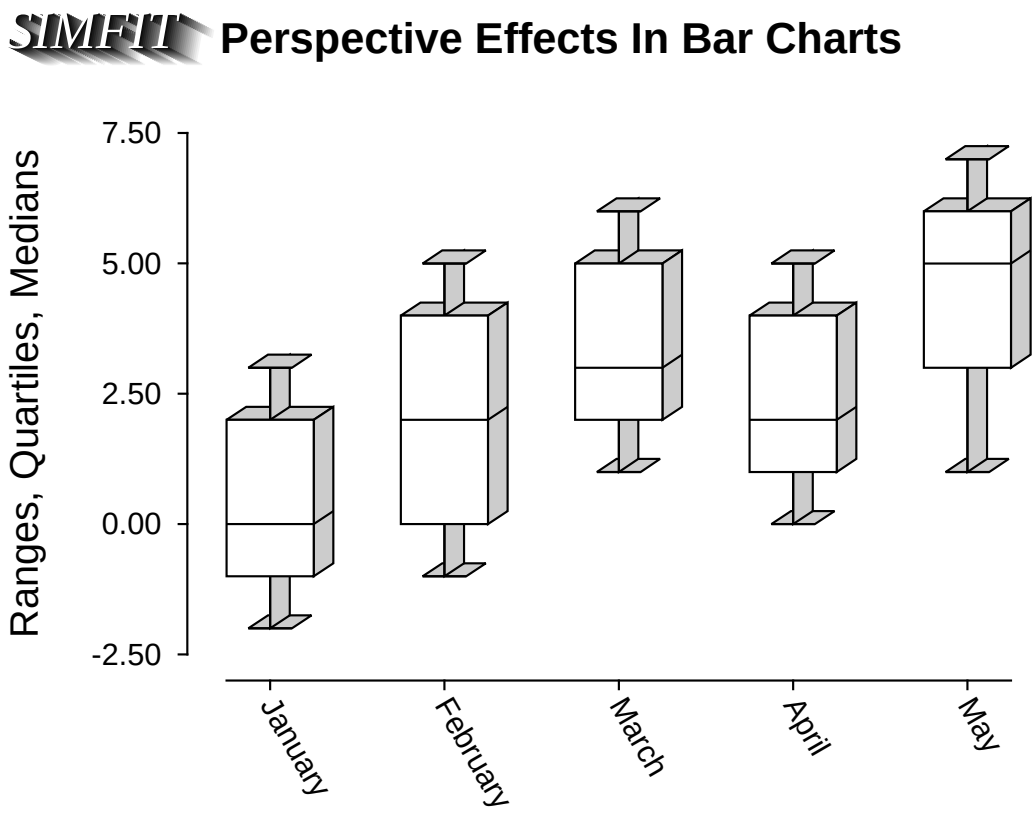
An explanation of the above three sections of data values follows.

1. Header: line 1: title, line 2: m = number of rows, n = number of columns
2. Data: line 3 onwards. Each line of data looks like this:-
   $x, y1, y2, y3, y4, y5, f, w, c$, and has the following interpretation:

- x = x coordinate for the bar (x in nondecreasing order)
- $y1$ = y for bottom of range (i.e. bottom error bar)
- $y2$ = y for lower quartile (i.e. bottom of box)
- $y3$ = y for median of data (i.e. divider for box)
- $y4$ = y for upper quartile (i.e. top of box)
- $y5$ = y for top of range (i.e. top error bar)
- f = fill-style (between 0 and 10)
- w = width (between 0 and 1)
- c = colour (between 0 and 71)

3. Labels: line $m + 3$: number of important trailing text lines, the first $m + 2$ of these being
   - a) first of all begin{labels} to indicate the start of the labels,
   - b) then the consecutive labels for data rows 1 to m,
   - c) then finally end{labels} to indicate the end of the labels.

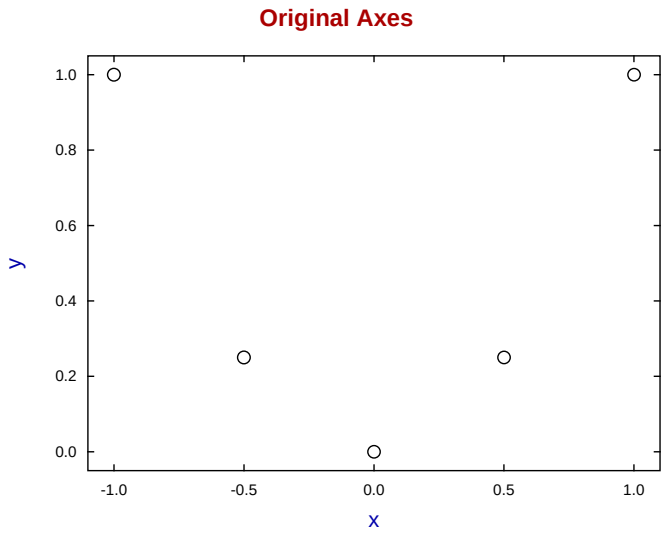
For example, reading `barchart.tf1` into program **simplot** followed by minor editing then installing a PostScript special created the next plot.



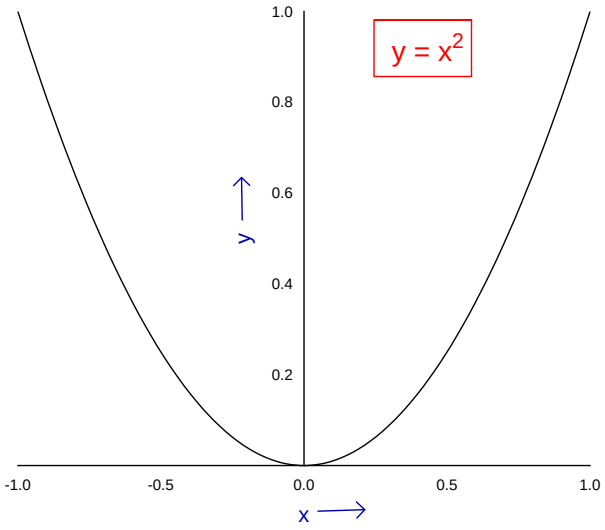
13.2.6 Standard plots

The easiest way to plot a simple graph is to open a text editor such as **notepad**, type in a two column data table with x in column 1 and y in column 2, copy to the clipboard, then input into program **simplot**. For instance, to plot the parabola $y = x^2$, you could type in this table to create the following default graph.

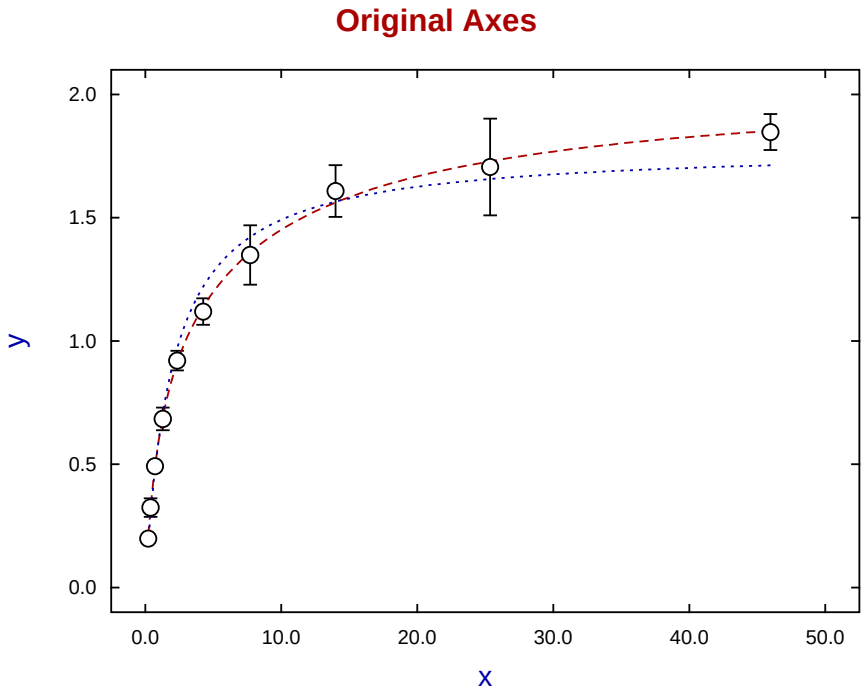
-1	1
-0.5	0.25
0	0
0.5	0.25
1	1



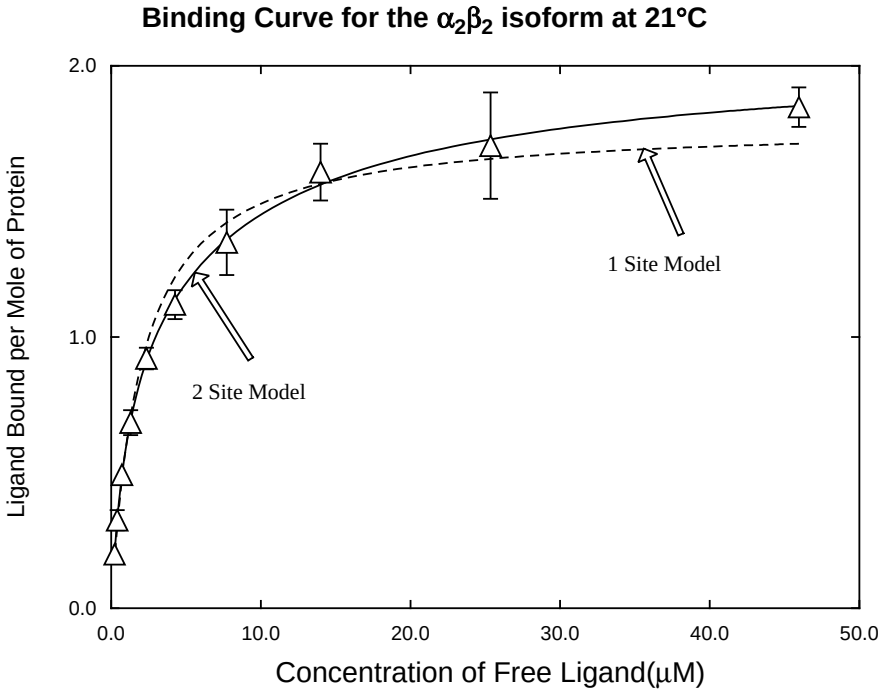
However, the following with 100 data points is probably more like what you had in mind.



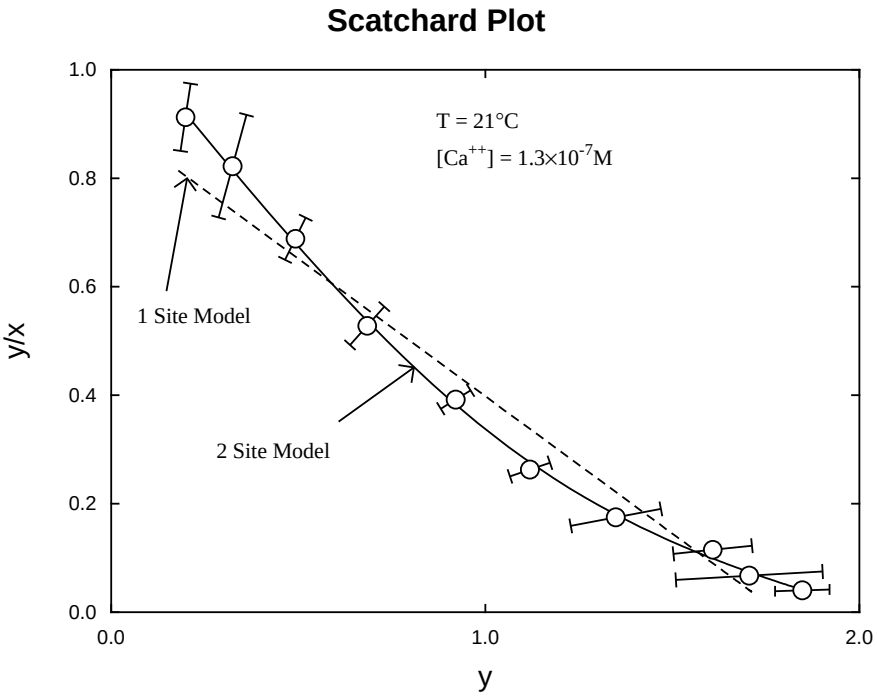
Of course you can copy and paste larger data sets from a spreadsheet program, but the superior way is to create SimFIT data files. These can be input in collections using library files, as in the next example created by reading the library test file `simfig1.tfl` into program **simplot** leading to this default graph.



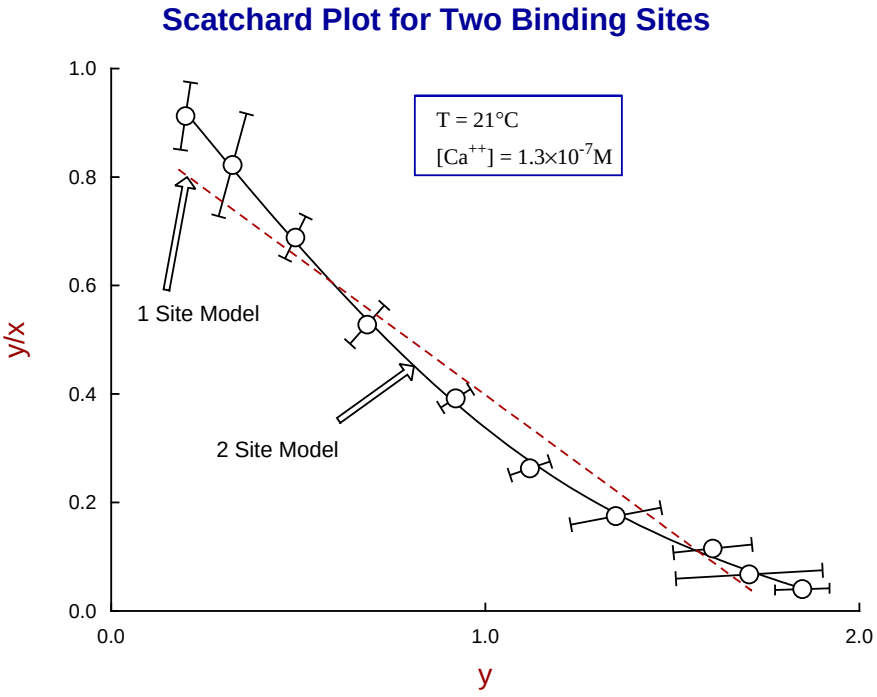
Since this was previously edited and the settings saved as the configuration file `w_simfig1.cfg`, this can now be input resulting in the next figure.



Alternatively the configuration file `w_simfig2.cfg` creates this figure.



This is simply the original data plotted in new coordinates and with additional features. Editing can then resume with this re-configured data in the same way as after reading in a SIMFIT metafile, as demonstrated in the next figure,



Advice

1. **Plotting a model equation**
   This is best achieved using program **makdat** with a model from the SIMFIT library or a user-supplied model file, as the range and number of points can be chosen interactively.
2. **Simple Graphics**
   During data analysis SIMFIT often displays a standard graph in simple graphics mode which only offers limited editing options and hardcopy possibilities. If there is reason to dedicate time and effort to make a superior graphics presentation, there is a [Advanced] option which allows users to to transfer to advanced graphics, or to add coordinate files to your project archive for retrospective use.
3. **Advanced graphics**
   From this interface extensive editing can be carried out before saving as a graphics file.
4. **Plotting multiple data sets**
   Groups of files for plotting are best collected into library files using coordinate files saved into your graphics project archive, as then they are certain to have the correct format.
5. **Saving a SIMFIT metafile**
   Once a graph has been edited you can store a configuration file, which is a template containing the editing details that can be input retrospectively to over-write the standard default parameters. Alternatively the editing details and the data can be saved to a SIMFIT metafile. Such metafiles can be input into program **simplot** to resume editing.
6. **Saving Windows quality hardcopy**
   The recommended order of preference is as follows.

   *.svg >> *.png > *.jpg > *.emf > *.tif >> *.bmp

   Use *.svg vector files for internet graphics and *.png for including into documents. Enhanced metafiles (*.emf) can also be included in documents, but care must be taken not to change the aspect ratio.
7. **Saving PostScript quality hardcopy**
   By far the superior way is to archive SIMFIT encapsulated PostScript files *.eps, because they are very compact, the resolution is device independent, they can be transformed into *.pdf and other graphics files, they can be edited in a text editor such as **notepad**, and there are many additional graphics procedures available. However, to get the most out of SIMFIT *.eps files you should have GhostScript installed and possibly a PostScript display program like Gsview.
8. **Using GhostScript**
   This wonderful free package can be used to transform SIMFIT *.eps files into *.pdf, *.png, and *.jpg that are much higher quality than anything you can get from Windows hardcopy. However *.svg files saved directly using the Windows hardcopy option are true vector graphics created by SIMFIT and are actually superior to the *.svg files created from *.eps by Ghostscript.

13.2.7 Double plots

Frequently different scales are required for the x or y axes and it is convenient to make a plot with, say two y axes. For instance, in column chromatography, with absorbance at 280nm representing protein concentration, at the same time as enzyme activity eluted, and the pH gradient. The following table is typical where absorbance could require a scale of zero to unity, while enzyme activity uses a scale of zero to eight, and pH could be on a scale of six to eight.

Fraction Number	*Absorbance*	*Enzyme Activity*	*Buffer pH*
1	0.0	0.1	6.0
2	0.1	0.3	6.0
3	1.0	0.2	6.0
4	0.9	0.6	6.0
5	0.5	0.1	6.2
6	0.3	0.8	6.7
7	0.1	1.5	7.0
8	0.3	6.3	7.0
9	0.4	8.0	7.0
10	0.2	5.5	7.0
11	0.1	2.0	7.2
12	0.1	1.5	7.5
13	0.3	0.5	7.5
14	0.6	1.0	7.5
15	0.9	0.5	7.5

If absorbance and activity were plotted on the same scale the plot would be dominated by activity, so you could change the units of enzyme activity to be compatible with the absorbance scale. However, to illustrate how to create a double graph, a plot with absorbance on the left hand axis and enzyme activity and pH together on the right hand axis will be constructed. Obviously this requires three separate objects, i.e., files for program **simplot**, and information to indicate which y axis is to be used for the individual files. For instance, you could create the following files using program **makmat**, and the data in the above table.
File 1: The first column together with the second column (as in test file `plot2.tf1`)
File 2: The first column together with the third column (as in test file `plot2.tf2`)
File 3: The first column together with the fourth column (as in test file `plot2.tf3`)

Then select program **simplot** and choose to make a double graph. Input the first file (absorbance against fraction) scaled to the left hand axis with the other two scaled to the right hand axis to get the first figure following. To transform the default **simplot** plot into the finished product of the second figure following proceed as follows:

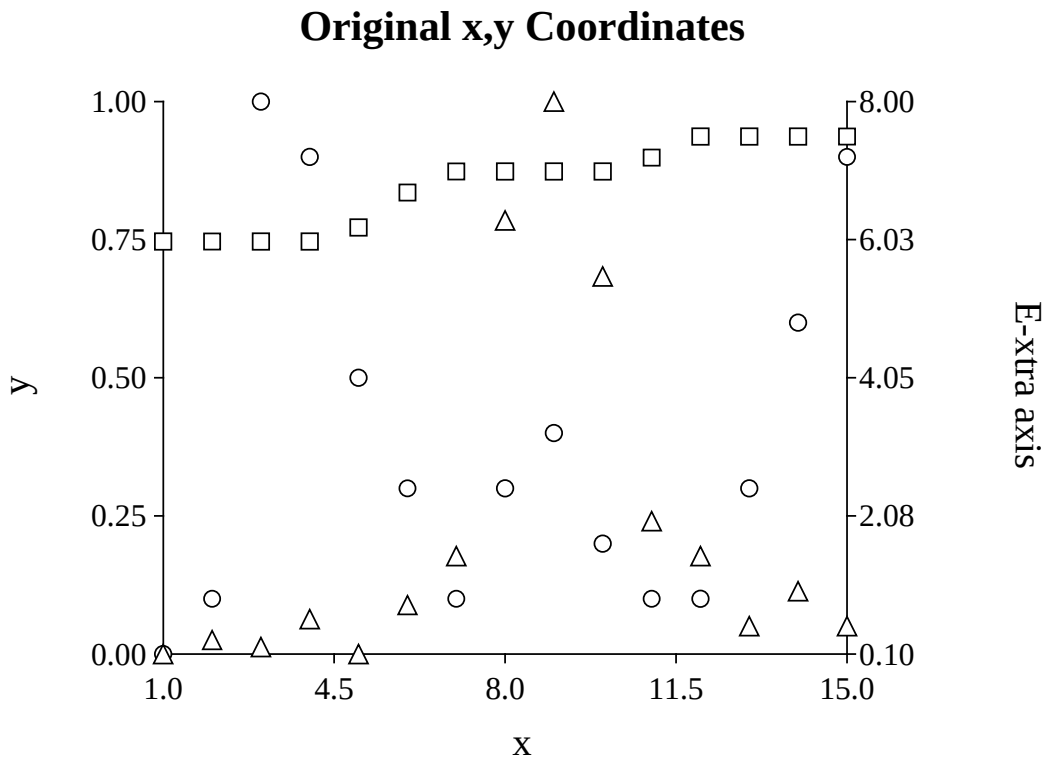
a) Edit the overall plot title and both plot legends.

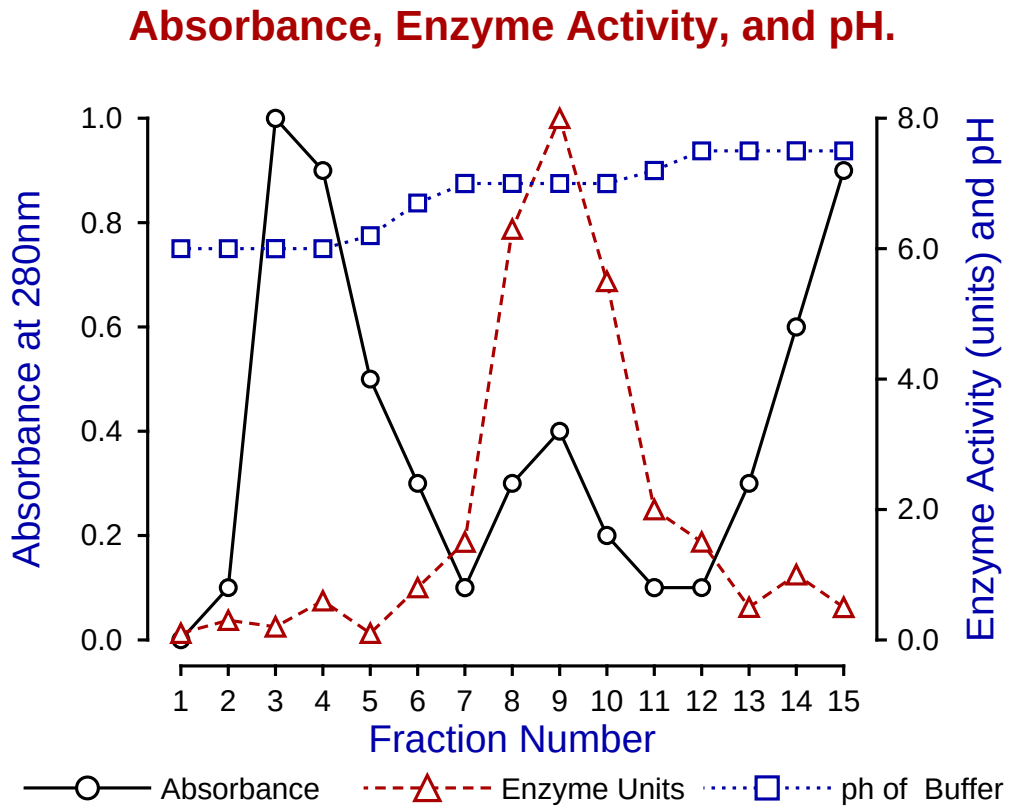
b) Edit the data ranges, notation and offset on the axes.

c) Edit the three symbol and line types corresponding to the three files.

d) Include an information panel and edit the corresponding keys.

e) Create the final PostScript file.

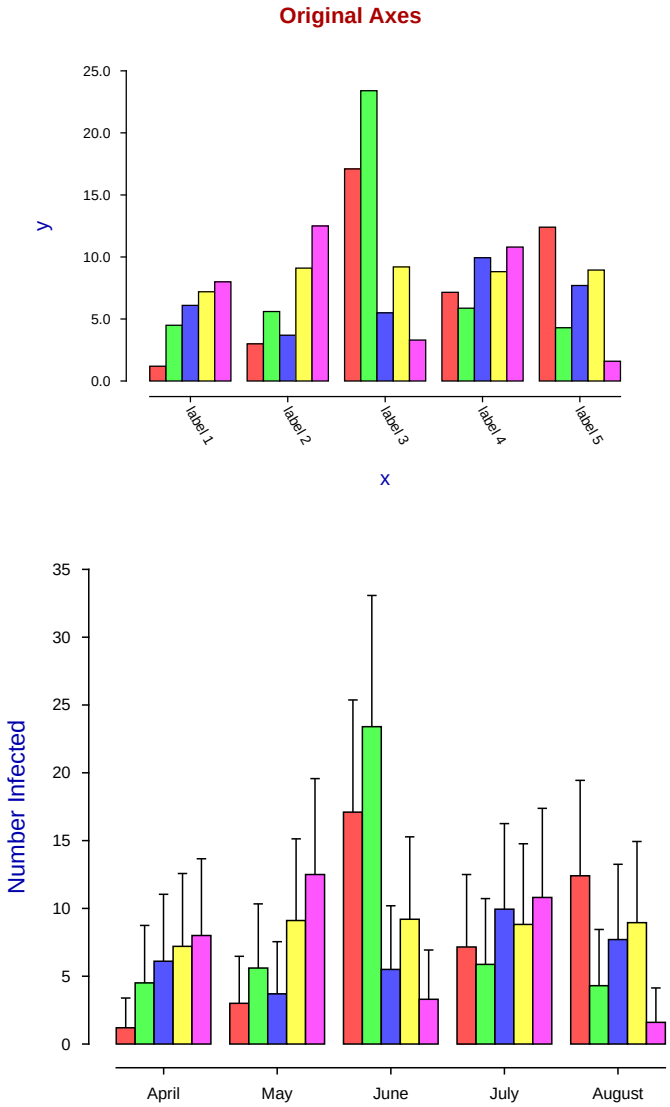

Original x,y Coordinates
1.00
0.75
0.50
0.25
0.00
y
8.00
6.03
4.05
2.08
0.10
Extra axis
1.0
4.5
8.0
11.5
15.0
x


Absorbance, Enzyme Activity, and pH.
Absorbance at 280nm
1.0
0.8
0.6
0.4
0.2
0.0
Enzyme Activity (units) and pH
8.0
6.0
4.0
2.0
0.0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
Fraction Number
Absorbance
Enzyme Units
ph of Buffer

13.2.8 Plotting error bars

Error bars with barcharts

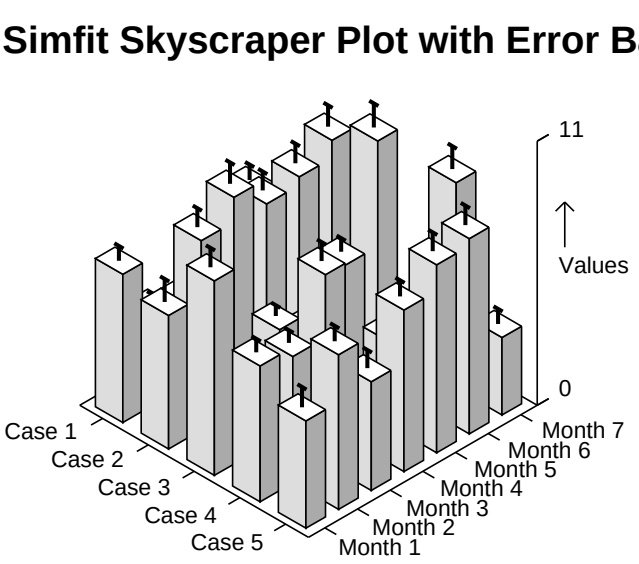
Barcharts with or without error bars can be created interactively from a table of values. For example, the upper figure below was generated from `matrix.tf1` as a default barchart using program **simplot**, while the lower figure has twice the square root of the bar values added.

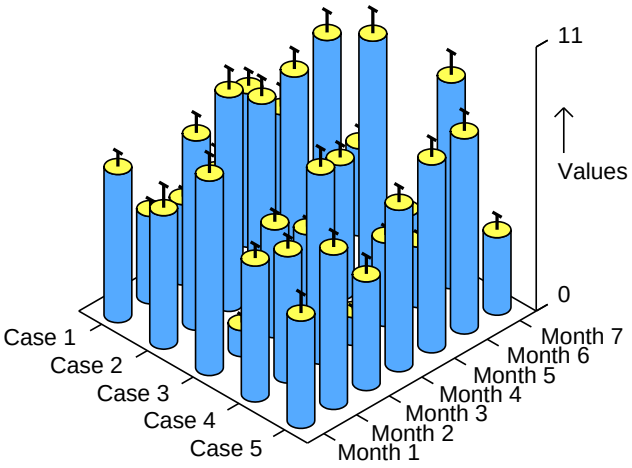


If the elements are measurements, the height of the bars would be means, while error bars would be calculated as 95% confidence limits, i.e., using a t distribution and assuming a normal distribution. Often one standard error of the mean is used instead of confidence limits to make the data look better, which is dishonest. If the elements are counts, approximate error bars could be added as twice the square root of the counts, i.e., assuming a Poisson distribution. Note that after creating barcharts from matrices in program **simplot** the temporary advanced barchart files generated to plot the bar chart can be saved.

Error bars with skyscraper and cylinder plots

Barcharts can be created for tables, $z(i, j)$ say, where cells are values for plotting as a function of x (rows) and y (columns). The x, y values are not required, as such plots usually require labels not numbers. The next figure shows the plot generated by **simplot** from the test file `matrix.tf2`.





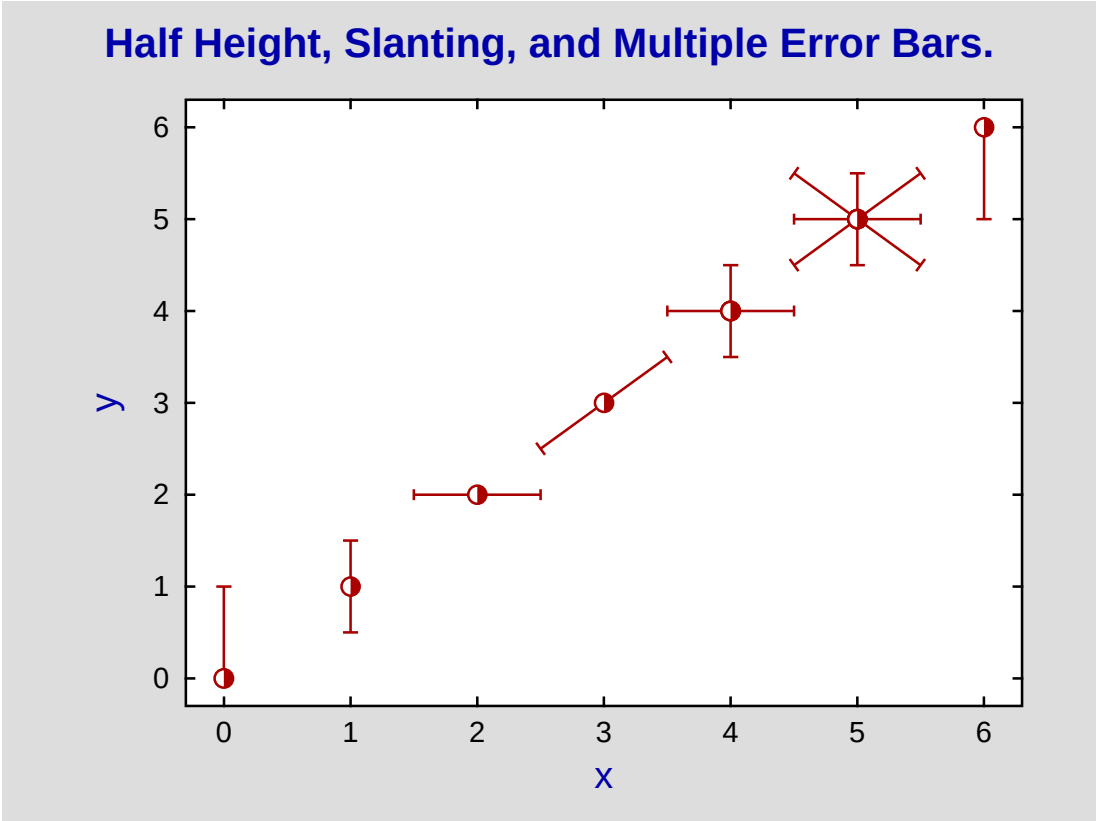
Errors are added from a file, and are calculated according to the distribution assumed. They could be twice square roots for Poisson counts, binomial errors for proportions or percentages, or they could be calculated from sample standard deviations using the t distribution for means. As skyscraper plots with errors are dominated by vertical lines, error bars are plotted with thickened lines, but a better solution is to plot cylinders instead of skyscrapers, as illustrated.

Slanting and multiple error bars

Error bar files can be created by program **editfl** after editing curve fitting files with replicates, and such error bars will be symmetrical, representing central confidence limits in the (x, y) space. But note that these error bars can become unsymmetrical or slanting as a result of a transformation, e.g., $\log(y)$ or Scatchard, using program **simplot**. Program **binomial** will, on the other hand, generate noncentral confidence limits, i.e., unsymmetrical error bars for binomial parameter confidence limits, and Log-Odds plots.

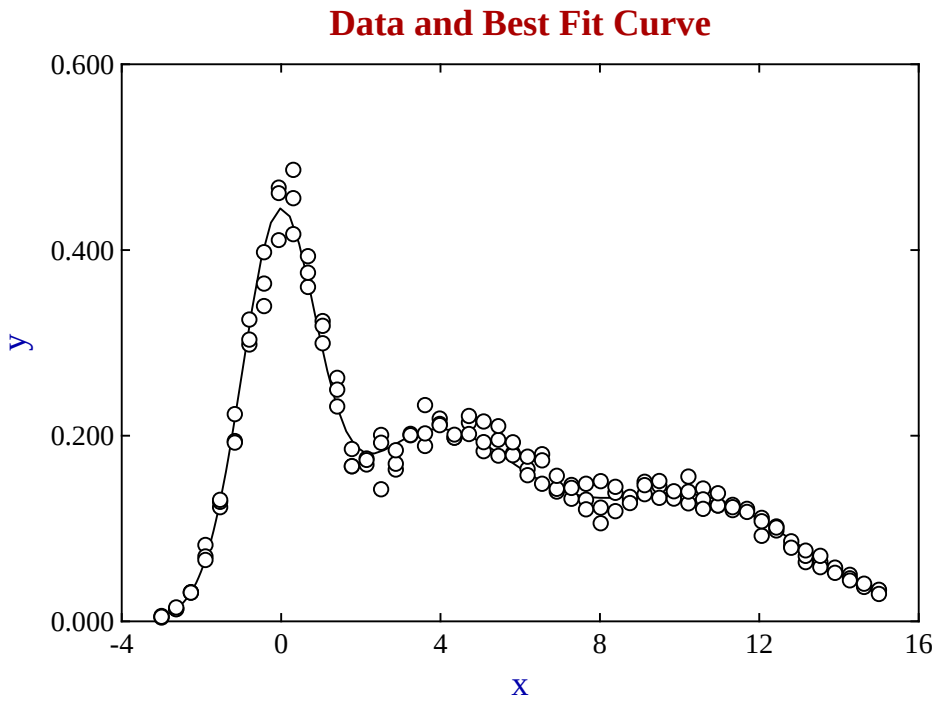
Sometimes it is necessary to plot asymmetrical error bars, slanting error bars or even multiple error bars. To do this, note that the error bar test file `errorbar.tf1` has four columns with the x coordinate for the plotting symbol, then the y-coordinates for the lower bar, middle bar and upper bar. However, the error bar test file `errorbar.tf2` has six columns shown in the table below, so that the (x_1, y_1), (x_2, y_2), (x_3, y_3) coordinates specified can create any type of error bar, even multiple error bars, as will be seen in the next figure.

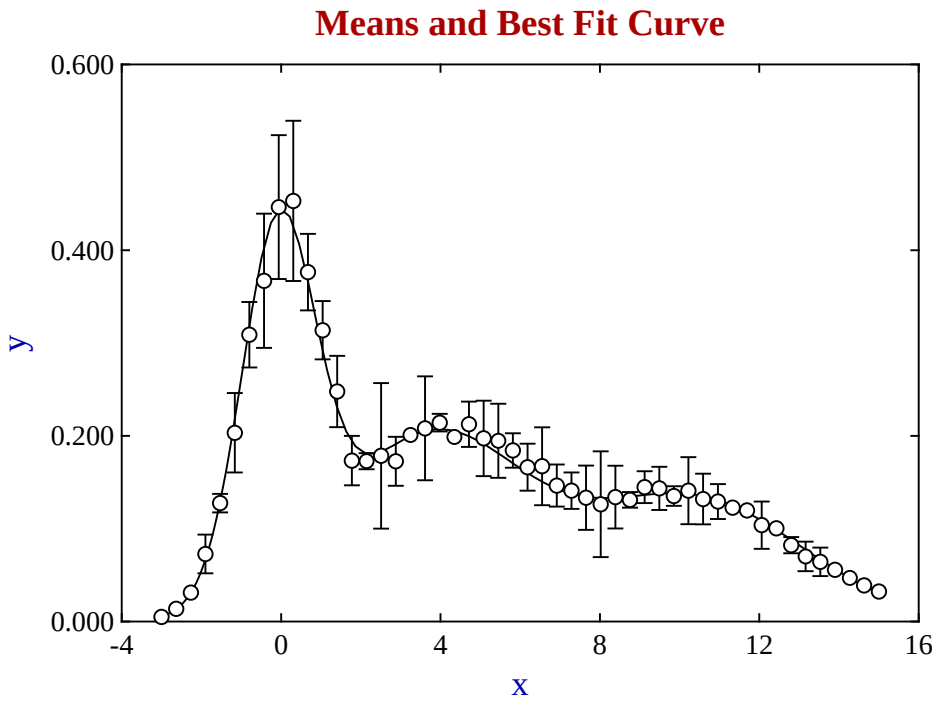
x_1	x_2	x_3	y_1	y_2	y_3
0.0	0.0	0.0	0.0	0.0	1.0
1.0	1.0	1.0	0.5	1.0	1.5
1.5	2.0	2.5	2.0	2.0	2.0
2.5	3.0	3.5	2.5	3.0	3.5
3.5	4.0	4.5	4.0	4.0	4.0
4.0	4.0	4.0	3.5	4.0	4.5
5.0	5.0	5.0	4.5	5.0	5.5
4.5	5.0	5.5	5.0	5.0	5.0
4.5	5.0	5.5	4.5	5.0	5.5
4.5	5.0	5.5	5.5	5.0	4.5
6.0	6.0	6.0	6.0	6.0	5.0



Calculating error bars interactively

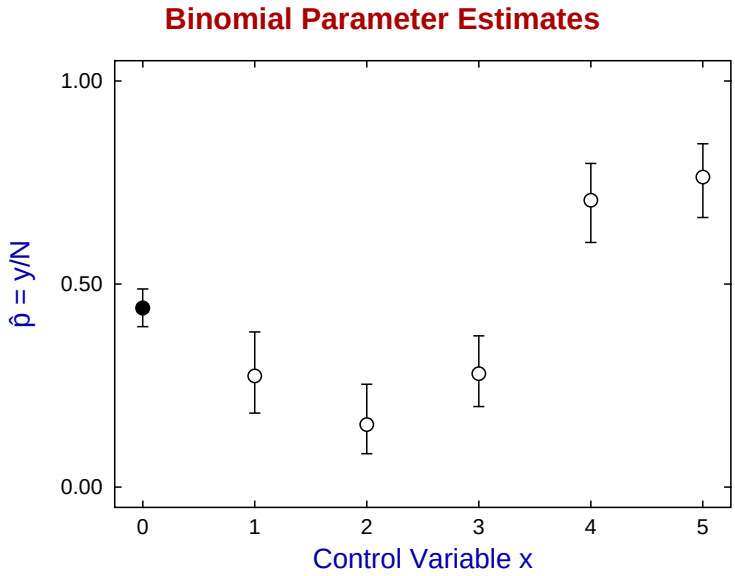
The next figure shows the best fit curve estimated by **qnfit** when fitting a sum of three Gaussians to the test file `gauss3.tf1`. Note that all the data must be used for fitting, not means. Programs **editfl** and **simplot** can generate error bar plotting files from such data files with replicates, as illustrated for 95% confidence limits.





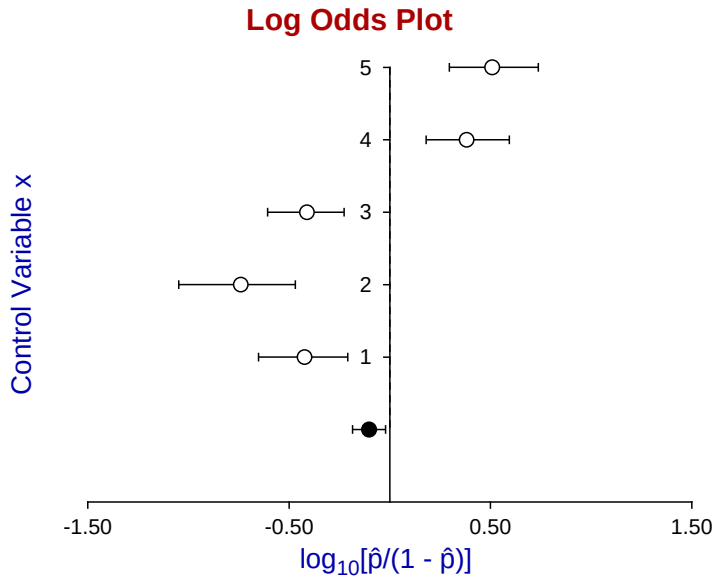
Plotting binomial error bars

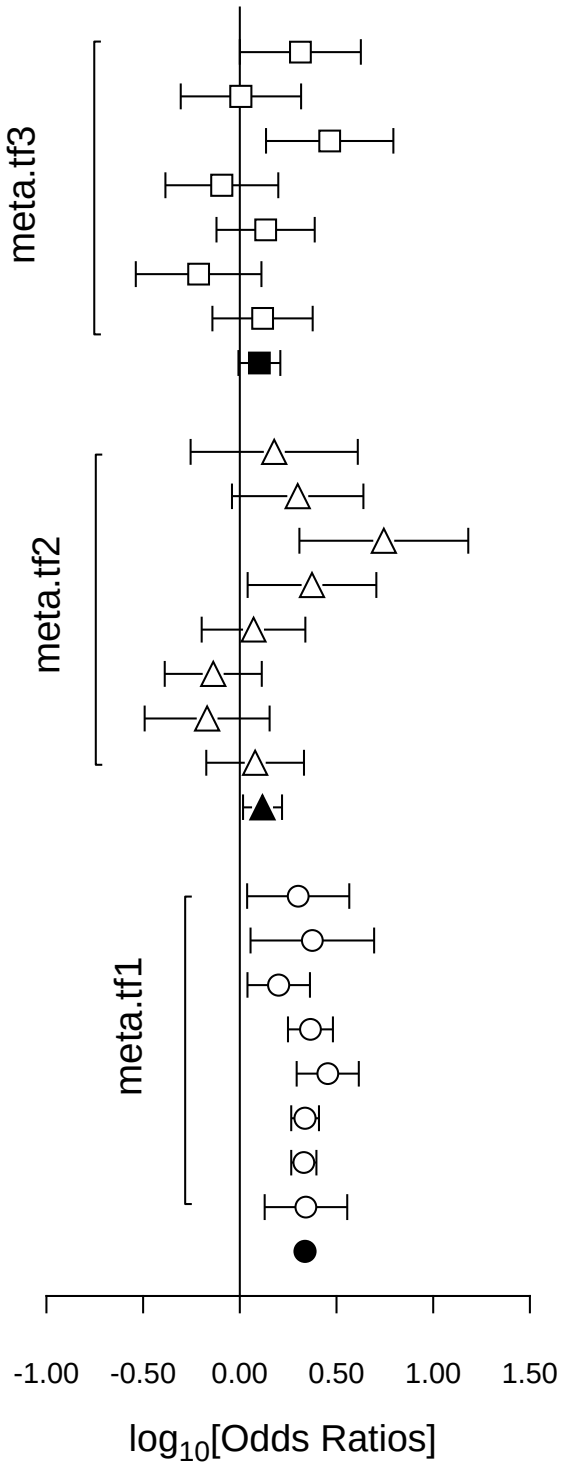
The plot below shows binomial parameter estimates for y successes in N trials. The error bars represent exact, unsymmetrical confidence limits, not those calculated using the normal approximation.



Plotting Log-Odds error bars

Binomial error bars can also be manipulated by transforming the estimates $\hat{p} = y/N$ and confidence limits. For instance, the ratio of success to failure (i.e. Odds $y/(N-y)$) or the logarithm (i.e. Log Odds) can be used, as in the next figure, to emphasize deviation from a fixed p value, e.g. $p = 0.5$ with a log-odds of 0. This figure was created from a simple log-odds plot by using the [Advanced] option to transfer the $x, \hat{p}/(1-\hat{p})$ data into **simplot**, then selecting a reverse y-semilog transform.





It is often useful to plot Log-Odds-Ratios, so the creation of the adjacent figure will be outlined.

(1) **The data**
Test files `meta.tf1`, `meta.tf2`, and `meta.tf3` were analyzed in sequence using the SIMFIT Meta Analysis procedure. Note that, in these files, column 3 contains spacing coordinates so that data will be plotted consecutively.

(2) **The ASCII coordinate files**
During Meta Analysis, $100(1-\alpha)\%$ confidence limits on the Log-Odds-Ratio resulting from a 2 by 2 contingency tables with cell frequencies n_{ij} can be constructed from the approximation $\hat{e}$ where

$$\hat{e} = Z_{\alpha/2}\sqrt{\frac{1}{n_{11}}+\frac{1}{n_{12}}+\frac{1}{n_{21}}+\frac{1}{n_{22}}}.$$

When Log-Odds-Ratios with error bars are displayed, the overall values (shown as filled symbols) with error bars are also plotted with a x coordinate one less than smallest x value on the input file. For this figure, error bar coordinates were transferred into the project archive using the [Advanced] option to save ASCII coordinate files.

(3) **Creating the composite plot**
Program **simplot** was opened and the six error bar coordinate files were retrieved from the project archive. Experienced users would do this more easily using a library file of course. Reverse y-semilog transformation was selected, symbols were chosen, axes, title, and legends were edited, then half bracket hooks identifying the data were added as arrows and extra text.

(4) **Creating the PostScript file**
Vertical format was chosen then, using the option to stretch PostScript files, the y coordinate was stretched by a factor of two.

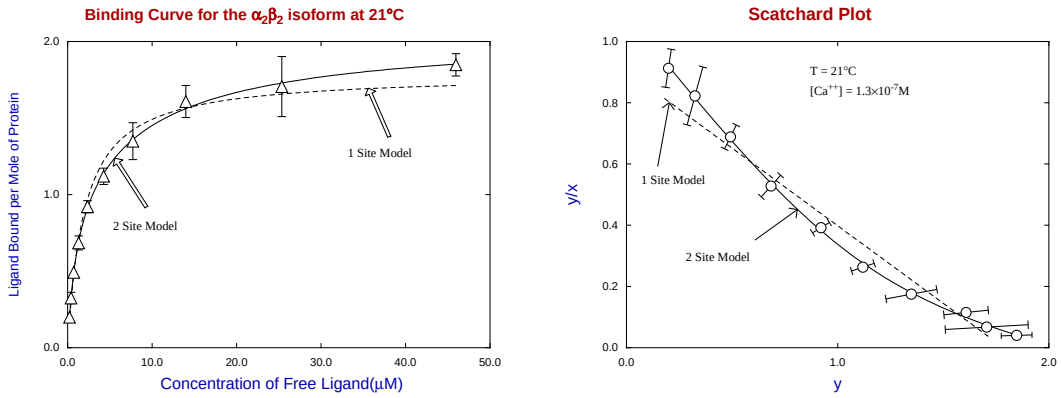
(5) **Editing the PostScript file**
To create the final PostScript file for LaTeX a tighter bounding box was calculated using **gsview** then, using **notepad**, clipping coordinates at the top of the file were set equal to the BoundingBox coordinates, to suppress excess white space. This can also be done using the [Style] option to omit painting a white background, so that PostScript files are created with transparent backgrounds, i.e., no white space, and clipping is irrelevant.

13.2.9 Plotting labels

No plot is complete without labels to identify the data in a title, with specific information in the legends, and possibly additional text within the data plotting range to identify special features. However, the labels may not be simple character strings but may contain subscripts, superscripts, or mathematical symbols, which require an associated template to indicate such additional enhancements, as discussed next.

Labels and corresponding templates

Consider the titles, legends, and additional text in the next figures from fitting a two site binding model displayed in standard space and Scatchard transformed space.



To appreciate how to edit labels and associate template keys to introduce subscripts, superscripts, maths, etc. consider the title, *x* legend and *y* legend in the standard plot are as follows

```
Binding Curve for the a2b2 isoform at 21@C
00000000000000000000006161000000000000060

Concentration of Free Ligand(lM)
00000000000000000000000000000300

Ligand Bound per Mole of Protein
00000000000000000000000000000000
```

while the additional text and templates for the Scatchard plot are as shown next.

```
T = 21@C
00000030

[Ca++] = 1.3*10-7M
000220000000300220
```

Clearly a 0 indicates a normal character, a template value 1 indicates a subscript while a value 2 indicates a superscript, and these changes would be made by altering the template key interactively.

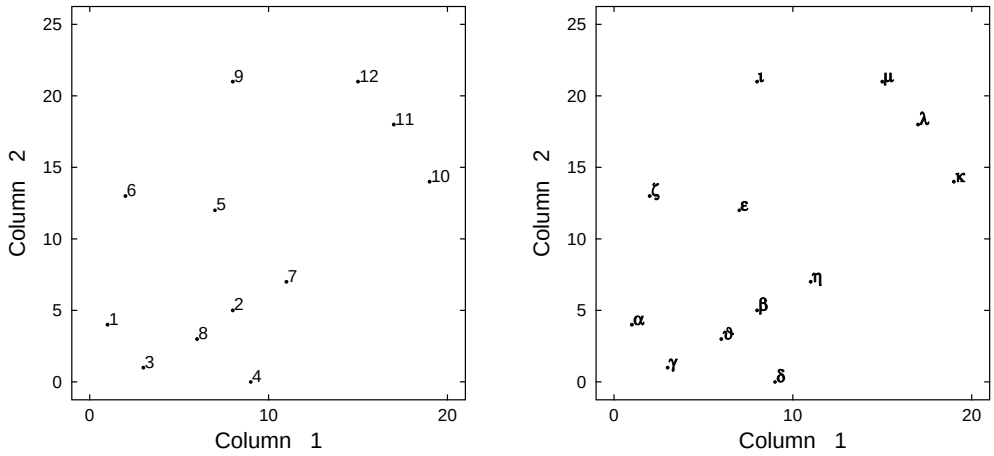
However the keys equal to 3 for maths or 6 for bold maths are done automatically by selecting the required character from a SimFIT symbol or maths character substitution table.

Labels in plots

Labels in SIMFIT plots are text strings (with associated template strings) that do not have arbitrary positions, but are plotted at default coordinates to identify the data. Some examples would be as follows.

- Labels adjacent to segments in a pie chart.
- Labels on the X axis to indicate groups in bar charts.
- Labels on the X axis to identify clusters in dendrograms.
- Labels plotted alongside symbols in 2D plots, such as principal components.

Test files such as `cluster.tf1` illustrate the usual way to supply labels appended to data files in order to over-ride the defaults set from the configuration options, but sometimes it is convenient to supply labels interactively from a file, or from the clipboard, and not all procedures in SIMFIT use the labels supplied appended to data files. The next figures illustrate this.

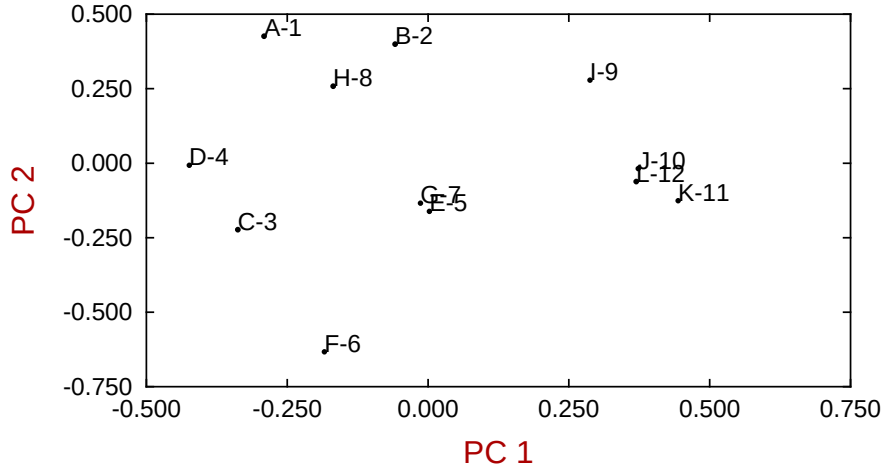


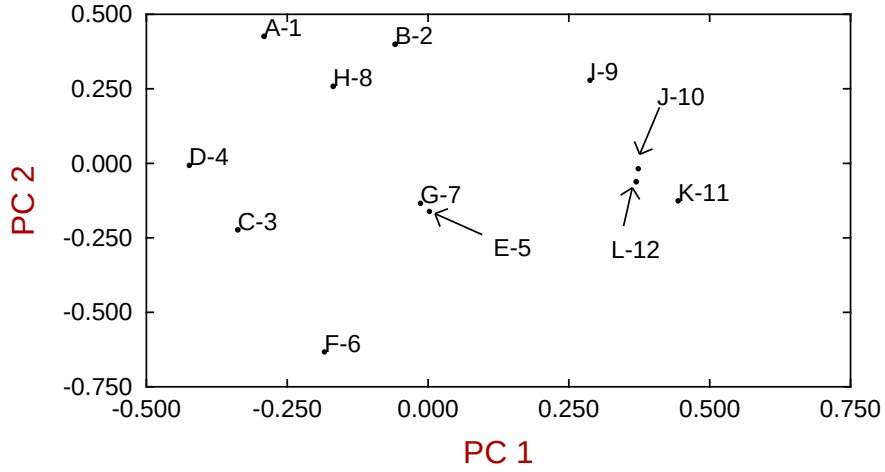
Test file `cluster.tf1` was input into the procedure for exhaustive analysis of a matrix in **simstat**, and the option to plot columns as an advanced 2D plot was selected. This created the left hand figure, where default integer labels indicate row coordinates. Then the option to add labels from a file was chosen, and test file `labels.txt` was input. This is just lines of characters in alphabetical order to overwrite the default integers. Then the option to read in a template was selected, and test file `templates.txt` was input. This just contains a succession of lines containing 6, indicating that alphabetical characters are to be plotted as bold maths symbols, resulting in the right hand figure. The recommended procedure follows.

1. Write the column of case labels, or row of variable labels, from your data-base or spread-sheet program into an ASCII text file.
2. This file should just consist of one label per line and nothing else (like `labels.txt`)
3. Paste this file at the end of your SIMFIT data matrix file, editing the extra line counter (as in `cluster.tf1`) as required.
4. If there are n lines of data, the extra line counter (after the data but before the labels) must be at least n to use this label providing technique.
5. Alternatively use the more versatile `begin{labels} ... end{labels}` technique.
6. Archive the labels file if interactive use is anticipated as in the previous figure.
7. If Special symbols or accents are required, a corresponding templates file with character display codes can be prepared.

Adjusting the position of labels

As an example, principal components for multivariate data can be explored by plotting scree diagrams and scattergrams after using the calculations options in program **simstat**. If labels are appended to the data file, as with `cluster.tf2`, they can be plotted, as in the next figures.





Labels similar to the ones that are usually plotted along the x axis are used to label the points, but displaced to one or other side of the plotting symbol for legibility. Colors are controlled from the [Colour] options as these are linked to the color of the symbol plotted, even if the symbol is suppressed. The font is the one that would be used to label the x axis if labels were plotted instead of numbers. Clearly arbitrary labels cannot be plotted at the same time on the x axis. Often it is required to move the labels because of clashes, as above. This is done by using the labels editing function, setting labels that clash equal to blanks, then using the normal mechanism for adding arbitrary text and arrows to label the coordinates in the principal components scattergram. To facilitate this process, the default text font is the same as the axes numbering font.

Alternatively, the plotting menus provide the option to displace any labels by defining parameters to shift individual labels horizontally or vertically. The movement is necessarily limited by a numerical scale or slider control, and some versions of SIMFIT allow label movement by dragging with the red arrow. Clearly, setting the additional x and y displacements to zero restores the label to the original position adjacent to the symbol being plotted.

13.2.10 Plotting mathematical equations interactively

Plotting mathematical equations over a range is often required and the programs and techniques available to do this are as follows.

1. Program **makdat**
   After selecting a model from the compiled library or as a user-defined model, plots can be displayed over a chosen range.
2. Program **deqsol**
   This is similar to the using program **makdat** but is preferred if it is wished to plot systems of nonlinear differential equations, or phase portraits or orbits for autonomous systems.
3. Program **usermod**
   This has similar functionality to program **makdat** except that it allows users to defines a model or set of models interactively. Once a model has been developed it can be archived for future use, so this is the only technique that will be described in this document.

Defining a mathematical model interactively

From the main SimFIT menu use the option [A/Z] to open program **usermod** and observe that there is an option to define a model interactively, and when this has been done the mathematical model can be checked for correct syntax, plotted over a chosen range or archived for retrospective use. Some simple examples to illustrate the functionality of program **usermod** will now be given. However note that you will have to be prepared to input the following values.

- The number of equations $NEQN \geq 1$
  The equations will be defined as $f(1), f(2), \ldots, f(NEQN)$.
- The number of variables $NVAR \geq 1$ (or differential equations which assumes $NVAR = 1$)
  The variables will be either $NVAR = 1$ using the symbol x for the independent variable to plot 2 dimensional curves, or $NVAR = 2$ using the symbols x and y for the independent variables to display 3 dimensional surfaces.
- The number of parameters $NPAR \geq 0$
  The parameters will be $p(1), p(2), \ldots, p(NPAR)$ and these can be defined and varied interactively if it is required to study the effect of parameter values on the plots.

