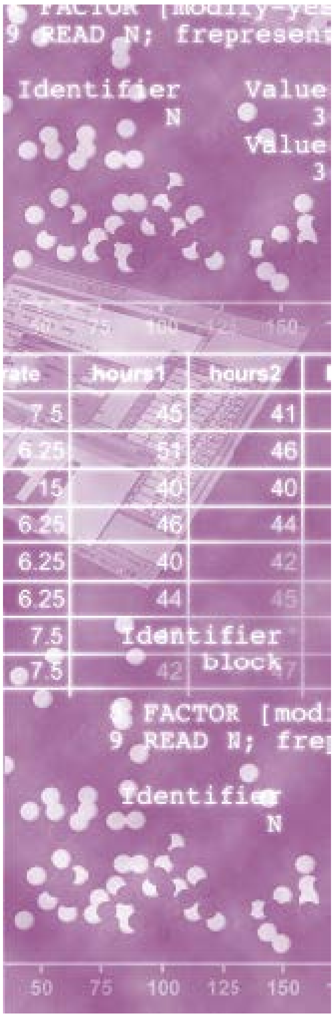
WAGENINGEN UR



Graphics Guide
GenStat for Windows 15th Edition

Jac T.N.M. Thissen

Biometris

Biometris 23.11.12

November 2012

Biometris quantitative methods in life and earth sciences

Biometris is the integration of the Centre for Biometry of Plant Research International and the Department of Mathematical and Statistical Methods of Wageningen University. Biometris, part of Wageningen University and Research center (Wageningen UR), was established on 20th June 2001.

For more information please visit the website: http://www.biometris.wur.nl

November 2012 Report number 23.11.12

CONTENTS

1 Introduction

The official documentation on the graphical display with GenStat can be found in:

Chapter 3 of the Guide, *Introduction to GenStat for Windows (15th Edition),* pages 55-62, (Reference 1)
Chapter 6 of the *Syntax and Data Management Guide*, pages 301-382, (Reference 2)
Reference Manual Part 1: Summary, pages 17-19. (Reference 3)

The guides, IntroGuide.pdf and SyntaxGuide.pdf, and the manual, Refman1.pdf, can be found in the Doc-subdirectory of Gen15Ed. They can also be consulted via clicking the Help button on the menu bar within GenStat:

Help → GenStat Guides → Introduction to GenStat for Windows…,
Help → GenStat Guides → Syntax and Data Management…, and
Help → Reference Manual → Summary …

Tip

In the documentation there is frequently a reference to special bars, panes and windows. In Appendix 7 there is an explanation.

In this document a sequence like Help → Reference Manual → Summary … means:
Click on the first word of the sequence (Help) on the menu bar, then click on the second word of the sequence (Reference Manual), an option on the Help submenu, then click on the third word of the sequence (Summary...), an option on the Reference Manual submenu, etc.

Furthermore, there is a lot of information in the Help pages. For instance, there is a table of colours with pre-defined names (see Paragraph 4.9.9) which can be found in the Help Search with “Graphics Colours”. Also there is information in the description of the PRINT directive about how to use typesetting commands in strings to represent Greek and mathematical symbols (see Appendix 3).

The aim of this document is to bring all this information together and to show a wide variety of examples to illustrate the graphical possibilities of GenStat. The next part of this Introduction comes from Reference 3; a main part of Chapter 2 , The GenStat graphics wizard, comes from Reference 1; Chapter 3, the GenStat graphics viewer comes from the Help; a main part of Chapter 4, Graphical display, comes from Reference 2; and Chapter 5 contains a lot of examples illustrating the various aspects of modifying the settings of options and parameters.

For not-experienced users of GenStat the Graphics Wizard in Chapter 2 is most useful but we will see that knowledge of the command language is a great help to produce the more complicated examples.

Tip

In Chapter 5 you will find remarks about the way the figures are inserted into this document, especially the figures of the examples.

GenStat can produce graphical output in **two distinctively different styles**.

These are ***line-printer* graphics** and ***high-resolution* graphics**. As the name suggests, **line-printer graphics** are designed for printing on ordinary printers, and are also suitable for display on terminals and PC screens. Thus no special equipment is required; also the plots form an integral part of the GenStat output, and can thus be interspersed with other results during the analysis of the data. With line-printer graphics, the standard character set, made up of letters, digits and punctuation characters, is used to produce a graphical representation of the data. This will be of *low resolution*, typically 24 rows by 80 columns for screen display, 132 by 48 or 80 by 60 for a printer; but this is often adequate for a quick assessment of the data, or for checking the assumptions of an analysis. Histograms, graphs and contour plots can be produced in this basic style. The relevant 3 directives and 1 procedure are:

LPGRAPH	produces scatter plots and line graphs
LPHISTOGRAM	plots histograms
LPCONTOUR	plots contour maps of two-way arrays of numbers
STEM	plots a stem-and-leaf chart

High-resolution graphics provide a more attractive alternative. Lines and points are plotted with far greater precision, and a wider range of plotting symbols can be used to enhance the output. Also most devices allow the use of colour. A wide range of plots can be produced: graphs and histograms in two or three dimensions, contour plots, shade diagrams, three-dimensional surfaces and pie charts. High-resolution graphics can be produced interactively on graphics terminals, workstations or PC screens. Some of these support graphical input, which can be used for example to allow interactive identification of outliers. Plots can be also saved in files using standard formats that are suitable for plotters or laser printers or for importing into wordprocessed documents.

GenStat *for Windows* has a **Graphics Wizard** (see Chapter 2) that allows you to select a high-resolution graph and customize its appearance. You can also modify many aspects of the graph, such as colours, line styles, plotting symbols, fonts and axes, interactively after it has been plotted.

However, even here you may find it useful to know the commands, in case you want to study the input log or to develop new types of display. The directives for high-resolution graphics have **two main purposes**. There are those that define the **"graphics environment"** for subsequent plots, and those that **do the plotting**. The default environment, set up at the start of a program, will often be satisfactory. However, you can **modify the environment** (see Paragraph 4.9) to customize the plots using the following commands:

DEVICE	switches between graphics devices
SETDEVICE	opens a graphical file on the basis of a file extension
FRAME	defines the positions and appearance of the plotting windows within the graphics frame
FFRAME	forms multiple windows in a plot-matrix for highresolution graphics
XAXIS	defines the x-axis in a graphical window
YAXIS	defines the y-axis in a graphical window
ZAXIS	defines the z-axis in a graphical window
AXIS	defines an oblique axis for high-resolution graphics
PEN	defines properties of graphics "pens"
GETRGB	provides a standard sequence of colours, defined by the initial defaults of the GenStat pens
COLOUR	defines the red, green and blue intensities to be used for the GenStat colours for certain graphics devices (is not longer used; only in combination with `SET [CMETHOD=standard]`)
DCOLOURS	forms a band of graduated colours for graphics
DHELP	provides information about the graphics environment
DKEEP	copies details of the graphics environment into GenStat data structures
DLOAD	loads the graphics environment settings from an external file
DSAVE	saves the current graphics environment settings to an external file

The directives for **plotting high-resolution graphs** are:

DGRAPH	produces scatter plots and line graphs
D3GRAPH	plots a 3-dimensional graph
DHISTOGRAM	plots histograms
BARCHART	plots bar charts
DCONTOUR	plots contour maps
DSHADE	plots a shade diagram of three-dimensional data
DSURFACE	draws a perspective plot of a two-way array of numbers
D3HISTOGRAM	plots three-dimensional histograms
DPIE	plots pie charts
DBITMAP	plots a bit map of RGB colours
DCLEAR	clears a graphics screen
DSTART	starts a sequence of related plots
DFINISH	ends a sequence of related plots
DDISPLAY	redraws the current graphical display

Tip

Other graphical facilities are provided by **procedures in the Library**: in Appendix 2 they are divided by module.

2 The GenStat graphics wizard

It is often helpful to explore the data using graphical displays. These can be used to examine the structure of the data or to display its distribution. Many types of graph can be accessed directly as options of the Graphics menu on the menu bar. Alternative GenStat has as a graphics wizard that can help you choose the right graph.

We will use the wizard to investigate some data collected in 1990 to investigate changing levels of air pollution. The principal measurement is the amount of sulphur in the air each day, but there are also associated measurements: the strength and direction of the wind, and whether or not it rained. The data are available in the file Sulphur.gsh and can be read using the Data file option on the Load menu from the Data menu bar. The Data View pane, shown in Figure 2.1, lists the structures loaded from the file. (You can view the Data View pane by selecting the Data View option on the View menu or the Display option on the Data menu). The data consist of two variates and two factors, each containing 114 values.

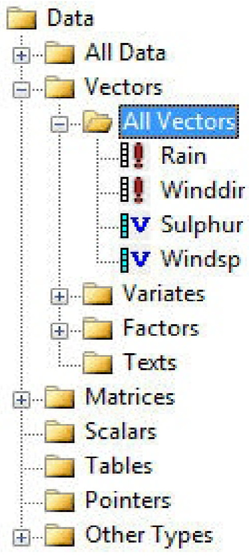


Figure 2.1

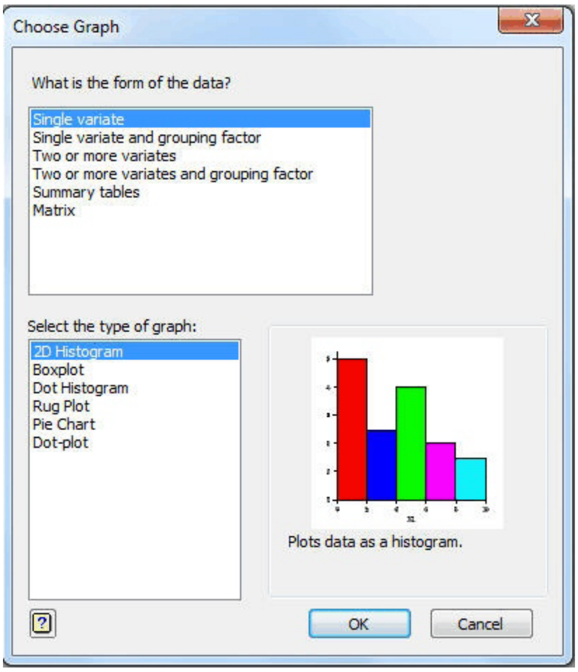


Figure 2.2

You open the wizard by selecting the Create Graph option from the Graphics menu on the menu bar. The first menu, shown in Figure 2.2, helps you to choose the graph. You indicate the type of data in the top half of the menu, and the corresponding choices are then shown in the bottom half. Here we want to investigate the (variate of) sulphur measurements, and have chosen to investigate their distribution using a histogram.

Clicking the OK button opens the menu shown in Figure 2.3.

The menu opens on the Data tab, where we can enter the data to be plotted. Here we have selected Sulphur in the Available data list, and clicked on the arrow to place it into the Data variates list.

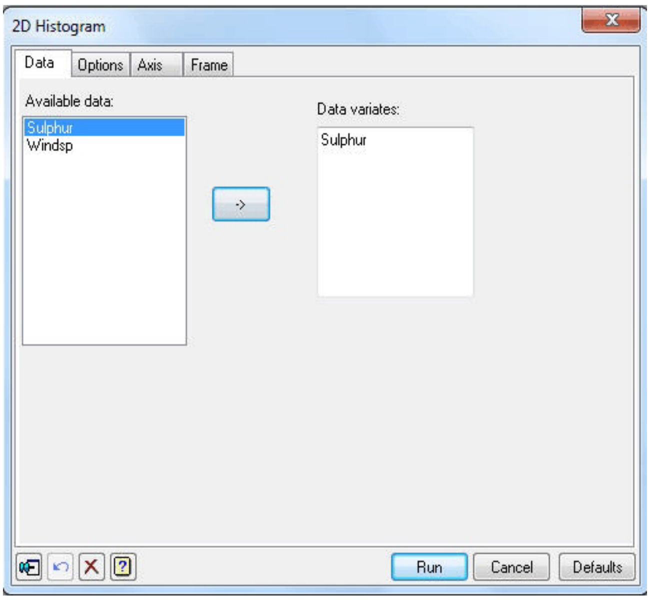


Figure 2.3

We now move to the Options tab in Figure 2.4, and enter a title, "Sulphur pollution", for the graph. We have selected the Use data values option for specifying the boundaries. This option lets GenStat select the number of groupings and their locations automatically. You can use the Number of groups option to specify a particular number of groups. The groups are then defined by intervals of equal width, spanning the range of values of the variate. Alternatively, if you want to have other intervals, you can define the boundaries explicitly by selecting the Limits option and entering either a variate containing the boundary values, or the list of values themselves.

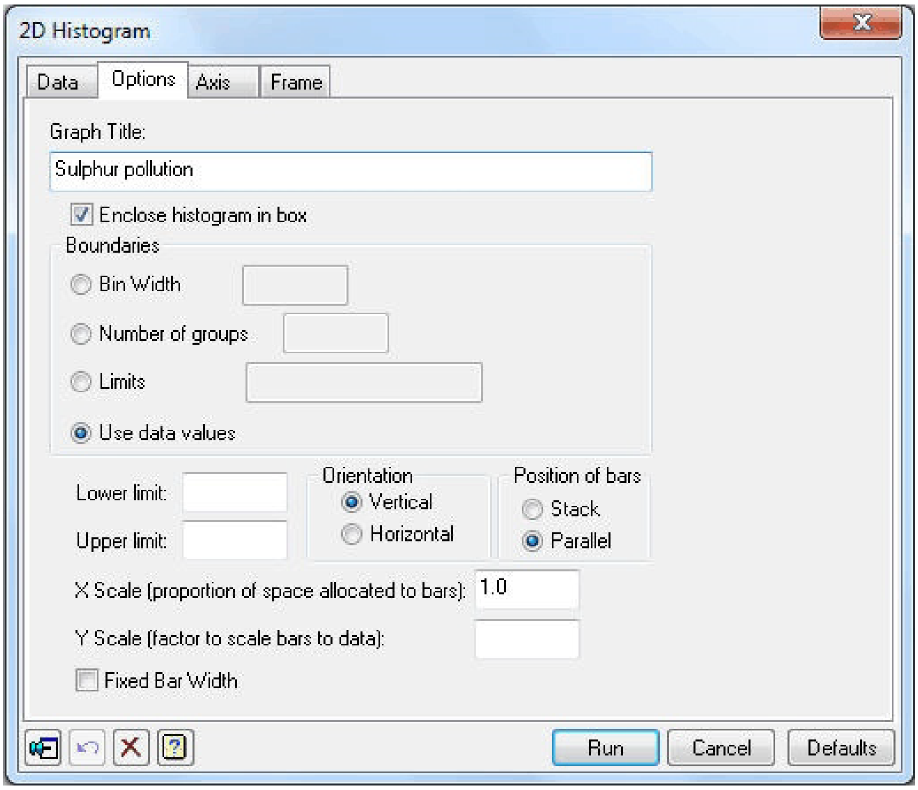


Figure 2.4

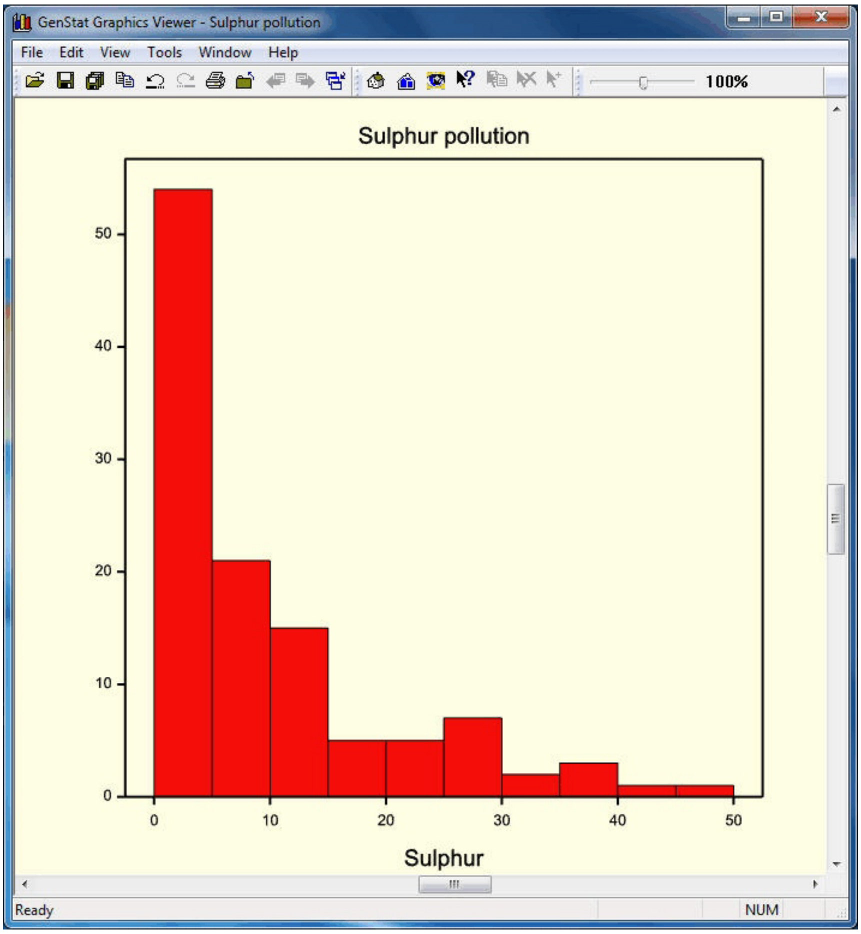


Figure 2.5

Clicking the Run button produces the histogram shown in Figure 2.5. This shows the numbers of observations in successive equal-width categories of the sulphur scale. Clearly sulphur has a skew distribution: there are many days with little or no sulphur in the air, and then decreasing numbers in successive categories with more and more sulphur.

Many statistical studies are concerned not with single variables, but with the relationships between several variables. With the pollution it is natural to ask questions like "Is there any effect of wind speed on the sulphur level?"

The most effective way to begin answering a question like this is generally to draw a *scatter plot* or *point plot*. This can be done by selecting Two or more variates option at the top of the Choose Graph menu, and then 2-D Scatter Plot as the type of graph. Clicking on OK opens the menu on the Data tab as shown in Figure 2.6. We wish to plot the sulphur levels against the wind speeds. So we enter Sulphur as the Y variate, and Windsp as the X variate.

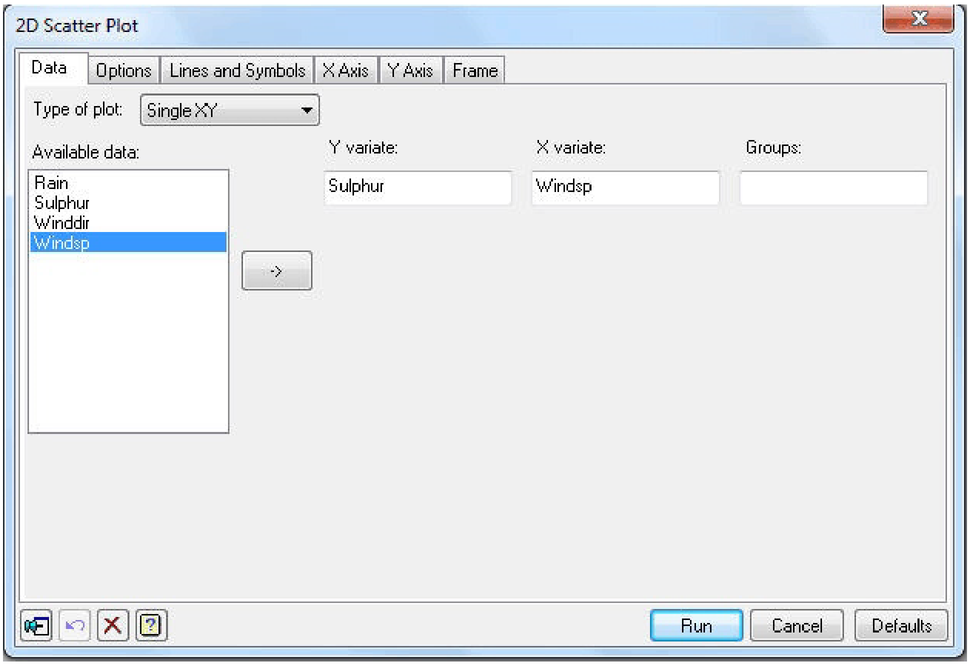


Figure 2.6

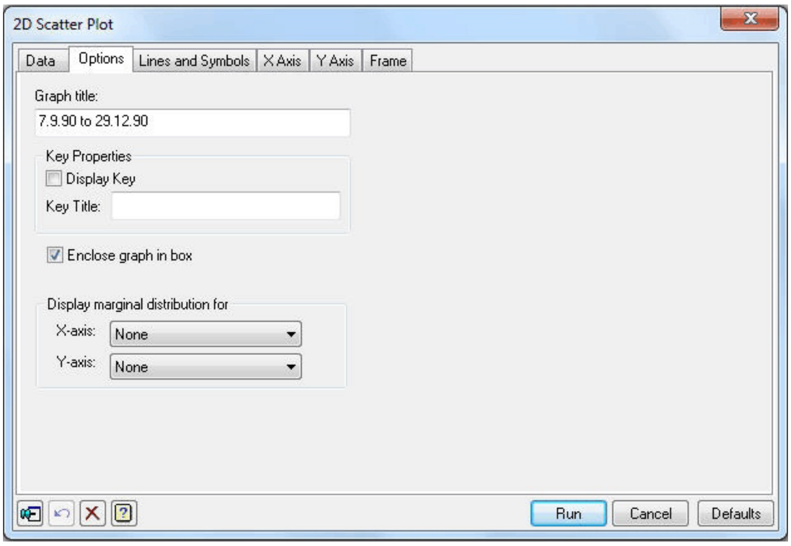


Figure 2.7

The Options tab is shown in Figure 2.7, where we have entered the title into the Graph Title box and unselected the option Display key so that no key is displayed on the graph. Clicking on Run produces the graph shown in Figure 2.8.

Many of the graphics menus contain axis tabs to allow you to modify the x- and y-axes. These allow you to set attributes such as a title, a lower limit, an upper limit, position of tick marks and the origin position through which to draw the other axis i.e., where to draw the x-axis across the yaxis, or vice-versa.

To illustrate the axis tabs, we open the Choose Graph menu, and again select Two or more variates option at the top of the menu, and 2-D Scatter Plot as the type of graph. On the Data tab we then select Sulphur as the Y variate, and Windsp as the X variate as before.

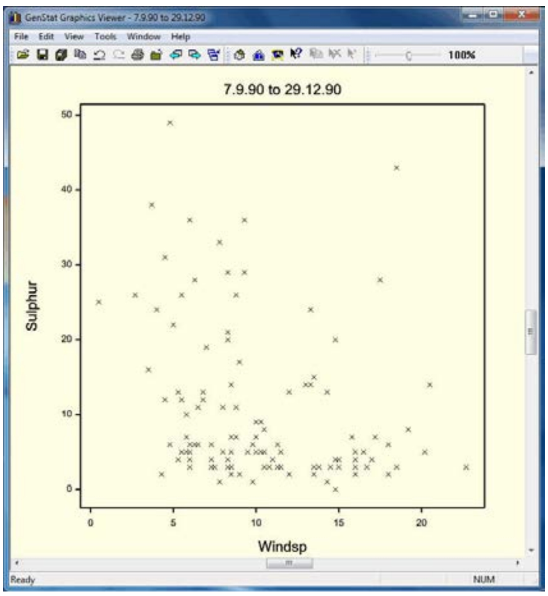


Figure 2.8

The X Axis tab is shown in Figure 2.9. To improve the plot, we select the Display Title option and enter the title as Wind speed m/s. We then enter the lower and upper bound values as 0 and 25.

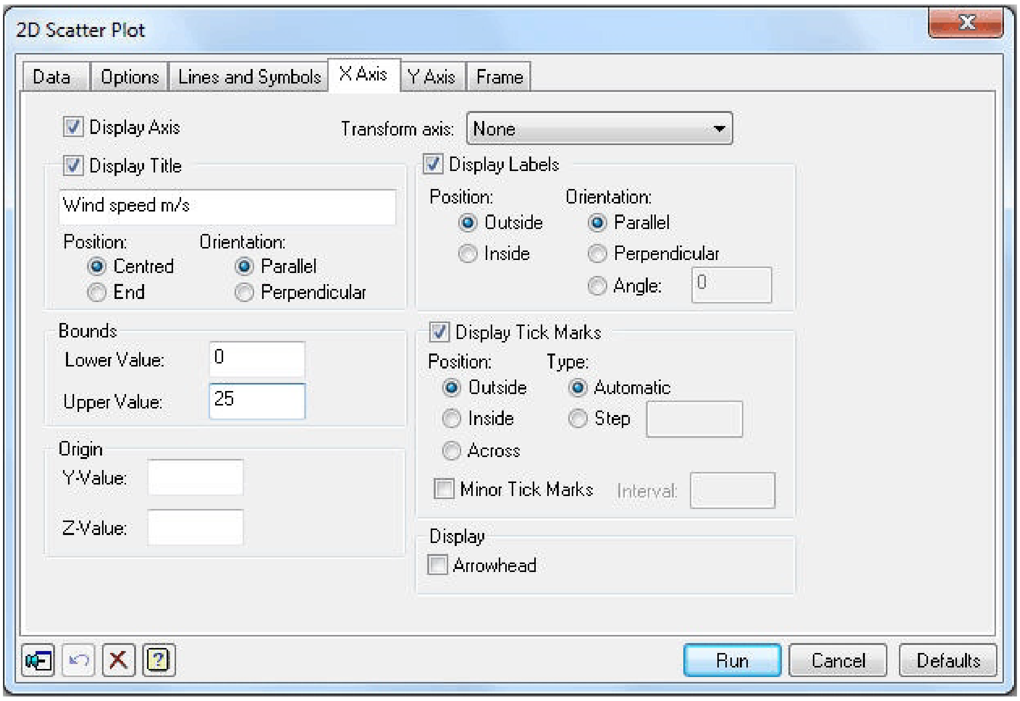
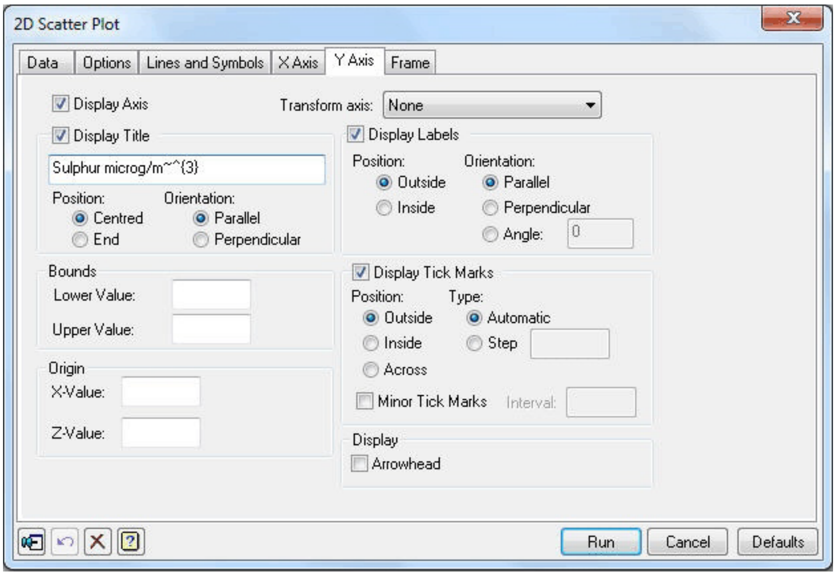


Figure 2.9



We now click on the Y Axis tab which produces an identical menu to the X Axis tab. The new menu is shown in Figure 2.10 where we have selected the Display Title option and entered the title Sulphur microg/m~^{3} into the space provided. The string ~^{3} is a special typesetting command to make 3 a superscript letter.

Tip More details on typesetting can be found In Appendix 3

Figure 2.10

You can set the style and colour of the symbols for the scatter plot using the Line and Symbols tab, shown in Figure 2.11. We have selected Plot 1 from the Graph list, and chosen Circle from the symbols. You can chose colours for the circle (in the Colour list box), and for its interior (in the Fill Colour list box). Here we have chosen dark blue circles filled with light blue.

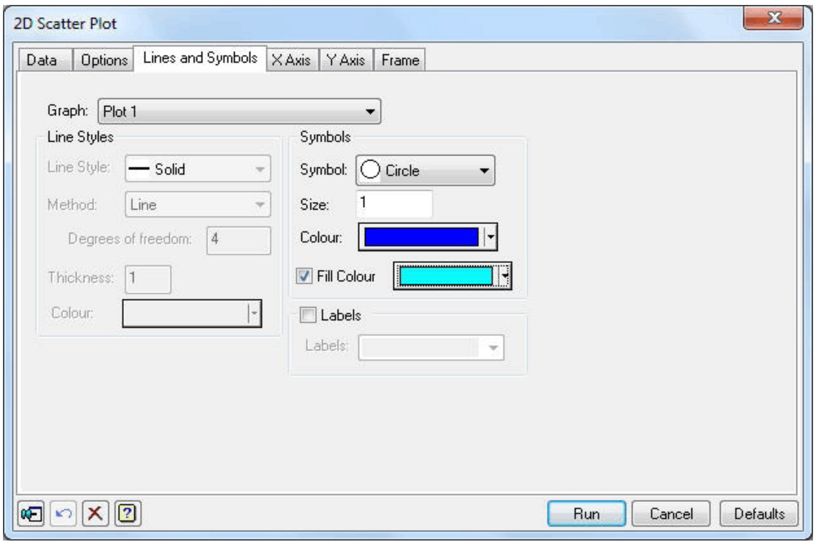


Figure 2.11

Finally, the Frame tab (Figure 2.12) allows you to control the positioning of the plot within the graphics "frame". Each graph is plotted into one of the "windows" in the frame. Here we are plotting into window 1 using its default position. This is defined to use the top 3/4 of the frame. (Window2, which is used if you include a key to the graph, uses the lower 1/4 of the frame.)
The Draw radio buttons allow you to put extra graphs onto an existing frame.

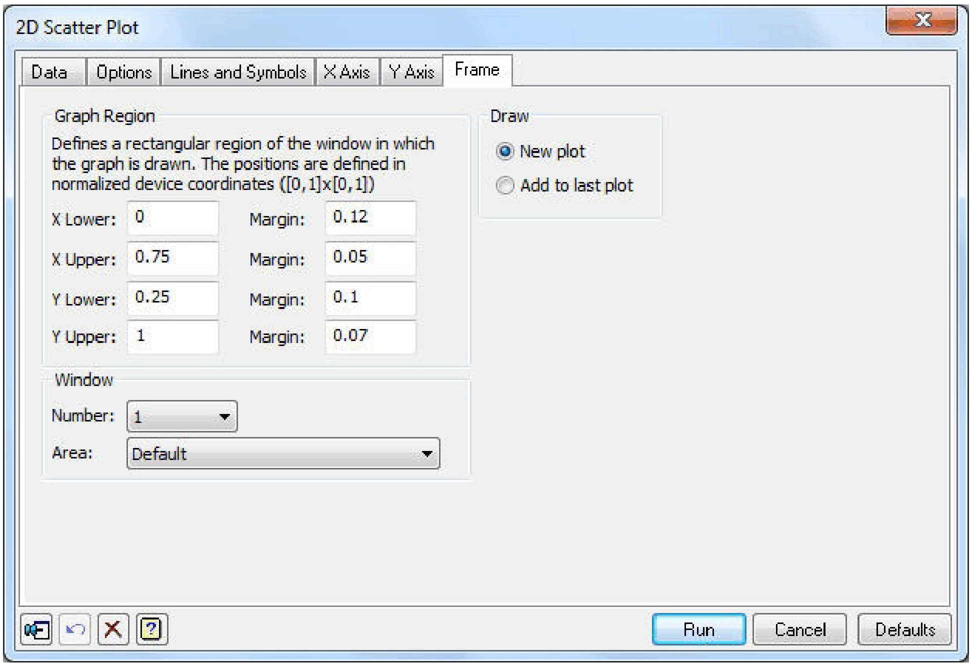


Figure 2.12

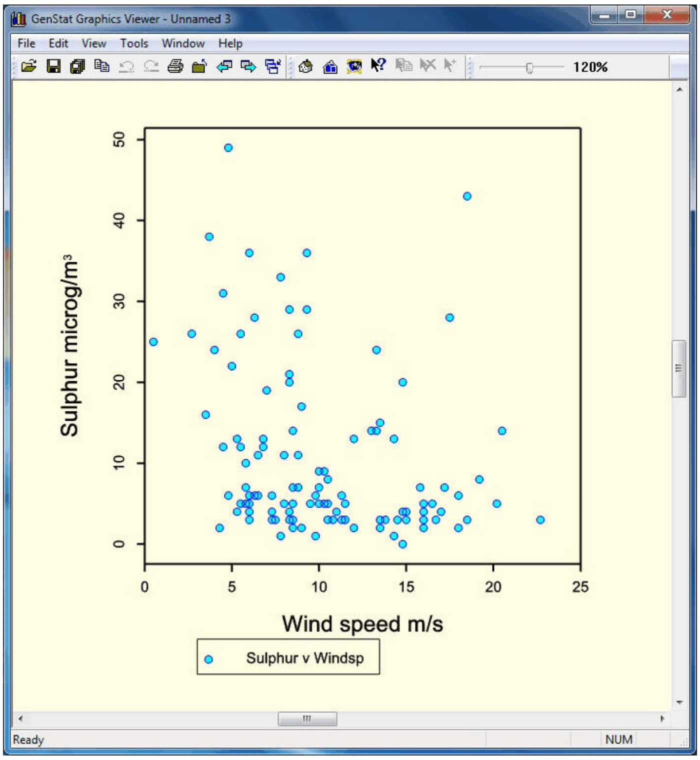


Figure 2.13

Clicking on Run produces the graph shown in Figure 2.13, where the points are now represented by filled circles.
Modifications to this graph, for example a forgotten title, can still be made by going into Edit mode, see Chapter 3.

3 The GenStat graphics viewer

Default the plots are dispayed in the GenStat Graphics Viewer. The GenStat Graphics viewer is a program that runs independently of GenStat and displays graphical output. It is started automatically when a graph is generated, and you can switch between GenStat and the graphics viewer at any time. The graphical images are produced in a vector format which is scaled to fit the current size of the GenStat Graphics window, which means that the full resolution can be maintained if the window is resized or printed.

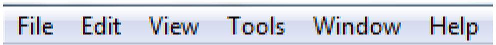
When using GenStat, the Graphics Viewer can be raised to the top of the display at any time by clicking on the Graphics button on the Standard toolbar: 

The GenStat graphics format also includes extensive meta-information about the plot and the data contained within it, allowing many aspects to be edited interactively on screen.

The viewer operates in **three modes – View, Edit and Child**.

The **View** mode has its own bars as you can see in Figure 2.13 of the previous Chapter.

Title bar 

Menu bar 

Toolbar 

Tip The Tools menu on the menu bar contains an Options submenu. Tick the 'Sequence number in Windows Title'-box in the Workspace Tab. This is very helpful if there are many plots.

When graphs are received from GenStat or opened via the File menu they will be displayed in the **View mode**. In this mode you can zoom the graph, view data information on different points, include a main title and save to other graphics formats, such as Enhanced Windows metafiles, bitmaps, JPEG, PostScript, and PNG. The viewer displays a single graph (or composite plot), and other graphs can be viewed by selecting the appropriate graph name from the Window menu or by using the Previous or Next arrows on the Toolbar.

To make changes to the graph you need to invoke the **Edit mode**. To do this, double-click on the plot you wish to edit or select Edit Graph from the Edit menu. The window title and background colour should change to indicate the new mode as shown in Figure 3.1.

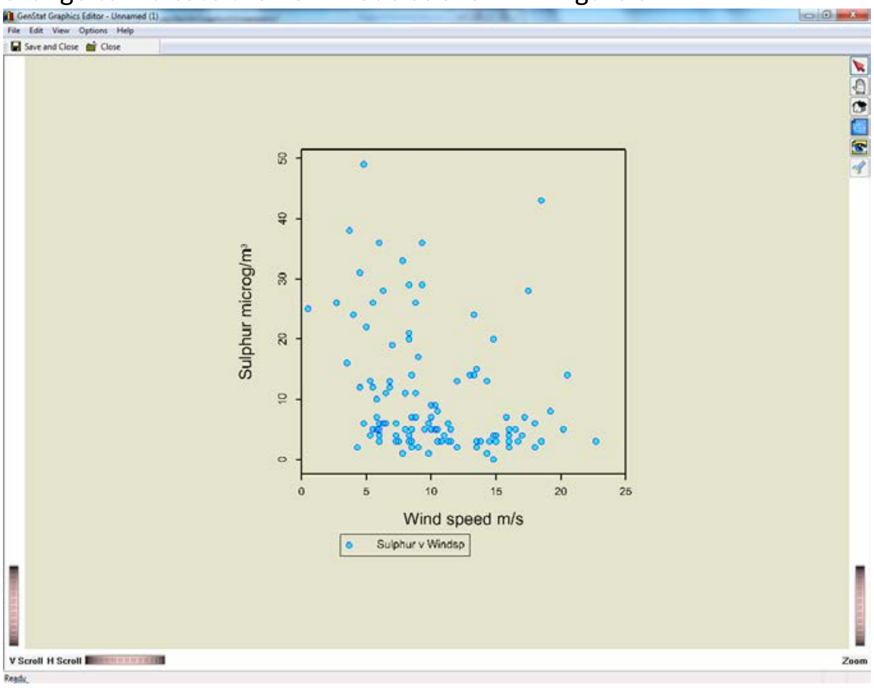


Figure 3.1

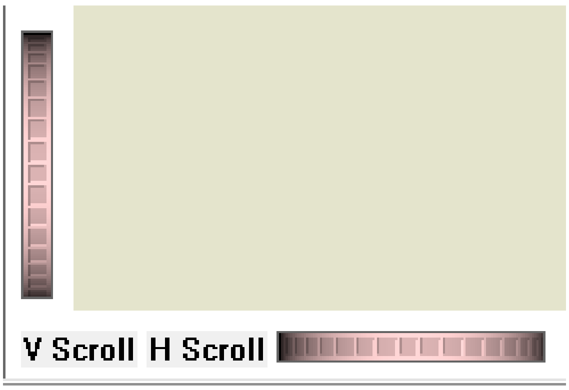
Within the edit mode you can either double-click on a graph or select one of the options from the Tools menu to edit the graph. For each graph you can change settings for the following:

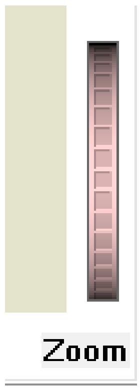
Layout	The layout of the graph
Key	The layout of the key or legend
X-Axis, Y-Axis and Z-Axis	For setting axes attributes

The following graph types have some additional options to control the display.
Graph options for scatter and line plots, Histogram options, Pie Chart options, Contour plot options, Surface plot options, Shade diagram options, 3D Histogram options.

The Scroll (V Scroll, H Scroll) rollers in the lower left corner (see Figure 3.2) can be used to move the graph left and right (H Scroll) or up and down (V Scroll). The Zoom roller in the lower right corner (see Figure 3.3) enlarges/shrinks the graph about its home point. To move a roller, click down on the roller button and drag your move left/right or up/down as appropriate.

In the upper right corner you can find the buttons shown in Figure 3.4





Select/Pick

View

Home

Set Home

View All

Seek

Figure 3.2 **Figure 3.3** **Figure 3.4**

Select/Pick Button **–** selects object manipulation or **pick mode** (and deselects camera or viewer mode). The cursor shape will change to an arrow as shown on the Select/Pick button. In this mode, the user is manipulating objects in the scene graph.

View Button – selects camera or **Editviewer mode** (and deselects object manipulation or pick mode). The cursor shape will change to an upright arrow.

Home Button – returns the camera to its home position (initial position if not reset).

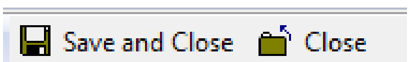
Set Home Button – resets the home position to the current camera position.

View All Button – brings the entire scene graph into view.

Seek Button – allows the user to select a new centre of rotation for the camera. When clicked on (and in Editviewer mode) the cursor changes to a crosshair. The next left mouse button press causes whatever is underneath the cursor to be selected as the new centre of rotation. Once the button is released, the camera animates to its new position.

Tip

The Edit mode of a 3-dimensional plot looks slightly different. At the end of this Chapter the differences are shown.

Once you have made any changes you can return to the viewer mode by using either selecting Save and Close or Close from the File menu respectively to retain or discard any changes. Another possibility is to use the buttons 

Graphs containing two or more individual components within a single display, such as a trellis plot, can be produced by GenStat. For these composite plots, the zoom and scroll controls operate on the entire display, whilst the built-in editor operates on individual component plots.

Whilst the viewer can only display a single graph (or composite plot), it maintains a list of graphs as they are generated within GenStat, making it easy to switch the display between the current plot and those generated earlier.

The Child mode allows you to view several plots at once. From the GenStat Graphics Viewer you can open any plot in a new Child Viewer. In the Child Viewer you cannot edit the plot or open other plots. You can still print, save and interact with the graph as in the main Graphics Viewer.

To open a new instance of the Child Viewer right-click on the plot you wish to view and select Display in Child Viewer from the Popup menu. The window title and background colour should change to indicate the new mode.

Individual displays can be saved to **GenStat metafiles**. This is a **private file format** that saves the graph and additional information, so that the graphs can be redisplayed or edited on another occasion, using the Graphics Viewer. These files have extension *gmf* and can be opened in the Graphics Viewer with the Open command in the File menu. This is not possible with the formats in the next line.

Tip

Another possibility is to save the GenStat **program** that produces the plot. Choose in File →Save as... →Save as type →GenStat Command Files (*.gen). This gives you the opportunity also to produce other plots because the data are in the program.

The contents of the Graphics window can also be saved to files in standard graphics formats, such as Enhanced Windows metafiles, bitmaps, JPEG, PostScript, and PNG.

GenStat graphics can also be saved directly to the various graphical file formats using the `DEVICE` and `OPEN` statements in command mode. The `DHELP` command lists the device numbers associated with each format.

As already mentioned in the Introduction, you may find it useful also to know the commands that produce the plots. If you use the Graphics Wizard, the commands can be found in the Input Log window. But you can also save the program. See the TIP above. The program that produces Figure 2.13 looks like:

```
GET       [SPECIAL = _special]
PEN       [RESET = YES] 1; THICKNESS = 2; SYMBOL = 1; COLOUR = RGB(0; 0; 0); YMISSING = break; \
          XMISSING = break
PEN       [RESET = YES] 2; LINESTYLE = 2; SYMBOL = 1; COLOUR = RGB(169; 169; 169); YMISSING = break; \
          XMISSING = break
PEN       [RESET = YES] 3; SYMBOL = 2; CLINE = RGB(0; 0; 0); CSYMBOL = RGB(0; 0; 255);\
          CFILL = RGB(0; 255; 255); CAREA = RGB(0; 0; 0); YMISSING = break; XMISSING = break
PEN       [RESET = YES] 4; JOIN = given; SYMBOL = 1; COLOUR = RGB(0; 0; 0); YMISSING = break; XMISSING = break
PEN       [RESET = YES] 5; COLOUR = RGB(0; 0; 0); SIZEMULTIPLIER = 1.5
PEN       [RESET = YES] 6; COLOUR = RGB(0; 0; 0)
PEN       [RESET = YES] 7; COLOUR = RGB(0; 0; 0)
PEN       [RESET = YES] 8; COLOUR = RGB(0; 0; 255)
FRAME     [RESET = yes] 1; YLOWER = 0.25; YUPPER = 1; XLOWER = 0; XUPPER = 0.75; \
          YMLOWER = 0.1; YMUPPER = 0.07; XMLOWER = 0.12; XMUPPER = 0.05; \
          BOX = include; PENTITLE = 5; PENKEY = 7; PENGRID = 2
XAXIS     [RESET = YES] window = 1; TITLE = 'Wind speed m/s'; TPOSITION = middle; TDIRECTION = parallel; \
          LOWER = 0; UPPER = 25; MPOSITION = outside; NSUBTICKS = 0; \
          LPOSITION = outside; LDIRECTION = parallel; PENTITLE = 5; PENAXIS = 1; PENLABELS = 7
YAXIS     [RESET = YES] window = 1; TITLE = 'Sulphur microg/m~^{3}'; TPOSITION = middle; TDIRECTION = parallel; \
          MPOSITION = outside; NSUBTICKS = 0; LPOSITION = outside; LDIRECTION = parallel; PENTITLE = 5; \
          PENAXIS = 1; PENLABELS = 7
VARIATE   _x_0, _y_0; !(14.8, 14.3, 5.5, 5, 4.5, 4.8, 4.3, 4, 9.3, 6.3, 5.8, 8.3, 16, 15.8, 16, 16.7, 9.5, 9.8, 12, 4.8, 2.7, \
          6.5, 13.5, 6, 10.5, 5.3, 18, 8.5, 15, 22.7, *, 8.5, 8.3, 14.3, 15, 10.5, 13.8, 8.5, 6, 16.5, 7.3, 9.8, 7.3, 5.5, 6, \
```

```
         11.3, 8.8, 8.5, 8.3, 14.5, 9, 11.5, 10, 11.3, 18.5, 16, 8.3, 13, 13.5, 10, 9, 7.3, 5.8, 8.5, 6, 8, 7.5, 10.5, 11, 16, \
         13.5, 20.2, 11.5, 15, 10.3, 7.8, 6.3, 6.8, 10, 5.5, 17, 19.2, 10, 6, 8.8, 9.3, 6.5, 6.8, 8.8, 8.3, 5.3, 10.3, 13.3, 7, \
         20.5, 17.5, 14.8, 18.5, 8.3, 0.5, 7.8, 3.5, 4.5, 3.7, 8, 11.5, 5.8, 14.8, 13.3, 10.8, 15, 17.2, 12, 18), !(0, 13, 12, \
         22, 12, 6, 2, 24, 36, 6, 10, 4, 3, 7, 2, 3, 5, 6, 13, 49, 26, 6, 3, 6, 8, 4, 6, 5, 3, 3, 10, 7, 3, 1, 4, 5, 3, 3, 3, 5, 3, 1,\
         6, 5, 5, 6, 11, 2, 3, 3, 2, 3, 7, 3, 3, 5, 29, 14, 15, 9, 17, 4, 7, 14, 4, 5, 3, 3, 4, 4, 2, 5, 3, 4, 5, 33, 28, 13, 5, 26, 4, \
         8, 9, 36, 7, 29, 11, 12, 26, 21, 13, 9, 24, 19, 14, 28, 20, 43, 20, 25, 1, 16, 31, 38, 11, 5, 5, 4, 14, 3, 3, 7, 2, 2)
PEN      3; METHOD = point;
DGRAPH   [WINDOW = 1] Y = _y_0; X = _x_0; PEN = 3; DESCRIPTION = 'Sulphur v Windsp'
SET      [DSAVE = _special['dsave']]
```

Clicking on the  button (a toggle) on the toolbar of the Graphics Viewer changes the cursor in an arrow with a question mark (only in a DGRAPH or D3GRAPH plot). If then the cursor is moved to a point in the graph information about that point appears. See Figure 3.5.

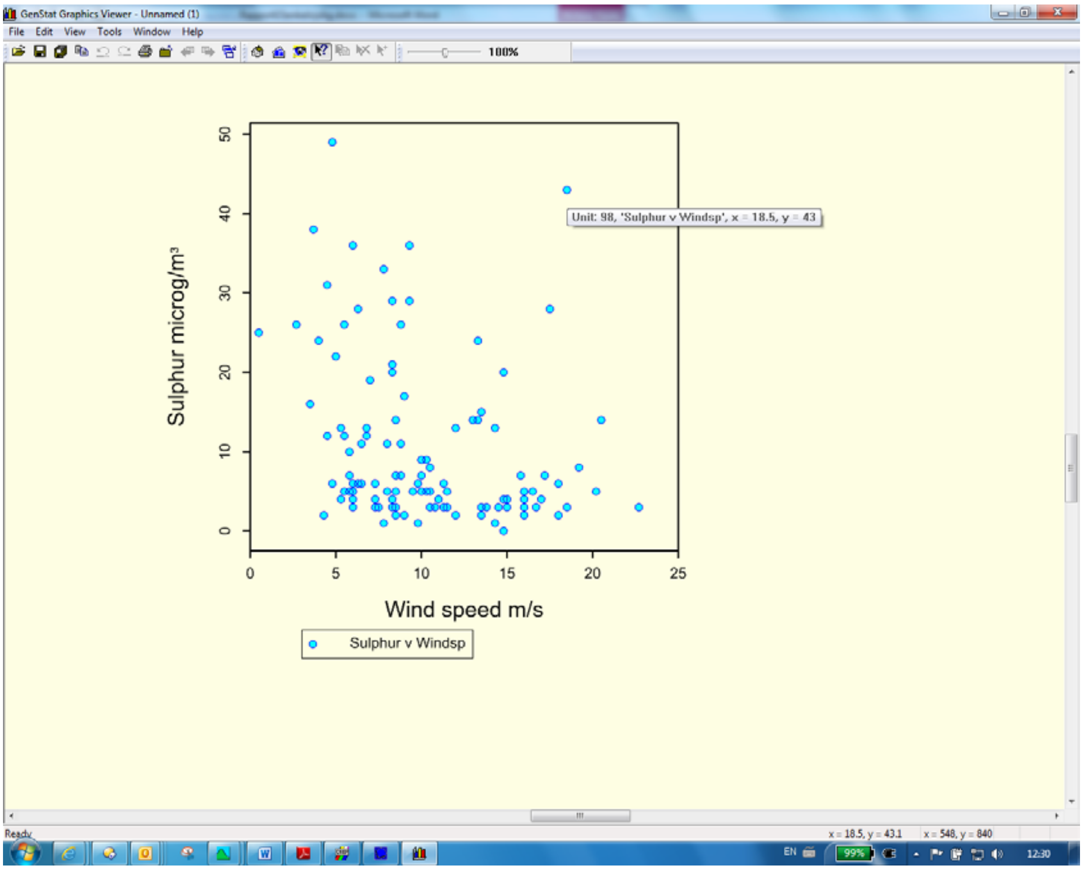
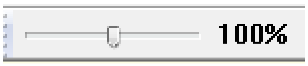


Figure 3.5

Clicking on a point modifies the colour. In Figure 3.6 this is done for 6 points. After clicking with the right mouse button in the graph a menu appears (also shown in Figure 3.6). Clicking on Copy Data Information copies information onto the clipboard.

With the Paste command this information is inserted in the next lines:

Unit	**Description**	**'X Coordinate'**	**'Y Coordinate'**	**Label**
20	**'Sulphur v Windsp'**	**4.8**	**49**	
98	**'Sulphur v Windsp'**	**18.5**	**43**	
96	**'Sulphur v Windsp'**	**17.5**	**28**	
95	**'Sulphur v Windsp'**	**20.5**	**14**	
30	**'Sulphur v Windsp'**	**22.7**	**3**	
100	**'Sulphur v Windsp'**	**0.5**	**25**	



With this slicer on the toolbar of the viewer the graph can be resized within the graphics window.

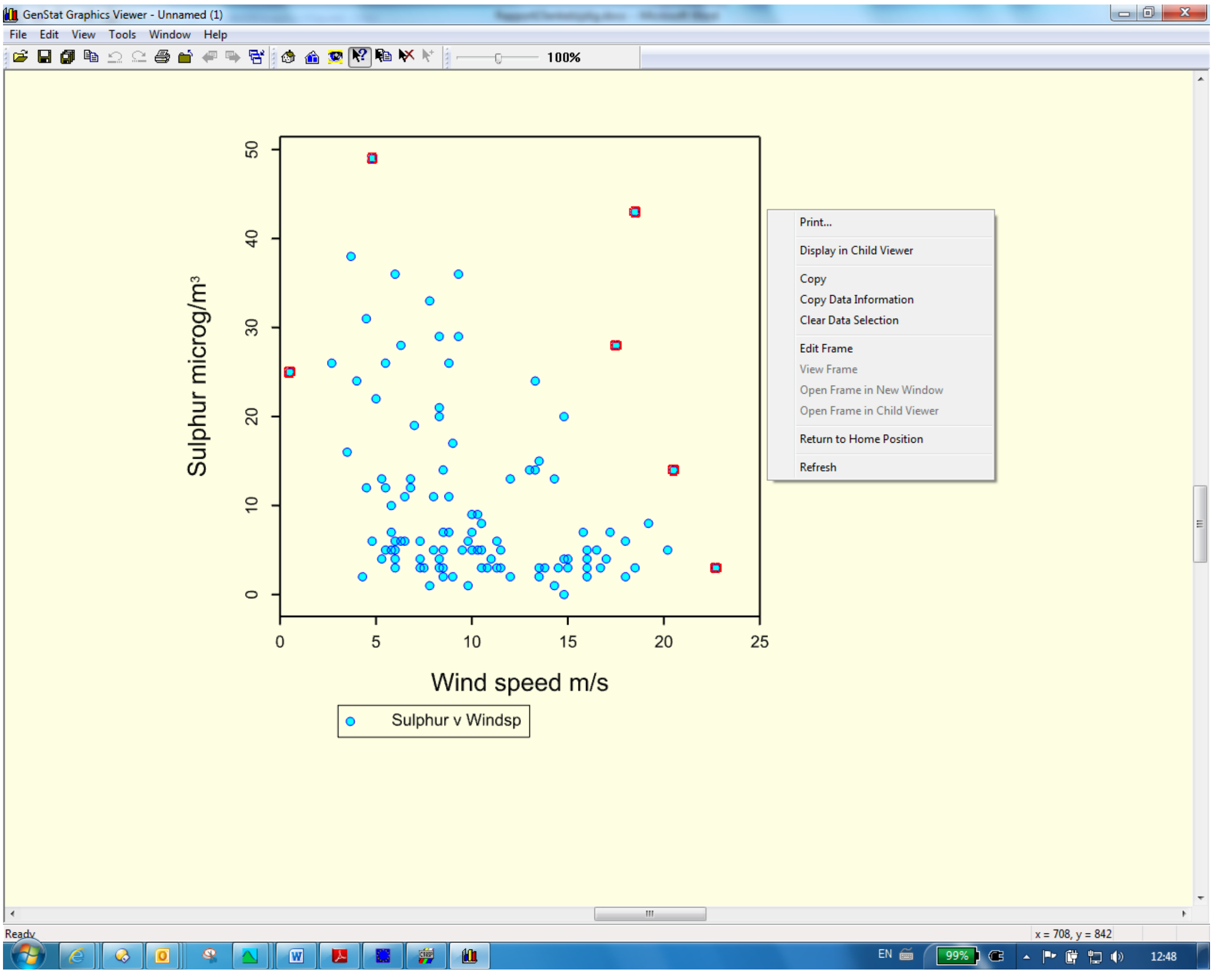


Figure 3.6

As we will see in Chapter 4 the DGRAPH directive has two parameters `HOTCOMPONENT` and `HOTDEFINITION`. By using these parameters the  button on the Toolbar of the viewer becomes available.

Tip

When you want to use files from subdirectories of GenStat in a command line, you can use %GENDIR% for the main directory specification. In the next program there is an example.

The next program is example DGRA-8.gen from the %GENDIR%/Examples subdirectory and uses the `HOTCOMPONENT` and `HOTDEFINITION` parameters. This example can also be run by:
Help → Examples →Commands... → Look for: DGRAPH →Regression line with hotpoints

```
" Example DGRA-8: plotting a regression line and the residuals.
  Uses hotpoints to illustrate size of residual for each point."

" File DGRA-8.DAT contains recordings of blood-pressure from a sample of
  38 women whose ages range from 20 to 80."

" Read the data recorded in the file."
FILEREAD  [NAME='%gendir%/examples/DGRA-8.DAT'] Age, Pressure

" Fit the linear model."
MODEL     Pressure
TERMS     Age
FIT       [PRINT=*] Age
"Save the fitted values"
RKEEP     FITTED=Fitted
```

```
" Construct matrix containing data values and the expected values of X."
CALC      nva=NVALUES(Age)
MATRIX    [ROWS=nva; COLUMNS=3] mx,my
CALCULATE mx$[*; 1,2,3],my$[*; 1,2,3] = Age,Age,!s(*),Pressure,Fitted,!s(*)
PEN       4,5; METHOD=line; JOIN=ascending,give; SYMBOL=0

" Set up hot definition matrix by creating a matrix with one column and
  values 1...nva.
  All values of the hot definition matrix have to be integers."
MATRIX    [ROWS=nva; COLUMNS=1; VALUES=1...#nva] h

" Plot Pressure and Fitted values. Set HOTDEFINITION so that Pressure is
  assigned the matrix h. This means that the first point of (X,Y) has the
  'hot value' 1, the second point of (X,Y) has the 'hot value' 2, etc. The
  numbers 1,2,3... come from the coresponding elements of the matrix h."
DGRAPH    [WINDOW=1] Y=Pressure, Fitted; X=Age; PEN=2,4; HOTDEFINITION=h,*

" Plot lines linking residuals to fitted line.
  Set HOTCOMPONENT so that each row of mx and my is assigned an integer
 which appears in the matrix h. For example, when the point associated with
 'hot value' 10 is clicked the line with HOTCOMPONENT 10 appears."
DGRAPH    [SCREEN=keep] Y=my$[1...#nva;*]; X=mx$[1...#nva;*]; PEN=5; \
           DISPLAY=hide; HOTCOMPONENT=1...nva; DESC=''

"To set up hotpoints so that all points appear whenever a data point is
 clicked, make all elements of the matrix h the same and set HOTCOMPONENT
equal to this value. For example:"
MATRIX    [ROWS=1; COLUMNS=1; VALUES=100] h2
PEN       [RESET=yes] 6; COLOUR='Green'; CFILL='Green'; SYMBOL=2; \
          LABEL='Click circle (in hot definition mode) for all lines'
DGRAPH    [SCREEN=keep] Y=107.5;X=20; HOTDEFINITION=h2; PEN=6; DESC=''
DGRAPH    [SCREEN=keep] Y=my; X=mx; PEN=5; DISPLAY=hide; DESC=''; \
           HOTCOMPONENT=100
```

Running this program gives the graph in Figure 3.7. After clicking the  button, clicking on some circles gives the light blue vertical lines in Figure 3.8 and clicking on the green circle gives the graph in Figure 3.9. Clicking again on the circles removes the lines again.

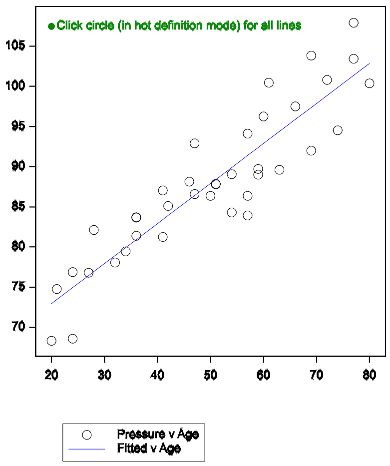


Figure 3.7

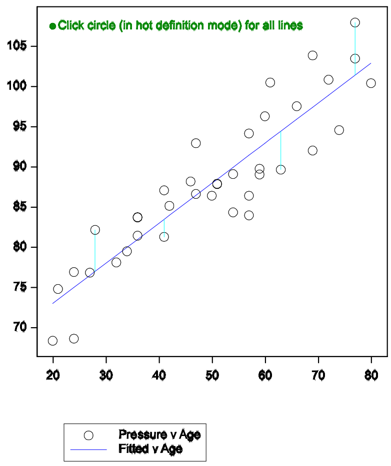


Figure 3.8

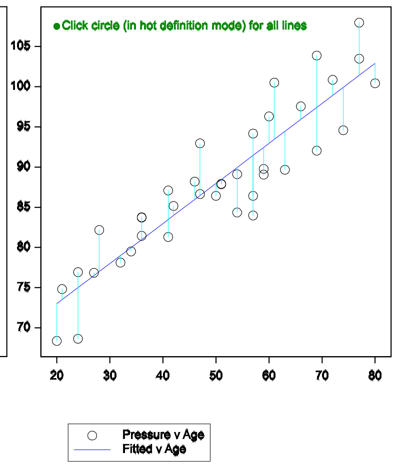


Figure 3.9

The Edit mode of a 3-dimensional plot differs as follows from a 2-dimensional plot:
Instead of moving the graph left and right or up and down the plot can now be rotated by the rollers in the lower left corner (see Figure 3.10). With the RotX roller the rotation is about the x-axis, with the RotY roller about the y-axis. With the Zoom roller (see Figure 3.11) the plot can be resized.
The arrows in the lower right corner (see Figure 3.12) are not present in a 2-dimensional plot.