Example 1: A quadratic equation

The mathematical model will be the quadratic

$$f(x) = x^2 - 1.$$

So you have to create a user-defined model with these characteristics:

- One equation $NEQN = 1$
- One variable $NVAR = 1$
- No parameters $NPAR = 0$

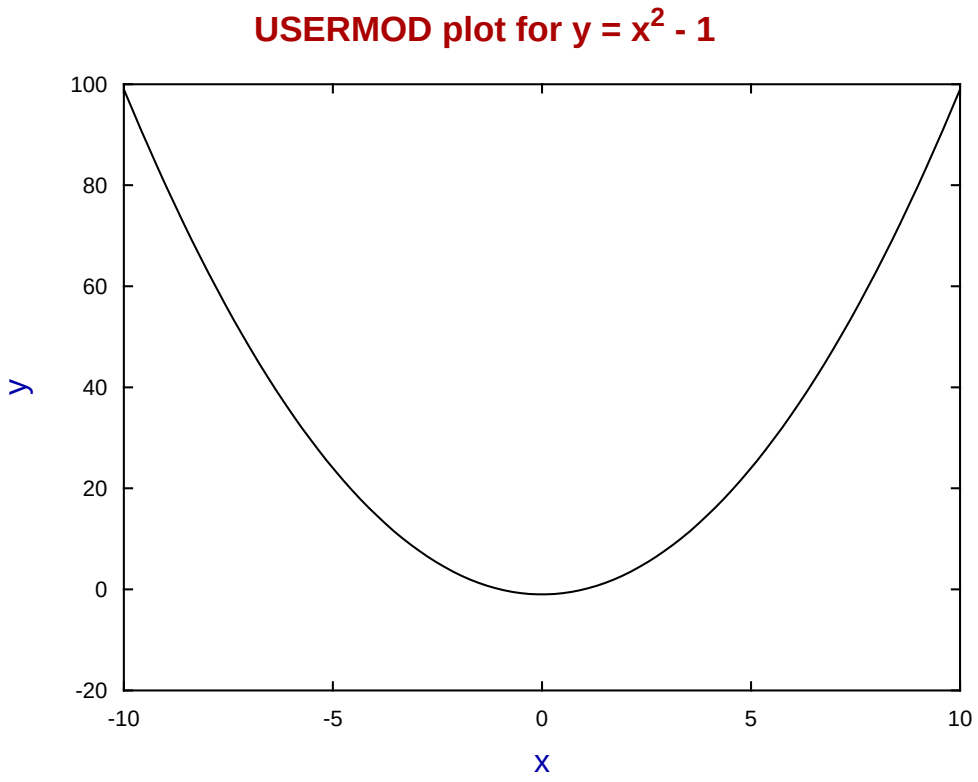
then the following unfinished model file will be displayed.

```
%
This is a default template for a user-defined-model file.
%
1 equation
1 variable
0 parameters
%
begin{expression}
f(1) =
end{expression}
%
```

The empty field is then filled in to replace the string `f(1) =` by `f(1) = x^2 - 1` as shown below.

```
%
The model y = x^2 - 1 = (x - 1)(x + 1).
%
1 equation
1 variable
0 parameters
%
begin{expression}
f(1) = x^2 - 1
end{expression}
%
```

This is then checked for consistency and the option is provided to plot the model as in the next figure.



Example 2: Four trigonometric functions

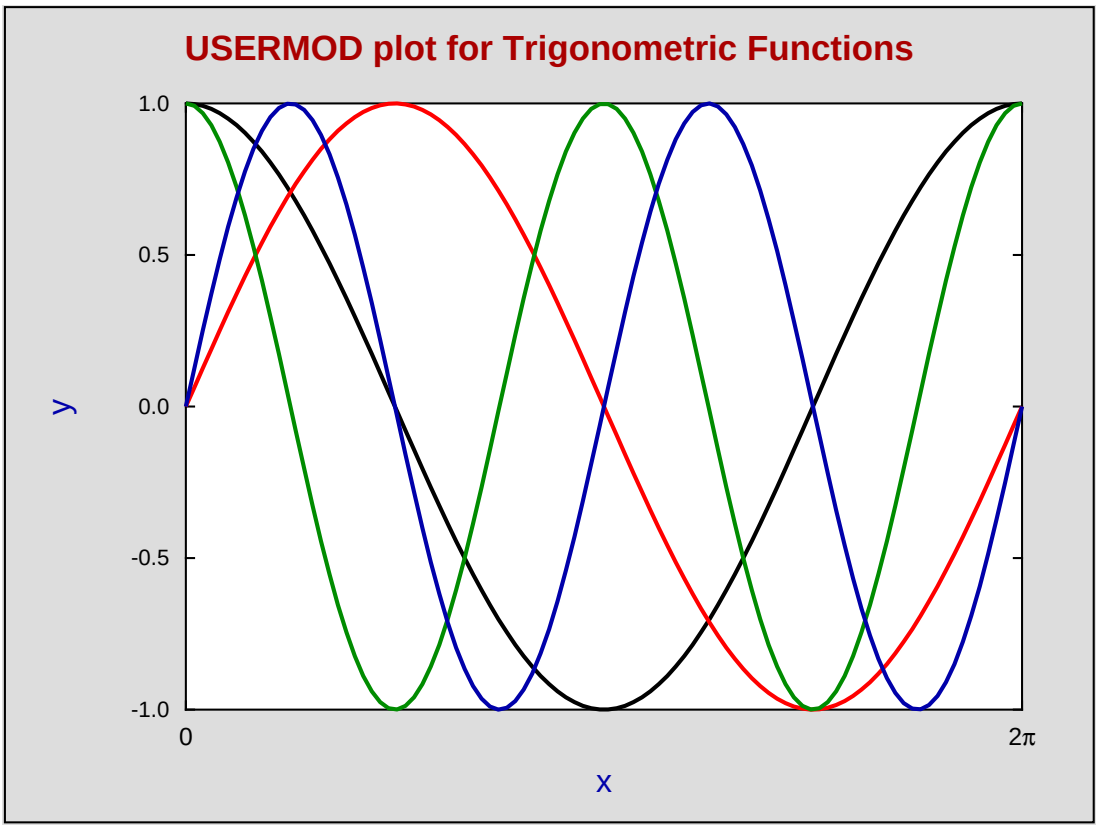
Selecting four functions of one variables with four parameters and then choosing

$$f_1(x) = p_1 \cos x,\ f_2(x) = p_2 \sin x,\ f_3(x) = p_3 \cos 2x,\ f_4(x) = p_4 \sin 2x$$

leads to the following model.

```
 %
f(1)=p(1)cos(x), f(2)=p(2)sin(x), f(3)=p(3)cos(2x), f(4)=p(4)sin(2x)
 %
 4 equations
 1 variable
 4 parameters
 %
 begin{expression}
 f(1) = p(1)cos(x)
 f(2) = p(2)sin(x)
 f(3) = p(3)cos(2x)
 f(4) = p(4)sin(2x)
 end{expression}
 %
```

This is then checked for consistency and the option is provided to plot the model as in the next figure using the default values of 1 for all the parameters.



The border, colors, line thicknesses, and title were added using the Advanced Graphics option and the x labels were plotted as characters instead of numbers and edited to show the range $(0 \le x \le 2\pi)$.

Example 3: A quadratic surface with contours

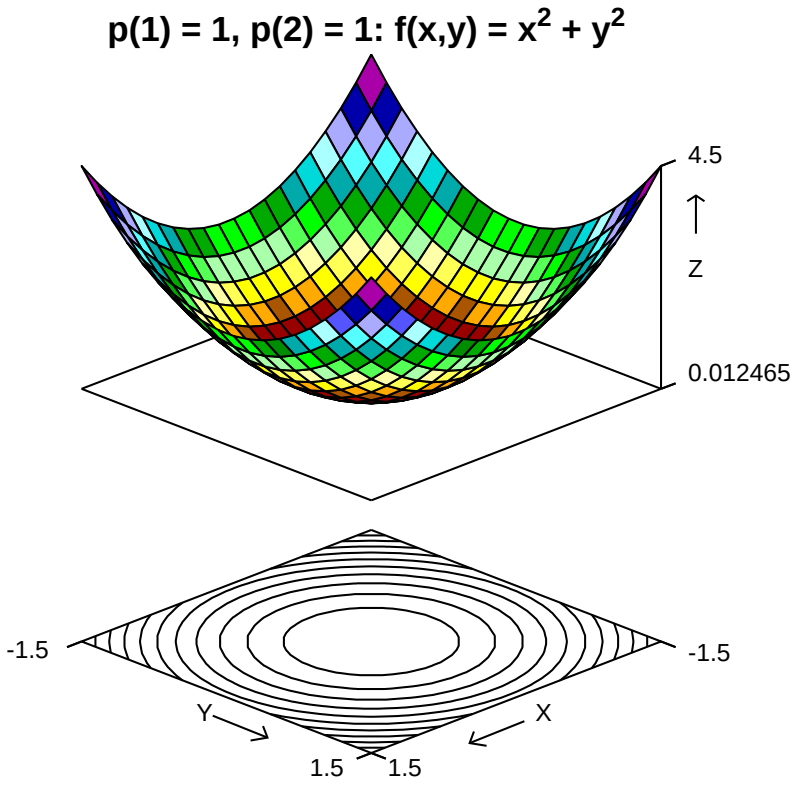
Select one function of two variables with two parameters then create the following model.

```
%
A function of two variables.
%
1 equation
2 variables
2 parameters
%
begin{expression}
f(1) = p(1)x^2 + p(2)y^2
end{expression}
%
```

Note that the default parameters are

$$p_1 = 1, p_2 = 1$$

which defines the following convex paraboloid.



However it is possible to edit the parameters using the same model to create a hyperboloid by setting

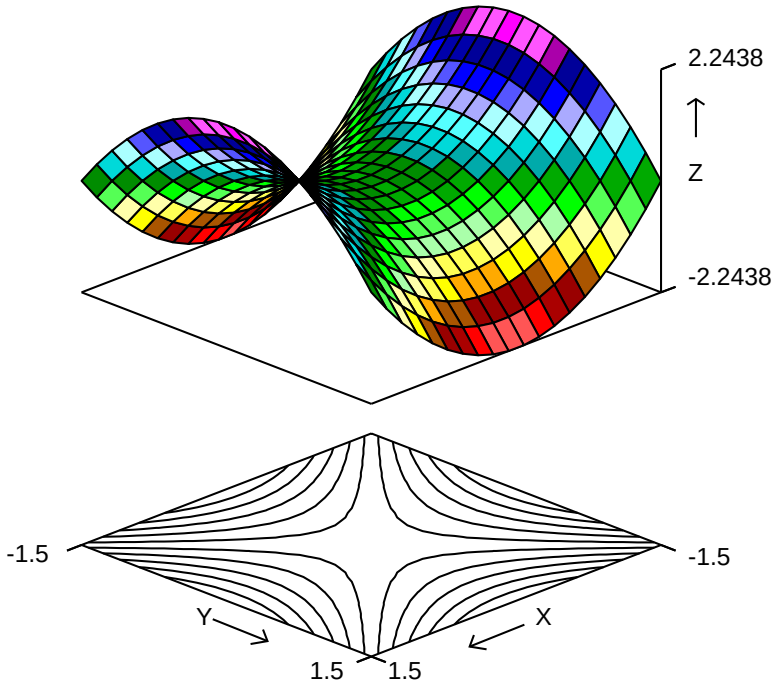
$$p_1 = 1, p_2 = -1$$

or a concave paraboloid by using

$$p_1 = -1, p_2 = -1$$

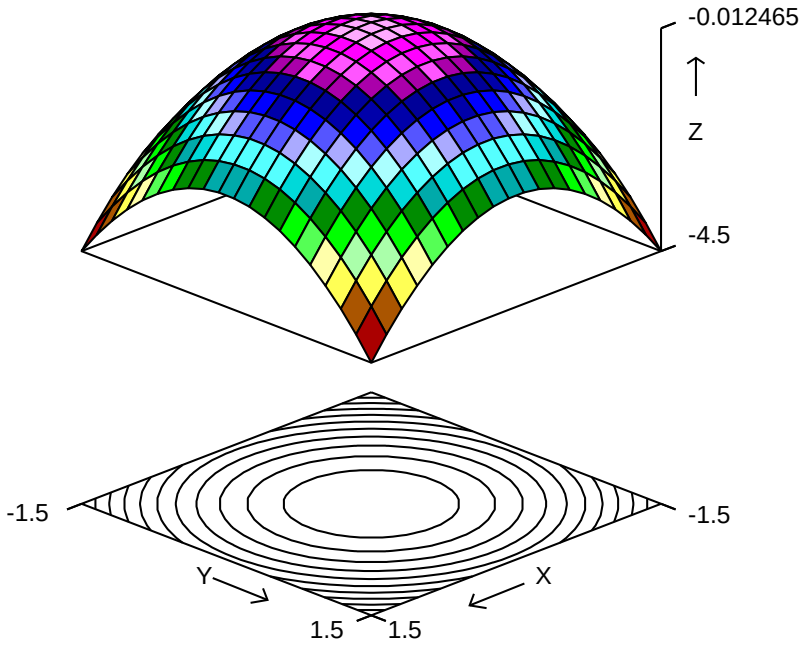
as demonstrated in the next two plots.

p(1) = 1, p(2) = -1: f(x,y) = x^2 - y^2



---

p(1) = -1, p(2) = -1: f(x,y) = - (x^2 + y^2)



13.3 Advanced graphics

Tutorials and worked examples for simulation, curve fitting, statistical analysis, and plotting.
https://simfit.uk
https://simfit.org.uk
https://simfit.silverfrost.com

13.3.1 Configuration files, templates, and metafiles

It is often required to sculpture a graph for archive use or publication, and this can prove tiresome if it has to be done repeatedly.

Consider, for example, the steps needed to generate a logodds plot from the default plot that SIMFIT creates.

- Transfer the (x, y) values, i.e. $\hat{p}/(1-\hat{p})$ as a function of x, into advanced graphics mode, i.e. the indirect **simplot** interface.
- Choose the reverse y-semilog plot.
- Change the title and legends.
- Suppress or move the y axis.
- Add labels or similar additional features.
- Save a graphics file.

SIMFIT provides the following procedures to facilitate such processes.

1. **Create a configuration file.**
   This will contain all the special features added to the default graph by editing in the advanced **simplot** interface. It will include all the extra features added such as arrows, labels, graphical objects, information panel, and any mathematical or accented text, but it will not contain any of the data plotted. In particular it will contain the ranges of data, x_{min} to x_{max} and y_{min} to y_{max}, that were being plotted.
2. **Create a metafile.**
   This will be a configuration file as just described, but in addition will have all the data added.
3. **Read in a configuration file in comprehensive mode.**
   This will restore all the graphics features stored in the configuration file, including the ranges of data. This means that a configuration file should only be read in when the data have the same ranges as when the configuration file was created, otherwise unwanted effects will be created.
4. **Read in a configuration file in template mode.**
   This will re-install all the graphical features saved in the configuration file except for the data ranges. This means that the template can be used with data sets with different data ranges than those in use when the configuration file was created.
5. **Read in a SIMFIT metafile into program simplot.**
   This will create a complete graph with all features and data that were present when the metafile was created, and this is the recommended way to interrupt editing a graph in the advanced **simplot** mode in order to resume later for retrospective editing.

   Note that, if a metafile is read directly into program **simplot**, it will create the full graph, but if it is read into advanced graphics mode it will just act as a configuration file.

Advice

If you decide to read in a configuration file then SIMFIT will advise you that you can, at that stage, create a configuration file so that, in the event that the configuration file introduced creates unsatisfactory effects, you can undo these and return to the previous state. If you do intend to use the configuration file method, you must read in the configuration file

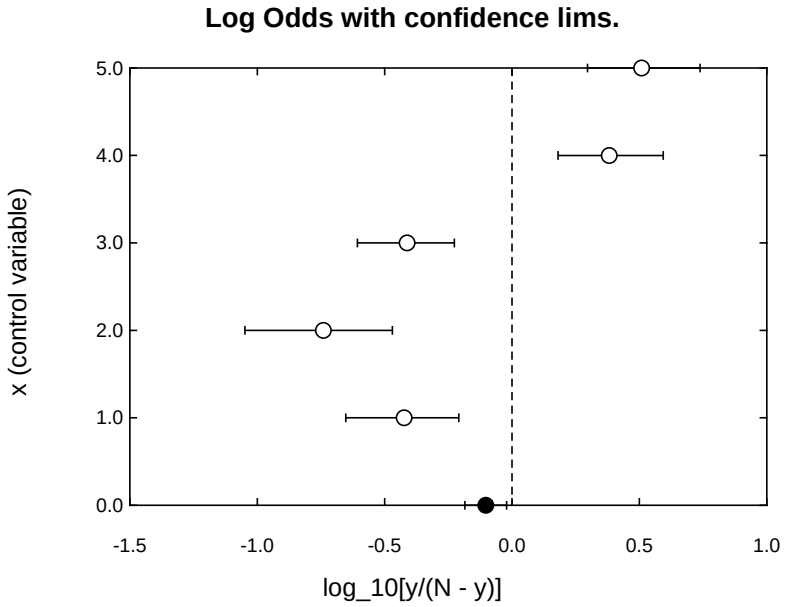
immediately the advanced interface has been opened, and appreciate the difference between reading in a configuration file in comprehensive or template mode, as follows.

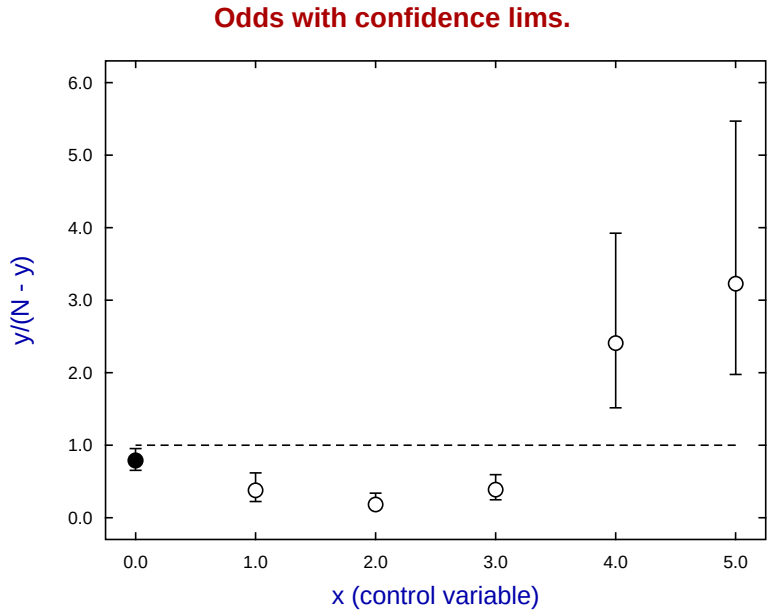
Using the ***comprehensive mode*** *should only be contemplated when the data being edited have exactly the same data ranges as those that were in operation when the configuration file was created and so, in general, it is usually safer to use the* ***template mode*** *so that the existing data ranges are not changed.*

13.3.2 Log Odds plot

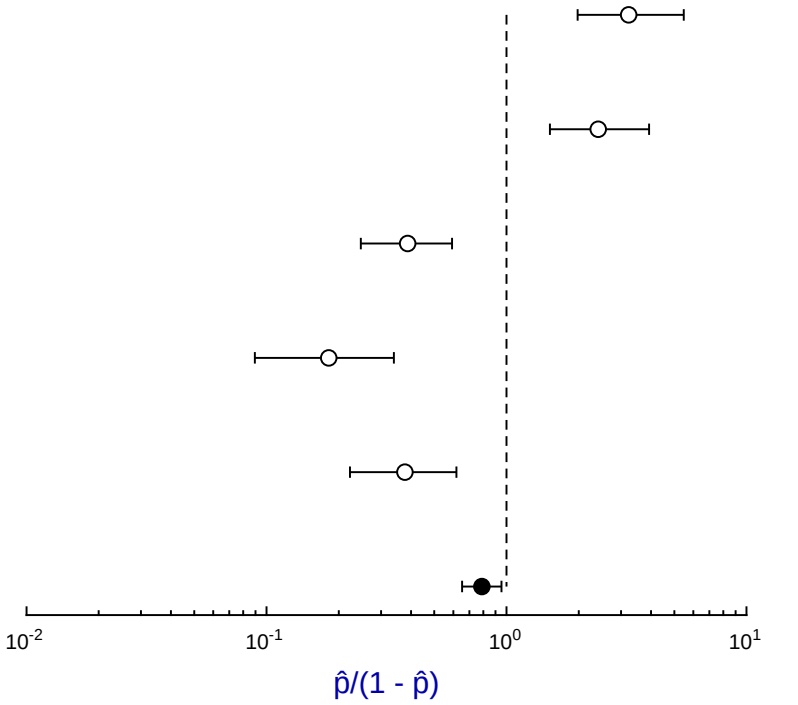
The log odds plot is used to display the ratio of success to failure in a sequence of binomial investigations, using a logarithmic scale to base 10 in order to facilitate analysis of the differences in orders of magnitude. Frequently the *y* axis is moved to a central position, or an additional vertical line is added to indicate the point where success and failure are equally likely.

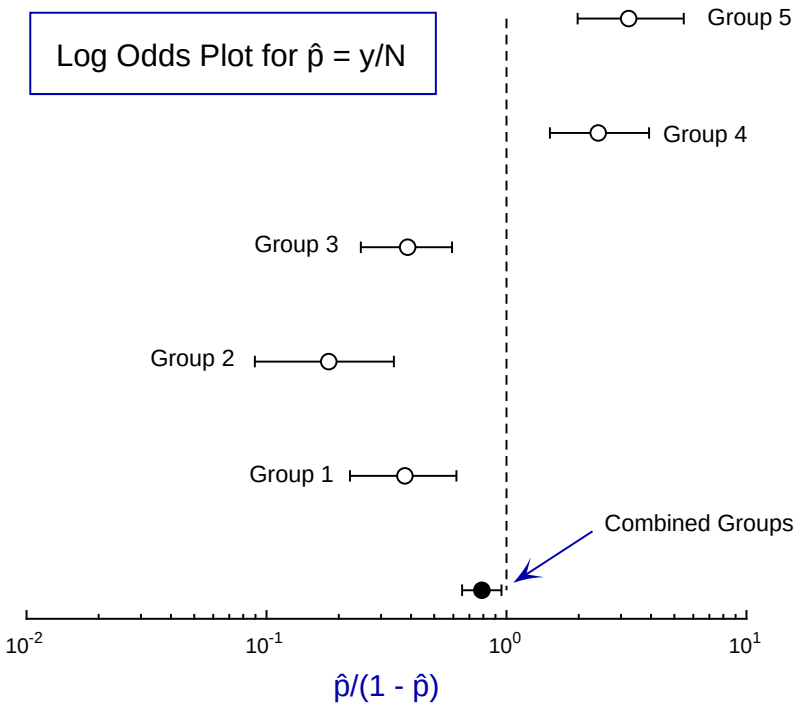
Using program **simstat** in binomial proportions mode with the default data set displays the following graph, which is followed by the results from transferring to advanced editing.





Reading in the configuration file `logodds.cfg` as a short cut to tedious editing results in the next graph, while creating the metafile `metafile.tf9` allows immediate production of the final graph either by directly reading the metafile into program **simplot**, or by reading it in as a configuration file to configure the default graph after transferring to advanced graphics.

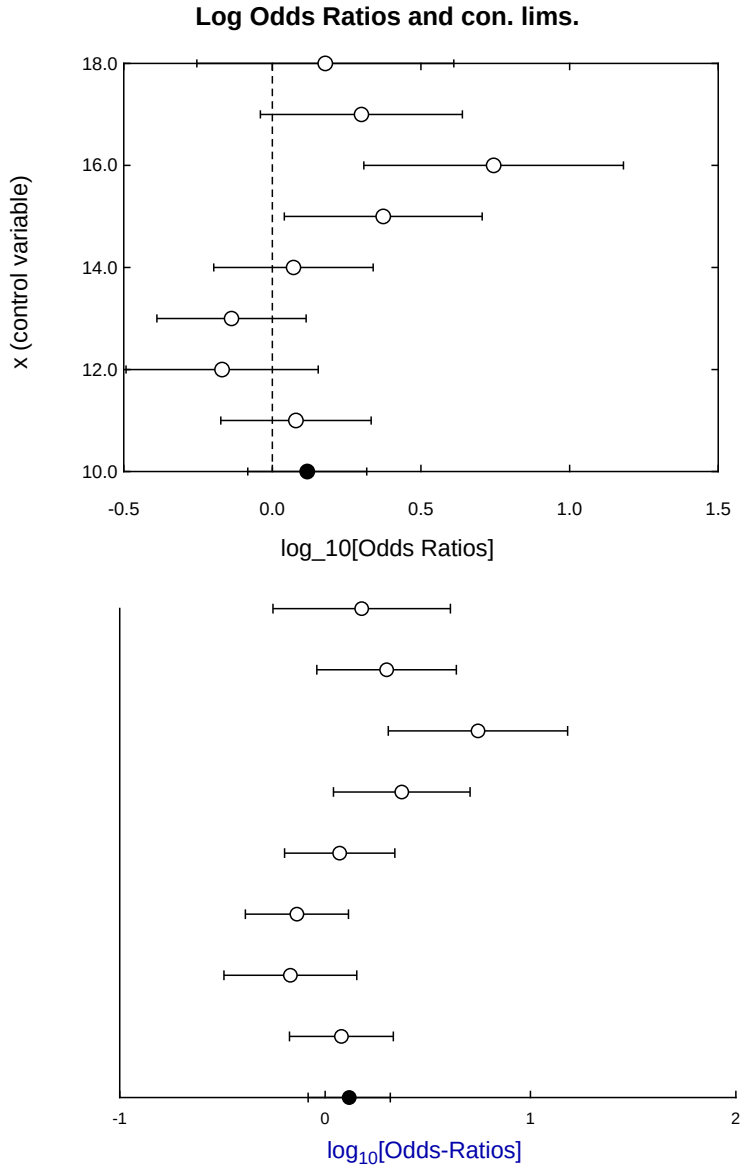




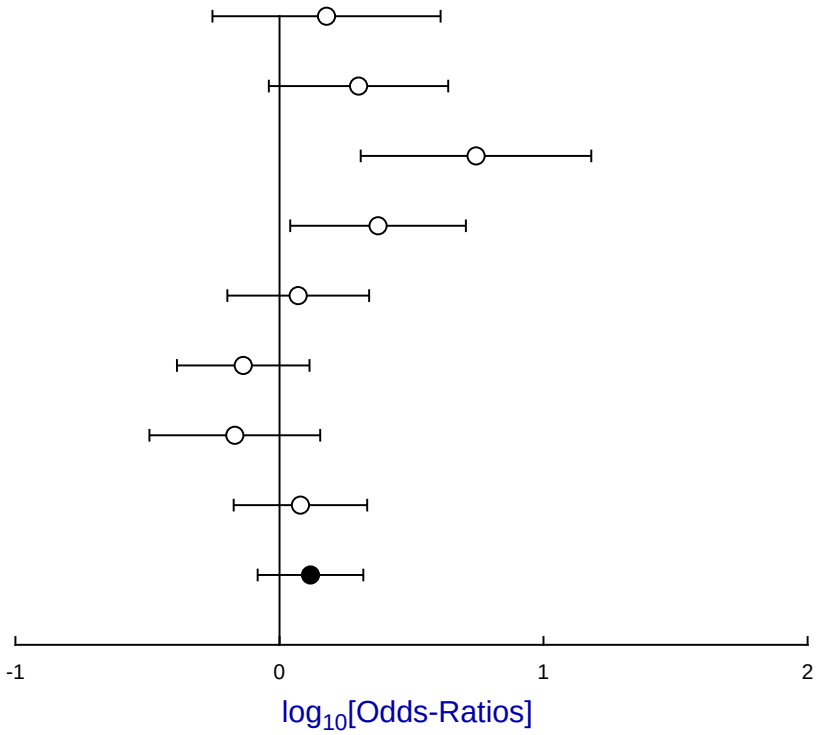
13.3.3 Log Odds Ratios Forest plot

The log odds ratios plot is used to display groups of ratios of odds (i.e. successes/failures) in sequences of binomial investigations, such as in meta analysis. Usually a logarithmic scale to base 10 is used in order to facilitate analysis of the differences in orders of magnitude. This tutorial also demonstrates an important difference between using graphics configuration files in **template** mode and in **comprehensive** mode.

Using program **simstat** in meta analysis mode with the test data in `meta.tf2` (instead of the default data set `meta.tf1`) displays the following graph, which is followed by the results from transferring to advanced editing then reading in the configuration file `logoddsratios.cfg` in the **template** mode.



However, reading in the configuration file `logoddsratios.cfg` in the **comprehensive** mode as a short cut to tedious editing results in the next more detailed graph.



The difference between this plot and the previous one highlights the difference between using a configuration file in **template** mode and in **comprehensive** mode.

In template mode the position and range of axes together with the number of tick marks is calculated from the existing data and is therefore applicable to any meta analysis data. In meta analysis we have pairs of binomial analyzes in order to determine if groups of pairs of data sets represent the same ratio of successes to failures, rather than equality of binomial parameter estimates. For instance, sample 1 would have y_1 successes in a sample size of N_1 giving the parameter estimate $\hat{p}_1 = y_1/N_1$, while sample 2 would generate the estimate $\hat{p}_2 = y_2/N_2$.

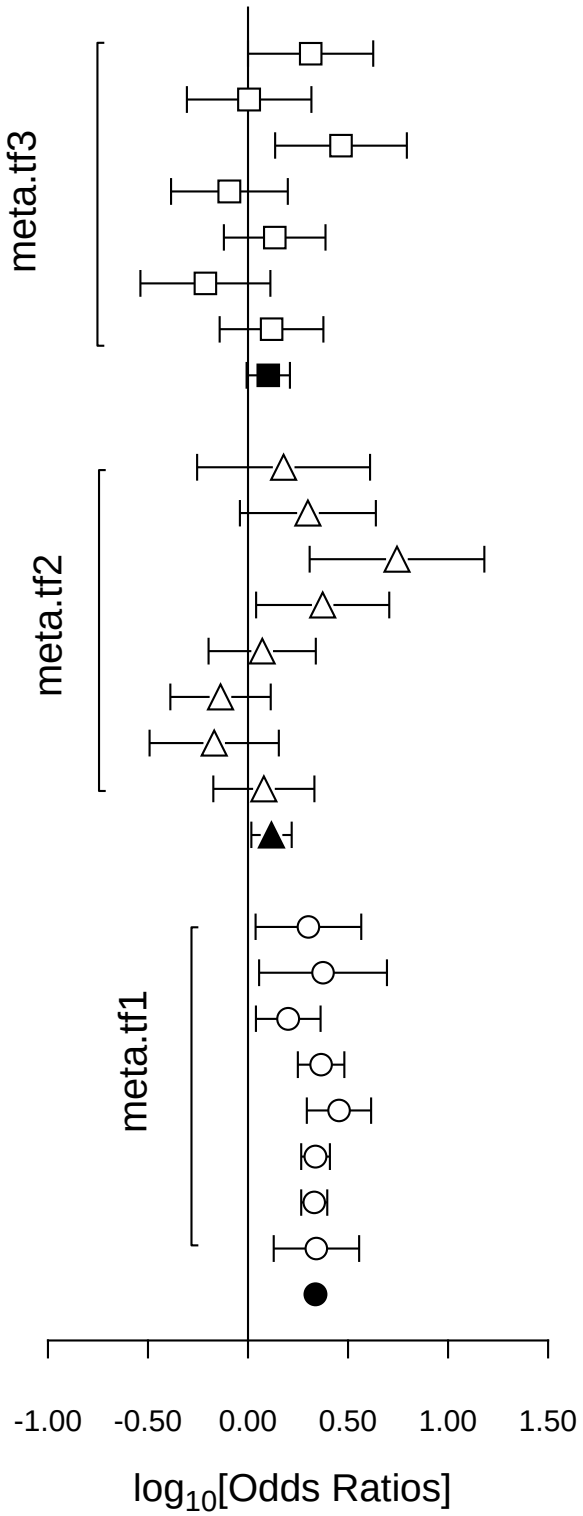
So the log odds ratios being compared between groups would be

$$\text{Log Odds Ratio in each group} = \log_{10}\left(\frac{\hat{p}_1/(1-\hat{p}_1)}{\hat{p}_2/(1-\hat{p}_2)}\right)$$

and this illustrates an important point.

Because the **comprehensive** mode imposes the coordinate ranges that were current when the configuration file was created for test data file `meta.tf2`, it re-imposes the choice that was made to suppress the original dashed line added to indicate equal odds within groups, i.e. log odds ratios of 0, move the y axis to a value of 0, and in addition to increase the range covered by the y axis to separate the pooled sample estimate from the x axis.

Another feature of plotting meta analysis data is the intention to display groups of such 2 by 2 contingency tables stacked vertically so as to emphasize differences an similarities between competing studies, for instance in so-called evidence based medicine, i.e. clinical trials. An example will now be presented to illustrate how to do this using SimFIT.



It is often useful to plot Log-Odds-Ratios, so the creation of the adjacent figure will be outlined.

(1) **The data**
Test files `meta.tf1`, `meta.tf2`, and `meta.tf3` were analyzed in sequence using the SIMFIT Meta Analysis procedure. Note that, in these files, column 3 contains spacing coordinates so that data will be plotted consecutively.

(2) **The ASCII coordinate files**
During Meta Analysis, $100(1-\alpha)\%$ confidence limits on the Log-Odds-Ratio resulting from a 2 by 2 contingency tables with cell frequencies n_{ij} can be constructed from the approximation $\hat{e}$ where

$$\hat{e} = Z_{\alpha/2}\sqrt{\frac{1}{n_{11}}+\frac{1}{n_{12}}+\frac{1}{n_{21}}+\frac{1}{n_{22}}}.$$

When Log-Odds-Ratios with error bars are displayed, the overall values (shown as filled symbols) with error bars are also plotted with a x coordinate one less than smallest x value on the input file. For this figure, error bar coordinates were transferred into the project archive using the [Advanced] option to save ASCII coordinate files.

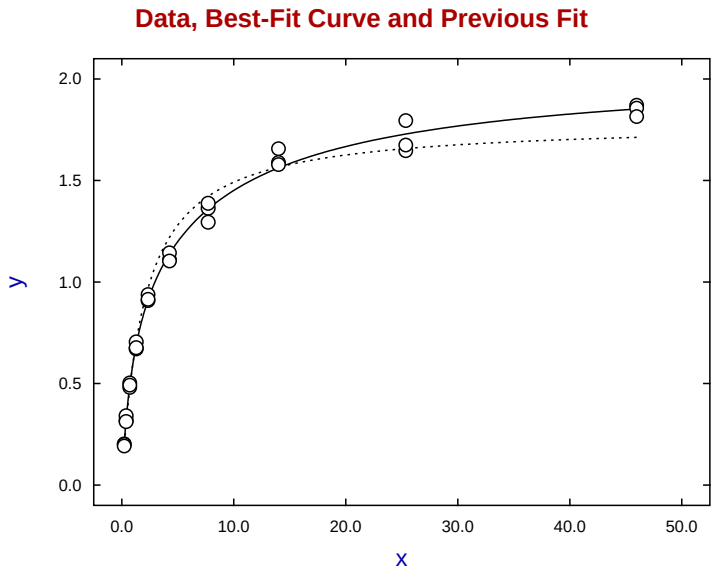
(3) **Creating the composite plot**
Program **simplot** was opened and the six error bar coordinate files were retrieved from the project archive. Experienced users would do this more easily using a library file of course. Reverse y-semilog transformation was selected, symbols were chosen, axes, title, and legends were edited, then half bracket hooks identifying the data were added as arrows and extra text.

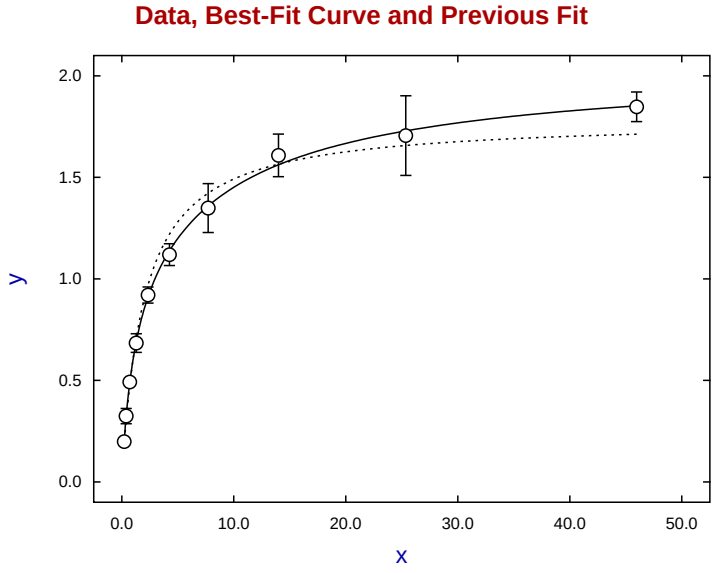
(4) **Creating the PostScript file**
Vertical format was chosen then, using the option to stretch PostScript files, the y coordinate was stretched by a factor of two.

(5) **Editing the PostScript file**
To create the final PostScript file for LaTeX a tighter bounding box was calculated using **gsview** then, using **notepad**, clipping coordinates at the top of the file were set equal to the BoundingBox coordinates, to suppress excess white space. This can also be done using the [Style] option to omit painting a white background, so that PostScript files are created with transparent backgrounds, i.e. no white space, and clipping is irrelevant.

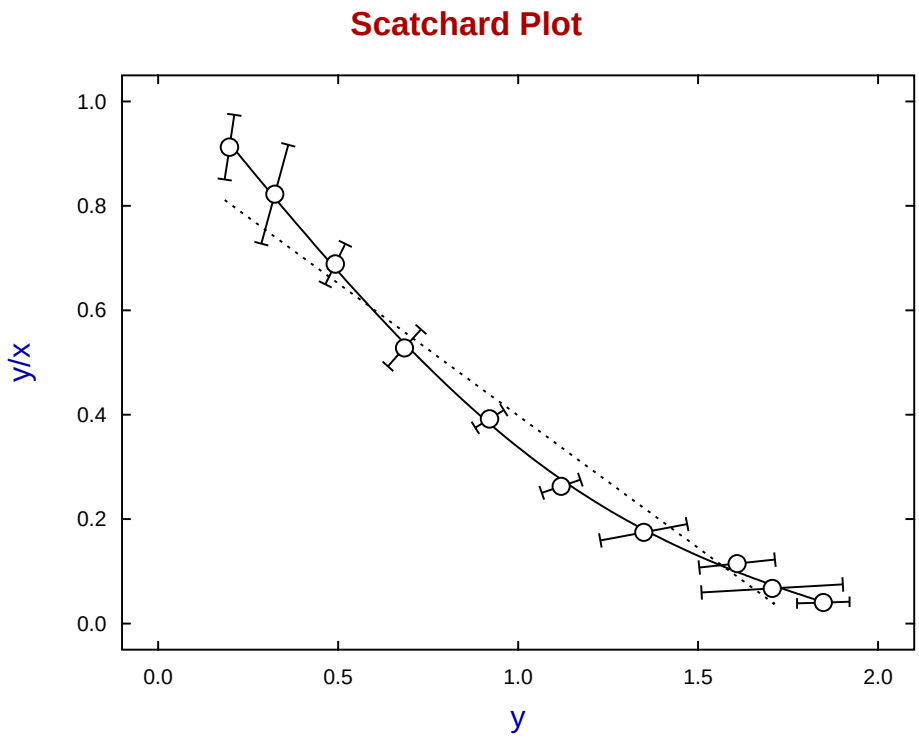
13.3.4 The Scatchard plot

The Scatchard plot was used historically to estimate binding constants by extrapolation, but nowadays parameter estimation and model discrimination would be performed using nonlinear regression and statistical analysis. However, the plot is still sometimes used to illustrate deviations from Michaelis-Menten kinetics or one-site binding (Childs,R.E and Bardsley,W.G. (1976) J. Theor. Biol. **63**, 1 – 18). This can be demonstrated by opening program **mmfit** and fitting first one then a sum of two Michaelis-Menten functions. Program **mmfit** provides an option to visualize a Scatchard plot but, to use additional features, the data would be transferred directly to advanced editing leading to the following plots. The first plot shows all the data, while the second results from interactive calculation of error bars for 95% confidence limits.

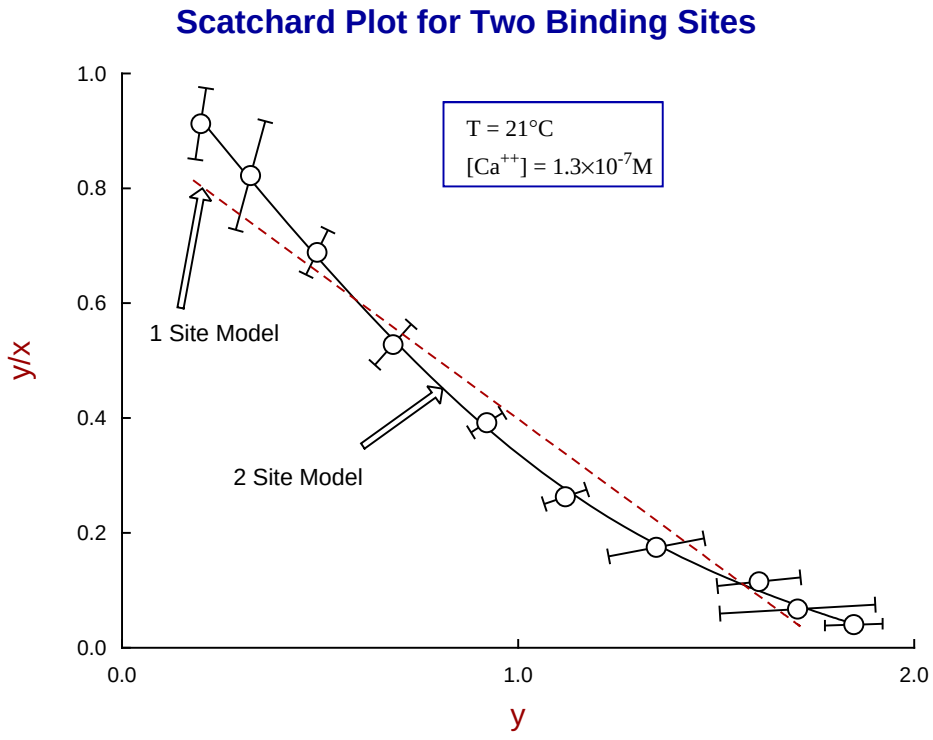




Selecting the Scatchard transformation then leads to the next graph where the best-fit curve is a conic section.

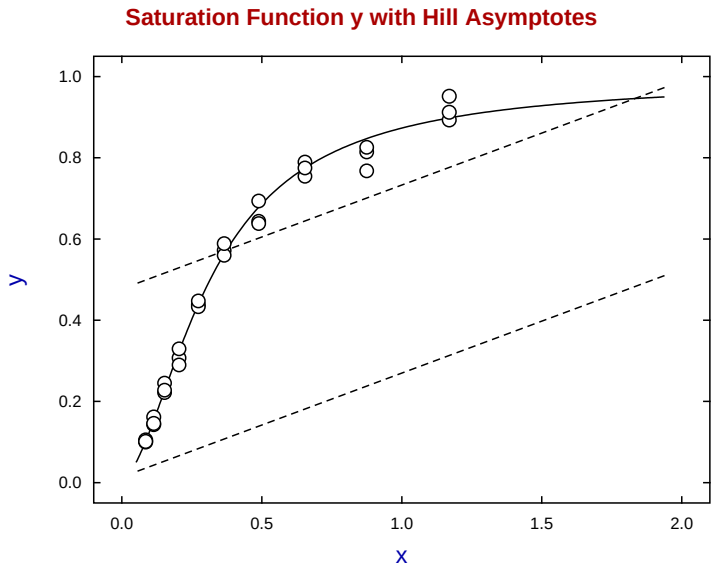


For further illustration, the configuration file `w_simfig1.cfg` was installed, followed by minor editing to create the next plot.

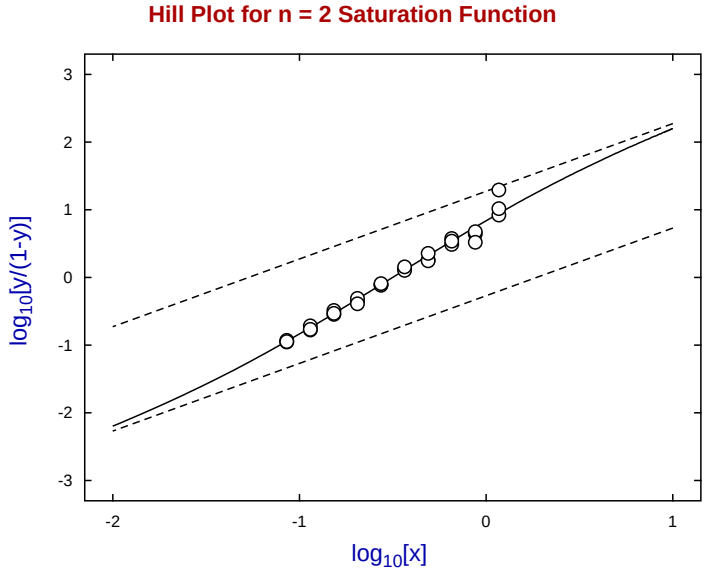


13.3.5 The Hill plot

The Hill plot is used to display cooperative effects in ligand binding to a protein or receptor with at least two linked sites. Sites with no linkage will show one-site binding isotherms and so, to explain how to create Hill plots, use program **sffit** to fit a two site model to the test file `sffit.tf4`. After fitting, a cooperativity analysis interface is presented and, choosing the Hill plot, generates the next plot with data, best-fit curve, and extreme points needed to plot the asymptotes in Hill space.



Choosing a Hill plot with asymptote $A = 1$ the next plot will be displayed, and this is followed by an introduction to the principles of cooperativity analysis needed to interpret Hill plots..



Theory

Ligand binding theory will be presented under the following headings.

1. Historical introduction
2. Binding polynomials
3. Definition of cooperativity
4. Factorability of the binding polynomial
5. Statistical interpretation of saturation functions
6. Cooperativity analysis

Historical Introduction

In 1910 Hill [1] proposed that the sigmoid binding curve for oxygen binding to haemoglobin could be analyzed in terms of the binding of n ligands in one step with no appreciable intermediates, i.e. the mass action description

$$Hb + nX \rightleftharpoons HbX_n.$$

This leads to the Hill equation describing the fractional saturation y as a function of concentration x, and the Hill plot of $\log[y/(1-y)]$ as a function of $\log x$ as follows

$$y = \frac{Kx^n}{1 + Kx^n}$$

$$\log\left(\frac{y}{1-y}\right) = n \log x + \log K.$$

It is now realized that the Hill equation is simply an empirical equation that is at best a poor approximation to any real binding situation since:

1. it is only an appropriate representation for a one-site binding process, i.e. for $n = 1$;
2. when $n < 1$ it has an infinite slope at the origin and cannot model any realistic binding situation;
3. when $n > 1$ it has zero slope at the origin and cannot model any realistic binding situation;
4. when n is not a positive integer it is pure nonsense; and
5. using it to discuss the effect of cooperativity on graphical features such as sigmoidicity in the $y(x)$ curve, or convexity in Lineweaver-Burke or Scatchard space, has resulted in considerable confusion.

Of course, before the days of computers and nonlinear regression, fitting a straight line to a Hill plot to get a non-integer value for the estimated slope was all that could be done, and this non-integer value was correctly taken to mean that this was a result of the model being incorrect.

Nowadays no one would dream of discussing cooperative binding in terms of the Hill equation or fitting a straight line to a Hill plot but, by a serendipitous coincidence, it turns out that the variable slope of the curve obtained by transforming a saturation curve into Hill space still provides an unambiguous definition of the sign and magnitude of cooperativity that has got nothing at all to do with the Hill equation. That is because, to use receptor terminology,

$$\frac{y}{1-y} = \frac{[\text{Bound}]}{[\text{Free}]}.$$

Binding polynomials and their Hessians

In 1925 Adair [2] improved the description of binding isotherms by defining binding constants for the individual binding events, and later it came to be appreciated that these have to be normalized by statistical factors in order to discuss the affinity of receptor for ligand in adjacent binding events. In 1967 Wyman [3] rationalized the situation by pointing out that, for a non-aggregating macromolecule with n binding sites and only one ligand x varied, there would be binding polynomial which would act like a partition function in that successive terms of degree i in the polynomial are proportional to the amount of macromolecule with i ligands attached.

So now the binding of ligands to receptors can be defined for all possible cooperative binding schemes in terms of a binding polynomial $p(x)$ in the free ligand activity x, as follows

$$\begin{aligned} p(x) &= 1 + K_1x + K_2x^2 + \cdots + K_nx^n \\ &= 1 + A_1x + A_1A_2x^2 + \cdots + \prod_{i=1}^{n} A_ix^n \\ &= 1 + \binom{n}{1}B_1x + \binom{n}{2}B_1B_2x^2 + \cdots + \binom{n}{n}\prod_{i=1}^{n} B_ix^n, \end{aligned}$$

where the only difference between these alternative expressions concerns the meaning and interpretation of the binding constants. The fractional saturation is just the scaled derivative of the log of the polynomial with respect to $\log(x)$, and an important auxiliary function is $h(x)$, the Hessian of the binding polynomial defined as follows

$$\begin{aligned} y(x) &= \left(\frac{1}{n}\right)\frac{d\log p(x)}{d\log x} \\ &= \left(\frac{1}{n}\right)\frac{xp'(x)}{p(x)} \\ h(x) &= npp'' - (n-1)p'^2. \end{aligned}$$

Definition of cooperativity

Given a binding polynomial of degree n there are $n-1$ cooperativity coefficients c_i defined as

$$c_i = B_{i+1} - B_i \text{ for } i = 1, 2, \ldots, n-1,$$

or alternatively as $\log(B_{i+1}/B_i)$, and the interpretation of these is perfectly clear: in a situation where $c_i > 0$ the macromolecule has greater affinity for binding the $i+1$th ligand after the ith ligand has been bound and it is perfectly reasonable to describe this as mechanistic positive cooperativity. Hence every binding situation for n ligands can be summarized by a succession of $n-1$ signs and it might be thought that during the actual saturation of macromolecule with ligand there would be a succession of phases with possibly differing cooperativity. For instance, the sequence $+-+$ might be supposed to give a saturation curve with positive, then negative, then positive cooperativity. Unfortunately the cooperativity coefficients cannot be interpreted in this way and they are not a unique indicator of the sign and magnitude of the type of cooperativity exhibited during the saturation process. The reason for this is simply that binding does not occur in a succession of isolated steps and at every stage for $0 < x < \infty$ every species that is possible is present, that is no ligands bound, one ligand bound, two ligands bound, etc. up to n ligands bound.

At every point in the range $0 < x < \infty$ there is a one site binding curve y_{app} with a uniquely defined apparent binding constant K_{app} according to the scheme

$$[\text{Free sites}] + X \rightleftharpoons [\text{Occupied sites}]$$

that is

$$y_{app}(x) = \frac{K_{app}x}{1 + K_{app}x}.$$

Surely all would agree that the sign and magnitude of cooperativity at that point in the saturation curve would depend on whether K_{app} is increasing or decreasing as a function of x. It turns out that

$$\begin{aligned} K_{app} &= \frac{p'(x)}{np(x) - xp'(x)} \text{ and} \\ \frac{dK_{app}}{dx} &= \frac{h(x)}{(np(x) - xp'(x))^2} \end{aligned}$$

so that increasing affinity (i.e. positive cooperativity) requires $h(x) > 0$, decreasing affinity (i.e. negative cooperativity) requires $h(x) < 0$ while at a point where $h(x) = 0$ cooperativity changes sign. Bardsley and Wyman [4] emphasized that the magnitude of the Hill slope with respect to 1 is the unambiguous indicator of cooperativity which also depends on the sign of the Hessian as follows

$$\frac{d\log[y/(1-y)]}{d\log x} = 1 + \frac{xh(x)}{p'(x)(np(x) - xp'(x))}.$$

and Wood and Bardsley [5] proved that the Hessian can have at most $n - 2$ positive zeros.

Zeros of the binding polynomial

If the n zeros of the binding polynomial are α_i then the fractional saturation y can be expressed as

$$y = \left(\frac{x}{n}\right)\sum_{i=1}^{n}\frac{1}{x-\alpha_i},$$

but further discussion depends on the nature of the zeros.

First observe that, for a set of m groups of receptors, each with n_i independent binding sites and binding constant k_i, then the zeros are all real and

$$p(x) = \prod_{i=1}^{m}(1+k_i x)^{n_i},$$

$$\text{and } y = \frac{1}{\sum_{i=1}^{m} n_i}\sum_{i=1}^{m}\frac{n_i k_i x}{1+k_i x},$$

so y is just the sum of simple binding curves, giving concave down double reciprocal plots, etc.

Actually Bardsley et al [6] and [7] proved that, if a binding polynomial factorizes into m polynomials p_i with positive coefficients according to

$$p(x) = p_1(x)p_2(x)\ldots p_m(x)$$

then the Hill plot slope cannot exceed that of the Hill plot slope for any of the individual factors. As a binding polynomial can always be factorized into a product of linear factors with real negative zeros and complex conjugate pairs forming quadratic factors it might be supposed that the Hill slope can never exceed two. However, if a binding polynomial of degree > 2 has complex conjugate zeros, the Hill slope may exceed two and there may be evidence of strong positive cooperativity. That is why Hill plot slopes up to a maximum of the degree of the binding polynomial can be achieved if there are quadratic factors with negative coefficients, corresponding to a group of at least three linked binding sites.

For instance, the binding polynomial for a four site Monod-Wyman-Changeux model is

$$p(\alpha) = \frac{1}{1+L}\left((1+\alpha)^4 + L(1+c\alpha)^n\right)$$

and this can factorize into the form

$$q(x) = (1 + a_1 x + b_1 x^2)(1 - a_2 x + b_2 x^2)$$

with $a_1 > 0, a_2 > 0, b_1 > 0, b_2 > 0$ under certain constraints so that the meaningless quadratic factor with a negative term allows Hill slopes greater than two.

Edelstein and Bardsley [8] subsequently explored the relationship between the Hill slope at half-saturation and the Hessian of the binding polynomial.

Statistical interpretation of saturation functions

The species fractional populations s_i which are defined for $i = 0, 1, \ldots, n$ as

$$s_i = \frac{K_i x^i}{K_0 + K_1 x + K_2 x^2 + \cdots + K_n x^n}$$

with $K_0 = 1$, are interpreted as the proportions of the receptors in the various states of ligation as a function of ligand activity. The species fractions defined as $y_i = is_i/n$ for $i = 1, 2, \ldots, n$ are the contributions of the species to the overall saturation. Note that

$$\sum_{i=0}^{n} s_i = 1, \text{, while}$$
$$\sum_{i=1}^{n} y_i = (1/n)d\log p/d\log x.$$

Such expressions are very useful when analyzing cooperative ligand binding data and they can be generated from the best fit binding polynomial after fitting binding curves with program **sffit**, or by interactive input of binding constants into program **simstat**. At the same time other important analytical results like factors of the Hessian and minimax Hill slope are also calculated.

The species fractional populations can be also used in a probability model to interpret ligand binding in several interesting ways. For this purpose, consider a random variable U representing the probability of a receptor existing in a state with i ligands bound. Then the the probability mass function, expected values and variance are

$$\begin{aligned} P(U = i) &= s_i \ (i = 0, 1, 2, \ldots, n), \\ E(U) &= \sum_{i=0}^{n} is_i, \\ E(U^2) &= \sum_{i=0}^{n} i^2 s_i, \\ V(U) &= E(U^2) - [E(U)]^2 \\ &= x\left(\frac{p'(x) + xp''(x)}{p(x)}\right) - \left(\frac{xp'(x)}{p(x)}\right)^2 \\ &= n\frac{dy}{d\log x}, \end{aligned}$$

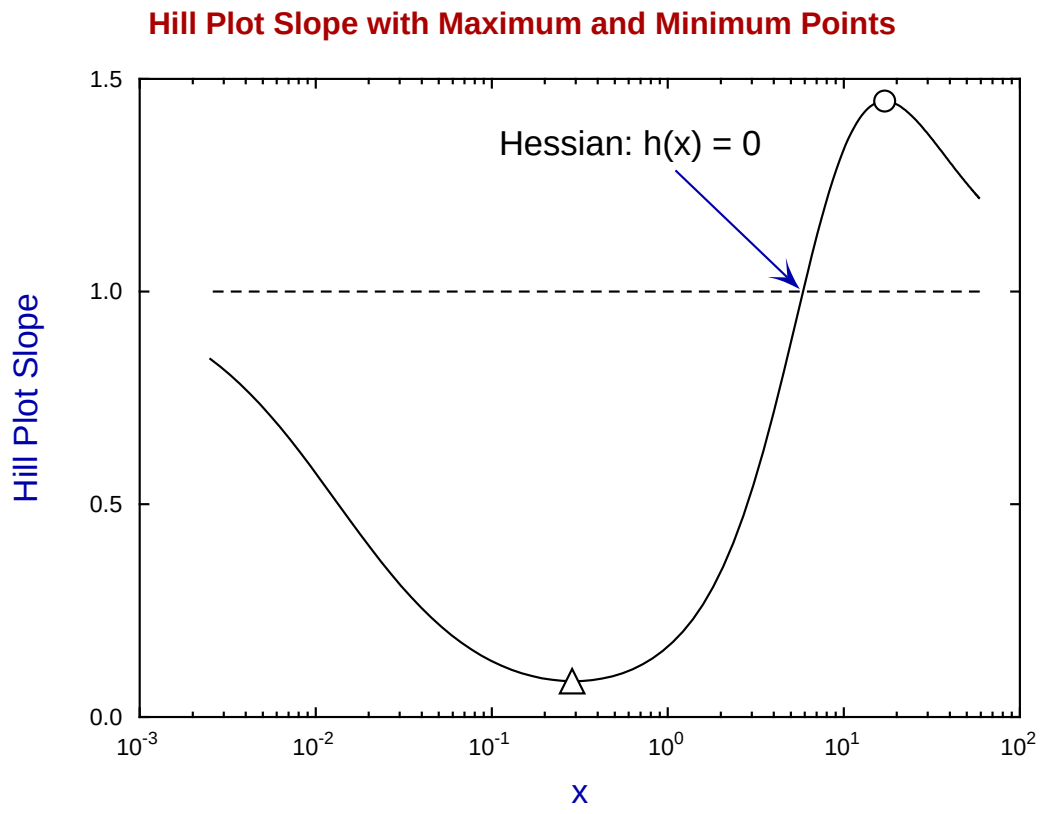
as fractional saturation y is $E(U)/n$. In other words, the slope of a semi-log plot of fractional saturation data indicates the variance of the number of occupied sites, namely; all unoccupied when $x = 0$, distribution with variance increasing as a function of x up to the maximum semi-log plot slope, then finally approaching all sites occupied as x tends to infinity. You can input binding constants into the statistical calculations procedure to see how they are mapped into all spaces, cooperativity coefficients are calculated, zeros of the binding polynomial and Hessian are estimated, Hill slope is reported, and species fractions and binding isotherms are displayed, as is done automatically after every $n > 1$ fit by program **sffit**.

Cooperativity analysis

After fitting a model, program **sffit** outputs the binding constant estimates in all the conventions and, when $n > 2$ it also outputs the zeros of the best fit binding polynomial and those of the Hessian of the binding polynomial $h(x)$. The positive zeros of $h(x)$ indicate points where the theoretical one-site binding curve coinciding with the actual saturation curve at that x value has the same slope as the higher order saturation curve, which are therefore points of cooperativity change. The SimFIT cooperativity procedure allows users to input binding constant estimates retrospectively to calculate zeros of the binding polynomial and Hessian, and also to plot species population fractions.

For instance, for 4 sites with $K_1 = 100, K_2 = 10, K_3 = 1,$ and $K_4 = 0.1$, the Hessian has a positive zero at $x = 5.86139$, the minimum Hill slope in the range plotted is 0.0842, at $x = 0.28607$, the maximum is 1.44479, at $x = 17.059$, and the slope at half saturation is 1.0847, at $x = 6.5808$.

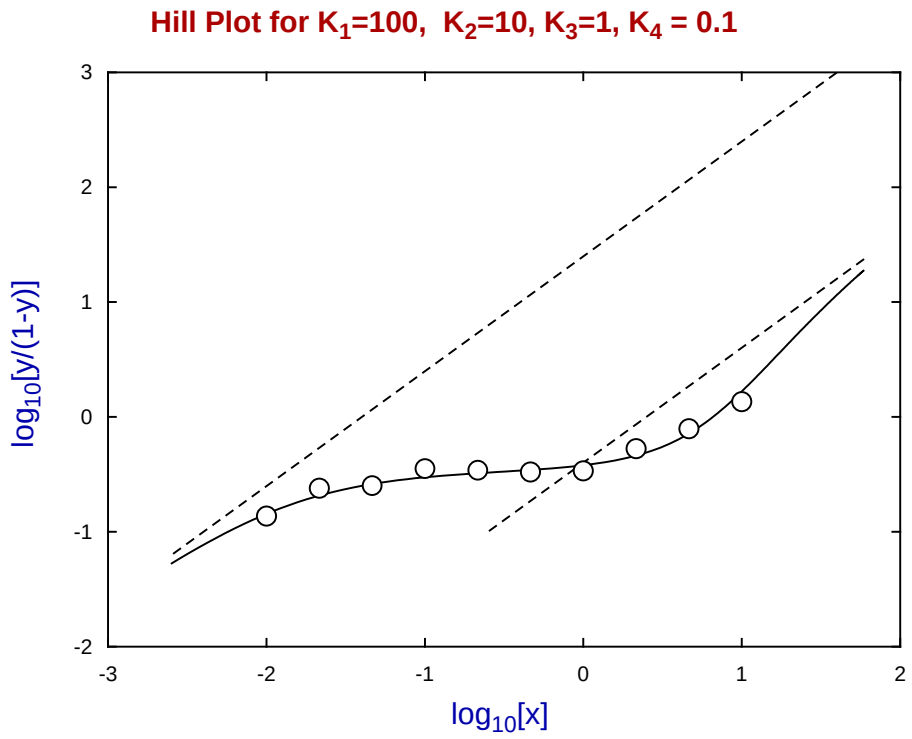
The next graph shows how the Hill plot slope varies with the maximum and minimum slopes indicated along with the point where the positive zero of the Hessian occurs.



The following graph shows the sort of complicated Hill plots that can be obtained when there are more than two cooperatively linked sites. The asymptotes are for the equation

$$y = \frac{kx}{1 + kx}$$

with $k = K_1/n$ as $x \to 0$ and $k = nK_n/K_{n-1}$ as $x \to \infty$.



References

[1] The possible effects of the aggregation of the molecules of haemoglobin on its dissociation curves.
Hill,A.V. (1910), *J.Physiol.* **40**, 4-7.

[2] The hemoglobin system. VI. The oxygen dissociation curve of hemoglobin.
Adair,G.S. (1925) *J.Biol. Chem.* **63**, 529-545.

[3] Allosteric Linkage.
Wyman,J. (1967), *J.Amer.Chem.Soc. 89.9* 2202-2218.

[4] Concerning the thermodynamic definition and graphical manifestations of positive and negative co-operativity.
Bardsley,W.G. & Wyman,J. (1978) *J. theor. Biol.* **72**, 373-376

[5] Critical points and sigmoidicity of positive rational functions.
Wood,R.M.W. & Bardsley,W.G. (1985) *Amer. Math. Month.* **92(1)**, 37-48

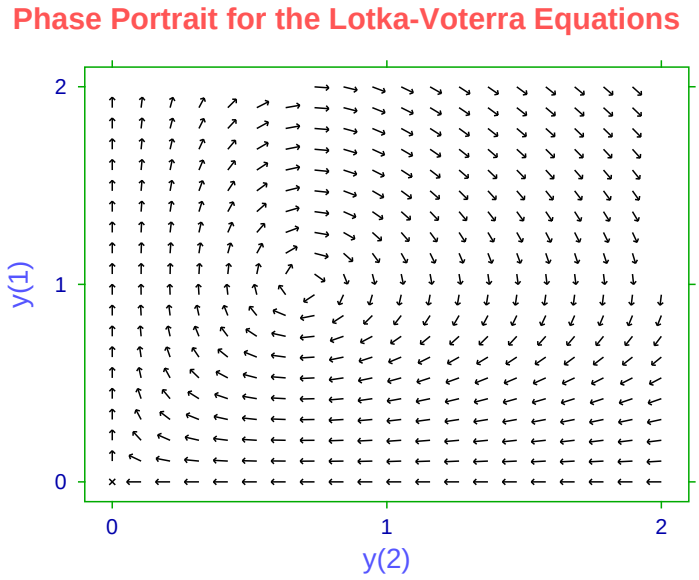
[6] Relationships between the magnitude of Hill plot slopes, apparent binding constants and factorability of binding polynomials and their Hessians.
Bardsley,W.G., Woolfson,R. & Mazat,J,-P. (1980) *J. theor. Biol.* **85**, 247-284

[7] Factorability of the Hessian of the binding polynomial. The central issue concerning statistical ratios between binding constants, Hill plot slope and positive and negative co-operativity.
Bardsley,W.G. & Waight,R.D. (1978) *J. theor. Biol.* **72**, 321-372

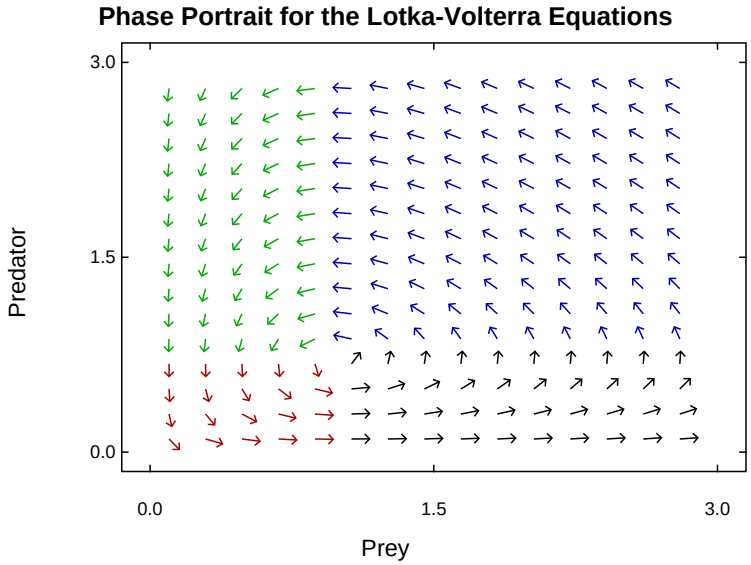
[8] Contributions of individual molecular species to the Hill coefficient for ligand binding by an oligomeric protein.
Edelstein,S.J & Bardsley,W.G. *J. Mol. Biol* (1997) **267**, 10–16

13.3.6 Vector field diagrams

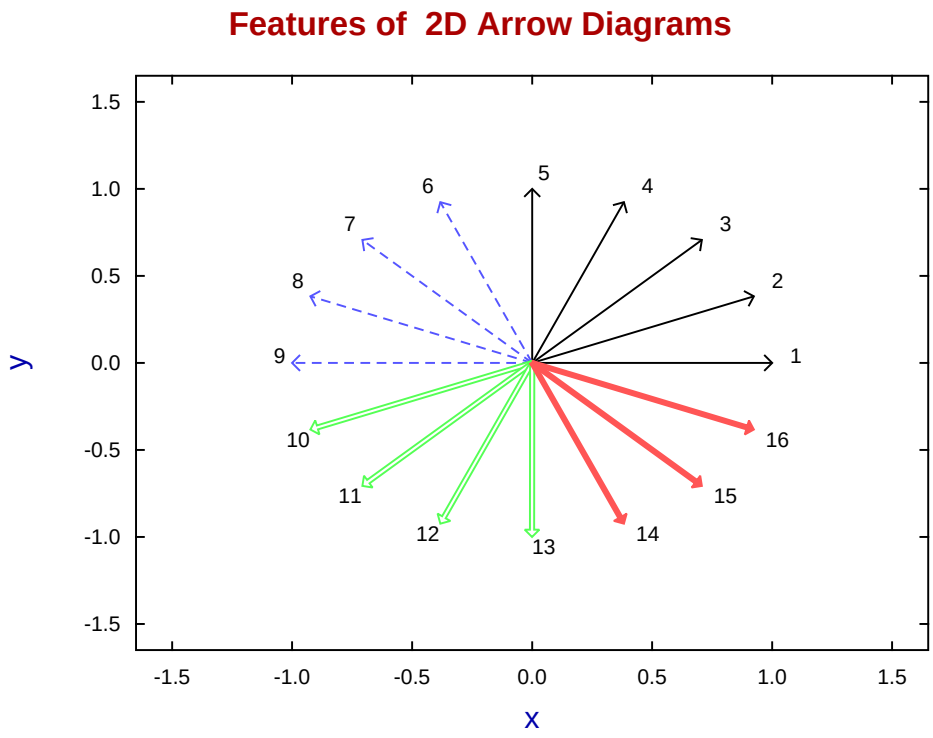
Vector field diagrams are used to indicate the strength and direction of fields as a function of position, where the definition of a field covers many situations. Consider the phase portrait technique for exploring the Lotka-Volterra predator-prey equations using program **deqsol** leading to the following diagram of vector directions.

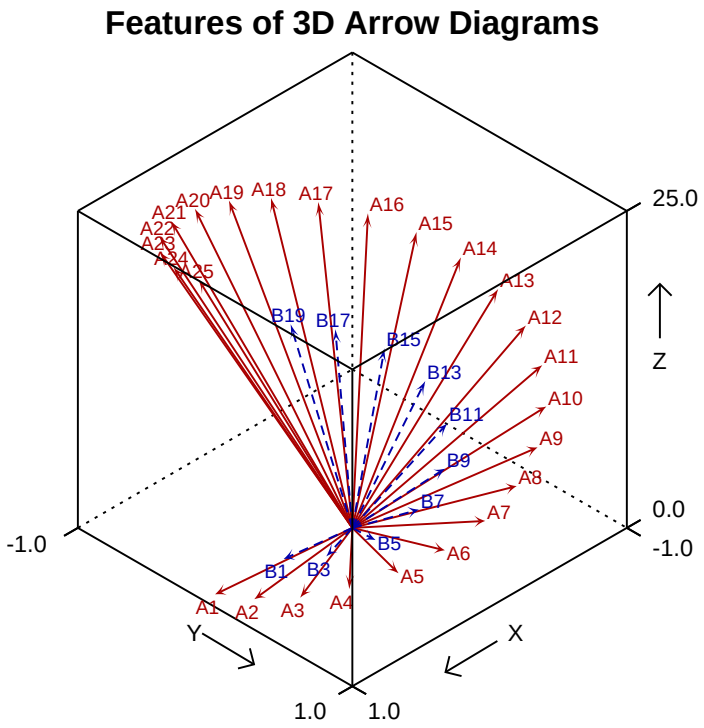


To improve the interpretation of such portraits, program **deqsol** also allows the direction of the gradients to be color-coded as in the next plot, or even to change the size of arrows to be proportional to the magnitude, although this feature has to be used sparingly.



SIMFIT does provide numerous options for creating 2D and 3D arrow diagrams as shown below, but such graphs are most easily produced using the SIMDEM package linked to the SIMFIT DLLs, which provides many extra procedures but does require programming ability





13.3.7 Plotting surfaces

Plotting a three-dimensional surface requires four steps as follows.

1. Define a mathematical model.
2. Fix the values of parameters in the model.
3. Choose the ranges of independent variables.
4. Decide on the number of divisions required.

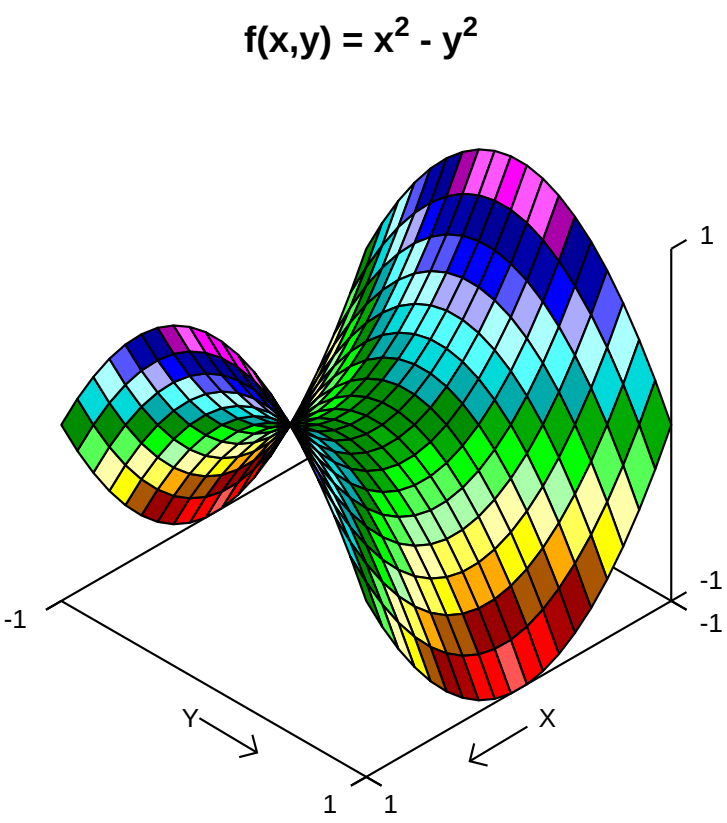
For example, open program **makdat** and choose a function of two variables, then select a polynomial which will have the following definition

$$f(x, y) = p_1 x + p_2 y + p_3 x^2 + p_4 xy + p_5 y^2.$$

In order to plot the function $z = x^2 - y^2$ you will have to fix the parameters as follows

$$p_1 = 0, p_2 = 0, p_3 = 1, p_4 = 0, p(5) = -1$$

then choose to plot a sensible range, e.g. $-1 \leq x \leq 1, -1 \leq y \leq 1$, say 20 divisions which will often be sufficient for a surface, to obtain the following plot (after some minor editing).



13.3.8 Plotting contours

Plotting a two-dimensional contour diagram of a three-dimensional surface requires four steps as follows.

1. Define a mathematical model.
2. Fix the values of parameters in the model.
3. Choose the ranges of independent variables.
4. Decide on the number of divisions required.

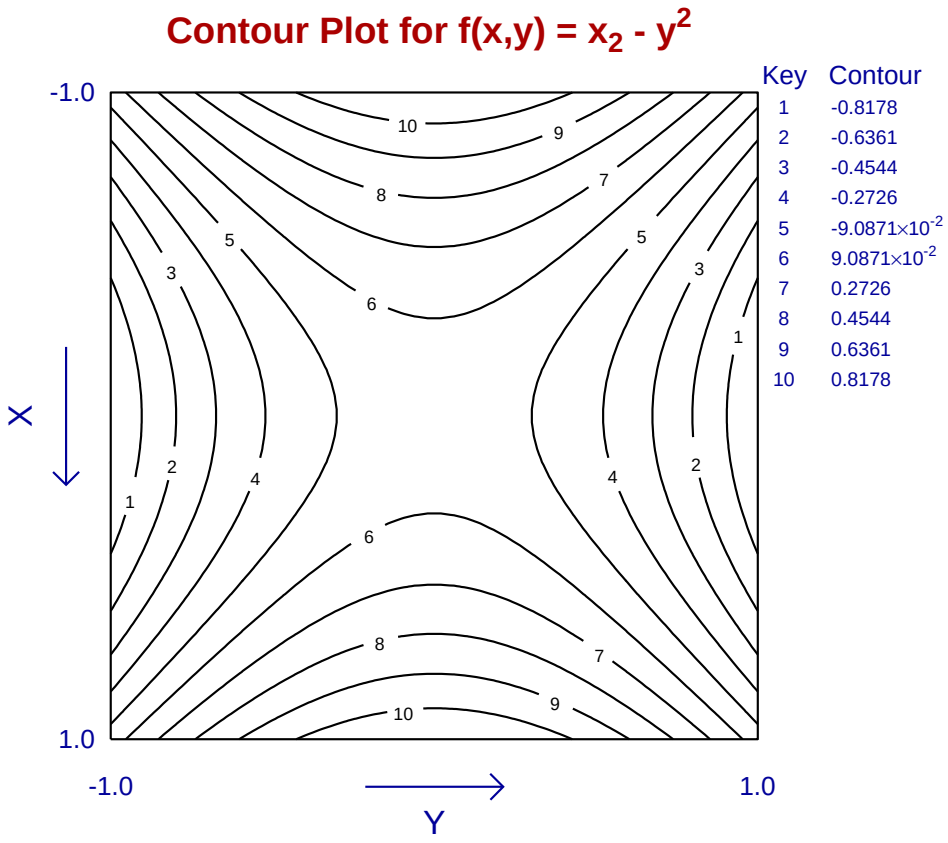
For example, open program **makdat** and choose a function of two variables, then select a polynomial which will have the following definition

$$f(x, y) = p_1 x + p_2 y + p_3 x^2 + p_4 xy + p_5 y^2.$$

In order to plot the function $z = x^2 - y^2$ you will have to fix the parameters as follows

$$p_1 = 0, p_2 = 0, p_3 = 1, p_4 = 0, p(5) = -1$$

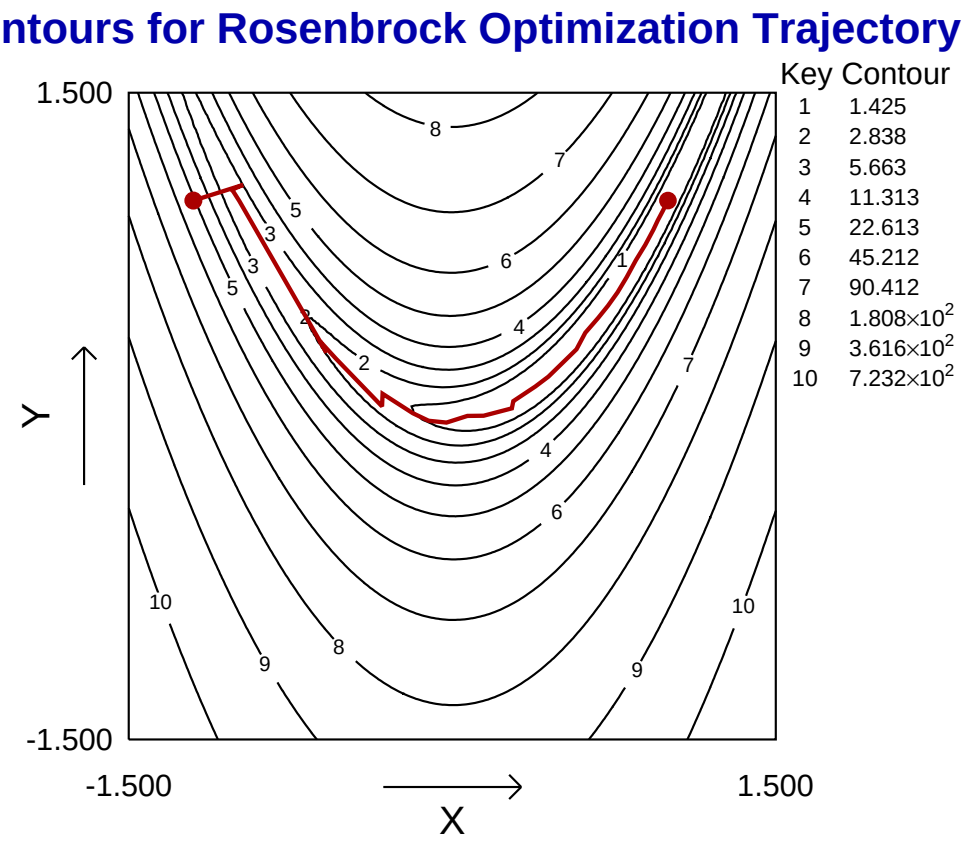
then choose to plot a sensible range, e.g. $-1 \leq x \leq 1, -1 \leq y \leq 1$, with say 50 divisions which will often be sufficient for a contour diagram, to obtain the following plot.



There are numerous points to consider when editing such a contour plot as follows.

1. While 20 divisions may be sufficient to plot a surface, a larger number of divisions, say at least 50, will be required for a contour plot.
2. Warnings will be output if insufficient divisions have been used leading to failure of the contouring algorithm.
3. There are numerous options to apply smoothing techniques with smaller numbers of divisions or complicated surfaces, but these cannot be expected to substitute for increasing the number of divisions.
4. The numbers indicating the contours and the table of contour values can be suppressed.
5. There are numerous options to choose the number and spacing of the contours. For instance, the default spacing of contour values in an arithmetic progression can be replaced by a geometric progression or even a user-supplied vector of proportions.
6. Color schemes can be used and it is possible to add additional features like arrows, extra text, graphical objects, or even additional curves to highlight trajectories.
7. In some cases it may be useful to plot the three-dimensional surface superimposed on a contour diagram but, when this is done, some of the editing functions for the contour diagram are not available.

The next plot illustrates the result of adding a trajectory to the contour diagram to illustrate the path taken by a constrained optimization algorithm.



13.3.9 Skyscraper and cylinder plots

Plotting a three-dimensional barchart as a skyscraper or cylinder diagram can be done using a mathematical model, or more conveniently by simply supplying a table of bar heights. Both techniques will be described.

Method 1: Using a mathematical model

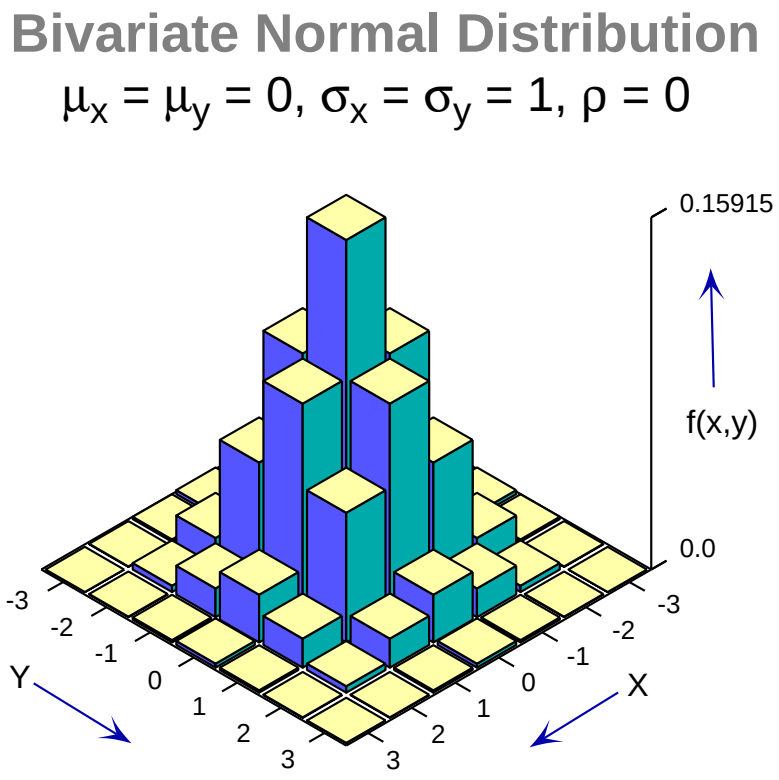
The following four steps are required.

1. Define a mathematical model.
2. Fix the values of parameters in the model.
3. Choose the ranges of independent variables.
4. Decide on the number of divisions required.

For example, open program **makdat** and choose a function of two variables, then select a bivariate normal distribution, $N_2(x, y)$, scaled and shifted which will have the following definition

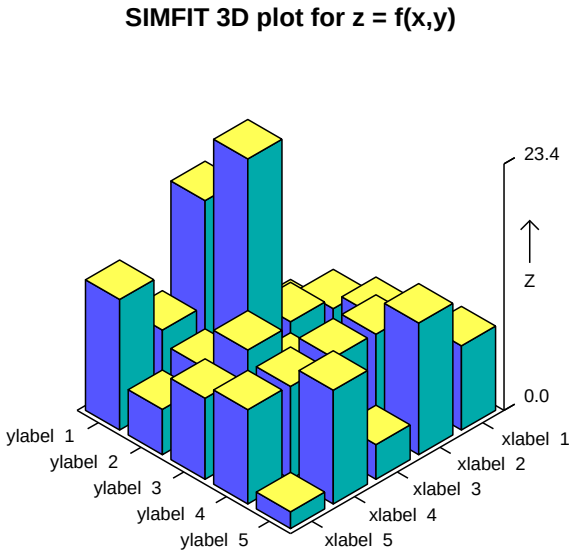
$$f(x, y) = p_6 N_2(x, y) + p_7, \text{ where } p_1 = \mu_x, p_2 = \sigma_x, p_3 = \mu_y, p_4 = \sigma_y, p_5 = \rho.$$

Choosing $p_1 = 0, p_2 = 1, p_3 = 0, p_4 = 1, p_5 = 0, p_6 = 1, p)7 = 0$ with a sensible range, e.g. $-3 \le x \le 3, -3 \le y \le 3$, and 7 divisions then creates the following diagram.

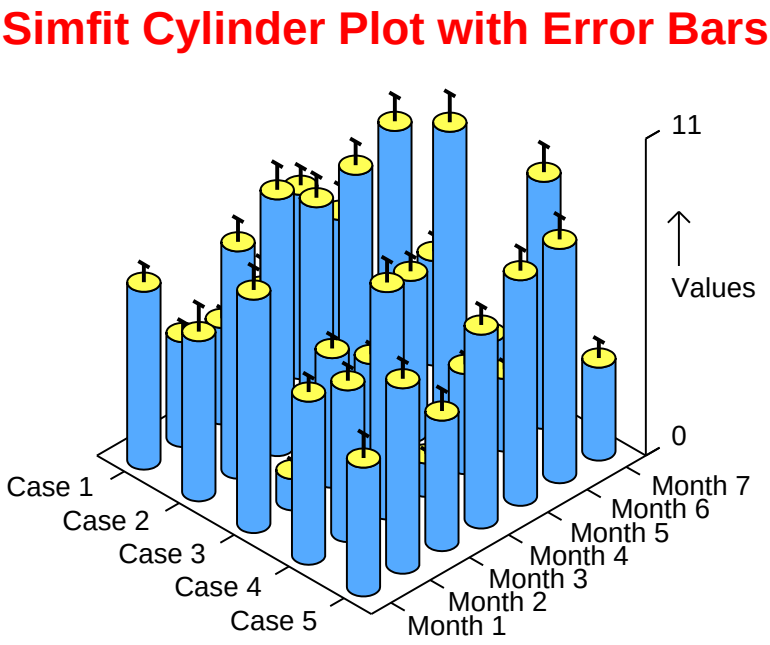


Method 2: Reading in a rectangular table of height values

Use the analysis of an arbitrary matrix option in program **simstat** or in program **simplot** and input the test file `matrix.tf1` which will generate the following default skyscraper diagram.



Alternatively the table of heights can be entered interactively from the console with program **simplot**. This technique is particularly valuable if it is wished to create a three dimensional barchart from a n by m matrix where the x and y axes are arbitrary groupings not coordinate values, and it is also possible to add a further file with a n by m matrix of errors to plot error bars as in the next figure.



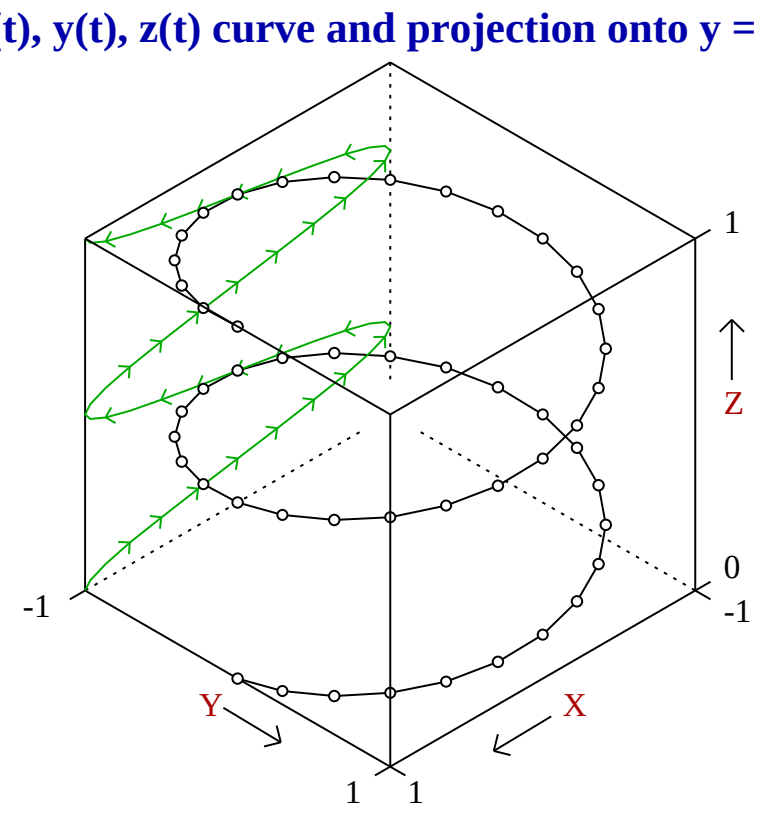
13.3.10 Plotting curves and data in three dimensions

Three dimensional space curves

Sets of x, y, z coordinates can be plotted in three dimensional space to represent either an arbitrary scatter of points, a surface, or a connected space curve. Arbitrary points are best plotted as symbols such as circles or triangles, surfaces are usually represented as a mesh of orthogonal space curves, while single space curves can be displayed as symbols or may be connected by lines. For instance, space curves of the form

$$x = x(t), y = y(t), z = z(t)$$

can be plotted by generating x, y, z data for constant increments of t and joining the points together to create a smooth curve as in the next figure.



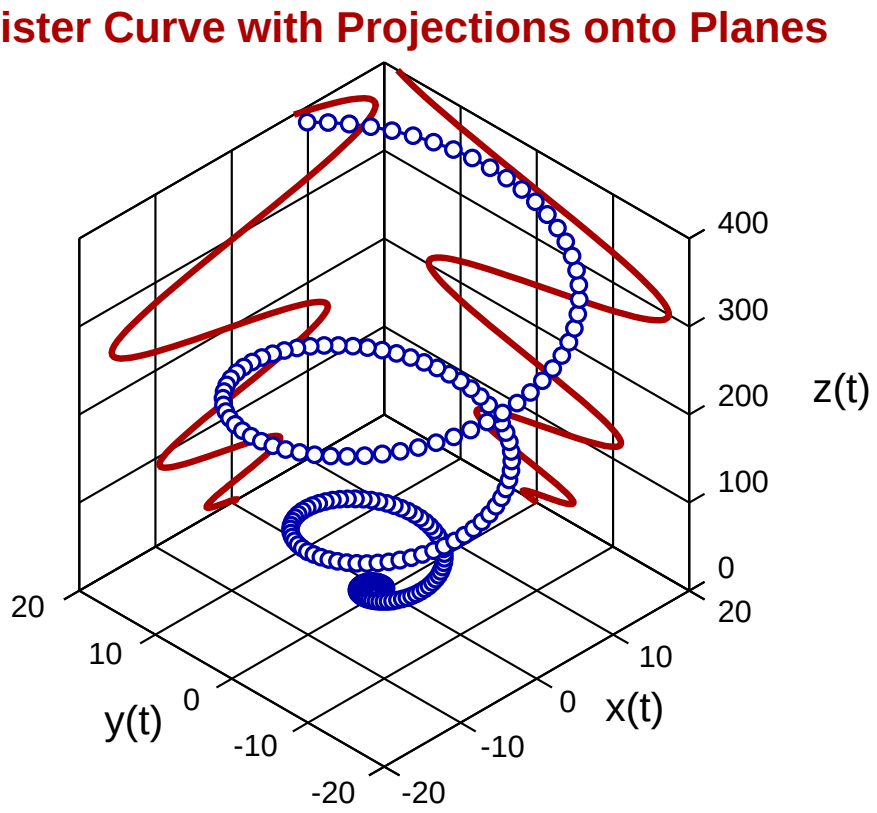
Such space curves can be generated quite easily by preparing data files with three columns of x, y, z data values, then displaying the data using the space curve option in **simplot**. However users can also generate space curves from $x(t), y(t), z(t)$ equations, using the option to plot parametric equations in **simplot** or **usermod**. The test file `helix.mod` shows you how to do this for a three dimensional helix. Note how the rear (x, y) axes have been subdued and truncated just short of the origin, to improve the three dimensional effect. Also, projections onto planes are generated by setting the chosen variable to a constant, or by writing model files to generate x, y, z data with chosen coordinates equal to the value for the plane.

Projecting space curves onto planes

Sometimes it is useful to project space curves onto planes for purposes of illustration. The next figure shows a simulation using **usermod** with the model file `twister.mod`. The parametric equations are

$$x = t\cos t, y = t\sin t, z = t^2$$

and projections are created by fixing one the variables to a constant value.



Note the following about the model file `twister.mod`.

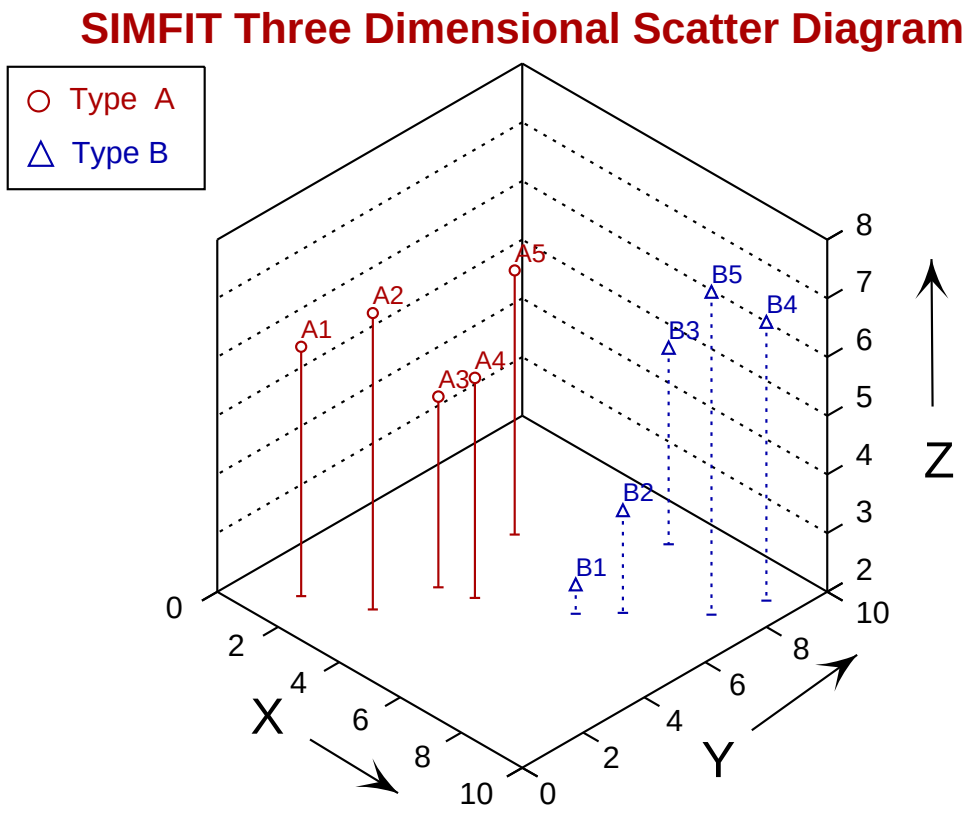
- There are 3 curves so there are 9 functions of 1 variable
- The value of x supplied is used as the parameter t
- Functions $f(1), f(4), f(7)$ are the $x(t)$ profiles
- Functions $f(2), f(5), f(8)$ are the $y(t)$ profiles
- Functions $f(3), f(6), f(9)$ are the $z(t)$ profiles

Also observe that the model parameters fix the values of the projection planes just outside the data range, at

$$p(1) = 20, p(2) = 20.$$

Three dimensional scatter diagrams

Often it is necessary to plot sets of x, y, z coordinates in three dimensional space where the coordinates are arbitrary and are not functions of a parameter t. This is the case when it is wished to illustrate scattering by using different symbols for subsets of data that form clusters according to some distance criteria. For this type of plotting, the sets of x, y, z triples, say principal components, are collected together as sets of three column matrices, preferably referenced by a library file, and a default graph is first created. The usual aim would be to create a graph looking something like this.



In this graph, the front axes have been removed for clarity, a subdued grid has been displayed on the vertical axes, but not on the base and perpendiculars have been dropped from the plotting symbols to the base of the plot, in order to assist in the identification of clusters.

Note that plotting symbols, minus signs in this case, have been added to the foot of the perpendiculars to assist in visualizing the clustering. Also, note that distinct data sets, requiring individual plotting symbols, are identified by a simple rule; data values in each data file are regarded as representing the same cluster, i.e. each cluster must be in a separate file.

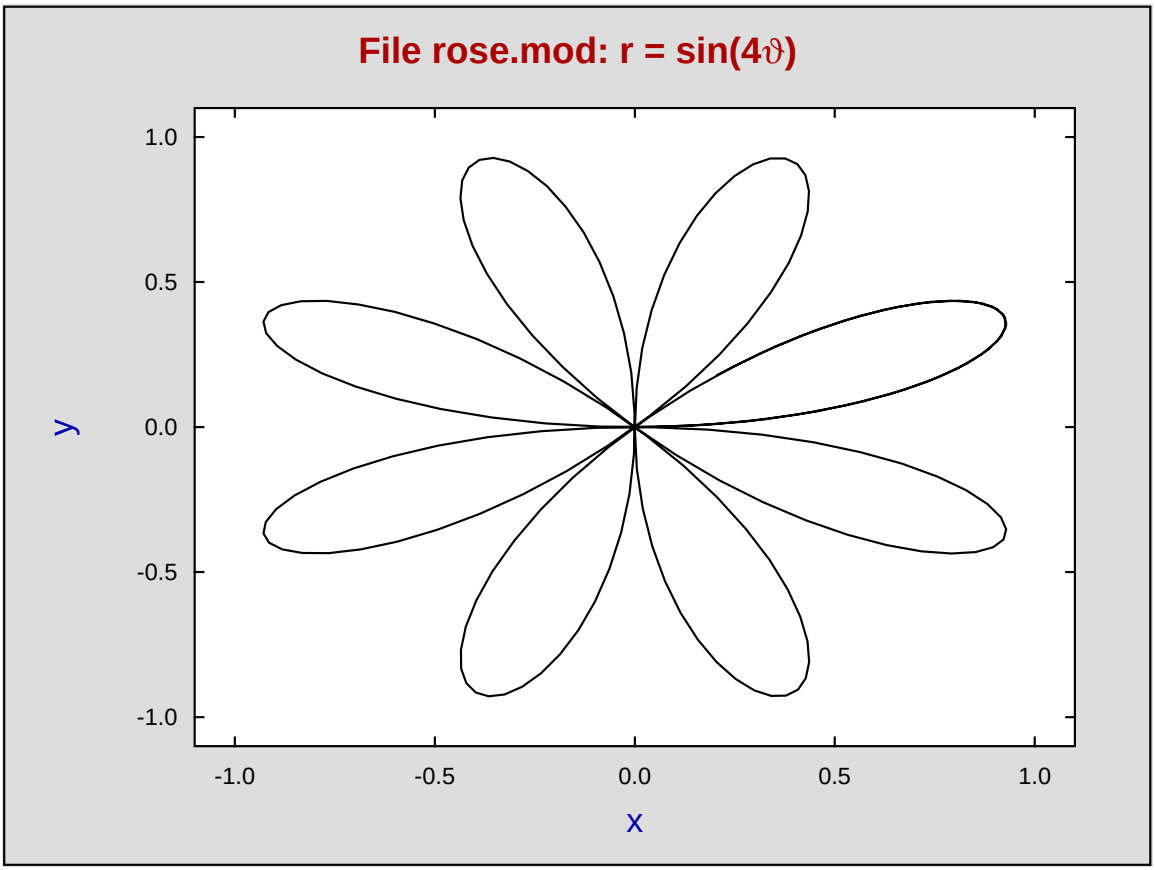
13.3.11 Parametric plots

Plotting models expressed in parametric form is often required,e.g. when the model cannot be expressed in standard form. SimFIT programs **simplot** and **usermod** can plot models in the following forms:
Plane curves as $r = r(\theta)$, Plane curves as $x = x(t), y = y(t)$, or Space curves as $x = x(t), y = y(t), z = z(t)$.

Plane curves as $r = r(\theta)$: Example 1

As an example consider the test file `rose_e.mod` for the eight leaved rose plotted below.

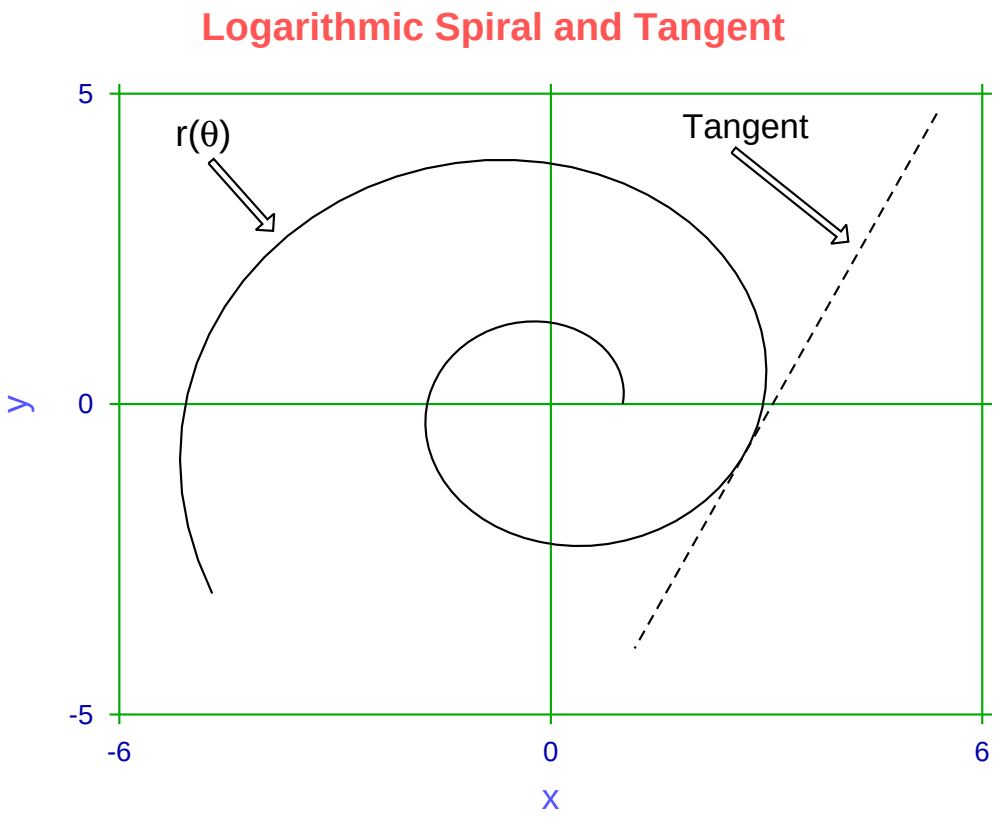
```
  %
Example: Eight leaved rose
r = A*sin(4*theta): where theta = x, r = f(1) and A = p(1)
 %
 1 equation
 1 variable
 1 parameter
 %
 begin{expression}
 f(1) = p(1)sin(4x)
 end{expression}
 %
```



Plane curves as $r = r(\theta)$: Example 2

The next plot illustrates the logarithmic spiral defined in SIMFIT model file `camalot_e.mod`.

```
%
Model: Logarithmic Spiral
   r = A*exp(theta*cot(p(1))): where theta = x, r = f(1)
   A = amplitude scaling factor
p(1) = angle in radians between radius vector and tangent
 %
 1 equation
 1 variable
 1 parameter
 %
 begin{expression}
 f(1) = exp{x/tan[p(1)]}
 end{expression}
 %
```



This profile is used in rock climbing camming devices such as Camalots and Friends to maintain a constant angle α between the radius vector for the spiral and the tangent to the curve, defined in `tangent.mod` as

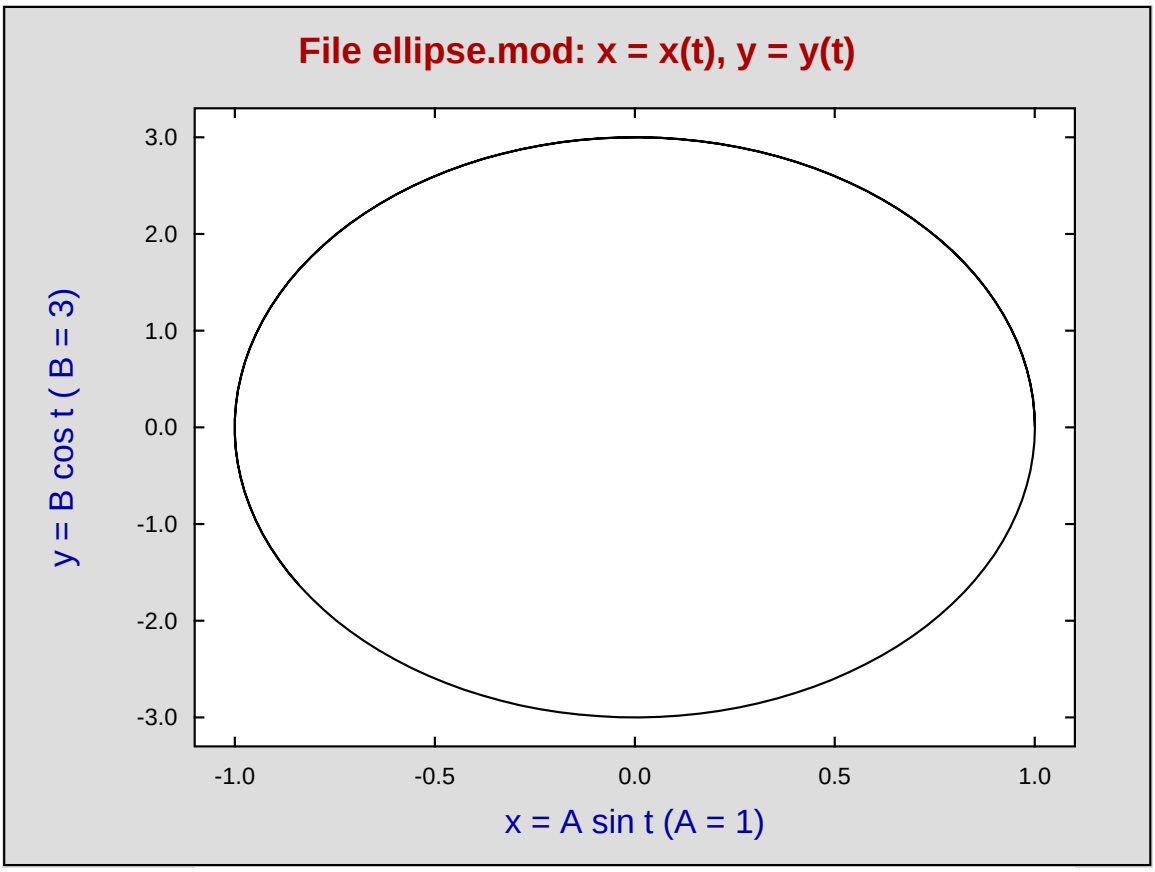
$$r = \frac{A\exp(\theta_0\cot\alpha)[\sin\theta_0 - \tan(\theta_0+\alpha)\cos\theta_0]}{\sin\theta - \tan(\theta_0+\alpha)\cos\theta}.$$

The figure above used $\alpha = p(1) = 1.4, \theta_0 = P(2) = 6$ and **usermod** to generate individual figures over the range $0 \le \theta = x \le 10$, then **simplot** plotted the ASCII text coordinates simultaneously, a technique that can be used to overlay any number of curves.

Plane curves as $x(t), y(t)$

The test file `ellipse_e.mod` defines an ellipse as now listed, followed by a plot where the eccentricity, which is evident from the ranges of the axes, is introduced by choosing $A = 1$ and $B = 3$.

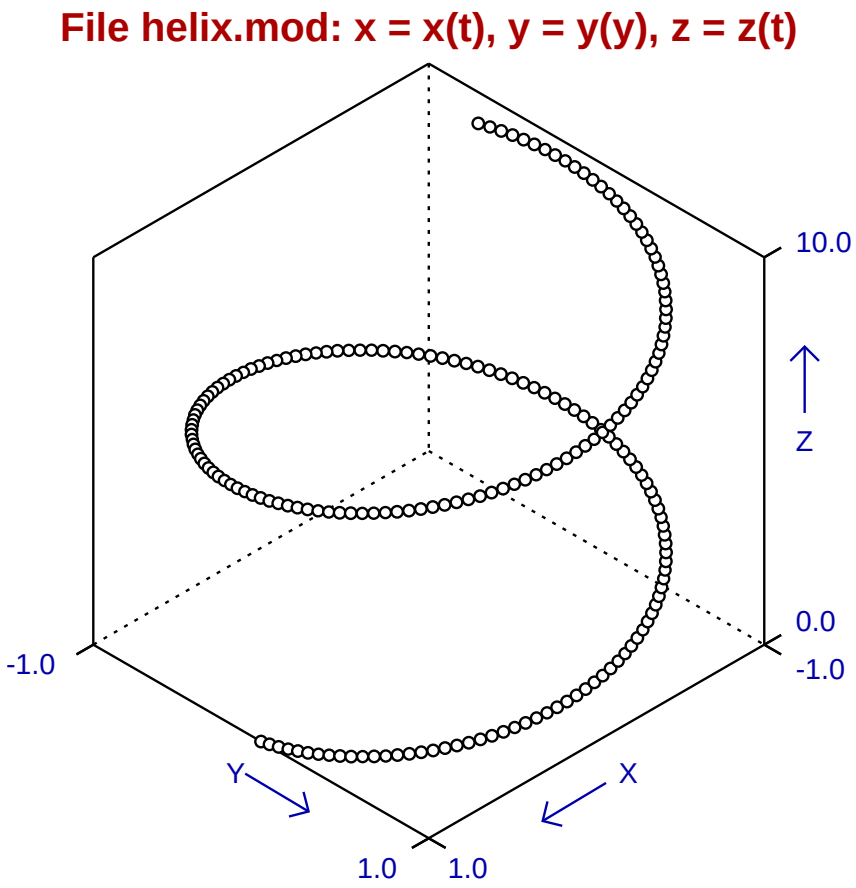
```
 %
Example: the ellipse
         X = A*cos(t), Y = B*sin(t)
         where: t = x, A = p(1), B = p(2)
                and X(t) = f(1), Y(t) = f(2)
 %
 2 equations
 1 variable
 2 parameters
 %
 begin{expression}
 f(1) = p(1)cos(x)
 f(2) = p(2)sin(x)
 end{expression}
 %
```



Space curves as $x(t), y(t), z(t)$

Note that, as with all parametric equations, program **simplot** transforms the parametric curve into sensible plotting parameters so that, as in this case with `helix_e.mod`, only minor editing of the data ranges is required.

```
 %
Example: the helix
         X = A*cos(t), Y = B*sin(t), Z = C*t
         where: t = x, A = p(1), B = p(2), C = p(3)
                and X(t) = f(1), Y(t) = f(2), Z(t) = f(3)
 %
 3 equations
 1 variable
 3 parameters
 %
 begin{expression}
 f(1) = p(1)cos(x)
 f(2) = p(2)sin(x)
 f(3) = p(3)x
 end{expression}
 %
```



Two dimensional families of curves

Users may need to plot families of curves indexed by parameters. For instance, diffusion of a unit mass of substance from an instantaneous plane source is described by the equation

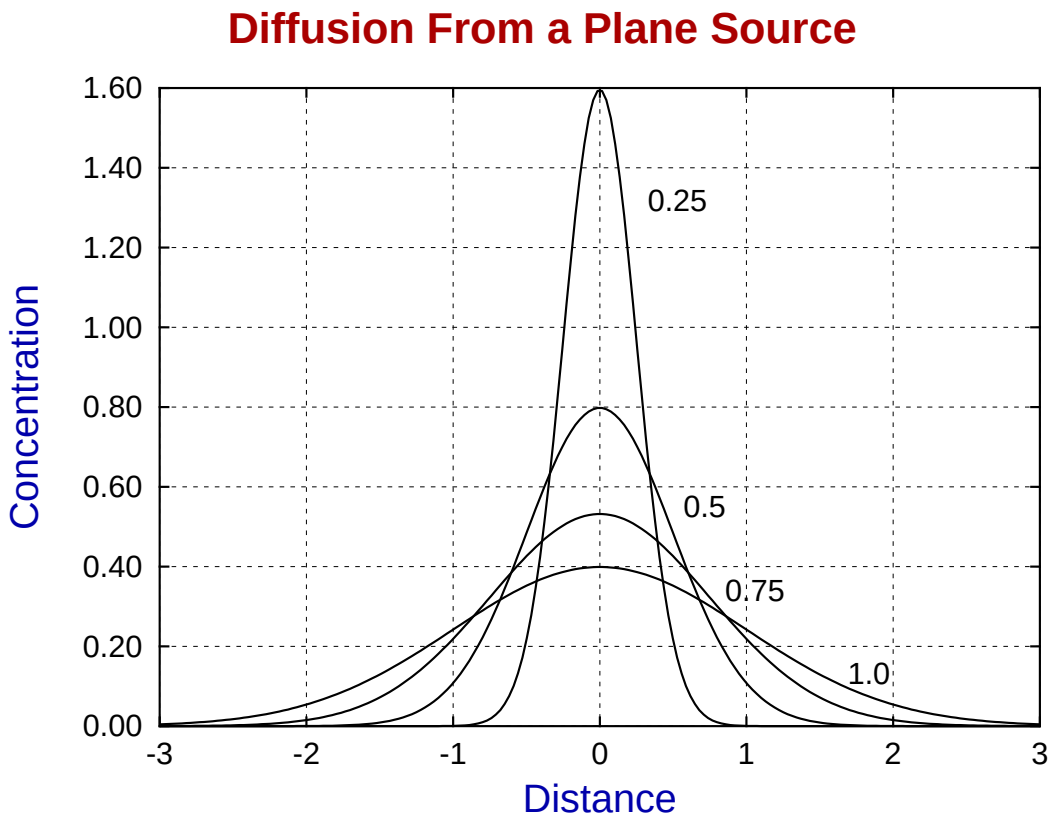
$$f(x) = \frac{1}{2\sqrt{\pi D t}} \exp\left(\frac{-x^2}{4Dt}\right)$$

which is, of course, a normal distribution with $\mu = 0$ and $\sigma^2 = 2Dt$, where D is the diffusion constant and t is time, so that $2Dt$ is the mean square distance diffused by molecules in time t. Now it is easy to plot the concentration $f(x)$ predicted by this equation as a function of distance x and time t given a diffusion constant D, by simulating the equation using **makdat**, saving the curves to a library file or project archive, then plotting the collected curves. However, there is a much better way using program **usermod** which has the important advantage that families of curves indexed by parameters can be plotted interactively. This is a more powerful technique which provides numerous advantages and convenient options when simulating systems to observe the behavior of the profiles as the indexing parameters vary.

The next figure shows the above equation plotted (in arbitrary units) using the model parameters

$$p_i = 2Dt_i, \text{ for } i = 1, 2, 3, 4$$

to display the diffusion profiles as a function of time. The plot was created using model file `family2d.mod`, which simply defines four identical equations corresponding to the diffusion equation but with four different parameters p_i. Program **usermod** was then used to read in the model, simulate it for the parameter values indicated, then plot the curves simultaneously.

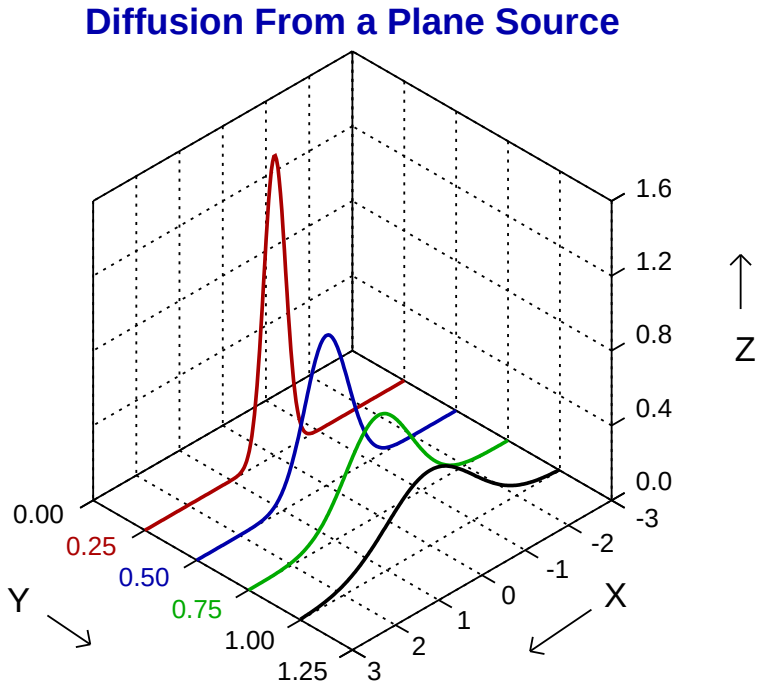


Three dimensional families of curves

Users may need to plot families of curves indexed by parameters in three dimensions. To show how this is done, the diffusion equation dealt with previously is reformulated, using $y = \sqrt{2Dt}$, as

$$z(x, y) = \frac{1}{y\sqrt{2\pi}} \exp\left\{-\frac{1}{2}\left(\frac{x}{y}\right)^2\right\}$$

and is plotted in the next figure for the same parameter values used before, but now as sections through the surface of a function of two variables.