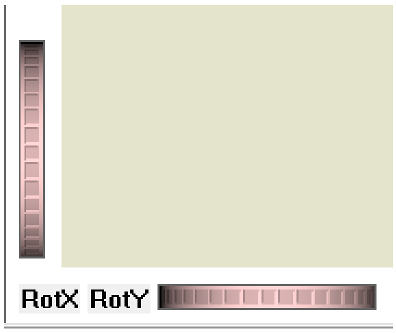


Figure 3.10

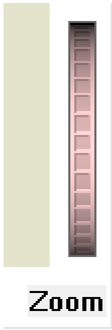


Figure 3.11

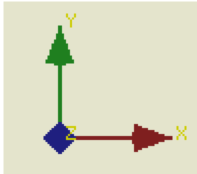


Figure 3.12

Select/Pick
View — The cursor in the Editview mode changes in two curved arrows.
Home
Set Home
View All
Seek
X Projection — Only in 3-dimensional graph
Y Projection — Only in 3-dimensional graph
Z Projection — Only in 3-dimensional graph

Figure 3.13

You have to use the the View button  if you want to rotate the D3GRAPH plot in another direction than about the x- or y-axis (with the RotX and RotY rollers). To illustrate this a 3-dimensional graph is produced by adding a third variable, Wind direction, to the 2 variables Sulphur and Wind speed in Figure 3.2. Figure 3.14 shows this plot in the Editview mode of the Graphics Viewer. By clicking on the figure with the left mouse button without releasing it you can rotate the plot.
When you stop with moving the mouse and release the left mouse button, a rotated plot is produced as shown in Figure 3.15.

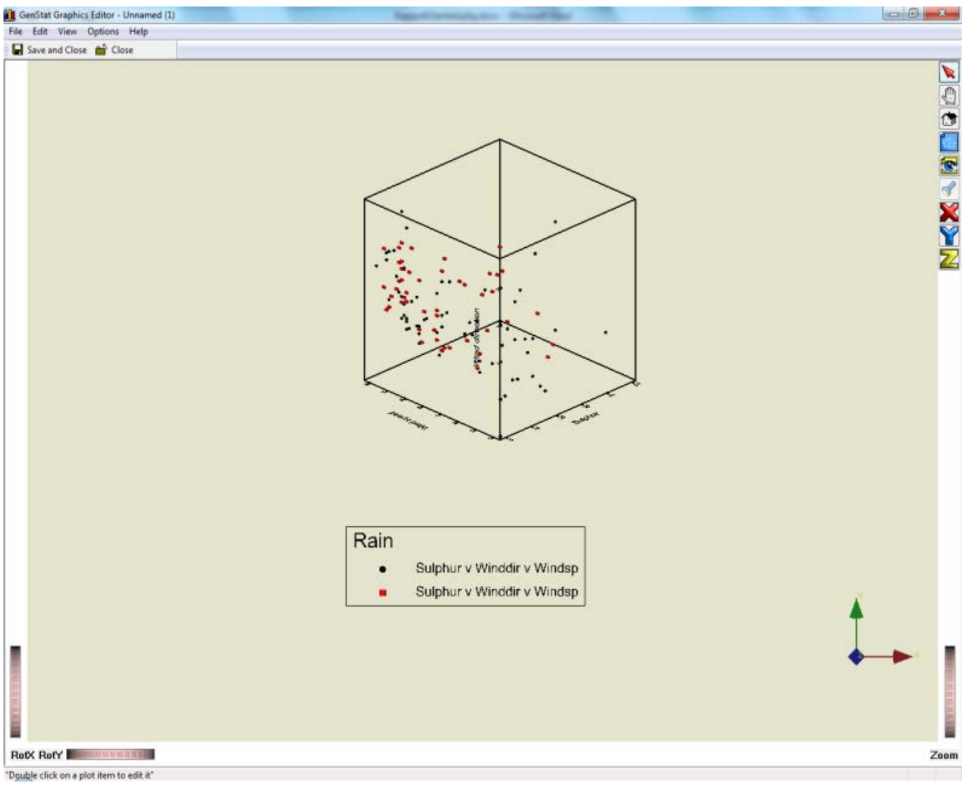


Figure 3.14

If you release the left mouse button without stopping with moving the mouse, the plot keeps rotating. You can stop the rotation then by clicking with the left mouse button.
The projection buttons, X Projection, Y Projection and Z Projection, can be used to make a projection in respectively the y-z plane, the x-z plane and the x-y plane. You can find the projections of Figure 3.14 in figures 3.16, 3.17 and 3.18.

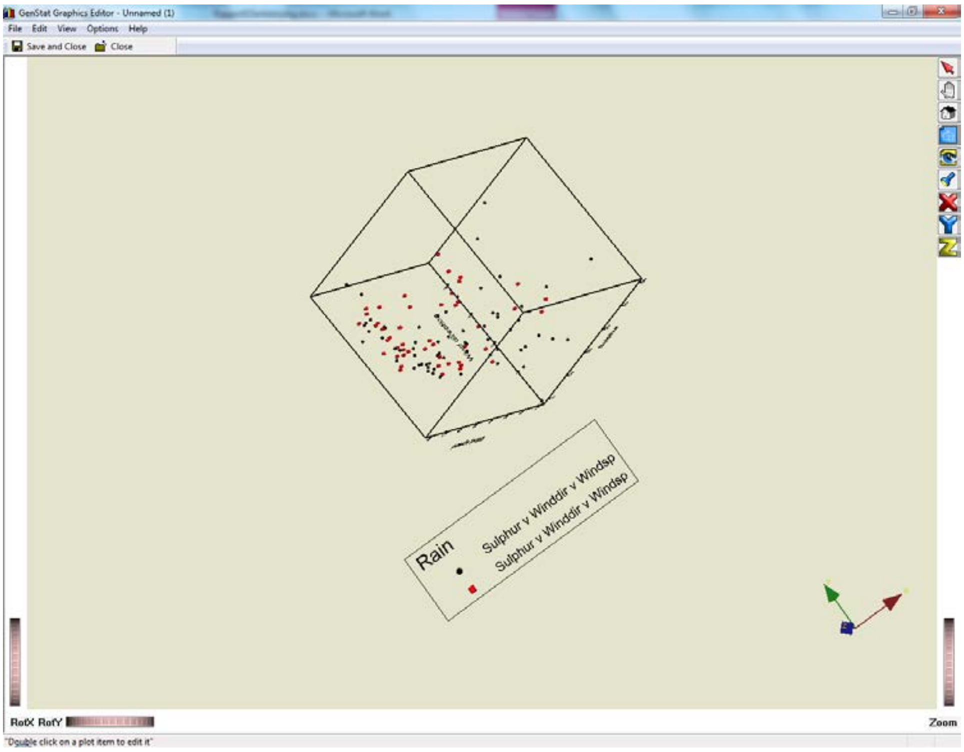


Figure 3.15

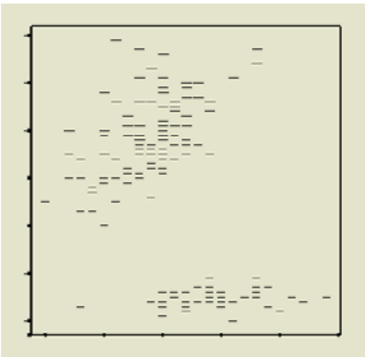


Figure 3.16: X projection

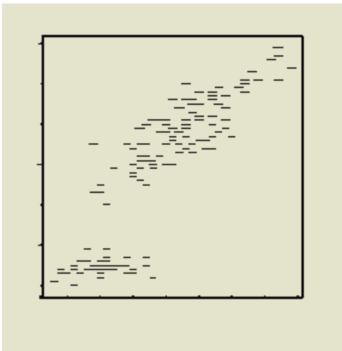


Figure 3.17 Y Projection

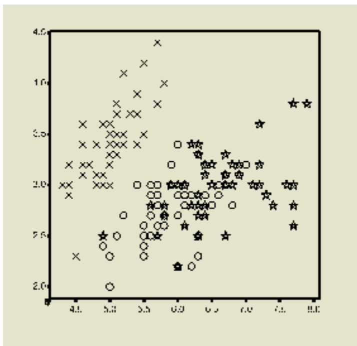


Figure 3.18: Z Projection

4 Graphical display

4.1 Introduction to high-resolution graphics

The DGRAPH directive is used in this section to introduce the structure of the high-resolution graphics in GenStat. A full description of DGRAPH is given in Section 4.2.1.

Before producing any high-resolution plots, you must first select an appropriate output *device*. This can either be screen-based, for interactive use, or it may send the output to a file in one of a number of standard formats suitable for plotters, printers or word-processed documents. Different versions of GenStat may support different types of device; inevitably there are minor differences in the details of their operation, and these are discussed further in the description of the DEVICE directive (4.9.1.1). The default graphics device is chosen to be the most appropriate for each version and, if this is suitable, no explicit action is required before you start plotting. The examples in this chapter from Reference 2 (see the Introduction) have all been generated using the HPGL device, available in all versions of GenStat. This stores output in a file which can then be sent to a printer or incorporated into other documents, as has been done in this chapter. The other figures in this guide and in the Examples chapter have been generated using the PNG device (see Appendix 4).

In the simplest use of DGRAPH, you just need to specify the x- and y-coordinates of the points to be plotted and, if required, a title for the plot. For example, the statement:

```
DGRAPH [TITLE='Scatter Plot'] Y1,Y2; X1,X2
```

generates the graph shown in Figure 4.1a. There are separate parameter lists for the y- and x-coordinates, which are processed in parallel so that the graph contains plots of Y1 versus X1 and Y2 versus X2. The TITLE option provides a title for the graph, which is drawn at the top of the plot. It can be up to 80 characters in length, and must consist of one line of text only.

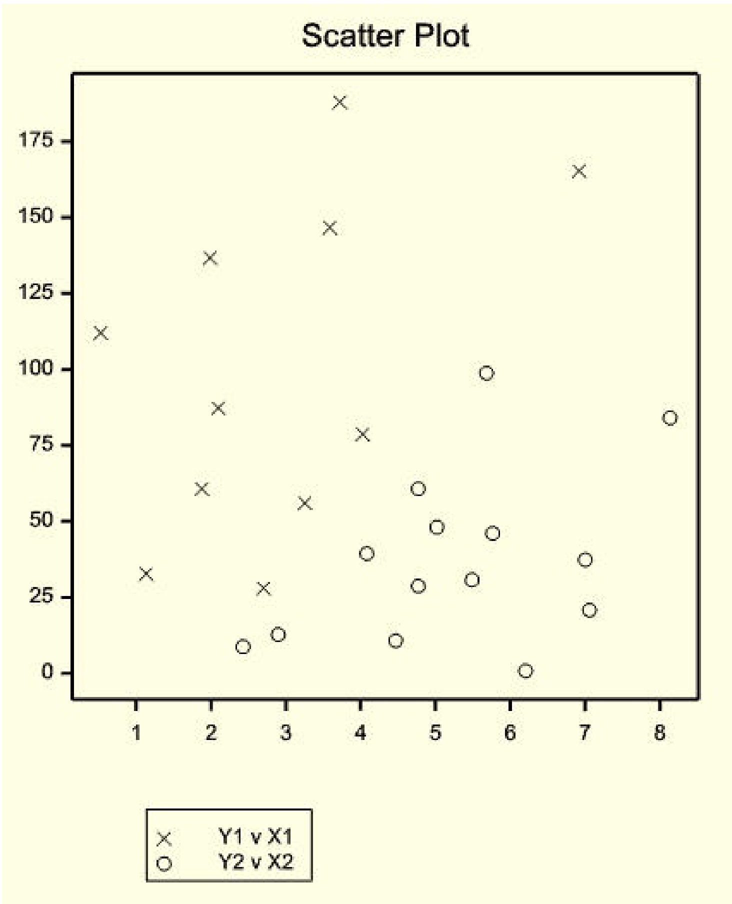


Figure 4.1a

However, there are many more aspects of the output that can be controlled when producing a graph, and it is not feasible to allow all of these to be specified by the options or parameters of DGRAPH. The syntax would have become very complicated, and you would have had to specify all the relevant settings every time that DGRAPH was used. Instead additional directives are used to set up and modify a graphical *environment* which contains most of the information required when plotting. Each time DGRAPH is used, it accesses the relevant information from this environment in order to determine how to construct the graph. Thus, to make a simple modification to a graph, for example to change the colour of the plotted symbols, you need make only that specific change; any other information that you have supplied previously will remain in force. This section illustrates some of the settings that can be used to control or modify the appearance of graphical output**. The complete description of the various elements of the environment and the directives that can be used to define them is in Section 4.9.**

All the elements of graphical output, such as symbols, lines, axes, titles, labels, annotation and filled polygons are drawn by *pens*, which have associated definitions covering various attributes, like colour and symbol type. The pen also indicates the plotting method, that is, what kind of plot is to be drawn. For example, the following statements can be used to plot the data and the fitted line from a regression of Logpress on Boiltemp:

```
PEN    1,2; METHOD=line,point; SYMBOL=0,1; COLOUR='black'
DGRAPH [TITLE='Simple Linear Regression'] Fitted,Logpress;\
       Boiltemp; PEN=1,2
```

This means that *pen* 1 will be used to plot a line through the points specified by Fitted and Boiltemp, and *pen* 2 will be used to plot the points specified by Logpress and Boiltemp. The corresponding output is shown in Figure 4.1b.

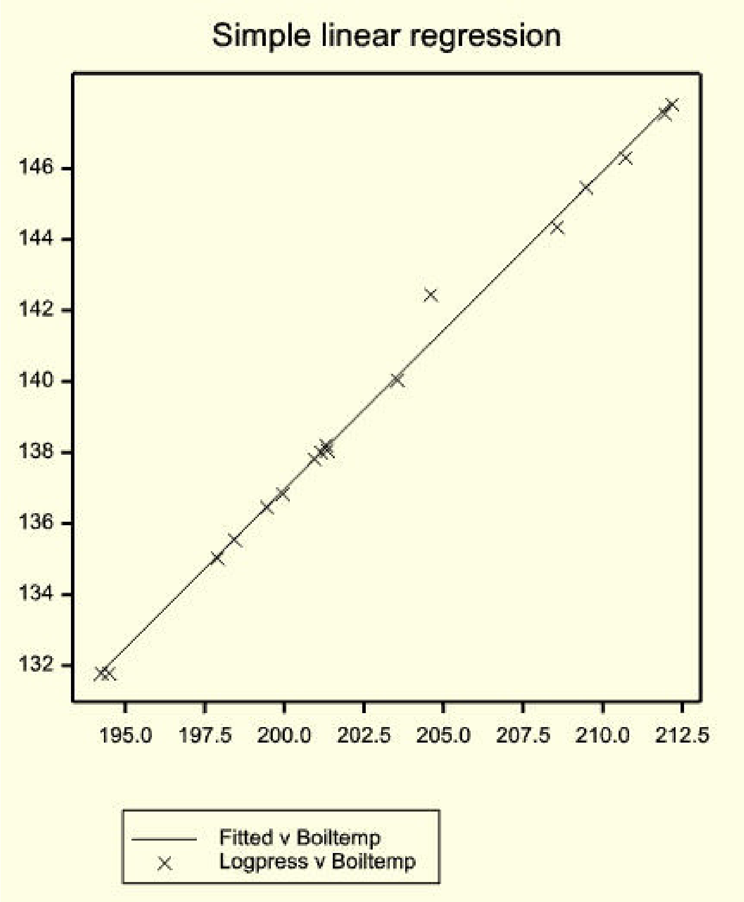


Figure 4.1b

Pens are available with numbers 1 up to 256, each with its own attribute settings, thus allowing a wide variety of styles within each plot. You can control which pens are used to plot the data, using the PEN parameter of DGRAPH as shown above. **The PEN directive (4.9.8) can be used to define the various attributes of each pen, such as the colour, symbol type and line style (whether lines should be full, dotted or dashed).** You can also specify labels to be plotted at each point, and control the size of symbols and text and the thickness of lines. When plotting a line you can switch off the symbols if you do not want to mark individual points, by setting SYMBOLS=0.

The XAXIS, YAXIS and ZAXIS directives (4.9.4, 4.9.5 and 4.9.6) allow you to specify the pens to be used for the axes and their titles, and the FRAME directive (4.9.3.1) enables you to specify the pen for the overall title. Unless you specify otherwise, GenStat uses pens with negative numbers for these features. Negatively numbered pens cannot be used for any other purposes, so this avoids any unintended side effects when pens are modified to change the main parts of the graph.

Tip

The purpose of the negative pen numbers (the default numbers of specific pen related options and parameters of different directives) is explained in the table of Appendix 1.

If the PEN parameter of DGRAPH is not specified, pen 1 is used for the first pair of Y and X structures, pen 2 for the second pair, and so on. So the pens need not have been specified explicitly in the DGRAPH statement above. The same convention applies to the pens used for different structures in histograms, pie charts and contour plots. The default settings for each pen are designed so that they will differ in appearance, for example by using different colours or line styles, depending on the output device. Thus, if you specify several data structures to appear in the same plot, the different sets of points or lines will be clearly distinguished by their different pens; see Figure 4.1a. The pens can be re-defined between DGRAPH statements in order to have a different effect each time.

You can also control the position and size of the graph. All graphical output is drawn in individual graphics *windows.* A window is a rectangular area of the screen. The position of the window is defined in terms of its lower and upper bounds in the vertical (y) and horizontal (x) directions using a data-independent coordinate system that ranges from 0.0 to 1.0 in each direction. You can use the default window positions defined by GenStat or you can use the FRAME directive (4.9.3.1) to define your own. The

WINDOW option of DGRAPH indicates which window is to be used for the plot and the KEYWINDOW option specifies the location of the key.

The FRAME statement in lines 25 and 26 of Example 4.1 defines window 4 to have dimensions that will ensure that the y- and x-axes are suitably in proportion to show the shape of the series (see for example Cleveland & McGill 1987). The setting KEYWINDOW=0 in DGRAPH in line 33 stops the key being displayed. The resulting graph is shown in Figure 4.1c.

Example 4.1

```
   2  " Wulfer's sunspot numbers, from Yule (1927)."
   3  READ      Sunspots

      Identifier    Minimum       Mean    Maximum       Values      Missing
       Sunspots      0.0000      44.76      154.4          176            0

  20  VARIATE   Year; VALUES=!(1749...1924); DECIMALS=0
  21  " Set the window size so that the y- and x-axis length are in the
 -22    ratio 0.065:1, by calculating the upper y-bound to be
 -23    0.065 * x-axis length + y-margin (see 4.9.3.1)."
  24  CALCULATE Yup = 0.065*0.9 + 0.1
  25  FRAME     4; YLOWER=0.0; YUPPER=Yup; XLOWER=0.0; XUPPER=1.0;\
  26            YMLOWER=0.1; YMUPPER=0.0; XMLOWER=0.1; XMUPPER=0.0
  27  VARIATE   Year; VALUES=!(1749...1924); DECIMALS=0
  28  " Set the window size so that the y- and x-axis length are in
 -29    the ratio 0.065:1, by calculating the upper y-bound to be
 -30    0.065 * x-axis length + y-margin (see 4.9.3.1)."
  31  CALCULATE Yup = 0.065*0.9 + 0.1
  32  FRAME     4; YLOWER=0.0; YUPPER=Yup; XLOWER=0.0; XUPPER=1.0;\
  33            YMLOWER=0.1; YMUPPER=0.0; XMLOWER=0.1; XMUPPER=0.0
  34  XAXIS     4; LOWER=1749; UPPER=1924; MARKS=!(1750,1800,1850,1900)
  35  YAXIS     4; LOWER=0; UPPER=175; MARKS=150
  36  PEN       1; METHOD=line; SYMBOL=0
  37  DGRAPH    [WINDOW=4; KEYWINDOW=0] Sunspots; Year
```

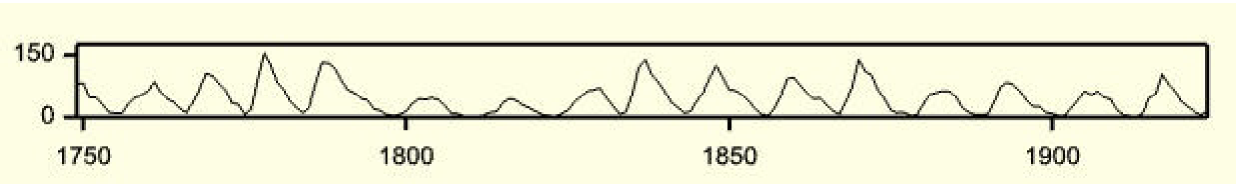


Figure 4.1c

Altogether, there are 256 windows, numbered from 1 up to 256. Windows are independent of one another and on most devices they are allowed to overlap or contain others. So it is possible to build up complex displays by a sequence of plotting commands. Details are given in Section 4.8.

For all the directives that produce graphics, the default window is window 1 and the default key window is window 2. The windows all have initial default sizes, as explained in 4.9.3.1. The default for window 2 is not very large. If you include too many variates in the plot, the key window may become full and a warning message will be printed. Also, very long identifier names or descriptions will be truncated if the width is insufficient. In either case you may want to use FRAME to increase the window size.

Each window has an associated definition for the axes that may be drawn in that window. The default definition will often be sufficient, but you can use the XAXIS, YAXIS and ZAXIS directives (4.9.4, 4.9.5 and 4.9.6) to control various aspects of the axes within each window, for example to add axis titles or to specify the spacing of tick marks or the position of labels. See lines 34 and 35 of Example 4.1, which define the lower and upper boundaries of the x- and yaxes, and the positions of the tick marks. These directives also allow you to specify which pens should be used for drawing the axes and adding annotation.

There are also ways in which you can control the output device. There are two options in DGRAPH (and in the other plotting directives) that can be used for this: SCREEN and ENDACTION. By default, when plotting a graph, GenStat will first clear the screen (or, equivalently, start a new page), but you can set option SCREEN=keep to preserve the current display. **DGRAPH and D3GRAPH have an additional setting, SCREEN=resize, that will adjust the bounds of the axes, if necessary, to include the new information.** Otherwise, the bounds are defined by the initial plot. DGRAPH can thus be used not only as a means of producing a self-contained picture, but also as a basic drawing tool to build up a complicated picture in several stages. Further details are given in Section 4.8

The ENDACTION option of DGRAPH is useful when producing graphs on an interactive graphical device; it specifies whether GenStat should pause at the completion of a graph, waiting for the user to press a key before continuing, or should immediately continue to the next statement. Where the screen has to switch between text and graphics modes, you would normally want to pause so that you could look at the graph; whereas using a windowed display, as on PCs running Windows, this is unnecessary unless several graphs are being drawn in succession, for example by a procedure. The default for ENDACTION, in the initial environment, uses the setting most suited to the current device. However, this can be modified by the DEVICE directive (4.9.1.1). ENDACTION is ignored when output is to a file.

4.2 High-resolution graphs in two and three dimensions

4.2.1 The DGRAPH directive

DGRAPH directive

Draws graphs on a plotter or graphics monitor.

Options

TITLE = *text*	General title; default *
WINDOW = *scalar*	Window number for the graphs; default 1
KEYWINDOW = *scalar*	Window number for the key (zero for no key); default 2
SCREEN = *string token*	Whether to clear the screen before plotting or to continue plotting on the old screen (clear, keep, resize); default clea
KEYDESCRIPTION = *text*	Overall description for the key; default *
ENDACTION = *string token*	Action to be taken after completing the plot (continue, pause); default * uses the setting from the last DEVICE statement
HOTMENU = *matrices*	Defines sets of "hot" components for the user to select as shown or hidden by a menu in the Graphics Viewer
HOTCHOICE = *string token*	Whether one or several "hot" components can be displayed at a time (one, several); default seve

Parameters

Y = *identifiers*	Vertical coordinates
X = *identifiers*	Horizontal coordinates
PEN = *scalars*, *variates* or *factors*	Pen number for each graph (use of a variate or factor allows different pens to be defined for different sets of units); default * uses pens 1, 2, and so on for the successive graphs
DESCRIPTION = *texts*	Annotation for key
YLOWER = *identifiers*	Lower values for vertical bars
YUPPER = *identifiers*	Upper values for vertical bars
XLOWER = *identifiers*	Lower values for horizontal bars
XUPPER = *identifiers*	Upper values for horizontal bars
YBARPEN = *scalars*, *variates* or *factors*	Pens to use to draw the vertical bars; default uses the pen defined by PEN; default -11
XBARPEN = *scalars*, *variates* or *factors*	Pens to use to draw the horizontal bars; default uses the pen defined by PEN;

	default -11
LAYER = *scalars*	"Layer" of the plot
UNITNUMBERS = *identifiers*	Specifies unit numbers to be used when points are selected in the graphics viewer; default * uses the actual unit numbers of the values in the X and Y structures
DISPLAY = *string tokens*	Whether to display each component initially in the graph (show, hide); default show
HOTCOMPONENT = *scalars*	Allows components of the graph (specified by pairs of Y and X settings) to be defined as "hot" components that can be shown or hidden through their association with "hot" points or using a menu in the Graphics Viewer
HOTDEFINITION = *matrices*	Defines how to use points defined by the Y and X parameters as "hot" points in the Graphics Viewer to allow the user to decide whether other components of the graph are shown or hidden
YAXIS = *identifiers*	Axis associated with the Y values; default is to use the yaxis of the window used for the plot **(is not available)**
XAXIS = *identifiers*	Axis associated with the X values; default is to use the xaxis of the window used for the plot **(is not available)**

The DGRAPH directive draws high-resolution graphs, containing points, lines or shaded polygons. The graph is produced on the current graphics device; this can be selected using the DEVICE directive as explained in 4.9.1.1. The WINDOW option defines the window, within the plotting area, in which the graph is drawn; by default this is window 1.

The Y and X parameters specify the coordinates of the points to be plotted; they must be numerical structures (scalars, variates, matrices or tables) of equal length. If any of the variates is restricted, only the subset of values specified by the restriction will be included in the graph. The restrictions are applied to the Y and X variates in pairs, and do not carry over to all the variates in a list. For example, suppose the variate Y1 is restricted but the variate Y2 is not. The statement

```
DGRAPH Y1,Y2; X
```

will plot the subset of values of Y1 against X, but all the values of Y2 against X. Conversely, if X were restricted the subset would be plotted for both Y1 and Y2. Any associated structures, like variates specified by the PEN parameter or factors used to provide labels for the points, must be of the same length as Y and X.

Each pair of Y and X structures has an associated pen, specified by the PEN parameter. By default, pen 1 is used for the first pair, pen 2 for the second, and so on. The type of graph that is produced is determined by the METHOD setting of that pen. This can be point, to produce a point plot or scatterplot; line to join the points with straight lines; monotonic, open or closed to plot various types of curve through the points; spline to plot a smoothing spline fitted to the points; or fill to produce shaded polygons. In the initial graphics environment, all the pens are defined to produce point plots. This can be modified using the METHOD option of the PEN directive (4.9.8). Other attributes of the pen can be used to control the colour, symbols and labels as described in 4.9.8.

With METHOD=fill, the points defined by the Y and X variates are joined by straight lines to form one or more polygons which are then filled using the brush style specified for the pen. The JOIN parameter of PEN determines the order in which the points are joined; with the default, ascending, the data are sorted into ascending order of x-values, while with JOIN=given they are left in their original order. There should be at least three points when using this method.

A warning message is printed if the data contain missing values. The effect of these depends on the type of graph being produced, as follows. If the method is point there will be no indication on the graph itself that any points were missing (but obviously none of the points with missing values for either the y- or x-coordinate can be included in the plot). If a line or curve is plotted through the points there will be a break wherever a missing value is found; that is, line segments will be omitted between points that are separated by missing values. When using METHOD=fill missing values will, in effect, define subsets of points, each of which will be shaded separately. Note, however, that the position of the missing values within the data will differ according to whether or not the data values have been sorted; this is controlled by the JOIN parameter of PEN, as described above. If the data are sorted, units with missing x-values are moved to the beginning.

The PEN parameter can also be set to a variate or factor, to allow different pens to be used for different subsets of the units. With a factor, the units with each level are plotted separately, **using the pen defined by the ordinal number of the level concerned**. If PEN is set to a variate, its values similarly define the pen for each unit. For example, if you fit separate regression lines to some grouped data, you can easily plot the fitted lines in just two statements, one to set up the pens and one to plot the data:

```
PEN     1...Ngroups; METHOD=line; SYMBOL=0
DGRAPH Fitted; X; PEN=Groups
```

By default, GenStat calculates bounds on the axes that are wide enough to include all the data; the range of the data is extended by five percent at each end, and the axes are drawn on the lefthand side and bottom edge of the graph. This can all be changed by the XAXIS and YAXIS directives (4.9.4 and 4.9.5), using the LOWER and UPPER parameters to set the bounds, and YORIGIN and XORIGIN to control the position of the axes. Other parameters allow you to control the axis labelling and style. **If the axis bounds are too narrow, some points may be excluded from the graph**, so that *clipping* occurs. If the plotting method is point, GenStat ignores points that are out of bounds. For other settings of METHOD, lines are drawn from points that are within bounds towards points that are out of bounds, terminating at the appropriate edge. Clipping may also occur if the method is monotonic, open or closed and you have left GenStat to set default axis bounds, because these methods fit curves that may extend beyond the boundaries. If this occurs, you should use the XAXIS and YAXIS directives to provide increased axis bounds.

When you use several DGRAPH statements with SCREEN=keep to build up a complex graph, the axes are drawn only the first time, and the same axes bounds are then used for the subsequent graphs. You should then define axis limits that enclose all the subsequent data. **Alternatively, if you set SCREEN=resize, the axes and their bounds will be adjusted**, if necessary, to enclose the additional information. Axes are drawn only if SCREEN=clear, or the specified window has not been used since the screen was last cleared, or the window has been redefined by a FRAME statement.

If you want to display several different types of measurement together on the same plot, you may want to associate some sets of y- and x-values with different axes from the standard axes of the plotting window. To do this, you must first use the AXIS directive to define another axis within the window. Then associate that axis with the relevant set of y-values (or x-values) by setting the YAXIS (or XAXIS) parameter of DGRAPH to the identifier of the axis, as defined by the IDENTIFIER parameter of AXIS.

DGRAPH allows *error bars* to be included in the plot. You might want to use these, for example, to show confidence limits on points that have been fitted by a regression. Error bars are requested by setting the YLOWER and YUPPER parameters to variates defining the lower and upper values for the error bar to be drawn at each point. For example, if you know the standard error for each point, you could calculate and plot the bounds as follows:

```
CALCULATE Barlow = Y - 1.96 * Err
&         Barhigh = Y + 1.96 * Err
DGRAPH    Y; X; YLOWER=Barlow; YUPPER=Barhigh
```

(this would give a 95% confidence interval assuming that the y-values come from a Normal distribution). The error bar is drawn from the lower point to the upper point at the associated xposition; the bar will be drawn even if the corresponding y-value (or y-variate) is missing. **If the lower value is missing, or the YLOWER parameter is not set, only the upper section of the bar is drawn**; likewise if the upper value is missing only the lower section is drawn. Similarly, parameters XLOWER and XUPPER allow you to plot horizontal bars at each point.

The YBARPEN and XBARPEN parameters define the pens to be used for the vertical and horizontal bars, respectively, with the default to use pen -11. Similarly to the PEN parameter, they can be set to either scalars, factors or variates. For each group of units defined by the setting of PEN, DGRAPH will use the first pen that it finds for that group in the setting supplied by YBARPEN and XBARPEN. (So YBARPEN and XBARPEN cannot define more detailed groupings of the points than those defined by PEN.) For example:

```
VARIATE [VALUES=1,1,2,2,3,3] Pvar
&       [VALUES=4,4,5,5,6,6] Ybvar
&       [VALUES=7,7,8,8,9,10] Xbvar
DGRAPH  Y; X; PEN=Pvar; YLOWER=Ylow; YUPPER=Yupp;\
        XLOWER=Xlow; XUPPER=Xupp; YBARPEN=Ybpen; XBARPEN=Xbpen
```

The first two points here will be plotted in pen 1 with vertical bar in pen 4 and horizontal bar in pen 7. The third and fourth points will be plotted in pen 2 with vertical bar in pen 5 and horizontal bar in pen 8. The fifth and sixth points will be plotted in pen 3 with vertical bar in pen 6 and horizontal bar in pen 9. Notice, that the horizontal bar for the sixth point will be plotted in pen 9 *not* pen 10, as it is in the same PEN group as the (earlier) fifth point which has pen 9 for the horizontal bar. However, if PEN is not set to a factor or variate, the YBARPEN and XBARPEN settings define the groups.

The KEYWINDOW option specifies the window in which the key appears; by default this is window 2. Alternatively, you can set KEYWINDOW=0 to suppress the key. The key contains a line of information for each pair of Y and X structures, written with the associated pen. This will indicate the symbol used, the line style (for a plotting method of line or curve) or a shaded block to illustrate the brush style (when METHOD=fill), the name of the structure (if any) defined by the LABELS parameter of PEN, and a description indicating the identifiers of the data plotted (for example Residuals v Fitted). Alternatively, you can supply your own key, using the DESCRIPTION parameter, and you can specify a title for the key using the KEYDESCRIPTION option. If you draw several graphs using SCREEN=keep or SCREEN=resize and the same key window, each new set of information is appended to the existing key, until the window is full.

If you have set the PEN parameter to a variate or factor in order to plot independent subsets of the data, the key will contain information for each subset**. If the LABELS parameter of PEN has been used to specify labels for the points, each line of the key will contain the label corresponding to the first value of the subset, rather than the identifier of the labels structure itself**. In lines 22 and 23 of Example 4.2.1 the factor Animal is used to label the points according to the type of animal. Alternatively, in lines 24 and 25 the animals are distinguished with different plotting symbols, by using the factor to specify different pens for the different types of animal. The resulting plots are in Figure 4.2.1.

Example 4.2.1

```
  2  " Use of factors for labels and pens."
  3  FACTOR [LABELS=!T(zebra,giraffe)] Animal
  4  READ   Animal,Height,Weight; FREPRESENTATION=labels

       Identifier   Minimum      Mean   Maximum     Values    Missing
           Height    0.3080     4.384     9.228         14          0
           Weight     18.74     111.9     181.6         14          0

       Identifier    Values   Missing    Levels
           Animal        14         0         2

 19  XAXIS  5,6; LOWER=0; UPPER=240
 20  YAXIS  5,6; LOWER=0; UPPER=10.5
 21  FRAME  7,8; YLOWER=0.3; YUPPER=0.5; XLOWER=0.075,0.59
 22  PEN    1; LABELS=Animal; SIZE=0.8
 23  DGRAPH [WINDOW=5; KEYWINDOW=7] Height; Weight
 24  PEN    1,2; SYMBOLS=2,7; LABELS=*
 25  DGRAPH [WINDOW=6; KEYWINDOW=8; SCREEN=keep] Height; Weight; PEN=Animal
```

The TITLE option can be used to provide a title for the graph. You can also put titles on the axes by using the TITLE parameter of the XAXIS and YAXIS directives.

The SCREEN option controls whether the graphical display is cleared before the graph is plotted and the ENDACTION option controls whether GenStat pauses at the end of the plot, as described at the start of this section.

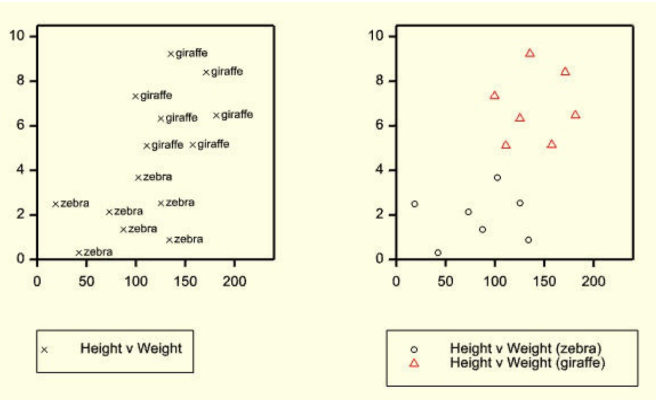


Figure 4.2.1

By default the sets of points defined by each pair of Y and X parameter settings are assumed to form separate, successive "layers" on the plot. So, if an area of the plot contains information (lines, symbols or labels) from several pairs of Y and X settings, **the information from the later settings will overlay the information from earlier settings**. You can control the orders of the layers by using the LAYER parameter to assign an explicit layer number to each pair of Y and X settings. The pairs of Y and X settings are then plotted in ascending order of layer numbers. **These layer numbers also work across DGRAPH statements when you add to a plot by setting option SCREEN=keep or SCREEN=resize**. So, for example, you can specify lower layer numbers to plot the new information "below" the layers formed by the earlier DGRAPH statement(s).

Usually all these components of the graph are shown when the graph is plotted. In GenStat *for Windows*, the Graphics Editor (which can be opened from the Edit menu on the menu bar of the Graphics Viewer) allows you to show or hide components, and the DISPLAY parameter of DGRAPH allows you to define whether a component should be shown or hidden in the initial graph displayed by the Graphics Viewer.

Alternatively, the Graphics Viewer itself can allow components to be shown or hidden, either by using their association with some "hot" points that have been defined on the graph, or by using a menu on its menu bar. These "hot" components are identified by defining a unique integer number for each one, using the HOTCOMPONENT parameter; if the component is not to be treated as "hot", HOTCOMPONENT should be left unset or given a missing value. Several pairs of Y and X parameter settings can be given the same number, so you can build up a "hot" component from more than one type of graphical item (e.g. from plotted points and shaded areas). "Hot" points are plotted within the graph using the Y, X and other parameters (e.g. PEN) in the usual way, as described above. The extra information, to define them as "hot", is supplied by setting the HOTDEFINITION parameter to a matrix with a row for each "hot" point, and a column for each type of "hot" component. The elements of the matrix specify the "hot" components to be associated with each "hot" point, using the numbers defined by the HOTCOMPONENT parameter. The menus in the Graphics Viewer can be made more informative, by defining textual labels for the rows and columns of the matrix (see the MATRIX directive); these are then used as annotation in the menus. Alternatively, if you set the HOTMENU option to a similar matrix, the Graphics Viewer will include a menu on its menu bar to allow users to choose whether "hot" components are shown or hidden. By default, users will be allowed to display several "hot" components at a time. However, you can set option HOTCHOICE=one to indicate that only one can be shown at a time. (The DISPLAY parameter should then be used to indicate which one, if any, should be shown on the initial graph.). **For an example see the program in Chapter 3 that produces Figures 3.7, 3.8 and 3.9.**

The Graphics Viewer also has a tool that allows you to select points, and copy their unit numbers onto the clipboard. **For a demonstration see Figures 3.5 and 3.6.** Usually these numbers are simply the locations of the plotted values in the X and Y structures. However, you can use the UNITNUMBERS parameter to supply other numbers. (This may be useful if, for example, you are plotting sorted values.)

4.2.2 The D3GRAPH directive

D3GRAPH directive

Plots a 3-dimensional graph.

Options

TITLE = *text*	General title; default *
WINDOW = *scalar*	Window number for the plots; default 1
KEYWINDOW = *scalar*	Window number for the key (zero for no key); default 2
ELEVATION = *scalar*	The elevation of the viewpoint relative to the surface; default 25 (degrees)
AZIMUTH = *scalar*	Rotation about the horizontal plane; the default of 225 degrees ensures that a point at the minimum x- and yvalue is nearest to the viewpoint
DISTANCE = *scalar*	Distance of the viewpoint from the centre of the grid on the base plane; default * ensures that the data points fill the viewing area
SCREEN = *string token*	Whether to clear the screen before plotting or to continue plotting on the old

	screen (clear, keep, resize); default clea
KEYDESCRIPTION = *text*	Overall description for the key; default *
ENDACTION = *string token*	Action to be taken after completing the plot (continue, pause); default * uses the setting from the last DEVICE statement

Parameters

X = *identifiers*	X-coordinates
Y = *identifiers*	Y-coordinates
Z = *identifiers*	Z-coordinates
PEN = *scalars*, *variates* or *factors*	Pen number for each graph (use of a variate or factor allows different pens to be defined for different sets of units); default * uses pens 1, 2, and so on for the successive graphs
DESCRIPTION = *texts*	Annotation for key
UNITNUMBERS = *identifiers*	Specifies unit numbers to be used when points are selected in the graphics viewer; default * uses the actual unit numbers of the values in the X and Y structures

---

The D3GRAPH directive produces high-resolution graphs, containing points, lines or filled shapes in three dimensions. The graph is produced on the current graphics device which can be selected using the DEVICE directive. The WINDOW option defines the window, within the plotting area, in which the graph is drawn; by default this is window 1.

The position of the viewpoint is specified in polar coordinates, using the options ELEVATION, DISTANCE and AZIMUTH. These define the angle of elevation, in degrees, above the base plane of the surface, distance from the centre of this plane, and angular position relative to the vertical z-axis, respectively.

The default settings of ELEVATION, DISTANCE and AZIMUTH have been chosen to produce a reasonable display of most situations; but if, for example, some parts of the plot are obscured they can be modified to obtain a better view. Altering the value of AZIMUTH will, in effect, rotate the plot in the horizontal plane about a vertical axis drawn through the centre of the plot; the default value of 225 degrees ensures that a point with the minimum x- and y-value would be at the corner nearest the viewpoint.

The X, Y and Z parameters specify the coordinates of the points to be plotted; they must be numerical structures (scalars, variates, factors, matrices or tables) of equal length. If any of the variates or factors is restricted, only the subset of values specified by the restriction will be included in the graph. The restrictions are applied to the X, Y and Z variates or factors in parallel sets, and do not carry over to other variates or factors in the list. Any associated structures, like variates specified by the PEN parameter or factors used to provide labels for the points, must be of the same length as X, Y and Z.

Each set of X, Y and Z structures has an associated pen, specified by the PEN parameter. By default, pen 1 is used for the first set, pen 2 for the second, and so on. **The type of graph that is produced is determined by the METHOD setting of that pen. This can be point,** to produce a point plot or scatterplot; **line** to join the points with straight lines; **or fill** to produce shaded objects. In the initial graphics environment, all the pens are defined to produce point plots. This can be modified using the METHOD option of the PEN directive (4.9.8). Other attributes of the pen can be used to control the colour, symbols and labels.

With METHOD=fill, the points defined by the X, Y and Z variates are joined by straight lines to form one or more polygons or polyhedrons which are then filled using the brush style specified for the pen. **The JOIN parameter of PEN is ignored for this directive. The points are plotted in the order in which they occur in the data.**

A warning message is printed if the data contain missing values. The effect of these depends on the type of graph being produced, as follows. If the method is point there will be no indication on the graph itself that any points were missing (but obviously none of the points with missing values for either the x-, y- or z-coordinate can be included in the plot). If a line is plotted through the points there will be a break wherever a missing value is found; that is, line segments will be omitted between points that are separated by missing values. When using METHOD=fill missing values will, in effect, define subsets of points, each of which will be shaded separately.

The PEN parameter can also be set to a variate or factor, to allow different pens to be used for different subsets of the units. With a factor, the units with each level are plotted separately, using the pen defined by the level concerned. If PEN is set to a variate, its values similarly define the pen for each unit. For example, if you fit separate regression lines to some grouped data, you can easily plot the fitted lines in just two statements, one to set up the pens and one to plot the data:

```
PEN     1...Ngroups; METHOD=line; SYMBOL=0
D3GRAPH Fitted; X1; X2; PEN=Groups
```

By default, GenStat calculates bounds on the axes that are wide enough to include all the data; the range of the data is extended by five percent at each end, and the axes are drawn on the lefthand side and bottom edge of the graph. This can all be changed by the XAXIS, YAXIS and ZAXIS directives (4.9.4, 4.9.5 and 4.9.6) using the LOWER and UPPER parameters to set the bounds, and XORIGIN, YORIGIN and ZORIGIN to control the positions of the axes. Other parameters allow you to control the axis labelling and style. If the axis bounds are too narrow, some points may be excluded from the graph, so that *clipping* occurs. If the plotting method is point, GenStat ignores points that are out of bounds. For other settings of METHOD, lines are drawn from points that are within bounds towards points that are out of bounds, terminating at the appropriate edge. Clipping may also occur if the method is monotonic, open or closed and you have left GenStat to set default axis bounds, because these methods fit curves that may extend beyond the boundaries. If this occurs you should use the relevant axis directive to provide increased axis bounds.

When you use several D3GRAPH statements with SCREEN=keep to build up a complex graph, the axes are drawn only the first time, and the same axes bounds are then used for the subsequent graphs. You should then define axis limits that enclose all the subsequent data. **Alternatively, if you set SCREEN=resize, the axes and their bounds will be adjusted, if necessary, to enclose the additional information.** Axes are drawn only if SCREEN=clear, or the specified window has not been used since the screen was last cleared, or the window has been redefined by a FRAME statement.

The KEYWINDOW option specifies the window in which the key appears; by default this is window 2. Alternatively, you can set KEYWINDOW=0 to suppress the key. The key contains a line of information for each pair of Y and X structures, written with the associated pen. This will indicate the symbol used, the line style (for a plotting method of line or curve) or a shaded block to illustrate the brush style (when METHOD=fill), the name of the structure (if any) defined by the LABELS parameter of PEN, and a description indicating the identifiers of the data plotted (for example Residuals v Fitted). Alternatively, you can supply your own key, using the DESCRIPTION parameter, and you can specify a title for the key using the KEYDESCRIPTION option. If you draw several graphs using SCREEN=keep or SCREEN=resize and the same key window, each new set of information is appended to the existing key, until the window is full.

If you have set the PEN parameter to a variate or factor in order to plot independent subsets of the data, the key will contain information for each subset. If the LABELS parameter of PEN has been used to specify labels for the points, each line of the key will contain the label corresponding to the first value of the subset, rather than the identifier of the labels structure itself.

The Graphics Viewer has a tool that allows you to select points, and copy their unit numbers onto the clipboard. Usually these numbers are simply the locations of the plotted values in the X and Y structures. However, you can use the UNITNUMBERS parameter to supply other numbers. (This may be useful if, for example, you are plotting sorted values.) This is the same as for a 2-dimensional plot. See for a demonstration Figures 3.5 and 3.6.

The TITLE option can be used to provide a title for the graph. You can also put titles on the axes by using the TITLE parameters of the XAXIS, YAXIS and ZAXIS directives (4.9.4, 4.9.5 and 4.9.6). The SCREEN option controls whether the graphical display is cleared before the graph is plotted and the ENDACTION option controls whether GenStat pauses at the end of the plot.

Example 4.2.2 shows the use of D3GRAPH to plot three of the variables in Fisher's Iris data; the resulting plot is in Figure 4.2.2.

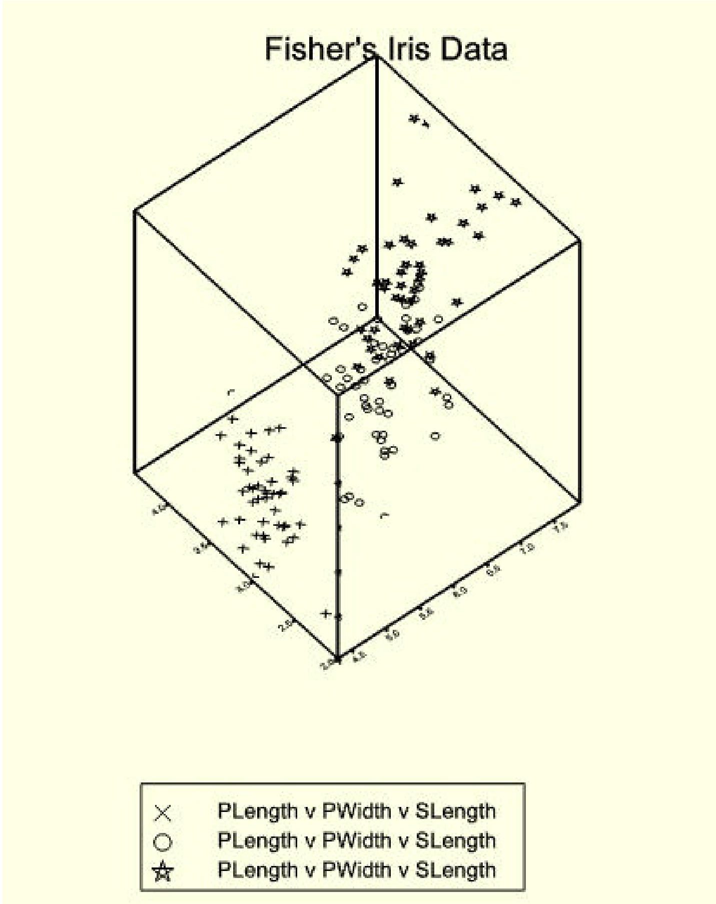


Figure 4.2.2

Example 4.2.2

```
 2  VARIATE [NVALUES=150] PLength,PWidth,SLength,SWidth
 3  READ PLength,PWidth,SLength,SWidth
```

Identifier	Minimum	Mean	Maximum	Values	Missing
PLength	4.300	5.843	7.900	150	0
PWidth	2.000	3.057	4.400	150	0
SLength	1.000	3.758	6.900	150	0
SWidth	0.1000	1.199	2.500	150	0

```
 4  FACTOR    [NVALUES=150; LABELS=!t(Setosa,Versicolor,Virginica); \
 5            VALUES=50(1,2,3)] Species
 6  PEN       1,2,3; SYMBOL=1,2,4; COLOUR='black'; SIZE=1.5
 7  D3GRAPH   [TITLE='Fisher''s Iris Data'; AZIMUTH=230; ELEVATION=50]\
 8            PLength; PWidth; SLength; PEN=Species
```

4.3 Histograms and bar charts

Histograms are used to represent the distribution of a set of data values. In the standard histogram, the range of the data is partitioned into consecutive intervals. The histogram has a bar for each interval, and the height of the bar represents the number of data values that the interval contains. Essentially, the histogram provides a graphical representation of a one-way table of counts; see Example 4.3.1a. The DHISTOGRAM directive (4.3.1) provides high-resolution plots of histograms with a single classifying factor. If you have two classifying factors, you can use the D3HISTOGRAM directive (4.4.4), which provides a plot in three dimensions: one for the lengths of the bars, and two for the classifying factors.

The bar chart differs from the histogram in that the factor classifying the table can represent any type of grouping, not simply a partitioning of a range of numerical values: for example a bar chart might present sales of a products in different areas, or sales of different types of product. Also, the table need not contain counts, but may contain any numerical values, for example profits (and losses), or yields of a crop. Bar charts with either one or two classifying factors can be plotted using the BARCHART directive (4.3.2).

4.3.1 The DHISTOGRAM directive

DHISTOGRAM directive

Draws histograms or bar charts on a plotter or graphics monitor.

Options

TITLE = *text*	General title; default *
WINDOW = *scalar*	Window number for the histograms; default 1
KEYWINDOW = *scalar*	Window number for the key (zero for no key); default 2
LIMITS = *variate*	Variate of group limits for classifying DATA variates into groups; default *
LOWER = *scalar*	For a DATA variate, this specifies the lower limit of the first bar; default * takes the minimum value of the variate
UPPER = *scalar*	For a DATA variate, this specifies the upper limit of the last bar; default * takes the maximum value of the variate
NGROUPS = *scalar*	When LIMITS and BINWIDTH are not specified, this defines the number of groups into which a DATA variate is to be classified; default is then 10, or the integer value nearest to the square root of the number of values in the variate if that is smaller
BINWIDTH = *scalar*	When LIMITS is unset the range of a DATA variate is split into equal intervals known as "bins" to form the groups, this option can set the bin widths (alternative is to set the number of groups using NGROUPS)
FIXEDBARWIDTH = *string token*	Whether to plot the histogram with bars of equal width (no, yes); default no
BARCOVERING = *scalar*	What proportion of the space allocated along the x-axis each bar should occupy; default * gives proportion 1 for a DATA variate, and 0.8 for a factor or table (thus giving a gap between each bar)
BARSCALE = *scalar*	Width of bar for which one unit of bar length represents one unit of data; default * uses the width of the narrowest bar
LABELS = *text*	Group labels; default *
APPEND = *string token*	Whether or not the bars of the histograms are appended together (yes, no); default no
ORIENTATION = *string token*	Direction of the plot (horizontal, vertical); default vert
OUTLINE = *string token*	Where to draw outlines (bars, perimeter); default bars
PENOUTLINE = *scalar*	Pen to use for the outlines; default -8
SCREEN = *string token*	Whether to clear the screen before plotting or to continue plotting on the old screen (clear, keep); default clea
KEYDESCRIPTION = *text*	Overall description for the key; default *
ENDACTION = *string token*	Action to be taken after completing the plot (continue, pause); default * uses the setting from the last DEVICE statement

Parameters

DATA = *identifiers*	Data for the histograms; these can be either a factor indicating the group to which each unit belongs, a variate whose values are to be grouped, or a one-way table giving the height of each bar
NOBSERVATIONS = *tables*	One-way table to save numbers in the groups
GROUPS = *factors*	Factor to save groups defined from a variate
PEN = *scalars* or *variates*	Pen number(s) for each histogram; default * uses pens 2, 3, and so on for the successive structures specified by DATA
DESCRIPTION = *texts*	Annotation for key

DHISTOGRAM plots high-resolution histograms or bar charts, depending on the input supplied by the DATA parameter. This can be either a list of variates, a list of factors or a list of one-way tables.

For a DATA variate, a histogram is produced. This summarizes the distribution of the variate by counting the number of values within a set of intervals defined by the LIMITS, NGROUPS or BINWIDTH options. The histogram contains a "bar" for each interval, with area proportional to the number of values found there.

Example 4.3.1a produces a histogram from a variate called Data, whose values are listed (in numerical order) in lines 4 and 5. The resulting graph is in Figure 4.3.1a.

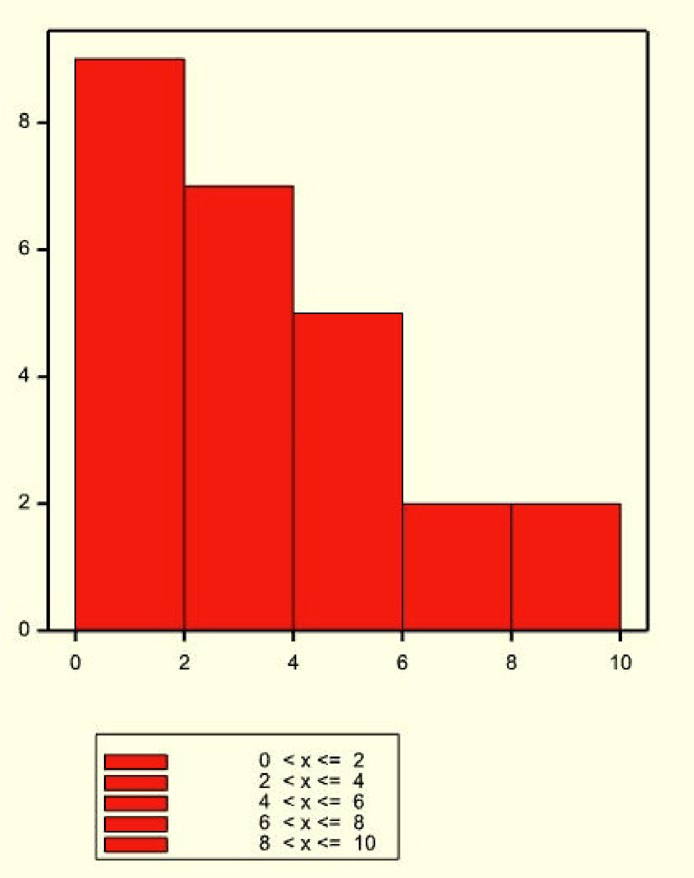


Figure 4.3.1a

Example 4.3.1a

```
 2  VARIATE Data
 3  READ    [PRINT=data,errors] Data
 4   0  1  1  1  1  2  2  2  2  3  3  4  4  4  4  4
 5   5  5  6  6  6  7  8  9  9  :
 6  DHISTOGRAM Data
```

You can define the boundaries between each interval using the LIMITS option. Alternatively, instead of setting LIMITS, you can specify the width of each interval using the BINWIDTH option. Or, instead of setting LIMITS or BINWIDTH, you can specify the number of groups using the NGROUPS option. **Finally, if none of these options is set, GenStat defines the number of groups to be 10, or the integer value nearest to the square root of the number of values in the first DATA variate if that is smaller**. The range of the histogram is specified by the LOWER and UPPER options. LOWER defines the lower limit of the first interval; by default this is set by making the width of the first bar equal to the width of the second bar, or it is the minimum value of the variates if that would otherwise be below the first bar. UPPER defines the upper limit of the last interval; by default this is set by making the width of the final bar equal to the width of the lastbut-one bar, or it is the maximum value of the variates if that would otherwise be above the final bar. The bars are perpendicular to the x-axis, and this is labelled with the positions of the interval boundaries.

Bar charts are given if DATA is set to factors or tables. These differ from histograms in that there is no longer the concept of dividing the x-axis into a set of contiguous intervals. Instead we have a set of bars located at equal intervals along the x-axis. Figure 4.3.1b shows an example, generated by Example 4.3.1b below.

If DATA is set to a list of factors, the bars are labelled by the labels, if available, or otherwise the levels of **the first factor.** If DATA is set to a list of tables, the labelling is given by the levels/labels of the factor **classifying the first table**. A DATA table defines the heights of each bars directly (from the value in the corresponding cell of the table). With a factor, GenStat first constructs a table giving the replications of the factor levels. So the height of each bar is equal to the number of units of the factor with the corresponding level of the factor.