This is, of course, a case of a family of parametric space curves projected onto the fixed values of y. Now the model file `family3d.mod` was used by program **usermod** to create this figure, using the option to plot n sets of parametric space curves, but you should observe a number of important facts about this model file before attempting to plot your own families of space curves.

- There are 4 curves so there are 12 functions of 1 variable
- Functions $f(1), f(4), f(7), f(10)$ are the parameter t, i.e. x
- Functions $f(2), f(5), f(8), f(11)$ are the y values, i.e. $\sqrt{2Dt}$
- Functions $f(3), f(6), f(9), f(12)$ are the z values, i.e. the concentration profiles

Finally, it is clear that n space curves require a model file that specifies $3n$ equations, but you should also realize that space curves cannot be plotted if there is insufficient variation in any of the independent variables, e.g. if all $y = k$, for some fixed parameter k.

The model files for diffusion curves

The model file family2d.mod defines two–dimensional curves as follows.

```
%
Normal pdf with mu = 0, sigma = p(i), i = 1, 2, ..., 4
to be used by program usermod to plot a 2D family of curves
for diffusion from a plane source where p(i) = sqrt(2Dt).
Try p(i) = sqrt(2Dt) = i/4, with -3 =< x =< 3.
%
4 equations
1 variable
4 parameters
%
1
pi
2
multiply
squareroot
divide
put(1)
x
p(1)
divide
2
power
0.5
multiply
negative
exponential
get(1)
multiply
p(1)
divide
f(1)
x
p(2)
divide
2
power
0.5
multiply
negative
exponential
get(1)
multiply
p(2)
divide
f(2)
x
p(3)
divide
2
power
0.5
multiply
negative
exponential
get(1)
```

```
multiply
p(3)
divide
f(3)
x
p(4)
divide
2
power
0.5
multiply
negative
exponential
get(1)
multiply
p(4)
divide
f(4)
%
```

The model file family3d.mod defines three–dimensional curves as follows.

```
%
4 space curves (12 equations) for x(i),y(i),z(i) as f(t).
Defines normal pdfs mu(i) = 0, sigma(i) = p(i), i=1,2,3,4
for program usermod to plot a 3D family of curves showing
diffusion from a plane source, where p(i) = sqrt(2Dt).
Try p(i) = sqrt(2Dt) (e.g. = i/4), with -3 =< t =< 3.
%
12 equations
1 variable
4 parameters
%
1
pi
2
multiply
squareroot
divide
put(1)                          store 1/srt(2*pi)
x
f(1)                             x(t) = x
p(1)
f(2)                             y(t) = p(1)
x
p(1)
divide
2
power
0.5
multiply
negative
exponential
get(1)
multiply
p(1)
divide
f(3)                             z(t) = function value
x
f(4)                             x(t) = x
```

```
p(2)
f(5)                          y(t) = p(2)
x
p(2)
divide
2
power
0.5
multiply
negative
exponential
get(1)
multiply
p(2)
divide
f(6)                          z(t) = function value
x
f(7)                          x(t) = x
p(3)
f(8)                          y(t) = p(3)
x
p(3)
divide
2
power
0.5
multiply
negative
exponential
get(1)
multiply
p(3)
divide
f(9)                          z(t) = function value
x
f(10)                         x(t) = x
p(4)
f(11)                         y(t) = p(4)
x
p(4)
divide
2
power
0.5
multiply
negative
exponential
get(1)
multiply
p(4)
divide
f(12)                          z(t) = function value
%
```

13.3.12 Adding text, arrows, and objects to graphs

In order to improve the usefulness of plots it is often valuable to be able to add text strings, arrows, and graphical objects to highlight features, which involves the following steps.

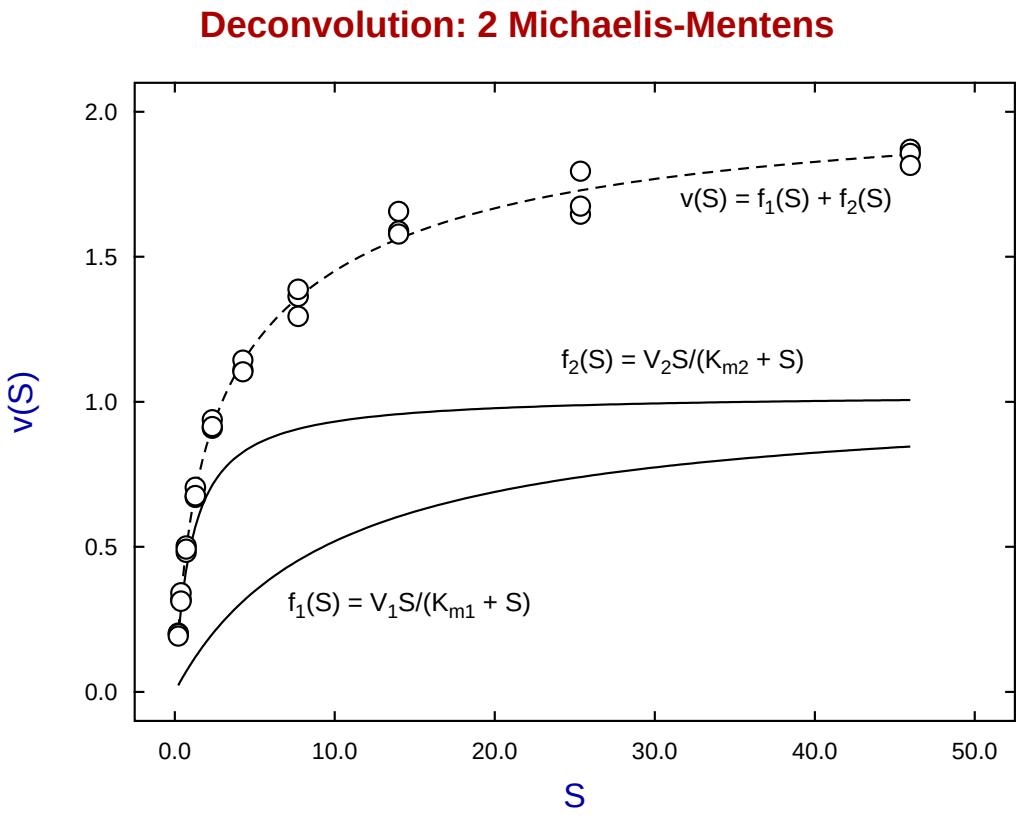
1. Display the plot in advanced graphics mode, then open a new text string using the [Text] button.
2. Type in the text string, then move it into position by dragging the red arrow icon to the coordinates required and selecting the [Text move] button.
3. Edit the string by selecting the font, size, colour, and orientation, including advanced editing if subscripts or similar font changes are required.
4. Arrows, lines, or boxes can then be added to remove ambiguities using the [A/L/B] button .
5. Individual graphical objects or an information panel can be added if appropriate.

This can be done for several text strings by treating each one individually.

Here is an example using **mmfit** to fit a sum of two Michaelis-Menten functions to data in test file `mmfit.tf4`, and then choosing the graphical deconvolution control to display the two components that constitute the best fit curve. Note that the text strings were initially typed in the form

```
f1 = V1S/(Km1 + S)
```

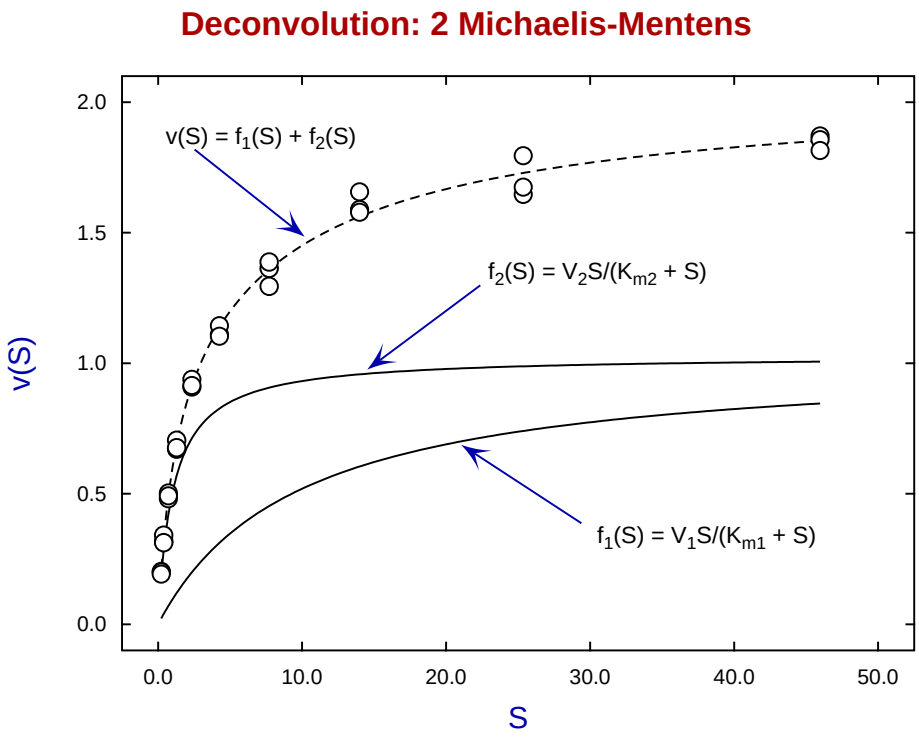
followed by selecting the advanced text editing option to introduce subscripts for indices with the expressions `f1`, `V1`, `Km1`, `f2`, `V2`, and `Km2`, etc. Note that a temporary grid can be drawn to aid the positioning and lining up of text strings.



Arrows, lines, hooks, and boxes differ from text strings in that they require two coordinates, namely

- first the red arrow icon is used to set the head position,
- then the red arrow icon is used to fix the tail position.

This is shown in the next figure which was derived from the previous one by simply changing the position of the text strings and adding script arrows.

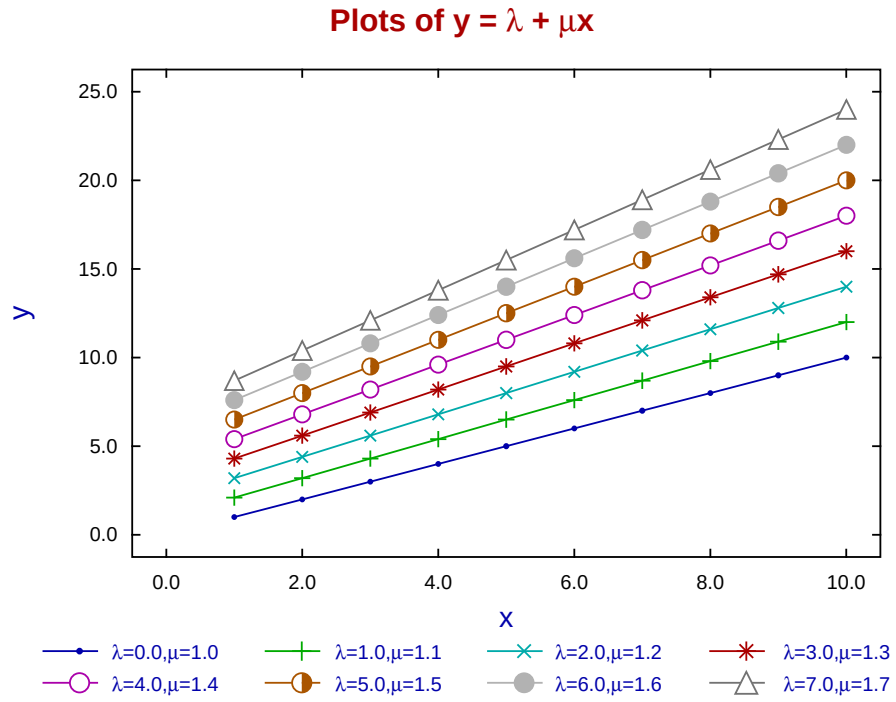


Note that the arrow, line, box options include many useful objects for illustrating graphs such as the following.

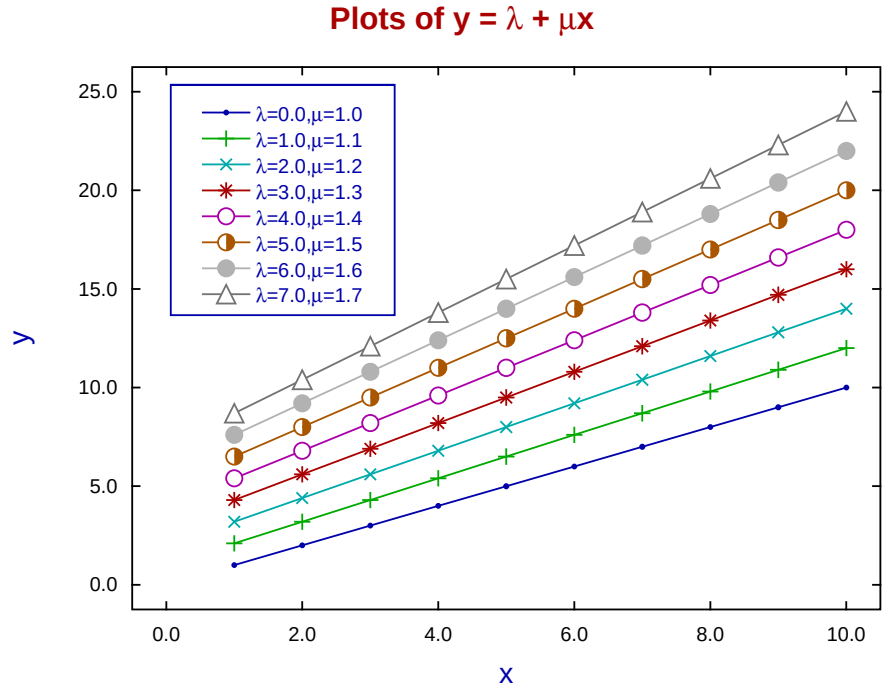
- Standard line arrows
- Hollow outline arrows
- Solid filled arrows.
- Script arrows.
- Headless arrows, i.e. lines.
- Oblique or horizontal rectangles.
- Horizontal Boxes for enclosing text.
- Three sided rectangles, i.e. hooks.

These are all positioned in the same way, that is, two coordinates must be set using the red arrow icon for the head and the tail. However there is a new feature when arrows, lines, or boxes have been selected: they can be displayed in the background underneath the data and best fit curves, or they can be displayed on top of the data and best fit curves which will often obscure features. This can be useful. For instance, a solid rectangle with background colour can be used to define an obscured region of a graph into which descriptive text or other illustrative material can be placed.

Graphical objects can be added to graphs in exactly the same way as arrows, but perhaps the most useful procedure is the ability to supply a panel containing descriptions of the data plotted, as in the next plots which used the [Panel] buttons.



Note that the panel can be at the bottom, at the side, or moved into the interior of the graph and optionally highlighted by being enclosed by a horizontal box as shown next.



14 PostScript graphics (EPS)

Tutorials and worked examples for simulation, curve fitting, statistical analysis, and plotting.
https://simfit.uk
https://simfit.org.uk
https://simfit.silverfrost.com

Figure 1 shows how the effect of plotting a positive 4:4 rational function in semi-logarithmic space can be enhanced by adding Greek letters, subscripts, and superscripts, and by using powers of ten on the X-legend.

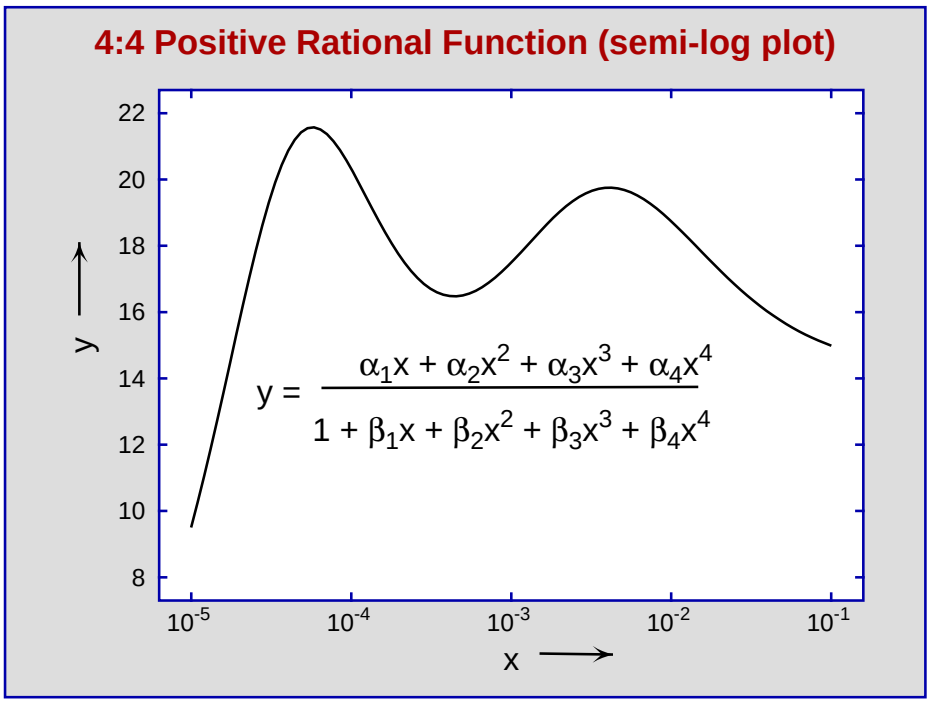


Figure 1: Plotting subscripts, superscripts, and math symbols

This document has details about how to use the SimFIT package to create such graphs with special characters and extra features as follows:

- Introduction to SimFIT EPS PostScript files;
- Plotting selected characters as subscripts and superscripts;
- Adding accents to letters in titles, legends, and labels;
- Replacing standard characters by mathematical symbols;
- Editing SimFIT PostScript files in a text editor;
- Using Postscript specials to add logos etc.; and
- Advanced techniques for LaTeX and PSFrag users.

14.1 Introduction to Simfit EPS PostScript files

There are essentially three types of image files as follows.

1. **Bitmaps and compressed bitmaps**

   Raw bitmaps (e.g., `.bmp`) are used to record the characteristics of every pixel in a display or hardcopy, typically a digital photograph. They are limited by the resolution of the captured image and are usually large, also the images break up by pixelation if they are enlarged. They are generally compressed into alternative formats (e.g., `.jpg`, or `.png`) where a certain loss of quality is offset by a great decrease in size. In scientific work they are mainly used for complicated diagrams, e.g., photographs of microscopic sections, where there are no distinct objects such as titles, legends, lines, curves, plotting symbols, etc.

2. **Vector graphics**

   Often scientific graphs consist only of lines for axes, curves, plotting symbols, and text for titles and legends with featureless backgrounds, so that storing a bitmap would be wasteful of space. However the main advantage of vector formats (e.g., `.eps`, `.svg`, or `.emf`) is that they are device-independent so they can be displayed or printed at any resolution with no loss of information. They can also be used to generate compressed bitmap files, but it should be noted that scientific graphs can sometimes generate vector hardcopy that is even more bulky than the corresponding bitmap if there are very large numbers of objects being plotted.

3. **Embedded bitmaps**

   Unfortunately vector graphics files can often consist of wrappers containing bitmaps which leads to files with a vector file extension that are actually no better than bitmaps. Some graphics programs export supposed `.eps` files as embedded bitmaps, so losing many of the advantages of true vector files. Also note that `.pdf` files created from vector `.eps` files using GhostScript retain some of the characteristics of actual vector files, but many programs simply distill graphs into `.pdf` files as embedded bitmaps.

Users of the SimFIT package are strongly urged to save all graphs as SimFIT `.eps` files because they have the following advantages.

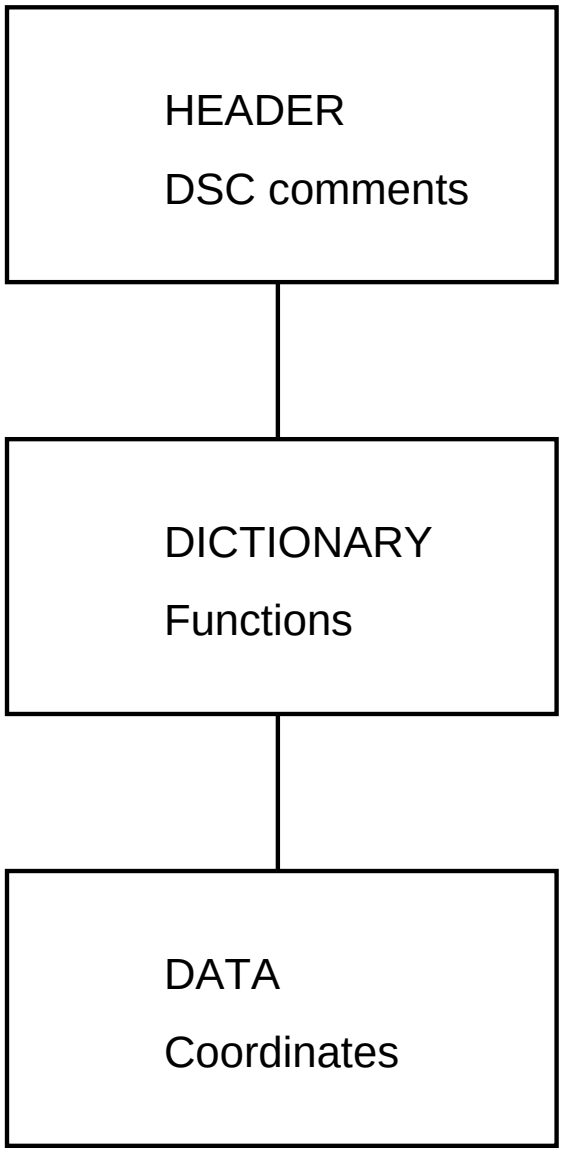
- They are true encapsulated PostScript vector files consisting of a single page with a BoundingBox.
- They are structured in such a way that they can be edited using a text editor to change dimensions, line types, symbols, colours, titles and legends, etc.
- They can be used retrospectively to create alternative types of image files.
- SimFIT provides facilities to edit such files, or make various types of collages.

Although SimFIT can make and edit `.eps` files with no additional software it will be found that, in order to make full use of the PostScript opportunities, it is necessary to download and install the GhostScript package, and also advisable to download and install the GSview package. GhostScript can be used to transform `.eps` files into other graphics formats, while GSview can be used to view and print them.

14.1.1 The sections of Simfit eps files

In order to be able to edit SimFIT `.eps` files it is necessary to appreciate that there are three distinct sections. The first section contains the BoundingBox coordinates that must be present in `.eps` files to specify the size of such one page graphs along with the document structuring comments. The second section is a dictionary containing the functions that can be used in SimFIT `.eps` files to draw lines and plotting symbols. The third section is a list of coordinates and colors where lines, symbols and text strings are to be drawn. This structure is summarized in the next diagram.

SIMFIT .eps file structure



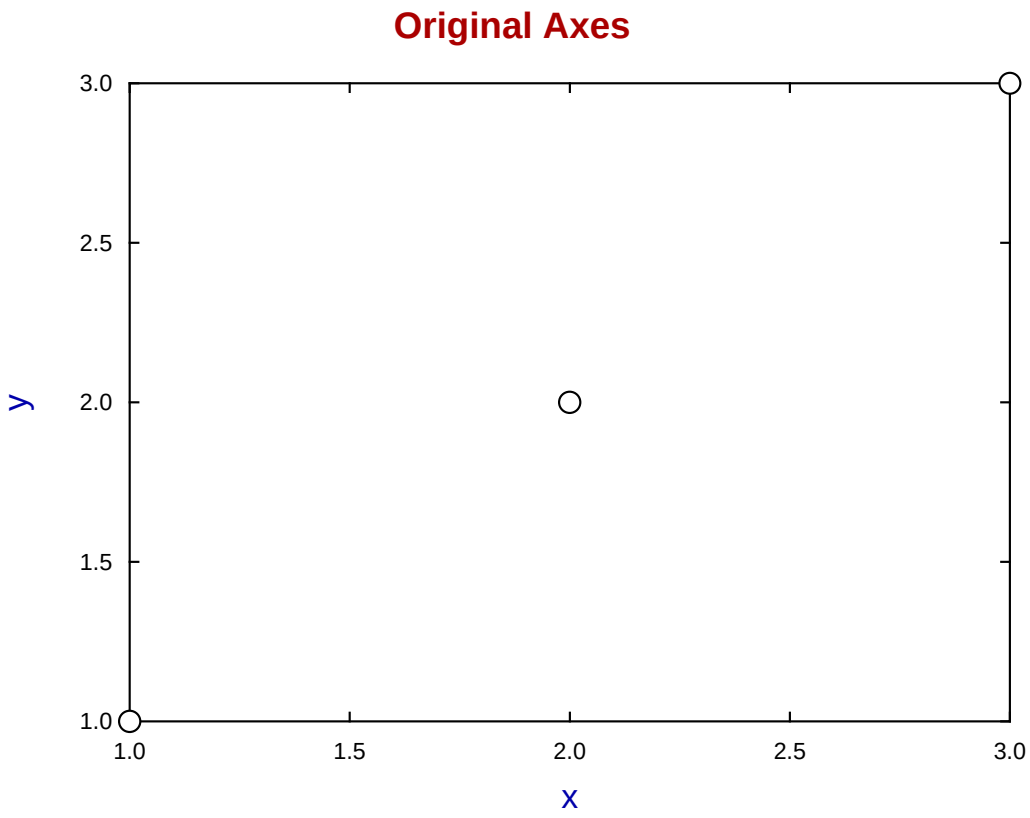
When editing SIMFIT PostScript files retrospectively the following rules must be obeyed until you become familiar with the structure.

1. Never change anything in the HEADER or DICTIONARY sections.
2. You can edit the title, legends, colors, and other display features in the DATA section.
3. Perform editing a step at a time and check the effects using GSview.
4. Make sure you save the file at each stage of editing.

Extensive descriptions about editing the DATA section will be found in the SIMFIT reference manual `w_manual.pdf` available from the SIMFIT website, but a number of simple examples will be given in the succeeding tutorials by way of a more gentle introduction.

14.1.2 A simple example

As an example consider the following simple default graph which is easily constructed by requesting to plot the coordinates (1,1), (2,2), and (3,3) using program **simplot**



Some of things a user might wish to change retrospectively could be as follows.

- Change the title
- Change the legends
- Change the plotting symbols
- Change the colors

Now SimFIT provides a PostScript editor program **editps** to perform such tasks but, in actual practise, it is easier to edit the PostScript file in a text editor such as **notepad** because of these three special features found in SimFIT Postscript files.

1. They are ASCII text files and can be edited in any text editor.
2. They are encapsulated PostScript files (`.eps`) and describe just a single page.
3. They have been uniquely designed to make such editing very easy.

As you will need to view the results of editing you will need a PS viewer such as **GSview** which requires an installed copy of **GhostScript**.

14.1.3 The header section

The following code is contained in the header.

```
%!PS-Adobe-3.0 EPSF-3.0
%%BoundingBox:    72   252   520   588
%%Creator: Simfit Version 7.2.8 (simfit.org.uk)
%%Title: colours=72/ISOLatin1Encoding/Accents/special/PSfrag/dict=300
%%CreationDate: Saturday, 15 April 2017
%%EndComments
%
%Start of SIMFIT PostScript file
%
save %save current state before clipping, etc.
    70   250   522   250   522   590    70   590%#clipping
newpath moveto lineto lineto lineto closepath clip newpath
   72.00  252.00 translate    0.07    0.07 scale    0.00 rotate%#portrait
   12.00 setlinewidth 0 setlinecap 1 setlinejoin [] 0 setdash
    2.50 setmiterlimit
%
%prolog(1) to (6) can be used by DVIPS as a header
%****cut the invariant prolog/header out from here
```

It includes the document structuring comments together with some technical instructions, and this section would only be edited by experienced users.

14.1.4 The dictionary section

This contains definitions for all the plotting functions, colors and fonts required to display the data contained in the data section, and this section would only be edited by very experienced users.

```
/SIMFIT 300 dict def SIMFIT begin
%
% prolog(1): definitions
%
/C{copy}def /D{def}def /E{exch}D /F{findfont}D /GR{grestore}D
/GS{gsave}D /M{moveto}D /N{newpath}D /P{pop}D /R{rmoveto}D
/S{scalefont setfont}D /d{dup}D /i{putinterval}D /p{put}D
%
% prolog(2): construct Greek/maths font
%
...
...
...
/ty-font /Helvetica      D%text right y-mid
/tz-font /Helvetica      D%text left  y-mid
/ti-size 204 D /xl-size 187 D /yl-size 187 D /zl-size 187 D
/tc-size 144 D /td-size 144 D /tl-size 144 D /tr-size 144 D
/ty-size 144 D /tz-size 144 D
/sb-size 0.75 D /sp-size 0.75 D%sub/superscript expansion
/y-down -0.33 sb-size mul D /y-up 0.33 sp-size div D%sub/sup shift
%
foreground thickness setlinewidth
```

14.1.5 The data section

This is first given in full and then the sections that are most likely to be edited are discussed in detail.

```
/background{c15}D
2 setlinecap
1070 671 5959 671 5959 4215 1070 4215 4 pc%#8
/ty-size ty-size   0.900 mul def
/tl-size tl-size   0.900 mul def
1070 671 1118 671 li%#4
5959 671 5911 671 li%#4
(1.0) 974 671 ty%#()2
(000) fx
1070 1557 1118 1557 li%#4
5959 1557 5911 1557 li%#4
(1.5) 974 1557 ty%#()2
(000) fx
1070 2443 1118 2443 li%#4
5959 2443 5911 2443 li%#4
(2.0) 974 2443 ty%#()2
(000) fx
1070 3329 1118 3329 li%#4
5959 3329 5911 3329 li%#4
(2.5) 974 3329 ty%#()2
(000) fx
1070 4215 1118 4215 li%#4
5959 4215 5911 4215 li%#4
(3.0) 974 4215 ty%#()2
(000) fx
/ty-size ty-size   1.111 mul def
/tl-size tl-size   1.111 mul def
/tc-size tc-size   0.900 mul def
/tl-size tl-size   0.900 mul def
1070 671 1070 719 li%#4
1070 4215 1070 4167 li%#4
(1.0) 1070 462 tc  %#()2
(000) fx
2292 671 2292 719 li%#4
2292 4215 2292 4167 li%#4
(1.5) 2292 462 tc  %#()2
(000) fx
3515 671 3515 719 li%#4
3515 4215 3515 4167 li%#4
(2.0) 3515 462 tc  %#()2
(000) fx
4737 671 4737 719 li%#4
4737 4215 4737 4167 li%#4
(2.5) 4737 462 tc  %#()2
(000) fx
5959 671 5959 719 li%#4
5959 4215 5959 4167 li%#4
(3.0) 5959 462 tc  %#()2
(000) fx
/tc-size tc-size   1.111 mul def
/tl-size tl-size   1.111 mul def
/ti-size ti-size   1.000 mul def
c4
(Original Axes) 3195 4467 ti%#title
(0000000000000) fx
/ti-size ti-size   1.000 mul def
/xl-size xl-size   1.000 mul def
c1
(x) 3515 192 xl%#x legend
```

```
(0) fx
/xl-size xl-size   1.000 mul def
/yl-size yl-size   1.000 mul def
(y) 501 2443 yl%#y legend
(0) fx
/yl-size yl-size   1.000 mul def
0 setlinecap
c0
1070 671 59 ce%#2
3514 2443 59 ce%#2
5959 4215 58 ce%#2
```

14.1.6 Editing the title and legends

Searching for the key word title locates the next section.

```
c4
(Original Axes) 3195 4467 ti%#title
(0000000000000) fx
/ti-size ti-size   1.000 mul def
/xl-size xl-size   1.000 mul def
c1
(x) 3515 192 xl%#x legend
(0) fx
/xl-size xl-size   1.000 mul def
/yl-size yl-size   1.000 mul def
(y) 501 2443 yl%#y legend
(0) fx
```

Interchanging the colors c1(blue) and c4(red) and altering the title and can then be done leading to the replacement section shown next.

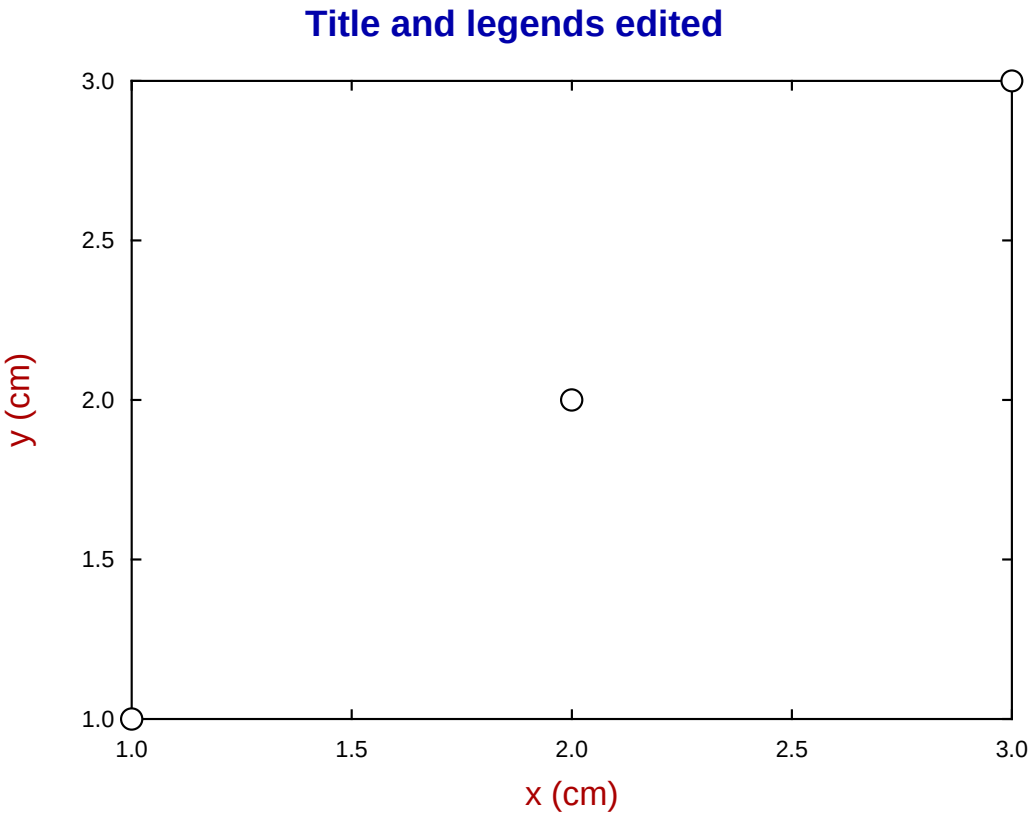
```
c1
(Title and legends edited) 3195 4467 ti%#title
(000000000000000000000000) fx
/ti-size ti-size   1.000 mul def
/xl-size xl-size   1.000 mul def
c4
(x \(mm\)) 3515 192 xl%#x legend
(00000000) fx
/xl-size xl-size   1.000 mul def
/yl-size yl-size   1.000 mul def
(y \(mm\)) 501 2443 yl%#y legend
(00000000) fx
```

Two comments are needed in order to understand the results of this editing.

1. When a text string such as a title or legend is edited it is necessary to make sure that the character key string underneath the text is padded or contracted if required to make sure the key has at least as many characters (in this case the 0 characters denoting a normal font) as the text string.

2. Since character strings in PostScript are enclosed in round brackets it is necessary to prefix any brackets introduced inside the string by the backslash escape character.

The edited file is shown next.



14.1.7 Editing line and symbol types

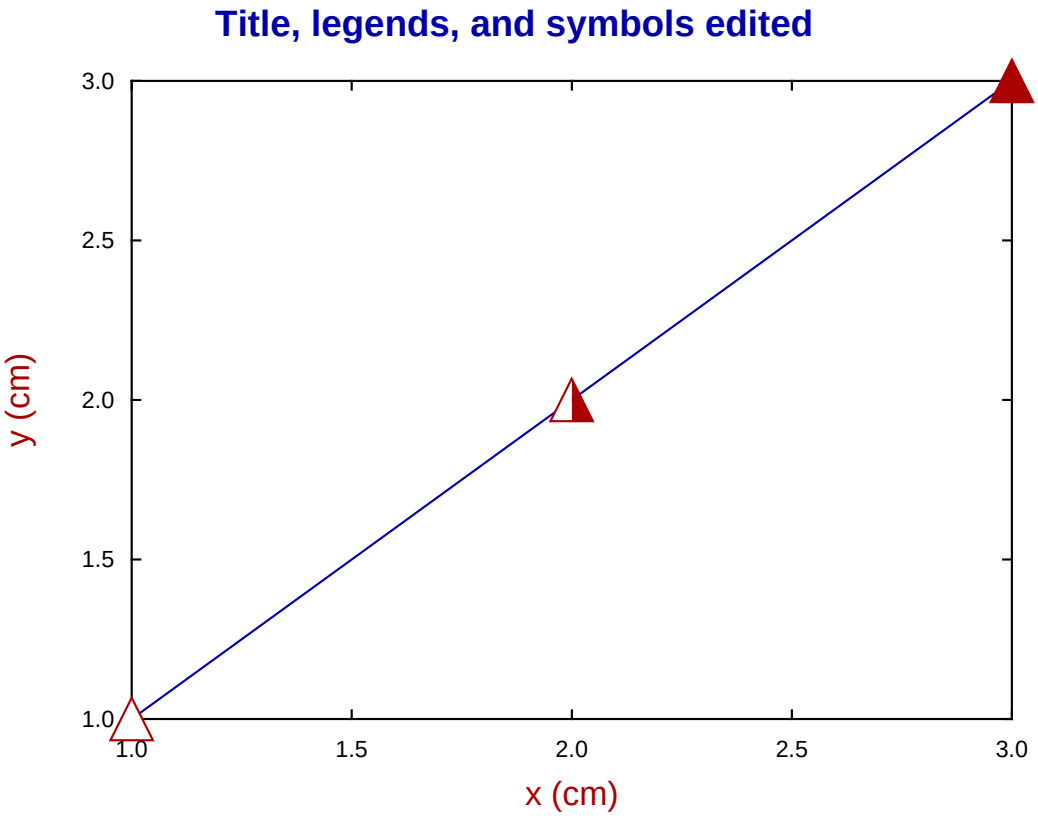
It is frequently required to change the size, colors, and types of lines and plotting symbols and the code in the data section defining the black (c0) empty circles (ce) is as follows.

```
c0
1070 671 59 ce%#2
3514 2443 59 ce%#2
5959 4215 58 ce%#2
```

For example, using the command c1 and li to add a blue line, altering c0 to c4 to change color to red, then changing empty circles (ce) into empty triangle (te), half-filled triangle (th), and filled triangle (tf), together with doubling the size of the symbols from 59 to 118, as in this code

```
c1
1070 671 5959 4215 li
c4
1070 671 118 te%#2
3514 2443 118 th%#2
5959 4215 118 tf%#2
```

creates the next graph.



Clearly users will require much more information to be able to edit all the features of SIMFIT PostScript files, and the necessary details along with numerous worked examples follows.

14.2 Plotting non-standard characters

The SIMFIT assumption is that users will want professional quality hardcopy and not be content with standard Windows graphics. In other words, if a permanent record is required for a displayed graph, then a *.eps file will be created for immediate use or retrospective conversion into a *.png or similar file for printing or incorporating into documents. Since the Windows display and PostScript files use different fonts, then the following details should be noted, otherwise the PostScript file will not have the same edited text strings as the Windows display.

The basic problem and the solution to it

As long as the characters to be plotted are from the standard 7-bit ASCII standard set, i.e. characters 32 to 126 then all that is required is to use the Simple editing control to edit text strings for titles, legends, labels, etc. European accented characters from the ISOLatin1 8-bit set that are entered from the keyboard or font table are also plotted correctly due to a built in transformation mechanism. However, to plot characters with arbitrary accents like hats and bars, superscripts, or subscripts, or to plot maths symbols or characters from the Greek alphabet, then the more advanced techniques have to be used. This is to ensure that what is displayed in the Windows bitmap will be the same as what is displayed in the PostScript output.

14.2.1 7-bit ASCII characters 33 to 126

These, together with the space character (32), are the standard non-accented characters allowed in the simple editing control.

```
! " # $ % & ' ( ) * + , - . /
0 1 2 3 4 5 6 7 8 9
: ; < = > ? @
A B C D E F G H I j K L M N O P Q R S T U V W X Y Z
[ \ ] ^ _ ‘
a b c d e f g h i j k l m n o p q r s t u v w x y z
{ | } ~
```

14.2.2 The basic character plotting technique

When SIMFIT displays a character string in a plot, then associated with every character is a key indicating if the the character is to be displayed in any special way. For instance in this string key pair

```
Area in cm2
00000000002
```

The 0 indicates a normal character while the 2 indicates a superscript, so the string will be displayed as

$$\text{Area in cm}^2$$

where the 2 is now shown as a superscript.

14.2.3 Advanced editing

Fortunately the user does not need to know any of these details as, when Advanced editing is selected, users can choose to either

- Edit the individual characters;
- Edit the individual keys, or;

❍ Select a character together with the associated key from a displayed table.

It is assumed that when users want a non-standard character they will first use the simple editing control to input a string, then select the advanced editing control to add details.

As another example, consider the numerator of the formula displayed in figure 1. This would first be entered by simple editing as the character string

```
a1x + a2x2 + a3x3 + a4x4
```

then the advanced control would be used to replace the characters by Greek equivalents, say from the Symbol or maths tables, and the keys would be edited for superscripts and subscripts as shown below

```
a1x + a2x2 + a3x3 + a4x4
31000031x200031x200031x2
```

followed by similar editing of the denominator to give

```
1 + b1x + b2x2 + b3x3 + b4x4
000031000031x200031x200031x2
```

Finally, adding a horizontal line as a an arow/line type of object would result in the display of a properly formatted rational function like this

$$\frac{\alpha_1 x + \alpha_2 x^2 + \alpha_3 x^3 + \alpha_4 x^4}{1 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \beta_4 x^4}.$$

The rest of this section is for those who are familiar with the PostScript language and wish to use the powerful PostScript techniques supported by SimFIT to create special effects.

14.2.4 Octal codes

For users who just want to use the simple editing control without resorting to the advanced editing control to plot accented characters not on their keyboard, it is possible to enter octal codes directly. For instance \361 will produce ñ because the character/key pair will automatically have an 8 placed under the backslash to indicate that the ISOLatin1 character set should be used instead of the Standard font. Note that users of the advanced editing mode will sometimes see a question mark ? or a blank instead of a 0 to represent a normal font, which happens when strings have been shortened or lengthened by the editing process.

The rule is that every character must be paired with a unique identifying key, and the PostScript file will only plot if this condition is satisfied. However, there is an important exception to this rule. When octal codes are used it is vital to realise that the back slash followed by three digits represents a single character, i.e. a single byte, so it only requires one key. As just remarked, a key must be placed under the backslash to indicate the font set required as below.

```
This is what n tilde looks like \361
00000000000000000000000000000008
```

which will plot as

This what n tilde looks like ñ

14.2.5 A detail about PostScript fonts

It has to be emphasized that the Windows bit-map display is using TrueType fonts, while the Postscript file will use PostScript fonts. Because these are not identical, it is possible that arrows and other graphical objects added to graphs will not have exactly the same appearance. The objects will have the same starting coordinates, but the lengths of the character strings will not be exactly equal, which can lead to unwanted overlap of objects and character strings in the PostScript display. Similarly the colours and line thickness will not be the same, and so it is vital to check a PostScript visual display in your PostScript viewer before saving a PostScript file.

14.3 PostScript procedures

The best way to use SimFIT graphics is to archive standard sized SimFIT PostScript files in portrait orientation and then, when required, manipulate them, followed by printing hardcopy, or by transforming into other formats, such as .png, for pasting into documents.

Most simple editing operations can be done using a text editor as described later, but for more extensive manipulations program **editps** can be used as now described.

14.3.1 Using editps to manipulate PostScript files

Several points must be mentioned concerning SimFIT EPS files and program **editps**.

1. An encapsulated PostScript file (*.eps) is a special type of self-contained one page PostScript file containing a BoundingBox with dimensions, so that the file can be easily manipulated for re-sizing and inclusion within documents.
2. All PostScript files created by SimFIT adhere to this convention.
3. Program **editps** will accept any such files, but some features will only work with SimFIT .eps files.

14.3.2 Editing Simfit Postscript files

After a SimFIT file has been loaded into **editps** it is possible to search for such items as the title, or legends, etc., and edit as required. However, as described later, it is much easier to do such editing using a simple text editor.

Further, note that this type of editing is restricted to SimFIT PostScript files which contain special markers to locate titles, legends, etc.

14.3.3 Rotating, re-sizing, and changing aspect ratios.

In addition program **editps** can be used to rotate or re-size SimFIT EPS PostScript files and perform shearing transformations. However, the main use is create various types of collages such as these.

- Simple collages.

  Here all the constituent files have identical BoundingBoxes, i.e., they are the same size, and the collage can be created directly.

- Freestyle collages.

  If the constituent files do not have identical BoundingBoxes then hand-crafting is required.

- Adding new files as insets to existing files.

  Here it must be realized that there are two cases.

  - The inserted child file can have an invisible background, when the parent file will show through.
  - The inserted child file can have an opaque background, when the parent graph will be obliterated where the child overlaps.

14.3.4 Creating simple collages

Figure 14.3.4 illustrates a simple collage created using **editps** from a set of SimFIT PostScript files assembled into a library file. If required, titles, labels, and extra text can be added by program **editps** to identify the sub-graphs or add further details. To create such collages it is advisable that all the files supplied should have the same dimensions and orientation, as this facility can only generate collages with fixed cell dimensions.

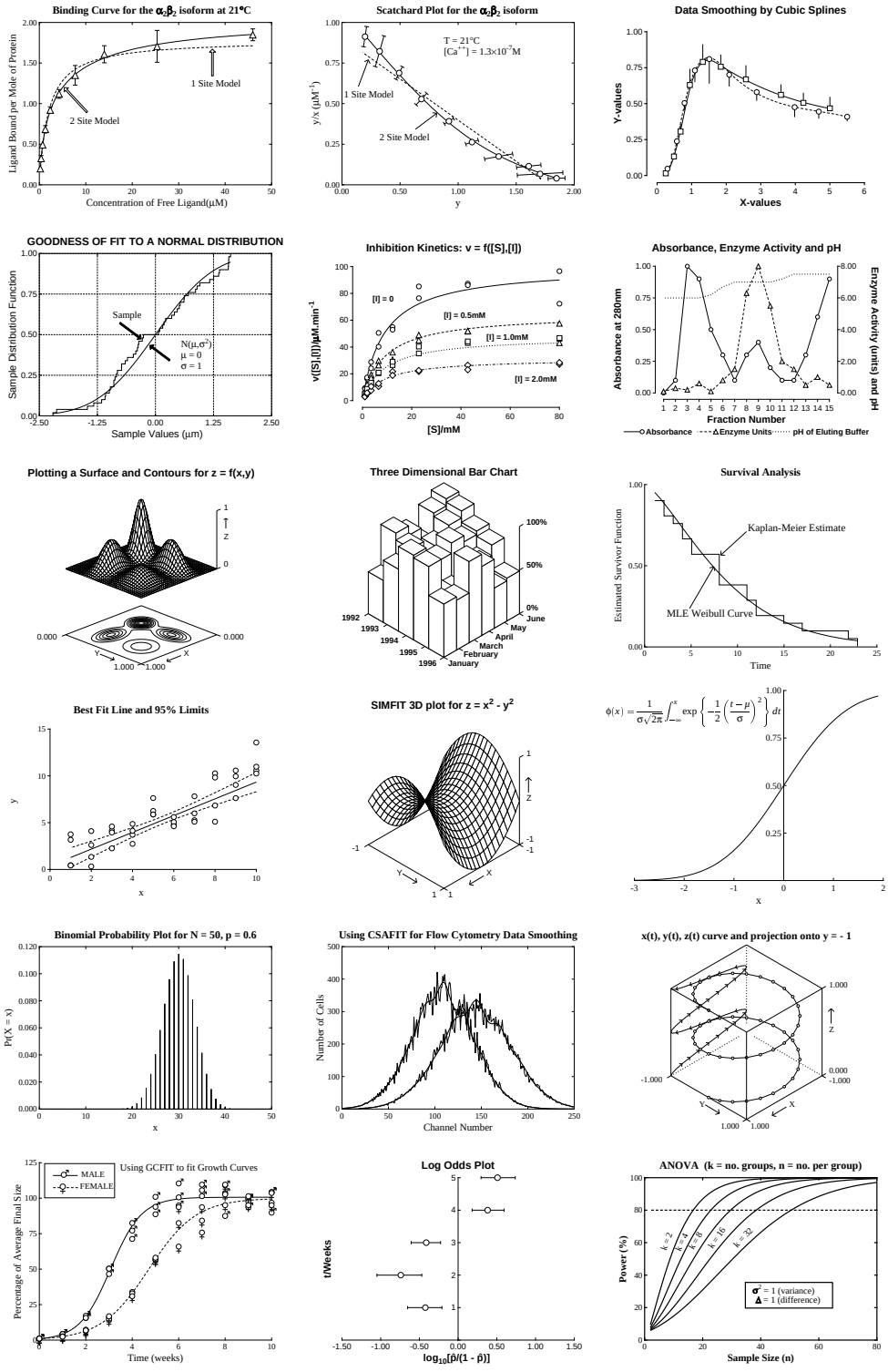


Figure 2: Collage 1

14.3.5 Creating freestyle collages

Plots that are incorporated into simple collages as just described must all have the standard default SimFIT portrait format, i.e., with identical standard BoundingBoxes. However, users often wish to collect plots of varying sizes together into an arbitrary pattern, as in figure 3.

This illustrates the value of changing font size and line thickness as graphs are re-sized. In general, it will be clear from figure 3 that, if a graph is to be reduced in size before building into a freestyle collage, it is a good idea to increase the size of fonts and thickness of lines.

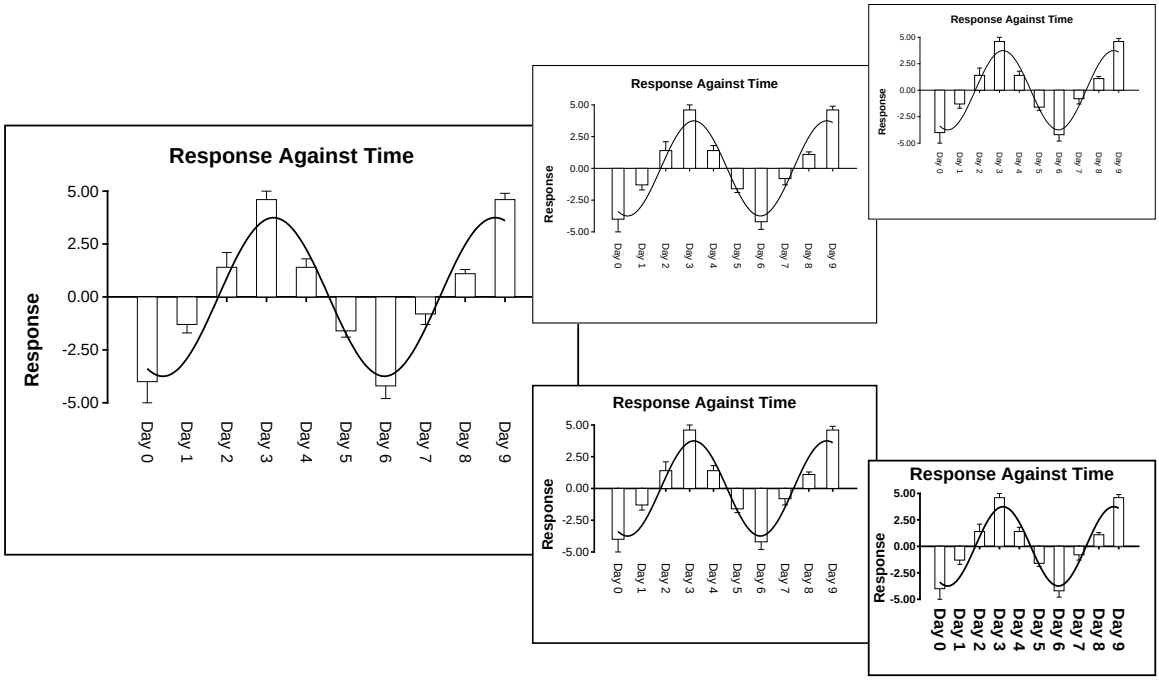


Figure 3: Collage 2

In the above figure the upper sub–figures are derived from the large figure by reduction, so the text becomes progressively more difficult to read as the figures scale down. In the lower sub-figures, however, line thicknesses and font sizes have been increased as the figure is reduced, maintaining legibility. Such editing can be done interactively, but SimFIT PostScript files are designed to make such retrospective editing easy. Clearly, the graphs should be edited individually before assembling into the final collage.

Figure 4 and figure 5 show further examples of how such collages can be assembled using **editps** in freestyle mode with a set of graphs that can have arbitrary dimensions and rotations. In addition to being able to move the sub–graphs into any positions, this procedure also allows interactive differential re-sizing of individual graphs.

There is an extremely important point to remember when creating freestyle collages: it is possible to create PostScript files from SimFIT where the background, if white, can be either transparent or opaque. Note that PostScript files with opaque white backgrounds, as in the above sub–figures, will obscure any graphs they overlay. Of course, sometimes this is desired, but often a transparent white background may be preferred. This choice is determined, of course, by the configuration option in use when the PostScript file is created

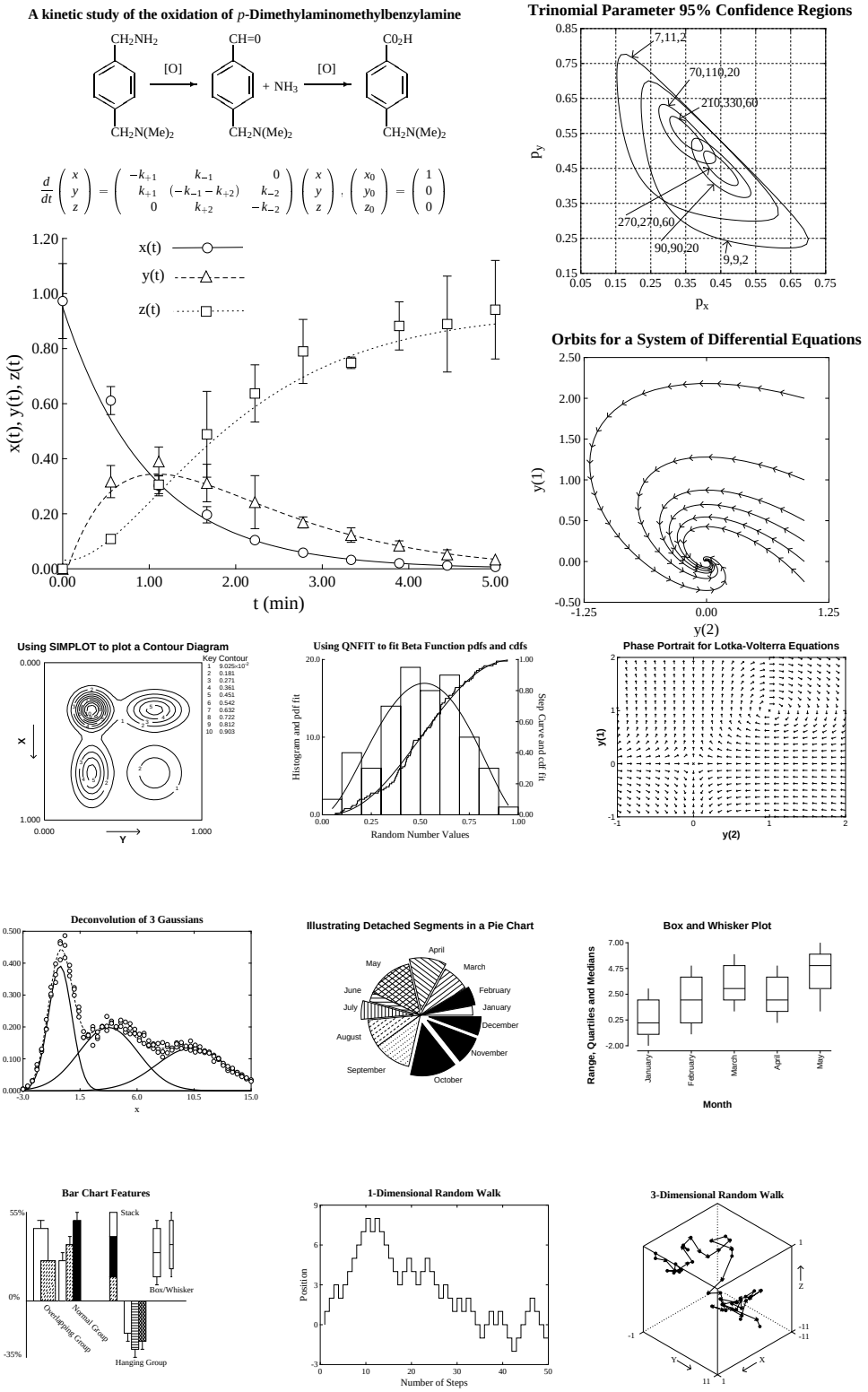


Figure 4: Collage 3

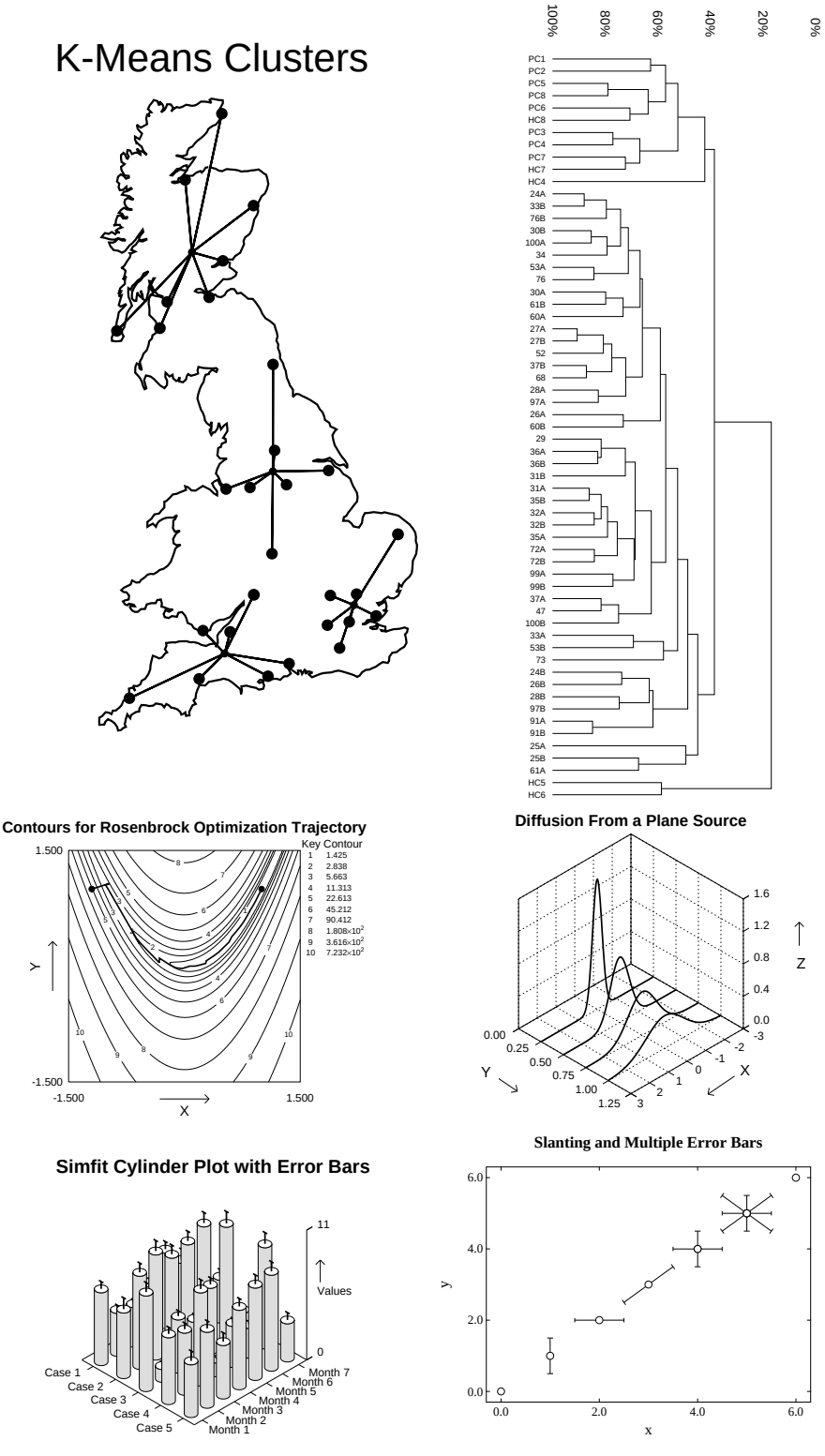


Figure 5: Collage 4

14.3.6 Subsidiary figures as insets

Figure 6 illustrates a special type of freestyle collage where a sub-graph is placed inside a parent graph. Sometimes it is best to enlarge the fonts and increase the line thicknesses when a sub-graph is going to be reduced in size in this way, and it is always important to remember the effects of opaque and transparent backgrounds that SIMFIT allows.

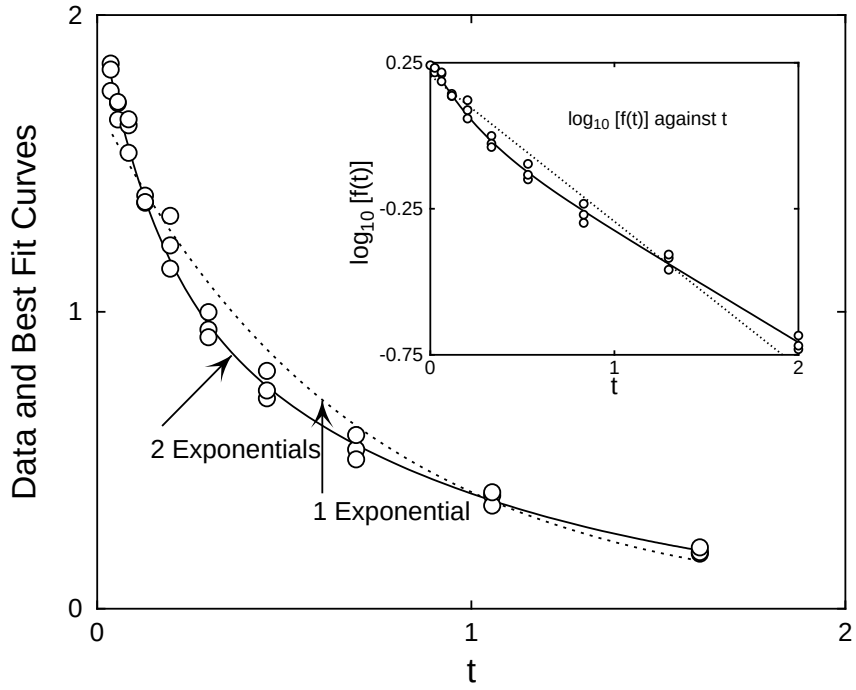


Figure 6: Subsidiary figures as insets

First of all the plots in figure 7 were created using **exfit** with test file `exfit.tf4` after fitting 1 exponential then 2 exponentials. Note that the line thickness and font size have been increased in the transformed plot as it is going to be reduced in the inset. Figure 8 was then created by **editps** using the option to create a

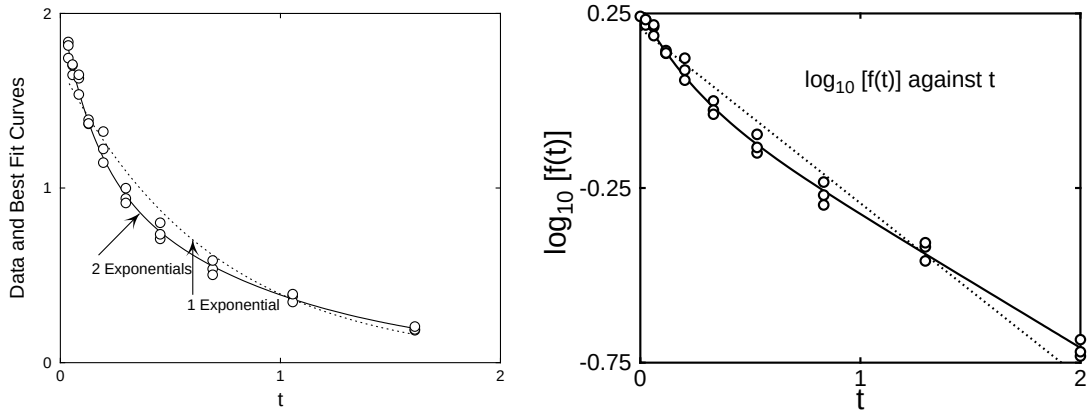


Figure 7: Insets 1: Exponential fitting and semilog transforms

freestylecollage. Note how, in the left hand plot the option to plot an opaque background even when white was selected and the transformed plot obscures the underlying main plot. In the right hand plot the option for a transparent background was used so that the main plot was not obscured. Both techniques are valuable when creating insets, and all that is now necessary to create figure 6 is to shrink the transformed plot and translate it to a more convenient location. A further point to note is that SimFIT plots have a border, which is obscuring more of the left hand main figure in figure 8 than seems necessary. When subsidary figures are going to be used in this way it is often advisable to use the option to clip the plot to trim away extra white space, or else use GsView to calculate a new BoundingBox in a transparent subsidiary plot by transforming ps into eps.

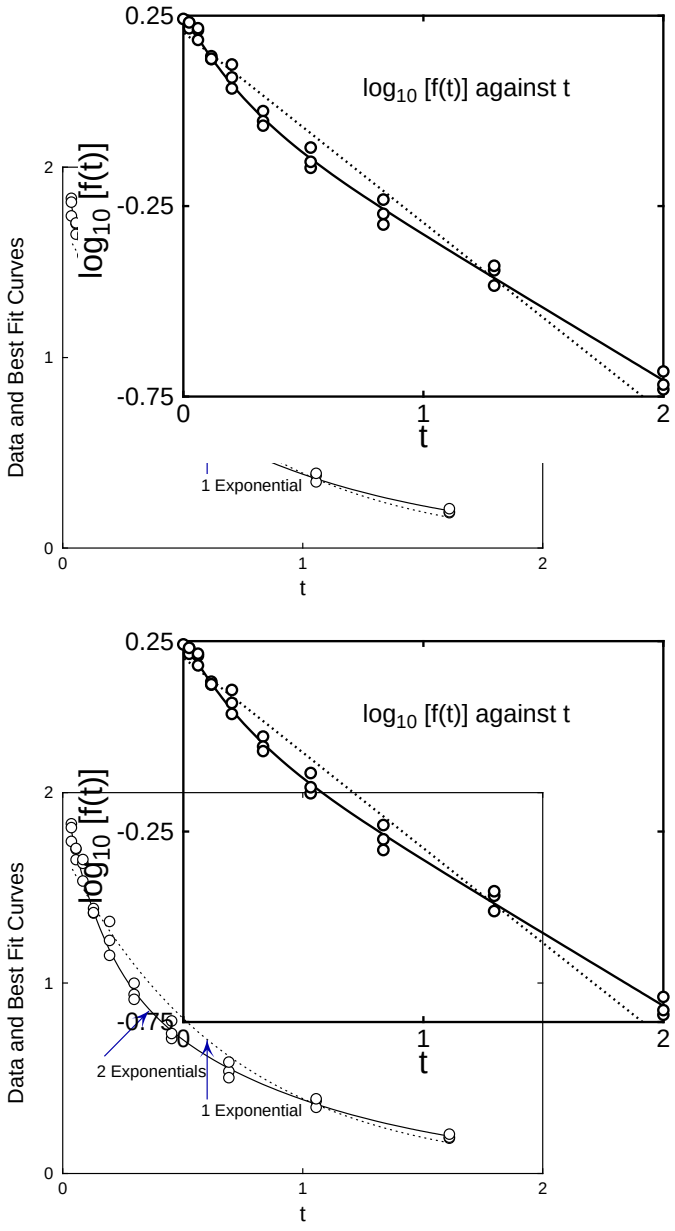
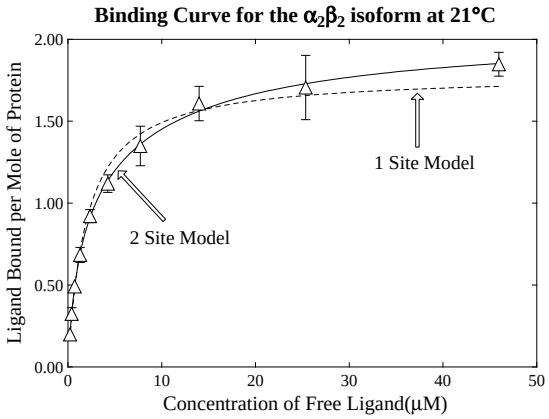
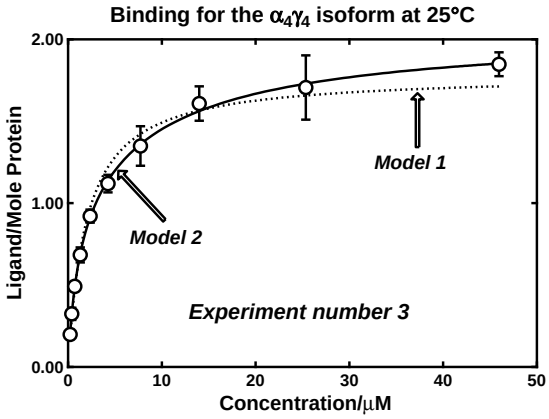


Figure 8: Insets 2: Opaque and tranparent backgrounds in insets

14.4 Editing Simfit PostScript files

One of the unique features of SIMFIT PostScript files is that the format is designed to make retrospective editing easy. A typical example of when this could be useful would be when a graph needs to be changed for some reason. Typically an experimentalist might have many plots stored as .eps files and want to alter one for publication or presentation. SIMFIT users are strongly recommended to save all their plots as .ps or .eps files, so that they can be altered in the way to be described. Even if you do not have a PostScript printer it is still best to save as .ps, then use GSview/Ghostscript to print or transform into another graphics format. Consider these next two figures, showing how a graph can be transformed by simple editing in a text editor, e.g. NOTEPAD.





This type of editing should always be done if you want to use one figure as a reduced size inset figure inside another, or when making a slide, otherwise the SIMFIT default line thickness will be too thin. Note that most of the editing to be described below can actually be done at the stage of creating the file, or by using program EDITPS. In this hypothetical example, we shall suppose that the experimentalist had realized that the title referred to the wrong isoform and temperature, and also wanted to add extra detail, but simplify the graph in order to make a slide using thicker lines and a bolder font. In the following sections the editing required to transform the SIMFIT example file `simfig1.ps` will be discussed, following a preliminary warning.

14.4.1 Warning about editing PostScript files

In the first place the technique to be described can only be done with SIMFIT PostScript files, because the format was developed to facilitate the sort of editing that scientists frequently need to perform. Secondly, it must be realized that PostScript files must conform to a very strict set of rules. If you violate these rules, then GSview/Ghostscript will warn you and indicate the fault. Unfortunately, if you do not understand PostScript, the warning will be meaningless. So here are some rules that you must keep in mind when editing.

- ❏ Always keep a backup copy at each successful stage of the editing.
- ❏ All text after a single percentage sign % to the line end is ignored in PostScript.
- ❏ Parentheses must always be balanced as in (figure 1(a)) not as in (figure 1(a).
- ❏ Fonts must be spelled correctly, e.g. Helvetica-Bold and not helveticabold.
- ❏ Character strings for displaying must have underneath them a vector index string of EXACTLY the same length.
- ❏ When introducing non-keyboard characters each octal code represents one byte.
- ❏ The meaning of symbols and line types depends on the function, e.g. `da` means dashed line while `do` means dotted line.

A review of the PostScript colours, fonts and conventions is also in the `w_readme` files. In the next sections it will be assumed that are running SimFIT and have a renamed copy of `simfig1.ps` in your text editor (e.g. notepad), and after each edit you will view the result using GSview/Ghostscript. Any errors reported when you try to view the edited file will be due to violation of a PostScript convention. The most usual one is to edit a text string without correctly altering the index below it to have exactly the same number of characters.

14.4.2 The percent-hash escape sequence

Later versions of SimFIT create PostScript files that can be edited by a stretch, clip, slide procedure, which relies on each line containing coordinates being identified by a comment line starting with %#. All text extending to the right from the first character of this sequence can safely be ignored and is suppressed for clarity in the following examples.

14.4.3 Changing line thickness and plot size

The following text will be observed in the original `simfig1.ps` file.

```
72.00  252.00 translate    0.07    0.07 scale    0.00 rotate
   11.00 setlinewidth 0 setlinecap 0 setlinejoin [] 0 setdash
    2.50 setmiterlimit
```

The postfix argument for setlinewidth alters the line width globally. In other words, altering this number by a factor will alter all the linewidths in the figure by this factor, irrespective on any changes in relative line thicknesses set when the file was created. The translate, scale and rotate are obvious, but perhaps best done by program EDITPS. Here is the same text edited to increase the line thickness by a factor of two and a half.

```
72.00  252.00 translate    0.07    0.07 scale    0.00 rotate
   27.50 setlinewidth 0 setlinecap 0 setlinejoin [] 0 setdash
    2.50 setmiterlimit
```

14.4.4 Changing PostScript fonts

In general the Times-Roman fonts may be preferred for readability in diagrams to be included in books, while Helvetica may look better in scientific publications. For making slides it is usually preferable to use Helvetica-Bold. Of course any PostScript fonts can be used, but in the next example we see how to change the fonts in `simfig1.ps` to achieve the effect illustrated.

```
/ti-font /Times-Bold  D%plot-title
/xl-font /Times-Roman D%x-legend
/yl-font /Times-Roman D%y-legend
/zl-font /Times-Roman D%z-legend
/tc-font /Times-Roman D%text centred
/td-font /Times-Roman D%text down
/tl-font /Times-Roman D%text left to right
/tr-font /Times-Roman D%text right to left
/ty-font /Times-Roman D%text right y-mid
/tz-font /Times-Roman D%text left  y-mid
```

The notation is obvious, the use indicated being clear from the comment text following the percentage sign % at each definition, denoted by a D. This is the editing needed to bring about the font substitution.

```
/ti-font /Helvetica-Bold D%plot-title
/xl-font /Helvetica-Bold D%x-legend
/yl-font /Helvetica-Bold D%y-legend
/zl-font /Helvetica-Bold D%z-legend
```

```
/tc-font /Helvetica-Bold D%text centred
/td-font /Helvetica-Bold D%text down
/tl-font /Helvetica-Bold D%text left to right
/tr-font /Helvetica-Bold D%text right to left
/ty-font /Helvetica-Bold D%text right y-mid
/tz-font /Helvetica-Bold D%text left  y-mid
```

Observing the scheme for colours (just before the fonts in the file) and text sizes (following the font definitions) will make it obvious how to change colours and text sizes.

14.4.5 Changing title and legends

Observe the declaration for the title and legends in the original file.

```
(Binding Curve for the a2b2 isoform at 21@C) 3514 4502 ti
(000000000000000000000006161000000000000060) fx
(Concentration of Free Ligand(lM)) 3514 191 xl
(000000000000000000000000000300) fx
(Ligand Bound per Mole of Protein) 388 2491 yl
(000000000000000000000000000000) fx
```

Note that, for each of the text strings displayed, there is a corresponding index of font substitutions. For example a zero prints the letter in the original font, a one denotes a subscript, while a six denotes bold maths. Since the allowed number of index keys is open-ended, the number of potential font substitutions is enormous. You can have any accent on any letter, for instance. This is the editing required to change the text. However, note that the positions of the text do not need to be changed, the font display functions work out the correct position to centre the text string.

```
(Binding for the a4c4 isoform at 25@C) 3514 4502 ti
(00000000000000006161000000000000060) fx
(Concentration/lM) 3514 191 xl
(0000000000000030) fx
(Ligand/Mole Protein) 388 2491 yl
(0000000000000000000) fx
```

Note that the \ character is an escape character in PostScript so, if you want to have something like an unbalanced parenthesis, as in `Figure 1 a)` you would have to write `Figure 1a\)`. When you create a PostScript file from SIMFIT it will prevent you from writing a text string that violates PostScript conventions but, when you are editing, you must make sure yourself that the conventions are not violated, e.g. use `c:\\simfit` instead of `c:\simfit`.

14.4.6 Deleting graphical objects

It is very easy to delete any text or graphical object by simply inserting a percentage sign % at the start of the line to be suppressed. In this way an experimental observation can be temporarily suppressed, but it is still in the file to be restored later if required. Here is the PostScript code for the notation on the left hand vertical, i.e. *y* axis in the file `simfig1.ps`.

```
910 1581 958 1581 li
6118 1581 6070 1581 li
(0.50) 862 1581 ty
(0000) fx
910 2491 958 2491 li
6118 2491 6070 2491 li
(1.00) 862 2491 ty
```

```
(0000) fx
910 3401 958 3401 li
6118 3401 6070 3401 li
(1.50) 862 3401 ty
(0000) fx
```

This is the text, after suppressing the tick marks and notation for $y = 0.5$ and $y = 1.5$ by inserting a percentage sign. Note that the index must also be suppressed as well as the text string.

```
%910 1581 958 1581 li
 %6118 1581 6070 1581 li
 %(0.50) 862 1581 ty
 %(0000) fx
910 2491 958 2491 li
6118 2491 6070 2491 li
(1.00) 862 2491 ty
(0000) fx
 %910 3401 958 3401 li
 %6118 3401 6070 3401 li
 %(1.50) 862 3401 ty
 %(0000) fx
```

14.4.7 Changing line and symbol types

This is simply a matter of substituting the desired line or plotting symbol key.

```
Lines    : li (normal) da (dashed) do (dotted) dd (dashed dotted) pl (polyline)
Circles  : ce (empty)  ch (half)   cf(full)
Triangles: te (empty)  th (half)   tf (full)
Squares  : se (empty)  sh (half)   sf (full)
Diamonds : de (empty)  dh (half)   df (full)
Signs    : ad (add)    mi (minus)  cr (cross)  as (asterisk)
```

Here is the original text for the dashed line and empty triangles.

```
5697 3788 120 da
933 1032 72 te
951 1261 72 te
984 1566 73 te
1045 1916 72 te
1155 2346 72 te
1353 2708 73 te
1714 3125 72 te
2367 3597 72 te
3551 3775 72 te
5697 4033 72 te
```

Here is the text edited for a dotted line and empty circles.

```
5697 3788 120 do
933 1032 72 ce
951 1261 72 ce
984 1566 73 ce
1045 1916 72 ce
1155 2346 72 ce
1353 2708 73 ce
```

```
1714 3125 72 ce
2367 3597 72 ce
3551 3775 72 ce
5697 4033 72 ce
```

14.4.8 Adding extra text

Here is the original extra text section.

```
/font /Times-Roman D /size   216 D
GS font F size S   4313  2874 M    0 rotate
(1 Site Model)
(000000000000) fx
/font /Times-Roman D /size   216 D
GS font F size S   1597  2035 M    0 rotate
(2 Site Model)
(000000000000) fx
```

Here is the above text after changing the font.

```
/font /Helvetica-BoldOblique D /size   216 D
GS font F size S   4313  2874 M    0 rotate
(Model 1)
(0000000) fx
/font /Helvetica-BoldOblique D /size   216 D
GS font F size S   1597  2035 M    0 rotate
(Model 2)
(0000000) fx
```

Here is the additional code required to add another label to the plot.

```
/font /Helvetica-BoldOblique D /size   240 D
GS font F size S   2250  1200 M    0 rotate
(Experiment number 3)
(00000000000000000000) fx
```

14.4.9 Changing colors

The definition of the 72 colors c0 to c71 in terms of red, green, blue components is in the file header and any color can be edited. Note that the colors c0 and c15 should be defined as black and white respectively, and perhaps the main editing involved would be to replace all appearances of colors other than c0 (and c15 for the background) by c0 in order to transform a colored file into a monochrome file, as these tend to look better in Word documents and Power Point. For instance, changing

```
c4
(Survival Analysis) 3195 4467 ti%#title
(00000000000000000) fx
```

into

```
c0
(Survival Analysis) 3195 4467 ti%#title
(00000000000000000) fx
```

would change a red title into a black title.

14.5 Standard fonts

All PostScript printers have a basic set of 35 fonts and it can be safely assumed that graphics using these fonts will display in GSview/Ghostscript and print on all except the most primitive PostScript printers. Of course there may be a wealth of other fonts available. The Times and Helvetica fonts are well known, and the monospaced Courier family of typewriter fonts are sometimes convenient for tables.

Times-Roman
!"#$%&’()*+,-./0123456789:;<=>?@
ABCDEFGHIJKLMNOPQRSTUVWXYZ[]^_‘
abcdefghijklmnopqrstuvwxyz{|}~

Times-Bold
!"#$%&’()*+,-./0123456789:;<=>?@
ABCDEFGHIJKLMNOPQRSTUVWXYZ[]^_‘
abcdefghijklmnopqrstuvwxyz{|}~

Times-Italic
!"#$%&’()+,-./0123456789:;<=>?@*
ABCDEFGHIJKLMNOPQRSTUVWXYZ[]^_‘
abcdefghijklmnopqrstuvwxyz{|}~

Times-BoldItalic
!"#$%&’()*+,-./0123456789:;<=>?@
ABCDEFGHIJKLMNOPQRSTUVWXYZ[]^_‘
abcdefghijklmnopqrstuvwxyz{|}~

Helvetica
!"#$%&’()*+,-./0123456789:;<=>?@
ABCDEFGHIJKLMNOPQRSTUVWXYZ[]^_‘
abcdefghijklmnopqrstuvwxyz{|}~

Helvetica-Bold
!"#$%&’()*+,-./0123456789:;<=>?@
ABCDEFGHIJKLMNOPQRSTUVWXYZ[]^_‘
abcdefghijklmnopqrstuvwxyz{|}~

Helvetica-Oblique
!"#$%&’()+,-./0123456789:;<=>?@*
ABCDEFGHIJKLMNOPQRSTUVWXYZ[]^_‘
abcdefghijklmnopqrstuvwxyz{|}~

Helvetica-BoldOblique
!"#$%&’()*+,-./0123456789:;<=>?@
ABCDEFGHIJKLMNOPQRSTUVWXYZ[]^_‘
abcdefghijklmnopqrstuvwxyz{|}~

14.5.1 Decorative fonts

Sometimes decorative or graphic fonts are required, such as pointing hands or scissors. It is easy to include such fonts using program Simplot, although the characters will be visible only if the plot is inspected using GSview/Ghostscript.

Symbol
!∀#∃%&∍()*+,−./0123456789:;<=>?≅
ΑΒΧΔΕΦΓΗΙϑΚΛΜΝΟΠΘΡΣΤΥςΩΞΨΖ[]⊥_
αβχδεφγηιφκλμνοπθρστυϖωξψζ{|}~

ZapfDingbats
✁✂✃✄✆✇✈✉☛☞✌✍✎✏✑✒✓✔✕✖✗✘✙✚✛✜✝✞✟✠✡✢✣✤✥✦✧★☆✪✫✬✭✮✯✰✱✲✳✴✵✶✷✸✹✺✻✼✽✾✿❀❁❂❃❄❅❆❇❈❉❊●❍■❑❒❐❒▲▼◆❖◗❙❚❵❛❜❝❞

ZapfChancery-MediumItalic
!'#$%&'()+,-./0123456789:;<=>?@*
ABCDEFGHIJKLMNOPQRSTUVWXYZ[]^_‘
abcdefghijklmnopqrstuvwxyz{|}~

Some extra characters in Times, Helvetica, etc.
æ(361)•(267)†(262)‡(263)¡(241)ƒ(246)œ(372)¿(277)˚(312)§(247)£(243)

Some extra characters in Symbol
∠(320)〈(341)〉(361)≈(273)↔(253)⇔(333)⇐(334)⇒(336)←(254)→(256)|(174)
⊗(304)⊕(305)°(260)÷(270)∈(316)…(274)∅(306)≡(272)ƒ(246)∇(321)≥(263)
∞(245)∫(362)≤(243)×(264)≠(271)∏(325)∂(266)±(261)√(326)∑(345)∪(310)

14.5.2 Plotting characters outside the keyboard set

To use characters outside the keyboard set you have to use the corresponding octal codes. Note that these codes represent just one byte in PostScript so, in this special case, four string characters need only one key character. For example, such codes as `\277` for an upside down question mark in standard encoding, or `\326` for a square root sign in Symbol, only need one index key. You might wonder why, if Simplot can put any accent on any character and there are maths and bold maths fonts, you would ever want alternative encodings, like the ISOLatin1Encoding. This is because the ISOLatin1Encoding allows you to use specially formed accented letters, which are more accurately proportioned than those generated by program Simplot by adding the accent as a second over-printing character, e.g. using `\361` for n tilde is more professional than overprinting.
All the characters present in the coding vectors to be shown next can be used by program Simplot, as well as a special Maths/Greek font and a vast number of accented letters and graphical objects, but several points must be remembered.

All letters can be displayed using GSview/Ghostscript and then Adobe Acrobat after distilling to pdf. Although substitutions can be made interactively from Simplot, you can also save a .eps file and edit it in a text editor. When using an octal code to introduce a non-keyboard character, only use one index key for the four character code. If you do not have a PostScript printer, save plots as .eps files and print from GSview/Ghostscript or transform into graphics files to include in documents.

Some useful codes follow, then by examples to clarify the subject. You will find it instructive to view `simfonts.ps` in the SIMFIT viewer and display it in GSview/Ghostcsript.

14.5.3 The StandardEncoding Vector

octal	0	1	2	3	4	5	6	7
\00x								
\01x								
\02x								
\03x								
\04x		!	"	#	$	%	&	’
\05x	(	)	*	+	,	-	.	/
\06x	0	1	2	3	4	5	6	7
\07x	8	9	:	;	<	=	>	?
\10x	@	A	B	C	D	E	F	G
\11x	H	I	J	K	L	M	N	O
\12x	P	Q	R	S	T	U	V	W
\13x	X	Y	Z	[	\	]	^	_
\14x	‘	a	b	c	d	e	f	g
\15x	h	i	j	k	l	m	n	o
\16x	p	q	r	s	t	u	v	w
\17x	x	y	z	{	|	}	~	
\20x								
\21x								
\22x								
\23x								
\24x		¡	¢	£	⁄	¥	*ƒ*	§
\25x	¤	'	“	«	‹	›	fi	fl
\26x		–	†	‡	·		¶	•
\27x	‚	„	”	»	…	‰		¿
\30x		`	´	ˆ	˜	¯	˘	˙
\31x	¨		°	¸		˝	˛	ˇ
\32x	—							
\33x								
\34x		Æ		ª				
\35x	Ł	Ø	Œ	º				
\36x		æ				ı		
\37x	ł	ø	œ	ß				

14.5.4 The ISOLatin1Encoding Vector

octal	0	1	2	3	4	5	6	7
\00x								
\01x								
\02x								
\03x								
\04x		!	"	#	$	%	&	’
\05x	(	)	*	+	,	−	.	/
\06x	0	1	2	3	4	5	6	7
\07x	8	9	:	;	<	=	>	?
\10x	@	A	B	C	D	E	F	G
\11x	H	I	J	K	L	M	N	O
\12x	P	Q	R	S	T	U	V	W
\13x	X	Y	Z	[	\	]	^	_
\14x	‘	a	b	c	d	e	f	g
\15x	h	i	j	k	l	m	n	o
\16x	p	q	r	s	t	u	v	w
\17x	x	y	z	{	|	}	~	
\20x								
\21x								
\22x	ı	`	´	ˆ	˜	¯	˘	˙
\23x	¨		°	¸		˝	˛	ˇ
\24x		¡	¢	£	¤	¥	¦	§
\25x	¨	©	ª	«	¬	-	®	¯
\26x	°	±	²	³	´	µ	¶	·
\27x	¸	¹	º	»	¼	½	¾	¿
\30x	À	Á	Â	Ã	Ä	Å	Æ	Ç
\31x	È	É	Ê	Ë	Ì	Í	Î	Ï
\32x	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×
\33x	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
\34x	à	á	â	ã	ä	å	æ	ç
\35x	è	é	ê	ë	ì	í	î	ï
\36x	ð	ñ	ò	ó	ô	õ	ö	÷
\37x	ø	ù	ú	û	ü	ý	þ	ÿ

14.5.5 The SymbolEncoding Vector

octal	0	1	2	3	4	5	6	7
\00x								
\01x								
\02x								
\03x								
\04x		!	∀	#	∃	%	&	∍
\05x	(	)	*	+	,	−	.	/
\06x	0	1	2	3	4	5	6	7
\07x	8	9	:	;	<	=	>	?
\10x	≅	A	B	X	Δ	E	Φ	Γ
\11x	H	I	ϑ	K	Λ	M	N	O
\12x	Π	Θ	P	Σ	T	Y	ς	Ω
\13x	Ξ	Ψ	Z	[	∴	]	⊥	_
\14x	‾	α	β	χ	δ	ε	ϕ	γ
\15x	η	ι	φ	κ	λ	μ	ν	ο
\16x	π	θ	ρ	σ	τ	υ	ϖ	ω
\17x	ξ	ψ	ζ	{	|	}	~	
\20x								
\21x								
\22x								
\23x								
\24x		ϒ	′	≤	⁄	∞	*f*	♣
\25x	♦	♥	♠	↔	←	↑	→	↓
\26x	°	±	″	≥	×	∝	∂	•
\27x	÷	≠	≡	≈	…	|	—	↵
\30x	ℵ	ℑ	ℜ	℘	⊗	⊕	∅	∩
\31x	∪	⊃	⊇	⊄	⊂	⊆	∈	∉
\32x	∠	∇	®	©	™	∏	√	·
\33x	¬	∧	∨	⇔	⇐	⇑	⇒	⇓
\34x	◊	〈	®	©	™	∑	⎛	⎜
\35x	⎝	⎡	⎢	⎣	⎧	⎨	⎩	⎪
\36x		〉	⌠	⌡	⎪	⎫	⎭	⎪
\37x	⎭	⎤	⎥	⎦	⎫	⎬	⎭	

14.5.6 The ZapfDingbatsEncoding Vector

octal	0	1	2	3	4	5	6	7
\00x								
\01x								
\02x								
\03x								
\04x								
\05x								
\06x				✓	✔	✕	✖	✗
\07x	✘		✚	✛	✜	†	✞	✟
\10x	✠	✡	✢	✣	✤	✥	✦	✧
\11x	★	☆	✪	✫	✬	✭	✮	✯
\12x	✰	✶	✱	✲	✳	✴	✵	✷
\13x	✸	✹	✺	✻	✼	✽	✾	✿
\14x	❀	❁	❂	❃	❄	❅	❆	❇
\15x	❈	❉	❊	❋	●	❍	■	❑
\16x	❒	❑	❒	▲	▼	◆	❖	◗
\17x	❙	❚	❚	❛	❜	❝	❞	
\20x	❨	❩	❪	❫	❬	❭	❮	❯
\21x	❰	❱	❲	❳	❴	❵		
\22x								
\23x								
\24x		❡	❢	❣	❤	❥	❦	❧
\25x	♣	♦	♥	♠	①	②	③	④
\26x	⑤	⑥	⑦	⑧	⑨	⑩	❶	❷
\27x	❸	❹	❺	❻	❼	❽	❾	❿
\30x	①	②	③	④	⑤	⑥	⑦	⑧
\31x	⑨	⑩	❶	❷	❸	❹	❺	❻
\32x	❼	❽	❾	❿	➔	→	↔	↕
\33x	➘	➙	➚	→	➜	→	→	➠
\34x	➠	➡	➢	➢	➤	➥	➦	➧
\35x	➨	➩	➪	➫	➬	➭	➮	➯
\36x		➱	➲	➳	➴	➵	➶	➷
\37x	➸	➹	➺	➻	➼	➽	➾	

14.6 Simfit character display codes

0 Standard font
1 Standard font subscript
2 Standard font superscript
3 Maths/Greek
4 Maths/Greek subscript
5 Maths/Greek superscript
6 Bold Maths/Greek
7 ZapfDingbats (PostScript) Wingding (Windows)
8 ISOLatin1Encoding (PostScript), Standard (Windows, almost)
9 Special (PostScript) Wingding2 (Windows)
A Grave accent
B Acute accent
C Circumflex/Hat
D Tilde
E Macron/Bar/Overline
F Dieresis
G Maths/Greek-hat
H Maths/Greek-bar
I Bold maths/Greek-hat
J Bold Maths/Greek-bar
K Symbol font
L Bold Symbol font

You will need non-keyboard characters from the standard font for such characters as a double dagger (‡) or upside down question mark (¿), e.g. typing `\277` in a text string would generate the upside down question mark (¿) in the PostScript output. If you want to include a single backslash in a text string, use `\\`, and also cancel any unpaired parentheses using `\(` and `\)`. Try it in program SIMPLOT and it will then all make sense. The ISOLatin1Encoding vector is used for special characters, such as `\305` for Angstrom (Å), `\361` for n-tilde (ñ), or `\367` for the division sign (÷), and, apart from a few omissions, the standard Windows font is the same as the ISOLatin1Encoding. The Symbol and ZapfDingbats fonts are used for including special graphical characters like scissors or pointing hands in a text string.

A special font is reserved for PostScript experts who want to add their own character function. Note that, in a document with many graphs, the prologue can be cut out from all the graphs and sent to the printer just once at the start of the job. This compresses the PostScript file, saves memory and speeds up the printing. Examine the manuals source code for this technique.

If you type four character octal codes as character strings for plotting non-keyboard characters, you do not have to worry about adjusting the character display codes, program SIMPLOT will make the necessary corrections. The only time you have to be careful about the length of character display code vectors is when editing in a text editor. If in doubt, just pad the character display code vector with question marks until it is the same length as the character string.

14.7 editps text formatting commands

Program **editps** uses the SimFIT convention for text formatting characters within included SimFIT .eps files but, because this is rather cumbersome, a simplified set of formatting commands is available within **editps** whenever you want to add text, or even create PostScript files containing text only. The idea of these formatting commands is to allow you to introduce superscripts, subscripts, accented letters, maths, dashed lines or plotting symbols into PostScript text files, or into collage titles, captions, or legends, using only ASCII text controls. To use a formatting command you simply introduce the command into the text enclosed in curly brackets as in: {raise}, {lower}, {newline}, and so on. If {anything} is a recognized command then it will be executed when the .eps file is created. Otherwise the literal string argument, i.e. anything, will be printed with no inter-word space. Note that no {commands} add interword spaces, so this provides a mechanism to build up long character strings and also control spacing; use {anything} to print anything with no trailing inter-word space, or use { } to introduce an inter-word space character. To introduce spaces for tabbing, for instance, just use {newline}{ }start-of-tabbing, with the number of spaces required inside the { }. Note that the commands are both spelling and case sensitive, so, for instance, {21}{degree}{C} will indicate the temperature intended, but {21}{degrees}{C} will print as 21degreesC while {21}{Degree}{C} will produce 21DegreeC.

14.7.1 Special text formatting commands, e.g. left

{left} ... use {left} to print a {
{right} ... use {right} to print a }
{%!command} ... use {%!command} to issue command as raw PostScript

The construction {%!command} should only be used if you understand PostScript. It provides PostScript programmers with the power to create special effects. For example {%!1 0 0 setrgbcolor}, will change the font colour to red, and {%!0 0 1 setrgbcolor} will make it blue, while {%!2 setlinewidth} will double line thickness. In fact, with this feature, it is possible to add almost any conceivable textual or graphical objects to an existing .eps file.

14.7.2 Coordinate text formatting commands, e.g. raise

{raise} ... use {raise} to create a superscript or restore after {lower}
{lower} ... use {lower} to create a subscript or restore after {raise}
{increase} ... use {increase} to increase font size by 1 point
{decrease} ... use {decrease} to decrease font size by 1 point
{expand} ... use {expand} to expand inter-line spacing by 1 point
{contract} ... use {contract} to contract inter-line spacing by 1 point

14.7.3 Currency text formatting commands, e.g. dollar

{dollar} $ {sterling} £ {yen} Y

14.7.4 Maths text formatting commands, e.g. divide

{divide} ÷ {multiply} × {plusminus} ±

14.7.5 Scientific units text formatting commands, e.g. Angstrom

{Angstrom} Å {degree} ∘ {micron} μ

14.7.6 Font text formatting commands, e.g. roman

{roman} {bold} {italic} {helvetica}
{helveticabold} {helveticaoblique} {symbol} {zapfchancery}

{zapfdingbats} {isolatin1}

Note that you can use octal codes to get extra-keyboard characters, and the character selected will depend on whether the StandardEncoding or IOSLatin1Encoding is current. For instance, \ 361 will locate an {ae} character if the StandardEncoding Encoding Vector is current, but it will locate a {ñ} character if the ISOLatin1Encoding Encoding Vector is current, i.e. the command {isolatin1} has been used previously. The command {isolatin1} will install the ISOLatin1Encoding Vector as the current Encoding Vector until it is cancelled by any font command, such as {roman}, or by any shortcut command such as {ntilde} or {alpha}. For this reason, {isolatin1} should only be used for characters where shortcuts like {ntilde} are not available.

14.7.7 Poor man's bold text formatting command, e.g. pmb?

The command {pmb?} will use the same technique of overprinting as used by the Knuth TEX macro to render the argument, that is ? in this case, in bold face font, where ? can be a letter or an octal code. This is most useful when printing a boldface character from a font that only exists in standard typeface. For example, {pmbb} will print a boldface letter b in the current font then restore the current font, while {symbol}{pmbb}{roman} will print a boldface beta then restore roman font. Again, {pmb\ 243} will print a boldface pound sign.

14.7.8 Punctuation text formatting commands, e.g. dagger

{dagger} † {daggerdbl} ‡ {paragraph} ¶ {subsection} §
{questiondown} ¿

14.7.9 Letters and accents text formatting commands, e.g. Aacute

{Aacute} Á	{agrave} à	{aacute} á	{acircumflex} â
{atilde} ã	{adieresis} ä	{aring} å	{ae} æ
{ccedilla} ç	{egrave} è	{eacute} é	{ecircumflex} ê
{edieresis} ë	{igrave} ì	{iacute} í	{icircumflex} î
{idieresis} ï	{ntilde} ñ	{ograve} ò	{oacute} ó
{ocircumflex} ô	{otilde} õ	{odieresis} ö	{ugrave} ù
{uacute} ú	{ucircumflex} û	{udieresis} ü	

All the other special letters can be printed using {isolatin1} (say just once at the start of the text) then using the octal codes, for instance {isolatin1}{\ 303} will print an upper case ntilde.

14.7.10 Greek text formatting commands, e.g. alpha

{alpha} α	{beta} β	{chi} χ	{delta} δ
{epsilon} ϵ	{phi} ϕ	{gamma} γ	{eta} η
{kappa} κ	{lambda} λ	{mu} μ	{nu} ν
{pi} π	{theta} θ	{rho} ρ	{sigma} σ
{tau} τ	{omega} ω	{psi} ψ	

All the other characters in the Symbol font can be printed by installing Symbol font, supplying the octal code, then restoring the font, as in {symbol}{\ 245}{roman} which will print infinity, then restore Times Roman font.

14.7.11 Line and Symbol text formatting commands, e.g. ce

{li} = line
{da} = dashed line
{do} = dotted line
{dd} = dashed dotted line
{ce}, {ch}, {cf} = circle (empty, half filled, filled)

{te}, {th}, {tf} = triangle (empty, half filled, filled)
{se}, {sh}, {sf} = square (empty, half filled, filled)
{de}, {dh}, {df} = diamond (empty, half filled, filled)
These line and symbol formatting commands can be used to add information panels to legends, titles, etc. to identify plotting symbols.

14.7.12 Examples of text formatting commands

{TGF}{beta}{lower}{1}{raise} is involved
TGFβ_1 is involved

y = {x}{raise}{2}{lower} + 2
$y = x^2 + 2$

The temperature was {21}{degree}{C}
The temperature was 21°C

{pi}{r}{raise}{decrease}{2}{increase}{lower} is the area of a circle
πr^2 is the area of a circle

The {alpha}{lower}{2}{raise}{beta}{lower}{2}{raise} isoform
The $\alpha_2\beta_2$ isoform

{[Ca}{raise}{decrease}{++}{increase}{lower}{]} = {2}{mu}{M}
$[Ca^{++}] = 2\mu M$

14.8 Scaling, rotating, and stretching

One of the features SIMFIT provides for EPS PostScript files is the ability to manipulate the metric features of the files as they are created, or retrospectively using program **editps**. The following procedures, for instance, are frequently required.

- Changing the aspect ratio and clipping;
- Scaling, rotating, and shearing;
- Adjusting fonts and line thicknesses if graphs are enlarged or reduced; and
- Stretching the white space between items plotted without altering the aspect ratios of characters or symbols.

This is possible because of two special features in SIMFIT PostScript files.

1. The BoundingBox and clipping coordinates, as in the typical header section shown next.

```
%!PS-Adobe-3.0 EPSF-3.0
%%BoundingBox:    72   252   520   588
%%Creator: Simfit Version 7.2.8 (simfit.org.uk)
%%Title: colours=72/ISOLatin1Encoding/Accents/special/PSfrag/dict=300
%%CreationDate: Saturday, 15 April 2017
%%EndComments
%
%Start of SIMFIT PostScript file
%
save %save current state before clipping, etc.
    70   250   522   250   522   590    70   590%#clipping
newpath moveto lineto lineto lineto closepath clip newpath
   72.00  252.00 translate    0.07    0.07 scale    0.00 rotate%#portrait
   12.00 setlinewidth 0 setlinecap 1 setlinejoin [] 0 setdash
    2.50 setmiterlimit
%
```

   The BoundingBox holds the overall dimensions of the EPS file in PostScript units (1/72 inches) while the rest defines the clipping and rotating parameters.

2. The special sections indicated by %#, as in the typical data section shown below.

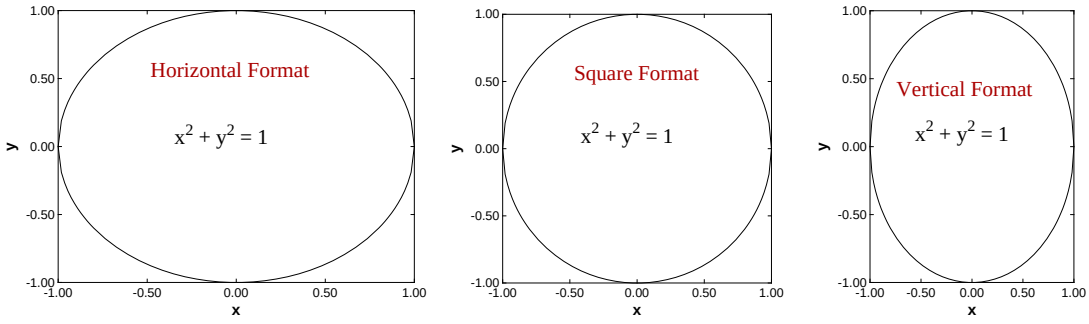
```
c0
1070 671 59 ce%#2
3514 2443 59 ce%#2
5959 4215 58 ce%#2
```

   Text following any % sign are taken as comments in PostScript files, so the parameters following the %# are specific SIMFIT stretching factors to alter the preceding coordinates. These stretch white space and alter the starting coordinates for symbols, lines, and character strings but not the inter-word separation, so preserving the aspect ratio of fonts.

Of course the brave may wish to alter these themselves using a text editor but, as this is very tedious and error-prone, the SIMFIT package provides functions to perform the editing

14.8.1 Alternative sizes, shapes and clipping

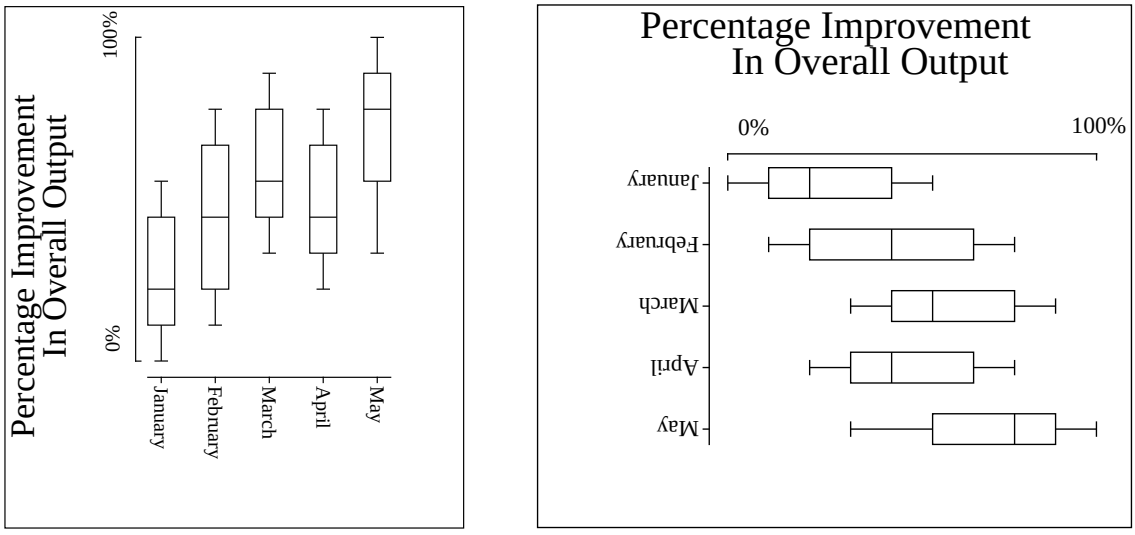
Plots can have horizontal, square or vertical format as in the next figure, and user-defined clipping schemes can be used. After clipping, SIMFIT adds a standard BoundingBox so all plots with the same clipping scheme will have the same absolute size but, when **GSview/Ghostscript** transforms ps into eps, it clips individual files to the boundary of white space and the desirable property of equal dimensions will be lost.



There is also a stretched format for long horizontal ribbon type graphs and a PostScript option to stretch white space without altering symbols and fonts, which is very useful with dense data sets such as dendrograms.

14.8.2 Rotated and re-scaled graphs

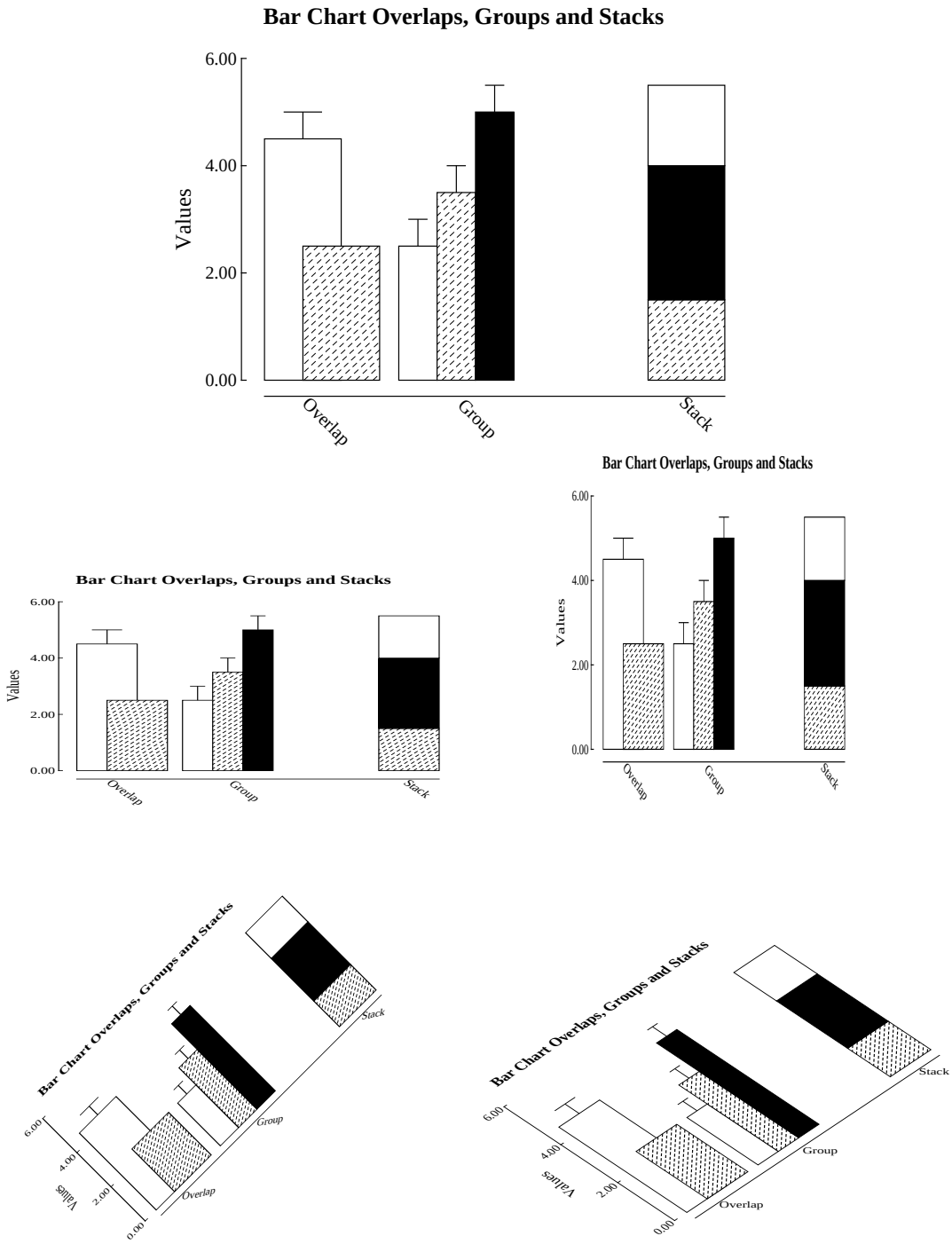
PostScript files can be read into **editps** which has options for re-sizing, re-scaling, editing, rotating, making collages, etc. In the next figure the box and whisker plot was turned on its side to generate a side-on barchart. To do this sort of thing you should learn how to browse a SIMFIT PostScript file in the SIMFIT viewer to read BoundingBox coordinates, in PostScript units of 72 to one inch, and calculate how much to translate, scale, rotate, etc.



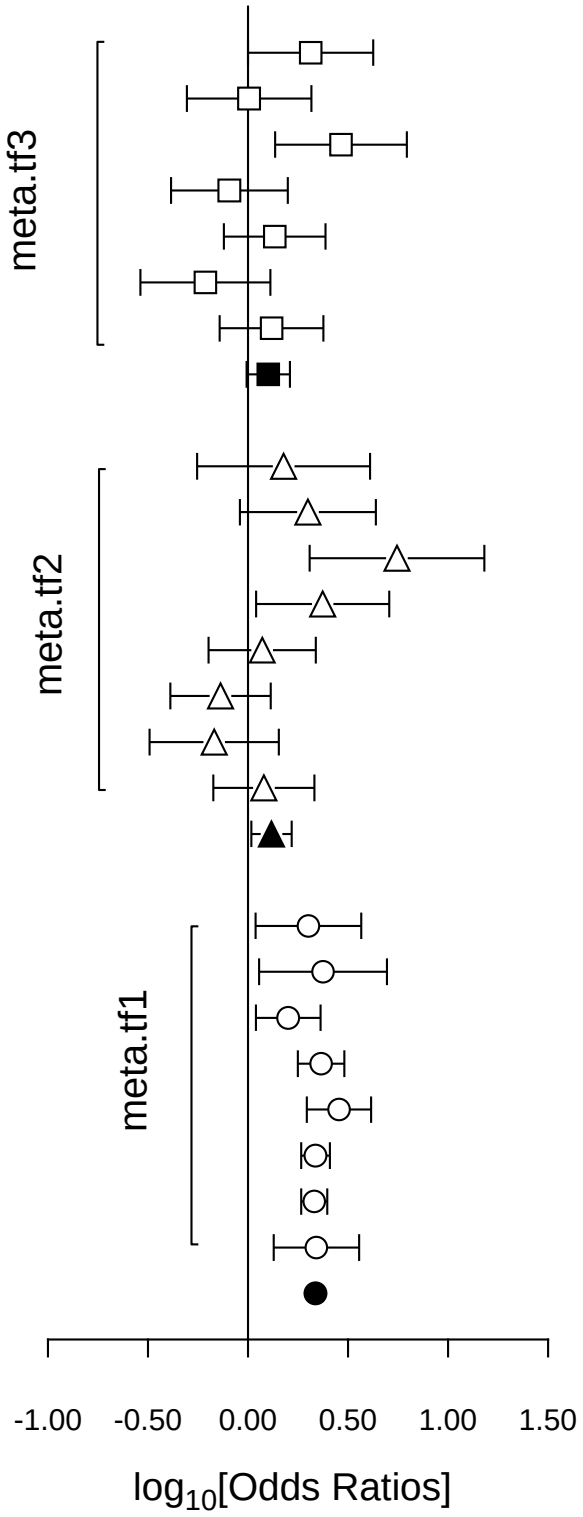
PostScript users should be warned that the special structure of SIMFIT PostScript files that allows extensive retrospective editing using **editps**, or more easily if you know how using a simple text editor like **notepad**, is lost if you read such graphs into a graphics editor program like Adobe Illustrator. Such programs start off by redrawing vector graphics files into their own conventions which are only machine readable.

14.8.3 Changed aspect ratios and shear transformations

The barchart in the figure below was scaled to make the X-axis longer than the Y-axis and vice-versa, but note how this type of differential scaling changes the aspect ratio as illustrated. Since rotation and scaling do not commute, the effect created depends on the order of concatenation of the transformation matrices. For instance, scaling then rotation cause shearing which can be used to generate 3-dimensional perspective effects as in the last sub-figure.



14.8.4 Plotting combined meta analysis results



It is often useful to plot Log-Odds-Ratios, so the creation of the adjacent figure will be outlined.

(1) **The data**
Test files `meta.tf1`, `meta.tf2`, and `meta.tf3` were analyzed in sequence using the SIMFIT Meta Analysis procedure. Note that, in these files, column 3 contains spacing coordinates so that data will be plotted consecutively.

(2) **The ASCII coordinate files**
During Meta Analysis, $100(1-\alpha)\%$ confidence limits on the Log-Odds-Ratio resulting from a 2 by 2 contingency tables with cell frequencies n_{ij} can be constructed from the approximation $\hat{e}$ where

$$\hat{e} = Z_{\alpha/2}\sqrt{\frac{1}{n_{11}}+\frac{1}{n_{12}}+\frac{1}{n_{21}}+\frac{1}{n_{22}}}.$$

When Log-Odds-Ratios with error bars are displayed, the overall values (shown as filled symbols) with error bars are also plotted with a x coordinate one less than smallest x value on the input file. For this figure, error bar coordinates were transferred into the project archive using the [Advanced] option to save ASCII coordinate files.

(3) **Creating the composite plot**
Program **simplot** was opened and the six error bar coordinate files were retrieved from the project archive. Experienced users would do this more easily using a library file of course. Reverse y-semilog transformation was selected, symbols were chosen, axes, title, and legends were edited, then half bracket hooks identifying the data were added as arrows and extra text.

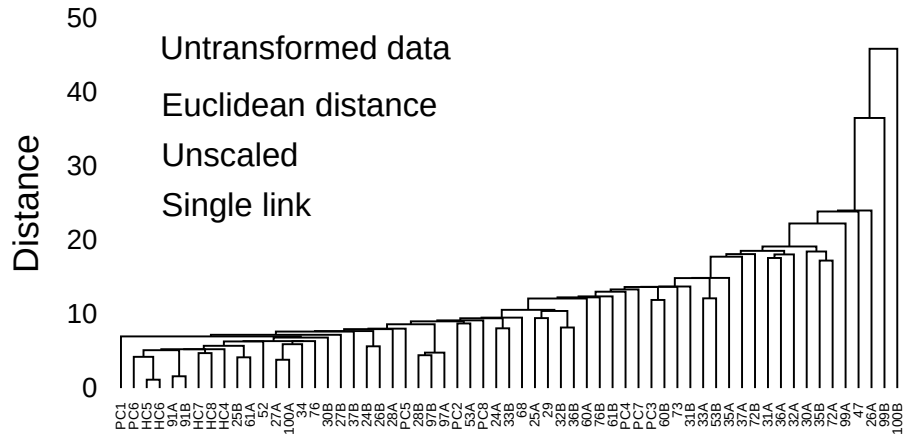
(4) **Creating the PostScript file**
Vertical format was chosen then, using the option to stretch PostScript files, the y coordinate was stretched by a factor of two.