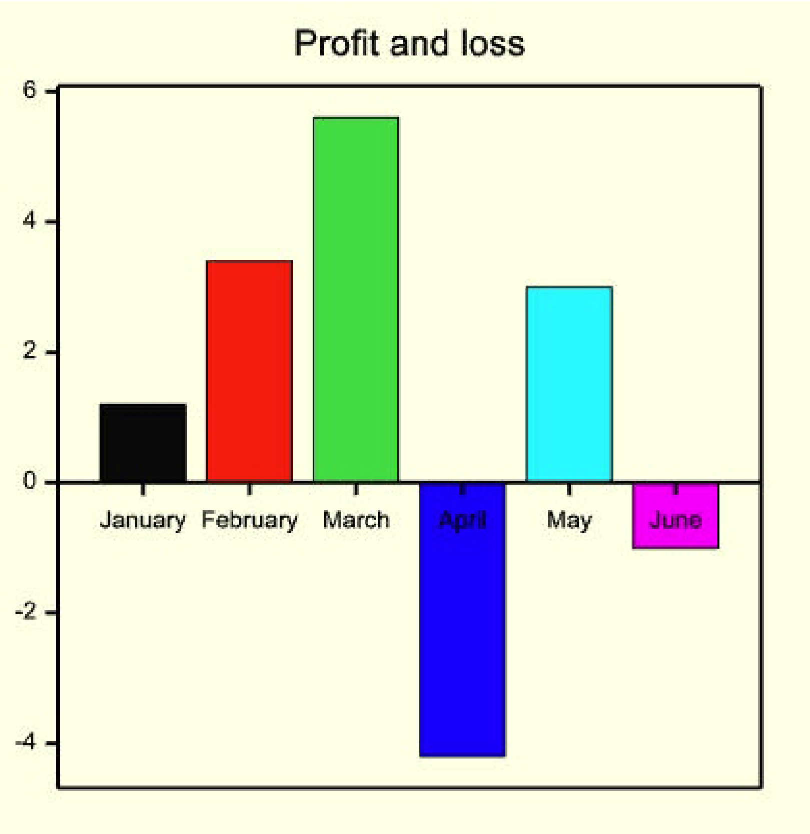


Figure 4.3.1b

Example 4.3.1b

```
2  FACTOR [LABELS=!t(January,February,March,April,May,June)] Month
3  TABLE  [CLASSIFICATION=Month; VALUES=1.2,3.4,5.6,-4.2,3.0,-1] Results
4  DHISTOGRAM [TITLE='Profit and loss'; WINDOW=1; KEYWINDOW=0] Results
```

The bars in a bar chart always have equal widths. With a histogram, the default is for the bar widths to be equal to the widths of the underlying intervals. However, you can request equal bar widths by setting option FIXEDBARWIDTH=yes. The BARCOVERING option indicates what proportion of the space allocated along the x-axis each bar should occupy. For a histogram the default is 1, while for bar charts it is 0.8 (thus giving a gap between each bar).

The BARSCALE option controls how the lengths of the bars correspond to units of data. **The length of each bar is calculated as (data-value × BARSCALE)/bar-width**. By default, BARSCALE is set to the width of the narrowest bar. So for that bar, the length will correspond directly to the data units.

The WINDOW option defines the window where the histogram is plotted, and the KEYWINDOW option similarly specifies where the key should appear. You can set either of these to zero if you want to suppress the corresponding output. Titles can be added to the histogram and key using the TITLE and KEYDESCRIPTION options respectively.

The APPEND option controls the form of display used when the DATA parameter specifies a list of structures. These parallel histograms can be produced in one of two styles. By default (APPEND=no) , the histogram contains a set of bars for each structure, drawn in parallel groups. This is used in line 15 of Example 4.3.1c to generate the histogram on the left-hand side of Figure 4.3.1c.

Alternatively, if you set APPEND=yes, the bars for the structures are concatenated into a single bar for each group, as in line 16 of Example 4.3.1c and the right-hand side of Figure 4.3.1c. The bottom portion of each bar then corresponds to the first structure, and the top to the last structure.

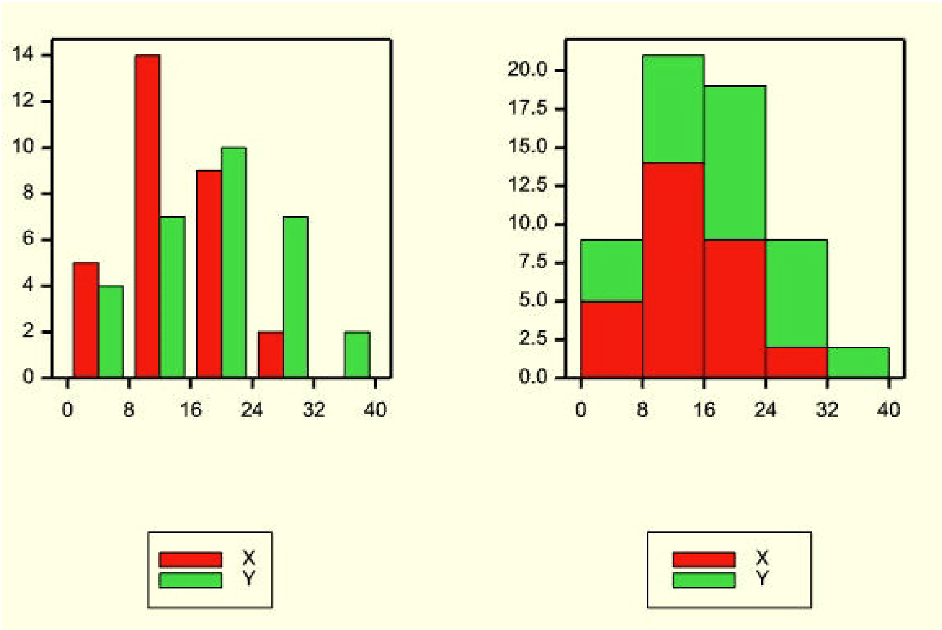


Figure 4.3.1c

Example 4.3.1c

```
 2  READ  X,Y

      Identifier    Minimum       Mean    Maximum      Values     Missing
               X      2.801      14.25      29.01          30           0
               Y      4.069      19.33      39.01          30           0

11  FRAME 7,8; YUPPER=0.45; XLOWER=0.2,0.7
12  PEN   1,2; BRUSH=2,9
13  XAXIS 5,6; MARKS=!(0,8...40)
14  YAXIS 5,6; MARKS=!(0,2...14),!(0,2.5...20)
15  DHISTOGRAM [WINDOW=5; KEYWINDOW=7; APPEND=no] X,Y
16  DHISTOGRAM [WINDOW=6; KEYWINDOW=8; SCREEN=keep; APPEND=yes] X,Y
17
18  " NB: X,Y=17,23+6,8*NED(URAN(123;30)) "
```

The ORIENTATION option controls whether the bars of the histogram are plotted vertically (the default) or horizontally. When ORIENTATION=horizontal, **the horizontal axis is taken to be the y-axis, so the same XAXIS and YAXIS settings can be used however the histogram is oriented**.

The bars for each structure are all shaded according to the pen or pens that have been specified for that structure, using the PEN parameter. You can set PEN to a scalar to define a single pen to be used for all the bars, or to a variate to define a different pen for each bar. If PEN is not set, GenStat uses the pens in turn, pen 2 for the first structure, pen 3 for the second structure, and so on, so that a different shading is used for each structure. The relevant aspects of the pens should be set in advance, if required, using the BRUSH and COLOUR parameters of the PEN directive (4.9.8). Generally, however, the default attributes of the pens will be satisfactory.

The OUTLINE option controls whether lines are drawn around the bars or around the perimeter of the histogram. These are drawn using the pen specified by the PENOUTLINE option (default -8). **You can suppress all the outlines by setting OUTLINE=***.

The axes of the histogram are formed automatically from the data. By default, the upper bound of the y-axis is set to be five percent greater than the height of the longest bar. If any of the bars has a negative height the lower bound is adjusted in a similar way, otherwise it is set to zero. As already mentioned, when the histogram is formed from a variate, the x-axis markings are set to indicate the limits of each bar or set of bars; when the data are provided in a factor the factor labels or levels are used to label the histogram bars, and when the bar heights are provided directly in a table the classifying factor of the table is used. You can control the form of the axes by using the XAXIS and YAXIS directives (4.9.4 and 4.9.5) to set the required attributes before the DHISTOGRAM directive is used.

The WINDOW parameter of XAXIS and YAXIS should be set to the window in which the histogram is to be plotted (controlled by the WINDOW option of DHISTOGRAM). The TITLE, LOWER, UPPER, MARKS and LABELS parameters control annotation. The UPPER parameter of YAXIS is particularly useful when you are plotting a series of histograms; by setting UPPER to a value larger than any of the bars in any of the histograms, you can ensure that they are all plotted on the same scale.

The histogram key consists of the title, if set by KEYDESCRIPTION, followed by a legend for each structure plotted. This consists of a small rectangle that is drawn in the same colour and brush style as that used in the histogram, followed by the identifier name or the piece of text specified by the DESCRIPTION parameter.

The SCREEN option controls whether the graphical display is cleared before the histogram is plotted and the ENDACTION option controls whether GenStat pauses at the end of the plot.

4.3.2 The BARCHART directive

BARCHART directive

Plots bar charts in high-resolution graphics.

Options

TITLE = *text*	General title; default *
WINDOW = *scalar*	Window number for the histograms; default 1
KEYWINDOW = *scalar*	Window number for the key (zero for no key); default 2
BARWIDTH = *scalar*, *variate* or *table*	Width(s) of the bars; default * sets equal widths to fill the x-axis
BARCOVERING = *scalar*	What proportion of the space allocated along the x-axis each bar should occupy; default * gives proportion 1 for a DATA variate, and 0.8 for a factor or table (thus giving a gap between each bar)
LABELS = *text*	Labels for the bars or groups of bars; default *
APPEND = *string token*	Whether or not the bars of the histograms are appended together (yes, no); default no
ORIENTATION = *string token*	Direction of the plot (horizontal, vertical); default vert
YSCALING = *string token*	What scale to use to label the y-axis (absolute, proportion, percentage); default abso
OUTLINE = *string token*	Where to draw outlines (bars, perimeter); default bars
PENOUTLINE = *scalar*	Pen to use for the outlines; default -9
SCREEN = *string token*	Whether to clear the screen before plotting or to continue plotting on the old screen (clear, keep); default clea
KEYDESCRIPTION = *text*	Overall description for the key; default *
ENDACTION = *string token*	Action to be taken after completing the plot (continue, pause); default * uses the setting from the last DEVICE statement

Parameters

DATA = *tables* or *variates*	Heights of the bars in each bar chart
ERRORBARS = *scalars*, *tables* or *variates*	Error bars to be plotted above the bars of each bar chart
LOWERERRORBARS = *scalars*, *tables* or *variates*	Heights of error bars plotted below the bars of each bar chart; if any of these is omitted, the corresponding setting of ERRORBARS is used as the default so that the error bars will have equal heights above and below the bars of the bar chart
GROUPS = *factors*	Which factor of a 2-way table to use as the groups factor; default uses the second classifying factor

PEN = *scalars*, *tables* or *variates*

Pen number(s) for each bar chart; default * uses pens 2, 3, and so on for the successive structures specified by DATA

PENERRORBARS = *scalars*, *tables* or *variates*

Pen number(s) for the error bars; default -11

DESCRIPTION = *texts* Annotation for key

BARCHART plots high-resolution bar charts. You can plot a single bar chart by setting the DATA parameter to a one-way table or a variate defining the heights of the bars. To plot several bar charts on the same graph, you can set DATA to a list of one-way tables or variates. These must all contain the same number of values, and any tables must be classified by the same factor. **Alternatively, you can set DATA to a two-way table**. The GROUPS parameter then specifies which of the two classifying factors is to be treated as the "groups" factor (**by default this is the second factor**). BARCHART now plots a bar chart for every level of the GROUPS factors, with bars defined by the other classifying factor; see Example 4.3.2 and Figure 4.3.2.

Labels can be supplied for the bars, using the LABELS option. If this is not set, the labels will be the labels or levels of the factor classifying the DATA tables, or the integers 1 upwards for a DATA variate.

By default, if there are several bar charts, they are plotted with their bars alongside each other. So BARCHART first plots the first bar of every bar chart, then the second bar, and so on. Alternatively, you can set option APPEND=yes to stack the bars into a single bar. The bottom portion of each bar then corresponds to the first bar chart, and the top to the last bar chart.

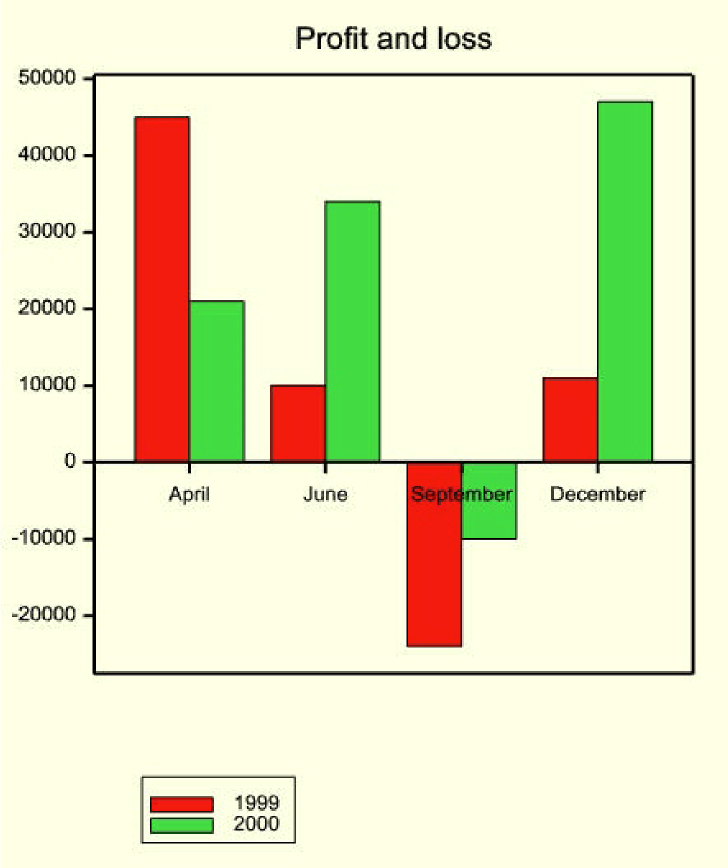


Figure 4.3.2

You can include error bars in a single bar chart or when several bar charts are plotted alongside each other, by specifying their heights with the ERRORBARS and LOWERERRORBARS parameters. The error bars take the form of a horizontal line joined by a vertical line of the specified height, above and below each bar. The ERRORBARS parameter specifies the heights of the error bars above the bars of the bar chart, and the LOWERERRORBARS parameter specifies the heights of the error bars below the bars**. If LOWERERRORBARS is not specified, the error bars are assumed to have the same heights below and above the bars**. You can set ERRORBARS and LOWERERRORBARS to a scalar if the heights are the same for every bar of a bar chart, or to a table or variate if different bars have error bars with different heights.

The ORIENTATION option controls whether the bars of the histogram are plotted vertically (the default) or horizontally. When ORIENTATION=horizontal, the horizontal axis is taken to be the y-axis, so the same XAXIS and YAXIS settings can be used however the histogram is oriented.

By default, GenStat uses pen 2 for the first bar chart, pen 3 for the second bar chart, and so on, so that a different colour or shading is used for each one. Alternatively, you can define your own colours or shading, using the PEN parameter. If you set PEN to a scalar, a single pen is used for all the bars. Alternatively, you can specify a variate or a table to define a different pen for each bar. The relevant aspects of the pens should be set in advance, if required, using the BRUSH and COLOUR parameters of the PEN directive (4.9.8). Generally, however, the default attributes of the pens will be satisfactory. Similarly, the PENERRORBARS parameter specifies the pen or pens to use for the error bars (default -11).

The bars in a bar chart usually have equal widths, defined to fill the available space along the x-axis. However, you can set your own widths by setting option BARWIDTH to either a scalar or a variate or table with as many values as the number of bars. The BARCOVERING option indicates what proportion of the space allocated along the x-axis each bar should occupy; the default is 0.8 (giving a gap between each bar).

The OUTLINE option controls whether lines are drawn around the bars or around the perimeter of the bar chart. These are drawn using the pen specified by the PENOUTLINE option (default -9). **You can suppress all the outlines by setting OUTLINE=* .**

The WINDOW option defines the window where the histogram is plotted, and the KEYWINDOW option similarly specifies where the key should appear. You can set either of these to zero if you want to suppress the corresponding output. Titles can be added to the histogram and key using the TITLE and KEYDESCRIPTION options respectively.

The SCREEN option controls whether the graphical display is cleared before the histogram is plotted and the ENDACTION option controls whether GenStat pauses at the end of the plot.

The axes of the plot are formed automatically from the data. By default, the upper bound of the y-axis is set to be five percent greater than the height of the longest bar. If any of the bars has a negative height the lower bound is adjusted in a similar way, otherwise it is set to zero. You can control the form of the axes by using the XAXIS and YAXIS directives to set the required attributes (such as titles) before the BARCHART directive is used. The YSCALING option controls the scale used to label the y-axis, with settings absolute, proportion or percentage; the default is absolute.

The key consists of the title, if set by KEYDESCRIPTION, followed information about each bar chart. You can specify a description for each bar chart using the DESCRIPTION parameter. If the DATA parameter was set to a list of one-way tables or variates, the default description takes the identifier of the table or variate. If DATA was set to a two-way table, the default descriptions are formed from the labels or levels of the GROUPS factor.

Example 4.3.2

```
 2  FACTOR   [LEVELS=!(1999,2000)] Year
 3  FACTOR   [LABELS=!t(April,June,September,December)] Month
 4  TABLE    [CLASSIFICATION=Year,Month; VALUES=45000,10000,-24000,11000,\
 5            21000,34000,-10000,47000] Results
 6  BARCHART [TITLE='Profit and loss'] Results
```

4.4 Plotting three-dimensional surfaces in high-resolution

The data for a three-dimensional surface consists of a grid of z-values or heights. The grid can be a rectangular matrix, a two-way table, or a pointer to a set of variates; the y-dimension is represented by the rows of the structure and the x-dimension by the columns**. In each case there must be at least three rows and three columns of data** (after allowing for any restrictions on a set of variates). **Missing values are not permitted**; that is, only complete grids can be displayed. If the grid is supplied as a table with margins, **these will be ignored when plotting the surface**.

GenStat provides **four** methods for plotting surfaces. A contour plot (DCONTOUR; 4.4.1) can be used to form a two-dimensional representation of the surface, in which contour lines are drawn to link points of equal height. The DSHADE directive (4.4.2) can produce a shade diagram. This is another two-dimension representation, in which each z-value is represented by a shaded rectangle indicating the value at that location, using either colour or shading density. This type of display is often used in a cluster analysis to display a similarity matrix, but it is also useful for the graphical display of spatial data. Alternatively, a three-dimensional representation of the surface, or *perspective view*, can be drawn using the DSURFACE directive (4.4.3), to display more fully the three-dimensional nature of the data. The grid can be viewed from any angle, allowing the investigation of features such as maxima, minima, valleys and plateaux. When the grid contains discrete data, a three-dimensional (or bivariate) histogram may be appropriate. This is produced using the D3HISTOGRAM directive (4.4.4), which forms the display by drawing cuboid blocks of the appropriate height at each (x,y) position.

DCONTOUR directive

Draws contour plots on a plotter or graphics monitor.

Options

`TITLE` = *text*	General title; default *
`WINDOW` = *scalar*	Window number for the plots; default 1
`KEYWINDOW` = *scalar*	Window number for the key (zero for no key); default 2
`YORIENTATION` = *string token*	Y-axis orientation of the plot (`reverse`, `normal`); default `reve`
`ANNOTATION` = *string token*	How to annotate the contours (`levels`, `ordinals`); default `ordi` if there is a key, and `leve` if there is no key
`SCREEN` = *string token*	Whether to clear the screen before plotting or to continue plotting on the old screen (`clear`, `keep`); default `clea`
`KEYDESCRIPTION` = *text*	Overall description for the key
`ENDACTION` = *string token*	Action to be taken after completing the plot (`continue`, `pause`); default * uses the setting from the last `DEVICE` statement

Parameters

`GRID` = *identifier*	Pointer (of variates representing the columns of a data matrix), matrix or two-way table specifying values on a regular grid
`PENCONTOUR` = *scalar*	Pen number to be used for the contours; default 1
`PENFILL` = *scalar* or *variate*	Pen number(s) defining how to fill the areas between contours, or 0 to leave the areas in the background colour; default 3
`PENHIGHLIGHT` = *scalar*	Pen number to use for highlighted contours; default 0 i.e. no highlighting
`HIGHLIGHTFREQUENCY` = *scalar*	Frequency at which contours are to be highlighted; default 10
`NCONTOURS` = *scalar*	Number of contours; default 10
`CONTOURS` = *variate*	Positions of contours
`INTERVAL` = *scalar*	Interval between contours
`DESCRIPTION` = *text*	Annotation for key

The orientation of the yaxis of the contour plot is controlled by the `YORIENTATION` option. **By default this is reversed, so that the element (1,1) of the grid is plotted at the top left-hand corner of the plot.** The grid is thus in the same order as it would be if it were printed. This is convenient in Example 4.4.1a and Figure 4.4.1a, which shows a contour plot of ammonium nitrate concentrations in soil cores. The y-values represent depth in the soil, and so it is appropriate that they increase down the page.

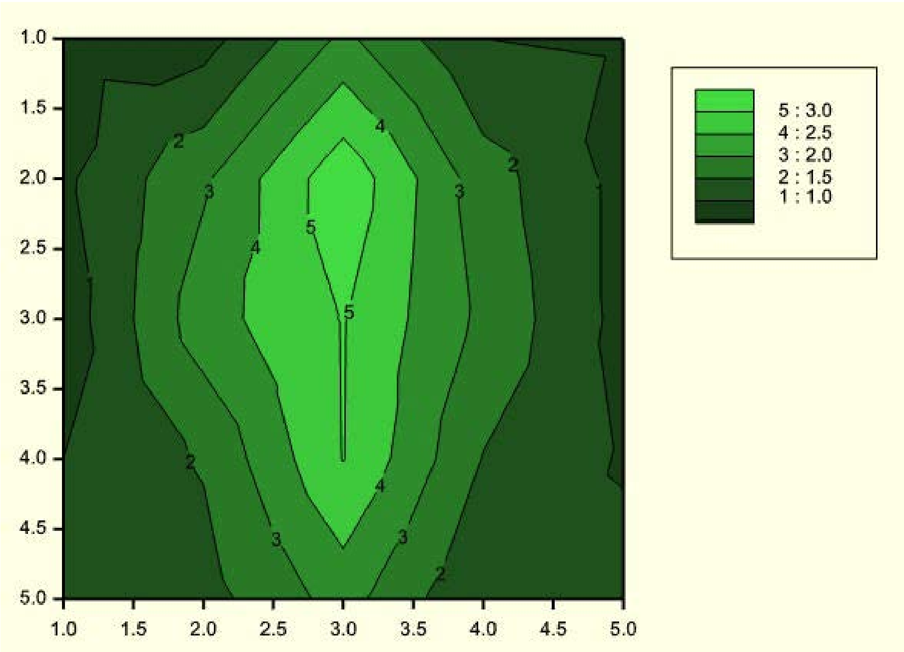


Figure 4.4.1a

Example 4.4.1a

```
 2  " Core samples were taken from a wetland rice experiment to examine
-3    the leaching of ammonium nitrate. Three three cores were taken at
-4    intervals of 5cm, and the concentration of ammonium nitrate was
-5    measured at depths of 4, 8, ... 20 cm. "
 6  VARIATE   [NVALUES=5] Core[1...5]
 7  READ      Core[]
```

Identifier	Minimum	Mean	Maximum	Values	Missing
Core[1]	5.000	8.200	11.00	5	0
Core[2]	6.000	67.60	195.0	5	0
Core[3]	129.0	940.6	2315	5	0
Core[4]	10.00	36.00	77.00	5	0
Core[5]	7.000	9.400	15.00	5	0

```
13  CALCULATE Core[] = LOG10(Core[])
14  FRAME     2; YLOWER=0.0; YUPPER=0.9; XLOWER=0.75; XUPPER=1.0
15  DCONTOUR Core
```

Normally the data will lie on a regular grid but you can also specify an irregular grid as shown in line 10 of Example 4.4.1c; the ROWS and COLUMNS options of the MATRIX directive are set to variates containing the appropriate x- and y- values when the matrix Function is declared.

The WINDOW option defines the window where the contours are plotted, and the KEYWINDOW option similarly specifies where the key should appear. The grid axes are scaled so that the y- and x-dimensions (rows and columns respectively) will match the dimensions of the specified window: **if you wish to preserve the "shape" of the grid you should use the FRAME directive (4.9.3.1)** to define a window whose y- and x-dimensions are in the same proportions as the grid dimensions, as shown in Example 4.4.1c. Titles can be added to these windows using the TITLE and KEYDESCRIPTION options. The SCREEN option controls whether the graphical display is cleared before the histogram is plotted and the ENDACTION option controls whether GenStat pauses at the end of the plot, as described in Section 4.1.

The heights of the contour lines are determined using the NCONTOURS, CONTOURS or INTERVAL parameters. The first possibility is to define the contours explicitly using the CONTOURS parameter. Alternatively, if CONTOURS is unset, INTERVAL can set the required interval between each contour. Or, if both CONTOURS and INTERVAL are unset, NCONTOURS defines the required number of lines. GenStat then partitions the range of data values accordingly to give NCONTOURS evenly-spaced contours (or fewer contours if there are insufficient distinct grid values).

The ANNOTATION option controls how the contours are labelled. The default is to label them by integers (ordinals) if there is a key, and by the actual heights (levels) if there is no key. Contour lines that are very short will not be labelled but their height can be determined from adjacent contours. Each line of the key occupies a space of height 0.02 (in normalized device coordinates; see 4.9.3.1), **and the key window by default has room for a heading and nine contour levels**. If necessary, the size of the window can be redefined using the FRAME directive.

The way in which the contour lines are drawn for each grid is determined by the pen that has been defined by the PENCONTOUR parameter of DCONTOUR; the default is to use pen 1. The relevant aspects of the pen should be set in advance, if required, **using the METHOD, COLOUR, LINESTYLE and THICKNESS parameters of the PEN directive (4.9.8).**

If the PENCONTOUR parameter is not used, the plotting method will be line, so that individual contours are made up of straight line segments. If curves are required, METHOD should be set to monotonic to use the method of Butland (1980), or open (or closed) to use the method of McConalogue (1970). Both these methods produce curves that are fitted to independent sets of interpolated points and **can thus produce contour lines that cross**, particularly if the supplied grid of data is coarse or in a region where the contour height is changing rapidly. If METHOD is set to other values, straight lines will be used to draw the contours.

The PENHIGHLIGHT parameter can specify a pen to use to highlight particular contours. The frequency of the highlighting is then determined by the HIGHLIGHTFREQUENCY parameter; by default every tenth contour is highlighted. This is illustrated in Example 4.4.1b and Figure 4.4.1b, where pen 2 is used to highlight every third contour.

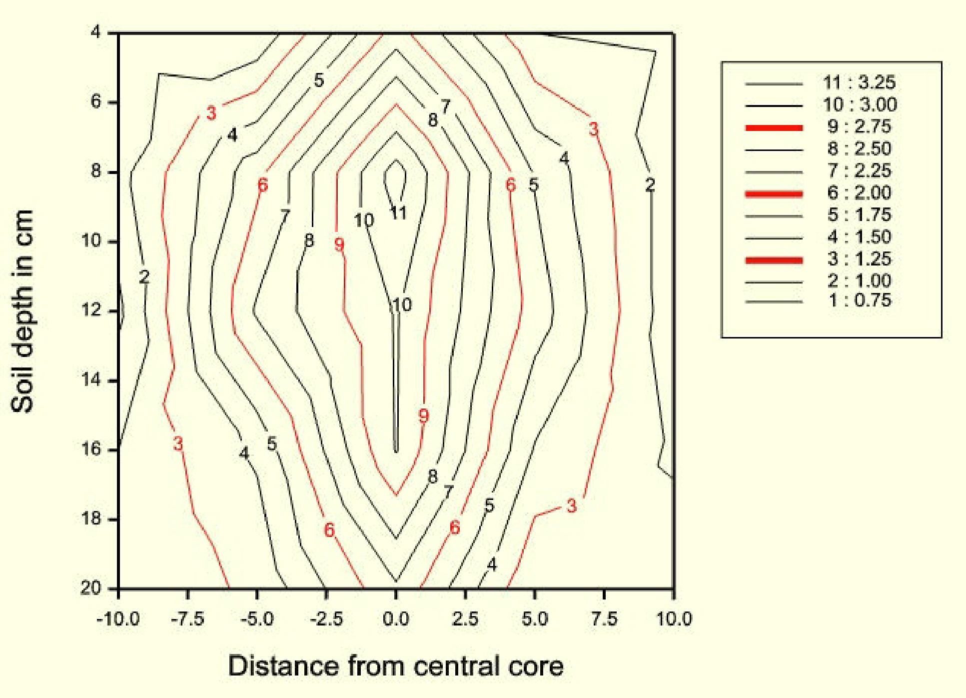


Figure 4.4.1b

Example 4.4.1b

```
16  XAXIS    1; TITLE='Distance from central core'; UPPER=10; LOWER=-10
17  YAXIS    1; TITLE='Soil depth in cm'; LOWER=4; UPPER=20; REVERSE=yes
18  PEN      2; LINESTYLE=1; THICKNESS=4
19  DCONTOUR Core; PENCONTOUR=1; PENFILL=0; PENHIGHLIGHT=2;\
20           HIGHLIGHTFREQUENCY=3; INTERVAL=0.25
```

The PENFILL parameter defines how to shade the areas between the contours**. If PENFILL is set to zero, as in Example 4.4.1b, there is no shading** i.e. the areas between the contours are left in the background colour. If PENFILL is set to a scalar, the shades are defined in increasing intensities of the colour of the specified pen. Alternatively, if PENFILL is set to a variate of length two, the pens are taken to define the shades at the minimum and maximum heights, and the other shades are interpolated between them. Finally, if PENFILL is set to a variate with more than two values, the shading uses the pens in the order in which they are given in the variate (recycling if insufficient pens are defined for the total number of contours). By default, PENFILL=3.

By default, on a colour device, the pens will be defined to use different colours, while on a monochrome device they will use different line styles. In line 18 of Example 4.4.1b, the PEN directive (4.9.8) specifies that pen 2 is to use a solid line style (like pen 1), and the THICKNESS is increased to produce the required highlighting.

By default, the axis bounds are determined from the grid. Normally the lower bound for each axis will be 1.0 and the upper bound will be the number of rows of the grid for the y-axis, and the number of columns for the x-axis. If a matrix is used to specify the grid, its row and column labels can be set to variates whose values will then be used to determine the axis bounds. The XAXIS and YAXIS directives (4.9.4 and 4.9.5) can be used to control how the axes are drawn (see Example 4.4.1b) or, by setting STYLE=none, to suppress them altogether.

In Example 4.4.1c, a matrix of function values is calculated over a regular range of y- and x-values, to produce the contour plot on the left-hand side of Figure 4.3.1c. The function is then recalculated on an irregular grid with the y- and x-values closest where the function is changing most rapidly, and the plot on the right-hand side is produced.

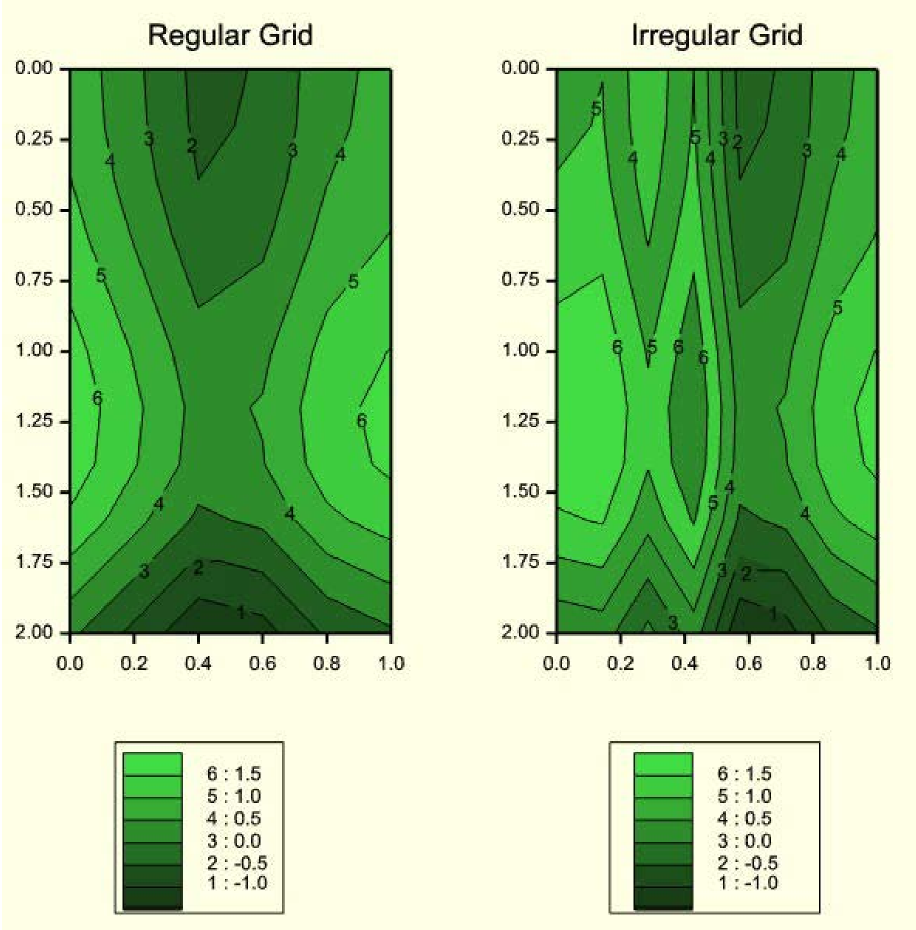


Figure 4.4.1c

Example 4.4.1c

```
VARIATE   Rows,Columns; VALUES=!(0.0,0.2...2.0),!(0.0,0.2...1.0)
CALCULATE Nrows,Ncolumns = NVALUES(Rows,Columns)
MATRIX    [ROWS=Rows; COLUMNS=Columns] X,Y,Function;\
          VALUES=!((#Columns)#Nrows),!(#Ncolumns(#Rows)),*
CALCULATE Function = COS(1/(X+0.1)**2) + SIN(Y**2)
FRAME     1; YLOWER=0.25; YUPPER=1; XLOWER=0; XUPPER=0.5
DCONTOUR  [TITLE='Regular Grid'] Function
VARIATE   Irregular; VALUES=!(0.0,0.1...0.4,0.6,0.8,1.0)
CALCULATE Nirregular = NVALUES(Irregular)
MATRIX    [ROWS=Rows; COLUMNS=Irregular] X,Y,Function;\
          VALUES=!((#Irregular)#Nrows),!(#Nirregular(#Rows)),*
CALCULATE Function = COS(1/(X+0.1)**2) + SIN(Y**2)
FRAME     3,4; YLOWER=0.25,0.0; YUPPER=1.0,0.25; XLOWER=0.5,0.675;\
          XUPPER=1
DCONTOUR  [TITLE='Irregular Grid'; WINDOW=3; KEYWINDOW=4; SCREEN=keep]\
          Function
```

4.4.2 The DSHADE directive

DSHADE directive

Plots a shade diagram of 3-dimensional data.

Options

`TITLE` = *text*	General title; default *
`WINDOW` = *scalar*	Window number for the graph; default 1
`KEYWINDOW` = *scalar*	Window number for the key (0 for no key); default 2

YORIENTATION = *string token*	Y-axis orientation of the plot (reverse, normal); default reve
GRIDMETHOD = *string token*	How to draw a grid around the elements of the matrix (none, present, complete); default pres
PENGRID = scalar	Pen to use for the grid; default -7
SCREEN = *string token*	Whether to clear the screen before plotting or to continue plotting on the old screen (clear, keep); default clea
KEYDESCRIPTION = *text*	Overall description for the key
ENDACTION = *string token*	Action to be taken after completing the plot (continue, pause); default * uses the setting from the last DEVICE statement

Parameters

GRID = *symmetric matrix*, *matrix*, *table* or *pointer to variates*	Data to be plotted
PEN = *scalar* or *variate*	How to draw each shade
LIMITS = *variate*	Boundary values for changes in shade
NGROUPS = *scalar*	Number of groups to form from the data values (i.e. number of different shades)
INTERVAL = *scalar*	Interval between changes in shade
DESCRIPTION = *text*	Annotation for key

DSHADE produces a shaded representation of a rectangular or symmetric matrix using highresolution graphics. Each element of the data matrix is represented by a shaded rectangle indicating the value at that location, using either colour or shading density. This type of display is often used in a cluster analysis to display a similarity matrix, but it is also useful for the graphical display of spatial data.

The data are specified by the GRID parameter, in either a matrix, a symmetric matrix (e.g. of similarities), a 2-way table or a pointer to a set of variates.

The range of data values corresponding to each shade are determined using the NGROUPS, the LIMITS or the INTERVAL parameter. The first possibility is to set LIMITS to a variate defining the boundaries on the data values where the shades change. Alternatively, if LIMITS is unset, NGROUPS can be used to define the required number of shades; GenStat then partitions the range of data values into that number of equal intervals (and shades each interval in a different way). Or, if both NGROUPS and LIMITS are unset, INTERVAL can set the interval between each change in shade. **Finally, if none of these parameters is set, GenStat uses a different shade for each distinct data value. Missing values are ignored, thus leaving blank areas in the plot.**

By default, the shades are drawn using pens 1, 2 onwards, with pen 1 being used for the lowest data values. Alternatively, you can specify the pen or pens explicitly, using the PEN parameter. If PEN is set to a scalar, the shades are defined in increasing intensities of the colour of the specified pen. Alternatively, if PEN is set to a variate of length two, the pens are taken to define the shades of the minimum and maximum data values, and the other shades are interpolated between them. Finally, you can set PEN to a variate with more than two values, and the shades use the pens in the order in which they are given in the variate (recycling if insufficient pens are defined for the total number of shades).

The shades are controlled by the current COLOUR and BRUSH settings of the pens. If the default settings do not produce a suitable display, these attributes should be set by a PEN statement before using DSHADE.

The GRIDMETHOD option specifies whether an outline should be drawn around each element of the matrix. The default setting, present, produces an outline for all values that are present; i.e. it ignores missing values. This is suitable where data have been sampled over an irregularly shaped area. Alternatively, with the complete setting, an outline is drawn around every element. Setting GRIDMETHOD=* stops the grid being drawn, which may be preferable if there are a large number of elements in the input data. The PENGRID option specifies which pen to use to draw the grid. The default is to use pen -7.

The YORIENTATION option controls the orientation of the y-axis. By default this is reversed, so that the data are in the same order as they would take if the data matrix were printed.

The `TITLE`, `WINDOW`, `SCREEN` and `ENDACTION` options are used to specify a title, the plotting window, whether the screen should be cleared first, and whether there should be a pause once the plotting is finished; as in other graphics directives. Similarly, the `KEYWINDOW` and `KEYDESCRIPTION` options and the `DESCRIPTION` parameters allow a key to be defined, if feasible for these plots with the current graphics device.

Example 4.4.2 uses `DSHADE` to display a similarity matrix for 16 types of Italian cars. The resulting graph is in Figure 4.4.2.

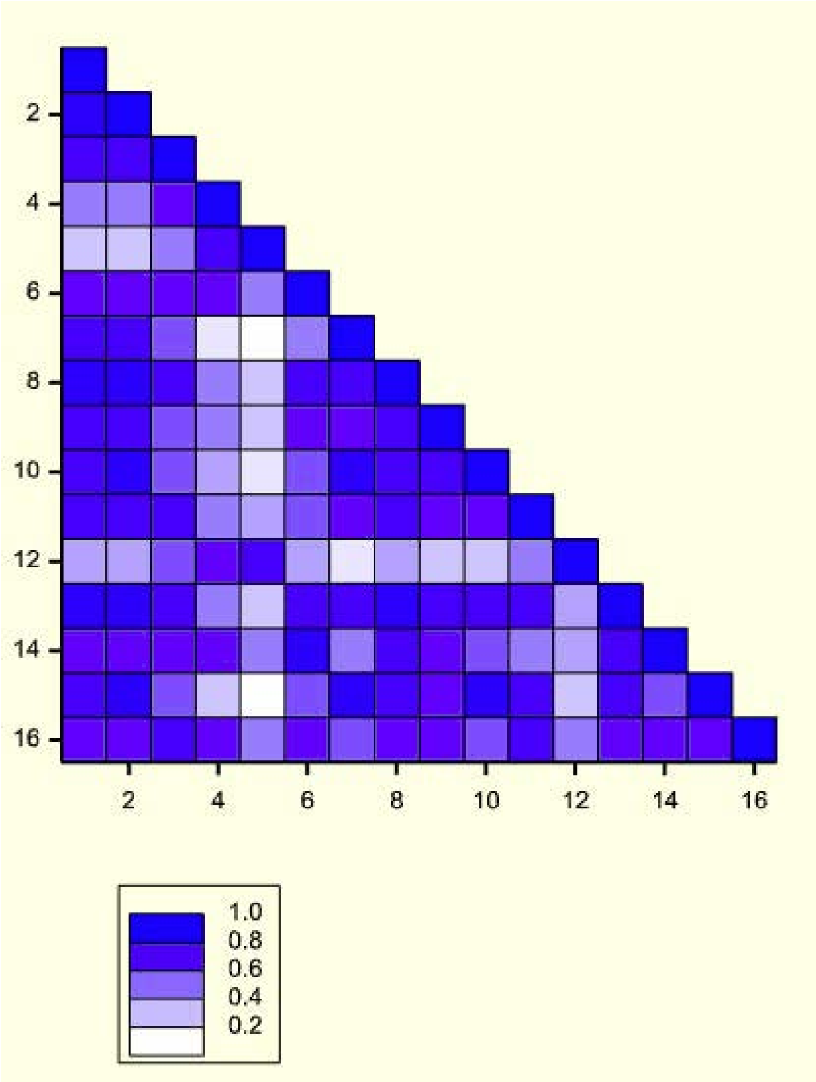


Figure 4.4.2

Example 4.4.2

```
SYMMETRIC [ROWS=!t(Estate,'Arna1.5','Alfa2.5',Mondialqc,\
          Testarossa,Croma,Panda,Regatta,Regattad,Uno,\
          X19,Contach,Delta,Thema,Y10,Spider)] Carsim
READ      Carsim
```

Identifier	Minimum	Mean	Maximum	Values	Missing
Carsim	0.1030	0.7249	1.000	136	0

```
FRAME  1; BOX=omit
PEN    9,10; COLOUR='white','blue'
DSHADE Carsim; INTERVAL=0.1; PEN=!(9,10)
```

4.4.3 The DSURFACE directive

DSURFACE directive
Produces perspective views of a two-way arrays of numbers.

Options

`TITLE` = *text*	General title; default *
`WINDOW` = *scalar*	Window number for the plots; default 1
`KEYWINDOW` = *scalar*	Window number for the key (zero for no key); default 2
`ELEVATION` = *scalar*	The elevation of the viewpoint relative to the surface; default 25 (degrees)
`AZIMUTH` = *scalar*	Rotation about the horizontal plane; the default of 225 degrees ensures that, with a square matrix M, the element M$[1;1] is nearest to the viewpoint
`DISTANCE` = *scalar*	Distance of the viewpoint from the centre of the grid on the base plane; default * gives a distance of 100 times the maximum of the x-range and the y-range
`ZSCALE` = *scalar*	Defines the scaling of the z-axis relative to the horizontal (x-y) axes; default 1
`SCREEN` = *string token*	Whether to clear the screen before plotting or to continue plotting on the old screen (`clear`, `keep`); default `clea`
`KEYDESCRIPTION` = *text*	Overall description for the key; default *
`ENDACTION` = *string token*	Action to be taken after completing the plot (`continue`, `pause`); default * uses the setting from the last `DEVICE` statement

Parameters

`GRID` = *identifier*	Pointer (of variates representing the columns of a data matrix), matrix or two-way table specifying values on a rectangular grid
`PEN` = *scalar*	Pen number to be used for the plot; default 1
`PENFILL` = *scalar* or *variate*	Pen number(s) defining how to fill the areas between contours (0 or * leaves the areas in the background colour); default 3
`PENMESH` = *scalar*	Pen number to use to draw the mesh (omitted if set to 0 or *); default 1
`PENSIDE` = *scalar*	Pen number to use to shade the sides of the surface (omitted if set to 0 or *); default *
`NCONTOURS` = *scalar*	Number of contours; default 10
`CONTOURS` = *variate*	Positions of contours
`INTERVAL` = *scalar*	Interval between contours
`DESCRIPTION` = *text*	Annotation for key

The `DSURFACE` directive produces a perspective (or *conical*) projection of a surface, showing the view from a particular viewpoint. The position of this viewpoint is specified in polar coordinates, using the options `ELEVATION`, `DISTANCE` and `AZIMUTH`. These define the angle ofelevation, in degrees, above the base plane of the surface, distance from the centre of this plane, and angular position relative to the vertical z-axis, respectively. This is illustrated in Figure 4.4.3a. The default settings of `ELEVATION`, `DISTANCE` and `AZIMUTH` have

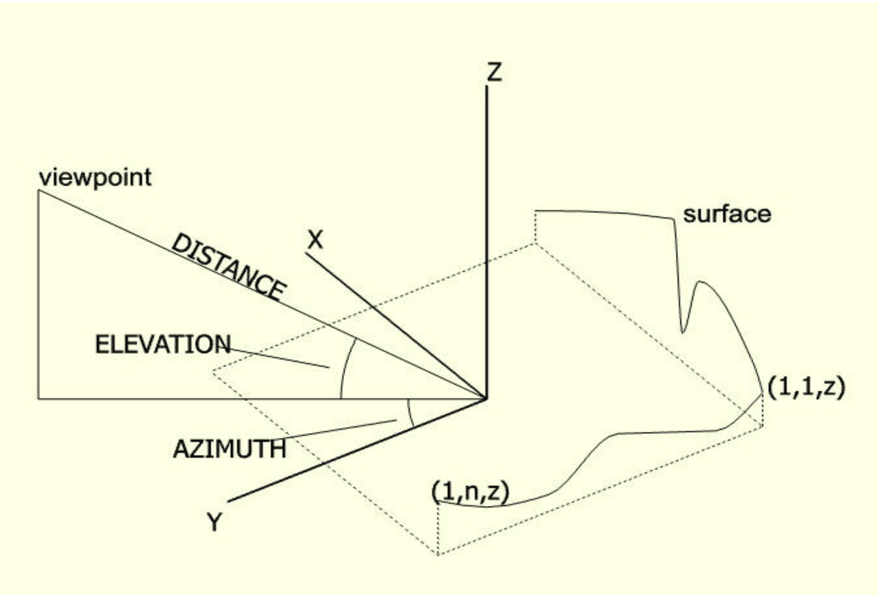


Figure 4.4.3a

been chosen to produce a reasonable display of most surfaces; but if, for example, some parts of the surface are obscured by high points they can be modified to obtain a better view. Altering the value of AZIMUTH will, in effect, rotate the surface in the horizontal plane about a vertical axis drawn through the centre of the grid; the default value of 225 degrees ensures that the element in the first row and column of the grid is at the corner nearest the viewpoint. Small values of DISTANCE produce a perspective view; larger values, like the default of 100 times the maximum of the x-range and the y-range, effectively put the viewpoint at infinity to produce an "orthographic parallel projection".

The ZSCALE option specifies a scaling factor for the z-axis (or vertical axis) of the plotted surface. Generally values between 0.5 and 2.0 are most successful; large values result in a flatter surface, while smaller values produce a steep surface, accentuating changes in the data.

The TITLE, WINDOW, SCREEN and ENDACTION options are used to specify a title, the plotting window, whether the screen should be cleared first, and whether there should be a pause once the plotting is finished; as in other graphics directives. Similarly, the KEYWINDOW and KEYDESCRIPTION options and the DESCRIPTION parameters allow a key to be defined, if feasible for these plots with the current graphics device.

The PEN parameter specifies the pen to be used to plot the surface (by default, pen 1). **The PEN directive can be used to modify the colour and the thickness of the pen**, but the other attributes of the pen are ignored.

The NCONTOURS, CONTOURS and INTERVAL parameters control the contours drawn on the surface, if these are available on the current graphics device. The first possibility is to define the contours explicitly using the CONTOURS parameter. Alternatively, if CONTOURS is unset, INTERVAL can set the required interval between each contour. Or, if both CONTOURS and INTERVAL are unset, NCONTOURS defines the required number of lines. GenStat then partitions the range of data values accordingly to give NCONTOURS evenly-spaced contours (or fewer contours if there are insufficient distinct grid values).

The PENFILL parameter defines how to shade the areas between the contours. If this is set to a scalar, the shades are defined in increasing intensities of the colour of the specified pen. Alternatively, if PENFILL is set to a variate of length two, the pens are taken to define the shades at the minimum and maximum heights, and the other shades are interpolated between them. Finally, you can set PENFILL to a variate with more than two values, and the shading uses the pens in the order in which they are given in the variate (recycling if insufficient pens are defined for the total number of contours). The default is to use pen 3. However, **if you set PENFILL to 0 or to a missing value, there will be no shading** (that is, the areas between the contours will be in the background colour).

The PENMESH parameter specifies a pen to be used to draw a mesh on the surface. This consists of lines marking the points of the surface that lie above a rectangular grid on the xy plane. By default pen 1 is used, but if you set PENMESH to 0 or to a missing value the mesh is omitted. The PENSIDE parameter defines the pen to use to shade the sides of the surface. There is no shading if this is set to 0 or a missing value, which is the default. The CFILL setting of the pen (see the PEN directive) specifies which colour is used.

Simple axes are drawn to indicate the directions in which x and y increase. The TITLE parameter of the XAXIS and YAXIS directives (4.9.4 and 4.9.5) can be used to add further annotation, as shown in lines 12 and 13 of Example 4.4.3 which produced the plots in Figures 4.4.3b and 4.4.3c. You can also use the UPPER parameter of ZAXIS (4.9.6) to truncate the grid, and the LOWER parameter to set the value for the base of the surface (line 16).

Example 4.4.3

```
 2  VARIATE   [VALUES=0.0,0.05...1.0] Values
 3  CALCULATE Nvalues = NVALUES(Values)
 4  MATRIX    [ROWS=Nvalues; COLUMNS=Nvalues] X,Y,Grid;\
 5            VALUES=!((#Values)#Nvalues),!(#Nvalues(#Values)),*
 6  CALCULATE Fx = EXP(-0.5*((X-0.3)/0.07)**2) \
 7                + 0.5*EXP(-0.5*((X-0.7)/0.12)**2)
 8  &         Fy = EXP(-0.5*((Y-0.3)/0.07)**2) \
 9                + 0.5*EXP(-0.5*((Y-0.7)/0.12)**2)
10  &         [PRINT=summary] Grid = Fx*Fy+0.1
```

Identifier	Minimum	Mean	Maximum	Values	Missing	
Grid	0.1000	0.1960	1.104	441	0	Skew

```
PEN        11; SIZE=4; COLOUR='black'
XAXIS      3; TITLE='The X axis'; PENTITLE=11
YAXIS      3; TITLE='The Y axis'; PENTITLE=11
ZAXIS      3; PENTITLE=11
DSURFACE   [WINDOW=3; KEY=0; TITLE='Default option settings'] Grid
ZAXIS      3; LOWER=0; UPPER=0.6
DSURFACE   [WINDOW=3; KEY=0; TITLE='Changes of viewpoint and z-axis';\
           AZIMUTH=120] Grid
```

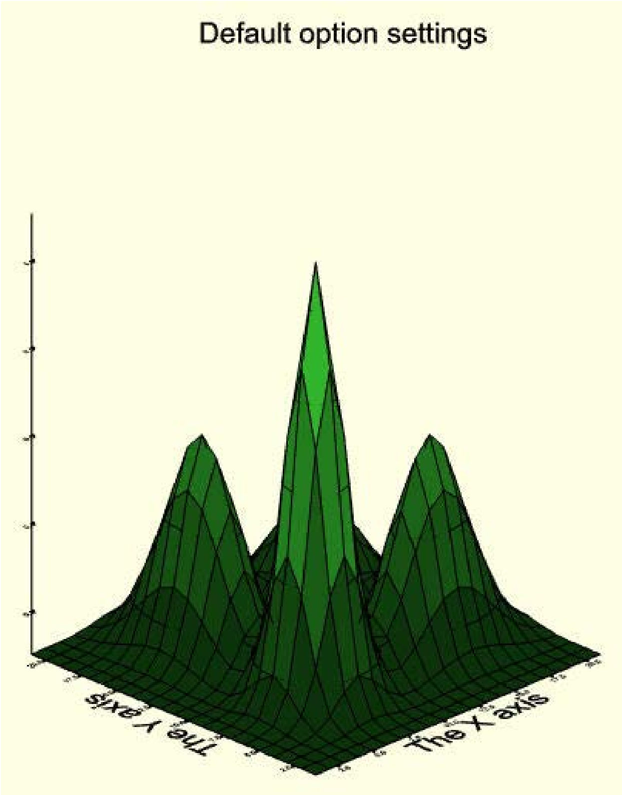


Figure 4.4.3b

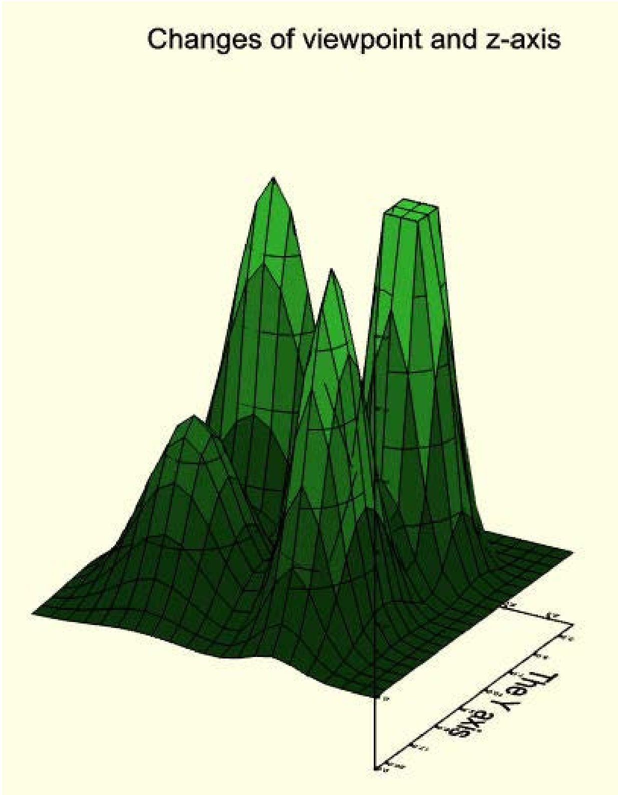


Figure 4.4.3c

4.4.4 The D3HISTOGRAM directive

D3HISTOGRAM directive

Plots three-dimensional histograms.

Options

TITLE = *text*	General title; default *
WINDOW = *scalar*	Window number for the plots; default 1
KEYWINDOW = *scalar*	Window number for the key (zero for no key); default 2
ELEVATION = *scalar*	The elevation of the viewpoint relative to the surface; default 25 (degrees)
AZIMUTH = *scalar*	Rotation about the horizontal plane; the default of 225 degrees ensures that, with a square matrix M, the element M$[1;1] is nearest to the viewpoint
DISTANCE = *scalar*	Distance of the viewpoint from the centre of the grid on the base plane; default * gives a distance of 100 times the maximum of the x-range and the y-range
SCREEN = *string token*	Whether to clear the screen before plotting or to continue plotting on the old screen (`clear`, `keep`); default `clea`
KEYDESCRIPTION = *text*	Overall description for the key; default *

ENDACTION = *string token*	Action to be taken after completing the plot (continue, pause); default * uses the setting from the last DEVICE statement

Parameters

GRID = *identifier*	Pointer (of variates representing the columns of a data matrix), matrix or two-way table specifying values on a regular grid
PEN = *scalar*	Pen number to be used for the plot; default 3
DESCRIPTION = *texts*	Annotation for key

The preceding subsection described how the DSURFACE directive can be used to produce a perspective view of a surface. D3HISTOGRAM provides an alternative way of displaying such data, which may be more appropriate for example if the grid contains counts.

The position of the point from which the histogram is viewed is specified in polar coordinates, using the options ELEVATION, DISTANCE and AZIMUTH as with DSURFACE (4.4.3). The TITLE, WINDOW, SCREEN and ENDACTION options operate in the usual way, to specify a title, the plotting window, whether the screen should be cleared first, and whether there should be a pause once the plotting is finished. Similarly, the KEYWINDOW and KEYDESCRIPTION options and the DESCRIPTION parameters allow a key to be defined, if feasible for these plots with the current graphics device. The PEN parameter specifies the pen to be used to plot the histogram (by default, pen 3). **The PEN directive can be used to modify the colour and the thickness of the pen, but the other attributes of the pen are ignored.**

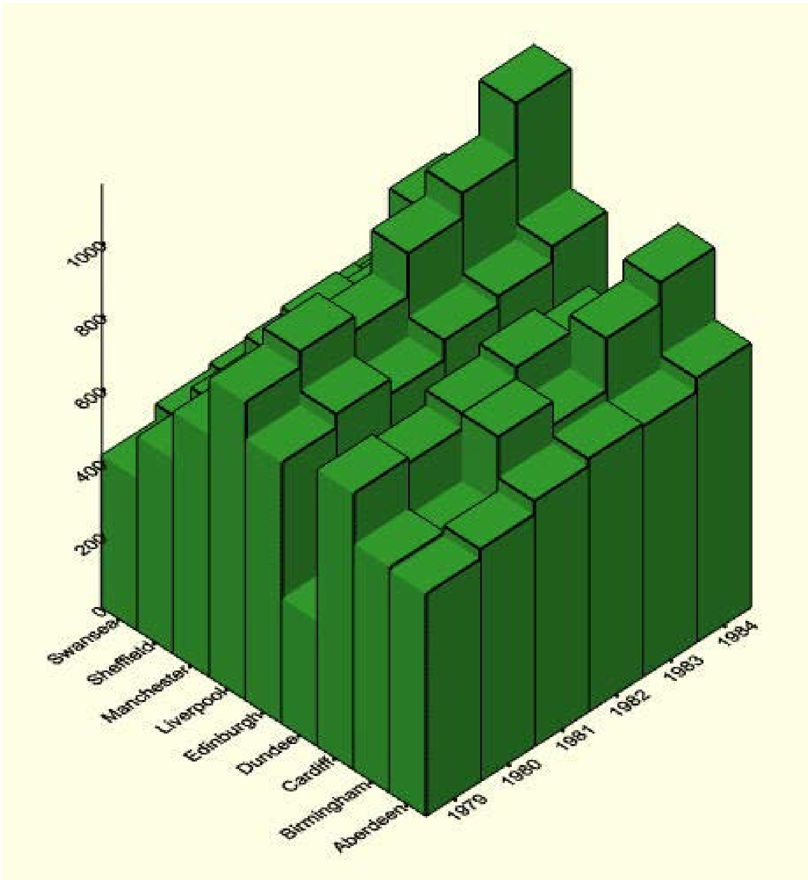


Figure 4.4.4

Example 4.4.4 illustrates the use of D3HISTOGRAM by displaying the table Sales formed in the example of this directive; the resulting graph is in Figure 4.4.4. The AZIMUTH and ELEVATION options are used to obtain a clearer view of the surface, and LOWER option of the ZAXIS directive (4.9.6) is used to set the minimum z-value to zero. Note that when the grid is not square, as in this example, the y- and x-axes are scaled appropriately. This is also the case when using DSURFACE.

The axis labelling is derived from the grid, using the classifying factors if it is a table or the row and column labels if it is a matrix. Alternative labels can be supplied using the LABELS parameters of the XAXIS and YAXIS directives (4.9.4 and 4.9.5). If axis labels are not available, either from the grid or from an XAXIS or YAXIS statement, plain axes will be drawn in the style used by DSURFACE; these can be labelled using the TITLE parameter of XAXIS and YAXIS.

Example 4.4.4

```
" Expand axis labels."
PEN   11; SIZE=2; COLOUR='black'
XAXIS 3; PENLABELS=11
```

```
YAXIS 3; PENLABELS=11
ZAXIS 3; LOWER=0; PENLABELS=11
D3HISTOGRAM [WINDOW=3; KEY=0; ELEVATION=40] Sales
```

4.5 Displaying pictures

4.5.1 The DBITMAP directive

DBITMAP directive
Plots a bit map of RGB colours.