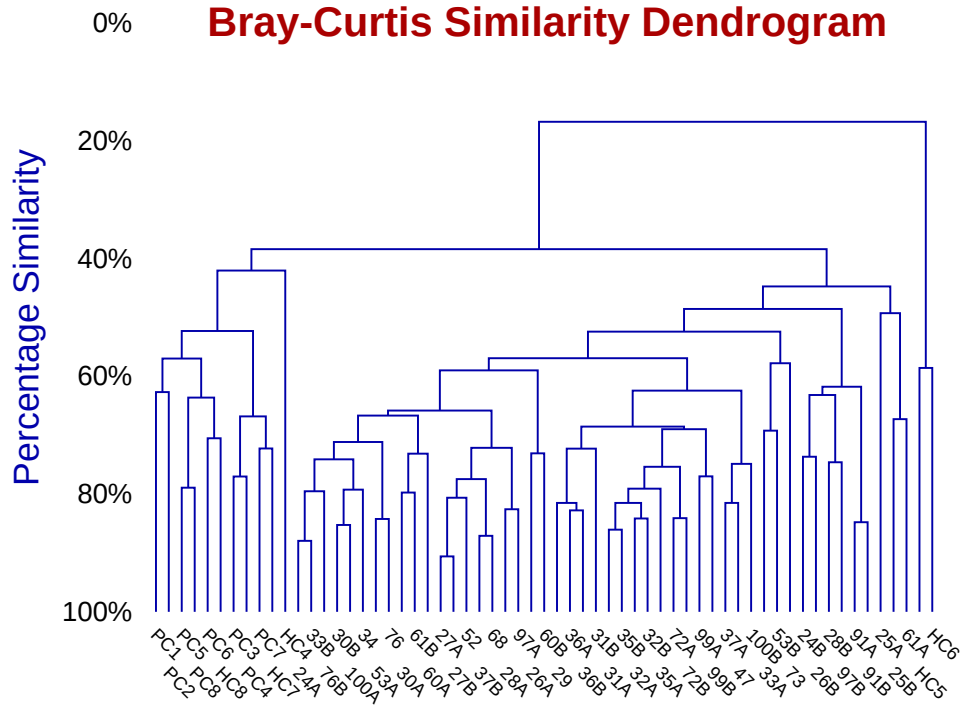
(5) **Editing the PostScript file**
To create the final PostScript file for LaTeX a tighter bounding box was calculated using **gsview** then, using **notepad**, clipping coordinates at the top of the file were set equal to the BoundingBox coordinates, to suppress excess white space. This can also be done using the [Style] option to omit painting a white background, so that PostScript files are created with transparent backgrounds, i.e. no white space, and clipping is irrelevant.

14.8.5 Plotting dendrograms: standard format

Dendrogram shape is arbitrary in two ways; the x axis order is arbitrary as clusters can be rotated around any clustering distance leading to 2^{n-1} different orders, and the distance matrix depends on the settings used. For instance, a square root transformation, Bray-Curtis similarity, and a group average link generates the second dendrogram in this figure from the first. The y plotted are dissimilarities, while labels are $100 - y$, which should be remembered when changing the y axis range.

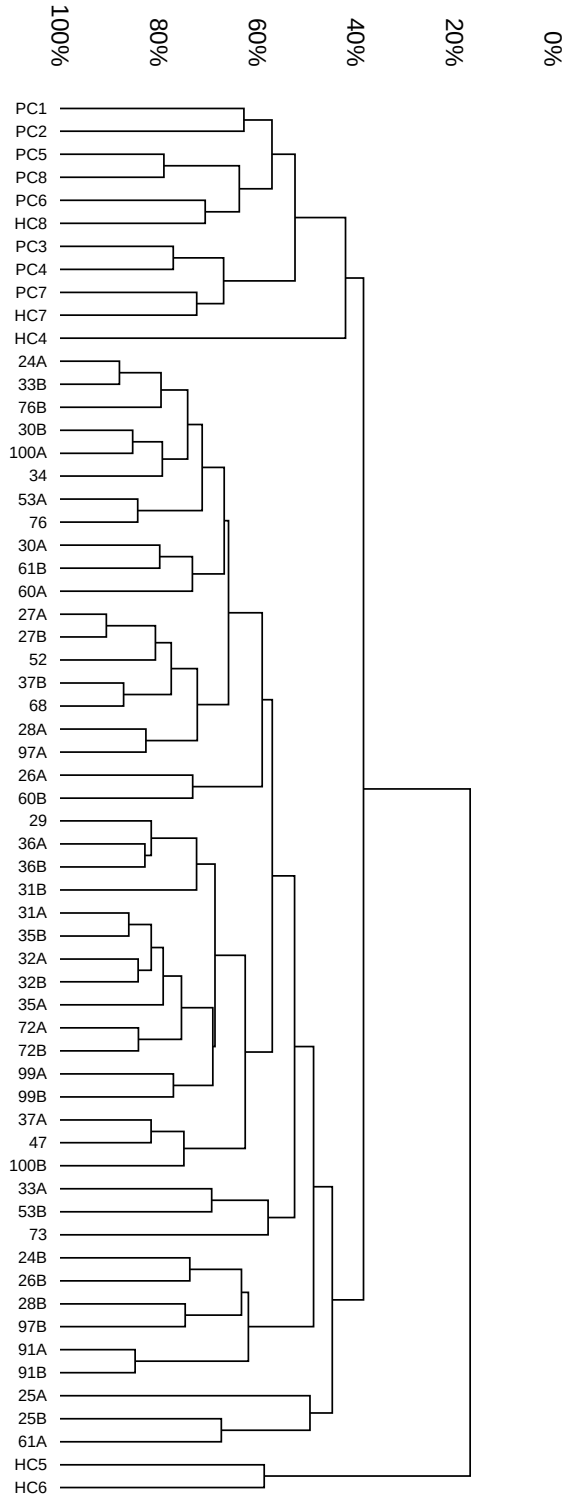
Users should not manipulate dendrogram parameters to create a dendrogram supporting some preconceived clustering scheme. You can set a label threshold and translation distance from the [X-axis] menu so that, if the number of labels exceeds the threshold, even numbered labels are translated, and font size is decreased.





14.8.6 Plotting dendrograms: stretched format

Sometimes dendrograms are more readable if the white space is stretched without distorting the labels. So SIMFIT PostScript graphs have a very useful feature: you can stretch or compress the white space between plotted lines and symbols without changing the line thickness, symbol size, or font size and aspect ratio. For instance, stretching, clipping and sliding procedures are valuable in graphs which are crowded due to overlapping symbols or labels, as in previous figures. If such dendrograms are stretched retrospectively using **editps**, the labels will not separate as the fonts will also be stretched so letters become ugly due to altered aspect ratios. SIMFIT can increase white space between symbols and labels while maintaining correct aspect ratios for the fonts in PostScript hardcopy and, to explain this, the creation of this stretrched figure will be described.

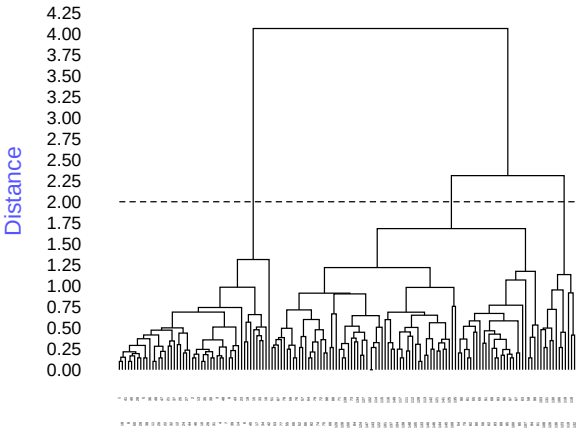


The title, legend and double x labeling were suppressed, and landscape mode with stretching, clipping and sliding was selected from the PostScript control using the [Shape] then [Landscape +] options, with an x stretching factor of two. Stretching increases the space between each symbol, or the start of each character string, arrow or other graphical object, but does not turn circles into ellipses or distort letters. As graphs are often stretched to print on several sheets of paper, sub-sections of the graph can be clipped out, then the clipped sub-sections can be slid to the start of the original coordinate system to facilitate printing.

If stretch factors greater than two are used, legends tend to become detached from axes, and empty white space round the graph increases. To remedy the former complication, the default legends should be suppressed or replaced by more closely positioned legends while, to cure the later effect, GSview can be used to calculate new BoundingBox coordinates (by transforming .ps to .eps). If you select the option to plot an opaque background even when white (by mistake), you may then find it necessary to edit the resulting .eps file in a text editor to adjust the clipping coordinates (identified by %#clip in the .eps file) and background polygon filling coordinates (identified by %#pf in the .ps file) to trim away unwanted white background borders that are ignored by GSview when calculating BoundingBox coordinates. Another example of this technique is with meta analysis plots, where it is also pointed out that creating transparent backgrounds by suppressing the painting of a white background obviates the need to clip away extraneous white space.

14.8.7 Plotting dendrograms: subgroups

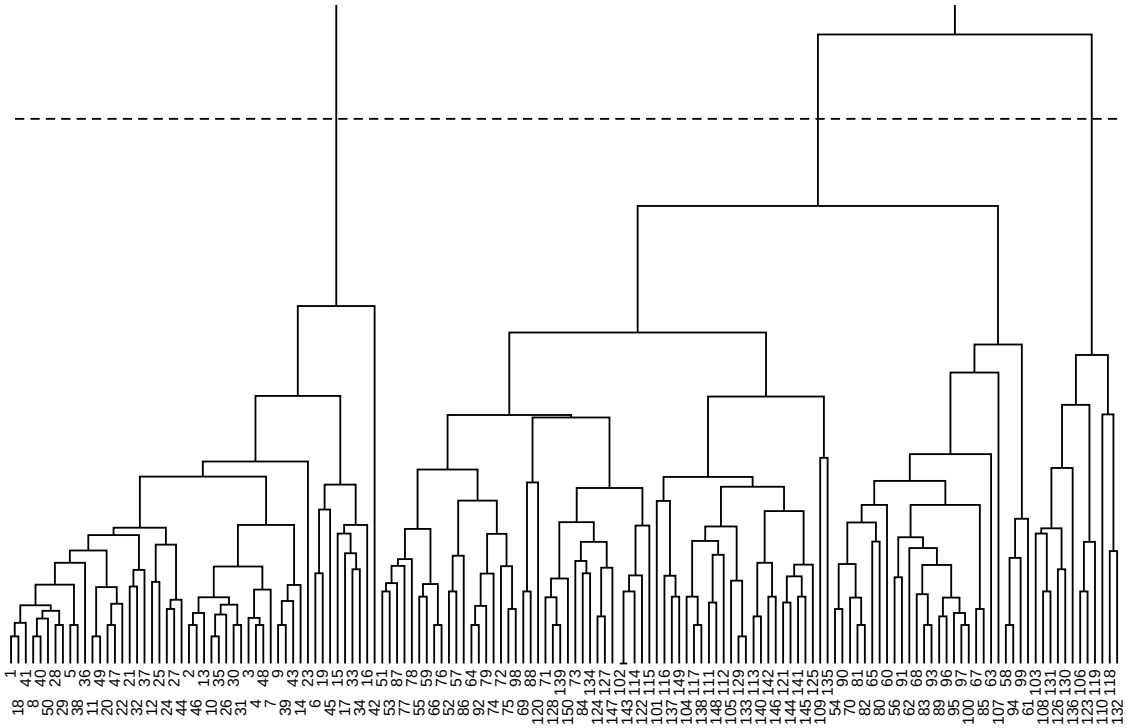
The procedures described can also be used to improve the readability of dendrograms where subgroups have been assigned by partial clustering. The next figure shows a graph from `iris.tf1` when three subgroups are requested, or a threshold is set corresponding to the horizontal dotted line. The figure was created by these steps.



First the title was suppressed, the y-axis range was changed to $(0, 4.25)$ with 18 tick marks, the (x, y) offset was canceled as this suppresses axis moving, the label font size was increased from 1 to 3, and the x-axis was translated to 0.8.
Then the PostScript stretch/slide/clip procedure was used with these parameters

$$x_{\text{stretch}} = 1.5$$
$$y_{\text{stretch}} = 2.0$$
$$x_{\text{clip}} = 0.15, 0.95$$
$$y_{\text{clip}} = 0.10, 0.60.$$

This generates the following graph.



Windows users without PostScript printing facilities must create a `*.eps` file using this technique, then use the SIMFIT procedures to create a graphics file they can use, e.g. `*.jpg`. Use of a larger font and increased x-stretching would be required to read the labels, of course.

14.9 PostScript specials

SimFIT PostScript files are designed to facilitate editing, and one important type of editing is to be able to specify text files, known as specials, that can modify the graph in an almost unlimited number of ways. This technique will now be described but, if you want to do it and you are not a PostScript programmer, do not even think about it; get somebody who has the necessary skill to do what you want. An example showing how to display a logo will be seen on page 797 and further details follow.

14.9.1 What specials can do

First of all, here are some examples of things you may wish to do with SimFIT PostScript files that would require specials.

- ❏ Replace the 35 standard fonts by special user-defined fonts.
- ❏ Add a logo to plots, e.g. a departmental heading for slides.
- ❏ Redefine the plotting symbols, line types, colours, fill styles, etc.
- ❏ Add new features, e.g. outline or shadowed fonts, or clipping to non-rectangular shapes.

When SimFIT PostScript files are created, a header subsection, called a prologue, is placed at the head of the file which contains all the definitions required to create the SimFIT dictionary. Specials can be added, as independent text files, to the files after these headings in order to re-define any existing functions, or even add new PostScript plotting instructions. The idea is is very simple; you can just modify the existing SimFIT dictionary, or even be ambitious and add completely new and arbitrary graphical objects.

14.9.2 The technique for defining specials

Any SimFIT PostScript file can be taken into a text editor in order to delete the existing header in order to save space in a large document, as done with the SimFIT manual, or else to paste in a special. However, this can also be done interactively by using the font option, accessible from the SimFIT PostScript interface. Since this mechanism is so powerful, and could easily lead to the PostScript graphics being permanently disabled by an incorrectly formatted special, SimFIT always assumes that no specials are installed. If you want to use a special, then you simply install the special and it will be active until it is de-selected or replaced by another special. Further details will be found in the on-line documentation and `w_readme` files, and examples of specials are distributed with the SimFIT package to illustrate the technique. You should observe the effect of the example specials before creating your own. Note that any files created with specials can easily be restored to the default configuration by cutting out the special. So it makes sense to format your specials like the SimFIT example specials `pspecial.1`, etc. to facilitate such retrospective editing. The use of specials is controlled by the file `pspecial.cfg` as now described. The first ten lines are Booleans indicating which of files 1 through 10 are to be included. The next ten lines are the file names containing the special code. There are ten SimFIT examples supplied, and it is suggested that line 1 of your specials should be in the style of these examples. You simply edit the file names in `pspecial.cfg` to install your own specials. The Booleans can be edited interactively from the advanced graphics PS/Fonts option. Note that any specials currently installed are flagged by the SimFIT program manager and specials only work in advanced graphics mode. In the event of problems with PostScript printing caused by specials, just delete `pspecial.cfg`. To summarise.

- ❏ Create the special you want to insert.
- ❏ Edit the file `psecial.cfg` in the SimFIT folder.
- ❏ Attach the special using the Postscript Font option.

14.9.3 Example codes for PostScript specials

To clarify the structure of SIMFIT PostScript specials, just consider the code for the first three examples distributed with the SIMFIT package. The file `psecial.1` simply adds a monochrome logo, the file `psecial.2` shows how to add color, while the file `psecial.3` makes more sweeping changes to the color scheme by reversing the definitions for black and white.

❏ **The PostScript special** `pspecial.1`

```
%file = pspecial.1: add monochrome simfit logo to plot
gsave
/printSIMFIT {0 0 moveto (SIMFIT) show} def
/Times-Italic findfont 300 scalefont setfont
300 4400 translate
.95 -.05 0
{setgray printSIMFIT -10 5 translate} for
1 1 1 setrgbcolor printSIMFIT
grestore
%end of pspecial.1
```

❏ **The PostScript special** `pspecial.2`

```
%file = pspecial.2: add yellow simfit logo to plot
gsave
/printSIMFIT {0 0 moveto (SIMFIT) show} def
/Times-Italic findfont 300 scalefont setfont
300 4400 translate
.95 -.05 0
{setgray printSIMFIT -10 5 translate} for
0 0 moveto (SIMFIT) true charpath gsave 1 1 0 setrgbcolor fill grestore
grestore
%end of pspecial.2
```

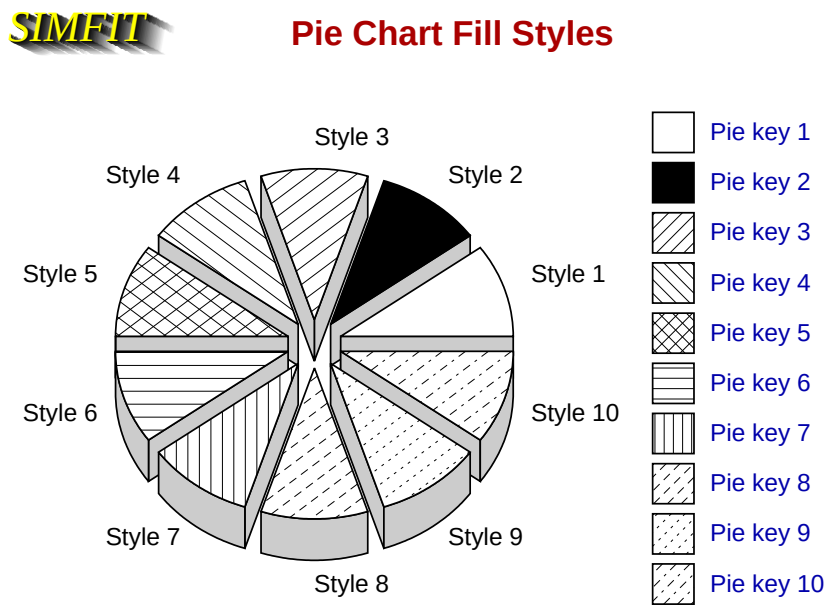
❏ **The PostScript special** `pspecial.3`

```
%file = pspecial.3: yellow-logo/blue-background/swap-black-and-white
/background{.5 .5 1 setrgbcolor}def
background
0 0 0 4790 6390 4790 6390 0 4 pf
/c0{1 1 1 setrgbcolor}def
/c15{0 0 0 setrgbcolor}def
/foreground{c0}def
gsave
/printSIMFIT {0 0 moveto (SIMFIT) show} def
/Times-Italic findfont 300 scalefont setfont
300 4400 translate
.95 -.05 0
{setgray printSIMFIT -10 5 translate} for
0 0 moveto (SIMFIT) true charpath gsave 1 1 0 setrgbcolor fill grestore
grestore
%end of pspecial.3
```

Remember, the effects of these specials are only visible in the PostScript files created by SIMFIT and not in any direct Windows quality hardcopy.

14.9.4 Example plots for PostScript specials

Figure 9 illustrates the end result of adding logos using such PostScript specials.



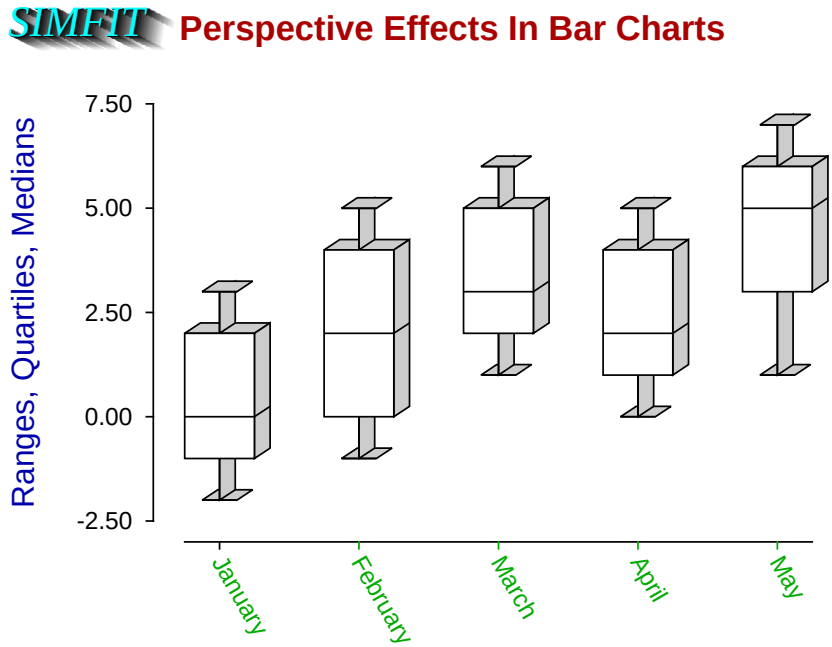


Figure 9: Using PostScript specials to add logos to plots

14.10 LaTeX options

14.10.1 Maths

You can add equations to graphs directly, but this will be a compromise, as specialized type setting techniques are required to display maths correctly. The LaTeX system is pre-eminent in the field of maths type-setting and the PSfrag system, as revised by David Carlisle and others, provides a simple way to add equations to SimFIT graphs. For figure 10, **makdat** generated a Normal cdf with $\mu = 0$ and $\sigma = 1$, then **simplot** created `cdf.eps` with the key `phi(x)`, which was then used by this stand-alone code to generate the figure, where the equation substitutes for the key. LaTeX PostScript users should be aware that SimFIT PostScript file format has been specially designed to be consistent with the PSfrag package but, if you want to then use GhostScript to create graphics file, say .png from .eps, the next section should be consulted.

```
\documentclass[dvips,12pt]{article}
   \usepackage{graphicx}
   \usepackage{psfrag}
   \pagestyle{empty}
\begin{document}
\large
\psfrag{phi(x)}{$\displaystyle
   \frac{1}{\sigma \sqrt{2\pi}}
   \int_{-\infty}^x \exp\left\{
  -\frac{1}{2} \left( \frac{t-\mu}{\sigma} \right)^2 \right\}\,dt$}
\mbox{\includegraphics[width=6.0in]{cdf.eps}}
\end{document}
```

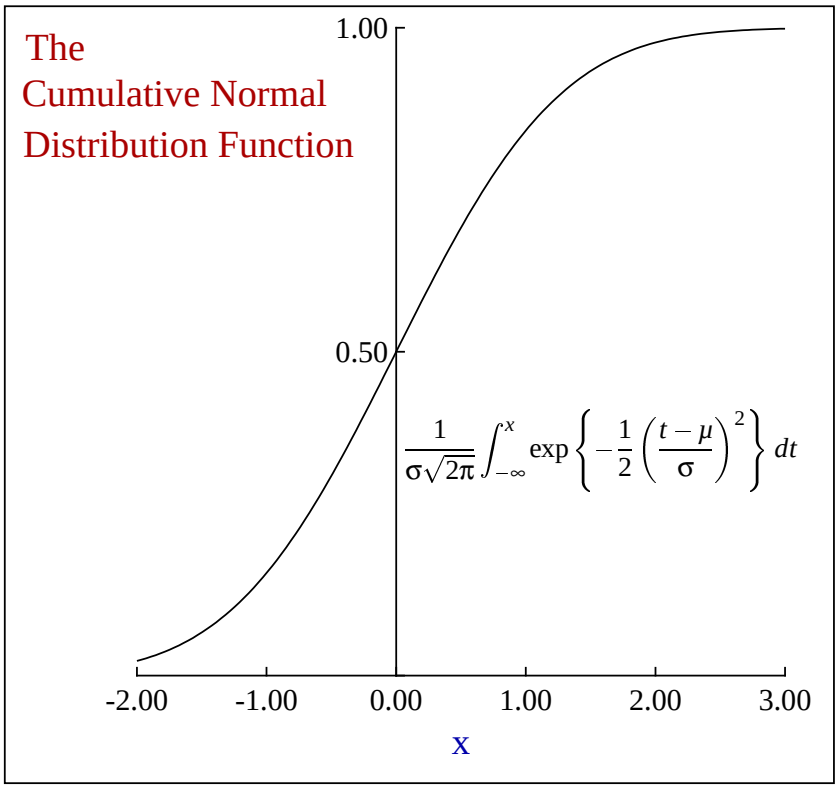


Figure 10: Plotting mathematical equations

14.10.2 Chemical Formulæ

LaTeX code, as below, is intended for document preparation and adds white space to the final .ps file. The easiest way round this complication is to add an outline box to the plot, as in figure 11. Then, after the .png file has been created, it can be input into, e.g., GIMP, for auto clipping to remove extraneous white space, followed by deletion of the outline box if required.

```
\documentclass[dvips,12pt]{article}
   \usepackage{graphicx}
   \usepackage{psfrag}
   \usepackage{carom}
   \pagestyle{empty}
\begin{document}
\psfrag{formula}
{\begin{picture}(3000,600)(0,0)
\thicklines
\put(0,0){\bzdrv{1==CH$_{2}$NH$_{2}$;4==CH$_{2}$N(Me)$_{2}$}}
\put(700,450){\vector(1,0){400}}
\put(820,550){[O]}
\put(1000,0){\bzdrv{1==CHO;4==CH$_{2}$N(Me)$_{2}$}}
\put(1650,400){+}
\put(1750,400){NH$_{3}$}
\put(2000,450){\vector(1,0){400}}
\put(2120,550){[O]}
\put(2300,0){\bzdrv{1==CO$_{2}$H;4==CH$_{2}$N(Me)$_{2}$}}
\end{picture}}
\mbox{\includegraphics{chemistry.eps}}
\end{document}
```

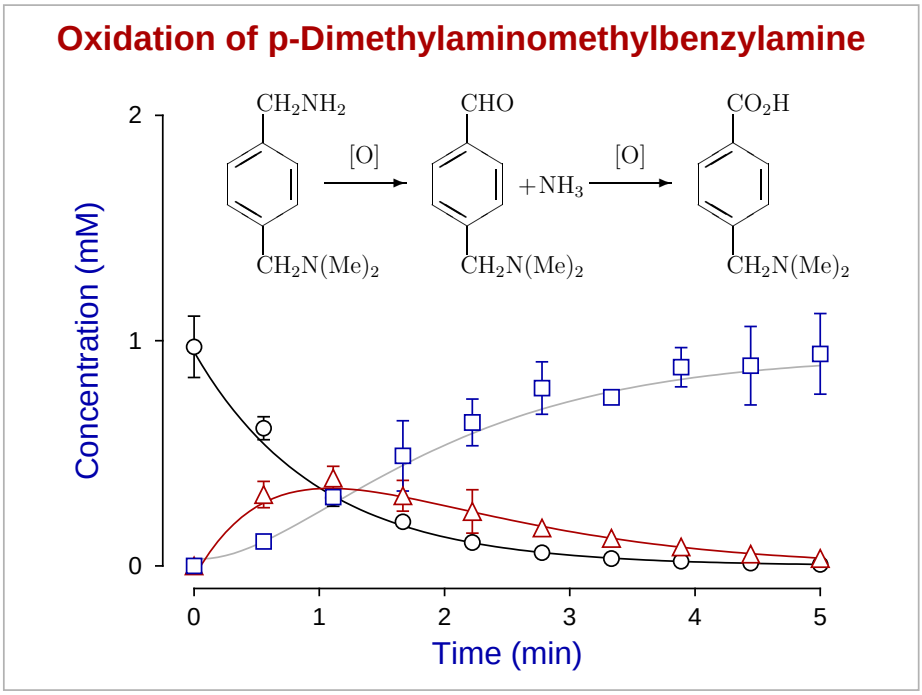


Figure 11: Plotting chemical structures

14.10.3 Composite graphs

The technique used to combine sub-graphs into a composite graph is easy. First use your drawing or painting program to save the figures of interest in the form of eps files. Then the SimFIT graphs and any component eps files are read into **editps** to move them and scale them until the desired effect is achieved. In figure 12, data were generated using **deqsol**, error was added using **adderr**, the simulated experimental data were fitted using **deqsol**, the plot was made using **simplot**, the chemical formulae and mathematical equations were generated using LaTeX and the final graph was composed using **editps**.

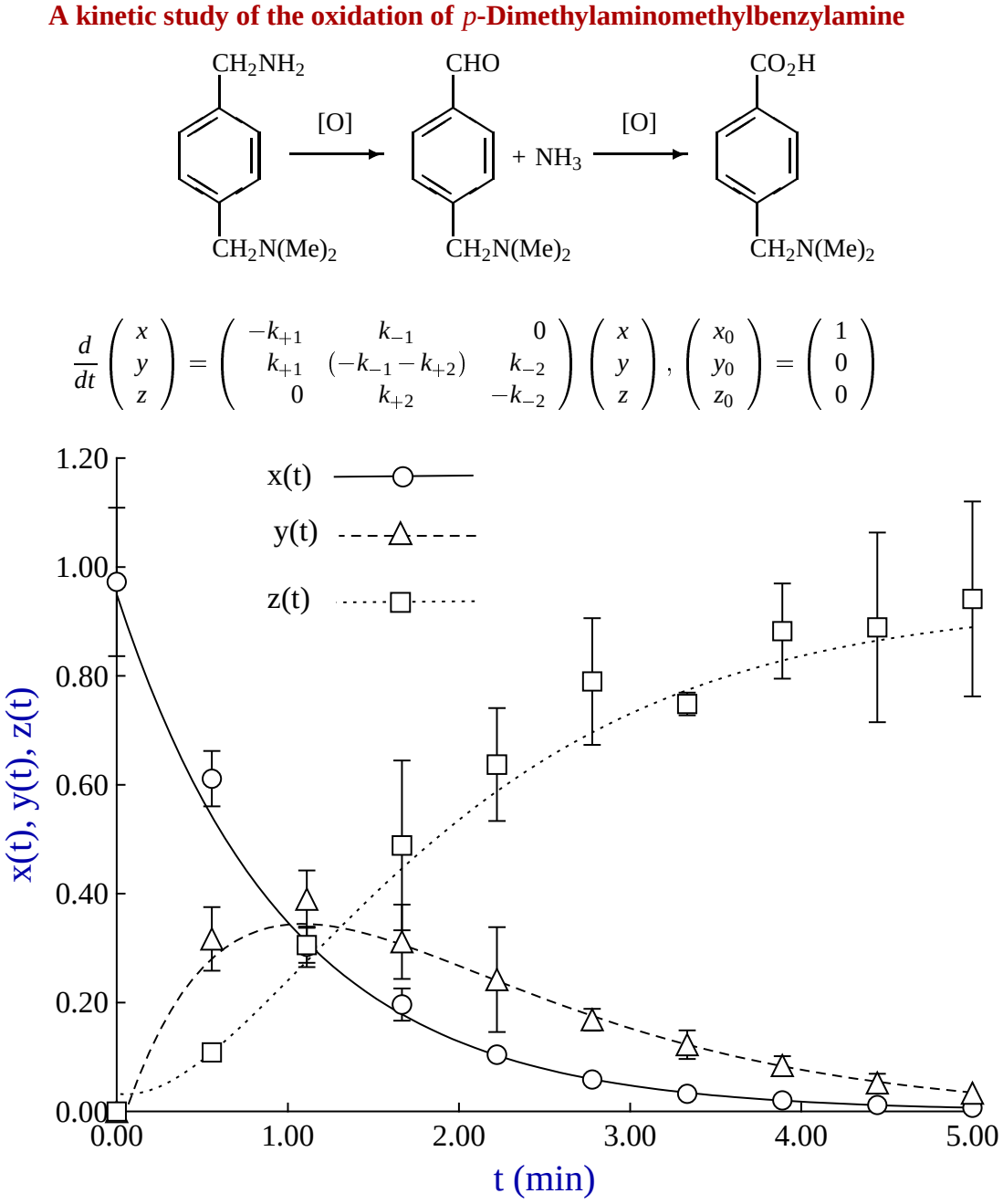


Figure 12: Chemical formulas

14.11 Creating collages, overlays, and insets

Sometimes it is useful to collect a numbers of graphics files together in order to create a single graphics file. This is particularly easy and valuable to do using SimFIT *.eps files since the manipulations are facilitated by the presence of BoundingBox coordinates in the header sections, also the composite graph will be compact and device–independent with full resolution for display or printing. The options available are provided by program **editps** and are as now listed.

1. **Strict collages**

   Here there must be a set of graphics files, say n, that all have the same size, orientation, and aspect ratio. For instance, the SimFIT default portrait setting. Because all files have the same dimensions then all that is required is to decide on the number of columns, say m, required for the collage. Clearly, for best results n should be a multiple of m, but this is purely for appearance. Once a collage has been composed there are three other options that can be used.

   (a) A label can be added to facilitate reference to the individual graphs.

   (b) Such labels would usually be a single number or letter but short labels can also be added, exercising care to view the resultant collage before finally saving.

   (c) A section of text can be added at the bottom of the collage to describe the items. This should be in simple text and will be transformed into PostScript before adding to the graph. In order to use subscripts, superscripts, font changes, or mathematical symbols, there is a special syntax described in the SimFIT reference manual `w_manual.pdf`, or the tutorial document `pscodes.pdf`.

2. **Freestyle collages**

   In this type of collage the graphs can be of any size, orientation, or aspect ratio, because the user is going to resize the graphs and then move them about individually. After selecting the files to be used, a display is created where the graphs are represented as numbered and colored rectangles. These can be selected at will, moved to a new positions, then enlarged or diminished as required. The collage can be viewed repeatedly during this procedure until a correct assembly has been obtained before saving the final composite file.

3. **Insets**

   This procedure is used when it is wished to insert one or more child graphs into a parent graph, often the same data plotted in alternative coordinates. There are three vital things to bear in mind when carrying out this procedure.

   (a) Font size and line thickness.

   Because inserted graphs are going to be reduced in size, it is important to consider saving the original child graphs graphs to be inserted using increased font and symbol size and line thickness. Otherwise the inserted graphs could be hard to read.

   (b) Clipping

   It is useful to consider clipping the graphs to be inserted to remove surrounding background which, by default, is added by SimFIT to create a margin. Alternatively, program **GSview** can be used to create a new BoundingBox. Actually it is very easy to edit the header of SimFIT PostScript files to alter the clipping and BoundingBox coordinates. Try it.

   (c) Background opacity.

   When a SimFIT *.eps file is created there is an option to make white backgrounds transparent or opaque. It is recommended to consider whether the inset graph is going to obliterate the section of the master graph it overlies, or if the master graph should remain visible.

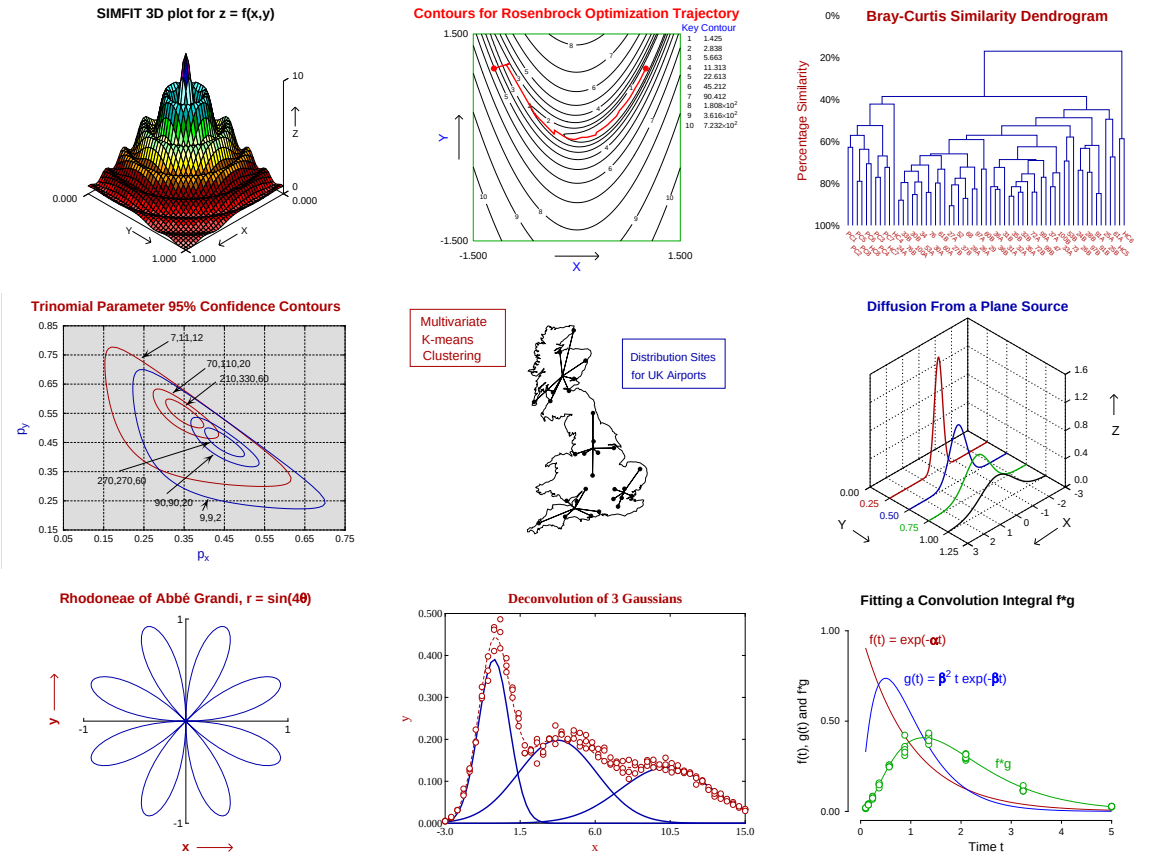
14.11.1 Example 1: Strict collages

The best way to input a collection of files is to create a library file for *.eps files using program **maklib**, and this was done to make the test library file `images.tfl` which contains this information.

```
Example of a EPS type library file
waves.eps
rosenbrock.eps
dendrogram.eps
trinom.eps
ukmap.eps
diffusion.eps
rose.eps
gauss3.eps
convolution.eps
```

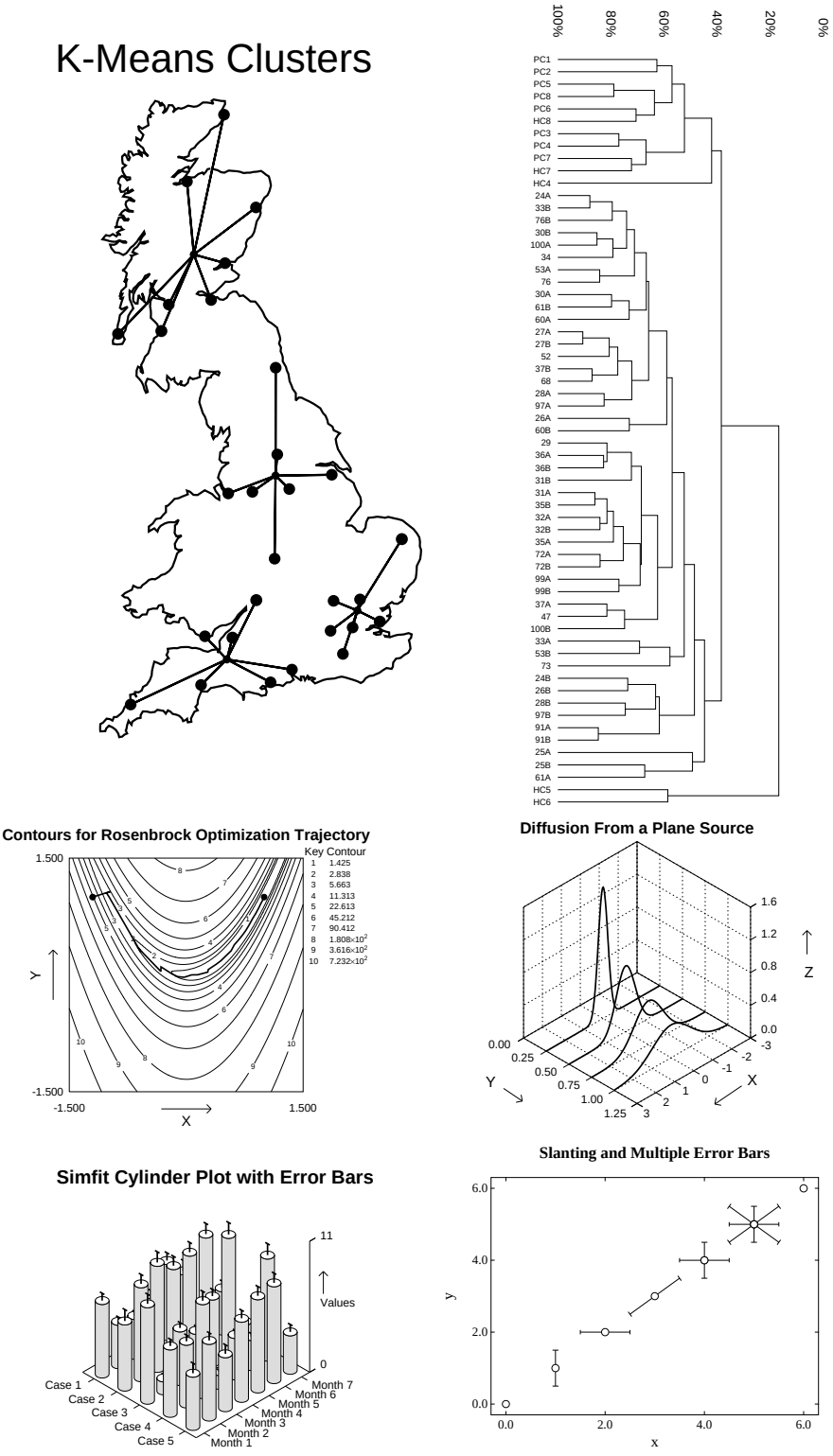
Note that the first line of a library file is an arbitrary title, while the next lines list the files in the order they will be used to make the collage. Normally these would be full paths, not local files names, and local file names are only used in this example because SIMFIT recognizes these short file names and can load the files.

Opening program **editps**, choosing a structured, i.e. strict collage using the [Demo] button on the file opening control, then requesting three columns gives the collage illustrated below.



14.11.2 Example 2: Freestyle collages

This figure illustrates a freestyle collage with different file sizes, aspect ratios and orientation.



14.11.3 Example 3: Insets

Figure 13 illustrates a special type of freestyle collage where a sub-graph is placed inside a parent graph. Sometimes it is best to enlarge the fonts and increase the line thicknesses when a sub-graph is going to be reduced in size in this way, and it is always important to remember the effects of opaque and transparent backgrounds that SimFIT allows.

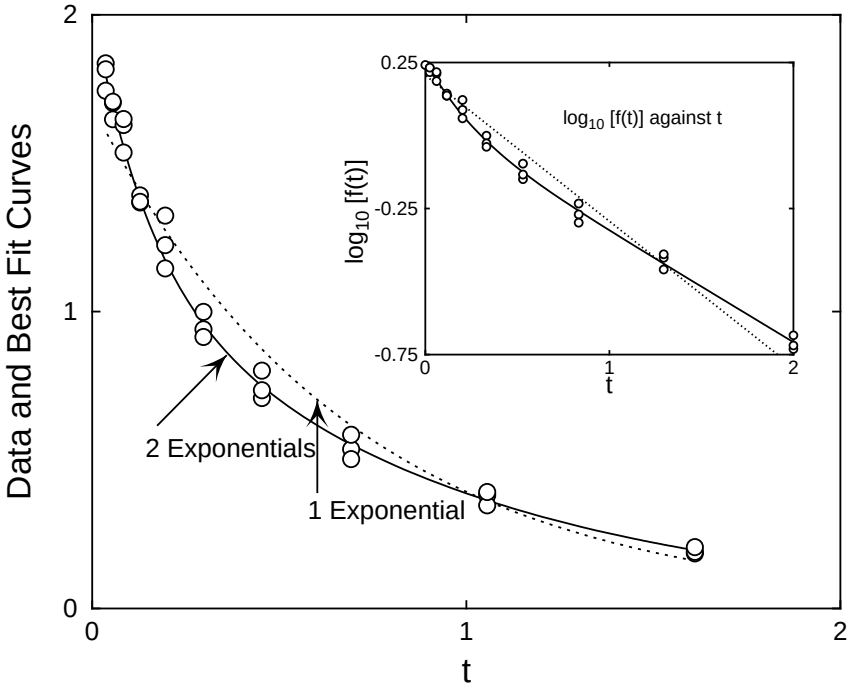


Figure 13: Subsidiary figures as insets

First of all the plots in figure 14 were created using **exfit** with test file `exfit.tf4` after fitting 1 exponential then 2 exponentials. Note that the line thickness and font size have been increased in the transformed plot as it is going to be reduced in the inset.

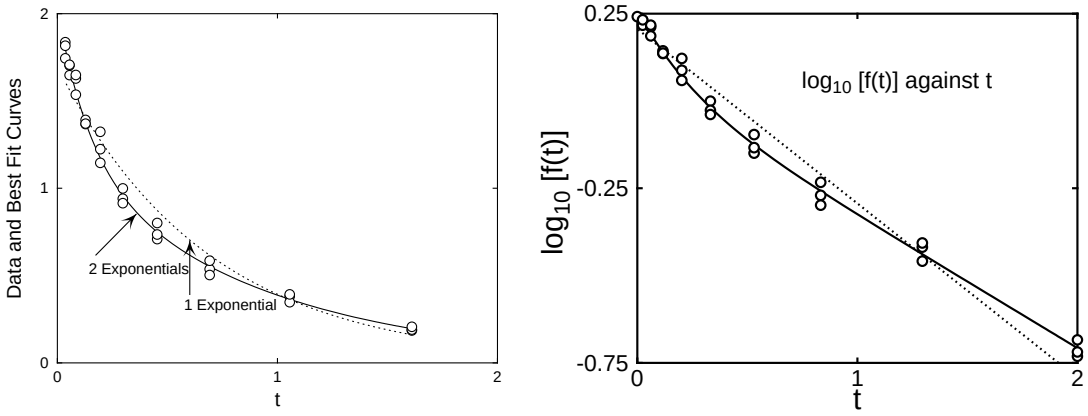


Figure 14: Insets 1: Exponential fitting and semilog transforms

Figure 15 was then created by **editps** using the option to create a freestyle collage. Note how, in the left hand plot the option to plot an opaque background even when white was selected and the transformed plot obscures the underlying main plot. In the right hand plot the option for a transparent background was used so that the main plot was not obscured. Both techniques are valuable when creating insets, and all that is now necessary to create figure 6 is to shrink the transformed plot and translate it to a more convenient location. A further point to note is that SimFIT plots have a border, which is obscuring more of the left hand main figure in figure 15 than seems necessary. When subsidiary figures are going to be used in this way it is often advisable to use the option to clip the plot to trim away extra white space, or else use program **GSview** to calculate a new BoundingBox in a transparent subsidiary plot by renaming *.eps as *.ps then transforming *.ps into *.eps.

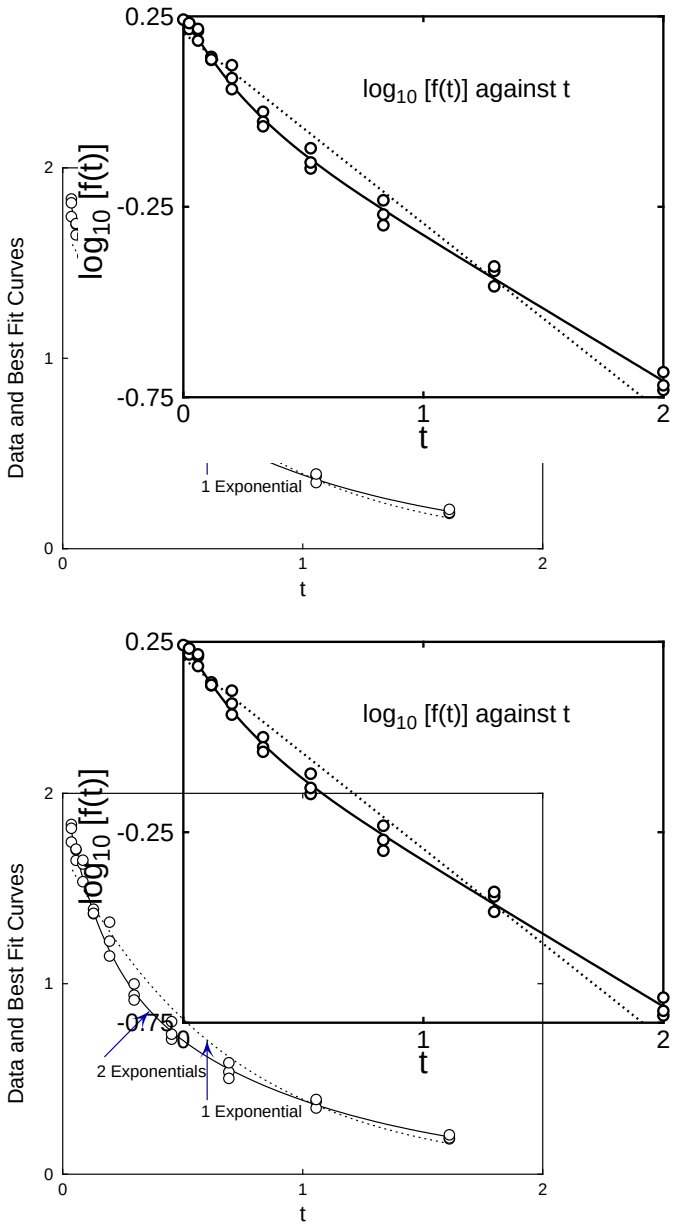


Figure 15: Insets 2: Opaque and transparent backgrounds in insets

14.11.4 Example 4: Adding labels to collages

Sometimes it is useful to add labels to identify sub-graphs in a collage or even to add extra text. Consider the effect of reading test file `editps.tfl` into program **editps** to create Figure 16.

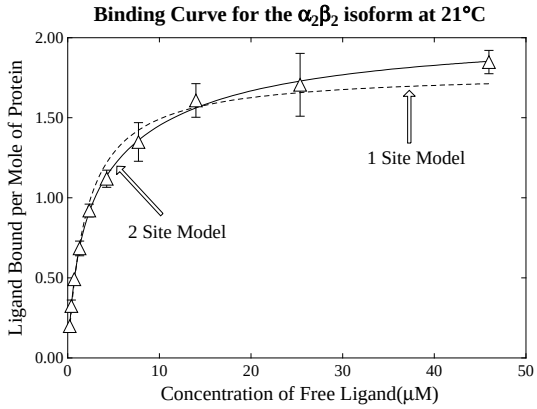


Figure A: Ligand Binding Data

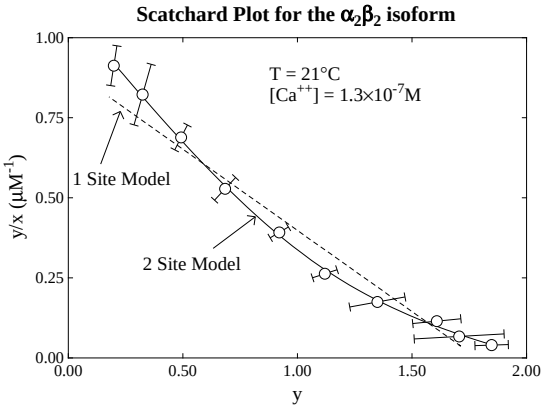


Figure B: Scatchard Transform

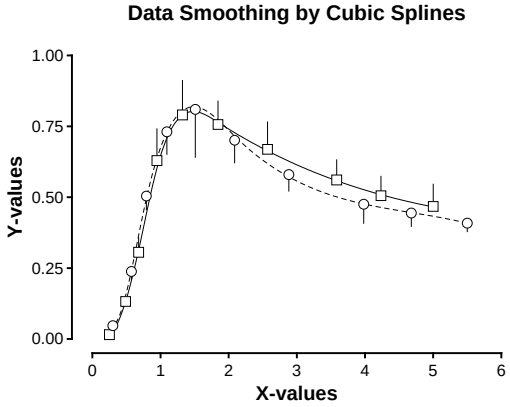


Figure C: Cubic Spline Smoothing

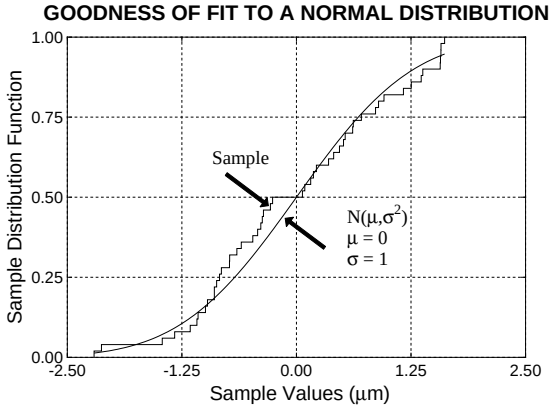


Figure D: Fitting a Normal Distribution

Demonstrating Labels and Additional Text

Figure A. Note maths subscripts in α_2 β_2
Figure B. Note the slanting error bars
Figure C. Note the confidence region indicators with no top or bottom bars
Figure D. Note maths superscript in N(μ, σ^2)

Figure 16: Adding labels and extra text

Once the collage has been created there are several options for adding the labels and extra text. Of course, as the composite file is to be created in PostScript then any commands added will have to be in PostScript. As a matter of fact raw Postscript can be added directly to the output file, but SimFIT also provides an interface to

define labels, titles, and text in any language except that special commands must be used to change fonts and introduce features such as maths, superscripts and subscripts. This feature which is demonstrated in Figure 16 will now be explained.
Once the collage has been assembled and the number of columns chosen, the following options are provided.

1. Defining labels as numbers, letters, or user–defined text.
2. Adding a title
3. Adding a text section

For instance, read in the test file `editps.tfl` into program **editps** then choose two columns, request to edit the text and cut and paste the next lines into the SIMFIT editor that has been opened.

```
{newline}
{newline}
{helveticabold}Demonstrating Labels and Additional Text{helvetica}{newline}
{newline}
Figure A. Note maths subscripts in {roman}{alpha}{lower}2{raise}{beta}{lower}2{raise}{newline}
{helvetica}Figure B. Note the slanting error bars{newline}
Figure C. Note the confidence region indicators with no top or bottom bars{newline}
Figure D. Note maths superscript in {roman}N\({mu},{sigma}{raise}2{lower}\)
```

This text will generate the title and additional text displayed in Figure 16. It uses the following commands that will be translated into Postscript.

- `{newline}` new line
- `{helveticabold}` bold font
- `{helvetica}` normal font
- `{roman}` roman font
- `{alpha}` α
- `{lower}` subscript
- `{raise}` superscript
- `{beta}` β
- `{mu}` μ
- `{sigma}` σ
- `{\(}` (
- `{\)}`)

Further details of the options available are described on page 785.

14.12 Editing PostScript colors

Graphics files for importing into documents, visual display, or printing can code for a large number of colors by using a color scheme, such as rgb, i.e., defining three values for red green blue intensities. Consider, for instance, the definition of black, red, green, blue and white, using three integers in the range 0 to 255 in Windows, the same range in Hexadecimal for the web, and the continuous range 0 to 1 in PostScript.

Colour	Windows	Hexadecimal	Postscript
Black	0, 0, 0	000000	0, 0, 0
Red	255, 0, 0	FF0000	1, 0, 0
Green	0, 255, 0	00FF00	0, 1, 0
Blue	0, 0, 255	0000FF	0, 0, 1
White	255, 255, 255	FFFFFF	1, 1, 1

However, such a wide range of colors can be confusing in scientific graphs where generally only a few strong colors are required for titles, legends, lines, and symbols, together with some subdued colors for backgrounds or plot borders. SIMFIT allows any possible color to be used for plotting by providing the following functionality.

1. **The color palette**

   72 colors are defined in `w_ps.cfg` in the `...Documents\Simfit\cfg` folder.

2. **Editing the colors**

   Users can edit any colors in this file to change defaults.

3. **The default colors**

   The SIMFIT scheme works as follows.

   (a) The first sixteen colors (0 to 15) correspond to the standard colors which would usually be sufficient for scientific graphs.
   It would not normally be necessary to edit these.

   (b) The next forty four colors (16 to 59) are variants of this scheme which includes a grey–scale selection.
   It would not normally be necessary to edit these.

   (c) The last twelve colors (60 to 71) can be adjusted by selecting the rgb numbers required, or using slider controls for color mixing available from the SIMFIT color palette control. These user-defined colors are provided so that twelve personally selected colors can be used.

   (d) In Windows hardcopy files such as *.png, *.jpg, *.emf, *.pdf, these colors can not be changed retrospectively.

   (e) In SIMFIT PostScript *.eps files any color can easily be changed retrospectively using a text editor, such as **notepad**.

4. **The PostScript header**

   At the start of a SIMFIT *.eps file is a list of all 72 colors defined as c0 to c71 using the PostScript command x y z setrgbcolor taking the three arguments x y z, which allows easy editing by simply inserting these color–changing commands at any point in the file.

5. **Changing colors in non–PostScript graphics files**

   It should be noted that, if graphics hardcopy is always archived as SIMFIT *.eps files, then the colors used in other types of files such as *.png, *.jpg, *.pdf, *.xps can be changed retrospectively, as can other features such as titles, legends, symbols, or line types, by simply editing the *.eps file in a text editor, followed by creating the type of hardcopy file required from within SIMFIT.

14.12.1 The sixteen standard colors (c0 to c15)

As all 72 colors are defined and can be manipulated in the same way it is enough to describe just the first 16 colors. So the next section shows how the standard colors are defined in the configuration file `w_ps.cfg`.

```
 0.0000, 0.0000, 0.0000
 0.0000, 0.0000, 0.6667
 0.0000, 0.6667, 0.0000
 0.0000, 0.6667, 0.6667
 0.6667, 0.0000, 0.0000
 0.6667, 0.0000, 0.6667
 0.6667, 0.3333, 0.0000
 0.7000, 0.7000, 0.7000
 0.4500, 0.4500, 0.4500
 0.3333, 0.3333, 1.0000
 0.3333, 1.0000, 0.3333
 0.3333, 1.0000, 1.0000
 1.0000, 0.3333, 0.3333
 1.0000, 0.3333, 1.0000
 1.0000, 1.0000, 0.3333
 1.0000, 1.0000, 1.0000
```

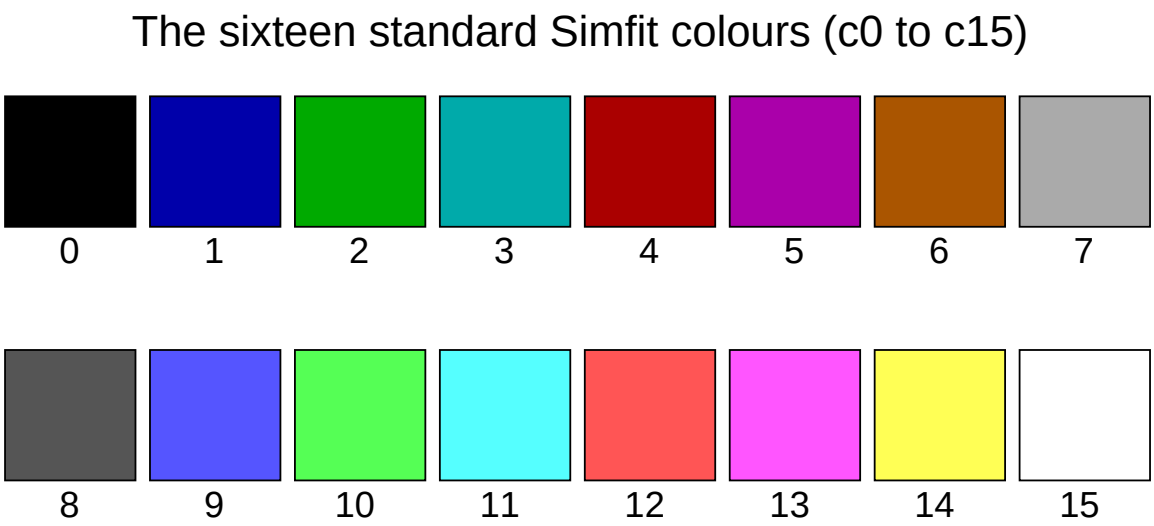
The closing comment section of `w_ps.cfg` summarizes the named standard colors as follows.

```
   red    green  blue
0  0.0000 0.0000 0.0000 black
1  0.0000 0.0000 0.6667 blue
2  0.0000 0.6667 0.0000 green
3  0.0000 0.6667 0.6667 cyan
4  0.6667 0.0000 0.0000 red
5  0.6667 0.0000 0.6667 magenta
6  0.6667 0.3333 0.0000 brown
7  0.6667 0.6667 0.6667 white
8  0.3333 0.3333 0.3333 dark grey
9  0.3333 0.3333 1.0000 light blue
10 0.3333 1.0000 0.3333 light green
11 0.3333 1.0000 1.0000 light cyan
12 1.0000 0.3333 0.3333 light red
13 1.0000 0.3333 1.0000 light magenta
14 1.0000 1.0000 0.3333 light yellow
15 1.0000 1.0000 1.0000 bright white
```

When a *.eps file is created the current 72 color definitions are read from `w_ps.cfg` and written to the *.eps file header section. So finally, here is how these standard colors are defined in the *.eps file headers using rgb as an abbreviation for setrgbcolor and D for define.

```
/c0{0.000 0.000 0.000 rgb}D /c1{0.000 0.000 0.667 rgb}D
/c2{0.000 0.667 0.000 rgb}D /c3{0.000 0.667 0.667 rgb}D
/c4{0.667 0.000 0.000 rgb}D /c5{0.667 0.000 0.667 rgb}D
/c6{0.667 0.333 0.000 rgb}D /c7{0.700 0.700 0.700 rgb}D
/c8{0.450 0.450 0.450 rgb}D /c9{0.333 0.333 1.000 rgb}D
/c10{0.333 1.000 0.333 rgb}D /c11{0.333 1.000 1.000 rgb}D
/c12{1.000 0.333 0.333 rgb}D /c13{1.000 0.333 1.000 rgb}D
/c14{1.000 1.000 0.333 rgb}D /c15{1.000 1.000 1.000 rgb}D
```

The SIMFIT color palette can be opened from the configuration option and is always made available when a color change is requested. So the next figure shows the first sixteen colors (c0 to c15) from this color palette.



Editing PostScript files

Note that, at any stage, you can open a SIMFIT *.eps file in a text editor such as **notepad** and add a new line to make that color the current color until the next time it is changed. For instance if the next line in a file

```
...
c4
...
```

is changed to

```
...
c12
...
```

then the current color will become light red instead of red.

Equally the new color command can be made explicitly as in

```
...
1.000 0.333 0.333 setrgbcolor
...
```

and, clearly, proceeding in this way any color change can be achieved.

Program **editps** provides many options for re-sizing and rotating SIMFIT *.eps files and has several editing opportunities. However, in the examples that follow, it is assumed that the procedure will be as follows

1) Save the *.eps as a backup or edit a copy
2) Edit the *.eps file in a text editor
3) View the edited file in, e.g., **gsview**
4) Save the final result

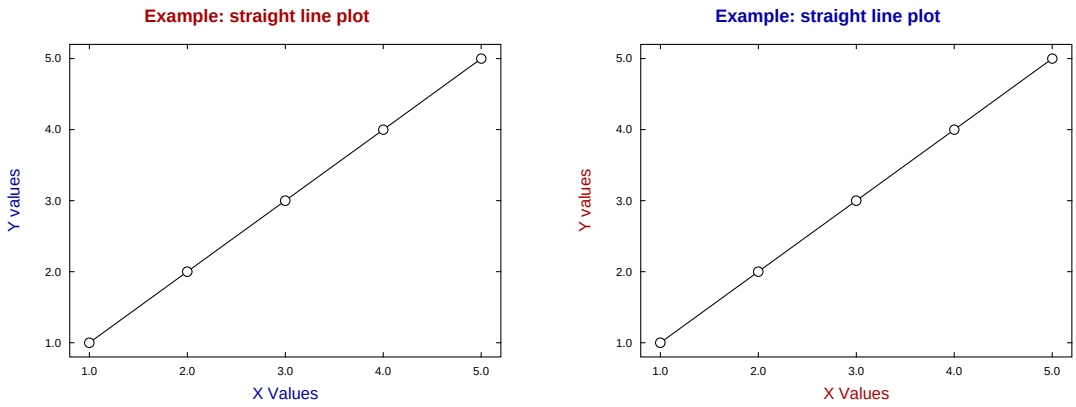
as, after a bit of practise, this is the easiest way to edit SIMFIT *.eps files.

14.12.2 Example 1: Changing colors in a title and legends

In this example, the *.eps file is opened in a text editor which is then searched for the expression `%#title` to move to the code in the file defining the title which is

```
c4
(Example: straight line plot) 3195 4467 ti%#title
(00000000000000000000000000000000) fx
/ti-size ti-size   1.000 mul def
/xl-size xl-size   1.000 mul def
c1
(X Values) 3515 192 xl%#x legend
(00000000) fx
/xl-size xl-size   1.000 mul def
/yl-size yl-size   1.000 mul def
(Y values) 501 2443 yl%#y legend
(00000000) fx
/yl-size yl-size   1.000 mul def
```

and which results in the title being colored red and the legends colored blue, as in the left hand figure below.

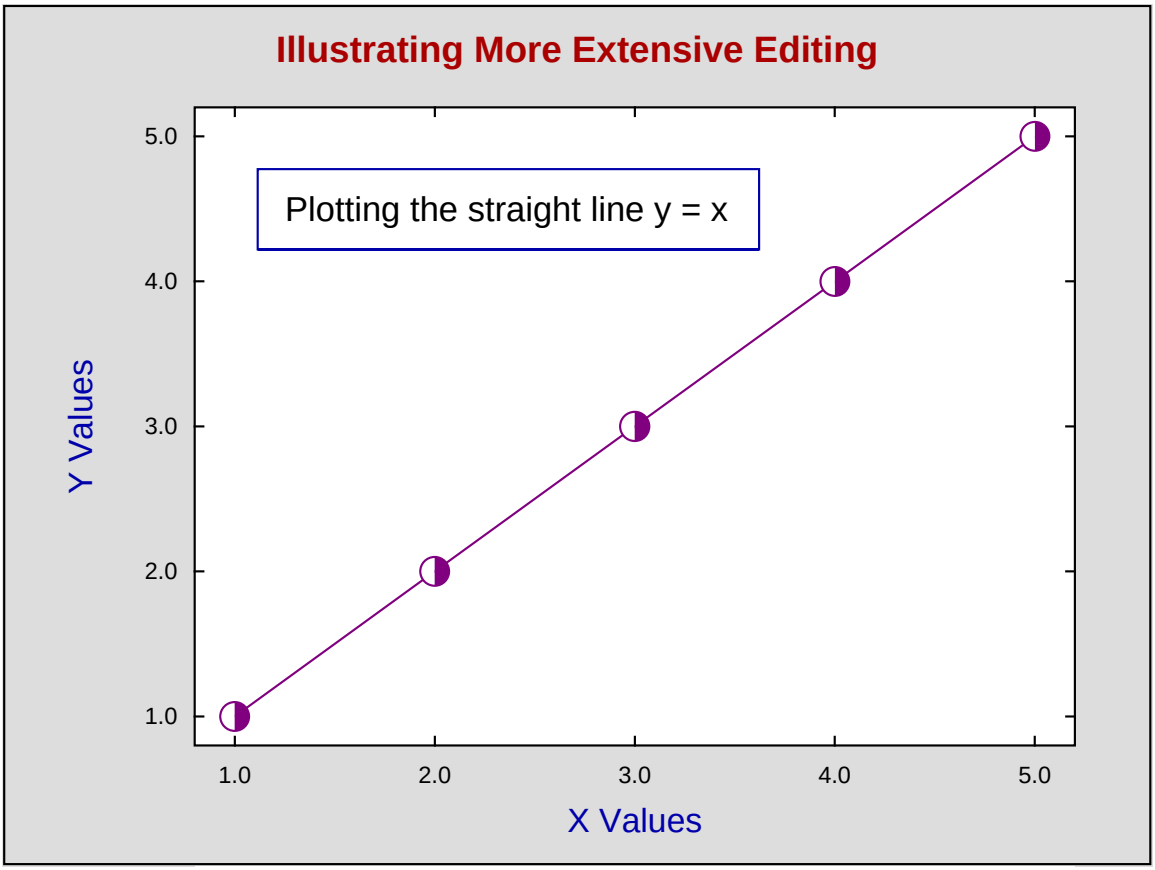


Simply interchanging the code for the red title and blue legend, i.e., replacing c4 (red) and c1 (blue) by c1 (blue) and c4 (red) as shown in the next section

```
c1
(Example: straight line plot) 3195 4467 ti%#title
(00000000000000000000000000000000) fx
/ti-size ti-size   1.000 mul def
/xl-size xl-size   1.000 mul def
c4
(X Values) 3515 192 xl%#x legend
(00000000) fx
/xl-size xl-size   1.000 mul def
/yl-size yl-size   1.000 mul def
(Y values) 501 2443 yl%#y legend
(00000000) fx
/yl-size yl-size   1.000 mul def
```

results in the blue title and red legends shown in the right hand plot.

14.12.3 Example 2: More extensive editing



First of all consider the steps that were required to obtain the graph above from the previous plots.

1. A grey border was added to the outside of the data–plotting area.
2. A black frame was added to surround the overall graph.
3. The title was edited.
4. The line and plotting symbols were colored, and the symbol type changed.
5. An information panel was added.

With experience it is easy to perform such operations directly using a text editor. However, the following procedure could be used to appreciate how to gain such experience.

- Create the original graph in **simplot** [Advanced Editing] and save as `file_1.eps`.
- Edit the data in **simplot** [Advanced Editing] and then save as `file_2.eps`.
- Open `file_1.eps` and `file_2.eps` in a file comparison editor, e.g., **notepad++**.
- See how to color the line and alter the symbol size and type.
- Copy and paste the new code sections from `file_2.eps` into `file_1.eps`.
- Copy the new codes to the clipboard and archive to be re–used as templates.

Here are the new sections that were copied from `file_2.eps` into `file_1.eps`, but with comments (following %) removed for clarity.

Code for the border using color c22 and pf (polygon filled)

```
c22
0 0 0 4790 1070 4790 1070 0 4 pf
5959 0 5959 4790 6390 4790 6390 0 4 pf
1070 0 1070 671 5959 671 5959 0 4 pf
1070 4215 1070 4790 5959 4790 5959 4215 4 pf
```

Code for the black frame using pc (polygon closed)

```
0 setlinejoin
12 12 6378 12 6378 4778 12 4778 4 pc
1 setlinejoin
```

Code for ti (title) and fx (character keys)

```
c4
(Illustrating More Extensive Editing) 3195 4467 ti
(0000000000000000000000000000000000) fx
```

Code for the pl (polyline) line)

```
c47
1292 832 2403 1638 3514 2443 4626 3248 5737 4054 5 pl
```

Code for larger size (80) ch (circle half filled)

```
c47
1292  832 80 ch
2403 1638 80 ch
3514 2443 81 ch
4626 3248 80 ch
5737 4054 80 ch
```

Code for the information panel

```
c1
1420 3426 4205 3426 4205 3872 1420 3872 1420 3426 4205 3426 6 pl
1 setlinejoin
c0
/font /Helvetica D /size   192 D
GS font F size S   1573  3585 M    0 rotate
(Plotting the straight line y = x)
(0000000000000000000000000000000000) fx
```

14.13 Creating hardcopy from eps files

There are essentially two extreme types of graphics files used to display or report scientific data: vector files and bit–map files but, because bit–map files are usually very large, several types of compression can be used. An explanation of which type of file to use follows, and this information is particularly valuable with SimFIT *.eps files because they can be used retrospectively to generate any type of graphics file.

14.13.1 Standard graphics files

1. Vector files

   Vector files simply contain the instructions to draw lines and text in various sizes and colors and an interpreter is needed to visualize or print the file. The great advantage with such files is the compactness as only the information essential to interpret the data is included. As the size of plot being produced depends on the replay system, as does the application of anti-aliasing to smooth out curves and improve the readability of fonts, these files are essentially device-independent. Three of the standard formats are as follows.

   (a) Encapsulated PostScript (*.eps)

   This a special form of PostScript file consisting of exactly one page but with certain restrictions such as the necessity for a BoundingBox, which simply records the coordinates of the rectangle containing the plot. This feature allows computer programs to easily resize, re-scale, rotate, make collages, etc., and explains the wide use with advanced type-setting systems, such as LaTeX.

   (b) Scalable Vector Graphics (*.svg)

   This is a special form of XML containing only markup commands and is especially recommended by the W3C for the internet. Because of the increasing use of *.svg files there are now many editors to enhance the graphs, which is a strong recommendation.

   (c) Enhanced Metafiles (*.emf)

   This is a dedicated Windows format and so can be imported easily into documents prepared by most word processing programs. However, some versions allow users to scale unsymmetrically which will change the aspect ratio and can lead to ugly effects.

2. Bit–map files

   Typical bit–map files (*.bmp) contain the characteristic values for each pixel in a display. Because of this they tend to be very large and of fixed resolution, and so tend to be used to record photographs of tissue sections and similar, where fine variations in color and brightness are important. They are very convenient for inserting into documents and for photo-editing, but it is normal to store in a compressed form especially when there is much redundant information. For instance, the background of a scientific graph consisting of lines, symbols, and text, accounts for a large portion of the bit–map information, but actually it can be represented entirely by the value of one pixel.

14.13.2 Compressed bit–maps

Compression always leads to loss of information and there are many compression algorithms. The most used format is the joint photographic group one (*.jpg), but for the web the portable network graphic format (*.png) is recommended because it is rather more lossless and better captures the lines and curves found in scientific graphs. As compression cannot be reversed and the resolution cannot be improved, it is advantageous to create *.jpg and *.png files from *.eps files because they can then be of much higher resolution than those saved as screen dumps.

14.13.3 Document files

Two standard document files that can also be used as graphics files are the portable document format (*.pdf) and the Microsoft Windows *.xps format.

The *.pdf format developed from PostScript but with the additional advantage that font information could be embedded. This means that *.pdf files created from SIMFIT *.eps files retain some of the the virtues of being compact and to a degree device independent. For instance, the SIMFIT reference manual (`w_manual.pdf`) has some 500 pages packed with hundreds of mathematical formulas, tables, and graphs but is only about 5MB: a very small fraction of what such a manual would be when produced by a typical document preparation program.

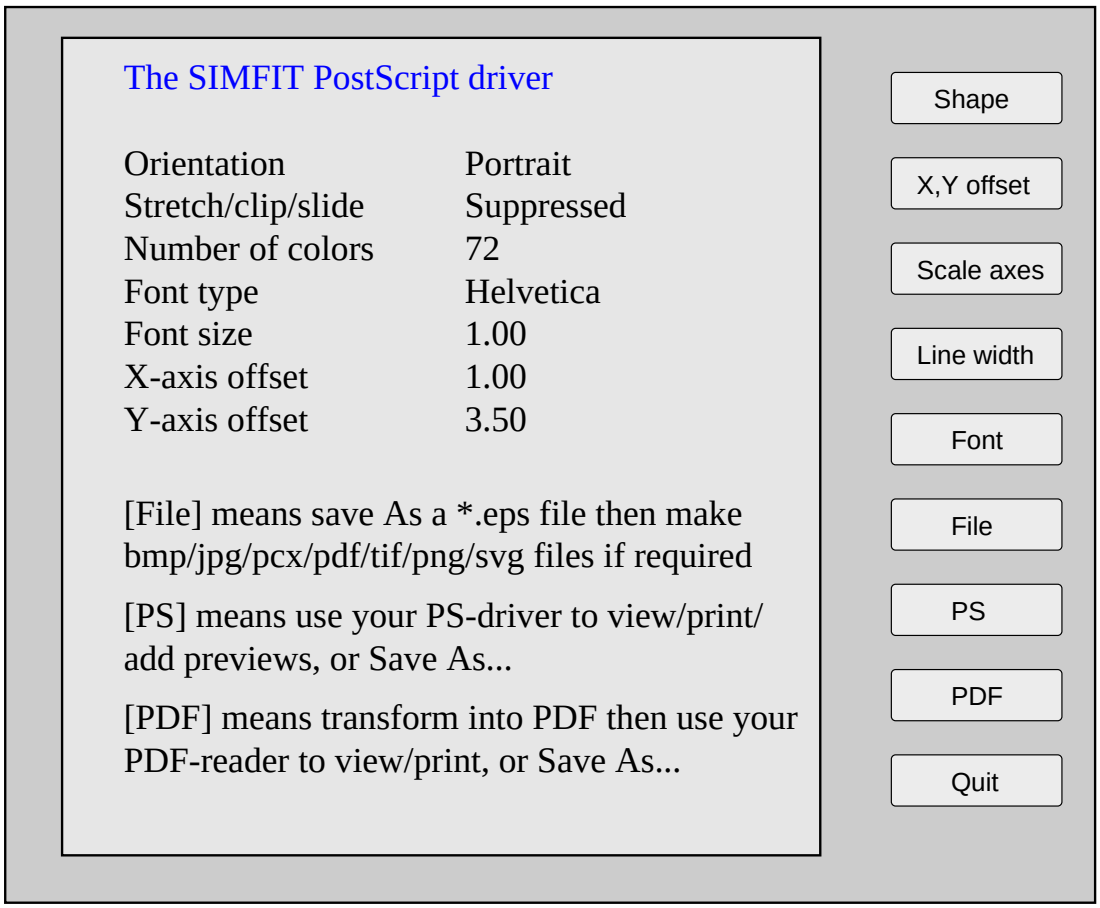
The *.xps format is a specialized ZIP archive file containing XML and was created to compete with the *.pdf format. It is only of value to users of Microsoft Office, but nevertheless such files can be created from SIMFIT *.eps files.

14.13.4 Warning

SIMFIT users should note that *.svg files created directly using the [Win] option or *.eps files using the [PS] option are true vector files. That is, they only consist of instructions that can be used to create hardcopy of any size and resolution for viewing or printing, and they contain no embedded bit–maps. Unfortunately, all the vector files in existence have the facility to embed bit–maps for pre-viewing, and this has been exploited by many computer programs, to the extent that *.eps files can be output as screen dumps from bit–map displays so as to constitute the total file without including vector–type information, so completely vitiating the whole purpose of vector graphic files format.

14.13.5 Using GhostScript to create graphics files form *.eps files

The recommended way to save graphics files from SIMFIT is to use the [PS] option first to open the PostScript interface, shown next.



From the [File] option it is possible to save a *.eps file which can then create transformed files immediately. Otherwise, use the main SIMFIT [Plot] control or else program **editps** retrospectively, to transform *.eps files into one of the following file types.

1. **PDF**

   This will create a high quality file preserving many of the *.eps vector file features.

2. **PNG**

   This is highly recommended as long as the effect of the dots per inch (DPI) setting is understood.

   If a low value for DPI is chosen, say 72 dpi, the resulting file will be satisfactory for a website thumbnail but not suitable for a document or visual display. Choosing a larger value, say 300 dpi, will be fine for documents but the resulting file size will be greater, while choosing, say > 600 dpi, will result in much bigger files with little further improvement in resolution. Anti-aliasing can be used to make curves smoother and fonts more readable.

   One great advantage of *.png files created this way instead of directly from the [Win] hardcopy control is that files saved from the [Win] control have a fixed maximum resolution limited by the display size, but files created using *.eps with GhostScript can have much higher resolution.

3. **JPG**

   Comments about DPI as for PNG files, but subsequent editing may be easier.

4. **XPS**

   Comments about DPI as for PNG files, but only useful for Word processors that can import *.xps files.

5. **PCX**

   Now an obsolete format.

6. **BMP**

   Comments about DPI as for PNG files, but can lead to massive files.

7. **TIFF**

   Now an obsolete format

8. **SVG**

   The quality available from GhostScript has been very poor so, unless this changes, it is preferable to use **Gsview**. In any case, SVG files saved directly from the [Win] option are true vector files. Note that, program **GSview** version 6.0 on uses MuPdf technology to generate high quality *.svg files from SIMFIT *.eps files.

14.13.6 Retrospective editing of graphics files

Very often it is wished to make changes to scientific graphs such as altering line thickness, color, dashed type into dotted, etc., or maybe replacing circles by triangles for clarity before incorporating into documents. It is very easy to make such changes to SIMFIT *.eps files using any text editor as the format was designed with this in mind, and the way to do this is described in the reference manual and tutorials.

Now there are many programs for editing graphics files such as *.jpgf or *.svg but it is obvious that such possibilities are not available in hardcopy files created from *.eps. Even editing SIMFIT *.eps in dedicated PostScript editors such as Adobe Illustrator destroys the ability to edit SIMFIT *.eps files retrospectively in a text editor.

15 Scalable vector graphics (SVG)

Tutorials and worked examples for simulation, curve fitting, statistical analysis, and plotting.
https://simfit.uk
https://simfit.org.uk
https://simfit.silverfrost.com

15.1 SVG: introduction

In order to appreciate the nature and use of scalable vector graphics files (*.SVG) it is useful to review the two document types, namely bitmap and vector, that are used to archive graphs and include them into computer generated documents or web pages.

15.1.1 Bitmaps

These files (*.BMP) contain the raw information for every pixel captured by a digital photograph or displayed on a computer screen. They have the following properties.

- The larger the number of pixels then the greater the detail recorded.
- Such files are very easy to display, include in documents, or print but, unless there is a large number of pixels, then lines and curves will appear stepped and fonts will pixelate.
- For complicated portraits, landscapes, capture of microscope fields, or 3D display of molecules etc. requiring shading such files are indispensable.
- Where there are appreciable homogeneous patches such as blue sky in a landscape then considerable compression into formats such those of the joint photographic expert group (*.JPG) or portable network graphics (*.PNG) can result in smaller files, but compression is not always lossless.
- Where a bitmap has been created using antialiasing to smooth out polygons and polylines as in text, curves, or lines then compression can lead to fuzzy curves or distorted characters. Traditional computer graphics hardly ever looked good on computer screens because hard edges at an angle would show as a series of steps (a distortion known as aliasing). This made even the simplest graph – such as a sine curve, look ugly. This problem has been largely eliminated by anti-aliasing (automatically employed by SIMFIT). However a bitmap of a specific size generated using anti-aliasing, never looks its best if it is ultimately displayed at another resolution. SVG completely sidesteps this problem because the necessary anti-aliasing is performed as an image is displayed (or printed), so that data are displayed correctly. This property of being device–independent is something that sets aside vector graphics from bitmap graphics.

15.1.2 Vector graphics

Scientific graphs largely consist of axes, curves, and plotting symbols, with small amounts of text, and vector graphic files simply contain the mathematical data such as coordinates necessary to reproduce the graph at any degree of magnification or compression without loss of information. Here is a summary of properties for the two main vector graphics files; encapsulated PostScript (*.EPS), and scalable vector graphics (*.SVG).

- The files are in text format, which means they can easily be edited retrospectively in text editors such as **notepad** in order to change, titles, legends, line types, plotting symbols, or colours.
- They are device independent so there is no loss of information on expansion or compression.
- They can be imported into LaTeX documents or include LaTeX code, so that high quality mathematical formulas and chemical structures can be incorporated.

- The free program **Ghostscript** can be used to convert EPS files into other formats, and similarly **Inkscape** can be used to visualize and transform SVG files.
- While EPS is the main import format for LaTeX documents, SVG is the recommended format for scientific graphs on the internet.

15.1.3 Bogus vector files

Note that many applications claim to transform bitmap and compressed bitmap files into vector files without loss of significant information, but this is almost impossible except for fairly simple images. Such applications usually just exploit a weakness in vector files that allows bitmaps to be inserted giving bogus vector files that are wrappers containing bitmaps. Similarly portable document files (*.PDF) were developed from PostScript and retain many PostScript features that can be exploited. For instance, using the SimFIT interface to GhostScript to transform SimFIT EPS files into PDF files yields PDF files that are effectively device independent, whereas using Windows to distil files into PDF merely creates bitmap files.

15.1.4 Using SVG files in SimFIT

The SVG format is very comprehensive as it has been specifically developed to be versatile for web use. For that reason few applications implement the whole standard and SimFIT is no exception, so it must be emphasized that SimFIT will only accept and manipulate SVG files according to the graph plotting functionality provided by Silverfrost FTN95 Clearwin+. This interface was written by David Bailey and it provides the following SVG file functionality for SimFIT users.

- The SVG files can be used on the web and opened by browsers such as **firefox**.
- The SVG files can be displayed and edited retrospectively by the program **EditSVG**.
- The SVG files created by other applications may cause warning messages, but in some circumstances files may use acceptable subsets of SVG facilities. To explore this option just open the SVG file with a text editor and splice the option Clearwin_output="1" into the line beginning <svg. It is easy to get this wrong, so it is probably best to file the result to a different file name.

15.1.5 Editing SVG files in SimFIT

The functionality provided by procedure **EditSVG** is now listed.

1. The only file types that can be used in this program are:

   (A) *.SVG files created by SimFIT, and other Clearwin+ output from Silverfrost FTN95 programs.

   (B) *.TEX files describing mathematical equations or chemical formulas. Such *.TEX files can be used to generate internal *.SVG files if **latex.exe**, **dvips.exe**, and **dvisvgm.exe** are on the path. For instance if users have a recent version of MikTeX available.

2. Files can be input from the console, by drag and drop, or by using library files.
3. Images can be enlarged or reduced by "right clicking" on the image.
4. Images can be freely positioned by dragging a window from any point on its surface. Using the view menu it is possible to add a graticule with optional "snap to nearest graticule intersection" facility. The graticule does not remain on the finished image.
5. After manipulating a file or a set of files the resulting composite image can be written out to a *.SVG, or *.PNG, file, or even to a *.ISVG rebuild–image file.
6. It can be used to create strict collages where every image is snapped to the nearest grid point, freestyle collages where images can be arranged in arbitrary positions, or overlays where smaller images can be inserted into larger ones.

15.1.6 Using LaTeX

Some examples of how to use these procedures within the SimFIT package from version 7.5.0 onwards using the test files provided follow. However the procedure used to create the SVG files using LaTeX should be noted.

To enlarge on the use of LaTeX it must be emphasized that the procedure to use LaTeX depends on whether the user has a fully functioning LaTeX installation on their machine. So there are three distinct cases.

1. **The direct method**

   There is a LaTeX installation so the user prefers to input a *.TEX file directly into program **EditSVG** whereupon the *.TEX file will be processed and the image will appear in the main window.

2. **The indirect method**

   There is a LaTeX installation but the user prefers to use the command line technique described subsequently to transform the *.TEX file into *.DVI then use **dvisvgm** to transform this into a stand alone *.SVG file. Note that filenames used in this procedure must be local files with no spaces in the file name.

3. **The remote user method**

   There is no LaTeXinstallation so a known LaTeX user will have to perform the transformation. Alternatively, a stand–alone *.SVG file, such as the demonstration files distributed with SimFIT will have to be used.

LaTeX is designed to create documents and, because of this, care is needed to remove much of the header information in order to create simple images that can be imported into **EditSVG**. This is much the same as the steps required to create a *.EPS from from a *.PS file but using the following commands.

To make DVI: use `latex myfile.tex` to create `myfile.dvi`.

To make PS: use `dvips myfile.dvi` to create `myfile.ps`.

To make SVG: use `dvisvgm -E myfile.ps` to create `myfile.svg`.

The argument `-E` indicates that a PostScript file is to be input. Note that white space can be trimmed from the resulting SVG file by **EditSVG**, or alternatively by using **Inkscape** or **GSview** (e.g. Version 5) to transform `myfile.ps` into `myfile.eps`, where the `BoundingBox` will automatically remove white space.

15.1.7 Important differences between EPS and SVG files

From within any SimFIT graph it is possible to create *.EPS files by first selecting [PS] then choosing [File], or to create *.SVG files by first selecting [Win] then choosing the [SVG] option. Any *.SVG file created in this way will be a fairly accurate representation of the display, as will any *.EPS files, except that there will be small but significant differences between them. That is unavoidable because the fonts used may differ slightly as the *.SVG file will use Windows fonts whereas the *EPS file will use PostScript fonts. In addition SimFIT allows users to set different global line thickness for EPS and SVG files as line thickness do not scale in exactly the same way in EPS and SVG files. Nevertheless it is useful to know how to create a *.SVG file from a *.EPS file and vice versa.

Fortunately such transformations can readily be carried out due to the widespread availability of Open Source programs such as **Inkscape** and **Cairo**. The usefulness of such transformations will be explained in subsequent tutorial sections.

15.2 SVG: Importing LaTeX maths equations

Sometimes it is required to use LaTeX to display a mathematical equation inside a scientific plot, and this document describes how to do this for the normal distribution cumulative distribution function $\Phi(x)$. Note that all the files mentioned in this document are distributed as SimFIT test files so that users simply wishing to create the final composed document can proceed directly to the last section describing how to use **EditSVG**.

15.2.1 The TEX source

This is the code contained in the file `latex_maths_equation.tex`

```
\documentclass[12pt]{article}
\usepackage{amsmath,bm}
\pagestyle{empty}
\begin{document}
\Large
\[
 \frac{1}{\sigma \sqrt{2\pi}}
   \int\limits_{-\infty}^x
   \exp -\left\{
  \frac{1}{2} \left( \frac{t-\mu}{\sigma} \right)^2 \right\}\,dt
\]
\end{document}
```

which displays the mathematical definition of $\Phi(x)$ as follows.

$$\frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{x} \exp - \left\{ \frac{1}{2} \left(\frac{t-\mu}{\sigma} \right)^2 \right\} dt$$

In order to import this formula into a graph using **EditSVG** this code must be used to create the corresponding SVG file `latex_maths_equation.svg`, the overall process being the following sequence of commands.

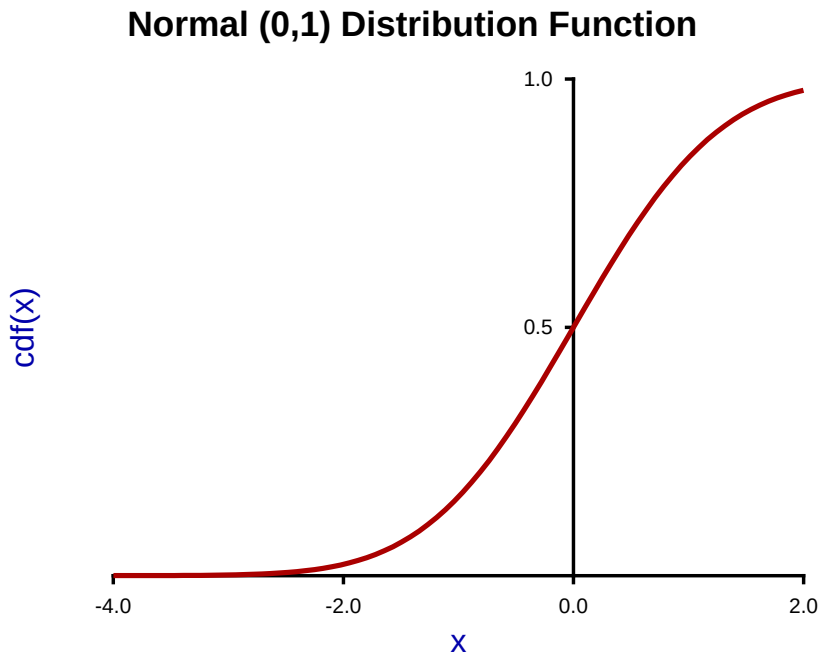
- **latex** `latex_maths_equation.tex`
- **dvips** `latex_maths_equation.dvi`
- **dvisvgm** `-E latex_maths_equation.ps`

The file `latex_maths_equation.svg` created is then ready to be imported into **EditSVG** but, alternatively, the source file `latex_maths_equation.tex` can be opened in or dragged and dropped directly onto **EditSVG** if there is a local installation of LaTeX.

It should be realized that, when using LaTeX in this way to create a SVG file, the command line must be used from a folder containing the *.TEX file required as a local file and not as a fully qualified path–filename to a remote source file. The program **EditSVG** circumvents this issue when importing LaTeX source by creating local copies of all files.

15.2.2 Creating the plot file

The file `latex_maths_plot.svg` with the $\Phi(x)$ profile to be used looks like this before the equation is added.



This figure was created using **makmat** by selecting to display the normal cumulative distribution $\Phi(x)$ with $\mu = 0$ and $\sigma^2 = 1$ for $-4 \leq x \leq 2$, and then transferring the resulting plot into the program **simplot** using the [Advanced] option to manipulate the title, legends, line–widths, and colors, etc.

Users wishing to avoid this process can simply read the SIMFIT metafile `latex_maths_plot.metafile` directly into the SIMFIT program **simplot**, or the SIMDEM program **simdem70**.

In either case the file is then saved as `latex_maths_plot.svg` using the [Win] or [SVG] option.

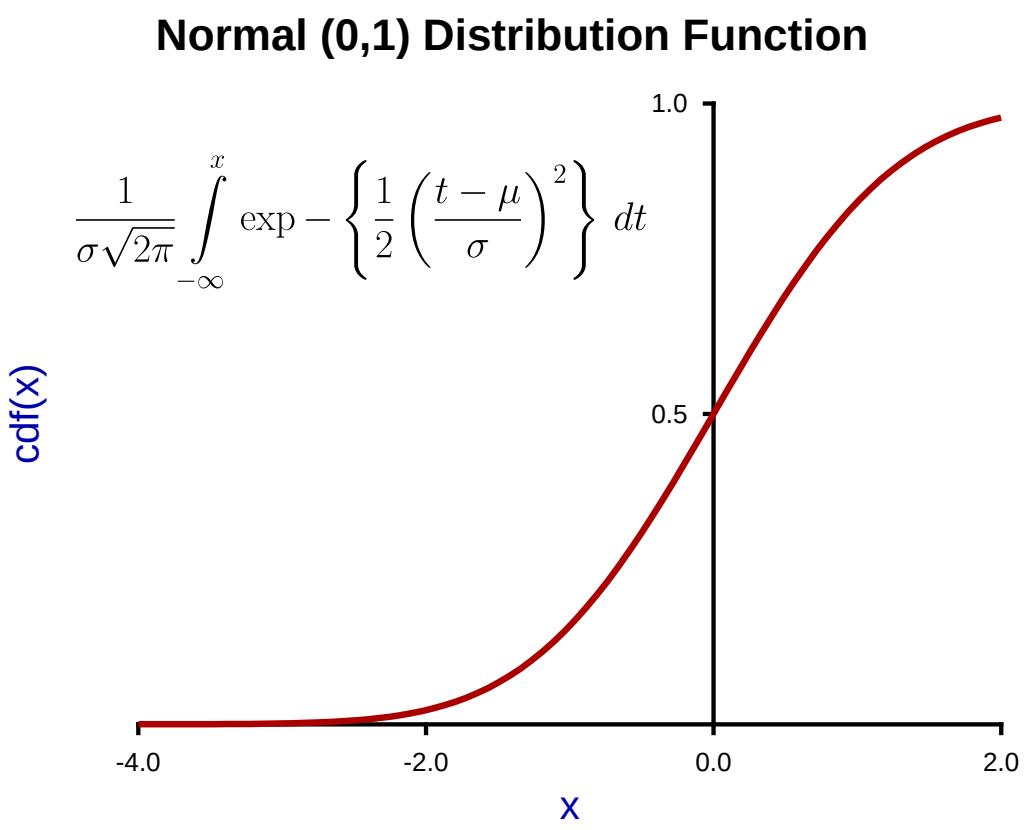
15.2.3 Joining the SVG files using EditSVG

First open program **EditSVG** then input the test file `latex_maths_plot.svg` to act as a background, then there are two possible options.

1. Input the test file `latex_maths_equation.svg` directly; or

2. read in the test file `latex_maths_equation.tex` which will then be used by LaTeX to generate an internal copy of `latex_maths_equation.svg`.

Finally, just use the mouse to move the equation into position and alter the scaling as required to obtain the final plot saved as `latex_maths.svg` and shown next.



15.2.4 Summary of files described in this section

The programs referred to in this document are as follows.

1. **InkScape** is an OpenSource program that takes in SVG files and can write out EPS and other files.

2. **EditSVG** is a SimFIT and SimDEM program that takes in SVG or TEX files and writes out SVG and other files.

3. **editPS** is a SimFIT and SimDEM program that takes in EPS files and writes out only EPS files.

4. The SimFIT program **simplot** and the SimDEM program **simdem70** take in SimFIT metafiles and write out either SVG or EPS files.

Further, the SimFIT test files (*.TEX and *.SVG) described in this document that can be used by program **EditSVG**, and those (*.EPS) that can be used by program **editPS** are now listed.

File name	Data included
`latex_maths_plot.metafile`	SimFIT or SimDEM metafile to create the plot without any equation
`latex_maths_equation.tex`	LaTeX source file for the maths equation with no plot
`latex_maths_equation.svg`	SVG file containing the formula only
`latex_maths_plot.svg`	SVG file containing the plot only
`latex_maths.svg`	SVG file containing both the equation and plot
`latex_maths_equation.eps`	EPS file containing the formula only
`latex_maths_plot.eps`	EPS file containing the plot only
`latex_maths.eps`	EPS file containing both the equation and plot

15.3 SVG: Importing LaTeX chemical formulas

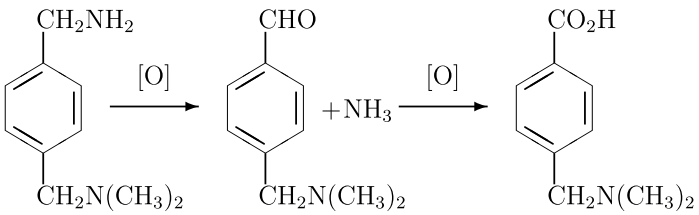
Sometimes it is required use LaTeX to display chemical structures inside a scientific plot, and this document describes how to do this using a condensed scheme for the oxidation of p-dimethylaminomethylbenzylamine. Note that all the files mentioned in this document are distributed as SimFIT test files so that users simply wishing to create the final composed document can proceed directly to the last section describing how to use **EditSVG**

15.3.1 The TEX source

This is the code contained in the file `latex_chemical_formula.tex`

```
\documentclass[12pt]{article}
\usepackage{carom}
\pagestyle{empty}
\begin{document}
{\begin{picture}(3000,600)(0,0)
\thicklines
\put(0,0){\bzdrv{1==CH$_{2}$NH$_{2}$;4==CH$_{2}$N(CH$_{3}$)$_{2}$}}
\put(700,450){\vector(1,0){400}}
\put(820,550){[O]}
\put(1000,0){\bzdrv{1==CHO;4==CH$_{2}$N(CH$_{3}$)$_{2}$}}
\put(1650,400){+}
\put(1750,400){NH$_{3}$}
\put(2000,450){\vector(1,0){400}}
\put(2120,550){[O]}
\put(2300,0){\bzdrv{1==CO$_{2}$H;4==CH$_{2}$N(CH$_{3}$)$_{2}$}}
\end{picture}}
\end{document}
```

which displays like this.



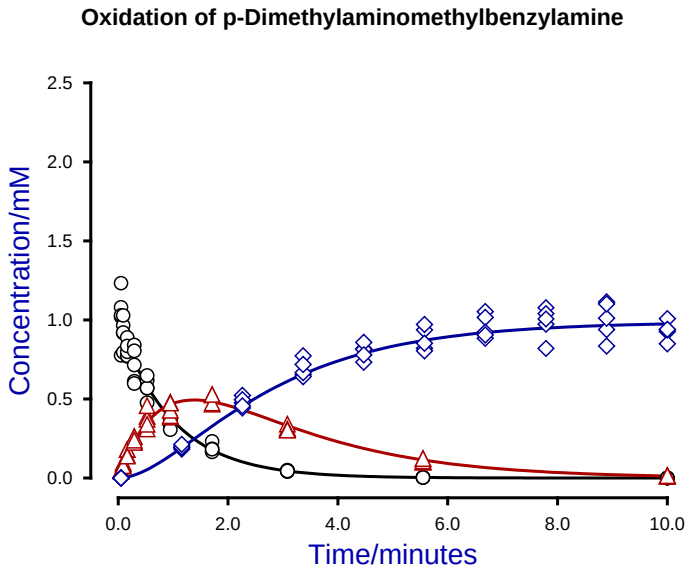
To import this formula into a graph using **EditSVG**, `latex_maths_equation.svg` can be made using the following commands, or `latex_maths_equation.tex` can be input directly into **EditSVG**.

- `latex latex_chemical_formula.tex`
- `dvips latex_chemical_formula.dvi`
- `dvisvgm -E latex_chemical_formula.ps`

The file `latex_chemical_formula.svg` created is then ready to be imported into **EditSVG** but, alternatively, the source file `latex_chemical_formula.tex` can be opened in or dragged and dropped directly onto **EditSVG** if there is a local installation of LaTeX. It should be realized that, when using LaTeX in this way to create a SVG file, the command line must be used from a folder containing the *.TEX file required as a local file and not as a fully qualified path–filename to a remote source file. The program **EditSVG** circumvents this issue when importing LaTeX source by creating local copies of all files.

15.3.2 Creating the plot file

The file `latex_chemical_plot.svg` with the time course data to be used looks like this before the equation is added.



This figure was created using **qnfit** fit three data sets for the consecutive reaction scheme

$$A \rightarrow B \rightarrow C$$

in the SimFIT test library file `consec3.tfl`, then fitted using the model in the model file `consec3.mod`.

After manipulating the line thicknesses, title, legend, and colors, the files `latex_chemical_plot.svg`, and `latex_chemical_plot.eps` were created to archive the graph. In addition the SimFIT metafile `latex_chemical_plot.metafile` was saved so that users wishing generate this plot can easily do so using the SimFIT program **simplot** or the SimDEM program **simdem70**. Users wishing to avoid this process can simply read the SimFIT metafile `latex_maths_plot.metafile` directly into the SimFIT program **simplot**, or the SimDEM program `simdem70`.

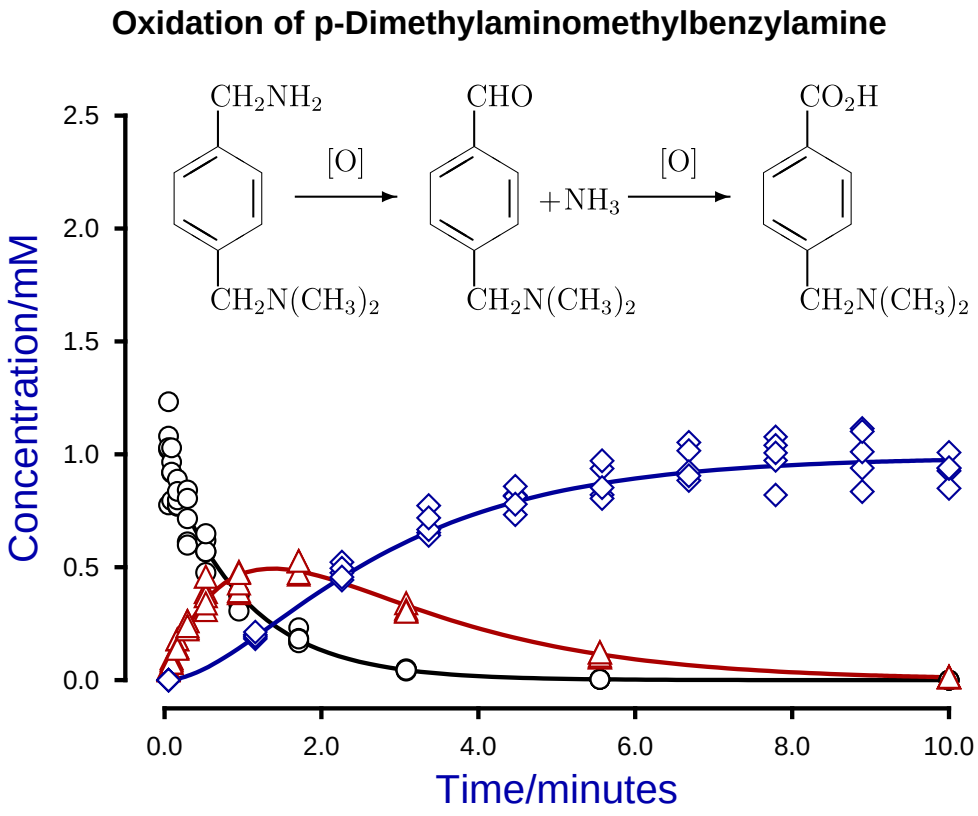
Joining the SVG files using EditSVG

Open program **EditSVG** then input the test file `latex_chemical_plot.svg`. Then there are two options.

1. Input the test file `latex_chemical_formula.svg` directly; or

2. read in the test file `latex_chemical_formula.tex` which will then be used by LaTeX to generate an internal copy of `latex_chemical_formula.svg`.

Finally, just use the mouse to move the equation into position and alter the scaling as required to obtain the final plot saved as `latex_chemistry.svg` and shown next.



15.3.3 Summary of files used in this section

The programs referred to in this document are as follows.

1. **InkScape** is an OpenSource program that takes in SVG files and can write out EPS and other files.
2. **EditSVG** is a SimFIT and SimDEM program that takes in SVG or TEX files and writes out SVG and other files.
3. **editPS** is a SimFIT and SimDEM program that takes in EPS files and writes out only EPS files.
4. The SimFIT program **simplot** and the SimDEM program **simdem70** take in SimFIT metafiles and write out either SVG or EPS files.

Further, the SimFIT test files (*.TEX and *.SVG) described in this document that can be used by program **EditSVG**, and those (*.EPS) that can be used by program **editPS** are now listed.

File name	Data included
`latex_chemical_plot.metafile`	SimFIT or SimDEM metafile to create the plot without any equation
`latex_chemical_formula.tex`	LaTeX source file for the maths equation with no plot
`latex_chemical_formula.svg`	SVG file containing the formula only
`latex_chemical_plot.svg`	SVG file containing the plot only
`latex_chemistry.svg`	SVG file containing both the formula and plot
`latex_chemical_formula.eps`	EPS file containing the formula only
`latex_chemical_plot.eps`	EPS file containing the plot only
`latex_chemistry.eps`	EPS file containing both the formula and plot

15.4 SVG: Importing SVG files into SVG files

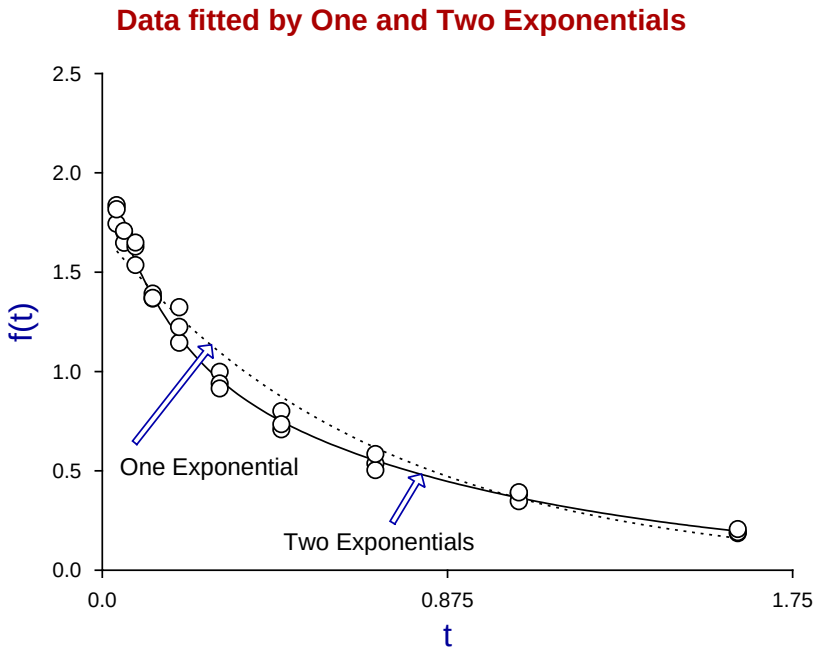
Sometimes it is required to import one SVG file into another SVG file,for example to create overlays, insets, or collages. This document describes how to do this using program **EditSVG** when fitting one then two exponential functions to a data set and plotting the best–fit curves to evaluate the improvement in fit.

15.4.1 Fitting exponential functions

Using SimFIT program **exfit** in the default mode to analyze data in the test file `exfit.tf4` for models of orders 1 and 2 fits the following exponential functions sequentially.

$$f_1(t) = Ae^{-Bt}$$
$$f_2(t) = \alpha_1 e^{-k_1 t} + \alpha_2 e^{-k_2 t}$$

Program **exfit** then outputs goodness of fit criteria and statistical tests to see if there is sufficient statistical evidence to accept the need to fit the additional parameters required by the two exponential model. In addition the following graph is plotted to illustrate the goodness of fit for both models.



The following SimFIT test files are provided to reproduce this fit.

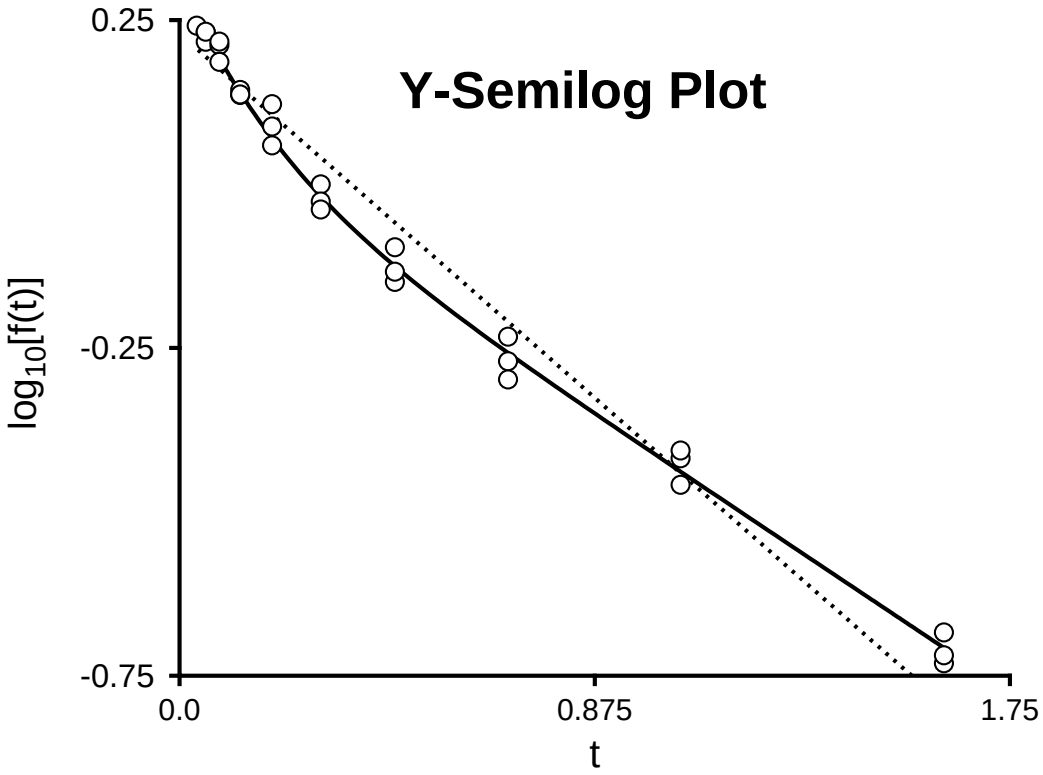
- `exfit.t4f`, the data for input into program **exfit**
- `exfit_normalplot.eps`, the above plot in EPS format for input into program **editps**
- `exfit_normalplot.svg`, the above plot in SVG format for input into program **EditSVG**
- `exfit_normalplot.metafile`, a metafile to create this plot in SimFIT program **simplot** or SimDEM program **simdem70**

The file `exfit_normalplot.svg` was also created at this stage to archive this plot.

15.4.2 Creating the log transform

Of course the previous plot was created from within program **exfit** by simply transferring the plot into SIMFIT advanced graphics mode then editing. Now one technique that can be used to check the relative fit for two exponentials as opposed to one is to plot a semilogarithm plot, which results in linearizing the single exponential function but not the double exponential model.

Within the advanced graphics environment this transformation was selected to produce the next plot.



As the intention was to insert this graph into the previous one the graph was edited as follows

- Suppressing the title
- Changing the legends
- Adding a subsidiary title within the graph
- Making the lines thicker so they do not look too narrow when the plot is reduced in size. As the figure is to be reduced by a factor of two, the line thicknesses were doubled. Sometimes it is useful to enlarge the legends or to replace using a bold font, and even the numbers can be enlarged if required.

The following SIMFIT test files are provided to reproduce this fit.

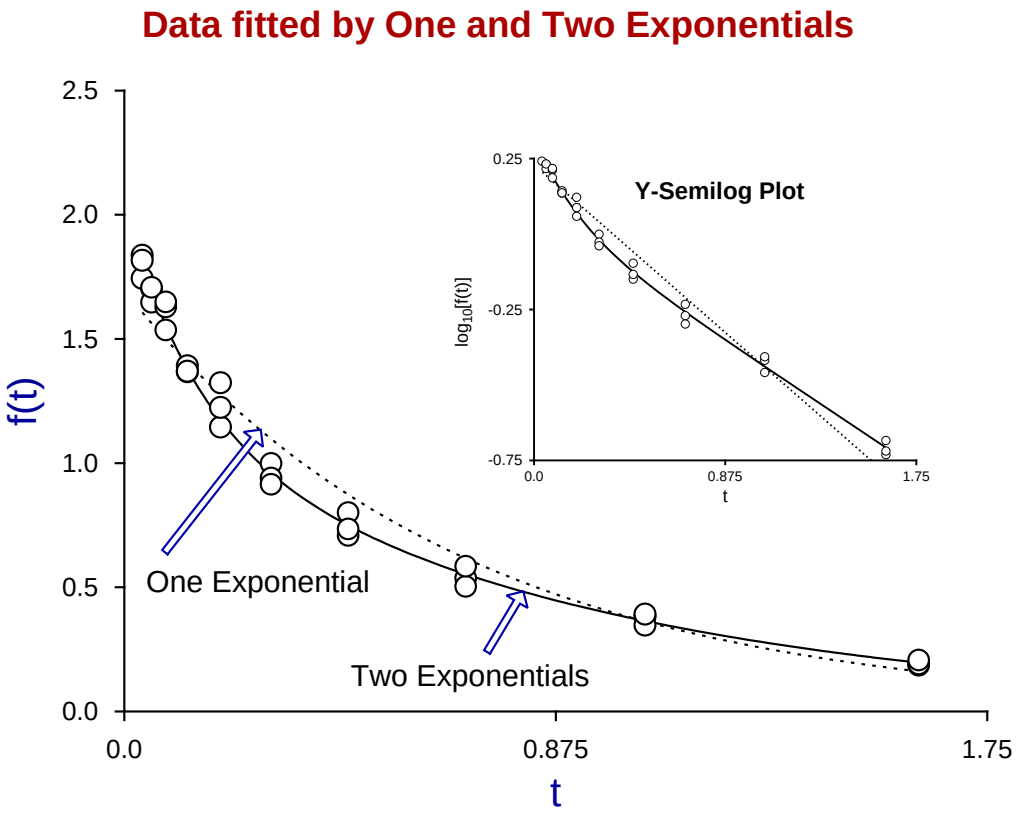
- `exfit_logplot.eps`, the above plot in EPS format for input into program **editps**
- `exfit_logplot.svg`, the above plot in SVG format for input into program **EditSVG**
- `exfit_logplot.metafile`, a metafile to create this plot in SIMFIT program **simplot** or SIMDEM program **simdem70**

The file `exfit.svg` was also created at this stage to archive the compound plot, as described next.

15.4.3 Joining the SVG files using EditSVG

Open program **EditSVG** then input the test file `exfit_normalplot.svg`.

Now input the file `exfit_logplot.svg` then just use the mouse to move the equation into position and alter the scaling as required to obtain the final plot saved as `exfit.svg` and shown next.



15.4.4 Summary of files used in this section

Finally, the SimFIT test files described in this document that can be used to create this plot are now listed.

File name	Data included
`exfit_normalplot.metafile`	SimFIT or SimDEM metafile to create the normal plot
`exfit_logplot.metafile`	SimFIT or SimDEM metafile to create the log plot
`exfit_normal.svg`	SVG file containing the normal plot
`exfit_logplot.svg`	SVG file containing the log plot
`exfit.svg`	SVG file containing both the final compound plot
`exfit_normalplot.eps`	EPS file containing the normal plot
`exfit_logplot.eps`	EPS file containing the log plot
`exfit.eps`	EPS file containing the final compound plot
`exfit.tf4`	Test file containing the data for fitting

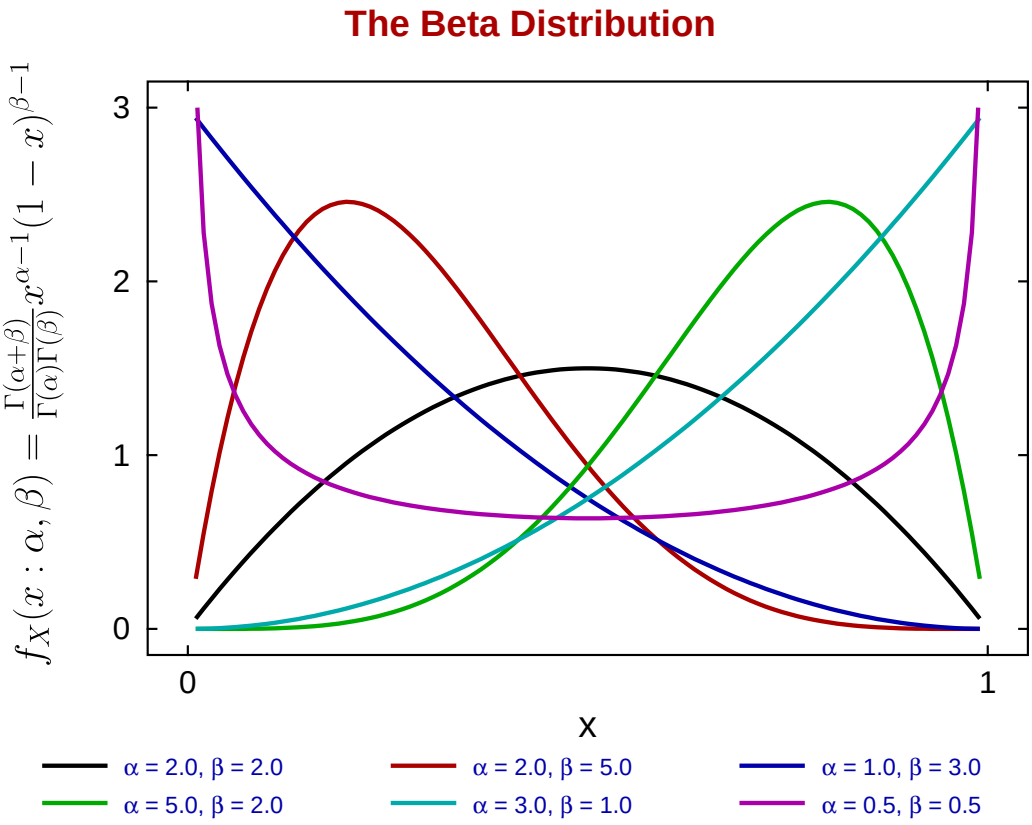
15.5 SVG: Using LaTeX to label SVG y axes

Sometimes it is required to use LaTeX to display a mathematical equation but with the formula rotated so it can be used as the y axis label inside a scientific SVG plot, and this document describes how to do this using the beta probability distribution as an example.