Options

TITLE = *text*	General title; default *
WINDOW = *scalar*	Window number for the graph; default 1
YORIENTATION = *string token*	Y-axis orientation of the plot (reverse, normal); default reve
GRIDMETHOD = *string token*	How to draw a grid around the elements of the matrix (present, complete); default * i.e. none
PENGRID = scalar	Pen to use for the grid; default -7
SCREEN = *string token*	Whether to clear the screen before plotting or to continue plotting on the old screen (clear, keep); default clea
ENDACTION = *string token*	Action to be taken after completing the plot (continue, pause); default * uses the setting from the last DEVICE statement

Parameters

BITMAP = *symmetric matrix*, *matrix*, *table*, *pointer to variates* or *variate*	Data to be plotted
ROWS = *variate*	Row indexes for a BITMAP variate
COLUMNS = *variate*	Column indexes for a BITMAP variate

DBITMAP displays a picture, represented as a 2-dimensional bit map of RGB colours (see 4.8.9). The data are specified by the BITMAP parameter. Data values in a regular two-way grid can be specified by supplying their RGB colours in either a matrix, a symmetric matrix, a 2-way table or a pointer to a set of variates. Alternatively, you can specify irregular data by setting BITMAP to a variate of colours, and the ROWS and COLUMNS parameters to variates defining their row and column indexes. **In GenStat *for Windows* you can form the bit map from an image file (JPG, GIF, TIF or PNG) using the IMPORT procedure and the matrix setting of the RGBMETHOD option, as shown in Example 4.5.1 and Figure 4.5.1.**

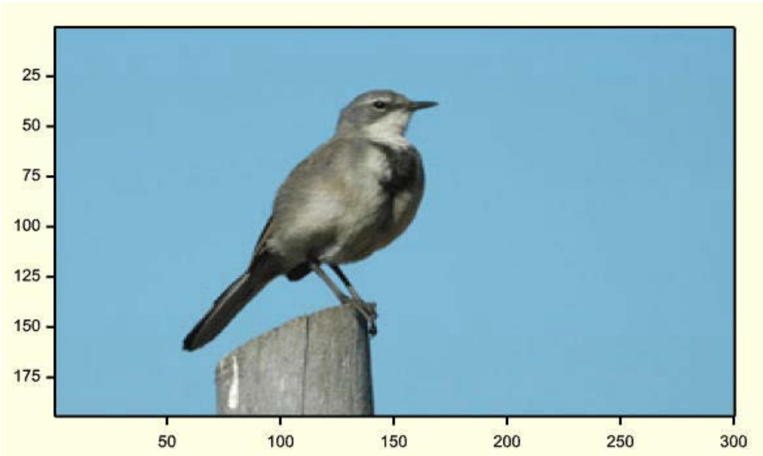


Figure 4.5.1

Example 4.5.1

```
IMPORT    [RGBMETHOD=matrix] file; COLUMNS='RGB'
" resize the window to match the dimensions of the bit map "
CALCULATE Nr = NROWS(RGB)
&         Nc = NCOLUMNS(RGB)
IF Nr < Nc
```

```
  FRAME   3; YUPPER=Nr/Nc
ELSE
  FRAME   3; XUPPER=Nc/Nr
ENDIF
DBITMAP   [WINDOW=3] RGB
```

The GRIDMETHOD option allows you to draw an outline around each element of the plot. The present setting produces an outline for all values that are present; i.e. it ignores missing values. This is suitable where data have been sampled over an irregularly shaped area. Alternatively, with the complete setting, an outline is drawn around every element. By default, no grid is drawn. The PENGRID option specifies which pen to use to draw the grid. The default is to use pen -7.

The YORIENTATION option controls the orientation of the y-axis. By default this is reversed, so that the data are in the same order as they would take if the data matrix were printed.

The TITLE, WINDOW, SCREEN and ENDACTION options are used to specify a title, the plotting window, whether the screen should be cleared first, and whether there should be a pause once the plotting is finished; as in other graphics directives.

4.6 Pie charts

4.6.1 The DPIE directive

DPIE directive

Draws a pie chart on a plotter or graphics monitor.

Options

TITLE = *text*	General title; default *
WINDOW = *scalar*	Window number for the pie chart; default 1
KEYWINDOW = *scalar*	Window number for the key (zero for no key); default 2
ANNOTATION = *string token*	Whether to annotate the slices by their percentages (percentages); default perc
OUTLINE = *string token*	Where to draw outlines (slices, perimeter); default slices
PENOUTLINE = *scalar*	Pen to use for the outlines; default -10
SCREEN = *string token*	Whether to clear the screen before plotting or to continue plotting on the old screen (clear, keep); default clea
KEYDESCRIPTION = *text*	Overall description for the key
ENDACTION = *string token*	Action to be taken after completing the plot (continue, pause); default * uses the setting from the last DEVICE statement

Parameters

SLICE = *scalars*	Amounts in each of the slices (or categories)
PEN = *scalars*	Pen number for each slice; default * uses pens 1, 2, and so on for the successive slices
DESCRIPTION = *texts*	Description of each slice

A pie chart is formed by taking the values of the scalars in the SLICE parameter, in order, and representing them by segments of a circle **starting at "three o'clock" and working in an anticlockwise direction**. The angle subtended by each segment (and thus the area of the segment) is proportional to the value of the corresponding scalar. The values may be raw data or can be expressed as percentages (by ensuring they total to 100).

The brush style and colour used for each segment can be controlled using the PEN parameter. By default, pen 1 is used for the first segment, pen 2 for the second segment, and so on. The default attributes of the pens are device specific, so that on a colour display the segments will be solid-filled using different colours, and on a monochrome device different hatching styles will be used. These can be modified using the PEN directive, as described in 4.9.8.

Lines 3 and 4 of Example 4.6.1 plot a pie chart with four slices, as shown in the top half of Figure 4.6.1.

Example 4.6.1

```
2  FRAME 1,2; YLOWER=0.5,0.0; YUPPER=1.0,0.5; XLOWER=0.0; XUPPER=1.0
3  DPIE  [WINDOW=1; KEYWINDOW=0] 24.7,98.8,74.1,49.4; \
4        DESCRIPTION='Administration','Sales','Marketing','Overheads'
5  DPIE  [WINDOW=2; KEYWINDOW=0; SCREEN=keep;\
6        ANNOTATION=percentage] 10,40,30,-20; \
7        DESCRIPTION='Administration','Sales','Marketing','Overheads'
```

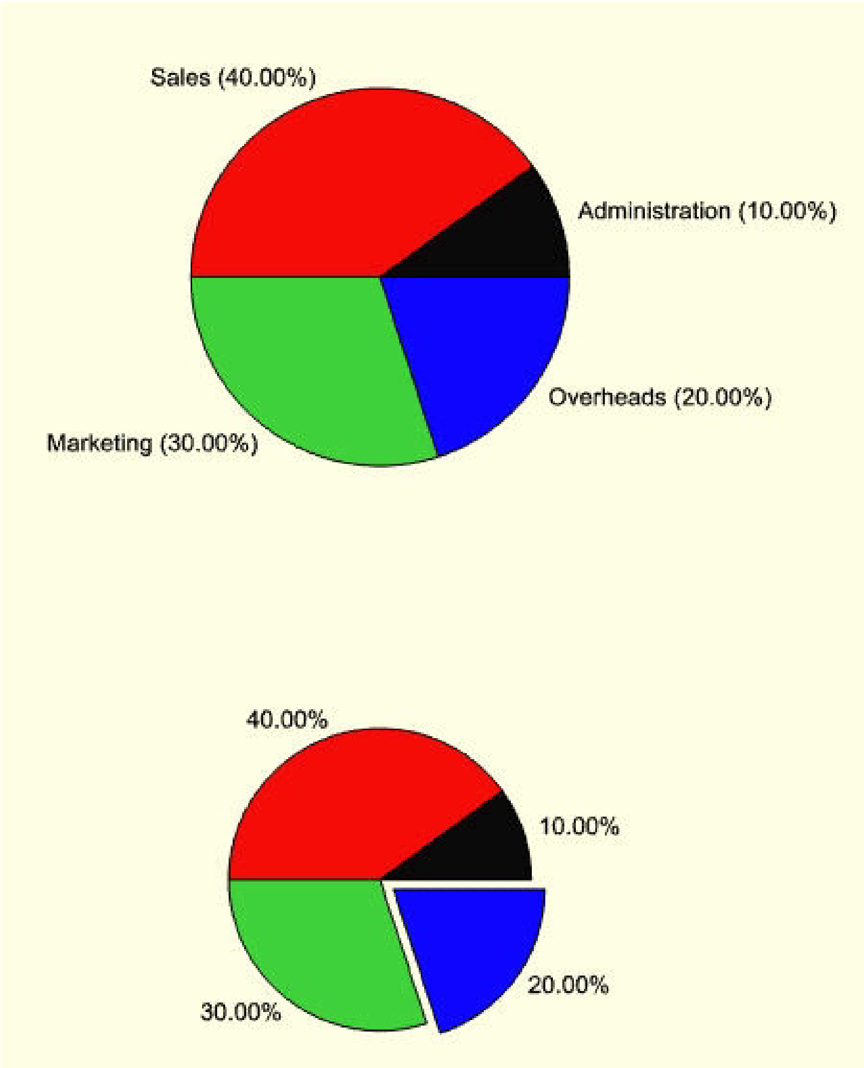


Figure 4.6.1

Individual segments can be displaced outwards from the centre, to obtain an "exploded" pie chart, as in the bottom half of Figure 4.6.1. The chosen segments are indicated by setting the corresponding scalars in the SLICE parameter list to negative values (see line 6 of Example 4.6.1).

The WINDOW and KEYWINDOW options specify the windows in which the pie chart and key are to be displayed. The shape of the pie chart is determined by the dimensions of the window; if it is not square the resulting pie chart will be elliptical.

Titles can be added using the TITLE and KEYDESCRIPTION options. The key produced for the pie chart is similar to that produced by the DHISTOGRAM directive. A shaded block is drawn for each segment, followed by the identifier name or the piece of text specified by the DESCRIPTION parameter. The key usually also gives the percentage contained by each slice, but you can suppress this by setting option ANNOTATION=*.

The OUTLINE option controls whether lines are drawn around the slices or around the perimeter of the pie chart. These are drawn using the pen specified by the PENOUTLINE option (default -10). You can suppress all the outlines by setting OUTLINE=* .

The SCREEN option controls whether the graphical display is cleared before the histogram is plotted and the ENDACTION option controls whether GenStat pauses at the end of the plot, as described at the start of this section.

4.7 Adding annotation to a graph

4.7.1 The DTEXT procedure

DTEXT procedure
Adds text to a graph (S.A. Harding).

Option

`WINDOW` = *scalar* — Window number of the graph; default 1

Parameters

Y = *variates* or *scalars* — Vertical coordinates
X = *variates* or *scalars* — Horizontal coordinates
TEXT = *texts* — Text to plot
PEN = *scalars*, *variates* or *factors* — Pens to use; default 1

The DTEXT procedure provides a convenient way of adding textual annotation or description to a plot. The text to plot is specified by the TEXT parameter. This can be either a single string, or a GenStat text structure containing several lines of text. The Y and X parameters specify where to plot the text, with scalars for a single string or line, or with variates for several lines. The PEN parameter specifies the pen or pens to use (default 1), and the WINDOW option specifies the window where the plot is taking place (default 1).

Example 4.7.1 uses DTEXT to insert the annotation onto the Venn diagram in Figure 4.7.1.

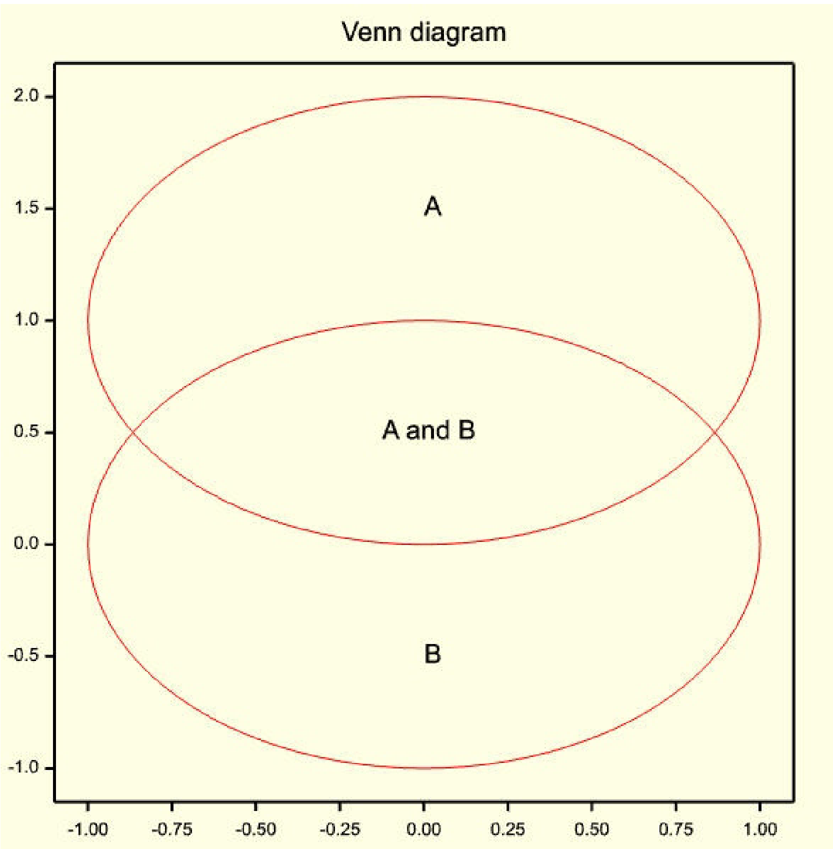


Figure 4.7.1

Example 4.7.1

```
VARIATE   [VALUES=0...100] theta
CALCULATE theta = 2 * C('pi') * theta / 100
&         X  = COS(theta)
&         Y1 = SIN(theta)
&         Y2 = Y1 + 1
PEN       2; METHOD=closed; SYMBOL=0; JOIN=given
DGRAPH    [TITLE='Venn diagram'; WINDOW=3; KEY=0] Y1,Y2; X; PEN=2
PEN       1; SIZE=1.5
DTEXT     [WINDOW=3] Y=1.5,0.5,-0.5; X=0,-0.125,0;\
          TEXT='A','A and B','B'; PEN=1
```

4.8 Multiple high-resolution plots

Many GenStat graphics commands have a SCREEN option, which can be set to keep to enable you to add new information to the current display. The output from each command is drawn in one or more graphics *windows*. There are 256 windows (see FRAME, 4.9.3.1). They are independent of one another, and most graphics devices allow you to display them simultaneously on the same graphics screen. On most devices you can also have windows that overlap or contain others. So you can plot to a sequence of windows (keeping the current display), and build up a multiple display with different graphs in adjacent windows. Several GenStat procedures use this facility: for example trellis plots are described in 4.8.3, and scatter-plot matrices in 4.8.4.

Alternatively, you may be able to plot new information in an existing window, and build up a complicated picture in several stages. However, there are limitations on what a single window can contain: you can use DGRAPH (4.2.1) any number of times, **but you can use no more than one other command**, which may be either DHISTOGRAM (4.3.1), DCONTOUR (4.4.1) or DSHADE (4.4.2). This approach is used in graphics procedures, like BOXPLOT or DDENDROGRAM, and is illustrated in Example 4.8 and Figure 4.8 where a graph and a histogram are plotted in window 3.

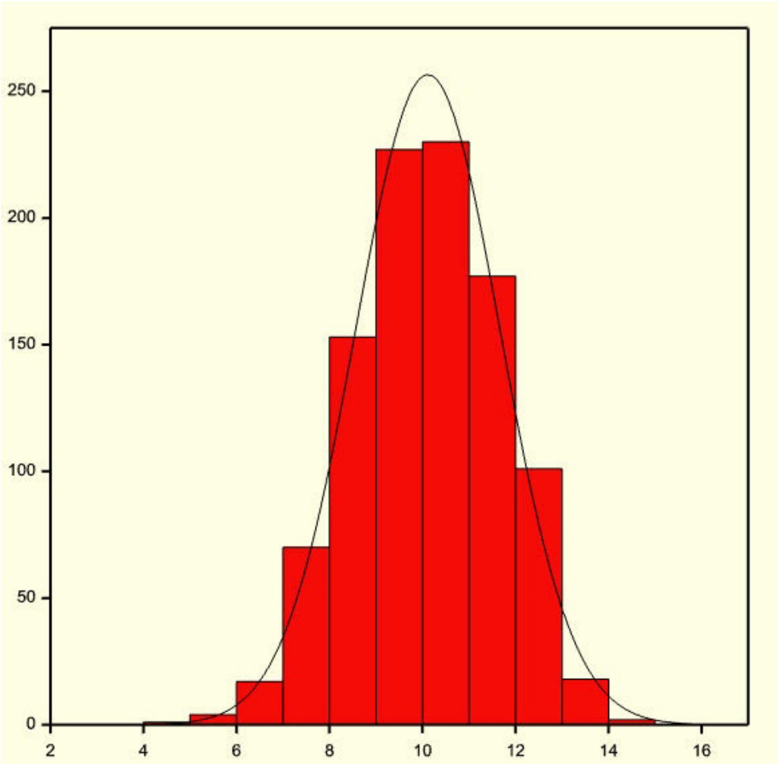


Figure 4.8

Example 4.8

```
  2  READ Data

       Identifier    Minimum       Mean   Maximum     Values    Missing
             Data      4.390      10.11     14.55       1000          0

104  VARIATE    [VALUES=2.0,2.1...17.0] X
105  CALCULATE  Mu = MEAN(Data)
106  &          Sigma = SQRT(VARIANCE(Data))
107  &          Y = 1 / (SQRT(2*C('PI')) * Sigma) * EXP(-0.5*((X-Mu)/Sigma)**2)
108  &          Y = Y * NVALUES(Data)
119  PEN        1; METHOD=monotonic; SYMBOL=0
110  XAXIS      3; LOWER=2.0; UPPER=17.0
111  YAXIS      3; LOWER=0.0; UPPER=275
112  DGRAPH     [WINDOW=3; KEYWINDOW=0] Y; X
113  VARIATE    [VALUES=3...16] Limits
114  DHISTOGRAM [WINDOW=3; KEYWINDOW=0; LIMITS=Limits; SCREEN=keep] Data
```

4.8.1 The DCLEAR directive: clearing the graphics screen

DCLEAR directive

Clears a graphics screen.

Options

DEVICE = *scalar*	Device whose screen is to be cleared; default is to clear the screen of the current graphics device
ENDACTION = *string token*	Action to be taken after clearing the screen (continue, pause); default * uses the setting from the last DEVICE statement

No parameters

When generating displays using a sequence of graphics commands, it may be convenient to clear the screen at the outset. Then the subsequent commands can all have option SCREEN=keep, which will simplify the programming particularly if they are in a FOR loop. Thus DCLEAR allows you to clear the screen of a graphics device so that the next plot produced on this device by any of the high-resolution commands will be drawn onto an empty screen. **All information about the current display, for example axis mappings, is also cleared from memory.** The DEVICE option indicates the device to be cleared; by default this is the current graphics device (as set by the DEVICE directive). The ENDACTION option controls what happens after clearing the screen. The default action is the setting specified by the most recent DEVICE statement.

4.8.2 The DSTART and DFINISH directives: sequences of high-resolution plots

---

DSTART directive

Starts a sequence of related high-resolution plots.

Option

TITLE = *text*	Overall title for the plots

DFINISH directive

Ends a sequence of related high-resolution plots.

No options or parameters

The most efficient way of generating a composite display is to define an explicit sequence of plots. The start of the sequence is indicated by a DSTART command, which can also supply an overall title for the plots. **This is plotted using pen -12.**

During the sequence the information from each graphics command is accumulated until GenStat finds a DFINISH command. GenStat then clears the screen and generates the display. **This improves efficiency, as no plotting takes place until the display is complete**. It also simplifies programming as the SCREEN option is irrelevant; any settings of the SCREEN option in the plotting directives during the sequence are ignored.

4.8.3 The TRELLIS procedure: Trellis plots

TRELLIS procedure

Does a trellis plot (S.J. Welham & S.A. Harding).

Options

GROUPS = *factors* or *variate*	Factors or variate defining the classification for the plots
GMETHOD = *string token*	Determines the method used to partition the range when GROUPS is set to a variate (equalspacing, quantiles, distinct, limits); default equal

NGROUPS = *scalar*	Determines the number of plots to be formed when GROUPS is set to a variate and GMETHOD is set to quantiles or equalspacing
LIMITS = *variate*	Limits to use to form groups from a GROUPS variate when GMETHOD=limits
OVERLAP = *scalar*	Proportion by which a GROUPS variate should overlap between plots (scalar in range 0 - 0.5); default 0
OMITEMPTY = *string token*	Whether to omit all empty plots from the array (all), or omit levels of a GROUPS factor where all plots are empty (levels), or keep all plots in the array (none); default level
PENGROUPS = *factors*	Defines factor combinations to be plotted in different colours, note that the number of colours available may differ between devices
NROWS = *scalar*	Specifies number of rows of plots to appear on one page; default determined automatically from GROUPS
NCOLUMNS = *scalar*	Specifies number of columns of plots to appear on one page; default determined automatically from GROUPS
TITLE = *text*	Supplies a title for the plot
FIRSTPICTURE = *string token*	Whether to put the first picture at bottom or top left of the grid (bottomleft, topleft); default topl
TMETHOD = *string token*	Whether to give plot titles as factor names with labels or just labels (names, labels); default names
YTITLE = *text*	Supplies an overall y-axis title
XTITLE = *text*	Supplies an overall x-axis title
YMARGIN = *scalar*	Relative size of margins for the y-axis labels on individual plots; default 0.04
XMARGIN = *scalar*	Relative size of margins for the x-axis labels on individual plots; default 0.04
TMARGIN = *scalar*	Relative size of margin for titles of individual plots; default 0.04
PENSIZE = *scalar*	Proportionate adjustment to the pen size for individual plot titles and axis labels; default 1
USEPENS = *string token*	Whether to use current pen definitions in the procedure (no, yes); default no
USEAXES = *string token*	Which aspects of the current axis definitions of window 1 to use (none, limits, style, marks, mpositions, transform); default none
NRMAX = *scalar*	Maximum number of rows on page; default 8
NCMAX = *scalar*	Maximum number of columns on page; default 8
KEYHEIGHT = *scalar*	Space in y-direction to use for key (0 to suppress key); default * i.e. determined automatically

Parameters

Y = *variates*	Y-values of the data to be plotted
X = *variates* or *factors*	X-values of the data to be plotted
METHOD = *string tokens*	Type of plot (point, line, mean, median, histogram, boxplot, spline); default poin
DESCRIPTION = *texts*	Annotation for key

TRELLIS plots one or more y-variates for each level generated by the GROUPS option, and arranges these plots in a grid (or trellis) arrangement on the page.

The data to be plotted are specified using the Y parameter. If more than one variate is specified, these will all be displayed on the same plots. **This means that e.g. data points can be plotted with means**. The type and method of plotting (points, lines, mean values, medians, histograms, boxplots or splines) is specified using the METHOD parameter. The default is METHOD=point. For methods point, line, mean, median and spline, a graph is produced of y-variates against x-variates, which are specified using the X parameter. When METHOD is set to mean or median, a line is drawn to join the mean or median data values at each value of the x-variate for each level of PENGROUP. In any of these cases, if PENGROUPS is set to one or more factors, a different pen will be used for each of the levels of the combined factors.

When METHOD=histogram, a histogram of the data values is drawn in each plot. In this case, options NGROUPS and LIMITS can be used to specify the number of groups in the histogram or the group limits, respectively. If more than one y-variate is specified, parallel histograms will be drawn for the variates. The PENGROUPS option is ignored when METHOD=histogram.

When METHOD=boxplot, a box-plot of the data values is drawn in each plot. If the PENGROUP option is set, parallel box plots will be drawn for each level of PENGROUPS within each plot. If multiple y-variates are set, parallel box plots will be drawn for each y-variate within each plot. **Note that you cannot specify parallel box plots by simultaneously using multiple y-variates and the PENGROUPS option**.

Other options control various aspects of the plot layout. The division of the data into separate plots is determined by the setting of the GROUPS option. This can be set to one or more factors, indicating that a separate plot should be drawn for each combinations of the factor levels. The OMITEMPTY option controls what happens if there are no data for some combinations. The default setting levels omits complete levels of any factor for which there are no data points, while the setting all omits all empty plots, i.e. plots where there are no data points. OMITEMPTY=none displays all plots regardless of whether or not they contain any data points.

If the GROUPS option is set to a variate, the plots will show the values of the data for different intervals in the range of the GROUPS variate. The GMETHOD, NGROUPS, LIMITS and OVERLAP options determine how many plots are displayed, and which data points they contain. The default option of GMETHOD is equalspacing. The distinct setting of GMETHOD converts the variate into a factor with a level (and thus a plot) for each distinct value of the variate. With equalspacing, the groups are defined by dividing the range of the GROUPS variate into the required number of intervals of equal length; while with quantiles, the intervals are defined so that each has an equal number of points, according to the ordering of the GROUPS variate. When GMETHOD is set to equalspacing or to quantiles, the number of groups to form can be specified by the NGROUPS option; if NGROUPS is not set, TRELLIS sets the number to the square root of the number of data values, or to the number of distinct values if this is smaller. Finally, when GMETHOD=limits, the LIMITS option specifies boundaries between the intervals; the first group then contains all data points with values of the GROUPS variate less than the first limit, the second group has all values greater than or equal to the first limit but less than the second limit, and so on.

The OVERLAP option allows the intervals of the GROUPS to overlap. The default overlap is 0, so there is no overlap between plots. If OVERLAP is set to 0.1, then 10% of the points (for PARTITION=quantiles) or 10% of the range (for PARTITION=equalspacing) will be in common between neighbouring plots. OVERLAP can be set anywhere in the range 0 (for no overlap) to 0.5.

The default layout on the page can be changed by using NROWS and NCOLUMNS to specify the number of rows of plots on the page and the number of columns of plots across the page respectively. By default the layout is arranged so that the area of the page used for plotting is maximized, with up to 64 plots per page. Options NRMAX and NCMAX can be used to override the default maximum numbers of rows and columns of plots, so that more can be produced on a page; the default value of both options is 8.

An overall title can be put at the head of each page using the TITLE option, and overall titles for the y- and x- axes can be specified using the YTITLE and XTITLE options respectively. By default the plots start at the top left of the page, but you can set option FIRSTPICTURE=bottomleft to start at the bottom left. When GROUPS is set to one or more factors, the plot titles are constructed by default with the factor name and label/level, but this can be restricted to just the label/level by setting option TMETHOD=label.

The margins and pen size are set to give a reasonable picture on the Windows PC implementation, but can be adjusted using options YMARGIN (space for y-axis labels), XMARGIN (space for x-axis labels), TMARGIN (space for plot titles) and PENSIZE (pen size for axis markings and plot titles).

By default the pen and axes attributes are determined automatically within the procedure. Some predefined attributes can be used, as indicated by the USEPENS and USEAXES options. Setting USEPENS to yes, requests all current pen definitions (for pens 1-29) to be used. The graphs are all plotted in Window 1, and various aspects of the current axes definitions of window 1 can be retained, as specified by the settings of the USEAXES option: limits allows y- and xaxis limits to be imposed for all plots (by the LOWER and UPPER parameters of XAXIS and YAXIS); style means that the current axis styles are used (ACTION parameter of XAXIS and YAXIS, together with the GRID option and BOX parameter of FRAME); marks means that the location and labelling of tick marks on both axes can be defined for the procedure (MARKS and LABELS parameters of XAXIS and YAXIS); mpositions means that the positions of tick marks on the axes can be defined (MPOSITION parameter of XAXIS and YAXIS); and transform uses any axis transformations (TRANSFORM parameter of XAXIS and YAXIS).

TRELLIS includes a key on each graphics page for plots other than boxplots if each window of the trellis contains more than plot (i.e. if there is more than one Y variate, or there is a PENGROUPS factor with more than one level). You can use the KEYHEIGHT option to control the size of the key in the y-direction, and setting this to zero will suppress the key. The DESCRIPTION parameter can be used to supply annotation for the key, in the same way as in the DGRAPH directive.

Example 4.8.3 uses TRELLIS to study the relationship between amounts of sulphur in the air and weather variables. Sulphur is plotted against (wind) Speed for every combination of factors Winddirection and Rain.

Example 4.8.3

```
" Comparison of air pollution and weather variables:
  sulphur levels against wind speed, wind direction, and rain."
FACTOR [LABELS=!t(N,NE,E,SE,S,SW,W,NW)] Direction
&      [LABELS=!t(no,yes)] Rain
READ   Sulphur,Speed,Direction,Rain; FREPRESENTATION=labels
```

Identifier	Minimum	Mean	Maximum	Values	Missing	
Sulphur	0.0000	10.46	49.00	114	0	Skew
Speed	0.5000	10.31	22.70	114	1	

Identifier	Values	Missing	Levels
Direction	114	1	8
Rain	114	0	2

```
TRELLIS [GROUPS=Direction,Rain; NROW=4;\
        TITLE='Sulphur versus wind speed';\
        YTITLE='Sulphur measurements';\
        XTITLE='Wind speed (km/h)'] Sulphur; Speed
```

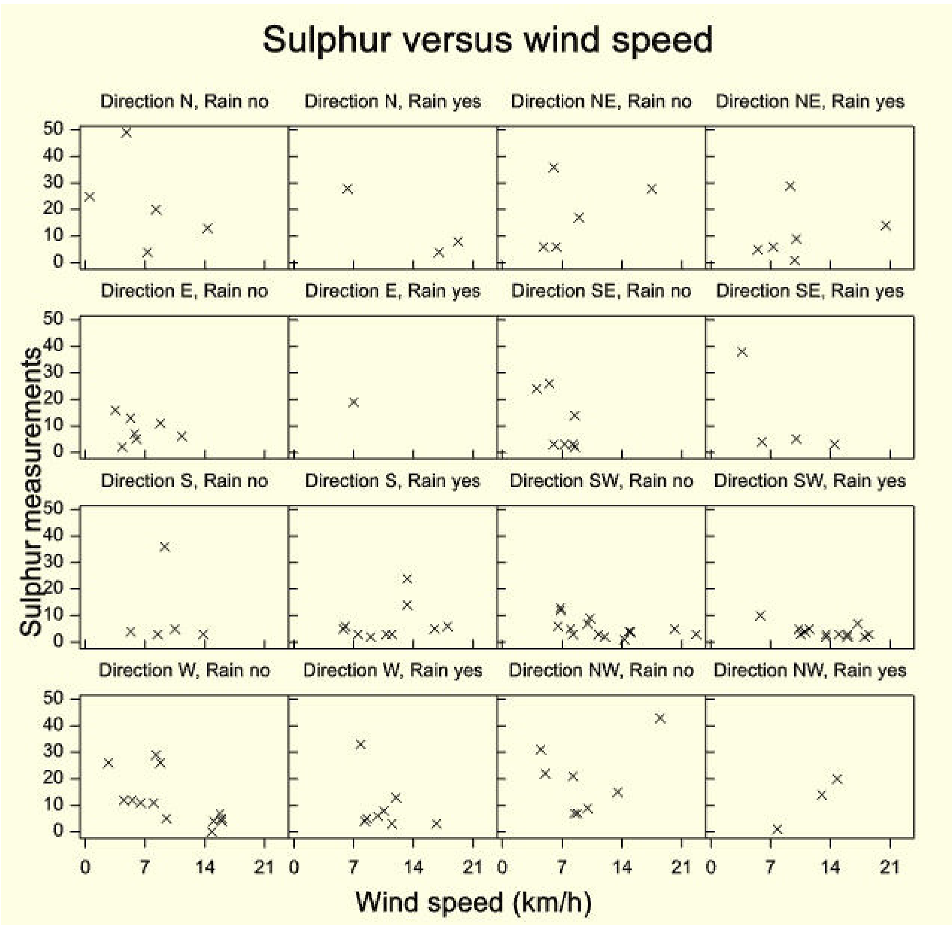


Figure 4.8.3

4.8.4 The DMSCATTER procedure: scatter-plot matrices

DMSCATTER procedure

Produces a scatter-plot matrix for one or two sets of variables (J. Ollerton & R.W. Payne).

Options

PLOT = *string tokens*	Additional information to include in the scatter plots (correlation, histograms); default *
SCALING = *string token*	How to scale the x- and y-axes (common, equal, none); default none
PEN = *scalar* or *variate* or *factor*	Pens to plot the scatter plots; default 1
PENHISTOGRAM = *scalar*	Pens to plot the histograms; if PEN is a factor the default plots the histogram for each group separately using the pen used for that group in the scatter plots, otherwise the default is to use pen 2
PENCORRELATION = *scalar*	Pen to use to write the correlations; default 1
PENTITLE = *scalar*	Pen to use to write the axis titles; default uses the pens currently defined for the axes in the windows that are used for the plots
PENAXIS = *scalar*	Pen to use to draw the axes; default uses the currently defined pens
PENLABELS = *scalar*	Pen to use to write the axis labels; default uses the currently defined pens
NROWS = *scalar*	Number of rows of graphs to put in a single frame (i.e. page); default puts them all in one frame
NCOLUMNS = *scalar*	Number of columns of graphs to put in a single frame; default uses the same value as NROWS
ASPECTRATIO = *scalar*	Ratio of the length of the y-axis to the length of the xaxis in each graph
FRAMESHAPE = *string token*	Shape of the plotting frame (landscape, portrait, square); default squa
MARGINSIZE = *scalar*	Specifies the size of the margins at the bottom and lefthand edge of the frame

Parameters

Y = *pointers*	Each pointer contains a set of variates and/or factors to be plotted
YTITLES = *texts*	Labels for the axes for the Y variates and factors, to use instead of their identifiers
YMARKS = *variates*, *scalars* or *pointers*	Marks to use on the axes for the Y variates and factors, if any of these contains missing values, the marks and their labels are suppressed for that variate or factor
X = *pointers*	Each pointer contains a set of variates and/or factors to be plotted as the x-variables in a rectangular scatter-plot matrix; if unset Y specifies both the x-variables and y-variables for a symmetric scatter-plot matrix
XTITLES = *texts*	Labels for the axes for the X variates and factors, to use instead of their identifiers
XMARKS = *variates*, *scalars* or *pointers*	Marks to use on the axes for the X variates and factors, if any of these contains missing values, the marks and their labels are suppressed for that variate or factor

---

Procedure DMSCATTER **produces two types** of scatter-plot matrix, using high-resolution graphics. For a symmetric scatter-plot matrix, the variates and/or factors to be plotted against each other must be specified, in a pointer, by the Y parameter. The scatter-plot contains a lower-triangular array of graphs, one for each pair of variables. Alternatively, for a rectangular scatter-plot matrix, there are two set of the variates and/or factors. The set that defines the y-values for the graphs are specified (in a pointer as before) by the Y parameter, and those that define the x-values for the graphs are specified (also in a pointer) by the X parameter. The scatter-plot now contains a rectangular array of graphs, one for each pair of x- and y-variables. If any of the variates or factors is restricted, only the units not excluded by the restriction will be plotted.

By default the identifiers of the relevant x- and y-variables are used for the titles of the axes at the lower and left-hand edges of the graphics frame (i.e. page). Alternatively, you define your own titles for the y-variables by setting the YTITLES to a text with a value for each Y variate or factor. Similarly, you can use the XTITLES parameter to supply your own titles for the X variates or factors.

The YMARKS parameter allows you to specify your own marks for the axes corresponding to the y-variables. (These are then used as the settings of the MARKS parameter of the YAXIS and XAXIS directives.) You can set YMARKS to single variate or scalar, if you want to use the same marks for every y-variable. Alternatively, you can set it to a pointer with a variate or factor for each Y variate or factor, if you want to specify different marks. If any of the variates or scalars contains missing values, the marks and their labels are suppressed on the corresponding axes. You can use the XMARKS parameter similarly, to specify axis marks for the x-variables.

The PEN option specifies the pens to be used to plot the graphs. The setting can be a scalar to plot all the points with the same pen, or a variate or a factor to use different pens. If PEN is set to a factor, a key is included in the plot to identify the correspondence between the pens and the groups. The default is to use pen 1.

The PLOT option allows you to specify extra information to be included in the plot, with settings:

correlation	prints the correlation of the pair of variables in each plot, at the top of the plot;
histograms	plots histograms of the variables down the diagonal of a symmetric scatter-plot matrix, or along the top and down the right-hand side of a rectangular scatter-plot matrix.

The PENHISTOGRAM option specifies the pens to plot the histograms. If PEN is a set to a factor, the default for PENHISTOGRAM plots histogram for each group, using the pen used for that group in the scatter plots. Otherwise the default is to use pen 2. The PENCORRELATION option specifies the pen to use to print the correlations; default 1.

The PENTITLE, PENAXIS and PENLABELS options define the pens to use for the titles of the x- and y-axes, for the axes themselves, and for their labels. If any of these is unset, the default is to use the pens already defined for that aspect of the axes in the windows used in the plot.

The SCALING option controls the scaling of the x- and y-axes, the settings:

equal	uses equal scaling for the x- and y-axes in each graph,
common	used exactly the same axes (upper and lower limits as well as scaling) for the axes in all the graphs,
none	defines all the axes independently (the default).

By default the plots are square, but you can request rectangular plots by setting the ASPECTRATIO option to the required value for the length of the y-axis divided by the length of the x-axis.

The MARGINSIZE option specifies the size of the margins at the bottom and left-hand edge of the graphics frame. If this is unset, the margins are defined automatically, using a smaller value if all the axis marks and labels on an edge have been suppressed.

The FRAMESHAPE option specifies the shape of the graphics frame, with settings:

landscape	for a frame of size 1.4 × 1.0 i.e. wider in the x- than the ydirection,
portrait	for a frame of size 1.0 × 1.4 i.e. wider in the y- than the xdirection,
square	for a frame of size 1.0 × 1.0.

Some graphics devices do not support the use of device cordinates greater than 1.0, so the default is FRAMESHAPE=square. (See FRAME (4.9.3.1) and DEVICE (4.9.1.1) for more information.)

By default the graphs are all plotted in a single frame (i.e. page), but you can specify the NROWS and NCOLUMNS options to split them across several frames. NROWS specifies the number of rows of plots to put in a single frame. The default is to fit them all into one frame. NCOLUMNS specifies the number of columns of plots to put in one frame. The default is to use the same value as NROWS.

An example is shown in Figure 4.8.4, which was generated by Example 4.8.4.

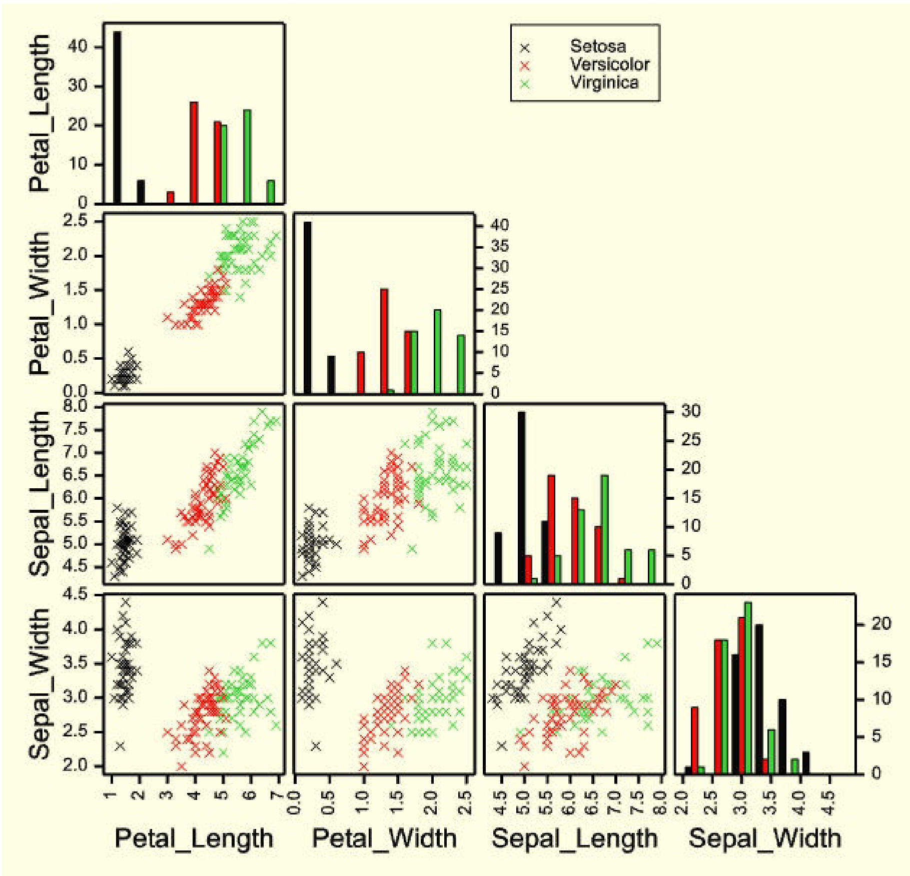


Figure 4.8.4

Example 4.8.4

```
SPLOAD    [PRINT=*] '%GENDIR%/Data/Iris.gsh'
CALCULATE PLength =(Petal_Length-MEAN(Petal_Length))/SQRT(VAR(Petal_Length))
&         PWidth=(Petal_Width - MEAN(Petal_Width))/SQRT(VAR(Petal_Width))
&         SLength =(Sepal_Length-MEAN(Sepal_Length))/SQRT(VAR(Sepal_Length))
&         SWidth = (Sepal_Width - MEAN(Sepal_Width))/SQRT(VAR(Sepal_Width))
POINTER   [VALUES=Petal_Length,Petal_Width,Sepal_Length,Sepal_Width]\
          Measurements
DMSCATTER [PLOT=histograms; PEN=Species] Measurements
```

4.9 The environment for high-resolution graphics

When you start GenStat an initial graphics environment is set up which contains default settings that are designed to be appropriate for the different types of plots. A graphics environment specifies how graphs are produced controlling aspects such as whether or not boxes are drawn around the plots, the display of the key, and the styles and the colours of outlines for graphs such as histograms or shade plots.

GenStat provides a range of environments to suit different tastes and situations. To see the choices, select the Graphics environments option from the Tools menu bar. This opens the menu shown in Figure 4.9.

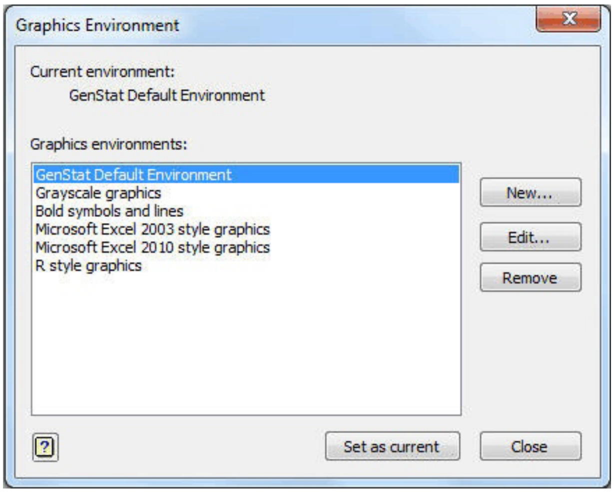


Figure 4.9

You can highlight another environment in the Graphics environments window, and then click on the Set as current button to use that instead. You can design your own graphics environment, by clicking on the New button. Alternatively, you can change an existing environment by clicking on the Edit button.

The directives described in the earlier sections of this chapter can display data in various ways. Implicit in all the discussion is the idea of a *graphics environment*, in which the displays are generated. This consists of a choice of graphics devices and a large number of parameters which control the appearance of the output. When you start GenStat an initial environment is created which contains default settings that are designed to be appropriate for the more common types of plot. This section describes the directives that allow you to modify the graphical environment in order to obtain more control over the appearance of your output. The descriptions of the commands earlier in this chapter indicate how the output will appear by default, and how it is affected by changes to the environment. The examples were chosen to illustrate the default display and some of the ways in which it can be modified by directives such as PEN (4.9.8), XAXIS (4.9.4) and YAXIS (4.9.5).

When you produce a high-resolution plot, the pictures are drawn on a *graphical device*, in a *graphical window*, using a *graphical pen*. Output can be produced on only one type of device at any time; however you can switch between different devices during a GenStat session so that, for example, you can experiment with various displays on the screen before sending some output to a file for printing as hard-copy. The device is selected using the DEVICE directive (4.9.1.1). **Note**, though, in GenStat *for Windows* you can save the display in these formats directly from the graphics viewer, **so the DEVICE statement is needed only if you want to run GenStat as a batch process**.

A graphics window is an area of the screen (or page on a plotter) that is used for plotting output. Many such windows can be used within a sequence of statements, so that several graphs may be plotted on a single screen. The position and size of the windows is defined using the FRAME directive (4.9.3.1). Associated with each window are the attributes of its axes. These control how axes are drawn by directives such as DGRAPH (4.2.1), DHISTOGRAM (4.3.1) and DCONTOUR (4.4.1). The XAXIS (4.9.4), YAXIS (4.9.5) and ZAXIS (4.9.6) directives can be used to control the various aspects of the axes associated with any specific window. These replace the AXES directive that was used, in releases before 4.2, to set attributes of the x- and y-axes. AXES is retained for compatibility, but it is less powerful than XAXIS and YAXIS. You can also include additional axes in a plot, and these can have oblique directions. They are defined using the AXIS (4.9.7) directive, and added into a particular graphics window using the FRAME directive (4.9.3.1).

Each part of the display is drawn using pens, each of which has attributes such as colour, line style and symbol type. In addition, the pen may be used to control how data is plotted, for example by requesting a straight line or a curve. The PEN directive (4.9.8) is used to set attributes of the different pens to be used in each graph.

The directives that define the environment change only the parameters that are mentioned explicitly; unspecified parameters retain their previous values (which may be the initial defaults). When you start a new job, the environment is reset to the initial default values. On the other hand, when you use RESUME to re-start an earlier session, the graphics environment will be loaded from the resume file. However, this does not affect the choice of output device (and associated file) which is preserved in both situations.

As the effects of these directives are additive, you need to keep aware of the current settings, and avoid unwanted side-effects which may occur, for example, if you use a pen that has earlier been modified in a way that is incompatible with its current use. This should not cause problems under ordinary circumstances. However, if you are using graphics in a general program or procedure there are various things you can do to make the graphics self-contained, and avoid side-effects. Each directive that modifies the environment includes a SAVE parameter that enables you to save the current settings of its particular aspect of the environment (frame, axes or pen) after making any modifications specified in the current statement. This enables you to check the current settings and reset particular attributes to their original values after a plot has been produced. The DKEEP directive can be used to obtain additional general information about the graphics devices and environment. The GET and SET directives allow the entire graphics environment to be stored in a pointer and later restored to its original state. For example, in a graphics procedure you might have the following statements:

```
GET           [SPECIAL=Special]
FRAME         1; YLOWER=0.3; YUPPER=0.6; XLOWER=0.3; XUPPER=0.6
YAXIS         1; LOWER=0; UPPER=100; TITLE='Percentages'
PEN           1...4; METHOD=line; LINESTYLE=1...4; SYMBOL=0; COLOUR=1,2,1,2
DGRAPH        Percent[1...4]; X; PEN=1...4
SET           [DSAVE=Special['dsave']]
ENDPROCEDURE
```

This can also be done automatically using the RESTORE option of PROCEDURE. Alternatively, you can save the current graphics environment settings to an external file using the DSAVE (4.9.11) directive, and reload then later using the DLOAD directive (4.9.11).

Information about the graphics environment can be displayed using the DHELP procedure.

DHELP procedure
Provides information about GenStat graphics (S.A. Harding).

No options

Parameter

TOPIC = *string tokens*	Lists the required graphics topics (current, possible); default poss

---

DHELP provides information about the GenStat high-resolution graphics. It has a single parameter called TOPIC, which supplies a list of strings indicating the topics about which you want information. The setting current gives details about the current settings of the graphics frames, windows, axes and pens, including the negatively numbered pens used as initial defaults. The other setting possible indicates the available frames, windows, axes and pens, and lists the available graphics devices (indicating which one is currently selected).

4.9.1 Devices

4.9.1.1 The DEVICE directive

DEVICE directive
Switches between (high-resolution) graphics devices.

No options

Parameters

NUMBER = *scalar*	Device number
ENDACTION = *string token*	Action to be taken after completing each plot (continue, pause)

ORIENTATION = *string token*

	Orientation of the pictures, if relevant (landscape, portrait); default * retains the current setting for this device
PALETTE = *string token*	How to represent colour (monotone, greyscale, grayscale, colour); default * retains the current setting for this device
SIZEPAGE = *string token*	Size of page for each screen (A4, A3); default * retains the current setting for this device
RESOLUTION = *scalar*	Specifies the height of the image for hard-copy output, in pixels

High-resolution graphics can be generated principally in two forms by GenStat: either on a screen that can operate in graphics mode or by sending output to a file. The screen-based operation is for use in interactive sessions, whereas file output is designed for later use outside GenStat: either to produce hard-copy on a plotter or laser-printer, or to re-display graphics on the screen, if appropriate software is available. Usually there is a choice of various kinds of screen type or file format. Each type of output, whether screen or file, is referred to as a device; thus, the first step in producing graphical output is selecting a device within GenStat that is appropriate for the hardware that you have available. GenStat has built-in interfaces to several different graphics devices. These vary according to the GenStat implementation. However, details of the devices, their characteristics and their associated numbers can be obtained from the DHELP procedure by typing the statement

```
DHELP possible
```

The output device is selected by the DEVICE statement. For example

```
DEVICE 4
```

selects the fourth available device.

If you have selected a file-based device you also need to open a file to receive the output, using the OPEN directive. This can be done before or after selecting the device, so long as the file has been opened before any output is generated. You can close the file when the graphics are complete; if you want to store separate items of graphical output in individual files you can use a sequence of OPEN and CLOSE statements. When opening or closing files for graphical output the CHANNEL parameter of the OPEN and CLOSE statements should be set to the device number specified by the DEVICE statement. For example:

```
OPEN   'PLOT.HPGL'; CHANNEL=4; FILETYPE=graphics
DEVICE 4
DGRAPH Y; X
CLOSE  4; FILETYPE=graphics
```

The default device, selected automatically when you start GenStat, is device 1: sometimes you may be able to specify an alternative device number and associated output file on the command line used to start GenStat (the local GenStat documentation should explain if this is possible).

You may get strange results if you try to generate graphics on a screen that is not designed for displaying graphics, or if you specify the wrong device type, as GenStat is not always able to detect the type of device or screen.

There need be little difference in your use of GenStat graphics on different devices as, by default, all the plotting symbols, brush styles and character output are *software-generated* using built-in graphics definitions that are supplied with GenStat. It may sometimes be advantageous, however, to use particular features of the device; for example, the use of *solid-fill* in histograms and pie charts. This can be generated by software on any device, using brush style 16 (see 4.9.8), but in many cases this can be performed in a fraction of the time by using the hardware instead. These device-specific features are usually selected by negative parameter settings (for example, by setting parameter SYMBOL=-3 in the PEN directive; 4.9.8). Naturally, selection of devicespecific attributes may lead to some differences in appearance of the output on different devices. Likewise different devices may have different initial default settings, in particular according to whether or not they support colour. Details of these device-specific properties are provided in the information provided by the DHELP procedure, as explained above.

The ENDACTION parameter, with settings continue and pause, controls the action taken by default at the end of each plot. When using a graphics terminal interactively it may be convenient to pause at the end of a plot to examine the screen. When you are ready to continue, pressing carriage-return or some equivalent key will switch the terminal back to text mode and the GenStat prompt will appear. The DHELP statement above should provide the precise details for each particular device. For some interactive devices, for example PCs or workstations with separate graphics windows, it may not be necessary to pause. Each device is initialized to either pause or continue when you start GenStat, according to the particular implementation. If you are running in batch mode the default will always be to continue.

You can repeat the DEVICE statement and set ENDACTION to pause or continue at any time that you wish to change the default action. Alternatively, each graphical directive has an ENDACTION option that controls the device at the end of that directive, without altering the general default setting. For example, if you wish to build up a complex display using several DGRAPH statements with option SCREEN=keep, you could set ENDACTION=continue in the DEVICE statement, and then put ENDACTION=pause in the final DGRAPH statement.

The ORIENTATION parameter can be used to specify landscape or portrait orientation of graphical output on PostScript and Interacter raster devices; portrait is the default. PALETTE can be set to monotone, to force all colours to be mapped to colour 1; this is the default for PostScript. Alternatively, PALETTE=colour produces colour PostScript output. The additional setting PALETTE=greyscale is as for monotone except that area filling (as in histograms) are shaded in grey tones, using the red component of the colour to define the grey intensity. The SIZEPAGE parameter allows you to specify A3 or A4 pagesizes for HPGL devices.

The RESOLUTION parameter specifies the height of the image for hard-copy output, in pixels. (This is equivalent to setting the image resolution in the Options menu of the GenStat Graphics Viewer.)

4.9.1.2 The SETDEVICE procedure

SETDEVICE procedure

Opens a graphical file on the basis of a file extension

Options

PRINT = *string token*	What to print (filename); default *
SURNAME = *text*	Surname of graphical file to open; default * uses the surname of the file on the current input channel
EXTENSION = *string token*	Extension of graphical file to create (hpg, eps, emf, jpg, tif, png, gmf or bmp); default png
SUBDIRECTORY = *text*	Subdirectory below the directory of the file on the current input channel in which the graphical files will be opened; default * does not use a subdirectory
FIRSTNUMBER = *scalar*	First sequence number for the graphical file name; default 1
NDIGITS = *scalar*	Number of zeros with which the sequence number of the graphics file is padded; default 2
PALETTE = *string token*	How to represent colour (monotone, greyscale, grayscale, colour); default colour
RESOLUTION = *scalar*	Specifies the height of the saved image in pixels; default 800
SEQUENCERESET = *string token*	Whether to continue with the sequence of graphical files (yes, no) default no
CREATEFILE = *scalar*	Logical expression indicating whether (1) or not (0) to create a new graphical file; default 1 creates a new graphics file
SAVEFILENAME = *text*	Saves the filename of the graphical file. This is only set to a new filename when the logical expression CREATEFILE is set to true

Parameters

FILENAME = *texts*	Name of the graphical file including one of the possible extensions .hpg, .eps, .emf, .jpg, .tif, .png, .gmf or .bmp
NUMBER = *scalars*	Saves the device number corresponding to the graphical format specified by parameter FILENAME or, if this is not set, by option setting EXTENSION

Procedure SETDEVICE is especially useful in batch jobs. In Release 15 there is already a SETDEVICE procedure, but in the Biometris Procedure Library there is an updated one. You can download this library from: http://www.biometris.wur.nl/UK/Software/Genstat+procedures/

Procedure SETDEVICE can be used in two different ways depending on whether the FILENAME parameter is set. In case FILENAME is set all options are ignored and a graphical file is opened with that name. The device number is set on the basis of the extension of the given filename. The table below shows the correspondence between the file extension and the device number.

Extension	Device	Type of file
.hpg	4	HPGL file
.eps	5	Encapsulated PostScript file
.emf	6	Windows Enhanced Meta File
.jpg	7	JPEG file
.tif	8	TIFF file
.png	9	Portable Network Graphics file
.gmf	10	GenStat Meta File
.bmp	11	Windows BitMaP file

PALETTE=colour, the default, produces coloured output and enables the use of the COLOUR directive to specify exactly the composition of the colours. Alternatively, the monotone setting forces all colours to be mapped to colour 1. The additional setting PALETTE=greyscale is as for monotone except that area filling (as in histograms) are shaded in grey tones, using the RED parameter of COLOUR to define the grey intensity. The RESOLUTION parameter specifies the height of the saved image in pixels. This is equivalent to setting the image resolution in the Options menu of the GenStat Graphics Viewer.

When the FILENAME parameter is not specified, the options settings determine automatic opening and closing of a sequence of graphics files. In case the SURNAME option is not set, the directory and surname of the file on the current input channel will be used to name the graphics file which will be opened. In case the SURNAME option is set, the given surname and the working directory will be used. The directory can be appended by a subdirectory, one level deep only, as specified by the SUBDIRECTORY option. The filename is then appended with '–', a sequence number and the EXTENSION option setting to give a new filename. The first sequence number is given by the setting of the FIRSTNUMBER option, and successive calls to SETDEVICE will update the sequence number. The NDIGITS option can be used to pad the sequence number with extra zeros. The corresponding graphics channel, according to the EXTENSION setting in the table above, is then closed and a new graphics file with the new filename is opened. The SEQUENCERESET option can be used to reset the sequence of filenames. Note that the option settings SURNAME, EXTENSION, SUBDIRECTORY, FIRSTNUMBER and NDIGITS are only relevant the first time SETDEVICE is called or when SEQUENCERESET=yes. In other cases the settings of a previous call to SETDEVICE are used.

By default a new plot file is always created, but this can be supressed by setting the CREATEFILE option to zero. This is especially convenient when multiple windows are used in a single graph, see the example program. The filename can be saved by means of the SAVEFILENAME option; this will only be set to a new filename when the logical expression CREATEFILE is set to 1 (true).

4.9.2 The DDISPLAY directive: re-displaying the graphics screen

---

DDISPLAY directive

Redraws the current graphical display.

Options

DEVICE = *scalar*	Device on which to redraw the display (on some systems it may only be possible to redisplay the picture on an interactive graphics device); default uses the current graphics device
ENDACTION = *string token*	Action to be taken after completing the plot (continue, pause); default * uses the setting from the last DEVICE statement

No parameters

This directive is provided to allow additional control of some interactive devices. In some of these the screen can operate in either text mode or graphics mode. GenStat will automatically switch the screen into the appropriate mode when starting or finishing a graph. Having returned to text mode after examining a graph you may later wish to have another look at the graph that was plotted. DDISPLAY will switch the screen back to graphics mode, thus re-displaying the graph. The ENDACTION option controls what happens after re-displaying the graph; normally with this type of device you would want to pause. The default action for DDISPLAY is the setting specified by the most recent DEVICE statement.

This directive has no effect when output is directed to a graphics file. For devices that do not operate in this dual-mode fashion, for example a graphics window under X-windows, DDISPLAY has no effect on the graphical display itself. It will however generate a pause if ENDACTION is set to request one.
Note that DDISPLAY does not actually re-plot the graphical output; it merely switches the screen into graphics mode, and assumes that your system has preserved the graphics image.

4.9.3 Positions of the plotting windows

4.9.3.1 The FRAME directive

FRAME directive

Defines the positions and appearance of the plotting windows within the frame of a high-resolution graph.

Options

GRID = *string tokens*	Specifies grid lines (xy, xz, yx, yz, zx, zy)
BOXFRAME = *string tokens*	Whether to include a box enclosing the entire frame (include, omit)
BACKGROUND = *scalars* or *texts*	Specifies the colour to be used for the background of the whole frame (where allowed by the graphics device)
RESET = *string token*	Whether to reset the window definition to the default values (no, yes); default no

Parameters

WINDOW = *scalars*	Window numbers
YLOWER = *scalars*	Lower y device coordinate for each window
YUPPER = *scalars*	Upper y device coordinate for each window
XLOWER = *scalars*	Lower x device coordinate for each window
XUPPER = *scalars*	Upper x device coordinate for each window
YMLOWER = *scalars*	Size of bottom margin (for x-axis labels); default 0.10
YMUPPER = *scalars*	Size of upper margin (for overall title); default 0.07
XMLOWER = *scalars*	Size of left-hand margin (for y-axis labels); default 0.12
XMUPPER = *scalars*	Size of right-hand margin; default 0.05
BACKGROUND = *scalars* or *texts*	Specifies the colour to be used for the background in each window (where allowed by the graphics device)
BOX = *string tokens*	Whether to include a box enclosing the plotted graphic (include, omit)
BOXSURFACE = *string tokens*	Box to include in a surface plot (full, bounded, omit)
BOXKEY = *string tokens*	Box to draw around key (full, bounded, omit)
PENTITLE = *scalars*	Pen to use to write the overall title; default -5
PENKEY = *scalars*	Pen to use for the key; default -6
PENGRID = *scalars*	Pen to use to draw the grid lines; default -4
SCALING = *string tokens*	How to scale the axis in each window (xyequal, xzequal, yzequal, xyzequal)

TPOSITION = *string tokens*	Position of title (right, left, center, centre)
CINTERIOR = *scalars* or *texts*	
	Specifies the colour to be used for the interior of each window (where allowed by the graphics device)
CFRAME = *scalars* or *texts*	Specifies the colour to be used for the frame of each window (where allowed by the graphics device)
CTITLE = *scalars* or *texts*	Specifies the colour to be used for the title bar of each window (where allowed by the graphics device)
AXES = *identifiers* or *pointers*	
	Additional oblique axes to include in each window; see Example 14
SAVE = *pointers*	Saves details of the current settings for the window concerned

You can define up to 256 different windows in which to plot graphics. Each window is a rectangular area of the screen which is defined using *normalized device coordinates* (NDC). For all devices, you can assume that a range of 0.0 to 1.0 will be available in both y- and xdirections, thus defining a 1 × 1 square to represent the plotting area. **On some devices the plotting area may extend further in either the y- or x-direction (but not both)**. Details can be obtained using the DHELP procedure, as explained at the start of Section 4.9. By keeping within the [0,1] range you can ensure that the window is always valid, whatever output device is selected. However, you may wish to use the extended area where possible on a particular device.

The mapping from NDC to physical coordinates on the current output device is performed internally, so the window definitions are independent of the actual size of device. The NDC coordinates are also completely independent of the values of the data that are to be plotted. (The locations of the points within the graph depend on how the axes of the graph are defined; see 4.9.4, 4.9.5 and 4.9.6).

When you use FRAME, any aspects of the windows that you do not specify explicitly retain the values that they had immediately before the FRAME statement. Alternatively, you can specify option RESET=yes to reset all these aspects to the default values, defined by GenStat at the start of each job.

To define a window, the upper and lower bounds are required in both y- and x-directions; thus defining both the position and the size of the window. For example

```
FRAME WINDOW=1; YLOWER=0.25; YUPPER=0.75; XLOWER=0; XUPPER=0.5
```

defines window 1 to be a square of size 0.5, whose bottom left corner is at the point (0.0,0.25), and whose top right corner is (0.5,0.75). This does not define the exact size of a graph plotted in this window, as margins may be required for the annotation and titles (see below).

If you do not specify all four values in the FRAME statement, the existing values are retained. A check is then made on the validity of the window bounds. The settings of YLOWER and XLOWER must be strictly less than those of YUPPER and XUPPER respectively; also, none of the bounds can be outside the permitted range, which is [0.0,1.0] on most graphics devices. **You cannot use * to reset a bound to the default value**; if you try to do so, GenStat will produce an error diagnostic. (Instead you can specify option RESET=yes, as explained above.)

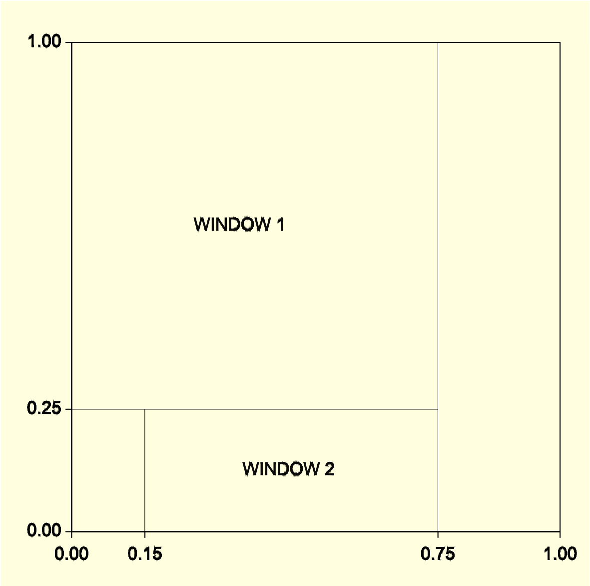


Figure 4.9.3.1a

All the windows have a default size defined when you start GenStat. Window 1 is the default window used for plots by DGRAPH, DCONTOUR, and so on, and is set up to be a square of size 0.75. The default key window is window 2, which is a rectangle of height 0.25 and width 0.75 located immediately below window 1. See Figure 4.9.3.1a. Windows 3 and 4 are the unit square [0,1]×[0,1] and windows 5, 6, 7 and 8 are the top-left, top-right, bottom-left, and bottom-right quarters respectively of the unit square. Windows 9, 10, 11 and 12 also divide the frame into quarters, but they have the full width (0 to 1) in the x-direction and quarter of the width in the y-direction, working from the top (i.e. 0.75 to 1 for window 9) to the bottom (i.e. 0 to 0.25 for window 12) of the frame. See Figures 4.9.3.1b and 4.9.3.1c for the positions of Windows 5 till 12. The remaining windows, from 13 to 256, also default to the unit square.

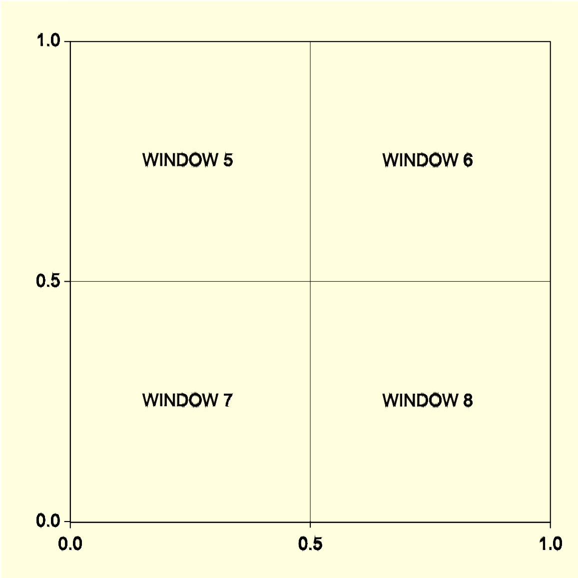


Figure 4.9.3.1b

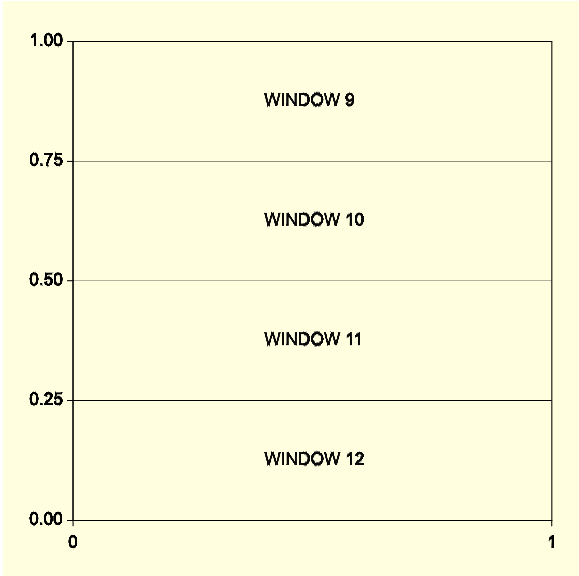


Figure 4.9.3.1c

You can use FRAME to modify the size or position of any of these windows.

Usually, a margin is provided around each plot so that there is room for the axes to be drawn, along with labelling and titles as specified by the XAXIS or YAXIS directives (4.9.4 and 4.9.5). In Figure 4.9.3.1d the default values of the 4 margins are shown. By default, the margin size is designed to allow sufficient room for annotation to be added using the standard character size, as defined by the SIZEMULTIPLIER or SMLABEL parameters of PEN (4.9.8). If you use XAXIS or YAXIS to control the plotting of axes explicitly you may wish to alter the size of the margins, either to increase the space used for the axes or, alternatively, to maximize the space available for the graph itself. For example, if you alter the size of the labelling, by explicitly defining the relevant axis pens, more space may be required for the axes; otherwise the labels may be clipped at the window bounds.

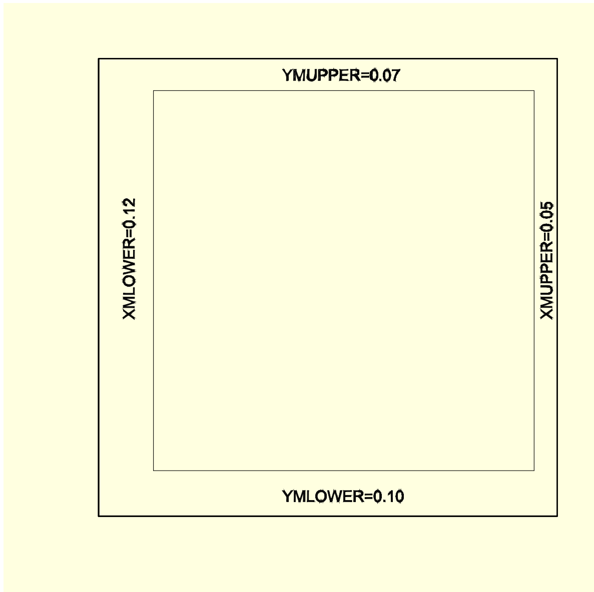


Figure 4.9.3.1d

The parameters YMLOWER, YMUPPER, XMLOWER and XMUPPER can be used to set the space (in NDC) for the bottom, top, left-hand and right-hand margins respectively, and have initial default settings of 0.10, 0.07, 0.12 and 0.05.

On most devices the background colours of the window may be modified by setting the BACKGROUND, CINTERIOR, CFRAME and CTITLE parameters. The BACKGROUND parameter can be used to define the colour for the whole background, while the other parameters define specific aspects (overriding any setting of BACKGROUND): CINTERIOR defines the colour of the interior of the plot (where the points are plotted), CFRAME defines the colour of the outer frame (outside the interior), and CTITLE is the colour of the title bar. The parameters can be set either to a text containing the name of one of GenStat's pre-defined colours (4.9.9), or to a scalar containing a number defining a colour using the RGB system; see 4.9.9. Similarly, the BACKGROUND option can define the background colour for the whole frame (which may include areas outside any of the windows).

The PENTITLE and PENKEY options allow you to define the pens to be used to write the overall title and the key in each window; the initial default is to use pen -5 and -6 respectively. The TPOSITION parameter can be used to specify the position of the title in each window: either left-justified, right-justified or centred. The initial default is that it is centred.

The GRID option allows you to request grid lines to be drawn in particular directions and planes (for all the windows listed by the WINDOW parameter). For example the setting xy requests lines in the xy plane running from the x-axis (that is, parallel to the y-axis), and the setting yx requests lines in the xy plane running from the y-axis (that is, parallel to the x-axis); so you can set both of these to obtain box markings in the xy plane. The PENGRID option specifies the pen to be used for the grid lines in each window; the initial default is to use pen -4. You must use the RESET option if you want to restore these pen numbers to the initial defaults. (**GenStat does not allow you to set negative pen numbers explicitly**.) The BOX parameter allows you to put a box around the window in plots other than surface plots; the initial default is to omit this. The box for a surface plot is controlled by the BOXSURFACE option, and can either be a full box enclosing the whole graph, or a bounded box enclosing just the surface; the initial default is that no box is drawn. The BOXKEY parameter can request that either a full or a bounded box be drawn around each key; the initial default is to omit the box. Finally, the BOXFRAME option controls whether or not a box is drawn around the entire frame; the initial default is to include the box.

The SCALING parameter enables you to request that scaling of the x-, y- or z-axes should be equal in each window. For example, the xyequal setting ensures that the x- and y-axes are scaled identically, the setting xyzequal ensures that all the axes have the same scaling, and so on.

The AXES parameter allows you to specify the identifier of an oblique axis (defined by the AXIS directive; see 4.9.7) that should be included in a window. See Example 14 in Chapter 5. If you want to include several axes, you can specify a pointer containing the identifiers of the required axes.

The current FRAME settings for a particular window can be saved in a pointer supplied by the SAVE parameter. The elements of the pointer are labelled to identify the components, as shown in Example 4.9.3.1.

Example 4.9.3.1

```
  2  FRAME 1; YLOWER=0.0; XUPPER=1.0; SAVE=Win1
  3  PRINT [ORIENTATION=across; RLWIDTH=18] Win1[]
```

	include
Win1['ylower']	0
Win1['yupper']	1.000
Win1['xlower']	0
Win1['xupper']	1.000
Win1['ymlower']	0.1000
Win1['ymupper']	0.07000
Win1['xmlower']	0.1200
Win1['xmupper']	0.05000
Win1['background']	-1
Win1['box']	include
Win1['boxsurface']	full
Win1['boxkey']	bounded
Win1['pentitle']	-5
Win1['penkey']	-6
Win1['pengrid']	-4
Win1['scaling']	
Win1['tposition']	centre
Win1['cinterior']	-1
Win1['cframe']	-1
Win1['ctitle']	-1

An alternative to FRAME, for setting up a plot-matrix of windows, is to use the FFRAME procedure. This uses FRAME internally to define windows in either a rectangular, square, lowersymmetric, uppersymmetric or diagonal pattern.

4.9.3.2 The FFRAME procedure

FFRAME procedure

Forms multiple windows in a plot-matrix for high-resolution graphics (P.W. Goedhart).

Options

PRINT = *string tokens*	Whether to display the layout and numbering of the plot-matrix in a table or in a high-resolution test-graph on the current device (table, testgraph); default *
ARRANGEMENT = *string token*	Type of plot-matrix (rectangle, square, lowersymmetric, uppersymmetric, diagonal); default rectangle
ROWS = scalar	Number of rows of plot-matrix; default 3
COLUMNS = scalar	Number of columns of plot-matrix; default 3
DIAGONALWINDOWS = *string token*	Whether to include or exclude the diagonal in symmetric plot-matrices (include, exclude); default include
SQUARESHAPES = *string token*	Whether to force the individual windows, excluding margins for annotation, to be square (yes, no); default no
STARTWINDOW = *scalar*	Specifies the number of the first window; default 1
TESTGRAPH = *variate*	Specifies windows to be displayed in a test-graph (if this option is set, only a test-graph is produced and all other settings are ignored); default *
NUMBERING = *string token*	Controls the way in which the individual windows are numbered (rowwise, columnwise); default rowwise
DEFINE = *string token*	Whether to define the windows within the procedure (windows, nothing); default wind
CLEARWINDOW = *scalar* or *variate*	Defines the windows for which the screen should be cleared; i.e. specifies the elements of the SCREEN pointer which are set to the single-values text 'clear', other element of SCREEN are set to 'keep'; default 1
RLOWER = *scalar*	Lowest y device coordinate; default 0
RUPPER = *scalar*	Highest y device coordinate; default 1
CLOWER = *scalar*	Lowest x device coordinate; default 0
CUPPER = *scalar*	Highest x device coordinate; default 1
RSKIP = *scalar*	Space between windows along the y-axis; default 0
CSKIP = *scalar*	Space between windows along the x-axis; default 0
MARGIN = *string tokens*	Sets the size of the margins for labels and titles (xtitle, ytitle, none, small); default *
YMLOWER = *scalar*	Size of bottom margin (x-axis labelling) in each window; default *
YMUPPER = *scalar*	Size of upper margin (overall title) in each window; default *
XMLOWER = *scalar*	Size of left-hand margin (y-axis labelling) in each window; default *
XMUPPER = *scalar*	Size of right-hand margin in each window; default *
RMLOWER = *scalar*	Additional size of bottom margin (x-axis labelling) in windows at the bottom of the plot-matrix; default 0
RMUPPER = *scalar*	Additional size of upper margin (overall title) in windows at the top of the plot-matrix; default 0
CMLOWER = *scalar*	Additional size of left-hand margin (y-axis labelling) windows at the left of the plot-matrix; default 0
CMUPPER = *scalar*	Additional size of right-hand margin in windows at the right of the plot-matrix; default 0
BACKGROUND = *text* or *scalar*	Specifies the colour to be used for the background in each window (where allowed by the graphics device); default 'background'

Parameters

NGRAPHS = *scalar*	To save the number of windows in the plot-matrix
SWINDOW = *pointer*	Pointer to save scalars with window numbers
SYLOWER = *pointer*	Pointer to save scalars with lower y device coordinates for each window
SYUPPER = *pointer*	Pointer to save scalars with upper y device coordinates for each window
SXLOWER = *pointer*	Pointer to save scalars with lower x device coordinates for each window
SXUPPER = *pointer*	Pointer to save scalars with upper x device coordinates for each window
SSCREEN = *pointer*	Pointer to save single-valued texts with value 'clear' or 'keep'; this depends only on the setting of the CLEARWINDOW option
SMYLOWER = *pointer*	Pointer to save scalars with size of bottom margins for each window
SMYUPPER = *pointer*	Pointer to save scalars with size of upper margins for each window
SMXLOWER = *pointer*	Pointer to save scalars with size of left-hand margin for each window
SMXUPPER = *pointer*	Pointer to save scalars with size of right-hand margin for each window

Procedure FFRAME supplements the FRAME directive with automatic definition of windows in a so-called plot-matrix. The ARRANGEMENT option defines the arrangement of the plot-matrix which can be in either a rectangle, square, lowersymmetric, uppersymmetric or diagonal. The number of rows and columns of the plot-matrix can be specified by options ROWS and COLUMNS. The COLUMNS option is only relevant when ARRANGEMENT=rectangle. The DIAGONALWINDOWS option defines whether or not the diagonal windows should be included for symmetric plot-matrices. The option setting SQUARESHAPES=yes forces each window, excluding margins for annotation, to be square.

By default the positions of the windows in the plot-matrix are defined within the procedure by means of a FRAME statement using windows 1, 2 ... Alternatively, you can use the STARTWINDOW option to start the window numbering at a different value. The windows are numbered by rows unless the NUMBERING option is set to columnwise. When the number of windows is larger than the maximum allowable number, the windows are not defined and a warning message is printed. In that case the parameters SYLOWER, SYUPPER, SXLOWER and SXUPPER can be used to save the y and x device coordinates of the windows for subsequent use in a FRAME statement. Likewise parameters SMYLOWER, SMYUPPER, SMXLOWER and SMXUPPER can be used to save the margins for each plot. Also, setting DEFINE=nothing does not define the windows. The NGRAPHS parameter saves the number of windows in the plot-matrix, and the SWINDOW parameter saves the window numbers.

Typically the screen should be cleared only for the first window in a plot-matrix. The SSCREEN parameter can be used to save the single-valued text 'clear' for the first window and 'keep' for all other windows. These texts can then be used to set the SCREEN option of the plot directives as is shown in the example. The CLEARWINDOW option can be employed in case the screen should be cleared for other windows.

By default the unit plot-square [0,1] x [0,1] is employed. Options RLOWER, RUPPER, CLOWER and CUPPER can be used to define a different plot-square. The RSKIP and CSKIP options can be used to increase the space between windows which is by default 0. The user must ensure that these options are set to sensible values.

The space used for labelling of axis and an overall title for each window can be controlled with options MARGIN, YMLOWER, YMUPPER, XMLOWER and XMUPPER options. Default values of these options ensure that an overall title and labels (not longer than 4 characters) along both axes are displayed when using the standard character size. However, the default values normally prohibit the display of titles along the axes. The settings xtitle and/or ytitle of the MARGIN option generate space for titles along the axes. The MARGIN option can have the following settings:

```
    MARGIN=*            YMLOW,YMUP,XMLOW,XMUP = 0.04, 0.04, 0.05, 0.01
    MARGIN=xtit         YMLOW,YMUP,XMLOW,XMUP = 0.09, 0.04, 0.05, 0.01
    MARGIN=ytit         YMLOW,YMUP,XMLOW,XMUP = 0.04, 0.04, 0.10, 0.01
    MARGIN=xtit,ytit    YMLOW,YMUP,XMLOW,XMUP = 0.09, 0.04, 0.10, 0.01
    MARGIN=none         YMLOW,YMUP,XMLOW,XMUP = 0
    MARGIN=small        YMLOW,YMUP,XMLOW,XMUP = 0.015
```

These values can be overridden by setting options YMLOWER, YMUPPER, XMLOWER, XMUPPER explicitly. The outer margins of the border plots in the plot-matrix can be increased by specifying options RMLOWER, CMLOWER, RMUPPER and RMLOWER. For example, you could expand these, as needed for a trellis-style plot, by

```
FFRAME    [ROWS=nwrows; COLUMNS=nwcols; MARGIN=none;\
          CMLOWER=0.04; YMLOWER=0.09; RMUPPER=0.04]
```

The background colour for all windows may be modified by the BACKGROUND option.

The PRINT option can be used to display the layout and numbering of the plot-matrix in a table or in a high-resolution test-graph on the current device. A test-graph can also be requested by setting the TESTGRAPH option to a variate with the window numbers to be displayed; all other settings are then ignored.

4.9.4 The XAXIS directive

There is a definition for the axes associated with each GenStat graphics window, which specifies how the axes are to be drawn when graphical output is produced in that window. The default definition for each set of axes requires some of the features to be determined from the data, as described below. Others have fixed defaults that are independent of the data. The XAXIS directive can be used to override the default action and specify particular aspects of the x-axis explicitly. Similarly, directives YAXIS (4.9.5) and ZAXIS (4.9.6) modify the y- and z-axis definitions, respectively.

XAXIS directive

Defines the x-axis in each window for high-resolution graphics.

Option

RESET = *string token*	Whether to reset the axis definition to the default values (no, yes); default no

Parameters

WINDOW = *scalars*	Numbers of the windows
TITLE = *texts*	Title for the axis
TPOSITION = *string tokens*	Position of title (middle, end) ; default midd
TDIRECTION = *string tokens*	Direction of title (parallel, perpendicular); default para
LOWER = *scalars*	Lower bound for axis
UPPER = *scalars*	Upper bound for axis
MARKS = *scalars* or *variates*	Distance between each tick mark (scalar) or positions of the marks along the axis (variate)
MPOSITION = *string tokens*	Positioning of the tick marks on the axis (inside, outside, across); ; default outs
LABELS = *texts* or *variates*	Labels at each major tick mark
LPOSITION = *string tokens*	Position of the axis labels (inside, outside) ; default outs
LDIRECTION = *string tokens*	Direction of the axis labels (parallel, perpendicular) ; default para
LROTATION = *scalars* or *variates*	Rotation of the axis labels
YORIGIN = *scalars*	Position on y-axis at which the axis is drawn
ZORIGIN = *scalars*	Position on z-axis at which the axis is drawn
PENTITLE = *scalars*	Pen to use to write the axis title; default -1
PENAXIS = *scalars*	Pen to use to draw the axis; default -2
PENLABELS = *scalars*	Pen to use to write the axis labels; default -3
ARROWHEAD = *string tokens*	Whether the axis should have an arrowhead (include, omit) ; default omit
ACTION = *string tokens*	Whether to display or hide the axis (display, hide) ; default disp
TRANSFORM = *string tokens*	Transformed scale for the axis (identity, log, log10, logit, probit, cloglog, square, exp, exp10, ilogit, iprobit, icloglog, root); default iden

LINKED = *scalars*	Linked axis whose definitions should be used for this axis in a 2-dimensional graph; default * i.e. none
MLOWER% = *scalars*	How large a margin to set between the lowest x-value and the lower value of the axis, if not set explicitly by LOWER (expressed as a percentage of the range of the xvalues) ; default 5
MUPPER% = *scalars*	How large a margin to set between the largest x-value and the upper value of the axis, if not set explicitly by UPPER (expressed as a percentage of the range of the xvalues) ; default 5
DECIMALS = *scalars* or *variates*	Number of decimal places to use for numbers printed at the marks
DREPRESENTATION = *scalars* or *variates*	Format to use for dates and times printed at the marks
VREPRESENTATION = *string tokens*	Format to use for numbers printed at the marks (decimal, engineering, scientific); default deci
YOMETHOD = *string tokens*	Method to use to set the position of the origin on the yaxis if not set explicitly by YORIGIN (upper, lower, center, centre)
ZOMETHOD = *string tokens*	Method to use to set the position of the origin on the zaxis if not set explicitly by ZORIGIN (upper, lower, center, centre)
REVERSE = *string tokens*	Whether to reverse the axis direction to run from upper to lower instead of the default lower to upper (yes, no); default no
SAVE = *pointers*	Saves details of the current settings for the axis concerned

All the parameters of XAXIS are relevant when using DGRAPH (4.2.1), **but for other directives only some of the parameters are used.**

The WINDOW parameter specifies the window whose axis definition is to be altered. WINDOW can be set to a list of window numbers, in which case the other parameter lists are cycled in parallel, in the usual way. By default, only those aspects specified by subsequent parameter lists are modified; any parameters that are not set will retain their current settings. Alternatively, you can specify option RESET=yes to reset the values of any parameters that are not set for each window, back to the default values that are set up by GenStat at the start of a job.

The LOWER and UPPER parameters specify the lower and upper bounds for the axis. By default, GenStat derives suitable axis bounds from the data, as described for the appropriate directive. You can obtain an inverted scale by setting parameter REVERSE=yes. The values specified with these parameters are on the scale of the data values that are plotted, and are independent of the normalized device coordinates used to define the window size in FRAME (4.9.3.1). The MLOWER% parameter controls the size of margin that is provided between the lower value of the axis and the smallest x-value, if the lower axis value is not set explicitly by LOWER. This is expressed as a percentage of the range of the x-values, and has the initial default of 5%. Similarly the MUPPER% parameter controls the size of the upper margin.

The YORIGIN parameter determines the value on the y-axis through which the axis is drawn. If its value is outside the y-axis bounds, the upper or lower bound is adjusted so that the axis will extend up to the specified origin. This applies whether you have set the bounds explicitly or have left GenStat to calculate them from the data. If YORIGIN is not set, the YOMETHOD parameter can specify how the position should be determined: either at the upper value on the y-axis, or the lower value, or in the centre. The initial default (if neither of these parameters has been specified) is to put the axis at the bottom of the y-axis, which will be the lower value unless the scale is reversed. The ZORIGIN and ZOMETHOD parameters set the position of the origin on the z-axis in a similar way.

You can specify a title for the axis using the TITLE parameter. **This is limited to a single line of characters**. The TPOSITION parameter controls whether the title is placed in the middle or at the end of the axis, and the DIRECTION parameter controls whether it is written parallel or perpendicular to the axis.

The axis is marked with a scale, determined automatically so that tick marks are evenly spaced and positioned to give "round" numbers for the scale values. You can set the MARKS parameter to a scalar to define the increment between tick marks. For example, setting MARKS=1.5 with bounds 10 and 2 causes tick marks to appear at 2, 3.5, 5, 6.5, 8 and 9.5. The interval must be a positive number, irrespective of the values of the bounds. Alternatively, you can set MARKS to a variate (with more than one value) to specify the actual positions of the tick marks on the axis. **Any values that lie outside the axis bounds are ignored**. The scale values printed next to the tick marks use a format that is determined automatically from the values, but if you set MARKS to a variate it will use the number of decimals specified in the variate declaration. If MARKS is unset or set to a scalar, you can use the NSUBTICKS parameter to specify a number of "subticks" to be drawn between each of the (major) tick marks.

When you set MARKS, you can also use the LABELS parameter to specify a set of labels to print at the (major) axis marks, instead of the numbers. For example,

```
TEXT     [VALUES=Mon,Tues,Wed,Thur,Fri,Sat,Sun] Day
VARIATE [VALUES=1...31] Month
XAXIS    1; MARKS=Month; LABELS=Day
```

The strings within the text are cycled if necessary, so the number of strings can be less than the number of tick marks. The DECIMALS parameter can set the number of decimal places to use if you are printing numbers at the marks. If the numbers represent dates or times, you should specify their formats using the DREPRESENTATION parameter of the numerical GenStat data structures as variates for example.. By default, numbers are printed in decimal form. If you would prefer scientific format you can set parameter VREPRESENTATION=scientific; numbers are then printed as a decimal number with absolute value less than 10, followed by an exponent (e.g. 3.4567 E4 for 34567). Alternatively, you can set VREPRESENTATION=engineering to use engineering format; the decimal number then has an absolute value less than 10000, so the exponent is a multiple of 3 (e.g. 34.567 E3 for 34567). With scientific or engineering formats, the DECIMALS parameter sets the number of significant figures to use rather than the number of decimal places.

The MPOSITION parameter controls the positioning of the tick marks, which can be drawn on the inside or the outside of the axis, or can be drawn across the axis. With the outside setting, the tick marks are drawn towards the outside of the plot; that is below the axis if the axis is in the lower half of the plot, or above the axis if it is in the top half of the plot. The aim is then to position the tick marks away from the main part of the plot, so that they interfere with the plotted points as little as possible. With the inside setting, the marks are drawn on the opposite side (that is, to the inside of the plot), while the across setting draws them across the axis. Similarly, the positioning of the scale markings or labels is controlled by the LPOSITION parameter, with settings inside or outside. The LDIRECTION parameter controls whether the scale markings or labels are written parallel or perpendicular to the axis. Alternatively, you can use the LROTATION parameter to specify the direction of the labels more precisely, as a rotation in degrees from the horizontal (i.e. parallel) direction. If LROTATION is specified, any setting of LDIRECTION is ignored.

Setting MARKS=* will return to the default positioning of the tick marks. Setting LABELS=* will switch off any labels previously specified. Setting MPOSITION=* will switch off any tick marks, and setting LPOSITION=* or LDIRECTION=* will switch off any labels.

The TRANSFORM parameter allows you to transform the scale of the axis. **The tick marks are still defined and labelled according to the original scale**, but their physical positions on the graph are transformed. So, for example, with TRANSFORM=log10, the equal physical distance between 1 and 10 would be the same as the distance between 10 and 100. The settings are the same as the names of the equivalent GenStat functions, with the addition of exp10 for the antilog transformation (i.e. 10), and square *x* for *x*2.

There are three parameters to control the pens to be used to draw the axis. These are PENTITLE, PENAXIS and PENLABEL, specifying the pen for the title, the axis and the labelling, respectively. The initial default is to use pens -1, -2 and -3 in every window. These pens are given negative numbers to allow them to be distinguished from the pens used for the contents of the plot. They are initially set up to use colour black, line style 1, thickness 1 and size 1. You can thus control which pens are used for drawing the axis in each window, and the attributes of those pens. For example, if no XAXIS statement has yet been given,

```
PEN    -1; LINESTYLE=4; COLOUR=2
```

will request that the titles in every window should be written in line style 4 and colour 2; while

```
PEN    29; LINESTYLE=3; COLOUR=4
XAXIS 1; PENAXIS=29
```

will change the appearance of just the x-axis in window 1, as pen 29 is not used for the other windows. You should of course be careful of side-effects when changing the pen numbers. For example, pen 29 may also have been modified for use in a DGRAPH statement and other attributes may have been set that are not wanted when drawing the axis. You must use the RESET option if you want to restore these pen numbers to the initial defaults. (GenStat does not allow you to set negative pen numbers explicitly.)

The ARROWHEAD parameter controls whether the axis is drawn with an arrowhead at the end. You may sometimes wish to use the axis definitions merely to control the positioning of the plot in the x-direction (using the UPPER and LOWER parameters), or you may wish to hide the axis temporarily in case it is obscuring information in the plot. You can do this by setting parameter ACTION=hide.

Axis annotation is plotted in the margins specified by the FRAME directive (4.9.3.1). You may wish to reduce the size of these margins if you have defined axes that use less space, for example by keeping within the area of the graph itself, or by omitting titles or labels. Space can thus be regained and used for plotting data. However, if the margins are too small the axis annotation may be "clipped" at the boundaries of the margins; if this happens, you can use FRAME to increase the margin size. The margins are used by DGRAPH (4.2.1), DHISTOGRAM (4.3.1) and DCONTOUR (4.4.1), but they are ignored by other directives.

The LINKED parameter is useful when you have several related plots in different windows within the frame. If, for example, you set LINKED=n, the attributes of the current x-axis will all be taken (at the time of plotting) from the definition of the x-axis for any 2-dimensional graph in window *n*. Also, you can edit the attributes of all the linked axes simultaneously in the graphics viewer in GenStat *for Windows*.

The current settings of the axis for a particular window can be saved in a pointer supplied by the SAVE parameter. The SAVE parameter The elements of the pointer are labelled to identify the components, as shown in Example 4.9.4.

Example 4.9.4

```
  2  XAXIS 7; TITLE='x-axis'; LOWER=2; UPPER=10; MARKS=1.5; SAVE=Axes7
  3  PRINT [ORIENTATION=across; RLWIDTH=19] Axes7[]
               Axes7['title']        x-axis
           Axes7['tposition']        middle
          Axes7['tdirection']      parallel
               Axes7['lower']          2.00
               Axes7['upper']         10.00
               Axes7['marks']         1.500
           Axes7['mposition']       outside
              Axes7['labels']
           Axes7['lposition']       outside
          Axes7['ldirection']      parallel
           Axes7['lrotation']             *
           Axes7['nsubticks']             0
             Axes7['yorigin']             *
             Axes7['zorigin']             *
            Axes7['pentitle']            -1
             Axes7['penaxis']            -2
           Axes7['penlabels']            -3
           Axes7['arrowhead']          omit
              Axes7['action']       display
           Axes7['transform']      identity
              Axes7['linked']             *
             Axes7['mlower%']         5.000
             Axes7['mupper%']         5.000
            Axes7['decimals']             *
     Axes7['drepresentation']             *
     Axes7['vrepresentation']      decimals
            Axes7['yomethod']             *
            Axes7['zomethod']             *
             Axes7['reverse']            no
```

The settings are those for the axis itself, so you should check that the axis is not linked to one in another window. (The 'linked' element contains the window number, or a missing value there is no link.) This facility is of most use within procedures, where you may wish to check or modify particular axis settings before constructing complicated graphs. Also, the DKEEP directive (4.9.10) allows you to extract the actual bounds used when plotting; these will be the bounds determined from the data if none have been defined explicitly by XAXIS.

4.9.5 The YAXIS directive

YAXIS directive

Defines the y-axis in each window for high-resolution graphics.

Option

RESET = *string token* — Whether to reset the axis definition to the default values (no, yes); default no

Parameters

WINDOW = *scalars* — Numbers of the windows
TITLE = *texts* — Title for the axis
TPOSITION = *string tokens* — Position of title (middle, end)); default midd
TDIRECTION = *string tokens* — Direction of title (parallel, perpendicular); default para
LOWER = *scalars* — Lower bound for axis
UPPER = *scalars* — Upper bound for axis
MARKS = *scalars* or *variates* — Distance between each tick mark (scalar) or positions of the marks along the axis (variate)
MPOSITION = *string tokens* — Positioning of the tick marks on the axis (inside, outside, across); default outs
LABELS = *texts* or *variates* — Labels at each major tick mark
LPOSITION = *string tokens* — Position of the axis labels (inside, outside); default outs
LDIRECTION = *string tokens* — Direction of the axis labels (parallel, perpendicular); default perp
LROTATION = *scalars* or *variates* — Rotation of the axis labels
NSUBTICKS = *scalars* — Number of subticks per interval (ignored if MARKS is a variate)
XORIGIN = *scalars* — Position on x-axis at which the axis is drawn
ZORIGIN = *scalars* — Position on z-axis at which the axis is drawn
PENTITLE = *scalars* — Pen to use to write the axis title; default -1
PENAXIS = *scalars* — Pen to use to draw the axis; default -2
PENLABELS = *scalars* — Pen to use to write the axis labels; default -3
ARROWHEAD = *string tokens* — Whether the axis should have an arrowhead (include, omit); default omit
ACTION = *string tokens* — Whether to display or hide the axis (display, hide); default disp
TRANSFORM = *string tokens* — Transformed scale for the axis (identity, log, log10, logit, probit, cloglog, square, exp, exp10, ilogit, iprobit, icloglog, root); default iden
LINKED = *scalars* — Linked axis whose definitions should be used for this axis in a 2-dimensional graph; default * i.e. none
MLOWER% = *scalars* — How large a margin to set between the lowest y-value and the lower value of the axis, if not set explicitly by LOWER (expressed as a percentage of the range of the yvalues); default 5
MUPPER% = *scalars* — How large a margin to set between the largest y-value and the upper value of the axis, if not set explicitly by UPPER (expressed as a percentage of the range of the yvalues); default 5
DECIMALS = *scalars* or *variates* — Number of decimal places to use for numbers printed at the marks
DREPRESENTATION = *scalars* or *variates* — Format to use for dates and times printed at the marks
VREPRESENTATION = *string tokens*

	Format to use for numbers printed at the marks (decimal, engineering, scientific); default deci
XOMETHOD = *string tokens*	Method to use to set the position of the origin on the xaxis if not set explicitly by XORIGIN (upper, lower, center, centre)
ZOMETHOD = *string tokens*	Method to use to set the position of the origin on the zaxis if not set explicitly by ZORIGIN (upper, lower, center, centre)
REVERSE = *string tokens*	Whether to reverse the axis direction to run from upper to lower instead of the default lower to upper (yes, no); default no
SAVE = *pointers*	Saves details of the current settings for the axis concerned

The syntax of YAXIS is identical to that of XAXIS (4.9.4), except that YAXIS has XORIGIN and XOMETHOD parameters which replaces the YORIGIN and YOMETHOD parameters of XAXIS. All the parameters are relevant when using DGRAPH (4.2.1), but for other directives only some of the parameters are used.

As in XAXIS, the WINDOW parameter specifies the window whose axis definition is to be altered. By default, only those aspects specified by subsequent parameter lists are modified, but you can specify option RESET=yes to reset the values of any parameters that are not set for each window, back to the default values that are set up by GenStat at the start of a job. The LOWER, UPPER, MLOWER% and MUPPER% parameters again specify the lower and upper bounds for the axis, the REVERSE parameter can reverse the axis, and the TITLE, TPOSITION and TDIRECTION parameter can define a title for the axis.

The XORIGIN parameter determines the value on the x-axis through which the axis is drawn. If its value is outside the x-axis bounds, the upper or lower bound is adjusted so that the axis will extend up to the specified origin. This applies whether you have set the bounds explicitly or have left GenStat to calculate them from the data. If XORIGIN is not set, the XOMETHOD parameter can specify how the position should be determined: either at the upper value on the x-axis, or the lower value, or in the centre. The initial default (if neither of these parameters has been specified) is to put the axis at the left-hand end of the x-axis, which will be the lower value unless the scale is reversed. The ZORIGIN and ZOMETHOD parameters set the position of the origin on the z-axis in a similar way, with the initial default that the axis is at the bottom of the z-axis.

The MARKS, NSUBTICKS, LABELS, DECIMALS, DREPRESENTATION and VREPRESENTATION parameters also operate as in XAXIS, to specify the markings on the axis, and their associated labels. The MPOSITION, LPOSITION, LDIRECTION and LROTATION parameters again control the positioning of the tick marks and labels. For a y-axis, the outside setting implies that the tick marks are drawn to the left of the axis if the axis is on the left-half side of the plot, or to the right of the axis if it is on the right-hand side. As in XAXIS, the TRANSFORM parameter allows you to transform the physical scale of the axis on the graph.

The ARROWHEAD parameter again controls whether the axis is drawn with an arrowhead at the end, and parameters PENTITLE, PENAXIS and PENLABEL specify the to be used for the title, the axis and the labelling, respectively. ACTION allows you to hide the axis, LINKED allows you to take all the axis settings from a (linked) axis in another window, and SAVE allows you to save the current settings defined for the axis. Further details are given in the description of XAXIS (4.9.4).

4.9.6 The ZAXIS directive

ZAXIS directive

Defines the z-axis in each window for high-resolution graphics.

Option

RESET = *string token*	Whether to reset the axis definition to the default values (no, yes); default no

Parameters

WINDOW = *scalars*	Numbers of the windows
TITLE = *texts*	Title for the axis
TPOSITION = *string tokens*	Position of title (middle, end); default midd
TDIRECTION = *string tokens*	

	Direction of title (parallel, perpendicular); default para
LOWER = *scalars*	Lower bound for axis
UPPER = *scalars*	Upper bound for axis
MARKS = *scalars* or *variates*	Distance between each tick mark (scalar) or positions of the marks along the axis (variate)
MPOSITION = *string tokens*	Positioning of the tick marks on the axis (inside, outside, across); default outs
LABELS = *texts* or *variates*	Labels at each major tick mark
LPOSITION = *string tokens*	Position of the axis labels (inside, outside); default outs
LDIRECTION = *string tokens*	Direction of the axis labels (parallel, perpendicular); default perp
LROTATION = *scalars* or *variates*	Rotation of the axis labels
NSUBTICKS = *scalars*	Number of subticks per interval (ignored if MARKS is a variate)
XORIGIN = *scalars*	Position on x-axis at which the axis is drawn
YORIGIN = *scalars*	Position on y-axis at which the axis is drawn
PENTITLE = *scalars*	Pen to use to write the axis title; default -1
PENAXIS = *scalars*	Pen to use to draw the axis; default -2
PENLABELS = *scalars*	Pen to use to write the axis labels; default -3
ARROWHEAD = *string tokens*	Whether the axis should have an arrowhead (include, omit); default omit
ACTION = *string tokens*	Whether to display or hide the axis (display, hide); default disp
MLOWER% = *scalars*	How large a margin to set between the lowest z-value and the lower value of the axis, if not set explicitly by LOWER (expressed as a percentage of the range of the z-values); default 5
MUPPER% = *scalars*	How large a margin to set between the largest z-value and the upper value of the axis, if not set explicitly by UPPER (expressed as a percentage of the range of the zvalues); default 5
DECIMALS = *scalars* or *variates*	Number of decimal places to use for numbers printed at the marks
DREPRESENTATION = *scalars* or *variates*	Format to use for dates and times printed at the marks
VREPRESENTATION = *string tokens*	Format to use for numbers printed at the marks (decimal, engineering, scientific); default deci
XOMETHOD = *string tokens*	Method to use to set the position of the origin on the xaxis if not set explicitly by XORIGIN (upper, lower, center, centre)
YOMETHOD = *string tokens*	Method to use to set the position of the origin on the yaxis if not set explicitly by YORIGIN (upper, lower, center, centre)
REVERSE = *string tokens*	Whether to reverse the axis direction to run from upper to lower instead of the default lower to upper (yes, no); default no
SAVE = *pointers*	Saves details of the current settings for the axis concerned

The syntax of ZAXIS is identical to that of XAXIS (4.9.4), except that ZAXIS has an XORIGIN parameter instead of the ZORIGIN parameter of XAXIS. **All parameters are relevant when using D3GRAPH (4.2.2), but for other directives only some of the parameters are used**.

The XORIGIN parameter determines the value on the x-axis through which the axis is drawn. If its value is outside the x-axis bounds, the upper or lower bound is adjusted so that the axis will extend up to the specified origin. This applies whether you have set the bounds explicitly or have left GenStat to calculate them from the data. If XORIGIN is not set, the XOMETHOD parameter can specify how the position should be determined: either at the upper value on the x-axis, or the lower value, or in the centre. The initial default (if neither of these parameters has been specified) is to put the axis at the left-hand end, which will be the lower value unless the scale is reversed.

4.9.7 The AXIS directive

AXIS directive

Defines an oblique axis for high-resolution graphics.

Option

RESET = *string token*	Whether to reset the axis definition to the default values (yes, no); default no

Parameters

IDENTIFIER = *identifiers*	Name to be used inside GenStat to identify each axis
TITLE = *texts*	Title for each axis
TPOSITION = *string tokens*	Position of title (middle, end); default midd
TDIRECTION = *string tokens*	Direction of title (parallel, perpendicular); default para
LOWER = *scalars*	Lower bound for each axis
UPPER = *scalars*	Upper bound for each axis
MARKS = *scalars* or *variates*	Distance between each tick mark (scalar) or positions of the marks along each axis (variate)
MPOSITION = *string tokens*	Positioning of the tick marks on each axis (inside, outside, across) ; default outs
LABELS = *texts* or *variates*	Labels at each major tick mark
LPOSITION = *string tokens*	Position of the axis labels (inside, outside); default outs
LDIRECTION = *string tokens*	Direction of the axis labels (parallel, perpendicular); default para
LROTATION = *scalars* or *variates*	Rotation of the axis labels
NSUBTICKS = *scalars*	Number of subticks per interval (ignored if MARKS is a variate); default 0
XZERO = *scalars*	Position of the axis origin in the x-dimension; default 0
YZERO = *scalars*	Position of the axis origin in the y-dimension; default 0
ZZERO = *scalars*	Position of the axis origin in the z-dimension; default 0
XSTEP = *scalars*	Step in the x-direction corresponding to a step of length one along the axis
YSTEP = *scalars*	Step in the y-direction corresponding to a step of length one along the axis
ZSTEP = *scalars*	Step in the z-direction corresponding to a step of length one along the axis
PENTITLE = *scalars*	Pen to use to write the axis title
PENAXIS = *scalars*	Pen to use to draw the axis
PENLABELS = *scalars*	Pen to use to write the axis labels
ARROWHEAD = *string tokens*	Whether the axis should have an arrowhead (include, omit); default omit
ACTION = *string tokens*	Whether to display or hide the axis (display, hide); default disp
TRANSFORM = *string tokens*	Transformed scale for the axis (identity, log, log10, logit, probit, cloglog, square, exp, exp10, ilogit, iprobit, icloglog, root); default iden
DECIMALS = *scalars* or *variates*	Number of decimal places to use for numbers printed at the marks
DREPRESENTATION = *scalars* or *variates*	Format to use for dates and times printed at the marks
VREPRESENTATION = *string tokens*	Format to use for numbers printed at the marks (decimal, engineering, scientific); default deci
SAVE = *pointers*	Saves details of the current settings for the axis concerned

The AXIS directive allows you to define an oblique axis for high-resolution graphics. You use the IDENTIFIER parameter to supply an identifier to store the axis definition. You can then use this as a setting of the AXES parameter of the FRAME directive (4.9.3.1) to display the axis in a particular graphics window. The position of the axis origin in the x-, y- and z-dimensions of the window is specified by the parameters XZERO, YZERO and ZZERO, respectively. The XSTEP, YSTEP and ZSTEP parameters define the size of the steps in the x-, y- and z-directions that corresponds to a step of length one along the axis. These six parameters thus define the location and direction of the axis. The other parameters operate as in the XAXIS directive (4.9.4).

4.9.8 The PEN directive

PEN directive

Defines the properties of "pens" for high-resolution graphics.

Options

RESET = *string token*	Whether to reset the pen definitions to their default values (no, yes); default no
BOXUNITS = *string token*	Units to use for text boxes (characters, distance); the default is to retain the existing setting

Parameters

NUMBER = *scalars*	Numbers associated with the pens
COLOUR = *texts* or *scalars*	Colour to use with each pen unless otherwise specified by the CSYMBOL, CLINE, CFILL or CAREA parameters
LINESTYLE = *scalars*	Style for line used by each pen when joining points
METHOD = *string tokens*	Method for determining line (point, line, monotonic, closed, open, fill, spline, polygon); default poin
SYMBOL = *texts*, *scalars*, *pointers* or *matrices*	Defines the plotting symbol for each pen, by a text or scalar for a pre-defined symbol, a pointer for a userdefined symbol, or a matrix to supply a bitmap
LABELS = *texts* or *factors*	Define labels that will be printed alongside the plotting symbols
ROTATION = *scalars* or *variates*	Rotation required for the plotting symbols and labels (in degrees); default 0
JOIN = *string tokens*	Order in which points are to be joined by each pen (ascending, given); default asce
BRUSH = *scalars*	Number of the type of area filling used with each pen when drawing pie charts or histograms
FONT = *scalars*	Font to be used for any text written by each pen; default 1
THICKNESS = *scalars*	Thickness with which any lines are drawn by each pen; default 1
SIZEMULTIPLIER = *scalars* or *variates*	Multiplier used in the calculation of the size in which to draw characters and symbols by each pen; default 1
CSYMBOL = *texts* or *scalars*	Colour to use with each pen when drawing symbols
CLINE = *texts* or *scalars*	Colour to use with each pen when drawing lines
CFILL = *texts* or *scalars*	Colour to use with each pen when filling areas inside hollow symbols
CAREA = *texts* or *scalars*	Colour to use with each pen when filling areas inside polygons and bars of histograms
SMSYMBOL = *scalars* or *variates*	Multiplier used in the calculation of the size in which to draw symbols by each pen
SMLABEL = *scalars* or *variates*	Multiplier used in the calculation of the size in which to draw labels by each pen
DFSPLINE = *scalars*	Number of degrees of freedom to use when METHOD=spline; default 4
YMISSING = *string token*	How to treat missing y-values when METHOD=spline (break, interpolate); default brea
XMISSING = *string token*	How to treat missing x-values when METHOD=spline (break, ignore);

	default brea
YLPOSITION = *string token*	How to position labels in the y-direction with respect to the points (above, centre, below, automatic); default auto
XLPOSITION = *string token*	How to position labels in the x-direction with respect to the points (left, centre, right, automatic); default auto
YLSIZE = *scalars* or *variates*	Sizes of the y-direction of the text boxes into which to plot labels; default 1
XLSIZE = *scalars* or *variates*	Sizes of the x-direction of the text boxes; default 1
YLOFFSET = *scalars* or *variates*	Offsets in the y-direction of the text boxes
XLOFFSET = *scalars* or *variates*	Offsets in the x-direction of the text boxes
BARTHICKNESS = *scalars*	Thickness with which any error bars are drawn by each pen; default 1
BARCAPWIDTH = *scalars*	Width of the cap drawn by each pen at the top and bottom of any error bars; default 1
DESCRIPTION = *texts*	Description for points plotted by the pen, to be used by the Data Information tool in the Graphics Viewer
TSYMBOL = *scalars*	Defines the transparency of symbols drawn by each pen, on a scale of 0 (opaque) to 255 (completely transparent); default 0
TLINE = *scalars*	Defines the transparency of lines drawn by each pen; default 0
TFILL = *scalars*	Defines the transparency to use when filling areas inside hollow symbols with each pen; default 0
TAREA = *scalars*	Defines the transparency to use when filling areas inside polygons and bars of histograms with each pen; default 0
SAVE = *pointers*	Saves details of the current settings for the pen concerned

Graphical displays are drawn using graphical *pens*. Certain pens are used by default, or you can specify other pens, as described in the preceding sections. The attributes of each pen, such as colour and symbol-type, determine how they are used to generate output. The initial defaults for each pen are device-specific, and are described at the end of this subsection. The PEN directive can be used to change these attributes so that you can modify the resulting display. Different attributes are relevant for different types of output, **for example symbols and labels are used only within DGRAPH and D3GRAPH** (and the graphics procedures that use them to construct their plots).

The NUMBER parameter lists the numbers of the pens, in the range 1 to 256 or -1 to -12, that you wish to redefine. By default, any aspects of these pens that are not set explicitly retain the values that they had immediately before the PEN statement. Alternatively, you can specify option RESET=yes to reset their definitions to the default values defined by GenStat at the start of each job.

Pens 1 to 256 are used for the information that is plotted in a graph (points, lines, and so on). In most of the graphics commands, the default is to use these pens in succession for the different structures that are plotted, so that the various data sets can easily be distinguished. **The negatively numbered pens** are used as the initial defaults for the axes and their associated marks and labels (see XAXIS; 4.9.4), and for gridlines, the overall title and the key (see FRAME; 4.9.3.1), or for default gridlines in shade plots (see DSHADE; 4.4.2), or for default outlines in histograms (see DHISTOGRAM; 4.3.1), bar charts (see BARCHART; 4.3.2) and pie charts (see DPIE; 4.6.1), or for error bars (see BARCHART; 4.3.2), or for the overall title (see DSTART; 4.8.2). They cannot be used for any other purposes. **See the table in Appendix 1 for an overview**.

The COLOUR, CSYMBOL, CLINE, CFILL and CAREA parameters specify the colours to be used by the pen. The COLOUR parameter can be used to define the colour for anything plotted by the pen, while the other parameters define specific aspects (overriding any setting of COLOUR): CSYMBOL defines the colour to be used for drawing symbols, CLINE defines the colour for lines, CFILL defines the colour for filling areas, and CAREA defines the colour for filling areas inside polygons and bars of histograms. The parameters can be set either to a text containing the name of one of GenStat's pre-defined colours (4.9.9), or to a scalar containing a number defining a colour using the RGB system, or to a hexadecimal digit defined in a string of the form '#abc', '0xabc' or '0Xabc' where abc is the hexadecimal digit. You can use the RGB function to construct these colour numbers from their red, green and blue components: for example

```
      CALCULATE xgold = RGB(255; 215; 0)
      PEN       2; CSYMBOL=xgold
```

sets xgold to the colour gold (which has red, green and blue values 255, 215 and 0 respectively) and uses this as the colour for symbols drawn in future by pen 2. The numbers give you access to the complete spectrum supported by most colour graphics devices. (Note, though, that they will automatically be mapped onto a grey scale if the device is defined with a grey-scale palette; see DEVICE). Alternatively, the pre-defined colours define the standard colours used by many web browsers, and mainly use the same names. The names, and their corresponding red, green and blue values, are listed in 4.9.9. **They can be given in either upper- or lower-case, or in any mixture, but they must not be abbreviated.**

The TSYMBOL, TLINE, TFILL and TAREA parameters accompany the parameters CSYMBOL, CLINE, CFILL and CAREA, respectively, and define the transparency of the corresponding colours. Their values are on a scale of 0 (opaque) to 255 (completely transparent). The pens have initial defaults of 0.

The SYMBOL parameter defines the symbol that is drawn at each point, for example by DGRAPH. You can mark different points with different symbols (for example to indicate groupings in the data) by setting the PEN parameter of DGRAPH to a variate or factor specifying a pen with the appropriate symbol for each point.

GenStat provides a choice of standard symbols that can be specified either by giving the name (in a text with a single value), or the number (in a scalar). See Figure 4.9.8a and the list below.

1 Cross
2 Circle
3 Plus
4 Star
5 Square
6 Diamond
7 Triangle
8 Nabla
9 Asterisk
10 Minus
11 Heavyminus
12 Heavyplus
13 Heavycross
14 Smallcircle
15 Tinycircle
16 Female
17 Male
18 Rhombus
19 Circlecross
20 Circleplus
21 Squarecross
22 Squareplus
-1 Sphere
-2 Cone
-3 Cylinder
-4 Cube

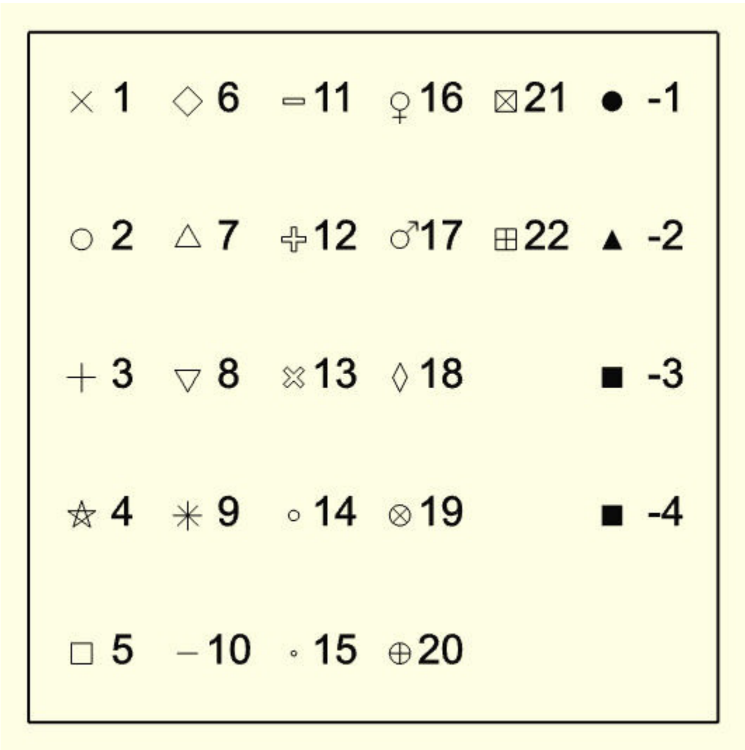


Figure 4.9.8a

Example 4.9.8a demonstrates the GenStat program which produced this figure. To get the LightYellow background you have to specify the BACKGROUND setting of FRAME.

Example 4.9.8a

```
FRAME       3 ; BACKGROUND= 'LightYellow'
VARIATE     [VALUES=5(1...4),2(5),4(6)] x
VARIATE     [VALUES=(5...1)4,5,4,5,4,3,2] y
```

```
VARIATE     [VALUES=1...22,-1,-2,-3,-4] vlabels
TXCONSTRUCT [TEXT=labels] ' ',vlabels
PEN         1...26 ; SYMBOLS=1...22,-1,-2,-3,-4 ; COLOUR=1 ; LABELS=#labels ;\
            SIZE=1.5
XAXIS       3 ; ACTION=hide ; UPPER=6.5
YAXIS       3 ; ACTION=hide
DGRAPH      [WINDOW=3 ; KEYWINDOW=0] Y=#y ; X=#x ; PEN=1...26
```

The final four symbols (numbered -1 to -4) are intended mainly for 3-dimensional plots, and may not be available on some devices. **You can set SYMBOL=0 if you do not want to plot symbols at the data points**, as for example if you only want to draw a line through the points. **You can also use SYMBOL=0 together with the LABELS parameter** (described below) to plot a character at the data points instead of a symbol. For example

```
PEN 1; SYMBOL=0; LABEL='A'
```

will identify the points plotted by pen 1 with the letter A.

To define a symbol of your own, you can set SYMBOL to a pointer containing a pair of variates defining the coordinates of a set of points to be joined by straight line segments. The points should be within a notional square with bounds -1.0 to 1.0 in each direction. The square is centred on the data point, and scaled to the same size as the standard symbols. Missing values can be included in the coordimates, to use separate pen strokes to draw the line segments. The final possibility is to set SYMBOL to a matrix of RGB colour values, representing a bitmap.

User-defined symbols are illustrated in Example 4.9.8b.

Example 4.9.8b

```
FRAME    [GRID=xy] 1...4; YLOWER=0.75; YUPPER=1.0; \
         XLOWER=0.0,0.25,0.5,0.75; XUPPER=0.25,0.5,0.75,1.0; \
         YMLOWER=0.05; YMUPPER=0.01; XMLOWER=0.05; XMUPPER=0.01; PENGRID=29
PEN      29; LINESTYLE=7; COLOUR='black'
XAXIS    1,2,3,4; LOWER=-1.2,0.8; UPPER=1.2,3.2; MARKS=1
YAXIS    1,2,3,4; LOWER=-1.2,0.8; UPPER=1.2,3.2; MARKS=1
VARIATE  Diamond[1]; VALUES=!(-1,0,1,0,-1)
&        Diamond[2]; VALUES=!(0,-0.5,0,0.5,0)
PEN      1; COLOUR='black'; METHOD=line; SYMBOL=0; JOIN=given; THICKNESS=2
&        2; COLOUR='black'; SYMBOL=Diamond; SMSYMBOL=2
DGRAPH   [WINDOW=1; KEYWINDOW=0] Diamond[1]; Diamond[2]; PEN=1
&        [WINDOW=2; SCREEN=keep] 1,2,3; 1,3,2; PEN=2
VARIATE  Arrow[1]; VALUES=!(0.0,1.0,0.75,*,1.0,0.75)
&        Arrow[2]; VALUES=!(0.0,0.0,-0.25,*,0.0,0.25)
PEN      3; COLOUR='black'; SYMBOL=Arrow; SIZE=!(2,2.5,3,2);\
         ROTATION=!(0,45,90,180); SMSYMBOL=3
DGRAPH   [WINDOW=3; KEYWINDOW=0; SCREEN=keep] Arrow[1]; Arrow[2]; PEN=1
&        [WINDOW=4] !(1.0,2.7,2.0,1.6); !(1.4,1.8,2.2,2.6); PEN=3
```

The definition of the arrow symbol in lines 15 and 16 illustrates how missing values can be included so that separate pen strokes are used to draw line segments. The plot produced by this example is shown in Figure 4.9.8b.