Note that all the files mentioned in this document are distributed as SimFIT test files so that users simply wishing to create the final composed document can proceed directly to the last section describing how to use **EditSVG**.

15.5.1 The beta probability density function

Consider, for example, the wide variety of shapes possible for the beta probability distribution as the two positive parameters α and β are varied as shown next.



This distribution is widely used in data analysis where a unimodal distribution is required as an empirical equation to model data as positive frequencies for a variable x that can be scaled into the range $0 \le x \le 1$.

The great advantage of this distribution is that for positive parameters α and β a great variety of shapes can be generated to illustrate and quantify skew and kurtosis with frequency histograms.

The LaTeX source

This is the LaTeX code contained in the file `latex_beta_pdf.tex` to generate the rotated formula.

```
\documentclass[12pt]{article}
\usepackage{amsmath}
\usepackage{graphicx}
\pagestyle{empty}
\begin{document}
\Large
\rotatebox{90}{$ f_X(x:\alpha,\beta) = \frac{\Gamma(\alpha +
\beta)}{\Gamma(\alpha)\Gamma(\beta)}x^{\alpha - 1}(1 - x)^{\beta - 1}$}
\end{document}
```

which displays the mathematical definition of the beta function (shown before rotation) as follows.

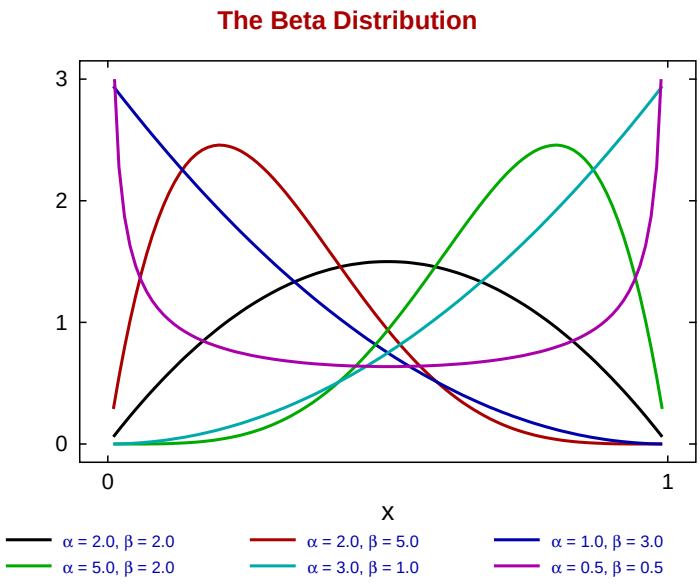
$$f_X(x:\alpha,\beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}x^{\alpha-1}(1-x)^{\beta-1}$$

In order to import this formula into a graph using **EditSVG** the code must be used to create the corresponding SVG file `latex_beta_pdf.svg`, the overall process being the following sequence of commands.

- **latex** `latex_beta_pdf.tex`
- **dvips** `latex_beta_pdf.dvi`
- **dvisvgm** `-E latex_beta_pdf.ps`

The file `latex_beta_pdf.svg` created is then ready to be imported into **EditSVG** but, alternatively, the source file `latex_beta_pdf.tex` can be opened in or dragged and dropped directly onto **EditSVG** if there is a local installation of LaTeX. When using LaTeX in this way to create a SVG file, the command line must be used from a folder containing the *.TEX file required as a local file and not as a fully qualified path–filename to a remote source file. The program **EditSVG** circumvents this issue when importing LaTeX source by creating local copies of all files.

15.5.2 Creating the plot file



The file `beta_pdf_plot.svg` with the $f_X(x : \alpha, \beta)$ to be used looks like the previous figure before the equation is added.

This figure was created using the SIMFIT program **makmat** by selecting to display the beta distribution $f_X(x : \alpha, \beta)$) with various values for the positive parameters α and β over the range $0.01 \leq x \leq 0.99$ so as to avoid the poles at either extreme. Users wishing to avoid this process can simply read the SIMFIT metafile `beta_pdf_plot.metafile` directly into the SIMFIT program **simplot**, or the SIMDEM program **simdem70**. In either case the file is then saved as `beta_pdf_plot.svg` using the [Win] or [SVG] option.

15.5.3 Joining the SVG files using EditSVG

First open program **EditSVG** then input the test file `beta_pdf_plot.svg` to act as a main plot, then there are two possible options.

1. Input the test file `latex_beta_pdf.svg` directly; or
2. read in the test file `latex_beta_pdf.tex` which will then be used by LaTeX to generate an internal copy of `latex_beta_pdf.svg`.

Finally, just use the mouse to move the equation into position and alter the scaling as required to obtain the final plot saved as `beta_pdf_with_equation.svg` shown previously at the start of this document.

15.5.4 Summary

The programs referred to in this document are as follows.

1. **EditSVG** is a SIMFIT and SIMDEM program that takes in SVG or TEX files and writes out SVG and other files.
2. The SIMFIT program **simplot** and the SIMDEM program **simdem70** take in SIMFIT metafiles and write out either SVG or EPS files.

Further, the SIMFIT test files (*.TEX and *.SVG) described in this document that can be used by program **EditSVG**, and those (*.EPS) that can be used by program **editPS** are now listed.

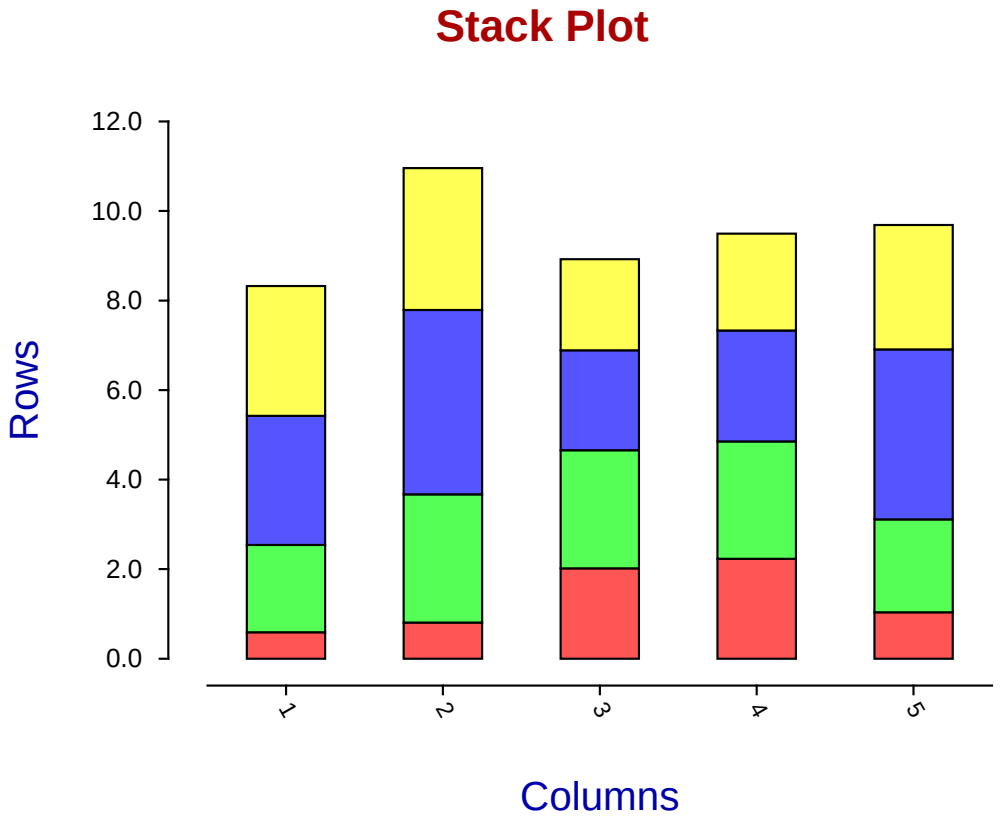
File name	Data included
`beta_pdf_plot.metafile`	SIMFIT or SIMDEM metafile to create the plot without any equation
`latex_beta_pdf.tex`	LaTeX source file for the beta_pdf equation with no plot
`latex_beta_pdf.svg`	SVG file containing the formula only
`beta_pdf_plot.svg`	SVG file containing the plot only
`beta_pdf_with_equation.svg`	SVG file containing both the equation and plot
`beta_pdf_with_equation.eps`	EPS file containing both the equation and plot only
`latex_beta_pdf.eps`	EPS file containing formula only
`beta_pdf_plot.eps`	EPS file containing the plot only

15.6 SVG: Editing using text editors, e.g., Notepad

It is frequently convenient to edit SVG files retrospectively, usually in order to change sizes, titles, legends, labels, line–types, line–widths, and colors, etc. Fortunately, SVG files, like EPS files, are in ASCII text format so they can be edited using any text editor that supports UTF8 characters, such as the Windows program **notepad** or better **notepad++**. The actual format is in XML which is similar to HTML but, in addition, the SVG files created by SimFIT have been designed with editing in mind. For instance, each individual markup code starts on a new line.

15.6.1 Titles and Legends

As a simple example consider the following stacked bar chart created by importing the SimFIT metafile `stack_plot.metafile` into SimFIT program **simplot** or SimDEM program **simdem70**.



On searching for the string "`Stack Plot`" in the file `stack_plot.svg` using a text editor we find the following markup at line 34, where the tokens have been shown on separate lines for clarity.

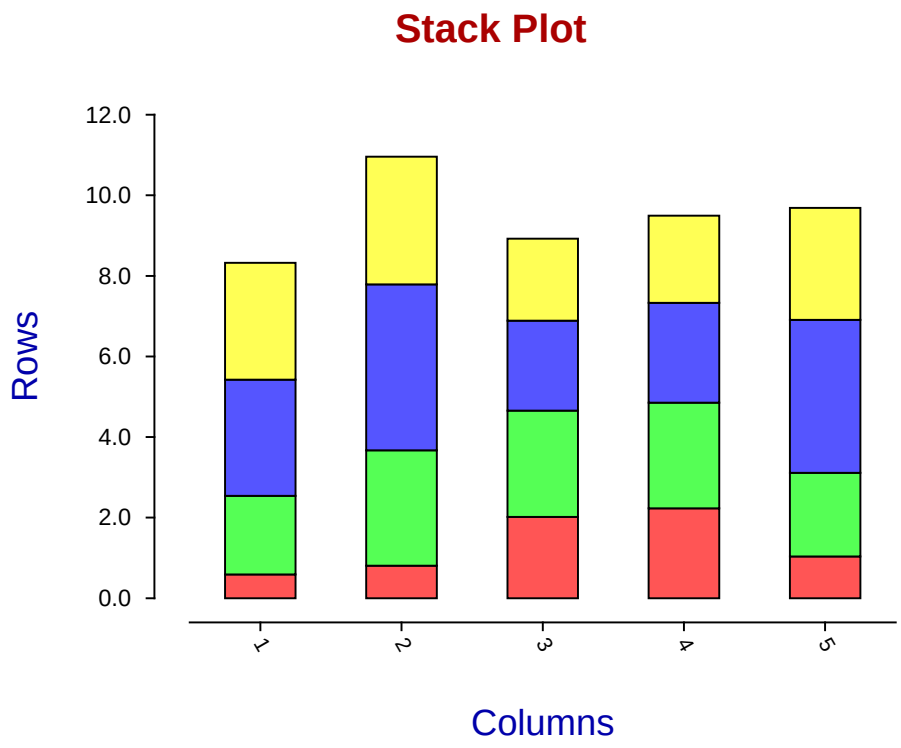
```
<text
x="420.00" y="44.72"
font-family="ARIAL"
font-size="47px"
font-weight="700"
fill="rgb(170,0,0)">
Stack Plot</text>
```

Note the key markup codes used in this line.

- <text ... ></text>

- font-family=
- font-size=
- font-weight=
- fill=

We might think the title could be more explanatory and would be better using a more subdued coloring given the amount of color already associated with the stack segments. So, after some editing in **notepad++**, re-displaying using **firefox** gives the next graph.



Lines 34, 35, and 36 now look like this (with line numbers added for clarity).

```
34 <text x="330.00" y="44.72" font-family="ARIAL" font-size="47px"
    font-weight="700" fill="rgb(0,0,170)">Bar Chart in Stacked Format</text>
35 <text x="504.00" y="788.92" font-family="ARIAL" font-size="42px"
    font-weight="400" fill="rgb(0,170,0)">Columns</text>
36 <text x="68.00" y="449.44" font-family="ARIAL" font-size="44px"
    font-weight="400" transform="rotate(270.00,68.00,460.00)"
    fill="rgb(0,170,0)">Rows</text>
```

The following editing will be apparent on inspection.

Line 34

The original short title "Stack Plot" has been changed to the longer title "Bar Chart in Stacked Format", the color has been changed from red rgb(170,0,0) to blue rgb(0,0,170), and the x-coordinate has been changed from 420.0 to 330.00 (so that the title remains centralized over the plot).

Line 35

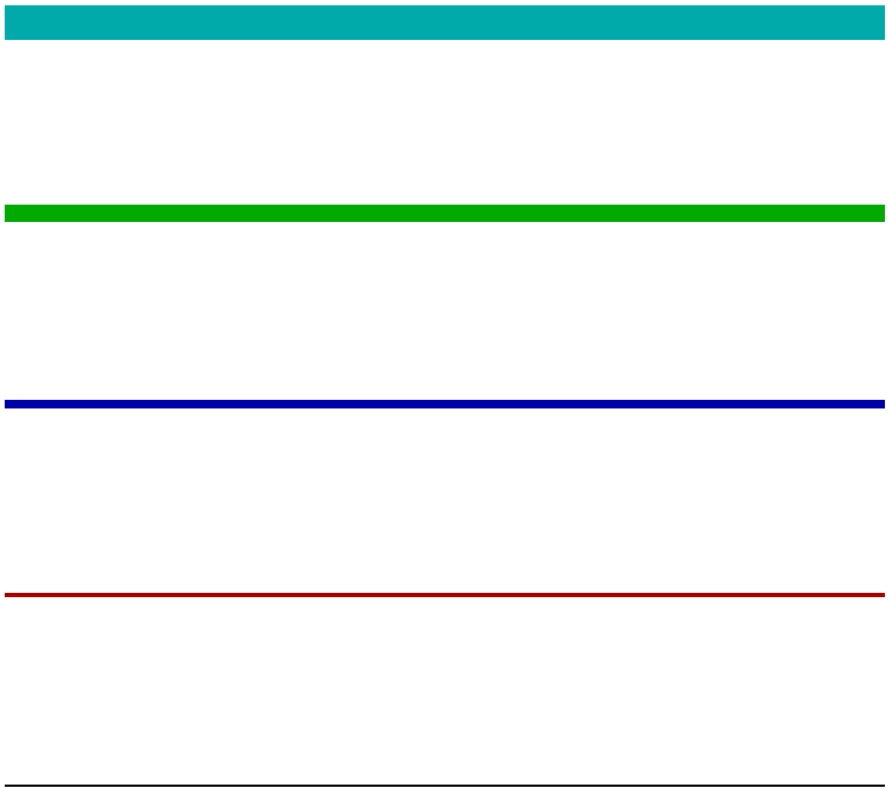
The x-legend color has been changed from blue rgb(0,0,170) to green rgb(0,170,0)

Line 36

The y-legend color has been changed from blue rgb(0,0,170) to green rgb(0,170,0)

15.6.2 Lines and Curves

Consider the next plot with five lines, each in a different color and line width.



This plot can easily be created by reading the SimFIT metafile `lines.metafile` into SimFIT program **simplot** or SimDEM program **simdem70**.

Note that in this plot the lines were created by drawing a polyline between two points which could easily have been drawn as a simple line. However a polyline was drawn as it would be more usual to change the line type, thickness, and color for a smooth curve which would then have made it more difficult to comprehend with the presence of a large number of coordinates, whereas only the end points are needed for a straight line.

The point is that, in order to search for a graphical object in a SVG file, the the markup code would be used. For instance, searching for the text string `<polyline points=` in the file `lines.svg` locates the following code on a single line, but here broken up into separate tokens for clarity.

```
<polyline points="196.81,823.62 1095.69,823.62"
style="fill:none;stroke:rgb(0,0,0) ;stroke-width:1.84"
stroke-linecap="butt"
stroke-linejoin="round" />
```

To understand this code the following summary is presented.

1. `<polyline points= ... />`

   This markup section starting with < and ending with /> defines all the properties of the polyline.

2. `style=`

   Here all the details of the curve to be drawn are to be found.

3. `:rgb(0,0,0)`

   This color convention is exactly as used for HTML, where red, blue, and green color components are defined on a scale from 0 to 255.

4. `stroke width:`

   This defines the line width in pixels.

5. `stroke-linecap=`

   This defines the way the ends of curves are finished off.

6. `stroke-linejoin=`

   This is where the way that the sections of the polyline making up the curve are to be connected together is defined.

In fact the code for drawing all five lines is as follows , where dots . . . are used to indicate the text omitted for clarity.

```
... style="fill:none;stroke:rgb(0,0,0)     ;stroke-width:1.84"  ...
... style="fill:none;stroke:rgb(170,0,0)   ;stroke-width:2.00"  ...
... style="fill:none;stroke:rgb(0,0,170)   ;stroke-width:4.00"  ...
... style="fill:none;stroke:rgb(0,170,0)   ;stroke-width:8.00"  ...
... style="fill:none;stroke:rgb(0,170,170) ;stroke-width:16.00" ...
```

Note that the order of these commands is in the direction from bottom to top, so that the first line (stroke-width:1.84) refers to the bottom line, while the last command (stroke-width:16) refers to the top line.

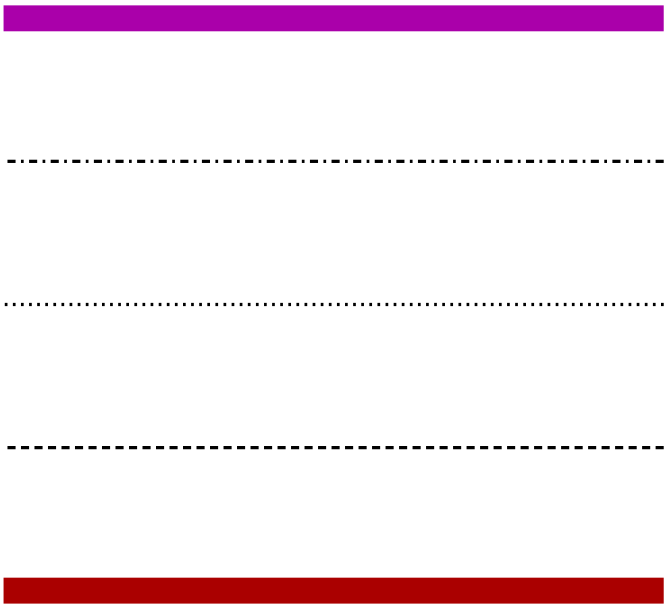
Another change that is often required is to swap between solid lines and dashed, dotted, or dash–dotted lines which requires the definition of a `stroke-dasharray` parameter as follows.

- `stroke-dasharray="18.00,12.00"` for dashed lines
- `stroke-dasharray="6.00,12.00"` for dotted lines
- `stroke-dasharray="18.00,12.00,6.00,12.00"` for dash–dotted lines

To demonstrate this technique, note that replacing this section of `lines.svg` by the next section

```
:rgb(170,0,0)   ;stroke-width:16.00"
:rgb(0,0,0)     ;stroke-width:2.00" stroke-dasharray="18.00,12.00"
:rgb(0,0,0)     ;stroke-width:2.00" stroke-dasharray="6.00,12.00"
:rgb(0,0,0)     ;stroke-width:2.00" stroke-dasharray="18.00,12.00,6.00,12.00"
:rgb(170,0,170) ;stroke-width:16.00"
```

generates the following graph with the dashed, dotted, and dash–dotted lines colored black.



15.6.3 Character Strings and Fonts

A common need is to reposition a character string, to edit the string, to change the font size, or to change the colour. Consider, for instance, the following diagram.

Here is the first character string with Arial/Helvetica at size 1.2, but broken into separate tokens for clarity.

```
<text x="289.00" y="62.48"
font-family="ARIAL"
font-size="23px"
font-weight="400"
fill="rgb(0,0,0)"
>This is Arial/Helvetica size = 1.2
</text>
```

If this is understood then editing such a SVG file will be simple. Here is what the rules are.

- `<text ...`

  This indicates the start of a new character string which includes several self–evident definitions such as these.

  - `font-family`
  - `font-size`
  - `font-weight`
  - `fill`

- `This is Arial/Helvetica size = 1.2`

  This is the actual string itself that is going to be displayed

- `</text>`

  This indicates the end of the instructions to display the string

The command to rotate 90 degrees, again with tokens separated for clarity, is as follows.

```
<text x="232.00" y="705.00"
font-family="ARIAL"
font-size="25px"
font-weight="700"
transform="rotate(270.00,232.00,711.00)"
font-style = "oblique"
fill="rgb(0,140,0)">
Arial BoldOblique rotated 90 degrees</text>
```

The following files are distributed with the SimFIT package in order to understand the previous details.

1. `lines.metafile, new_lines.metafile, fonts.metafile`

   These can be used to generate the figures using the SimFIT program **simplot** or the SimDEM program **simdem70**

2. `lines.eps, new-lines.eps, fonts.eps`

   PostScript graphics files

3. `lines.svg, new-lines.svg, fonts.svg`

   SVG graphics files

Note that, to edit text strings containing non–ASCII characters as in the lower strings using Symbol font, an editor supporting UTF8 must be used.

15.7 SVG: Creating collages

Sometimes it is required to combine several SVG files together to create a collage, i.e., a single SVG file containing subgraphs arranged into fixed or arbitrary positions. To do this, subsidiary SVG files are input into program **EditSVG**, then rearranged and scaled as necessary. When satisfied, program **EditSVG** can be used to output the composite graph as a SVG file (*.SVG).

Note that, after constructing a collage, program **EditSVG** also provides the facility to output re–build files (*.ISVG). These simply contain a list of SVG files used to compose the collage along with positions, scaling, and flags to indicate if white space borders are to be clipped from graphs, etc.

Such collages can be classified into several types as follows.

- A fixed or strict collage

  Here all the subgraphs have the same size and shape. For instance, all square, or all in landscape format, or all in portrait aspect ratio.

- An arbitrary or free–style collage

  In this case the subgraphs can be of arbitrary size and shape.

- Using ribbon graphs

  On occasions graphs created in square, portrait, or landscape, format are too compressed and it is necessary to apply differential stretching into a non standard format. This involves scaling the length of lines without altering the aspect ratio of the fonts used in titles, legends or plot labels. For instance with dendrograms or forest plots.

Several collages from the SimFIT tutorials section concerning SVG are now shown to illustrate some typical possibilities.

- Collage 1.

  Freestyle non–overlapping type showing a collection of arbitrary mathematical equations and chemical formulas.

- Collage 2.

  Freestyle inlay type illustrating how to combine LaTeX maths with visual display of data.

- Collage 3.

  Freestyle inlay type illustrating how to combine a LaTeX chemical scheme and graph with data and best-fit curves.

- Collage 4.

  Strict type displaying illustrations from the SimFIT tutorials SVG section.

- Collage 5.

  Differential stretching type illustrating how to stretch the x or y axes while maintaining constant aspect ratios for characters required for ribbon graphs.

15.7.1 Collage 1: Miscellaneous LaTeX examples

$$\begin{matrix} 0 & 1 \\ 1 & 0 \end{matrix} \quad \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$$

$$\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \quad \begin{Bmatrix} 1 & 0 \\ 0 & -1 \end{Bmatrix}$$

$$\begin{vmatrix} a & b \\ c & d \end{vmatrix} \quad \begin{Vmatrix} i & 0 \\ 0 & -i \end{Vmatrix}$$

$$\sqrt{1+\sqrt{1+\sqrt{1+\sqrt{1+\sqrt{1+\sqrt{1+x}}}}}}$$

$$\iint\limits_V \mu(v,w)\,du\,dv$$

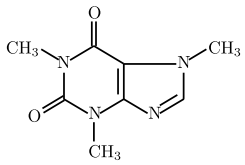
$$\iiint\limits_V \mu(u,v,w)\,du\,dv\,dw$$

$$\int\limits_V \cdots \int \mu(z_1,\ldots,z_k)\,\mathbf{dz}$$

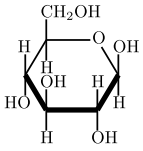
$$\lim_{x\to 0} \frac{\sin^2(x)}{x^2} = 1$$

$$\varliminf_{n\to\infty} |a_{n+1}|/|a_n| = 0$$

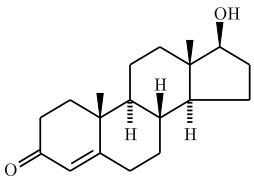
$$\varinjlim (m_i^\lambda \cdot M)^* \leq \varprojlim_{A/p\to\lambda(A)} A_p \leq 0$$



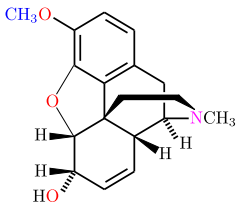
Caffeine



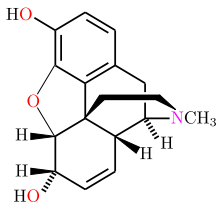
Glucose



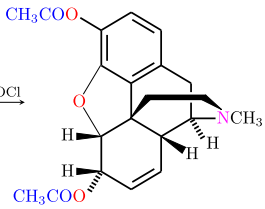
Testosterone



Codeine



Morphine



Heroin

15.7.2 Collage 2: LaTeX maths

$$S_n = \sum_{k=1}^{n} X_i, \quad k = 0, 1, 2, \ldots, n$$

$$\text{Binomial Distribution: } P(S_n = k) = \binom{n}{k} p^k (1-p)^{n-k}$$

$$E(S_n) = np$$

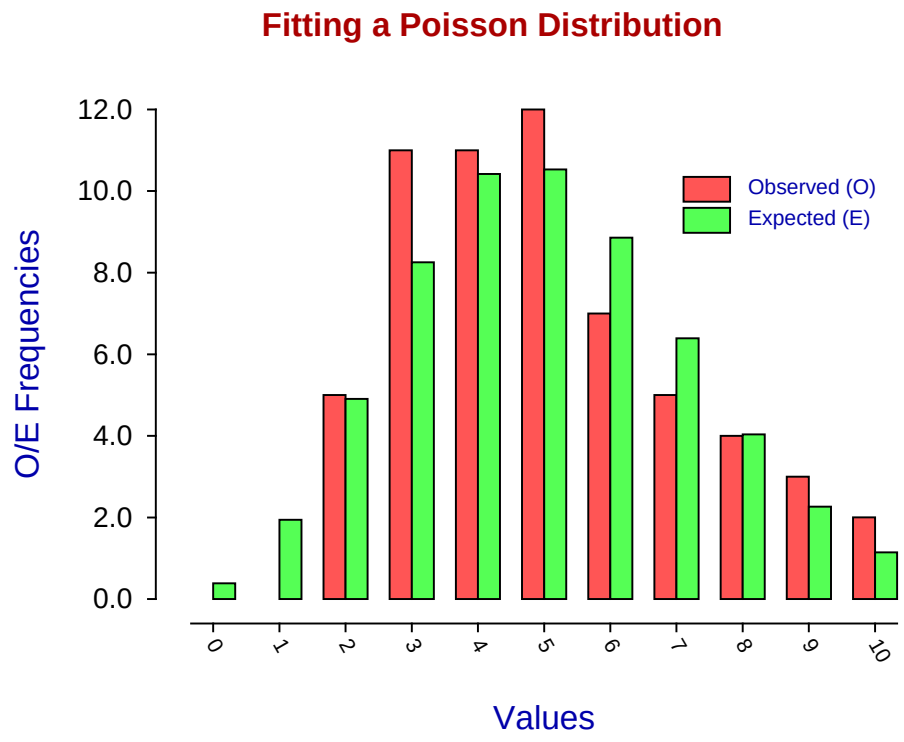
$$V(S_n) = np(1-p)$$

$$\text{Poisson Distribution: } P(Y = k) = \frac{\lambda^k}{k!} \exp(-\lambda), \quad y = 0, 1, 2, \ldots,$$

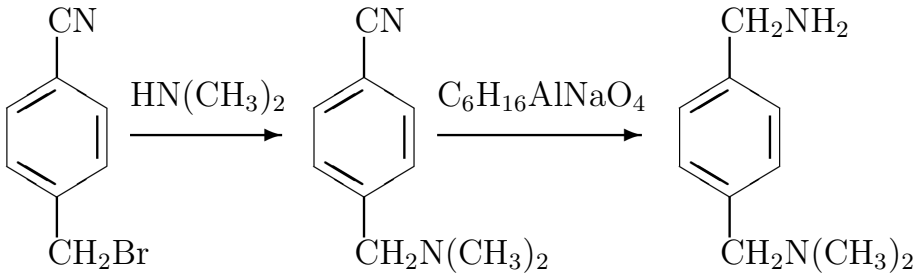
$$E(Y) = \lambda$$

$$V(Y) = \lambda$$

$$\lim_{n \to \infty, p \to 0} \binom{n}{k} p^k (1-p)^{n-k} = \frac{(np)^k}{k!} \exp(-np), \quad np > 0$$



15.7.3 Collage 3: LaTeX chemistry

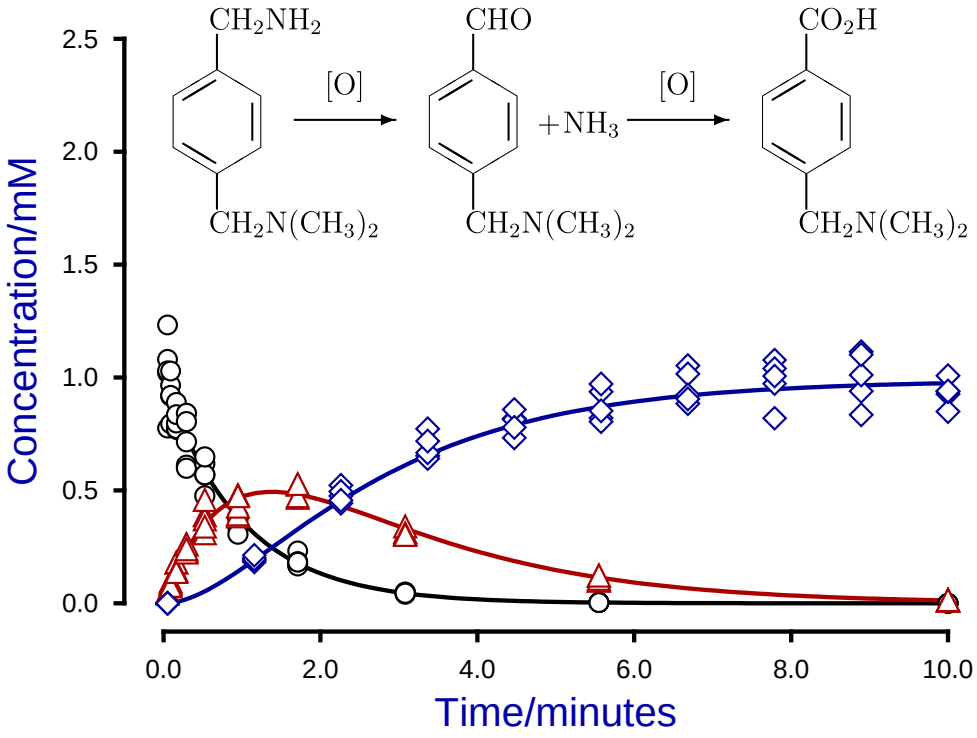


Chemical synthesis of p-dimethylaminomethlbenzylamine

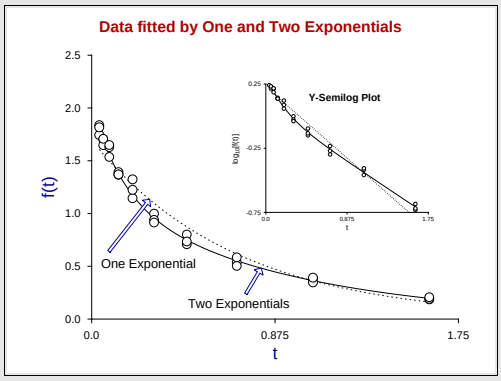
using

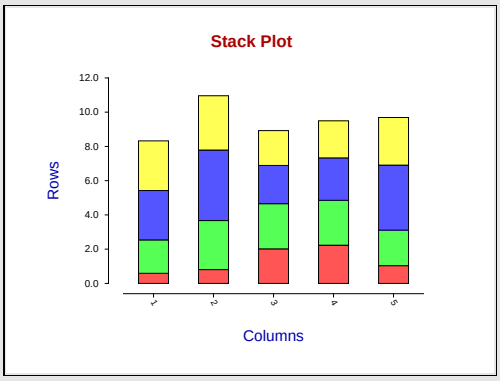
p-cyanobenzyl bromide, dimethylamine and Red-Al

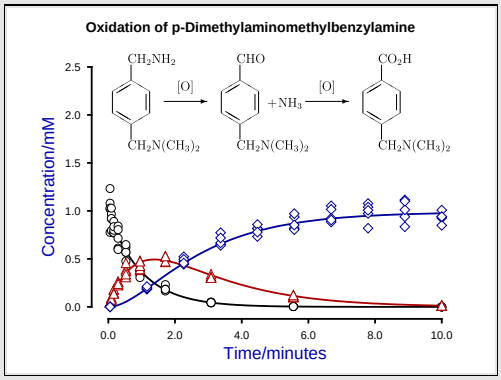
Oxidation of p-Dimethylaminomethylbenzylamine

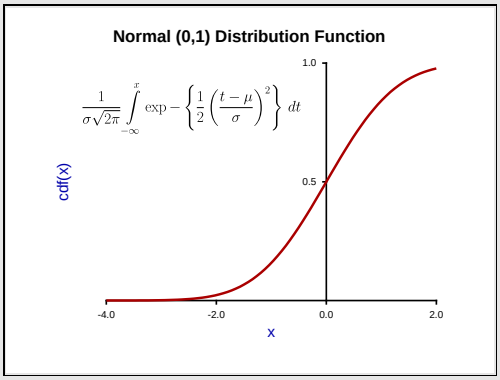


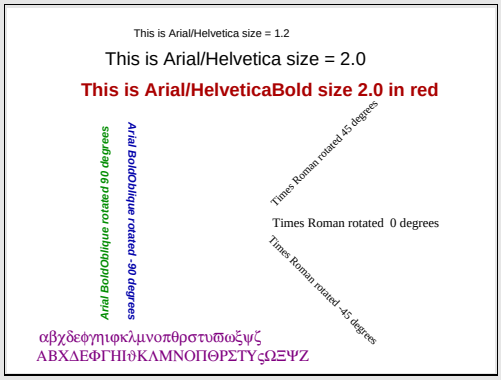
15.7.4 Collage 4: Tutorial examples

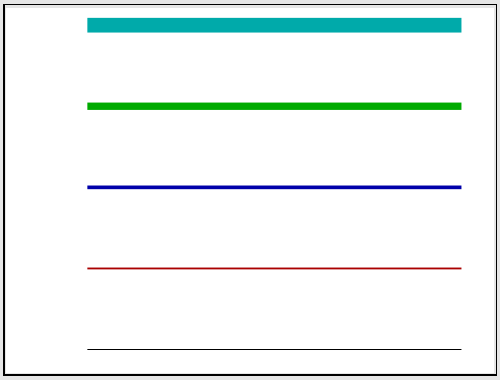












$$S_n = \sum_{k=1}^{n} X_i, \quad k = 0, 1, 2, \ldots, n$$

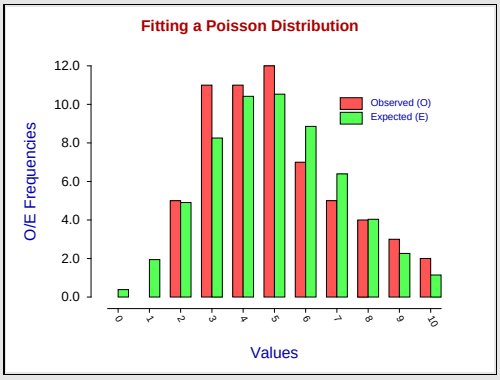
Binomial Distribution: $P(S_n = k) = \binom{n}{k} p^k (1-p)^{n-k}$

$$E(S_n) = np$$
$$V(S_n) = np(1-p)$$

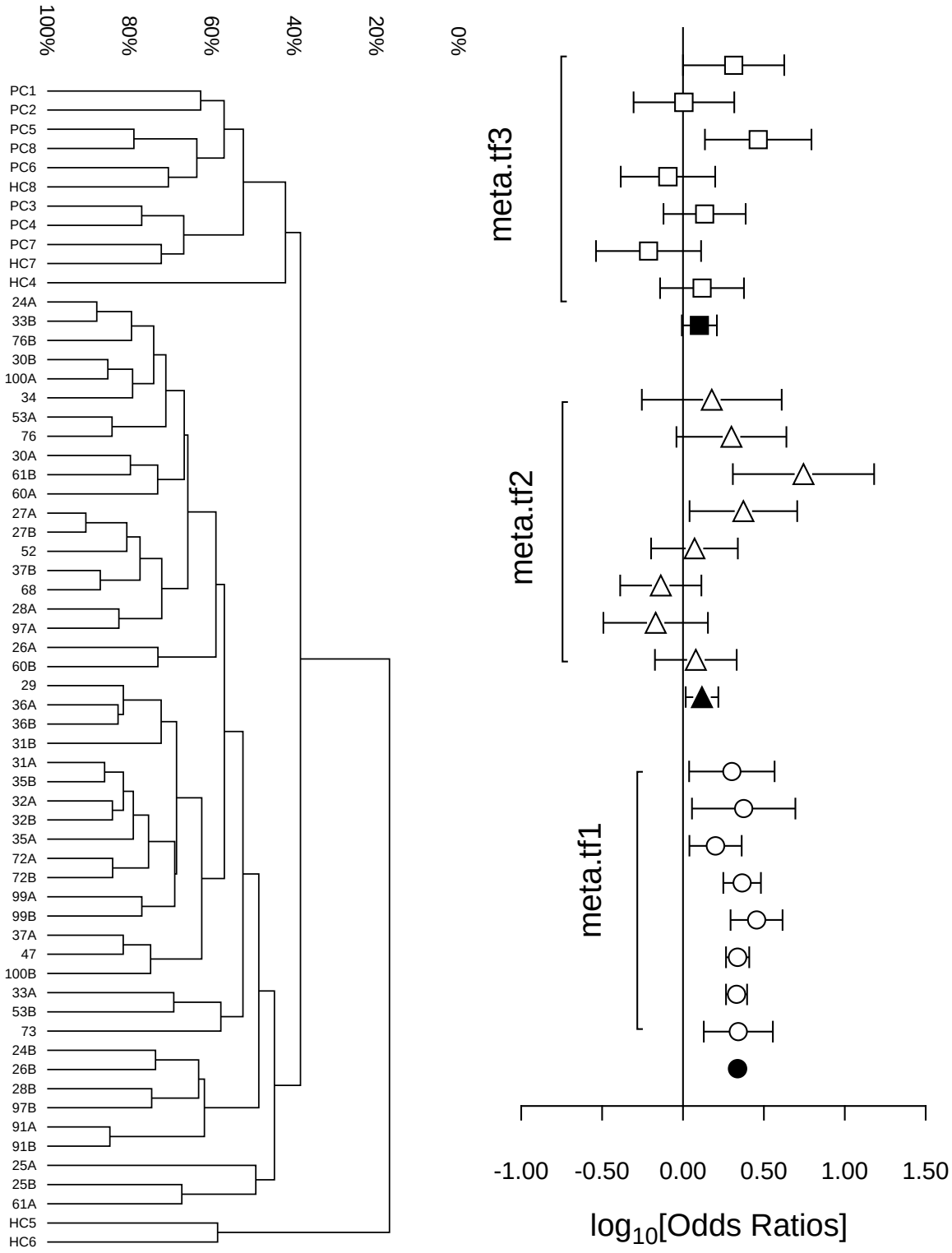
Poisson Distribution: $P(Y = k) = \frac{\lambda^k}{k!} \exp(-\lambda), \quad y = 0, 1, 2, \ldots,$

$$E(Y) = \lambda$$
$$V(Y) = \lambda$$

$$\lim_{n \to \infty, p \to 0} \binom{n}{k} p^k (1-p)^{n-k} = \frac{(np)^k}{k!} \exp(-np), \quad np > 0$$



15.7.5 Collage 5: Differential scaling to create ribbon graphs



15.8 SVG: Differential scaling examples

Usually scientific graphs are in landscape, square, or portrait aspect ratios as they are intended for inclusion in documents. However there many occasions where the data to be displayed are very extensive which would lead to overcrowding of curves or overlapping of labels and plotting symbols. Examples would be the display of dendrograms, forest plots, time series, or spectra where it would be useful to display data with extreme aspect ratios and attached scroll bars to scan the graph vertically or horizontally, as with chart–paper. Fortunately internet graphics using scalable vector graphics (SVG) allows this as all browsers support the SVG format, and SimFIT provides facilities to stretch out overcrowded SVG graphs.

Unfortunately, where a graph is in bitmap, compressed bitmap, or vector format with landscape, square, or portrait aspect ratio, it is not possible to merely stretch the graph as this would lead to pixellation, and distortion of characters and plotting symbols, e.g., circles becoming ellipses, squares becoming rectangles, etc., so a special type of stretching procedure is required.

SimFIT allows users to sculpture plots in advanced 2D format by editing symbols, line–types, labels, colors, titles and legends, but then to save in SVG format followed by interactively applying differential scaling until a satisfactory aspect ratio has been achieved before saving to a new SVG file which will automatically be displayed with scroll bars in browsers.

So differential scaling as defined in this way requires several steps in SimFIT as follows.

1. Sculpture the graph in advanced 2D graphics
2. Choose the [SVG] option to view the current plot as SVG
3. Select values for X_scale and Y_scale and view the outcome
4. When satisfied save as a new differentially scaled SVG file

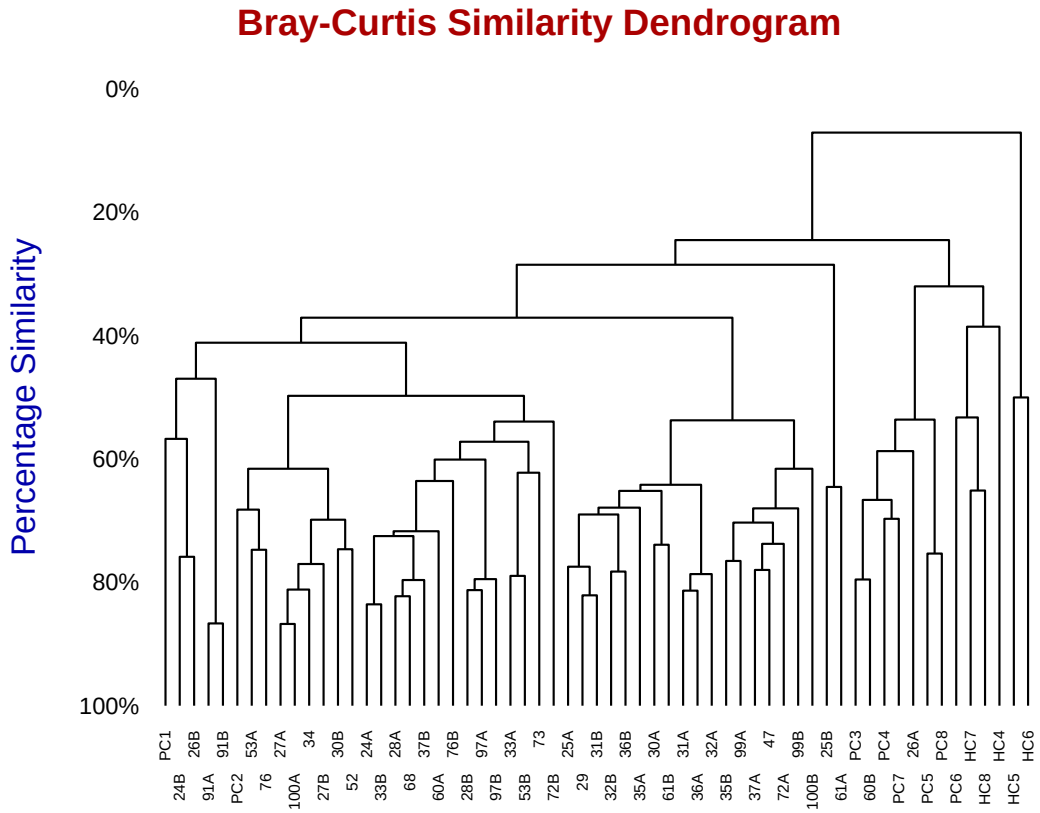
Note that the differentially scaled SVG files will have the following characteristics.

- The fonts and inter–character spacings will remain unchanged
- The line type and thickness will remain unchanged
- The plotting symbols will maintain their aspect ratios
- Only the white space between the lines, symbols and labels will change
- However there are likely to be unexpected effects if extreme scaling is used. For instance.
  - With horizontally stretched graphs a centralized title or X–legend may only become visible after horizontal scrolling.
  - With vertically stretched graphs a centralized Y–legend may only become visible after vertical scrolling.
  - In extreme cases like this it is best to delete the offending title or legend and add a new text string in an appropriate position, i.e., close to the top left of the plot.
  - Moving text, arrows, or information panels by eye with the red arrow will never be precise and this will become increasingly obvious at as the scaling increases. Positioning can be improved by using the graticule function, i.e., the mesh of intersecting lines created from the [Style] button.

Some of the unexpected effects and avoidance of such issues will be clear by detailing several examples that will be discussed next and, to understand this material, it should be combined with viewing the page `https://simfit.org.uk/svg.html`.

15.8.1 A normal dendrogram

This is a very simple example where differential scaling is scarcely necessary but, as it does illustrate several points, creation of the next plot requires some explanation.



The steps required to generate this default plot are as follows.

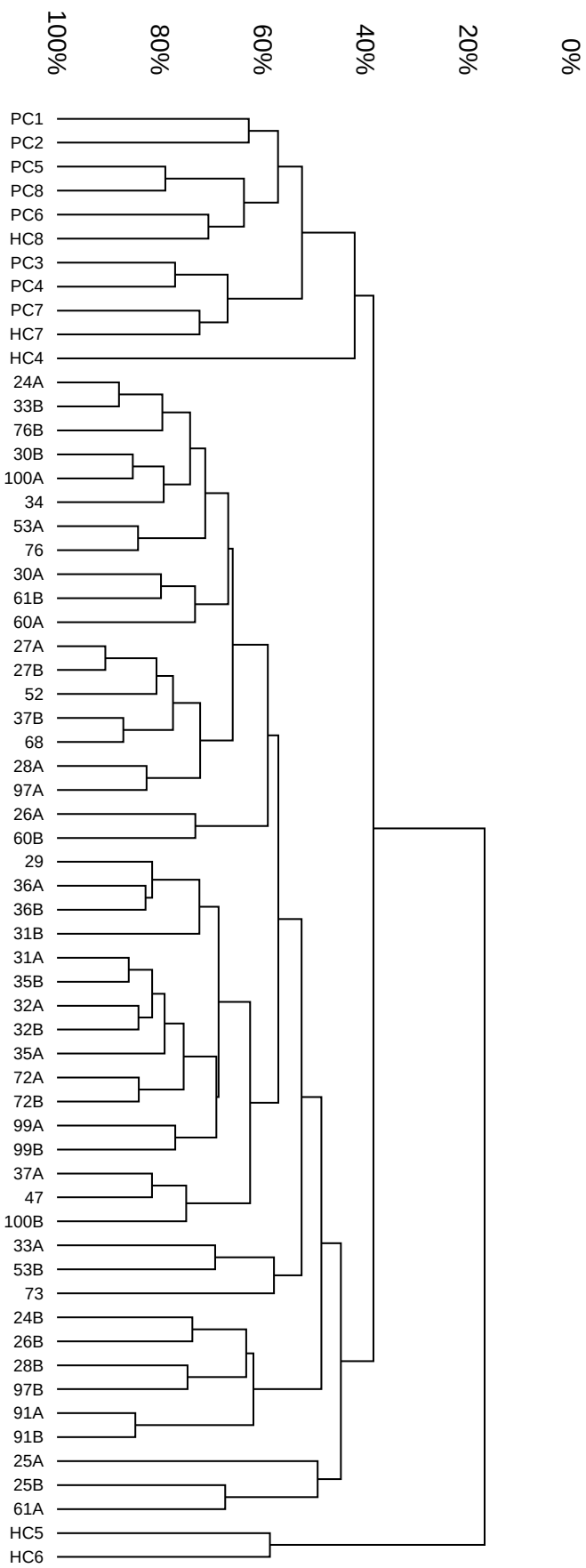
1. The SimFIT test data file `cluster.tf2` contains a multivariate data set that can be used to generate a dendrogram either by input into the SimFITprogram **simstat** then choosing multivariate statistics to create a distance matrix with a chosen metric followed by plotting a dendrogram with labels, or by input into **simplot** then opening the option to create statistical graphs.

2. The type of dendrogram required can be selected as the Bray-Curtis similarity type, which is often used for such biological data. This would, of course, always be a user–selected decision that would depend on the data and either statistical arguments or simply the visual appearance preferred.

3. The title and legend would normally be edited at this stage as required.

4. It should be noted that, with this example, SimFIT has automatically chosen to create a double label system with rotated X axis labels in order to create a legible X–axis labeling system without the labels overlapping.

The next plot illustrates the effect of differential X–axis scaling, where the title and Y–legend have been suppressed in order to display the graph in as large a size as is possible in a document.



To generate this particular figure these steps were taken.

The dendrogram was created in SimFIT advanced 2D graphics

The title was suppressed.

The *Y*–legend was suppressed.

The double label threshold was increased using the `[Labels]`option followed by choosing to edit the *X*-axis labels
This allows the labels to be displayed on one line.

The font size for the *Y*–axis numbers and the *X*–axis labels was increased.

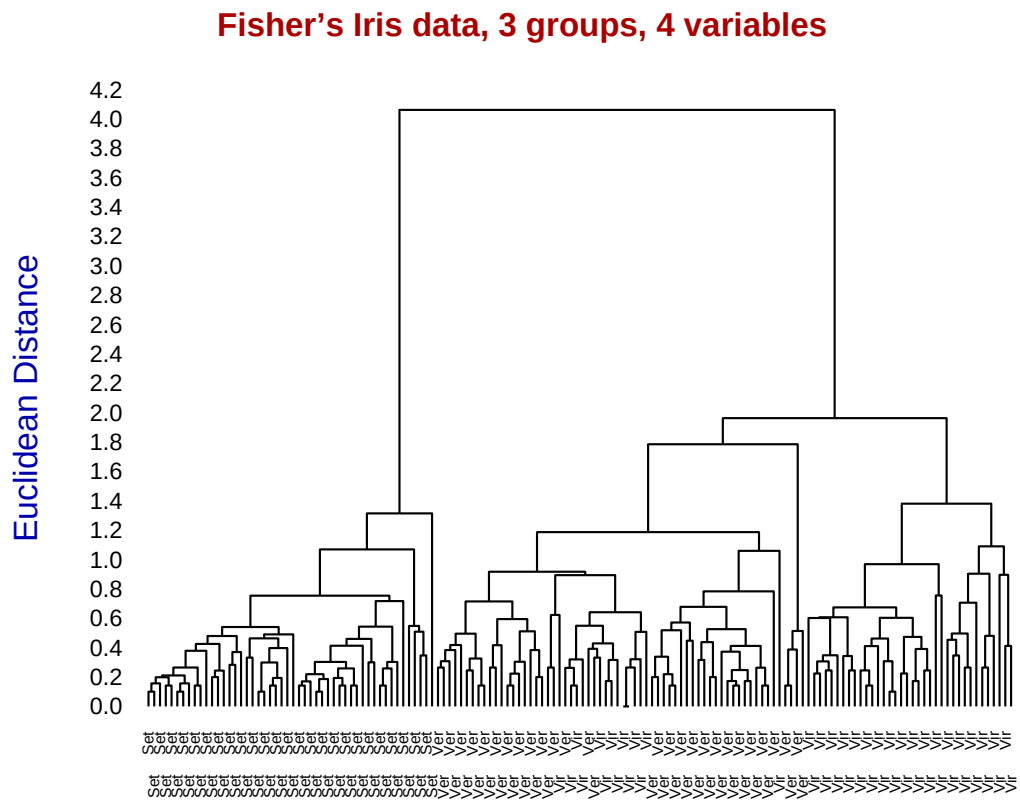
To improve legibility the *X*-axis scaling factor X_scale was then increased to 3.
Of course the graph becomes too large to display in this document so the scrolling version with title and *Y*–legend has to be observed using the [SVG] option from the main page at
`https://simfit.org.uk`.

In order to view with the labels in standard orientation rather than rotated it would then be necessary to increase the *X*–scaling factor to a larger value than 3 so as to maintain legibility and prevent labels overlapping.

The re–scaled and edited graph was saved to a new file which is displayed to the left, after rotation.

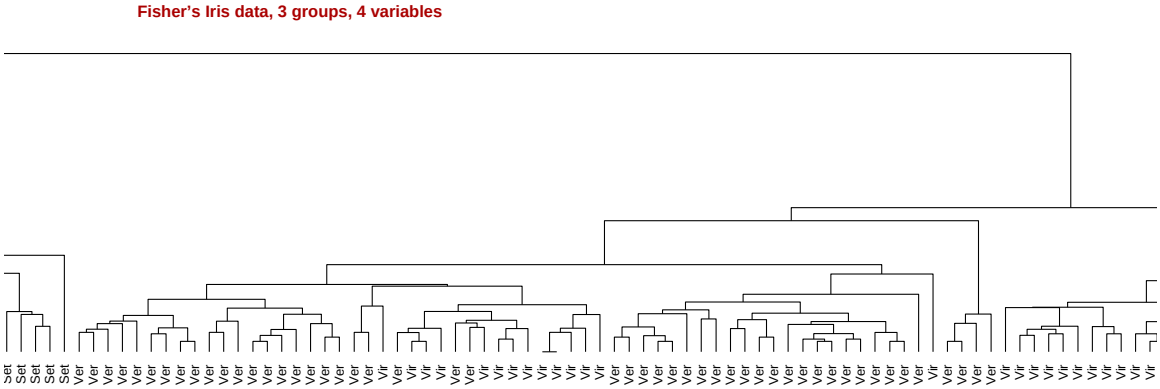
15.8.2 A crowded dendrogram

Proceeding as before but using the SimFIT test file `iris.tf1` displays the famous Fisher iris multivariate data as the following dendrogram.



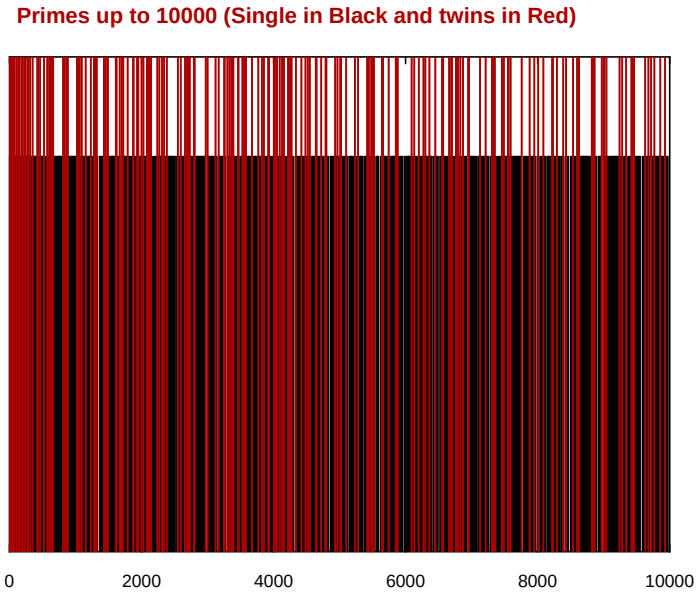
Here the labels are too crowded to read easily but placing all the labels on the same line instead of on double lines then stretching by using X_scale = 5 creates a much more legible dendrogram as will be seen from the SimFIT website. Unfortunately this is then too wide to display in full in this document without serious reduction.

However, using the SimFIT Postscript technique to clip a section out of such an expanded graph (which is described in the reference manuals) does allows the display of an arbitrary section clipped out of the center of the stretched graph as shown next (after some reduction).

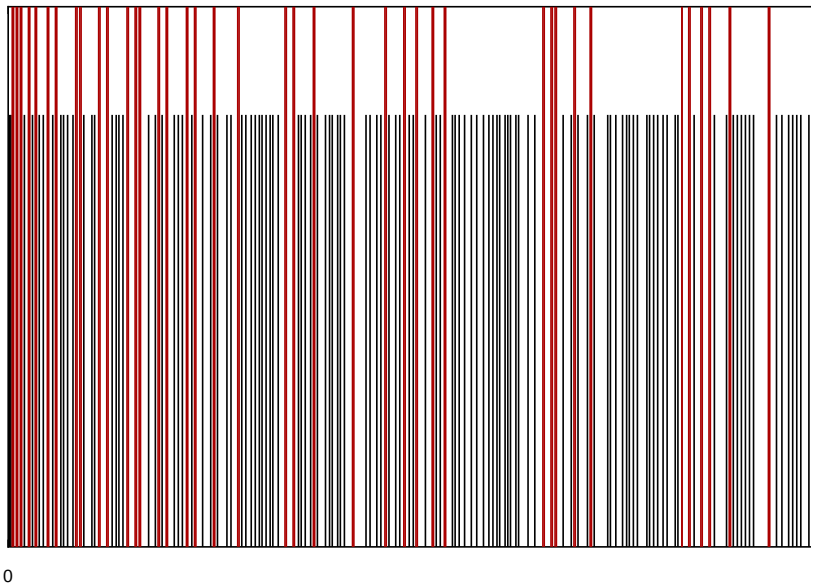


15.8.3 An extremely crowded plot

To create the chart–paper type of plot for primes displayed on the website the test files `primes.tf1` and `primes.tf2` were input into SIMFIT program **simplot** followed by editing the Y–axis to have range (0, 1.25), the data being plotted with no lines but with solid barchart–type symbols colored black for `primes.tf1` and red for `primes.tf2`, to create the next figure (after suppressing X and Y legends and Y labels and using integers as X–labels).



Using the SIMFIT Postscript facility described for the previous example to stretch by a factor of 10 and then clip out the start of this graph to an EPS file (shown below) indicates the effect of differential scaling to separate out the vertical bars from the solid mass of color in the plot (shown above) without differential scaling. This is seen more clearly in the scrolling example stretched by a factor of 20 on the website.



Index