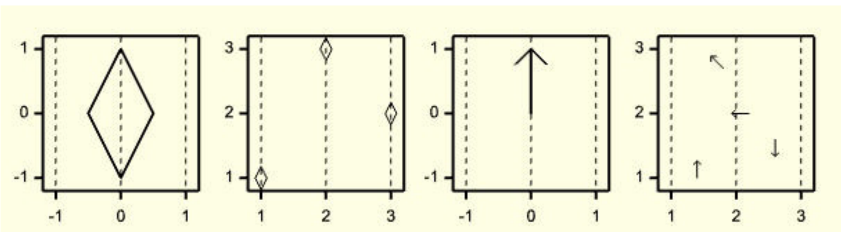


Figure 4.9.8b

You can mark different points with different symbols (for example to indicate groupings in the data) by setting the PEN parameter of DGRAPH (4.2.1) to a variate or factor specifying a pen with the appropriate symbol for each point.

You can also use the LABELS parameter to label each point with a string or a number. The LABELS parameter can be set to a single string to plot the same label at every point, or text structure with same number of values as the Y and X variates that are being plotted. Alternatively LABELS can be set to a factor;

the factor labels are then used, if available, otherwise the levels. This provides another means of representing grouped data. The positioning of the labels with respect to the points is controlled by the `YLPOSITION` and `XLPOSITION` parameters. The initial default is to determine the positions automatically according to their type (e.g. labels for points, or for tick marks on the y-axis, or on the x-axis, and so on).

The graphical symbols are drawn so that they are centred at the specified position. If `LABELS` are specified they are aligned alongside the markers, unless you have set `SYMBOLS=0` to suppress the markers, in which case the labels start from the specified (x,y) position. For compatibility with previous releases of GenStat you can also set `SYMBOLS` to a factor or text, which has the same effect as setting `LABELS` with `SYMBOLS=0`.

The GenStat Graphics Viewer with GenStat *for Windows* has a "Data Information" tool that allows you to display information about each point when you place the cursor over the point. See the explanation between figures 3.5 and 3.6. If you want to replace the default information, you can set the `DESCRIPTION` parameter to a text (with one line for each point) containing your own information.

The `METHOD` parameter specifies the type of object to be plotted: points, lines or filled polygons. The initial default for every pen, `METHOD=point`, will result in points being plotted using the corresponding symbols, labels, colours and fonts. Various types of line can be drawn through the plotted points; either straight lines (`line` and `polygon`) or smooth curves (`monotonic`, `open`, `closed` and `spline`). **The `line` and `polygon` settings differ in that, with `polygon`, a line is drawn also to connect the first and last points**. The `monotonic` setting specifies that a smooth single-valued curve is to be drawn through the data points. The name is derived from the requirement that the x-values (rather than the fitted curve) must be strictly monotonic, so that there is only one y-value for each distinct x-value. To ensure this, a copy of the data is made and sorted before the curve is fitted. This setting is recommended for plotting curves fitted to data, for example with `FITCURVE`. You should ensure that the points are close enough for the plotted line to be a reasonable approximation. When you know the functional form of the curve, it may be advantageous to calculate extra points. The `open` and `closed` settings specify that a smooth, possibly multi-valued, curve is to be drawn through the data points, using the method of McConalogue (1970); the resulting curve is rotationally invariant, although it is not invariant under scaling. The `closed` setting connects the last point to the first. McConalogue's method (`open` or `closed`) is more suited to the situation where the plotted curve is intended to represent the shape of an object. Alternatively, the `spline` setting plots a smoothing spline fitted through the points. The `DFSPLINE` parameter specifies how many degrees of freedom to use in the spline (initial default 4). The `YMISSING` parameter controls whether to break the spline at a missing y-value or to interpolate y-value, and the `XMISSING` parameter controls whether to break the spline at a missing x-value or to ignore the point; the initial default for both parameters is to break the spline. The setting `METHOD=fill` joins the data points by straight lines to produce one or more polygons. Each polygon is then shaded in the style specified by BRUSH (see below). The plotting method also determines how contours will be drawn, as described in 4.4.1. Also, the combination of `SYMBOLS=0` and `METHOD=point` will produce no plotting at all (and no warning) within `DGRAPH`.

If the requested plotting method produces a line through the points, the `LINESTYLE` parameter will specify what sort of line is drawn (for example a solid, dotted or dashed line). The type of line style is denoted by a number in the range 1 up to 10. Figure 4.9.8c illustrates some of the line styles available. The exact appearance of the different line styles is device-specific, and there are not necessarily 10 different line styles available on a particular device, but line style 1 should always produce a solid line.

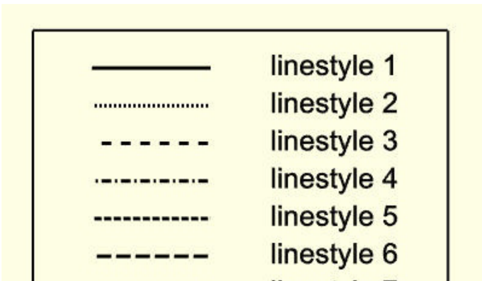


Figure 4.9.8c

The `JOIN` parameter controls the order in which points are connected when lines are to be drawn or the points define a polygon to be shaded. `Given` requests that the data are to be plotted in the order in which they are stored, whereas `ascending` implies that the data are copied and sorted so that the x-values are in ascending order before plotting. This parameter is ignored when `METHOD=monotonic`, as this requires that the data must always be sorted.

The BRUSH parameter is used on some **monochrome devices** to controls how areas are shaded when METHOD is set to fill, or when plotting histograms and pie charts. There are 16 available patterns indicated by the integers 1 to 16, as shown in Figure 6.9.8d of Reference 2 (see the Introduction). In general, the higher the number, the denser the hatching, and the longer such areas take to plot. The device-specific brush styles are generally faster, and produce smaller output files; however results are not guaranteed to be the same on every type of device. The CFILL parameter defines which colour is used by the pen to fill the areas.

The THICKNESS parameter allows you to specify an amount by which the standard thickness of plotted lines is to be multiplied. This allows you to increase the thickness of lines, perhaps to highlight some feature of a plot, as illustrated in the contour plot in Figure 4.4.1b. You can also use thickness to emphasize the axes, by redefining the appropriate pen. For some devices, it is not possible to control the thickness of plotted lines; the THICKNESS parameter is then ignored.

The default size of characters and symbols is determined from the dimensions of the current window. The SIZEMULTIPLIER parameter can be used to modify the sizes of both of these, by specifying a value by which this default size is to be multiplied. Alternatively, you can use the SMSYMBOL parameter to modify just the symbol size, or the SMLABEL parameter to modify just the size of characters in labels. For example when plotting a graph in a small window you may wish to increase the size of annotation in order to make it legible. They can each be set to a scalar, or to a variate to allow the different points to be scaled in different ways.

The ROTATION parameter controls the angle (in degrees) at which to plot text or user-defined symbols. The initial setting of zero will produce text "conventionally" orientated. You can set ROTATION to a scalar value that will apply to all points, or to a variate that allows a different angle to be used at each point. ROTATION is used in the last line of Example 4.9.8b to plot a userdefined symbol at different angles.

In most implementations, including GenStat *for Windows*, the FONT parameter can be set to an integer between 1 and 25 to select different fonts for text appearing as titles, axis annotation, plotting symbols and key information. The initial default for each pen is font 1, which is the default font. You can list the other fonts allocations using the DHELP procedure (4.9). If one of these is unavailable on your computer, the default font is used instead. The FONT parameter also works with the following allocations:

CAPITAL LETTERS

FONT 01	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
FONT 02	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
FONT 03	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
FONT 04	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
FONT 05	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
FONT 06	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
FONT 07	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
FONT 08	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
FONT 09	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
FONT 10	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
FONT 11	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
FONT 12	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
FONT 13	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
FONT 14	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
FONT 15	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
FONT 16	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
FONT 17	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
FONT 18	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
FONT 19	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
FONT 20	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
FONT 21	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
FONT 22	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
FONT 23	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
FONT 24	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
FONT 25	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

Font	Mapping						
01	Default font (see Tools|Options|Fonts in Graphics Viewer)						
02	Arial	08	Comic Sans MS	14	Impact	20	Sylfaen
03	Arial Black	09	Courier New	15	Lucida Console	21	Tahoma
04	Arial Narrow	10	Franklin Gothic Medi	16	Lucida Sans Unicode	22	Times New Roman
05	Book Antiqua	11	Garamond	17	Microsoft Sans Serif	23	Trebuchet MS
06	Bookman Old Style	12	Georgia	18	Monotype Corsiva	24	Verdana
07	Century Gothic	13	Haettenschweiler	19	Palatino Linotype	25	Arial

LOWER CASE LETTERS

FONT 01 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
FONT 02 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
FONT 03 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
FONT 04 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
FONT 05 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
FONT 06 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
FONT 07 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
FONT 08 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
FONT 09 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
FONT 10 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
FONT 11 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
FONT 12 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
FONT 13 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
FONT 14 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
FONT 15 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
FONT 16 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
FONT 17 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
FONT 18 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
FONT 19 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
FONT 20 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
FONT 21 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
FONT 22 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
FONT 23 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
FONT 24 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
FONT 25 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

The current settings of each pen can be saved in a pointer supplied by the SAVE parameter. The elements of the pointer are labelled to identify the components, as shown in Example 4.9.8c.

Example 4.9.8c

```
 2  PEN    8; LABELS='observation'; SYMBOL=8; JOIN=given; SAVE=Pen8
 3  PRINT [RLWIDTH=19; ORIENTATION=across] Pen8[]; FIELDWIDTH=18
        Pen8['boxunits']          characters
          Pen8['colour']            16747520
       Pen8['linestyle']                   *
          Pen8['method']               point
          Pen8['symbol']                   8
          Pen8['labels']         observation
        Pen8['rotation']                0.00
            Pen8['join']               given
           Pen8['brush']                   *
            Pen8['font']                   1
       Pen8['thickness']                1.00
            Pen8['size']                1.00
         Pen8['csymbol']            16747520
           Pen8['cline']            16747520
           Pen8['cfill']                  -1
           Pen8['carea']            16747520
        Pen8['smsymbol']                1.00
         Pen8['smlabel']                1.00
        Pen8['dfspline']                4.00
```

Pen8['ymissing']	break
Pen8['xmissing']	break
Pen8['ylposition']	automatic
Pen8['xlposition']	automatic
Pen8['barthickness']	1.00
Pen8['barcapwidth']	1.00
Pen8['tsymbol']	0.00
Pen8['tline']	0.00
Pen8['tfill']	0.00
Pen8['tarea']	0.00
Pen8['fontname']	***

Note that the saved values for line style and brush style are missing values. This is how the initial default settings are represented; the actual values used for these attributes when plotting will depend on the output device, unless they are set explicitly (as with SYMBOL in this example).

The standard text fonts, graphical symbols and brush styles are software generated. However, you can set negative values for these parameters of the PEN directive to select device-specific alternatives. For each parameter, the device-specific settings have the same range as the standard settings; thus you can select symbols -1 to -9, fonts -1 to -25, and brush styles -1 to -16. If fewer device-specific settings are actually available, the settings are taken in turn, and then recycled. Where a feature has no device-specific settings on a particular device, the standard form is used instead (for example, font -3 appearing as font 3). **Device-specific font numbers cannot be used within the in-line typesetting system**; GenStat will use either the standard fonts or the corresponding device-specific fonts depending on the base font originally specified by the PEN directive. In some cases, device-specific symbols or fonts may be of fixed size; the SIZEMULTIPLIER or SMLABEL parameters will then have no effect, and some of the typesetting commands may not function correctly. The help information should indicate when device-specific fonts are of fixed size (see 4.9.1.1). Although the device-specific settings are likely to be different from device to device, they are arranged to be consistent where possible, so that for example brush style -1 will select solid fill, if available.

By default GenStat uses software generated symbols and fonts. You can list the initial defaults for the colours using the GETRGB procedure, as explained in 4.9.9.1. On a grey-scale device, the colours are mapped automatically to shades of grey, while on a monochrome device all colours except white are plotted as black. By default, on colour and grey-scale devices, symbol 1 is used for all pens. On monochrome displays, the default is to use symbols 1 to 22 in turn: symbol 1 for pen 1, symbol 2 for pen 2, and so on. When solid fill and colour (or grey-scale) are available, the default brush style is -1, in different colours for each pen. Otherwise, by default, the pens use different software-generated brushes.

4.9.9 Colours

In the 11th edition changes have been made to the way that colours are specified in GenStat graphics. Unfortunately it was impossible to do this without having some affect on existing GenStat programs. Any programs that set colour attributes or used the `COLOUR` directive may be affected, although graphics generated by library procedures should be OK as these have all been updated. Any new programs will need to be coded differently and this will prevent them working in older versions of GenStat.

The `COLOUR` directive is no longer used. In other directives colour numbers between 0 and 255 are treated as RGB values, effectively all shades of blue between black and full blue.

There should be no noticeable difference in programs where graphics are drawn without reference to the colour parameters, i.e. using the defaults. Where colours are specified the effect will be to lose all or most of the colour, as the display will be drawn in very dark shades of blue that are effectively black when viewed on a PC.

Where great inconvenience results from this change the `SET` command (option `CMETHOD`) can be used to request old-style interpretation of colour numbers, using a standard lookup. In this case the `COLOUR` directive will function as before and values in the range 0 - 255 will refer to colours defined in this way. Thus old code should run more or less as before. The new colour values and string representations will also be recognised under this setting, with the exception that values between 0 and 255 (i.e. pure shades of blue) will be treated as old-style colour numbers.

Colours in GenStat graphs are specified in terms of red, green and blue components each of which can take any integer value in the range 0 – 255. These are combined to form a single RGB value which represents any one of 16,277,716 individual possibilities. The following table lists some common combinations with pre-defined names which will be suitable for most applications; these names are used by the graphics menus and can also be used as colour values in graphics directives such as `PEN`. The names listed here are those designated for the X Window System and are also recognised by most or all web browsers. The physical representation of any particular RGB value will depend on the quality of the display device and some colours may be indistinguishable in practice. However, this table should help with selection of suitable combinations; please scroll down to see the full range of colours.
The names of the standard pre-defined GenStat colours are listed below with their corresponding red, green and blue values for use e.g. in the RGB function.

Colour Name	R	G	B
Red Colours			
IndianRed	205	92	92
LightCoral	240	128	128
Salmon	250	128	114
DarkSalmon	233	150	122
LightSalmon	255	160	122
Crimson	220	20	60
Red	255	0	0
FireBrick	178	34	34
DarkRed	139	0	0
Pink Colours			
Pink	255	192	203
LightPink	255	182	193

Colour Name	R	G	B
Green Colours			
GreenYellow	173	255	47
Chartreuse	127	255	0
LawnGreen	124	252	0
Lime	0	255	0
LimeGreen	50	205	50
PaleGreen	152	251	152
LightGreen	144	238	144
MediumSpringGree	0	250	154
SpringGreen	0	255	127
MediumSeaGreen	60	179	113
SeaGreen	46	139	87
ForestGreen	34	139	34

HotPink	255	105	180
DeepPink	255	20	147
MediumVioletRed	199	21	133
PaleVioletRed	219	112	147

Orange Colours

LightSalmon	255	160	122
Coral	255	127	80
Tomato	255	99	71
OrangeRed	255	69	0
DarkOrange	255	140	0
Orange	255	165	0

Yellow Colours

Gold	255	215	0
Yellow	255	255	0
LightYellow	255	255	224
LemonChiffon	255	250	205
LightGoldenrodYellow	250	250	210
PapayaWhip	255	239	213
Moccasin	255	228	181
PeachPuff	255	218	185
PaleGoldenrod	238	232	170
Khaki	240	230	140

Purple Colours

Lavender	230	230	250
Thistle	216	191	216
Plum	221	160	221
Violet	238	130	238
Orchid	218	112	214
Fuchsia	255	0	255
MediumOrchid	186	85	211
MediumPurple	147	112	219
BlueViolet	138	43	226

Green	0	128	0
DarkGreen	0	100	0
YellowGreen	154	205	50
OliveDrab	107	142	35
Olive	128	128	0
DarkOliveGreen	85	107	47
MediumAquamarine	102	205	170
DarkSeaGreen	143	188	143
LightSeaGreen	32	178	170
DarkCyan	0	139	139
Teal	0	128	128

Blue Colours

Aqua	0	255	255
Cyan	0	255	255
LightCyan	224	255	255
PaleTurquoise	175	238	238
Aquamarine	127	255	212
Turquoise	64	224	208
MediumTurquoise	72	209	204
DarkTurquoise	0	206	209
CadetBlue	95	158	160
SteelBlue	70	130	180
LightSteelBlue	176	196	222
PurwaBlue	155	225	255
PowderBlue	176	224	230
LightBlue	173	216	230
SkyBlue	135	206	235
LightSkyBlue	135	206	250
DeepSkyBlue	0	191	255
DodgerBlue	30	144	255
CornflowerBlue	100	149	237
RoyalBlue	65	105	225

DarkViolet	148	0	211
DarkOrchid	153	50	204
DarkMagenta	139	0	139
Purple	128	0	128
Indigo	75	0	130
SlateBlue	106	90	205
DarkSlateBlue	72	61	139
MediumSlateBlue	123	104	238

Brown Colours

Cornsilk	255	248	220
BlanchedAlmond	255	235	205
Bisque	255	228	196
NavajoWhite	255	222	173
Wheat	245	222	179
BurlyWood	222	184	135
Tan	210	180	140
RosyBrown	188	143	143
SandyBrown	244	164	96
Goldenrod	218	165	32
DarkGoldenrod	184	134	11
Peru	205	133	63
Chocolate	210	105	30
SaddleBrown	139	69	19
Sienna	160	82	45
Brown	165	42	42
Maroon	128	0	0

Blue	0	0	255
MediumBlue	0	0	205
DarkBlue	0	0	139
Navy	0	0	128
MidnightBlue	25	25	112

White Colours

White	255	255	255
Snow	255	250	250
Honeydew	240	255	240
MintCream	245	255	250
Azure	240	255	255
AliceBlue	240	248	255
GhostWhite	248	248	255
WhiteSmoke	245	245	245
Seashell	255	245	238
Beige	245	245	220
OldLace	253	245	230
FloralWhite	255	250	240
Ivory	255	255	240
AntiqueWhite	250	235	215
Linen	250	240	230
LavenderBlush	255	240	245
MistyRose	255	228	225

Grey Colours

Gainsboro	220	220	220
LightGrey	211	211	211
Silver	192	192	192
DarkGray	169	169	169
Gray	128	128	128
DimGray	105	105	105
LightSlateGray	119	136	153
SlateGray	112	128	144
DarkSlateGray	47	79	79
Black	0	0	0

In addition the **string 'Background'** can be used to refer to the background colour that has been defined (e.g. by FRAME) for the particular part of the screen where the pen is being used. Another useful setting is the **string 'Transparent'**. This is used as the initial default for the background colours of the graphics windows.

Tip

Instead of these standard colours the complete windows colour palette can be found with the Graphics Viewer. This is explained below.

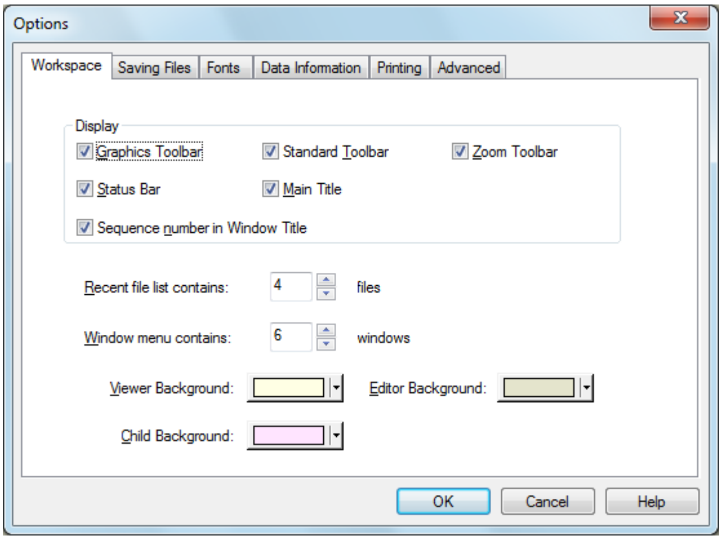


Figure 4.9.1

Open the Graphics Viewer with the button on the Toolbar of GenStat. Click on Tools on the Toolbar of the Graphics Viewer and then on Options... Click on the Nabla symbol to the right of the Viewer Background box in the Workspace Tab (see Figure 4.9.1).

Then click on the Other... button in Figure 4.9.2.

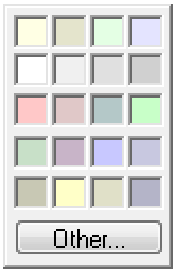


Figure 4.9.2

This opens the Colour Palette shown in Figure 4.9.3. To select a colour from the Colour Palette you must use the following two steps:

Click on the colour in the spectrum that you want to use (shown by the hollow cross).

Select the intensity from the slider on the right hand side (shown by the small triangle).

The colour is shown in the Color|Solid-box.

The Red, Green and Blue values of the chosen colour are shown in the boxes at the right. From these values the RGB value can be calculated with the RGB function (4.9.9.3).

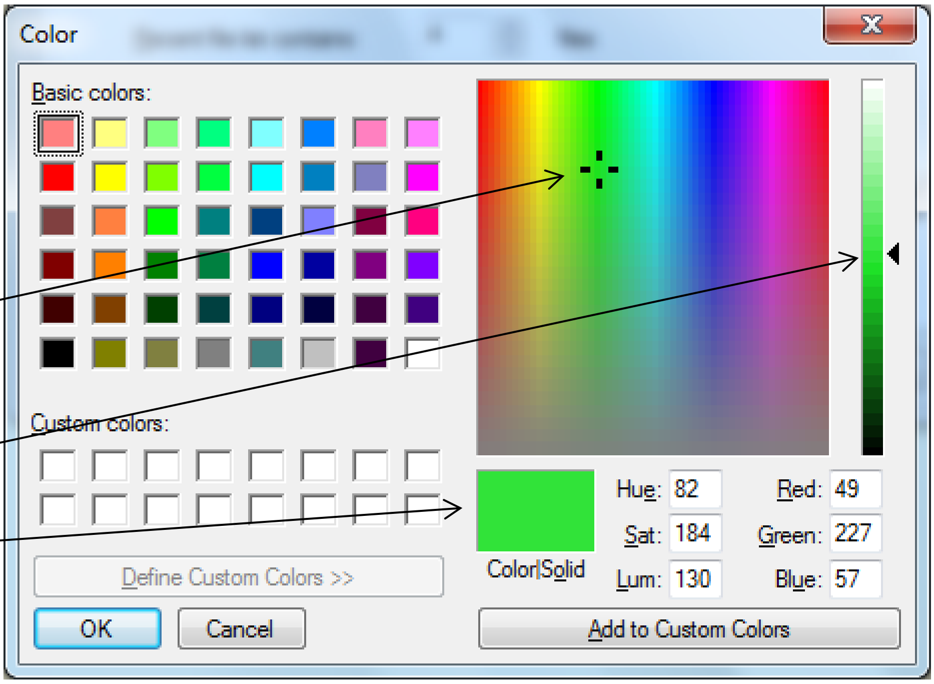


Figure 4.9.3

The initial defaults for the colours of the pens (4.9.8) defines a standard sequence of colours, that is also used to set default colours in procedures like AGRAPH. This can be accessed using the GETRGB procedure.

4.9.9.1 The GETRGB procedure

GETRGB procedure

Gets the RGB values and names of the initial default graphics colours of the GenStat pens (R.W. Payne).

No options

Parameters

COLOUR = *scalars* or *variates* — Colour numbers

RGB = *scalars* or *variates* — RGB values

NAME = *texts* — Names of nearest colours

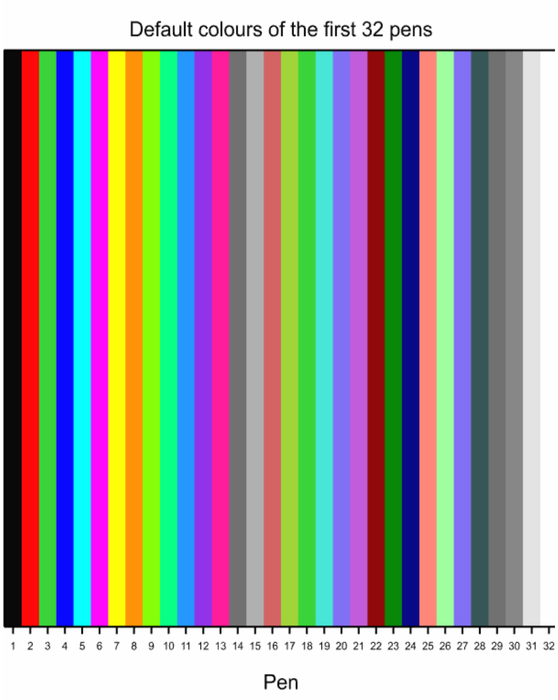
The COLOUR parameter specifies a scalar or variate containing the pen number(s) whose initial default colours are required. The RGB parameter saves a scalar or variate containing the corresponding colours, expressed as RGB values (see PEN). The NAME parameter saves a text containing the name of the nearest colour.

Example 4.9.9.1 uses GETRGB to display the first 32 colours in the sequence.

Example 4.9.9.1

```
VARIATE     [VALUES=1...32] Ncolour
GETRGB      Ncolour; RGB=RGB; NAME=Name
CALCULATE   Red = RED(RGB)
CALCULATE   Green = GREEN(RGB)
CALCULATE   Blue = BLUE(RGB)
PRINT       Ncolour,Name,RGB,Red,Green,Blue; DECIMALS=0;\
            FIELD=8,14,9,4,5,5
```

Ncolour	Name	RGB	Red	Green	Blue
1	Black	0	0	0	0
2	Red	16711680	255	0	0
3	LimeGreen	3329330	50	205	50
4	Blue	255	0	0	255
5	Aqua	65535	0	255	255
6	Fuchsia	16711935	255	0	255
7	Yellow	16776960	255	255	0
8	DarkOrange	16747520	255	140	0
9	Chartreuse	8388352	127	255	0
10	SpringGreen	65407	0	255	127
11	DodgerBlue	2003199	30	144	255
12	BlueViolet	9055202	138	43	226
13	DeepPink	16716947	255	20	147
14	DimGray	6908265	105	105	105
15	DarkGray	11119017	169	169	169
16	IndianRed	13458524	205	92	92
17	YellowGreen	10145074	154	205	50
18	LimeGreen	3329330	50	205	50
19	Turquoise	4251856	64	224	208
20	MediumSlateBlue	8087790	123	104	238
21	MediumOrchid	12211667	186	85	211
22	DarkRed	9109504	139	0	0
23	Green	32768	0	128	0
24	Navy	128	0	0	128
25	Salmon	16416882	250	128	114
26	PaleGreen	10025880	152	251	152
27	MediumSlateBlue	8087790	123	104	238
28	DarkSlateGray	3100495	47	79	79
29	DimGray	6908265	105	105	105



Note: Pen 3 and 18 have the same colour so as Pen 14 and 29, and Pen 20 and 27

30	Gray	8421504	128	128	128
31	Gainsboro	14474460	220	220	220
32	White	16777215	255	255	255

```
 7  MATRIX      [ROWS=1 ; COLUMNS=NVALUES(RGB)] mat ; VALUES=RGB
 8  YAXIS       3 ; MARKS=!(-1) ; LOWER=1 ; UPPER=50
 9  XAXIS       3 ; TITLE= 'Pen' ; MARKS=1
10  DBITMAP     [WINDOW=3 ; TITLE= 'Default colours of the first 32 pens'] mat
```

Alternatively, you can define your own sequences of colours using the DCOLOURS procedure.

4.9.9.2 The DCOLOURS procedure

DCOLOURS procedure

Forms a band of graduated colours for graphics (P.W. Goedhart).

Options

METHOD = *string token*	Type of colour band required (spectral, blackbody, linear); default line
PLOT = *string token*	What to plot (testgraph); default *

Parameters

START = *scalar* or *text*	Start value for the colour band; default * gives an appropriate default for the METHOD concerned
END = *scalar*, *text* or *variate*	End value(s) for the colour band; default * gives an appropriate default for the METHOD concerned
GAMMA = *scalar* or *variate*	The gamma-correction exponent(s) for the colour band; default 1
NCOLOURS = *scalar* or *variate*	Number(s) of colours in the colour band; default 20
RGB = *variates*	Saves the RGB colour values of each colour band
RED = *variates*	Saves the red component of the RGB colour values
GREEN = *variates*	Saves the green component of the RGB colour values
BLUE = *variates*	Saves the blue component of the RGB colour values
TITLE = *text*	General title for each test graph; default forms an informative title automatically
WINDOW = *scalar*	Window number for each test graph; default 1
SCREEN = *string token*	Whether to clear the screen before plotting each test graph or to continue plotting on the old screen (clear, keep); default clea

Procedure DCOLOURS creates a colour band by interpolating between start and end colour values. You can save the RGB colours of the band, in a variate, using the RGB parameter. Alternatively, you can save the red, green and blue components of the colours using the RED, GREEN and BLUE parameters (again in variates).

A test graph displaying the colour band can be requested by setting option PLOT=testgraph. The WINDOW parameter supplies the window number for the plot (default 1). The TITLE parameter can supply a title for the test graph; if this is not set, a suitable title is generated automatically. You can set parameter SCREEN=keep to plot the test graph on an existing screen; by default the screen is cleared first.

The METHOD option provides a choice of three different types of colour band. The default, METHOD=linear, forms the colours by interpolating between start and end RGB values. The start value is specified by the START parameter, as either a scalar defining an RGB colour value, or a text containing the name of one of the pre-defined GenStat colours (see the PEN directive for the available names, or search for "Graphics Colours" in the on-line help). You can set the END parameter to a single scalar or text (giving either the RGB value or the name of the colour) to define the band as a single sequence of colours. Alternatively, you can define a variate or a text with several values to form the band from several sequences of colours. At each END colour, DCOLOURS then begins a new sequence running from that colour to the next END colour. The default values for START and END are 'white' and 'black'.

Setting METHOD=spectral forms an approximate rainbow spectrum for wavelengths between 380 nm and 780 nm. There can now be only a single sequence of colours. The START and END parameters specify the start and end wavelengths, as scalars, with default values of 380 and 780.

The final setting, METHOD=blackbody, forms colours of hot objects with temperatures between 500 K and 11000 K. Again, only a single sequence of colours is allowed. The START and END parameters specify the start and end temperatures, as scalars, with default values of 500 and 11000.

The NCOLOURS parameter specifies the number of colours in each sequence of colours, as a scalar for the spectral or blackbody methods, or as either a scalar or a variate for the linear method; the default is 20.

The red, green and blue values in each sequence are assumed by default to vary linearly with wavelength, temperature or red/green/blue components. Alternatively, you can use the GAMMA parameter to specify the power for a power transformation (default 1). It must be set to a scalar for the spectral or blackbody methods, and to either a scalar or a variate for the linear method. Its values must lie in the interval [0.25, 4].

The number of values specified by each set of END, GAMMA and NCOLOURS parameters can be different. However, the number of values in the setting of the END parameter determines the number of colour sequences in the band, and the values in the GAMMA setting and NCOLOURS setting are recycled as required.

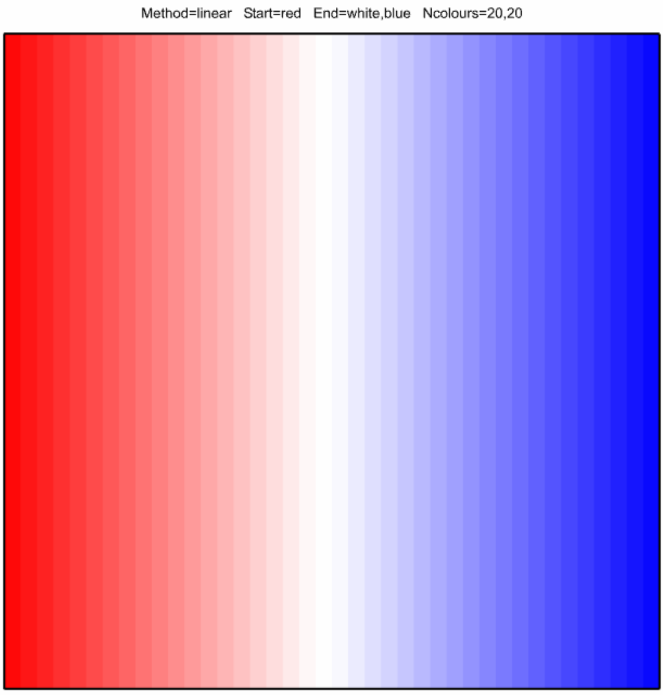


Figure 4.9.9.2

Example 4.9.9.2 uses DCOLOURS to form a sequence of colours running from red to white and then to blue. The colour map is shown in Figure 4.9.9.2.

Because the default value for NCOLOURS equals 20 and the END parameter is a text of length 2, the number of formed colours equals 40.

Example 4.9.9.2

```
DCOLOURS [METHOD=linear; PLOT=testgraph] START='red'; END=!t(white,blue);\
         WINDOW=3; RGB=RGBseq ; RED=red ; GREEN=green ; BLUE=blue
PRINT    !(1...40),RGBseq, red, green, blue ; DECIMALS=0
```

	RGBseq	red	green	blue
1	16711680	255	0	0
2	16715021	255	13	13
3	16718362	255	26	26
4	16721960	255	40	40
5	16725301	255	53	53
6	16728899	255	67	67
7	16732240	255	80	80
8	16735581	255	93	93
9	16739179	255	107	107
10	16742520	255	120	120
11	16746118	255	134	134
12	16749459	255	147	147
13	16753057	255	161	161
14	16756398	255	174	174
15	16759739	255	187	187

16	16763337	255	201	201
17	16766678	255	214	214
18	16770276	255	228	228
19	16773617	255	241	241
20	16777215	255	255	255
21	15921919	242	242	255
22	15066623	229	229	255
23	14211327	216	216	255
24	13421823	204	204	255
25	12566527	191	191	255
26	11711231	178	178	255
27	10855935	165	165	255
28	10066431	153	153	255
29	9211135	140	140	255
30	8355839	127	127	255
31	7500543	114	114	255
32	6711039	102	102	255
33	5855743	89	89	255
34	5000447	76	76	255
35	4145151	63	63	255
36	3355647	51	51	255
37	2500351	38	38	255
38	1645055	25	25	255
39	789759	12	12	255
40	255	0	0	255

4.9.9.3 The RGB, RED, GREEN, BLUE and GRAY graphics functions

The GETRGB procedure (4.9.9.1) gets only the RGB values (and names) of **the initial default graphics colours** of the GenStat pens. But if you have found your own colour in red, green and blue values (for example according to the TIP in 4.9.9) you can calculate the corrsponding RGB value with the RGB function. The other way around, if you know the RGB value, then you can calculate the red, green and blue components of that value with respectively the RED, GREEN and BLUE functions.

`RGB(r;g;b)` and `RGB(t)`	calculate RGB colour values. If there are three arguments, `r`, g and b, these must be numerical structures specifying the red, green and blue components of the colour (which must all be in the range 0-255). If there is a single argument, `t`, this must be a text containing names of the standard GenStat colours.
`RED(x)`	calculates the red components of the RGB colour value in x.
`GREEN(x)`	calculates the green components of the RGB colour value in x.
`BLUE(x)`	calculates the blue components of the RGB colour value in x.
`GRAY(x)`	calculates RGB colour values from the values on the gray scale in x (synonym GREY).

4.9.10 The DKEEP directive: accessing details of the graphics environment

DKEEP directive

Saves information from the last plot on a particular device.

No options

Parameters

DEVICE = *scalars*	The devices for which information is required, if the scalar is undefined or contains a missing value, this returns the current device number
WINDOW = *scalars*	Window about which the information is required; default * gives information about the last window
XLOWER = *scalars*	Lower bound for the x-axis in last graph in the specified device and window

XUPPER = *scalars*	Upper bound for the x-axis in last graph in the specified device and window
YLOWER = *scalars*	Lower bound for the y-axis in last graph in the specified device and window
YUPPER = *scalars*	Upper bound for the y-axis in last graph in the specified device and window
ZLOWER = *scalars*	Lower bound for the z-axis in last graph in the specified device and window
ZUPPER = *scalars*	Upper bound for the z-axis in last graph in the specified device and window
FILE = *scalars*	Returns the value 1 or 0 to indicate whether a file is required for this device
DESCRIPTION = *texts*	Description of the device
DREAD = *scalars*	Returns the value 1 or 0 to indicate whether graphical input is possible from this device
ENDACTION = *texts*	Returns the current ENDACTION setting ('continue' or 'pause')

DKEEP provides information that can be used in general programs and procedures to control the graphical output. For the specified device you can determine whether it generates screen output or uses a file, whether graphical input is possible, a description of the device (as printed by DHELP; see the start of Section 4.9), the current ENDACTION setting, and details of the axis bounds.

The device for which the information is required is specified by the DEVICE parameter. If you specify a scalar containing a missing value, this will be set to the number of the current graphics device. You can then test whether an output file is needed and open one accordingly, as shown in Example 4.9.10a.

Example 4.9.10a

```
DEVICE 4
READ   Y,X,Y2
```

Identifier	Minimum	Mean	Maximum	Values	Missing
Y	46.46	68.11	89.95	20	0
X	0.9400	4.867	8.877	20	0
Y2	38.00	62.37	82.65	20	0

```
SCALAR Device
DKEEP  DEVICE=Device; FILE=File; DESCRIPTION=Name
PRINT  Name,Device,File
```

Name	Device	File
HPGL	4	1

```
IF File
  OPEN 'Graph.hpg'; CHANNEL=Device; FILETYPE=graphics
ENDIF
```

When writing a procedure you can find out if axes bounds have been set explicitly, using the SAVE parameter of AXES. This information may then be used when setting up the axes for other graphs. However, if the bounds were not set, but have been evaluated from the data (or if the axes have subsequently been redefined) the information in the save structure will not be of any use. The actual values used when plotting are recorded internally, for each window of each device, and can be accessed using the XLOWER, XUPPER, YLOWER, YUPPER, ZLOWER and ZUPPER, parameters of DKEEP.

Example 4.9.10b

```
DGRAPH [WINDOW=5;KEYWINDOW=7] Y; X
" Now set up window 6 to have the same bounds as window 5,
  so that Y2 is plotted on the same scale as Y."
DKEEP  Device; WINDOW=5; YLOWER=Ymin; YUPPER=Ymax; XLOWER=Xmin; \
       XUPPER=Xmax
PRINT  Ymin,Ymax,Xmin,Xmax
```

Ymin	Ymax	Xmin	Xmax
44.29	92.12	0.5431	9.274

```
XAXIS  6; LOWER=Xmin; UPPER=Xmax
YAXIS  6; LOWER=Ymin; UPPER=Ymax
DGRAPH [WINDOW=6; KEYWINDOW=8; SCREEN=keep] Y2; X
CLOSE  Device; FILETYPE=graphics
```

4.9.11 The DSAVE and DLOAD directives: storing and recovering the graphics environment

Once you have defined the graphics environment for a particular type of plot, you may want to save it for use with that type of plot in the future. The DSTORE allows you to save the current graphics environment settings in an external file, and the DLOAD directive allows you to read them back into GenStat.

DSAVE directive

Saves the current graphics environment settings to an external file.

No options

Parameters

FILENAME = *text*	File in which to save the environment settings
DESCRIPTION = *text*	Description for these settings

DLOAD directive

Loads the graphics environment settings from an external file.

No options

Parameter

text	File from which to lead the environment settings

4.10 Line-printer graphics

Prior to Release 10.1 there were three directives for line-printer output: GRAPH, HISTOGRAM and CONTOUR. In Release 10.1, these were given a prefix LP for clarity, to become LPGRAPH (4.10.1), LPHISTOGRAM (4.10.2) and LPCONTOUR (4.10.3). The original names GRAPH, HISTOGRAM and CONTOUR are currently retained as synonyms, but they may be phased-out or used for highresolution plots in future releases.

The directives have options and parameters to modify the annotation, the symbols used, the size of plot, and so on. Several options apply generally to all three directives and are described now. Others are more specific and are left until the descriptions of the relevant directives.

Normally, output goes to the current output channel, but you can use the CHANNEL option to direct it to another . For example, when you are working interactively, you might want to send a graph to a secondary output file so that you can print it later. Unlike some directives (for example, PRINT) you cannot save the output in a text structure.

The TITLE option lets you set an overall title for the output; graphs and contour plots can also have individual axis titles, specified by the YTITLE and XTITLE options. You can supply the text settings of these options directly, in a string, or give them as the identifier of a pre-defined text structure. For example:

```
LPGRAPH     [XTITLE='Nitrogen Applied (kg/ha)'] Yield; Nitrogen
```

or

```
TEXT        Experiment
READ        [CHANNEL=2; SERIAL=yes; SETNVALUES=yes] Experiment,Data
LPHISTOGRAM [TITLE=Experiment] Data
```

GenStat prints the y-axis title as a column of characters down the left-hand side of a graph or contour plot. New lines are ignored, so that strings within a text are concatenated. GenStat truncates the title if necessary: the maximum possible number of characters is the number of rows of the frame plus 4. The xaxis title is printed below the graph; the maximum number of characters is the number of columns of the frame plus four: long strings are truncated whereas short strings are centred.

4.10.1 The LPGRAPH directive

LPGRAPH directive

Produces point and line graphs using character (i.e. line-printer) graphics.

Options

CHANNEL = *scalar*	Channel number of output file; default is current output file
TITLE = *text*	General title; default *
YTITLE = *text*	Title for y-axis; default *
XTITLE = *text*	Title for x-axis; default *
YLOWER = *scalar*	Lower bound for y-axis; default *
YUPPER = *scalar*	Upper bound for y-axis; default *
XLOWER = *scalar*	Lower bound for x-axis; default *
XUPPER = *scalar*	Upper bound for x-axis; default *
MULTIPLE = *variate*	Numbers of plots per frame; default * i.e. all plots are on a single frame
JOIN = *string token*	Order in which to join points (ascending, given); default asce
EQUAL = *string tokens*	Whether/how to make bounds equal (no, scale, lower, upper); default no
NROWS = *scalar*	Number of rows in the frame; default * i.e. determined automatically
NCOLUMNS = *scalar*	Number of columns in the frame; default * i.e. determined automatically
YINTEGER = *string token*	Whether y-labels integral (yes, no); default no
XINTEGER = *string token*	Whether x-labels integral (yes, no); default no

Parameters

Y = *identifiers*	Y-coordinates
X = *identifiers*	X-coordinates
METHOD = *string tokens*	Type of each graph (point, line, curve, text); if unspecified, poin is assumed
SYMBOLS = *factors* or *texts*	For factor SYMBOLS, the labels (if defined), or else the levels, define plotting symbols for each unit, whereas a text defines textual information to be placed within the frame for METHOD=text or the symbol to be used for each plot for other METHOD settings; if unspecified, * is used for points, with integers 1-9 to indicate coincident points, ' and . are used for lines and curves
DESCRIPTION = *texts*	Annotation for key

The simplest form of the LPGRAPH directive produces a point plot (or scatterplot as it is sometimes called). It can also be used to plot lines and curves, and text can be added for extra annotation. The data are supplied as y- and x-coordinates in separate parameter lists.

In Example 4.10.1a, the identifiers Y and X are variates of equal length; GenStat uses their values in pairs to give the coordinates of the points to be plotted.

Example 4.10.1a

```
2  VARIATE [VALUES=-16,-7,9,16,7,-8,-12,-5,0,10,4,-4,-3,3,16] X
3  &       [VALUES=0,-14,-12.5,0,14,0,12,0,-10,-9,5,6,-6,-1.5,16] Y
4  LPGRAPH Y; X
```

```
      -+---------+---------+---------+---------+---------+---------+---
      I                                                               I
      I                                                               I
      I                                                               I
 15.0 I                                          *              *     I
      I          *                                                    I
      I                                                               I
      I                                                               I
      I                       *             *                         I
      I                                                               I
  0.0 I   *             *    *                                  *     I
      I                                   *                           I
      I                         *                                     I
      I                                                               I
      I                              *                *               I
      I                                             *                 I
-15.0 I                  *                                            I
      -+---------+---------+---------+---------+---------+---------+---
   -18.0     -12.0      -6.0       0.0       6.0      12.0      18.0

                      Y  v. X using symbol *
```

By default, if you specify several identifiers, GenStat plots them all in the same frame a pair at a time; for example

```
LPGRAPH Y[1...3]; X[1,2]
```

superimposes plots of Y[1] against X[1], Y[2] against X[2], and Y[3] against X[1]. The usual rules governing the parallel expansion of lists apply here: the length of the Y parameter list determines the number of plots within the frame, and the X parameter list is recycled if it is shorter. To generate several frames from one LPGRAPH statement you can use the MULTIPLE option, described below.

The identifiers supplied by the Y and X parameters need not be variates, but can be any numerical structures: scalars, variates, factors, tables or matrices. The only constraints are that the pairs of structures must have the same numbers of values, and that tables must not have margins.

There are four types of graph available, controlled by the METHOD parameter: point (the default), line, curve and text.

A line plot is one in which each point is joined to the next by a straight line. Alternatively, using the curve method, cubic splines are used to produce a smoothed curve through the data points. This does not represent any model fitted in the statistical sense, but as long as the data points are not too widely spaced (especially where the gradient changes quickly) the plotted curve should be a good representation of the underlying function.

By default, GenStat sorts the data so that the x-values are in ascending order before any line or curve is drawn through the points. However, if you set option JOIN=given, the points are joined in the order in which they occur in the data; if there are then any missing values there will be breaks in the line at each missing unit.

Plots produced with METHOD set to either line or curve do not include markings for the data points themselves; you should plot these separately if they are required, as shown in Example 4.10.1b. Here W is plotted against V twice, first with the curve method and then with the point method. **It is best to plot the line first, so that the symbols for individual points will overwrite those used for the line or curve**.

Example 4.10.1b

```
   4 VARIATE [VALUES=-0.1,0.1...0.9] V
   5 &       [VALUES=5.5,9.9,8.7,2.3,1.3,5.5] W
   6 LPGRAPH [TITLE='Point and curve plot'; NROWS=16; NCOLUMNS=61] W,W; V;\
   7         METHOD=curve,point; SYMBOLS=*,'X'; \
   8         DESCRIPTION='Fitted curve ...',*
```

The fourth plotting method is text. You can use this to place an item of text within a graph as extra annotation. For example:

```
SCALAR  Xt,Yt; VALUE=20,10
TEXT    [VALUES='Y=aX+b'] T
LPGRAPH Y,Yt; X,Xt; METHOD=line,text; SYMBOLS=*,T
```

This plots a line, defined by the variates Y and X, as described above. In addition, the text T is printed within the frame starting at the coordinates defined by the scalars Yt and Xt. As these statements show, the SYMBOLS parameter then specifies the text that is to be plotted. The text is truncated as necessary, if positioned too close to the edge of the graph.

With other methods SYMBOL defines the plotting symbol to be used to mark either points or lines on the graph. The default symbol for points is the asterisk, and for lines is a combination of dots and single quotes: you can see these in the earlier examples. If several points coincide, GenStat replaces the asterisk by a digit between 2 and 9, representing the number of coincidences, with 9 meaning nine or more. For point plots, the SYMBOLS parameter can be set to either a text or a factor. If you specify a text with a single string, the string is used to label every point; otherwise, the text must have one string for each point.

By default, GenStat automatically calculates the extent of the axes from the data to be plotted, in such a way that all the data are contained within the frame. You can set one or more of the bounds for the axes by options YLOWER, YUPPER, XLOWER and XUPPER. By setting the upper bound of an axis to a value that is less than the lower bound, you can reverse the usual convention for plotting in which the y-values increase upwards and the x-values increase to the right. Setting the options YINTEGER and XINTEGER constrains the axis markings to be integral, if possible.

The EQUAL option allows you to place constraints on the bounds for the axes. The default setting no (meaning no constraint) uses the boundary values as set by the options or calculated from the data. The settings lower and upper constrain the lower or upper bounds of the two axes to be equal: for example, to plot the line $y=x$ along with the data, setting EQUAL=lower will ensure that it will pass through the bottom left-hand corner of the frame. The scale setting adjusts the y-bounds and x-bounds so that the physical distance on one axis corresponds as closely as possible to physical distance on the other: for example, so that one centimetre will represent the same distance along each axis.

Normally each LPGRAPH statement produces one frame, and GenStat sets the size so that it will fill one screen or line-printer page, based on the settings of WIDTH and PAGE from OPEN or OUTPUT, or their defaults if these have not been specified. When output is going to a file the graph will be placed on a new page, unless this has been disabled using OUTPUT, JOB or SET. The size of the graph is defined in terms of the number of characters in each row and the number of rows in the frame, a row being one line of output. You can adjust the size of the frame by using the NROWS and NCOLUMNS options; the minimum allowed is three rows and three columns, and the maximum number of columns is 17 characters less than the width of the output channel (to leave room for axis markings and titles). There is no maximum on the number of rows. By default, the number of columns is 101, subject to the maximum above, and the number of rows is the number of lines per page, less 8, to allow room for annotation. By defining the page size in advance you can avoid having to specify the numbers of rows and columns when you wish to plot many graphs.

The automatic axis scaling aims to find axis markings that are at reasonable values, but because the markings appear at fixed character positions this may not always be possible. If both upper and lower axis bounds are set, or EQUAL is set in conjunction with axis bounds, or you have requested integral axis markings, there may be conflicting constraints on the axis scaling. If the resultant axis markings then require several decimal places, you may be able to obtain better values by slight adjustments to the numbers of rows or columns.

The MULTIPLE option lets you generate several frames (separate graphs) from one statement. If there is room, the graphs can be printed alongside each other, for example to produce a two-by-two array of plots on a line-printer page. The option should be set to a variate whose elements define the number of graphs to plot in each frame and the number of values in the variate determines the number of frames to be output. For example,

```
LPGRAPH [MULTIPLE=!(2,1,2)] A,B,C,D,E; X[1...3]
```

will produce three frames; the first containing A against X[1] and B against X[2], the second containing C against X[3] and the third containing D against X[1] and E against X[2]. The sum of the values in the MULTIPLE list gives the total number of structures required to form the plots, which must therefore be equal to the length of the Y parameter list. The X list will be recycled if necessary, as here.

By default, each graph will fit the page (as if it had been produced by an individual LPGRAPH statement). However, if you set the NCOLUMNS option to a suitably small value, GenStat may be able to fit more than one frame across the page. The MULTIPLE option will then produce the graphs side by side. Remember that 17 columns are automatically added to provide annotation, and five blank columns are used to separate multiple graphs in parallel. This means that, for example, setting NCOLUMNS=20 will produce two graphs in parallel on a screen of width 80, and three graphs when output to a file of width 121 or more.

You can annotate the graph by using the TITLE, XTITLE and YTITLE options described at the beginning of this section. If none of these are set, a simple key will be produced below the graph, as in Example 4.10.1a, which lists the identifiers and plotting symbols for each pair of Y and X structures. You can obtain your own key by setting the DESCRIPTION parameter, which supplies a line of text for each plot, as in Example 4.10.1b.

4.10.2 The LPHISTOGRAM directive

LPHISTOGRAM directive

Produces histograms using character (i.e. line-printer) graphics.

Options

CHANNEL = *scalar*	Channel number of output file; default is the current output file
TITLE = *text*	General title; default *
LIMITS = *variate*	Variate of group limits for classifying variates into groups; default *
NGROUPS = *scalar*	When LIMITS is not specified, this defines the number of groups into which a data variate is to be classified; default is the integer value nearest to the square root of the number of values in the variate
LABELS = *text*	Group labels
SCALE = *scalar*	Number of units represented by each character; default 1

Parameters

DATA = *identifiers*	Data for the histograms; these can be either a factor indicating the group to which each unit belongs, a variate whose values are to be grouped, or a one-way table giving the number of units in each group
NOBSERVATIONS = *tables*	One-way table to save numbers in the groups
GROUPS = *factors*	Factor to save groups defined from a variate
SYMBOLS = *texts*	Characters to be used to represent the bars of each histogram
DESCRIPTION = *texts*	Annotation for key

LPHISTOGRAM plots histograms or bar charts, depending on the input supplied by the DATA parameter: either a list of variates, a list of factors or a list of one-way tables. Histograms are formed from variates to provide quick and simple visual summaries of the data that they contain. The data values are divided into several groups, which are then displayed as a histogram consisting of a line of asterisks for each group. The number of asterisks in each line is proportional to the number of values assigned to that group; this figure is also printed at the beginning of each line.

Example 4.10.2a

```
  2  VARIATE Data
  3  READ Data

     Identifier    Minimum       Mean    Maximum      Values     Missing
           Data     0.0000      3.960      9.000          25           0

  5  LPHISTOGRAM Data
```

```
Histogram of Data
-----------------

             -   2  9 *********
           2 -   4  7 *******
           4 -   6  5 *****
           6 -   8  2 **
           8 -      2 **

Scale:  1 asterisk represents 1 unit.
```

You can specify a list of variates, to obtain a parallel histogram. For each group one row of asterisks is printed for each variate, labelled by the corresponding identifier.
As shown in Example 4.10.2b, the variates are sorted according to the same intervals. There is no need for them all to have the same numbers of values.

Example 4.10.2b

```
   6  VARIATE Data2
   7  READ Data2
```

Identifier	Minimum	Mean	Maximum	Values	Missing
Data2	0.0000	3.225	8.000	40	0

```
  10  LPHISTOGRAM Data,Data2

Histogram of Data and Data2
---------------------------

               - 1.5  Data    5 *****
                      Data2   9 *********

           1.5 - 3.0  Data    6 ******
                      Data2  14 **************

           3.0 - 4.5  Data    5 *****
                      Data2   7 *******

           4.5 - 6.0  Data    5 *****
                      Data2   8 ********

           6.0 - 7.5  Data    1 *
                      Data2   1 *

           7.5 -      Data    3 ***
                      Data2   1 *

Scale:  1 asterisk represents 1 unit.
```

You can use the NGROUPS option to specify the number of groups in the histogram; GenStat will then work out appropriate limits, based on the range of the data, to form intervals of equal width. For example:

```
    LPHISTOGRAM [NGROUPS=5] Data
```

Alternatively, you can define the groups explicitly, by setting the LIMITS option to a variate containing the group limits. In Example 4.10.2c, Limits is a variate with seven values, producing a histogram in which the data is split into eight groups: #1, 1-2, 2-3, 3-5, 5-7, 7-8, 8-10, >10. The upper limit of each group is included within that group, so the group 3-5, for example, contains values that are greater than 3 and less than or equal to 5. The values of the limits variate are sorted into ascending order if necessary, but the variate itself is not changed.

Example 4.10.2c

```
  11  VARIATE [VALUES=1,2,3,5,7,8,10] Glimits
  12  LPHISTOGRAM [LIMITS=Glimits] Data

Histogram of Data grouped by Glimits
------------------------------------

             -   1.00  5 *****
        1.00 -   2.00  4 ****
        2.00 -   3.00  2 **
        3.00 -   5.00  7 *******
        5.00 -   7.00  4 ****
        7.00 -   8.00  1 *
        8.00 -  10.00  2 **
       10.00 -         0

Scale:  1 asterisk represents 1 unit.
```

You can use the LABELS option to provide your own labelling for the groups of the histogram. It should be set to a text vector of length equal to the number of groups. If neither NGROUPS nor LIMITS has been set, the number of groups is determined from the number of values in the LABELS structure. If LABELS is also unset, the default number of groups is chosen as the integer value nearest to the square root of the number of values (as in Example 4.10.2a where 25 values are sorted into five groups), up to a maximum of 10. Alternatively, procedure AKAIKEHISTOGRAM provides a more sophisticated method of generating histograms, using Akaike's Information Criterion (AIC) to generate an optimal grouping of the data.

If the DATA parameter is set to a factor or a one-way table, the histogram takes the form of a bar chart. There is now no longer the concept of dividing the x-axis into a set of contiguous intervals. Instead we have a set of bars located at various positions along the x-axis.

To form a bar chart from a factor, GenStat counts the number of units that occur with each level of the factor; thus the number of groups is the number of levels of the factor and the value for each group is the corresponding total. The labels of the factor (if present) are used to label the groups, as shown in Example 4.10.2d. Otherwise GenStat uses the factor levels.

Example 4.10.2d

```
  13  TEXT [VALUES=apple,banana,peach,cherry,pear,orange] Name
  14  FACTOR [LEVELS=6; LABELS=Name; NVALUES=32] Fruit
  15  READ Fruit
```

Identifier	Values	Missing	Levels
Fruit	32	0	6

```
  17  LPHISTOGRAM Fruit

Histogram of Fruit
------------------


 apple  3 ***
banana  2 **
 peach  8 ********
cherry  5 *****
  pear  8 ********
orange  6 ******

Scale:  1 asterisk represents 1 unit.
```

When GenStat plots the histogram of a one-way table, the number of groups is the number of levels of the factor classifying the table and the values of the table indicate the number of observations in each group. The labels or levels of the classifying factor are again used to label the histogram.

The LABELS option can also be used when producing a histogram from a factor or table. It should be set to a text of length equal to the number of levels of the factor or classifying factor. When producing a parallel histogram the data structures must all be of the same type: variate, factor or table. Variates and factors may be restricted, in which case only the subset of values specified by the restriction will be included in the histogram; however, unlike many directives, restrictions do not carry over to the other structures listed by the DATA parameter. If parallel histograms are to be formed from several factors, they must all have the same number of levels, and the labels or levels of the first factor will be used to identify the groups. Likewise, if you are forming parallel histograms from several tables, they must all have the same number of values, and the classifying factor of the first table will define the labelling of the histogram.

The SYMBOLS parameter can specify alternative plotting characters to be used instead of the asterisk. For example:

```
    LPHISTOGRAM Variate; SYMBOLS='+'
```

You can specify a different string for each structure in a parallel histogram. If you specify strings of more than one character, GenStat uses the characters in order, recycled as necessary, until each histogram bar is of the correct length.

Example 4.10.2e

```
  18  LPHISTOGRAM Data; SYMBOLS='X-O-'

Histogram of Data
-----------------

              -    2  9 X-O-X-O-X
            2 -    4  7 X-O-X-O
            4 -    6  5 X-O-X
            6 -    8  2 X-
            8 -       2 X-

Scale:  1 character represents 1 unit.
```

You can use the DESCRIPTION parameter to provide a text for labelling the histogram instead of the identifiers of the DATA structures.

Normally one asterisk will represent one unit. However, if there are many data values and the groups become large, GenStat may not be able to fit enough asterisks into one row. It will then alter the scaling so that one asterisk represents several units. You can set the scaling explicitly using the SCALE option; the value specified is rounded to the nearest integer, and determines how many units should be represented by each asterisk.

LPHISTOGRAM has two output parameters that allow you to save information that has been generated during formation of the histogram. The NOBSERVATIONS parameter allows you to save a one-way table of counts that contains the number of observations that were assigned to each group; the missing-value cell of this table will contain a count of the number of units that were missing and that therefore remain unclassified. When producing a histogram from a variate, you can use the GROUPS parameter to specify a factor to record the group to which each unit was allocated.

4.10.3 The LPCONTOUR directive

LPCONTOUR directive

Produces contour maps of two-way arrays of numbers using character (i.e. line-printer) graphics.

Options

CHANNEL = *scalar*	Channel number of output file; default is current output file
INTERVAL = *scalar*	Contour interval for scaling; default * i.e. determined automatically

`TITLE` = *text*	General title; default *
`YTITLE` = *text*	Title for y-axis; default *
`XTITLE` = *text*	Title for x-axis; default *
`YLOWER` = *scalar*	Lower bound for y-axis; default 0
`YUPPER` = *scalar*	Upper bound for y-axis; default 1
`XLOWER` = *scalar*	Lower bound for x-axis; default 0
`XUPPER` = *scalar*	Upper bound for x-axis; default 1
`YINTEGER` = *string token*	Whether y-labels integral (`yes`, `no`); default `no`
`XINTEGER` = *string token*	Whether x-labels integral (`yes`, `no`); default `no`
`LOWERCUTOFF` = *scalar*	Lower cut-off for array values; default *
`UPPERCUTOFF` = *scalar*	Upper cut-off for array values; default *

Parameters

`GRID` = *identifiers*	Pointers (of variates representing the columns of a data matrix), matrices or two-way tables specifying values on a regular grid
`DESCRIPTION` = *texts*	Annotation for key

A contour plot provides a way of displaying three-dimensional data in a two-dimensional plot. The data values are supplied as a rectangular array of numbers that represent the values of the variable in the third dimension, often referred to as *height* or the *z-axis*. The first two dimensions (x and y) are the rows and columns indexing the array; the complete three-dimensional data set is referred to as a *surface* or *grid*. Contours are lines that are used to join points of equal height, and usually some form of interpolation is used to estimate where these points lie. The resulting contour plot is not necessarily very "realistic" when compared to surface plots (4.4.3), but it has the advantage that the entire surface can easily be examined, without the danger of some parts being obscured by high points or regions.

You might use contour plots for example when you have data sampled at points on a regular grid, such as the concentrations of a trace element or nutrient in the soil. Contours are also very useful when fitting nonlinear models, when they can be used to study two-dimensional slices of the likelihood surface, to help find good initial estimates of the parameters.

`LPCONTOUR` produces output for a line printer by using cubic interpolation between the grid points to estimate a z-value for each character position in the plot. Each value is reduced to a single digit in the range 0 ... 9, according to the rules described below. To produce the contour plot only the even digits are printed: you can then see the contours as the boundaries between the blank areas and the printed digits.

In Example 4.10.3a, a function of two variables is calculated, and the shape of the function is displayed with `LPCONTOUR`. Titles have been given to the x-axis and the y-axis, and there is an overall title giving the algebraic form of the function.

Example 4.10.3a

```
  2  MATRIX    [ROWS=5; COLUMNS=7] X,Y; VALUES=!((1...7)5),!(7(1...5))
  3  CALCULATE Zvalues = (X-2.5)*(X-6)*X - 10*(Y-3)*(Y-3)
  4  LPCONTOUR [TITLE='Z(x,y) = x*(x-2.5)*(x-6) - 10*(y-3)**2';\
  5            YTITLE='Y values'; XTITLE='X values'] Zvalues

 Contour plot of Zvalues at intervals of 8.400

 ** Scaled values at grid points **
   -3.8690   -4.2857   -5.2976   -6.1905   -6.2500   -4.7619   -1.0119
   -0.2976   -0.7143   -1.7262   -2.6190   -2.6786   -1.1905    2.5595
    0.8929    0.4762   -0.5357   -1.4286   -1.4881    0.0000    3.7500
   -0.2976   -0.7143   -1.7262   -2.6190   -2.6786   -1.1905    2.5595
   -3.8690   -4.2857   -5.2976   -6.1905   -6.2500   -4.7619   -1.0119

                 Z(x,y) = x*(x-2.5)*(x-6) - 10*(y-3)**2
```

```
          0.0000    0.1667    0.3333    0.5000    0.6667    0.8333    1.0000
           '         '         '         '         '         '         '

     1.000-66666             4444444444               4444444   666  888-
           666666666666666          44444444444444444444      666   88
                       6666666666                         66666   88   0
           8888888            6666666666              666666    888  00
           88888888888888          6666666666666666666666     888   00
                   88888888888           666666666666       8888   00  2
     0.750-              888888888                        8888   000  22-
           0000             8888888888                 888888   000   22
Y          0000000000          88888888888          8888888    000   22
           0000000000000         8888888888888888888888888     000  222
v          00000000000000          8888888888888888888888     000   22
a          000000000000000          888888888888888888888     000   22
l    0.500-0000000000000000         88888888888888888888     0000   22  -
u          000000000000000          888888888888888888888     000   22
e          00000000000000          8888888888888888888888     000   22
s          0000000000000         8888888888888888888888888     000  222
           0000000000          88888888888          8888888    000   22
           0000             8888888888                 888888   000   22
     0.250-              888888888                        8888   000  22-
                   88888888888           666666666666       8888   00  2
           88888888888888          6666666666666666666666     888   00
           8888888            6666666666              666666    888  00
                       6666666666                         66666   88   0
           666666666666666          44444444444444444444      666   88
     0.000-66666             4444444444               4444444   666  888-
           '         '         '         '         '         '         '

                                     X values
```

The GRID parameter can be set to a matrix, a two-way table (with the first factor defining the rows), or a pointer to a set of variates each containing a column of data. We explain the conventions in terms of a matrix as input, but similar rules apply to the other structures. When reading or printing a matrix the origin of the rows and columns (row 1, column 1) appears at the top left-hand corner. However, in forming the contour plot the rows are reversed in order so that the first row of the matrix is placed at the bottom of the contour; thus the origin of the contour is located, according to the usual conventions, at the bottom left-hand corner of the plot. (By default, the DCONTOUR directive reverses the rows of the grid in the same way, but it also has an ORIENTATION option that allows you to plot with the normal orientation; see 4.4.1.)

LPCONTOUR scales the grid values by dividing by the contour interval. The scaled grid values are then converted to single digits by taking the remainder modulo 10 and truncating the fractional part. In Example 4.10.3a, the first grid value is -32.5, which is divided by the interval size (8.4) to obtain -3.869; this becomes 6.131 when taken modulo 10, and then 6 after truncation. To aid interpretation of the plot, the array of scaled values is printed out.

The INTERVAL option allows you to set the interval between contour lines. For example, if the grid values range from 17 to 72 and the interval is set to 10, contour lines (the boundaries between blank space and printed digits) will occur at grid values of 20, 30, 40, 50, 60 and 70. By default, the interval is determined from the range of the data in order to obtain 10 contours.

The UPPERCUTOFF and LOWERCUTOFF options can be used to define a window for the grid values that will form the contours. All values above or below these are printed as X. Setting either UPPERCUTOFF or LOWERCUTOFF will change the default contour interval, as the range of data values is effectively curtailed.

You can use the TITLE, YTITLE and XTITLE options to annotate the contour plot. If you specify several grids, these will be plotted in separate frames and the text of the TITLE option will appear at the top of each one. You should thus use TITLE only to give a general description of what the contours represent. The DESCRIPTION parameter can be used to add specific descriptions to be printed at the bottom of each individual plot.

The YUPPER and YLOWER options allow you to set upper and lower bounds for the y-axis; thus generating axis labels that reflect the range of values over which the grid was observed or evaluated.

Setting `YINTEGER=yes` will ensure the labels are printed as integers, if possible. The default axis bounds are 0.0 and 1.0. The options `XLOWER`, `XUPPER` and `XINTEGER` similarly control labelling of the x-axis.

Example 4.10.3b shows how a contour plot can be produced from a set of variates. In line 22, the values of the variates are inverted, using the `REVERSE` function, and y-axis labelling set up so that *depth* increases as you read down the plot. (The same data are plotted in Figure 4.4.1a using the `DCONTOUR` directive, but there the `YORIENTATION` option is used to reverse the y-axis.)

Example 4.10.3b

```
  6  " Core samples were taken from a wetland rice experiment to examine
 -7    the leaching of ammonium nitrate. Three three cores were taken at
 -8    intervals of 5cm, and the concentration of ammonium nitrate was
 -9    measured at depths of 4, 8, ... 20 cm. "
 10  VARIATE   [NVALUES=5] Core[1...5]
 11  READ      Core[]
```

Identifier	Minimum	Mean	Maximum	Values	Missing
Core[1]	5.000	8.200	11.00	5	0
Core[2]	6.000	67.60	195.0	5	0
Core[3]	129.0	940.6	2315	5	0
Core[4]	10.00	36.00	77.00	5	0
Core[5]	7.000	9.400	15.00	5	0

```
 17  TEXT      [VALUES=' Samples taken 40 days after placement ', \
 18            ' of 2 grams supergranule urea. '] Coredesc
 19  CALCULATE Core[] = LOG10(REVERSE(Core[]))
 20  LPCONTOUR [YTITLE='Soil depth in cm';  \
 21            XTITLE='Distance from central core'; \
 22            YINTEGER=yes; XINTEGER=yes; \
 23            YUPPER=4; YLOWER=20; XUPPER=10; XLOWER=-10] \
 24            Core; DESCRIPTION=Coredesc

Contour plot of Core at intervals of 0.267

** Scaled values at grid points **
   3.1704    2.9193    7.9179    3.7515    3.5799
   3.3880    7.1797   12.6222    6.4687    3.1704
   2.6222    8.5911   11.3557    7.0772    3.1704
   3.7515    5.7926   11.3091    5.5415    3.5799
   3.9068    4.8808    8.2790    3.7515    4.4121
```

```
         -10        -5         0         5        10
           '         '         '         '         '

         4- 2222222222  4  66      66  44          -
              222     44 66  88 888  66 444
                   444  6  88  0   8  66 4444
                 444  66 88  00 00  88 66  444
S              4444  66 88 00    00  8  6   444
o             444   66 88 00   2  00  8 66  4444
i        8-  444   66  8  0  2222  00 8  66  444    -
l            444  66  88 00  2222  00 88 66   44
             44  66  88  00  2222  00 88 666  444
d          2 44  66  88 000   222  00 88  66  444
e          2 44  6  88  000       000 88  66   44
p          2 44 66  88  000       00  88  66   44
t        12-2 44 66  88   000     000  88  66   44   -
h          2 44  66  88  0000    000  88 666  444
           2 44  666  88  000    00  88  66   444
i             44  666  88  00    00  88  66  4444
n             444  666 88  000   00  88 66   444
              4444   66 88  00   00  8  6   4444
c        16-    4444   6  8  00  000 88 66  4444     -
m               44444  66 88  00 00  8  6  4444
                 44444  6  88  000  88 66 4444
                 44444  66 88   0  88  6  444
                 444444 66  88    88  66 444        4
                 444444  66  888888  66  44         4
         20-    4444444  666   88   66  44         44-
           '         '         '         '         '

                    Distance from central core

        Samples taken 40 days after placement
            of 2 grams supergranule urea.
```

4.10.4 The STEM procedure

STEM procedure

Produces a simple stem-and-leaf chart (J. Ollerton & S.A. Harding).

No options

Parameters

`DATA` = *variates*	Data values for each plot
`NDIGITS` = *scalars*	Number of digits in the leaves of each plot
`STEMUNITS` = *scalars*	Scale units for the stem values in each plot

`STEM` produces a simple stem-and-leaf chart of a variate of data. The stems indicate leading digits and the leaves indicate subsequent digits. By default, the leaves are formed from single digits; the parameter `NDIGITS` can be used to specify the number of digits in each leaf if more than one is required. The `STEMUNITS` parameter can be used to specify the units represented by the stem values. By default, this is determined from the data so that the display will fit within a single screen or page of output. Small values of `STEMUNITS` (in comparison to the range of the data) should be avoided as they may generate far too many lines of output. The display produced by `STEM` is restricted to the current output width; any lines that have to be truncated at the right-hand margin are terminated by >, indicating their continuation.

In Example 4.10.4 stem-and-leaf charts are made of 18 Prices. In the first chart `NDIGIT=1` giving leaf digits of 1; in the second chart `NDIGIT=2` giving leaf digits of 2.

Example 4.10.4

```
   2  VARIATE [NVALUES=18] Prices
   3  READ    [PRINT=data] Prices
   4      250  150  795  895  696 1699 1499 1099 1693
   5     1166  688 1333  895 1775  895 1895  795  806 :
   6  STEM    Prices; NDIGIT=1
```

Stem-and-leaf display for Prices
Number of observations: 18. Minimum: 150.0. Maximum: 1895.0.
Stem units: 100, leaf digits: 1 (the value 150.0 is represented by 1|5)

```
1      1|5
1      2|5
0      3|
0      4|
0      5|
2      6|89
2      7|99
4      8|0999
0      9|
1     10|9
1     11|6
0     12|
1     13|3
1     14|9
0     15|
2     16|99
1     17|7
1     18|9
```

```
   7  STEM    Prices; NDIGIT=2
```

Stem-and-leaf display for Prices
Number of observations: 18. Minimum: 150.0. Maximum: 1895.0.
Stem units: 100, leaf digits: 2 (the value 150.0 is represented by 1|50)

```
1      1|50
1      2|50
0      3|
0      4|
0      5|
2      6|88,96
2      7|95,95
4      8|06,95,95,95
0      9|
1     10|99
1     11|66
0     12|
1     13|33
1     14|99
0     15|
2     16|93,99
1     17|75
1     18|95
```

5 Examples

In this chapter a lot of figures are shown to demonstrate the various possibilities of the GenStat high quality graphics. The DGRAPH directive offers the most possibilities. The examples contain both the GenStat program as well as the produced figure.

Some of the programs have so many lines that the figure is placed on the next page. For printing two pages per page (or viewing this document in Word with View → Two Pages) the layout is so, that the program and the figure can be seen simultaneously. For printing double-sided-one-page-per-page you might insert a blank page before this Chapter 5 to get program and figure side by side.

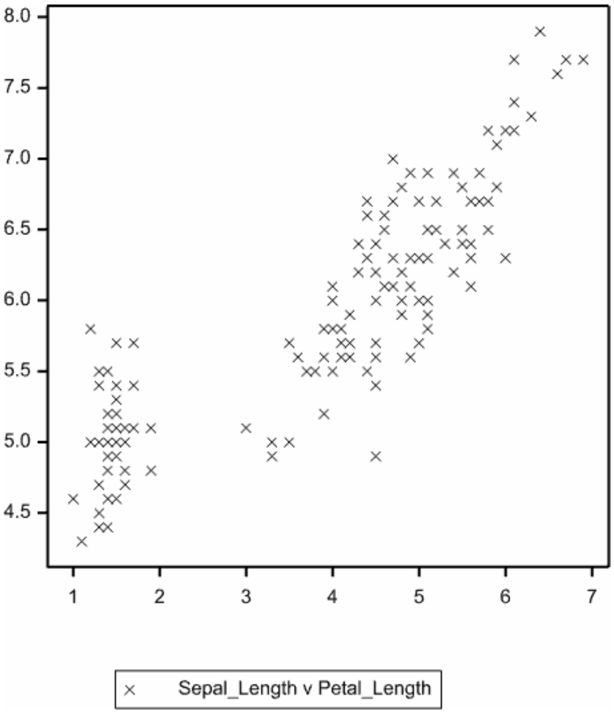
The examples 1 up to 10 are the default figures of the 10 available graphical directives. The next examples demonstrate various option and parameter settings of the available directives to modify the environment and various option and parameter settings of plot directives and procedures. The last examples show you figures produced by procedures of GenStat's general procedure library.

Tip

The examples of procedures (and directives) can also be run with the Help button on the Menu bar: Help → Examples → Commands...

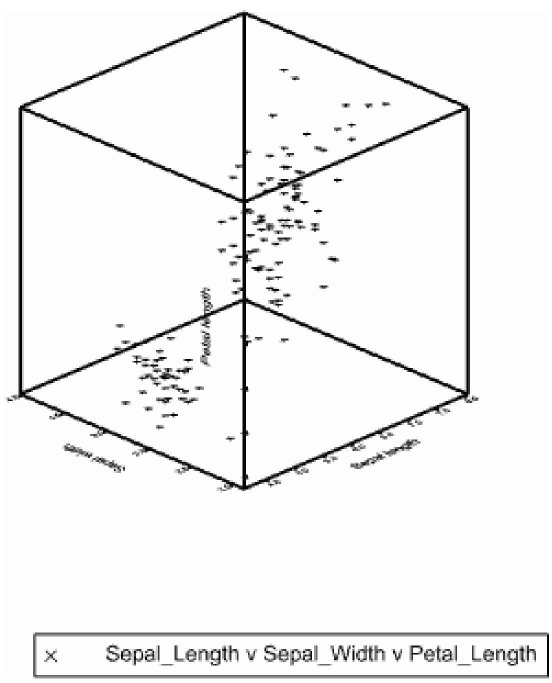
In Appendix 4 the conclusion is drawn that the formats EMF and EPS and the Copy and Paste method seem to give the best quality **printed on paper**. They also have minimal margins. However, after inserted them into Word, **the view on the screen is too bold**. Therefore the figures of the examples are saved as png-files and inserted in this document with Insert → Picture. Figure 13.5 is the only figure of the examples that is inserted with the Copy and Paste method because the PNG format looses the title along the y-axis.

```
JOB         'Example 1: Default output from DGRAPH'
SPLOAD      'C:/Program Files (x86)/Gen15ed/Data/Iris.gsh'
DGRAPH      Y=Sepal_Length ; X=Petal_Length
STOP
```



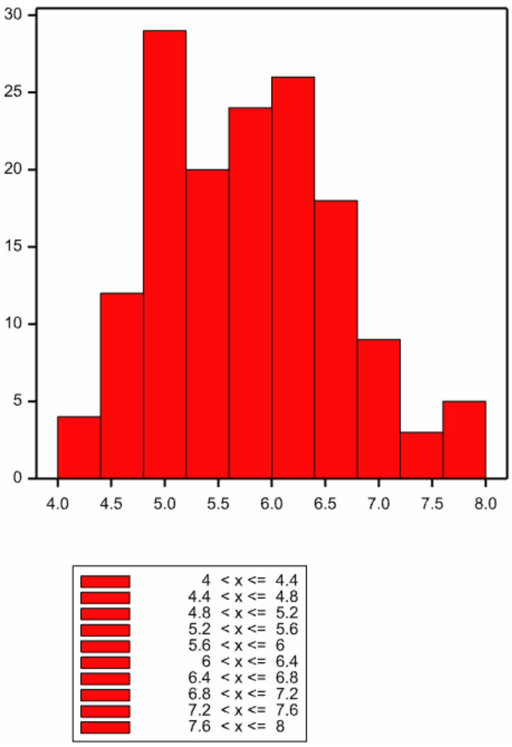
Example 1: Default output from DGRAPH

```
JOB         'Example 2: Default output from D3GRAPH'
SPLOAD      'C:/Program Files (x86)/Gen15ed/Data/Iris.gsh'
D3GRAPH     X=Sepal_Length; Y=Sepal_Width; Z=Petal_Length
STOP
```



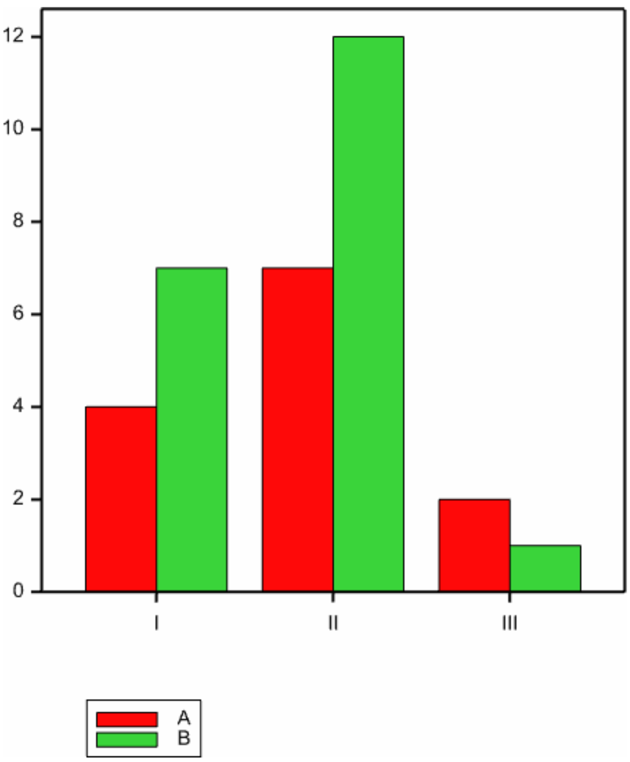
Example 2: Default output from D3GRAPH

```
JOB           'Example 3: Default output from DHISTOGRAM'
SPLOAD        'C:/Program Files (x86)/Gen15ed/Data/Iris.gsh'
DHISTOGRAM    Sepal_Length
STOP
```



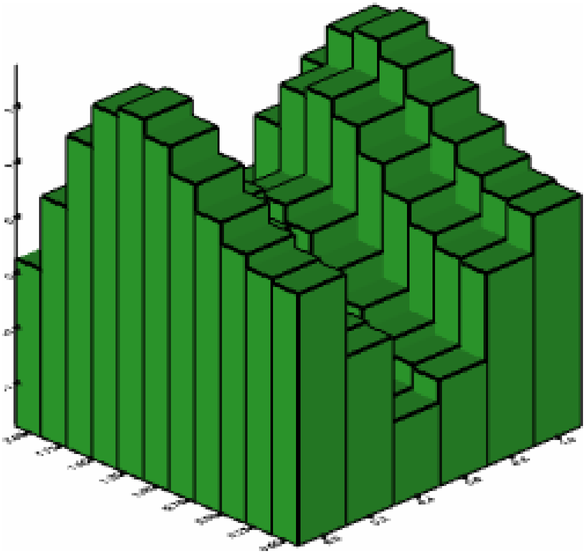
Example 3: Default output from DHISTOGRAM

```
JOB         'Example 4: Default output from BARCHART'
FACTOR      [LABELS= !T(A, B)] f1
FACTOR      [LABELS= !T(I, II, III)] f2
TABLE       [CLASS= f1, f2; VALUES= 4, 7, 2, 7, 12, 1] tab
BARCHART    tab
STOP
```



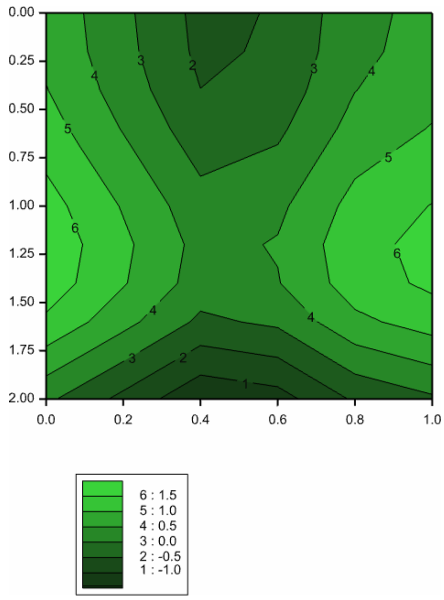
Example 4: Default output from BARCHART

```
JOB          'Example 5: Default output from D3HISTOGRAM'
VARIATE      [VALUES= 0, 0.2 ... 2] Rows
&            [VALUES= 0, 0.2 ... 1] Columns
MATRIX       [ROWS= Rows; COLUMNS= Columns] \
             X, Y, Function
&            [VALUES= (#Columns)11] X
CALCULATE    Y $ [*; 1...6] = Rows
&            Function = COS( 1 / ( X + 0.1) ** 2) + SIN( Y ** 2)
D3HISTOGRAM  Function
STOP
```



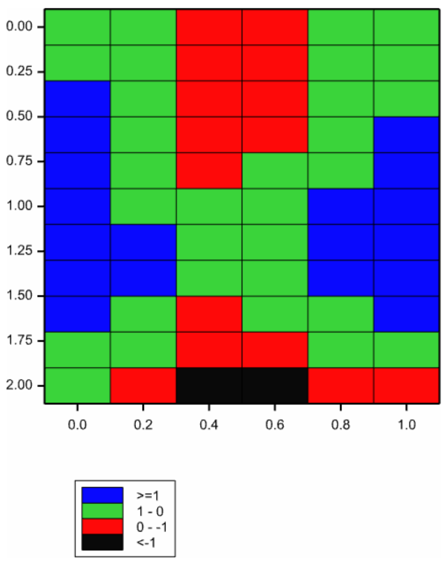
Example 5: Default output from D3HISTOGRAM

```
JOB          'Example 6: Default output from DCONTOUR'
VARIATE      [VALUES= 0, 0.2 ... 2] Rows
&            [VALUES= 0, 0.2 ... 1] Columns
MATRIX       [ROWS= Rows; COLUMNS= Columns] \
             X, Y, Function
&            [VALUES= (#Columns)11] X
CALCULATE    Y $ [*; 1...6] = Rows
&            Function=COS(1 / ( X + 0.1)** 2)+SIN(Y ** 2)
DCONTOUR     Function
STOP
```



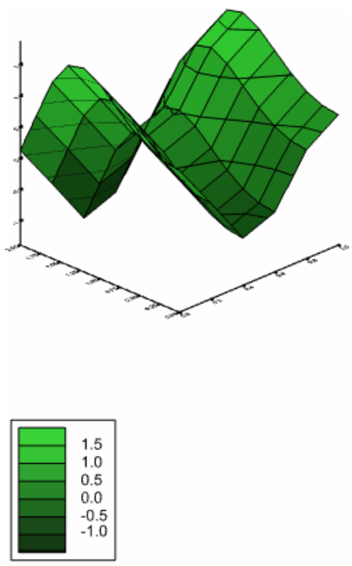
Example 6: Default output from DCONTOUR

```
JOB          'Example 7: Default output from DSHADE'
VARIATE      [VALUES= 0, 0.2 ... 2] Rows
&            [VALUES= 0, 0.2 ... 1] Columns
MATRIX       [ROWS= Rows; COLUMNS= Columns] \
             X, Y, Function
&            [VALUES= (#Columns)11] X
CALCULATE    Y $ [*; 1...6] = Rows
&            Function = COS( 1 / ( X + 0.1) ** 2) + SIN( Y ** 2)
DSHADE       Function ; NGROUPS=6
STOP
```



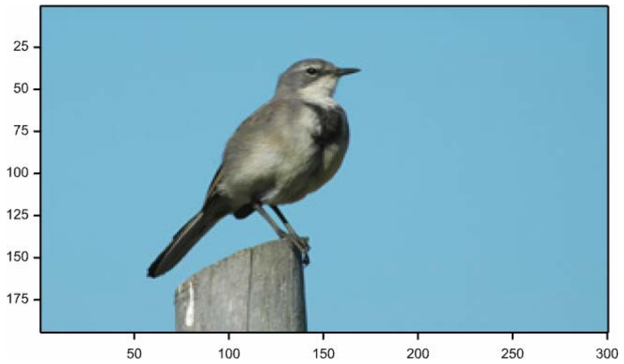
Example 7: Default output from DSHADE

```
JOB          'Example 8: Default output from DSURFACE'
VARIATE      [VALUES= 0, 0.2 ... 2] Rows
&            [VALUES= 0, 0.2 ... 1] Columns
MATRIX       [ROWS= Rows; COLUMNS= Columns] \
             X, Y, Function
&            [VALUES= (#Columns)11] X
CALCULATE    Y $ [*; 1...6] = Rows
&              Function=COS(1 / ( X + 0.1)**2)+SIN(Y**2)
DSURFACE       Function
STOP
```



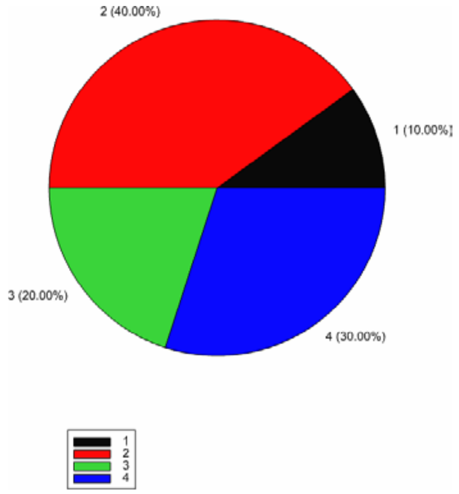
Example 8: Default output from DSURFACE

```
JOB         'Example 9: Default output from DBITMAP'
IMPORT      [RGBMETHOD=matrix] \
             '%GENDIR%/Examples/GuidePart1/CapeWagtail.jpg'; \
            COLUMNS='RGB'
" resize the window to match the dimensions of the bit map "
CALCULATE   Nr = NROWS(RGB)
&           Nc = NCOLUMNS(RGB)
FRAME       3; YUPPER= Nr / Nc
DBITMAP     [WINDOW=3] RGB
STOP
```



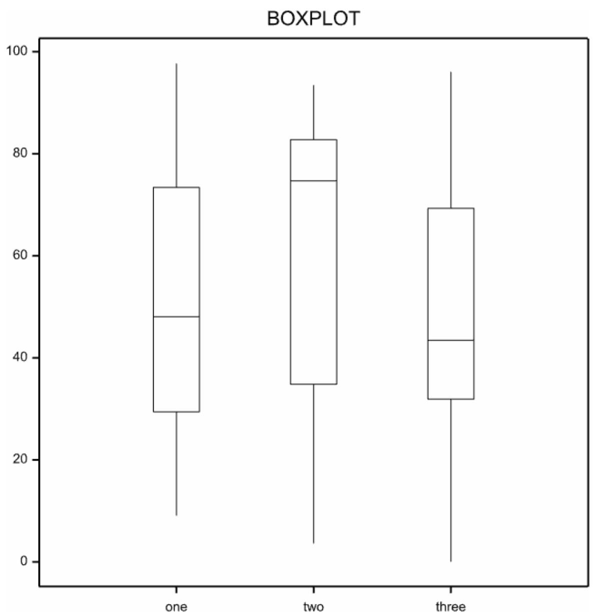
Example 9: Default output from DBITMAP

```
JOB     'Example 10: Default output from DPIE'
DPIE    SLICE= 1, 4, 2, 3
STOP
```



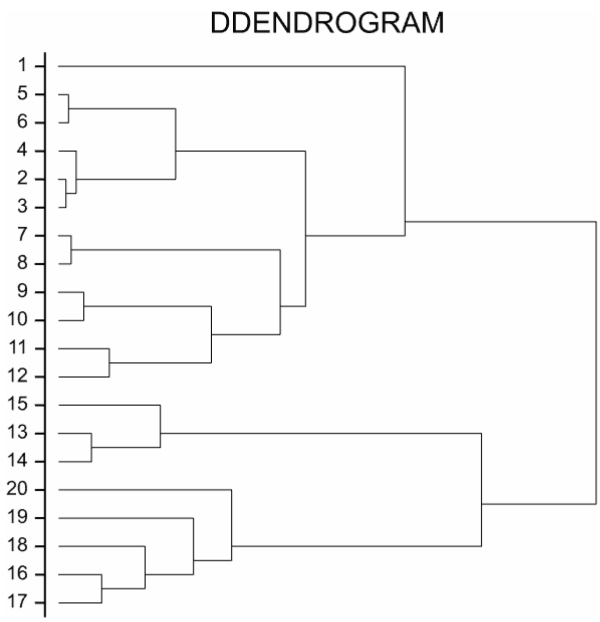
Example 10: Default output from DPIE

```
JOB         'Example 11: Procedure BOXPLOT'
CALCULATE   Data = URAND(1387; 45) * 100
FACTOR      [LABELS= !T( one, two, three); \
            VALUES= 15(1...3)] Group
BOXPLOT     [TITLE='BOXPLOT'] \
            Data; GROUPS= Group
STOP
```



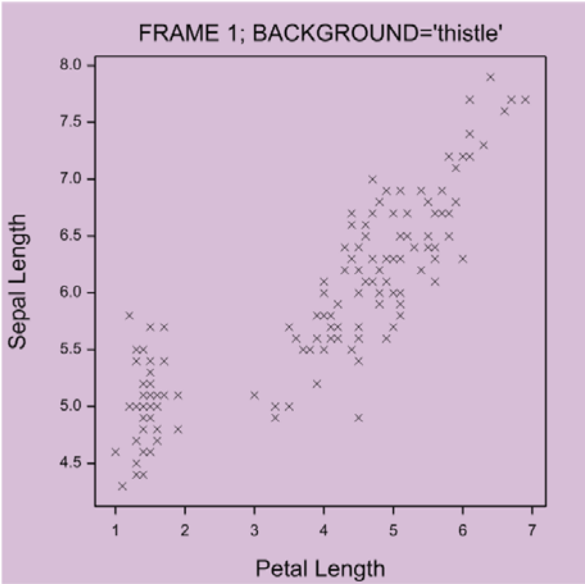
Example 11: Output from procedure BOXPLOT

```
JOB           'Example 12: Procedure DDENDROGRAM'
MATRIX        [ROWS= 19; COLUMNS= 3] Am
CALCULATE     Am$[*;1] = !(2,5,7,2,9,13,16,11,16,\
                          13,2,16,9,16,7,2,1,13,1)
&             Am$[*;2] = !(3,6,8,4,10,14,17,12,18,\
                          15,5,19,11,20,9,7,2,16,13)
&             Am$[*;3] = !(3,4,5,7,10,13,17,20,34,40,46,\
                          53,60,68,87,97,136,166,211)
DDENDROGRAM [STYLE= centroid ; METHOD= distances] \
              DATA= Am ; TITLE= 'DDENDROGRAM'
STOP
```

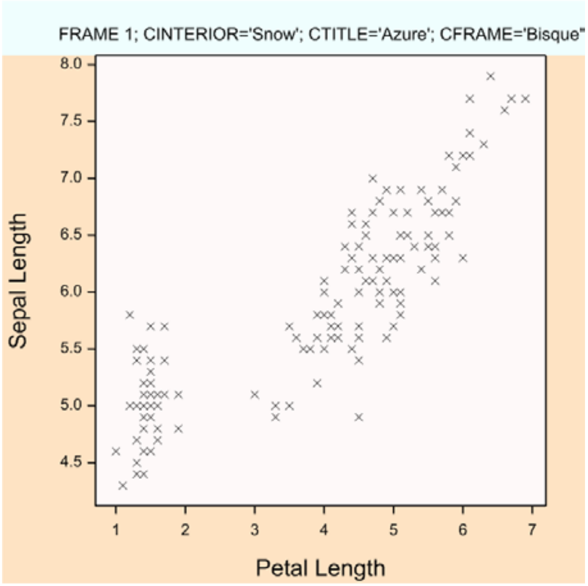


Example 12: Output from procedure DDENDROGRAM

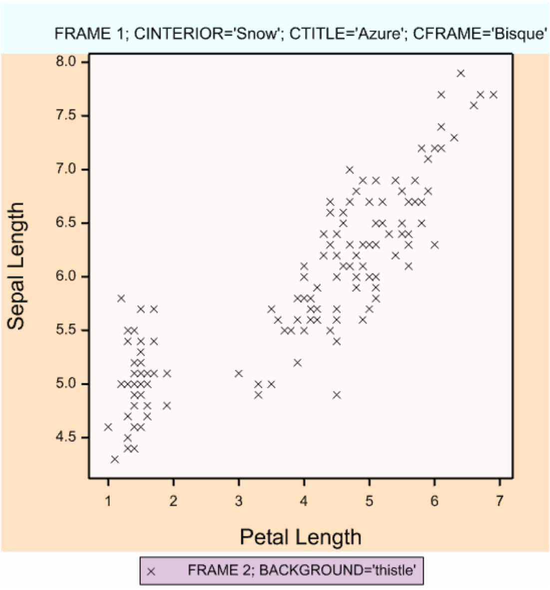
```
JOB       'Example 13
          Directive FRAME: parameters BACKGROUND or CINTERIOR, CTITLE, CFRAME
                           parameters BOX and BOXKEY
                           option BOXFRAME'
SPLOAD    '%GENDIR%/Data/Iris.gsh'
XAXIS     1,3 ; TITLE= 'Petal Length'
YAXIS     1,3 ; TITLE= 'Sepal Length'
"Example 13.1"
FRAME     1; BACKGROUND='thistle'
DGRAPH    [WINDOW=1; KEYWINDOW=0 ; \
          TITLE='FRAME 1; BACKGROUND=''thistle'''] \
          Sepal_Length ; X=Petal_Length
"Example 13.2"
FRAME     1; CINTERIOR='Snow'; CTITLE='Azure'; CFRAME='Bisque'
DGRAPH    [WINDOW=1; KEYWINDOW=0 ; TITLE= \
          'FRAME 1; CINTERIOR=''Snow''; CTITLE=''Azure''; CFRAME=''Bisque'''] \
          Sepal_Length ; X=Petal_Length
"Example 13.3"
FRAME     2; BACKGROUND='thistle'
DGRAPH    [WINDOW=1; KEYWINDOW=2; TITLE= \
          'FRAME 1; CINTERIOR=''Snow''; CTITLE=''Azure''; CFRAME=''Bisque'''] \
          Sepal_Length ; X=Petal_Length ; \
          DESCRIPTION='FRAME 2; BACKGROUND=''thistle'''
"Example 13.4"
FRAME     2; CINTERIOR='Snow'; CTITLE='Azure'; CFRAME='Bisque'
DGRAPH    [WINDOW=1; KEYWINDOW=2; TITLE= \
          'FRAME 1; CINTERIOR=''Snow''; CTITLE=''Azure''; CFRAME=''Bisque'''] \
          Sepal_Length ; X=Petal_Length ; \
          DESCRIPTION='FRAME 2 ; CINTERIOR=''Snow'''
"Example 13.5"
FRAME     [BOXFRAME=include] 1; BOX=omit; BACKGROUND='Azure'
FRAME     [RESET=yes] 2; BOXKEY=omit
DGRAPH    [WINDOW=1; KEYWINDOW=2; TITLE= \
          'FRAME [BOXFRAME=include] 1; BOX=omit; BACKGROUND=''Azure'''] \
          Sepal_Length ; X=Petal_Length ; \
          DESCRIPTION='FRAME 2; BOXKEY=omit'
"Example 13.6"
FRAME     [BOXFRAME=include] 3; BACKGROUND= 'Azure'
DGRAPH    [WINDOW=3; KEYWINDOW=0; \
          TITLE='FRAME [BOXFRAME=include] 3'] \
          Sepal_Length ; X=Petal_Length
STOP
```



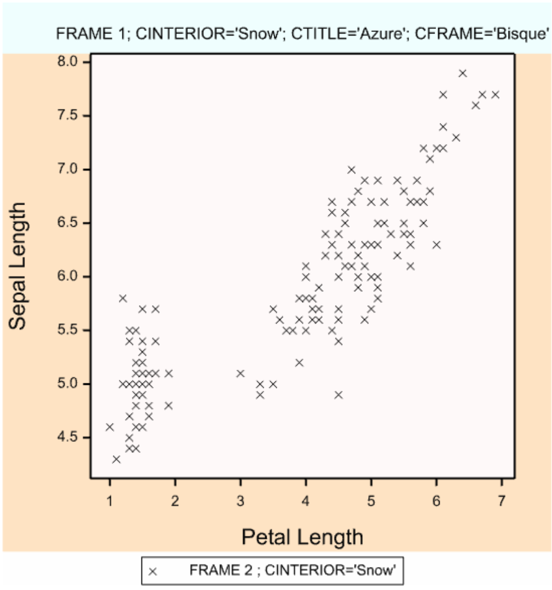
Example 13.1



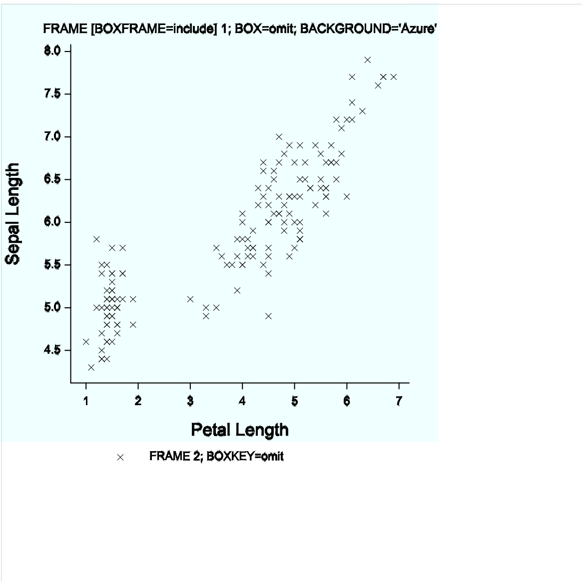
Example 13.2



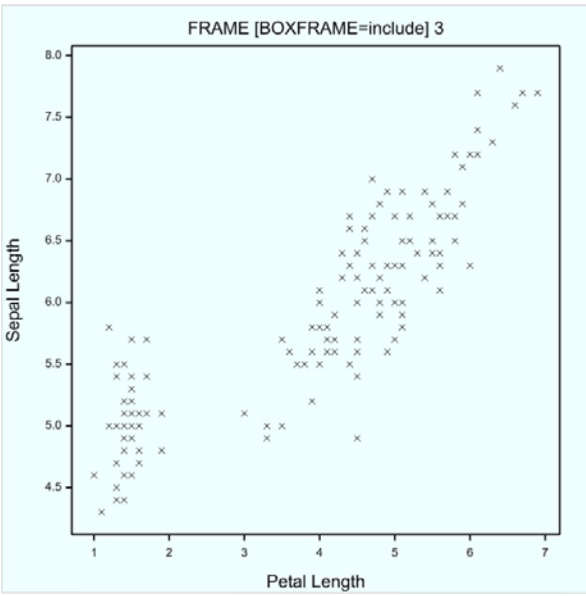
Example 13.3



Example 13.4

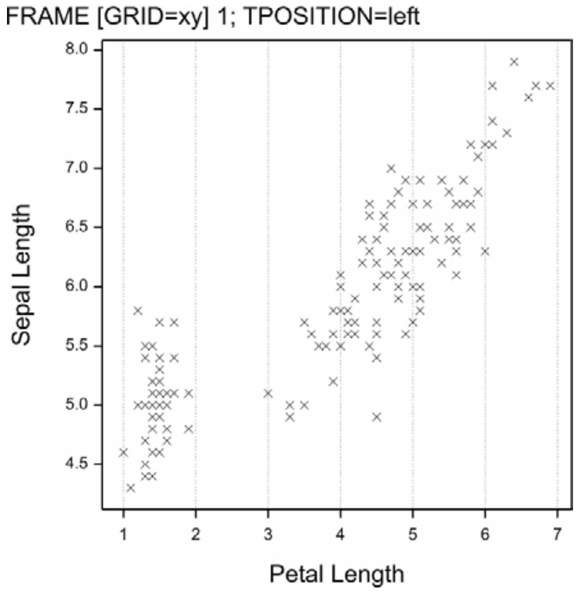


Example 13.5

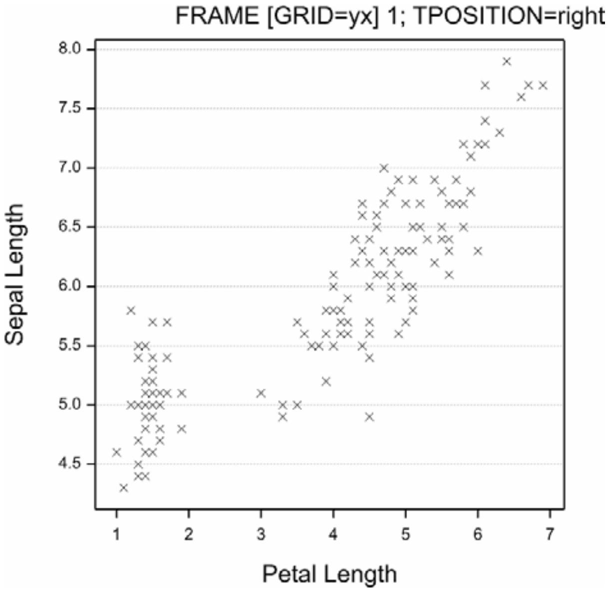


Example 13.6

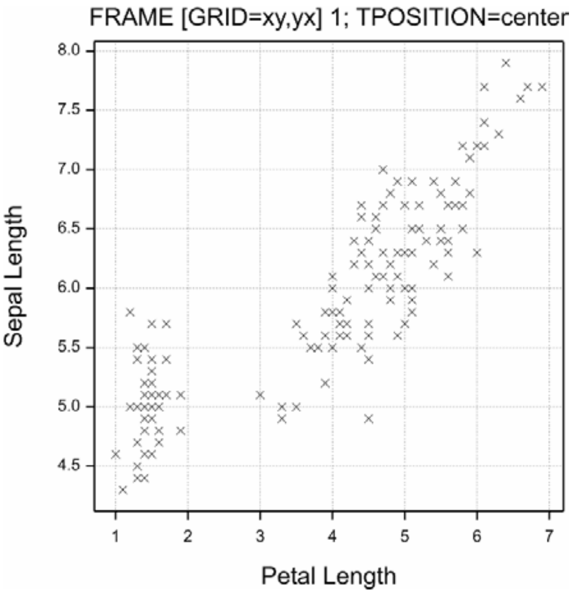
```
JOB       'Example 14
          Directive FRAME: option GRID, parameters SCALING, TPOSITION
                           parameter AXES together with Directive AXIS
          Directives XAXIS and YAXIS: parameters ACTION, LOWER and UPPER'
SPLOAD    '%GENDIR%/Data/Iris.gsh'
XAXIS     1,3 ; TITLE= 'Petal Length'
YAXIS     1,3 ; TITLE= 'Sepal Length'
"Example 14.1"
FRAME     [GRID=xy] 1; TPOSITION=left
DGRAPH    [WINDOW=1; KEYWINDOW=0; \
          TITLE= 'FRAME [GRID=xy] 1; TPOSITION=left'] \
          Sepal_Length ; X=Petal_Length
"Example 14.2"
FRAME     [GRID=yx] 1; TPOSITION=right
DGRAPH    [WINDOW=1; KEYWINDOW=0; \
          TITLE= 'FRAME [GRID=yx] 1; TPOSITION=right'] \
          Sepal_Length ; X=Petal_Length
"Example 14.3"
FRAME     [GRID=xy,yx] 1; TPOSITION=center
DGRAPH    [WINDOW=1; KEYWINDOW=0; \
          TITLE= 'FRAME [GRID=xy,yx] 1; TPOSITION=center'] \
          Sepal_Length ; X=Petal_Length
"Example 14.4"
FRAME     1; SCALING=xyequal
DGRAPH    [WINDOW=1; KEYWINDOW=0; \
          TITLE= 'FRAME [GRID=xy,yx] 1; SCALING=xyequal'] \
          Sepal_Length ; X=Petal_Length
"Example 14.5"
" AXIS example - plot a 30/60/90 set square "
AXIS      lower; LOWER=0; UPPER=2; MARKS=0.5; XZERO=1; YZERO=1;\
          XSTEP=1; YSTEP=0
AXIS      oblique; LOWER=0; UPPER=4; MARKS=0.5; MPOSITION=inside;\
          LPOSITION=inside; XZERO=1; YZERO=1; XSTEP=COS(RADIANS(60));\
          YSTEP=SIN(RADIANS(60))
AXIS      upper; LOWER=0; UPPER=SQRT(16-4); MARKS=0.5; XZERO=3; YZERO=1;\
          XSTEP=0; YSTEP=1
XAXIS     [RESET=yes] 1; LOWER=0; UPPER=5
YAXIS     [RESET=yes] 1; LOWER=0; UPPER=5
FRAME     [RESET=yes] 1; AXES=!p(lower,oblique,upper)
DGRAPH    [WINDOW=1; KEYWINDOW=0; \
          TITLE= 'Oblique axis from 0 to 4'] 1; 1
"Example 14.6"
XAXIS     1; ACTION=hide
YAXIS     1; ACTION=hide
FRAME     1; BOX=omit
DGRAPH    [WINDOW=1; KEYWINDOW=0; \
          TITLE= 'Oblique axis from 0 to 4'] 1; 1
STOP
```



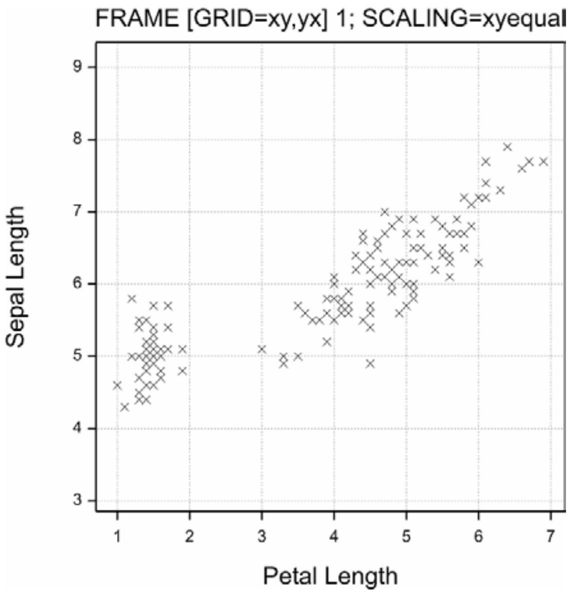
Example 14.1



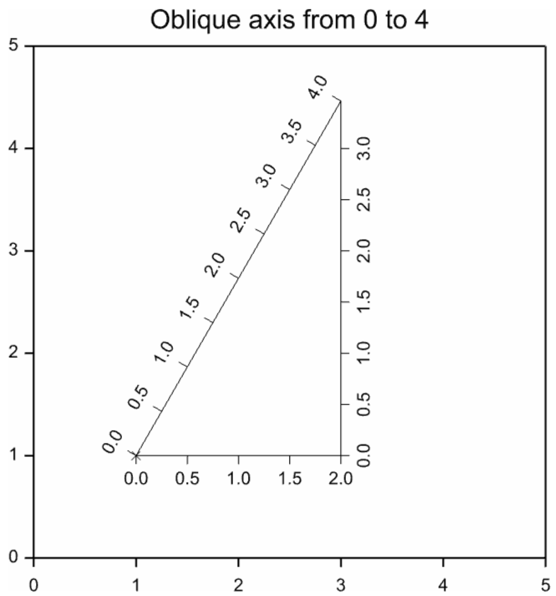
Example 14.2



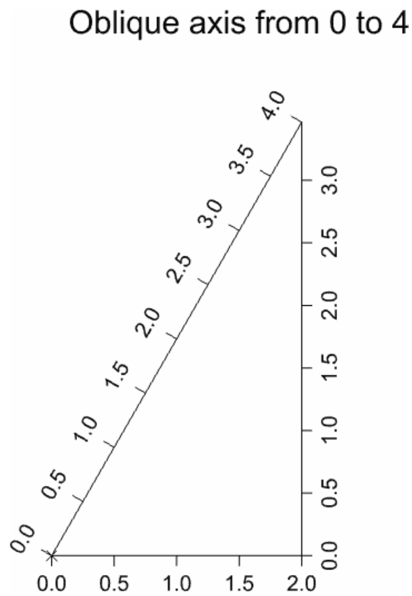
Example 14.3



Example 14.4

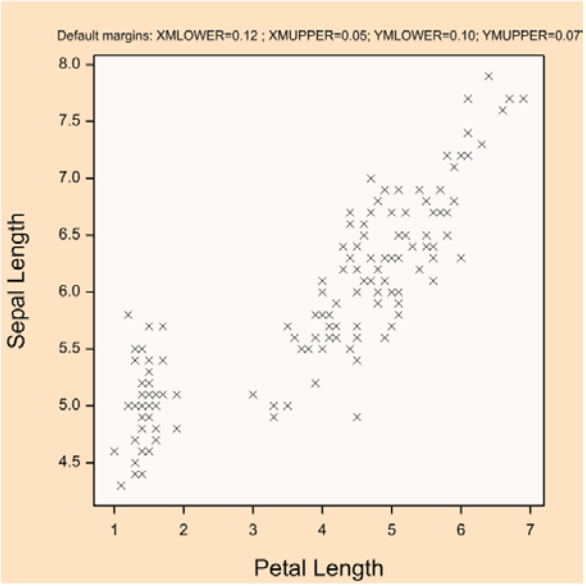


Example 14.5

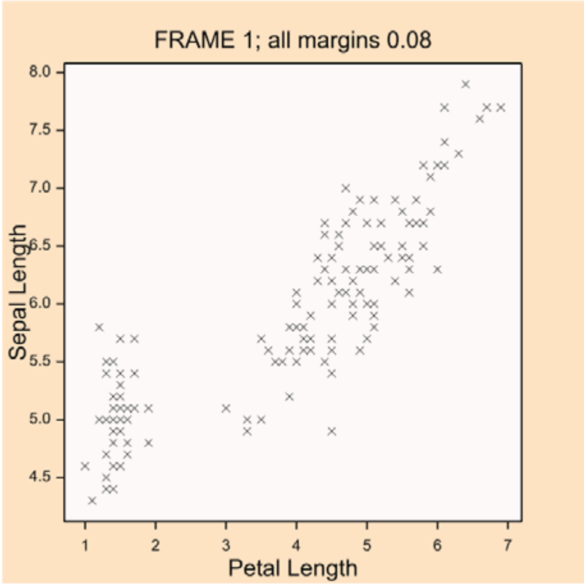


Example 14.6

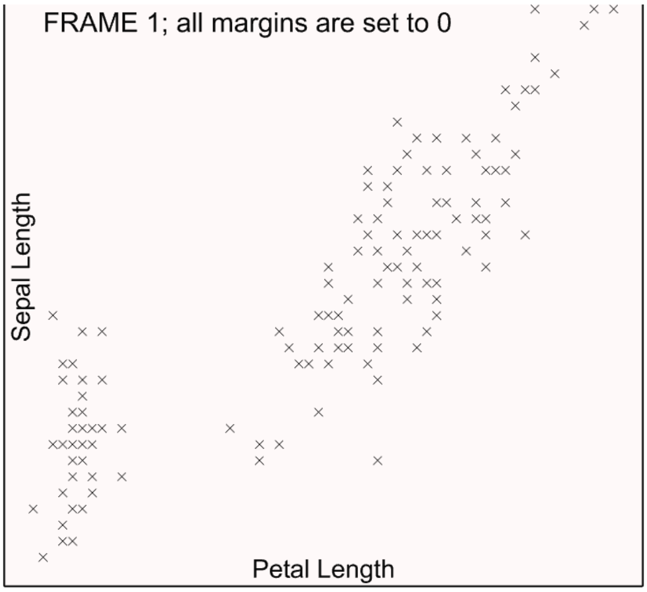
```
JOB       'Example 15
          Directive FRAME: parameters YMLOWER,YMUPPER,XMLOWER,XMUPPER
          Procedure DTEXT
          Directive XAXIS: parameters LDIRECTION and LROTATION'
SPLOAD    '%GENDIR%/Data/Iris.gsh'
XAXIS     1; TITLE= 'Petal Length'
YAXIS     1; TITLE= 'Sepal Length'
"Example 15.1 default margins"
FRAME     1; XMLOWER=0.12 ; XMUPPER=0.05; YMLOWER=0.10; YMUPPER=0.07; \
          CINTERIOR='Snow'; CTITLE='Bisque'; CFRAME='Bisque'
"NOTE: pen size for the title is decreased while the title is too long"
DGRAPH    [WINDOW=1; KEYWINDOW=0 ; TITLE= \
  'Default margins: XMLOWER=0.12 ; XMUPPER=0.05; YMLOWER=0.10; YMUPPER=0.07']\
          Sepal_Length ; X=Petal_Length
"Example 15.2"
FRAME     1; XMLOWER=0.08 ; XMUPPER=0.08; YMLOWER=0.08; YMUPPER=0.08; \
          CINTERIOR='Snow'; CTITLE='Bisque'; CFRAME='Bisque'
DGRAPH    [WINDOW=1; KEYWINDOW=0 ; TITLE= \
          'FRAME 1; all margins 0.08'] Sepal_Length ; X=Petal_Length
"Example 15.3"
FRAME     1; XMLOWER=0 ; XMUPPER=0; YMLOWER=0; YMUPPER=0; \
          CINTERIOR='Snow'; CTITLE='Bisque'; CFRAME='Bisque'
DGRAPH    [WINDOW=1; KEYWINDOW=0 ; TITLE= \
          'FRAME 1; all margins 0'] Sepal_Length ; X=Petal_Length
PEN       1; SIZE=1.6
DTEXT     7.6; X=1.1; TEXT='FRAME 1; all margins are set to 0'
"Example 15.4"
FRAME     [RESET=yes] 1
CALCULATE Petal_Length = 1000*Petal_Length
XAXIS     1; MARKS=500
DGRAPH    [WINDOW=1; KEYWINDOW=0 ; TITLE= \
          'Note: The labels of the x-axis are automatically decreased']\
          Sepal_Length ; X=Petal_Length
"Example 15.5"
XAXIS     1; MARKS=500; LDIRECTION=perpendicular
DGRAPH    [WINDOW=1; KEYWINDOW=0 ; TITLE= \
          'LDIRECTION of XAXIS is set to perpendicular'] \
          Sepal_Length ; X=Petal_Length
"Example 15.6"
XAXIS     [RESET=yes] 1; TITLE= 'Petal Length'; MARKS=500 ; LROTATION=30 ; \
          PENLABELS=4
DGRAPH    [WINDOW=1; KEYWINDOW=0 ; TITLE= \
          'Parameter LROTATION of XAXIS'] Sepal_Length ; X=Petal_Length
STOP
```



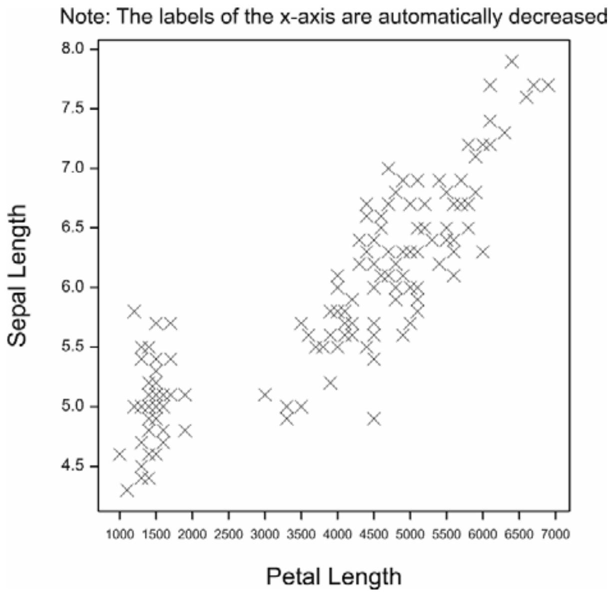
Example 15.1



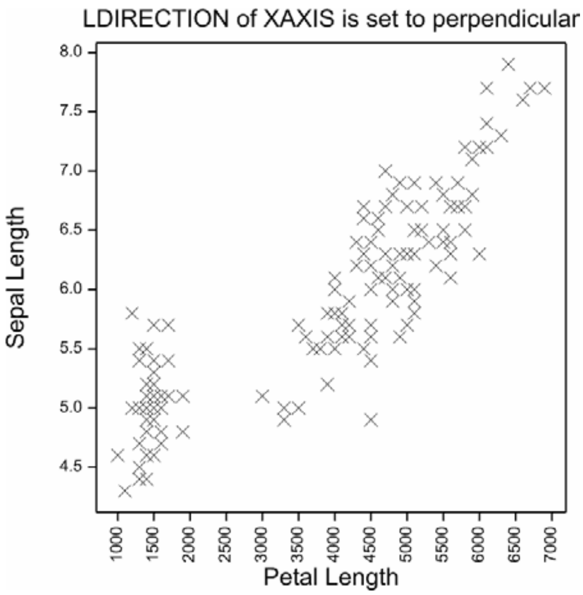
Example 15.2



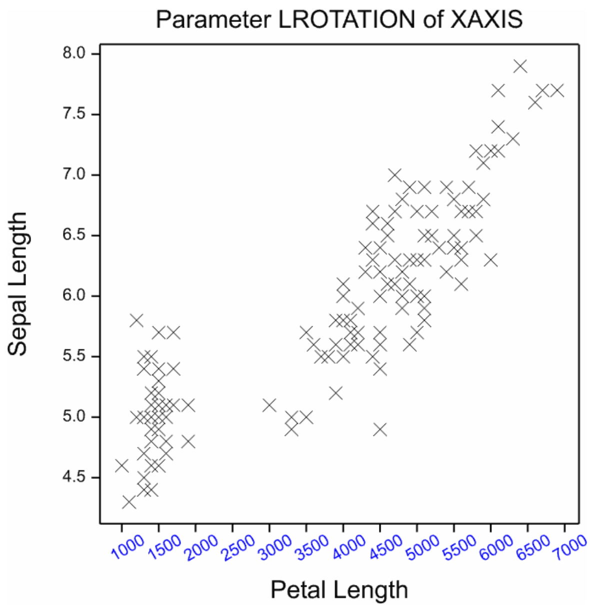
Example 15.3



Example 15.4

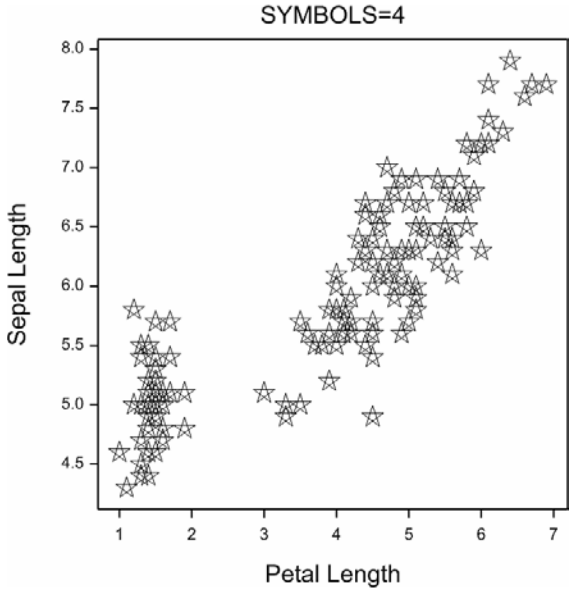


Example 15.5

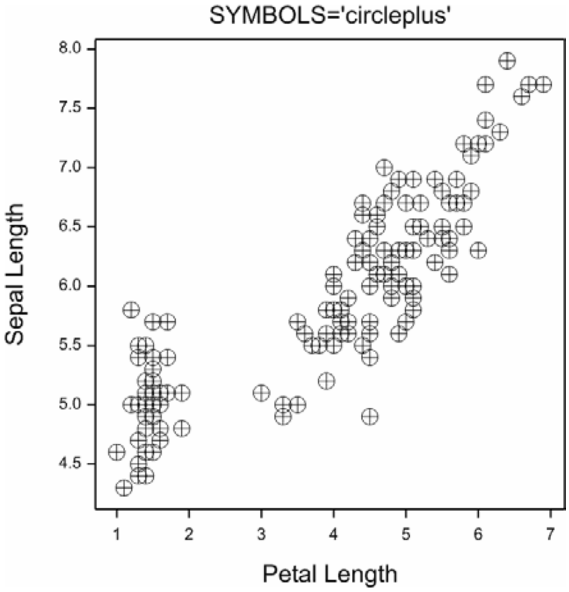


Example 15.6

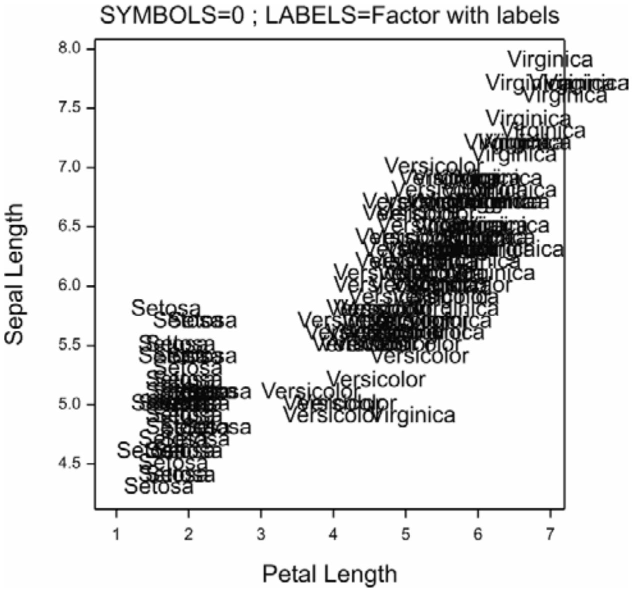
```
JOB       'Example 16
          Directive PEN: parameter SYMBOLS and LABELS'
SPLOAD    '%GENDIR%/Data/Iris.gsh'
XAXIS     1,3 ; TITLE= 'Petal Length'
YAXIS     1,3 ; TITLE= 'Sepal Length'
"Example 16.1"
PEN       1; SYMBOLS=4; SIZE=2
DGRAPH    [WINDOW=1; KEYWINDOW=0; \
          TITLE= 'SYMBOLS=4'] \
          Sepal_Length ; X=Petal_Length
"Example 16.2"
"The text settings of SYMBOLS can be found at the left of Figure 4.9.8a "
PEN       [RESET=yes] 1; SYMBOLS='circleplus'; SIZE=2
DGRAPH    [WINDOW=1; KEYWINDOW=0; \
          TITLE= 'SYMBOLS=''circleplus'''] \
          Sepal_Length ; X=Petal_Length
"Example 16.3"
PEN       1; SYMBOLS=0 ; LABELS=Species; SIZE=1.4
DGRAPH    [WINDOW=1; KEYWINDOW=0; \
          TITLE= 'SYMBOLS=0 ; LABELS=Factor with labels'] \
          Sepal_Length ; X=Petal_Length
"Example 16.4"
FACTOR    [LEVELS=3] SpeciesLev
CALCULATE SpeciesLev = Species
PEN       [RESET=yes] 1; SYMBOLS=0 ; LABELS=SpeciesLev; SIZE=2
DGRAPH    [WINDOW=1; KEYWINDOW=0; \
          TITLE= 'SYMBOLS=0 ; LABELS=Factor with levels'] \
          Sepal_Length ; X=Petal_Length
"Example 16.5"
TXCONSTR  [TEXT=tSpecies] Species
CONCATEN  tSpecies ; WIDTH=4
PEN       [RESET=yes] 1; SYMBOLS=0 ; LABELS=tSpecies; SIZE=1.4
DGRAPH    [WINDOW=1; KEYWINDOW=0; \
          TITLE= 'SYMBOLS=0 ; LABELS=text'] \
          Sepal_Length ; X=Petal_Length
"Example 16.6"
GETRGB    !(2...5); RGB=rgb
MATRIX    [ROWS=2; COLUMNS=2] mat ; VALUES=rgb
PEN       [RESET=yes] 1; SYMBOLS=mat; SIZE=2
DGRAPH    [WINDOW=1; KEYWINDOW=0; \
          TITLE= 'SYMBOLS=matrix to supply a bitmap'] \
          Sepal_Length ; X=Petal_Length
STOP
```



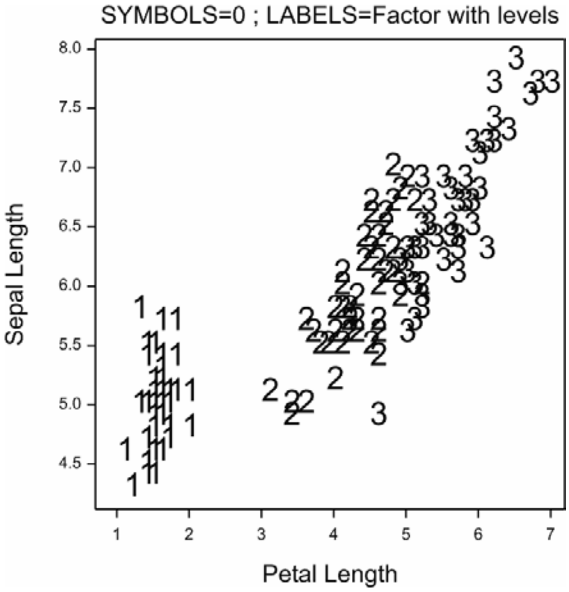
Example 16.1



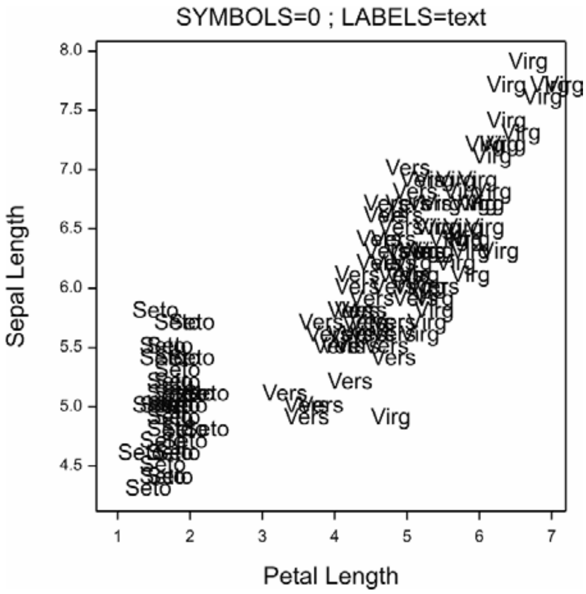
Example 16.2



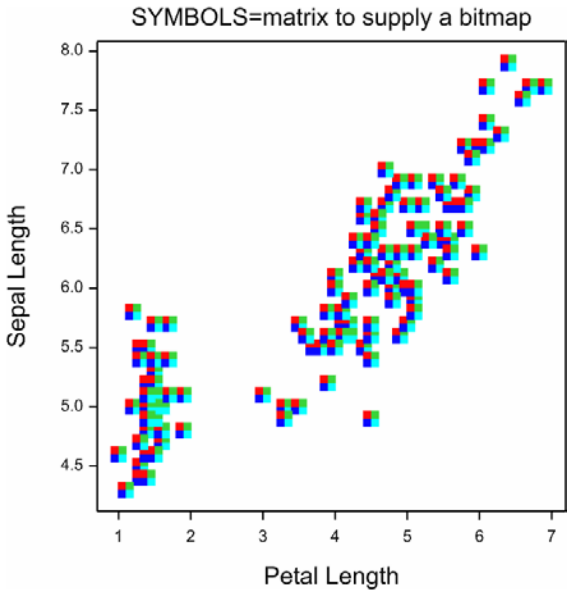
Example 16.3



Example 16.4

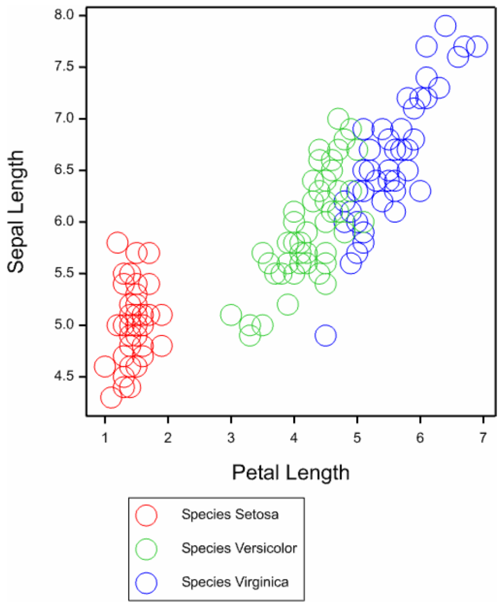


Example 16.5

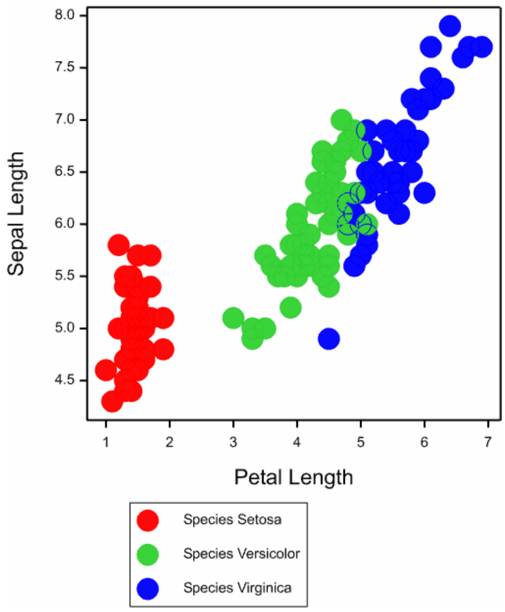


Example 16.6

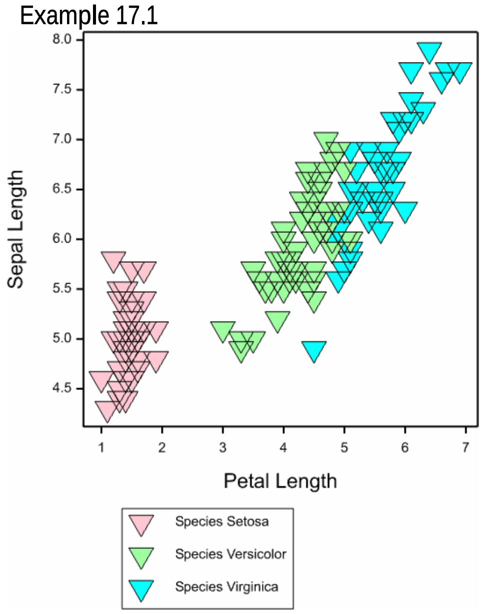
```
JOB       'Example 17
          Directive PEN: parameters COLOUR and CFILL
          Procedure GETRGB
          Function RGB'
SPLOAD    '%GENDIR%/Data/Iris.gsh'
XAXIS     1,3 ; TITLE= 'Petal Length'
YAXIS     1,3 ; TITLE= 'Sepal Length'
"Example 17.1 Specifying colours with colour names"
PEN       1...3; SYMBOLS=2; COLOUR= 'red', 'limegreen','blue' ; SIZE=3
DGRAPH    Sepal_Length ; X=Petal_Length; PEN=Species; DESCRIPTION= \
          !t('Species Setosa','Species Versicolor','Species Virginica')
"Example 17.2 Specifying colours with RGB values from procedure GETRGB "
" GETRGB gets the RGB values of the initial colours of the pens "
GETRGB    2...4; RGB=cred,climegreen,cblue "RGB values: 16711680, 3329330, 255"
PEN       1...3; SYMBOLS=2; COLOUR= cred, climegreen,cblue; \
          CFILL=cred,climegreen,cblue
DGRAPH    Sepal_Length ; X=Petal_Length; PEN=Species; DESCRIPTION= \
          !t('Species Setosa','Species Versicolor','Species Virginica')
"Example 17.3 Specifying colours with RGB values calculated by RGB function"
" The RGB function gets the RGB values of the colour names in paragraph 4.9.9 "
CALCULATE cpink,cpalegreen,caqua = RGB('pink','palegreen','aqua')
PEN       1...3; SYMBOLS=8; COLOUR=0; CFILL=cpink,cpalegreen,caqua
DGRAPH    Sepal_Length ; X=Petal_Length; PEN=Species; DESCRIPTION= \
          !t('Species Setosa','Species Versicolor','Species Virginica')
"Example 17.4 Specifying colours with RGB function directly (see CFILL)"
PEN       1...3; SYMBOLS=8; COLOUR=16711680, 3329330, 255; \ red,limegreen,blue
          CFILL=RGB('pink','palegreen','aqua')
DGRAPH    Sepal_Length ; X=Petal_Length; PEN=Species; DESCRIPTION= \
          !t('Species Setosa','Species Versicolor','Species Virginica')
"Example 17.5"
PEN       1...3; SYMBOLS=5; COLOUR=RGB(255,50,0 ; 0,205,0 ; 0,50,255); \
          CFILL=RGB(255 ; 255 ; 0)
DGRAPH    Sepal_Length ; X=Petal_Length; PEN=Species; DESCRIPTION= \
          !t('Species Setosa','Species Versicolor','Species Virginica')
"Example 17.6"
VARIATE   [VALUES= -1, 1,  1, -1, -1] square[1]
&         [VALUES= -1, -1, 1, 1, -1] square[2]
VARIATE   [VALUES= 3(-1, -.5 ... 1), (-1, 1, *)2] vert[1]
&         [VALUES= (-1, 1, *)5, (-1, -1, *), (1, 1, *)] vert[2]
CALCULATE vert[1]$[3, 6...15] = 0/0
VARIATE   [VALUES= (-1, 1, *)5, (-1, -1, *), (1, 1, *)] hori[1]
&         [VALUES= 3(-1, -.5 ... 1), (-1, 1, *)2] hori[2]
CALCULATE hori[1]$[3, 6...15] = 0/0
VARIATE   [VALUES= #vert[1], #hori[1]] verthori[1]
VARIATE   [VALUES= #vert[2], #hori[2]] verthori[2]
" The size of the self-made symbol square is larger than the size of the
 predefined symbol 5 "
PEN       [RESET=y] 1...3; SYMBOLS= square, vert, verthori; \
          COLOUR=RGB(255,50,0 ; 0,205,0 ; 0,50,255) ; SIZE=2
DGRAPH    Sepal_Length ; X=Petal_Length; PEN=Species; DESCRIPTION= \
          !t('Species Setosa','Species Versicolor','Species Virginica')
STOP
```



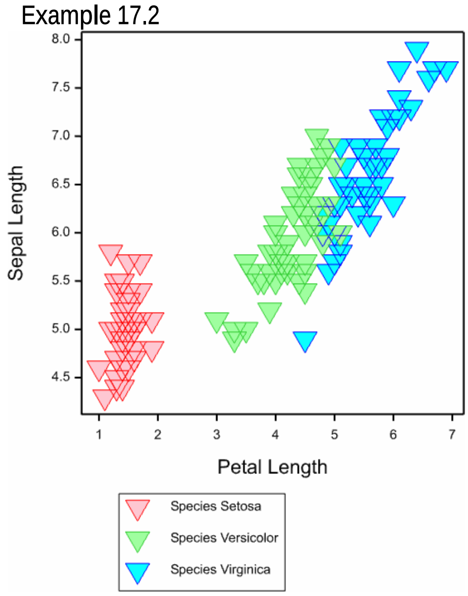
Example 17.1



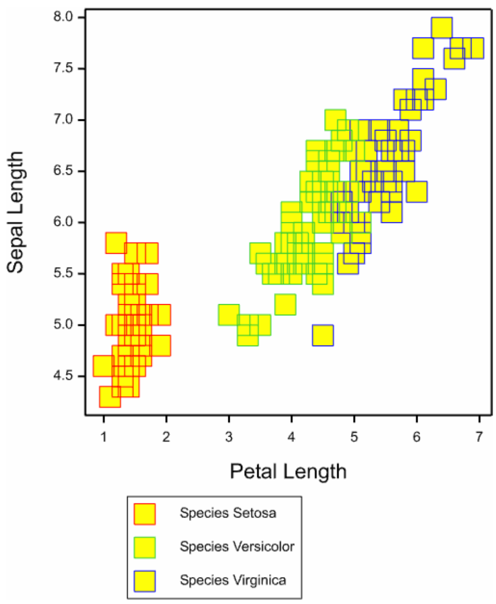
Example 17.2



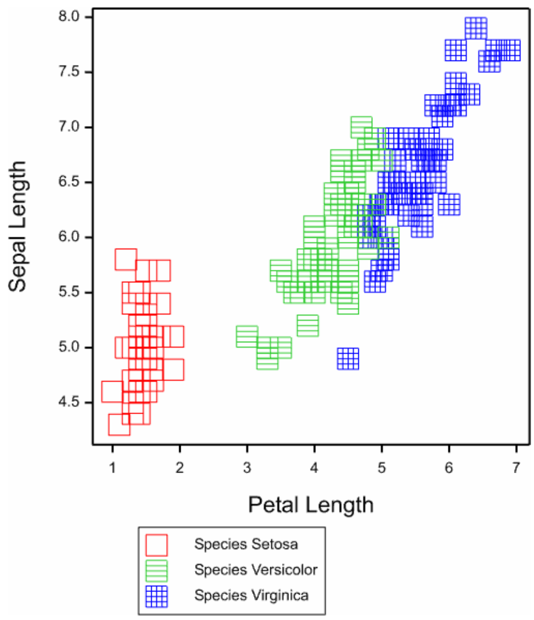
Example 17.3



Example 17.4

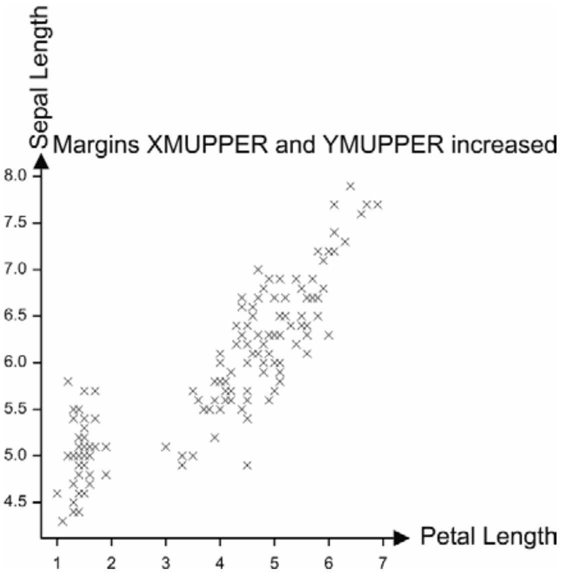


Example 17.5

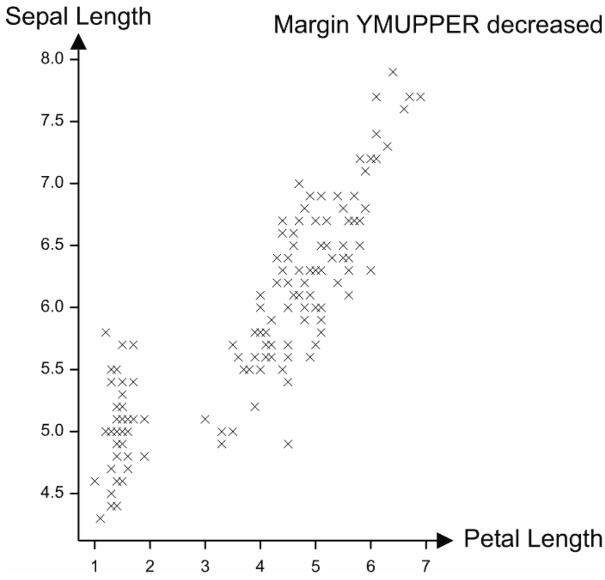


Example 17.6

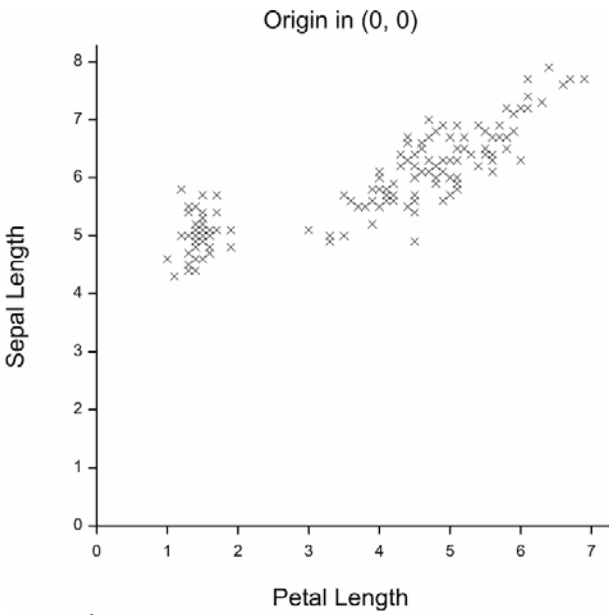
```
JOB       'Example 18
          Directives XAXIS and YAXIS: parameters TPOSITION, TDIRECTION, ARROW
                                      parameters XORIGIN, YORIGIN
                                      parameters MPOSITION, LPOSITION'
SPLOAD    '%GENDIR%/Data/Iris.gsh'
"Example 18.1"
FRAME     1; XMUPPER=0.20; YMUPPER=0.20; BOX=omit; TPOSITION=right
XAXIS     1 ; TITLE= 'Petal Length'; TPOSITION=end; ARROW=include
YAXIS     1 ; TITLE= 'Sepal Length'; TPOSITION=end; ARROW=include
DGRAPH    [WINDOW=1; KEYWINDOW=0; \
          TITLE= 'Margins XMUPPER and YMUPPER increased'] \
          Sepal_Length ; X=Petal_Length
"Example 18.2"
FRAME     1; YMUPPER=0.06; TPOSITION=right
YAXIS     1 ; TDIRECTION=perpendicular
DGRAPH    [WINDOW=1; KEYWINDOW=0; TITLE= 'Margin YMUPPER decreased'] \
          Sepal_Length ; X=Petal_Length
"Example 18.3"
FRAME     [RESET=y] 1; YMLOWER= 0.08; YMUPPER= 0.05; XMLOWER= 0.09; \
          XMUPPER= 0 ; BOX=omit
YAXIS     [RESET=y] 1 ; XORIGIN= 0 ; TITLE= 'Sepal Length'
XAXIS     [RESET=y] 1 ; YORIGIN= 0 ; TITLE= 'Petal Length'
DGRAPH    [WINDOW= 1; KEYWINDOW= 0; TITLE= 'Origin in (0, 0)'] \
          Y= Sepal_Length; X= Petal_Length
"Example 18.4"
YAXIS     [RESET=yes] 1 ; XORIGIN= 1 ; TITLE= 'Sepal Length' ; LOWER=4
XAXIS     [RESET=yes] 1 ; YORIGIN= 4 ; TITLE= 'Petal Length' ; LOWER=1
DGRAPH    [WINDOW= 1; KEYWINDOW= 0; TITLE= 'Origin in (1, 4)'] \
          Y= Sepal_Length; X= Petal_Length
"Example 18.5"
YAXIS     [RESET=yes] 1 ; XORIGIN= 4 ; TITLE= 'Sepal Length'
XAXIS     [RESET=yes] 1 ; YORIGIN= 6 ; TITLE= 'Petal Length'
DGRAPH    [WINDOW= 1; KEYWINDOW= 0; TITLE= 'Origin in (4, 6)'] \
          Y= Sepal_Length; X= Petal_Length
"Example 18.6"
FRAME     [RESET=y] 1; XMLOWER= 0.05; XMUPPER= 0.10 ; BOX=omit
YAXIS     [RESET=yes] 1 ; XORIGIN= 8 ; TITLE= 'Sepal Length'; UPPER=9
XAXIS     [RESET=yes] 1 ; YORIGIN= 9 ; TITLE= 'Petal Length'; \
          MPOSITION= inside; LPOSITION= inside; UPPER=8
DGRAPH    [WINDOW= 1; KEYWINDOW= 0; TITLE= 'Origin in (8, 9)'] \
          Y= Sepal_Length; X= Petal_Length
STOP
```



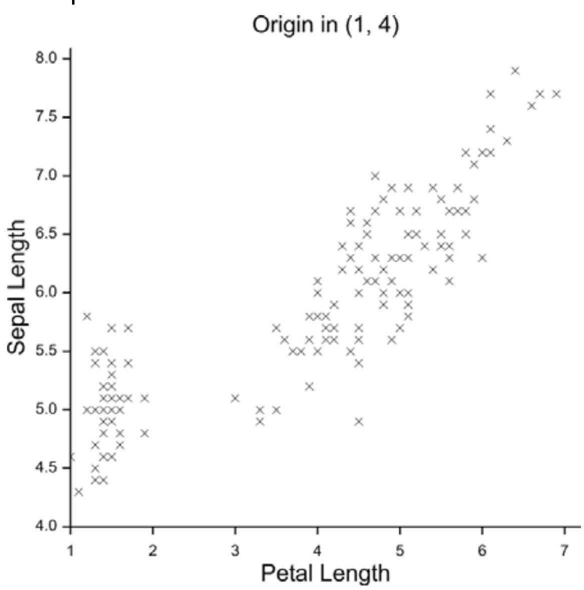
Example 18.1



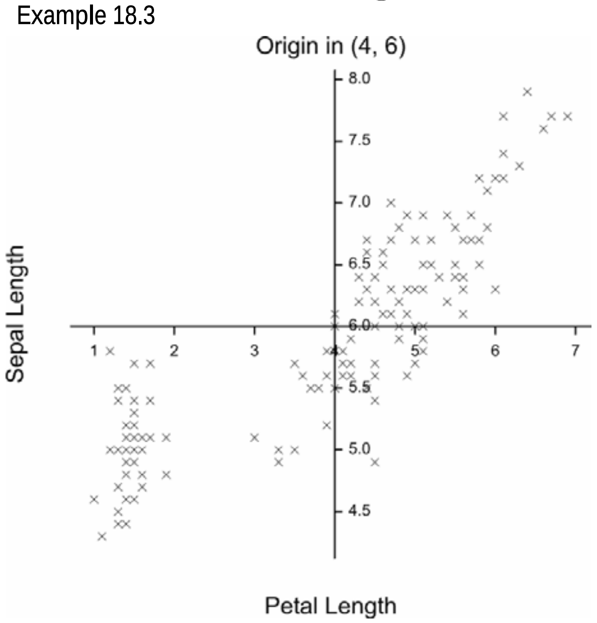
Example 18.2



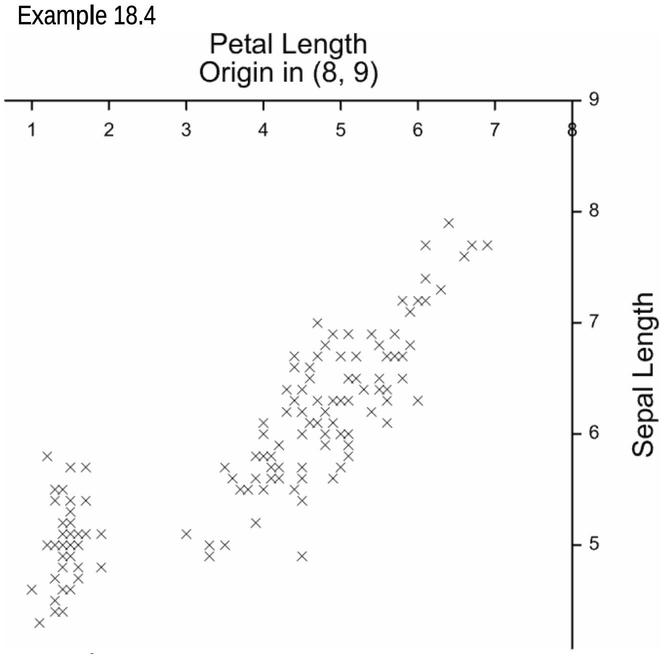
Example 18.3



Example 18.4

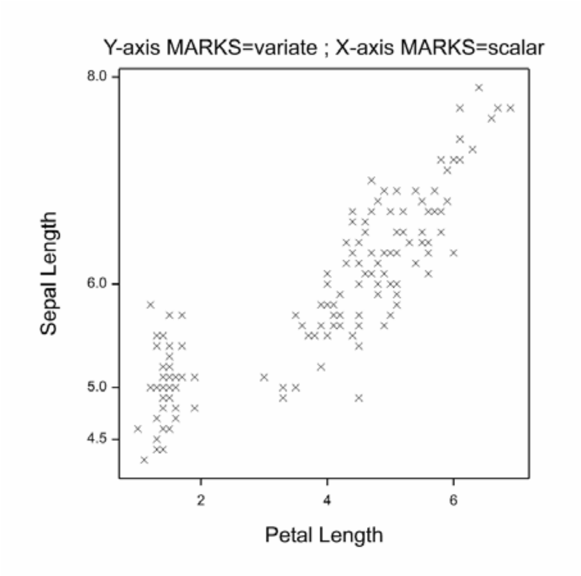


Example 18.5

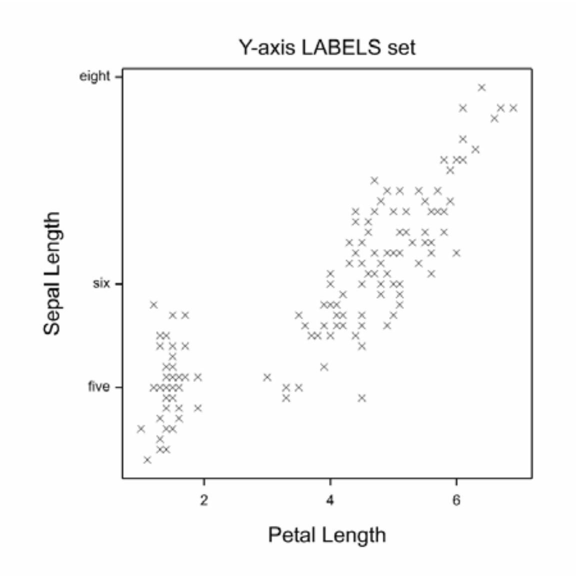


Example 18.6

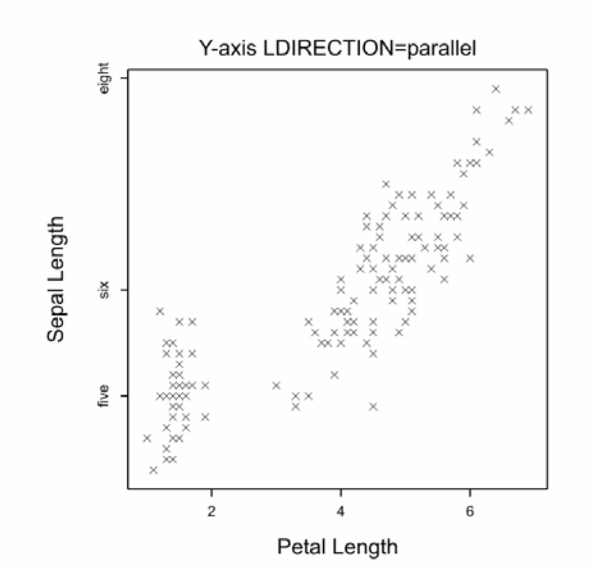
```
JOB       'Example 19
          Directives XAXIS and YAXIS: parameters MARKS, LABELS, MPOSITION
                                      parameters LPOSITION, LDIRECTION
                                      parameters NSUBTICKS, REVERSE'
SPLOAD    '%GENDIR%/Data/Iris.gsh'
"Example 19.1"
XAXIS     1; TITLE= 'Petal Length'; MARKS=2
YAXIS     1; TITLE= 'Sepal Length'; MARKS=!(4.5,5,6,8)
DGRAPH    [WINDOW=1; KEYWINDOW=0; \
          TITLE= 'Y-axis MARKS=variate ; X-axis MARKS=scalar '] \
          Sepal_Length ; X=Petal_Length
"Example 19.2"
XAXIS     1; TITLE= 'Petal Length'; MARKS=2
YAXIS     1; TITLE= 'Sepal Length'; MARKS=!(5,6,8); LABELS=!t(five,six,eight)
DGRAPH    [WINDOW=1; KEYWINDOW=0; \
          TITLE= 'Y-axis LABELS set'] Sepal_Length ; X=Petal_Length
"Example 19.3"
YAXIS     1; MARKS=!(4,5,6,8); LABELS=!t(four,five,six,eight); LDIRECTION=para
DGRAPH    [WINDOW=1; KEYWINDOW=0; \
          TITLE= 'Y-axis LDIRECTION=parallel'] Sepal_Length ; X=Petal_Length
"Example 19.4"
XAXIS     1; MARKS=2; MPOSITION=across
YAXIS     1; MARKS=!(4,5,6,8); LABELS=!t(four,five,six,eight); MPOSITION=inside
DGRAPH    [WINDOW=1; KEYWINDOW=0; \
          TITLE= 'Y-axis MPOSITION=inside ; X-axis MPOSITION=across'] \
          Sepal_Length ; X=Petal_Length
"Example 19.5"
XAXIS     1; MARKS=2; NSUBTICKS=1; MPOSITION=outside
YAXIS     [RESET=y] 1; TITLE= 'Sepal Length'; NSUBTICKS=4; MPOSITION=outside
DGRAPH    [WINDOW=1; KEYWINDOW=0; \
          TITLE= 'Y-axis NSUBTICKS=4 ; X-axis NSUBTICKS=1'] \
          Sepal_Length ; X=Petal_Length
"Example 19.6"
YAXIS     1; REVERSE=yes
DGRAPH    [WINDOW=1; KEYWINDOW=0; \
          TITLE= 'Y-axis REVERSE=yes'] Sepal_Length ; X=Petal_Length
STOP
```



Example 19.1

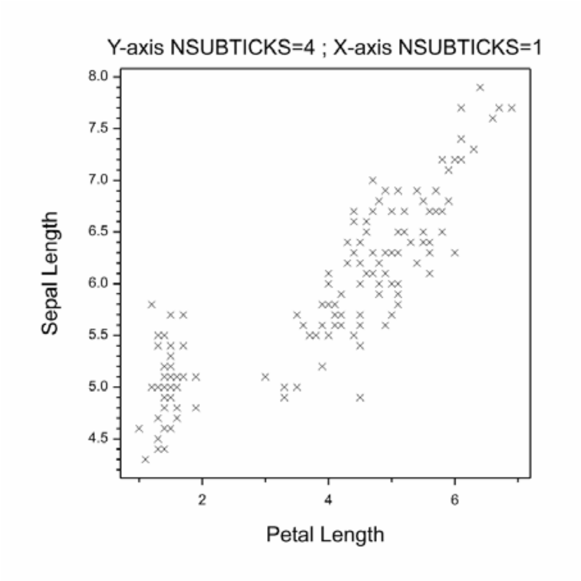


Example 19.2

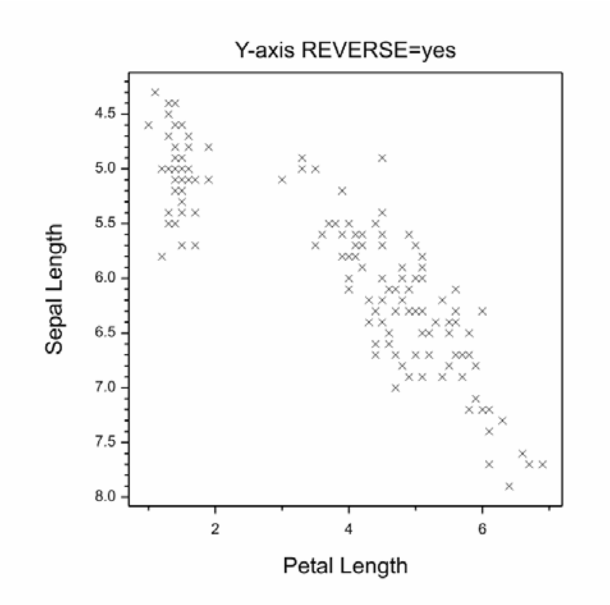


Example 19.3

Example 19.4

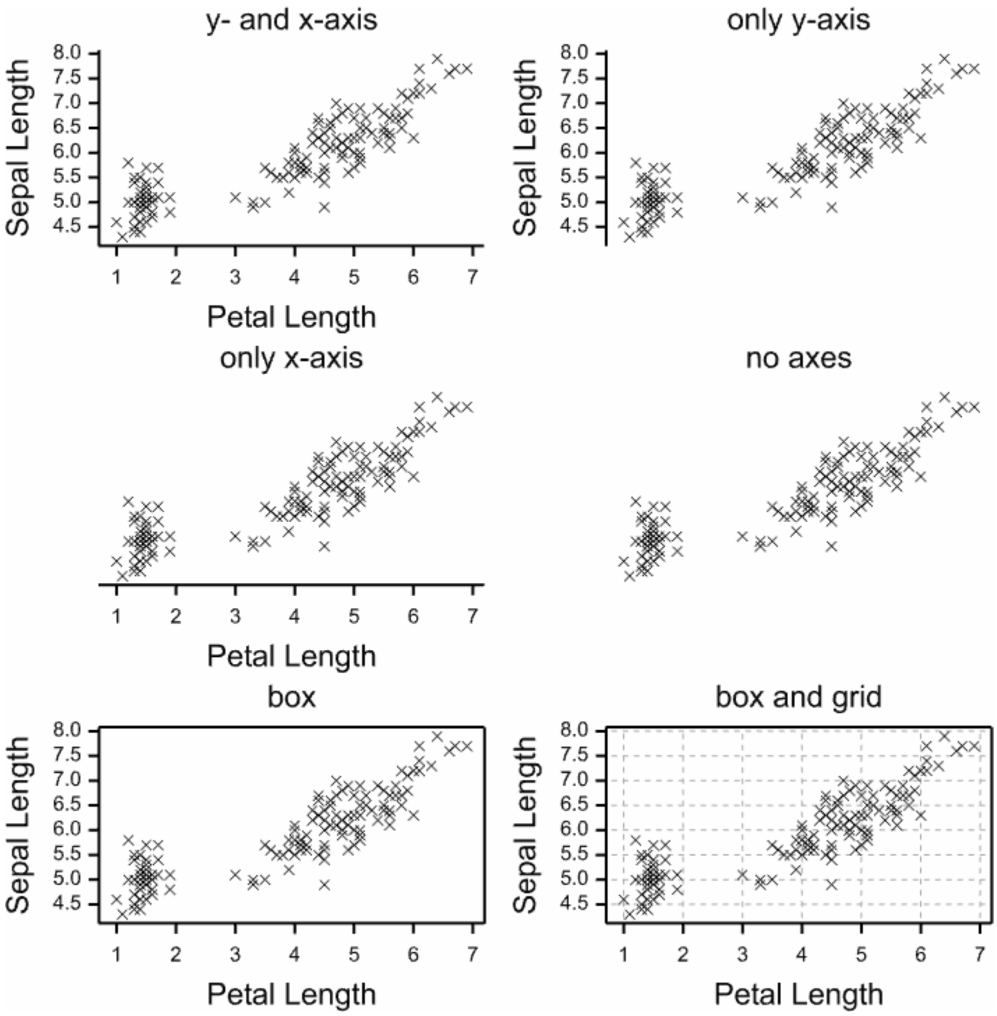


Example 19.5



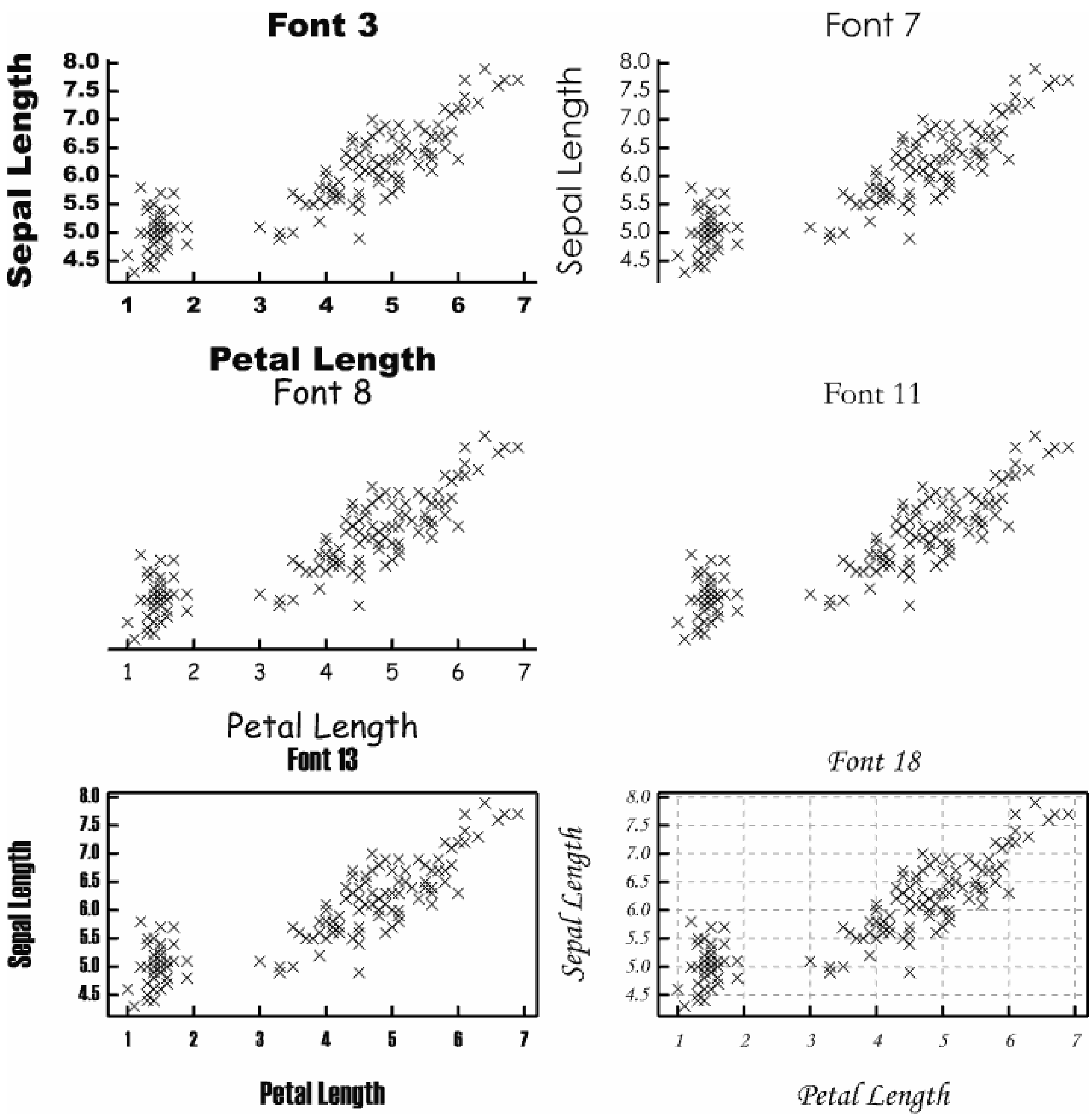
Example 19.6

```
JOB      'Example 20
         Directive FRAME: parameters YLOWER, YUPPER, XLOWER and XUPPER
         Directives XAXIS and YAXIS: parameter ACTION
         Directive DGRAPH: option SCREEN'
SPLOAD   'C:/Program Files/Gen15ed/Data/Iris.gsh'
FRAME    WINDOW= 1; YLOWER= 0.667;  YUPPER= 1;     XLOWER= 0;   XUPPER= 0.5
&        WINDOW= 2; YLOWER= 0.667;  YUPPER= 1;     XLOWER= 0.5; XUPPER= 1
&        WINDOW= 3; YLOWER= 0.333;  YUPPER= 0.667; XLOWER= 0;   XUPPER= 0.5
&        WINDOW= 4; YLOWER= 0.333;  YUPPER= 0.667; XLOWER= 0.5; XUPPER= 1
&        WINDOW= 5; YLOWER= 0;      YUPPER= 0.333; XLOWER= 0;   XUPPER= 0.5
&        WINDOW= 6; YLOWER= 0;      YUPPER= 0.333; XLOWER= 0.5; XUPPER= 1
" NOTE:  the settings of YLOWER, YUPPER, XLOWER and XUPPER in previous FRAME
         commands remain unchanged in the next FRAME commands"
FRAME    1...6; YMLOWER=0.09; YMUPPER=0.05; XMLOWER=0.10; XMUPPER=0.02
FRAME    1...4; BOX=omit
FRAME    [GRID=xy,yx] 6
XAXIS    WINDOW=1...6; TITLE='Petal Length'; ACTION=(display, hide)2, (display)2
YAXIS    WINDOW=1...6; TITLE='Sepal Length'; ACTION=2(display, hide), (display)2
PEN      -4; LINESTYLE=7     " Default pen number for drawing the grid "
DGRAPH   [WINDOW=1; KEYWINDOW=0; TITLE= 'y- and x-axis'] \
         Y= Sepal_Length; X= Petal_Length
&        [WINDOW=2; SCREEN= keep; TITLE= 'only y-axis'] \
         Y= Sepal_Length; X= Petal_Length
&        [WINDOW=3; TITLE='only x-axis']  Y= Sepal_Length; X= Petal_Length
&        [WINDOW=4; TITLE='no axes']      Y= Sepal_Length; X= Petal_Length
&        [WINDOW=5; TITLE='box']          Y= Sepal_Length; X= Petal_Length
&        [WINDOW=6; TITLE='box and grid'] Y= Sepal_Length; X= Petal_Length
STOP
```



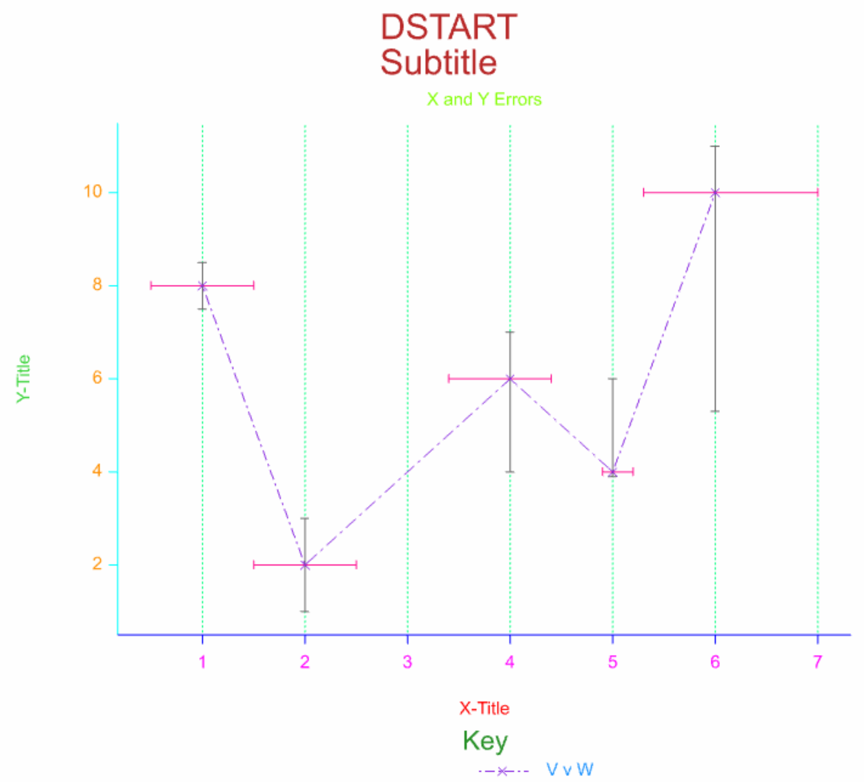
Example 20

```
JOB     'Example 21
        Procedure FFRAME: the same 6 windows as those in Example 15
        Directive PEN: parameter FONT (see all fonts in paragraph 4.9.8)'
SPLOAD  'C:/Program Files/Gen15ed/Data/Iris.gsh'
FFRAME  [ROWS= 3; COLUMNS= 2; MARGIN= xtitle, ytitle] ngraphs; SSCREEN= screen
FRAME   1...4; BOX= omit
FRAME   5; BOX= include
FRAME   [GRID=xy,yx] 6; BOX= include
XAXIS   WINDOW= 1...6; TITLE= 'Petal Length'; MPOSITION= inside ; \
        ACTION= (display, hide)2, (display)2
YAXIS   WINDOW= 1...6; TITLE= 'Sepal Length'; MPOSITION= inside ; \
        ACTION= 2(display, hide), (display)2
PEN     -4; LINESTYLE= 7
FOR     ii= 1...ngraphs; jj= 3, 7, 8, 11, 13, 18
  TXCONST [TEXT=ss]  'Font ',jj
  PEN      -1, -2, -3, -5; FONT= jj
  DGRAPH  [WINDOW= ii; KEYWINDOW= 0; TITLE= ss; SCREEN= #screen[ii] ] \
           Y= Sepal_Length; X= Petal_Length
ENDFOR
STOP
```



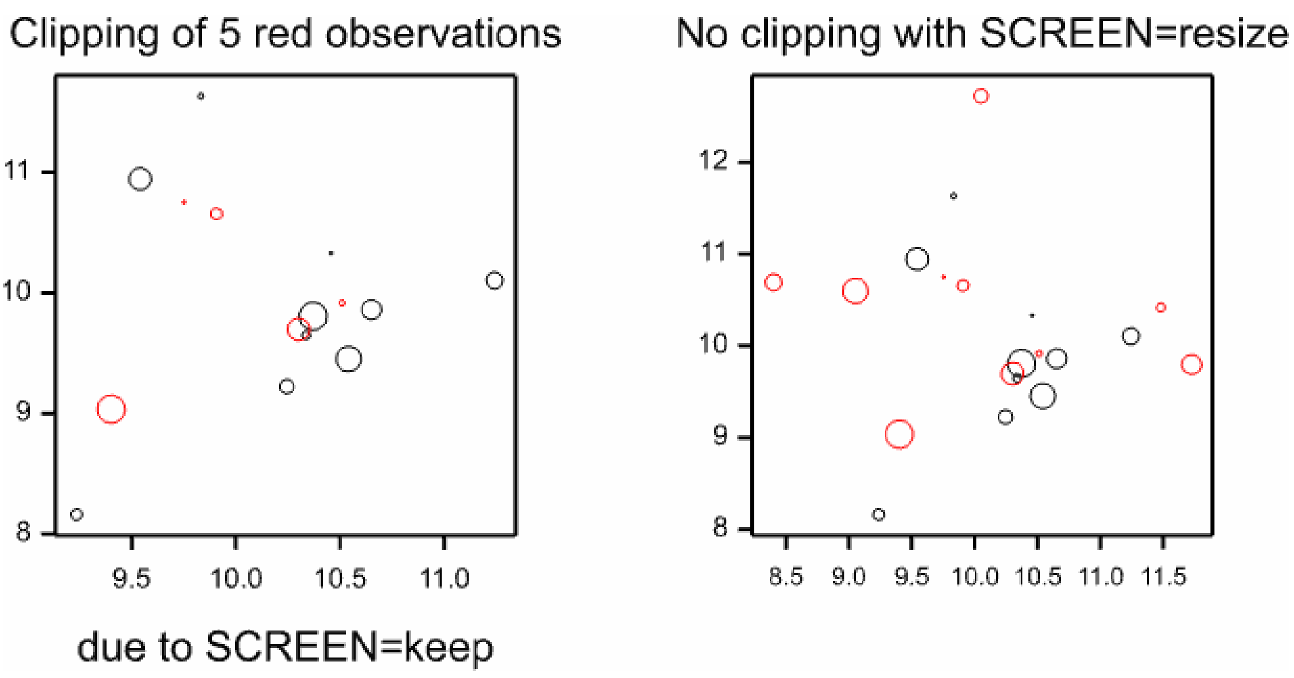
Example 21

```
JOB     'Example 22
     Different pens for different parts of a graph: see the table in Appendix 1'
VARIATE V ; VALUES=!(2,6,4, 8,10)
VARIATE W ; VALUES=!(2,4,5, 1, 6)
VARIATE YL; VALUES=!(1,4,3.9,7.5,5.3)
VARIATE YH; VALUES=!(3,7,6,8.5,11)
VARIATE XL; VALUES=!(1.5,3.4,4.9,0.5,5.3)
VARIATE XH; VALUES=!(2.5,4.4,5.2,1.5,7)
FRAME   1,2; XLOWER=0,0.5; XUPPER=1
XAXIS   1; TITLE= 'X-Title'; PENTITLE= 2 ; PENAXIS= 4 ; PENLABELS= 6
YAXIS   1; TITLE= 'Y-Title'; PENTITLE= 3 ; PENAXIS= 5 ; PENLABELS= 8
FRAME   [GRID=xy] 1; PENTITLE= 9; PENGRID=10; BOX=omit
FRAME   2; PENKEY=11; BOXKEY=omit
PEN     10; LINESTYLE=2
PEN     12; LINESTYLE=4; METHOD=line; JOIN=ascending
PEN     -5, -12; COLOUR= 'Forestgreen','Firebrick'
DSTART  [TITLE= 'DSTART ~{break}Subtitle']
" In this example the following pens are used:
  pen 2 and 3 for plotting the titles along X- and Y-axis,
  pen 4 and 5 for plotting the X-axis and Y-axis,
  pen 6 and 8 for plotting the labels along X- and Y-axis,
  pen 9, 10 and 11 for plotting the title, grid and key
  pen 12 for plotting the points and line,
  pen 13 for plotting the x-bars,
  pen 14 for plotting the y-bars.
  pen -12 for plotting the overall title
  The default colours of the pens are used except the colours of pen -5 and -12"
DGRAPH  [WINDOW=1; KEYWINDOW=2; TITLE='X and Y Errors' ; \
        KEYDESCRIPTION= 'Key'] Y=V; X=W; YLOWER=YL; YUPPER=YH; XLOWER=XL; \
        XUPPER=XH; PEN=12; XBARPEN=13; YBARPEN=14
STOP
```



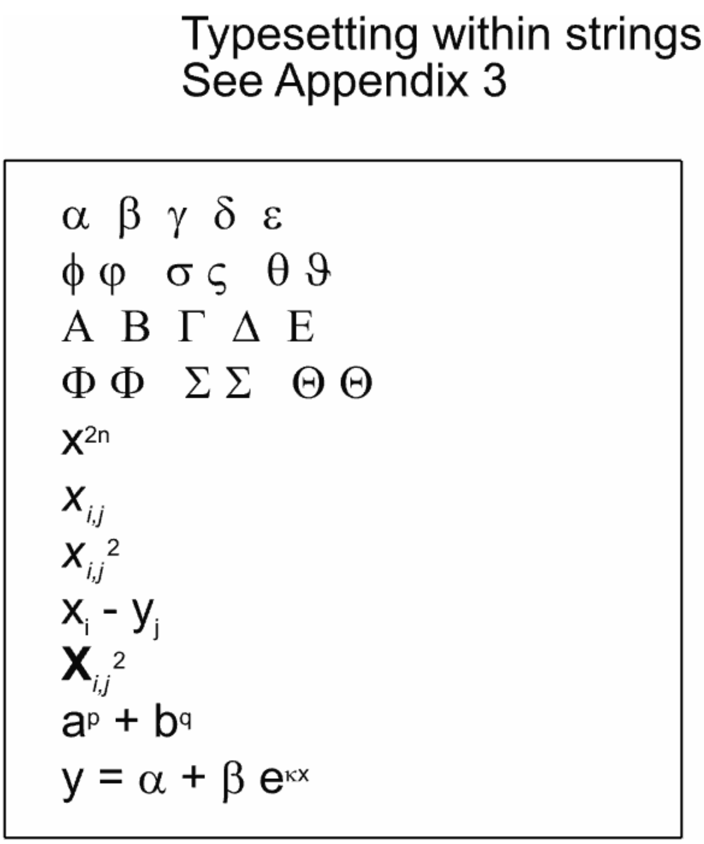
Example 22

```
JOB          'Example 23
             Directive DGRAPH: option SCREEN
                       with setting RESIZE no clipping'
CALCULATE    u = URAND(12345;1)
CALCULATE    x1,y1 = GRNORMAL(10;10;1)
CALCULATE    size = !(1...10)*2/10
XAXIS        5; TITLE='due to SCREEN=keep' ; MARK=0.5
YAXIS        5; MARK=1
PEN          1,2 ; SIZE=size ; SYMBOLS=2
DGRAPH       [WINDOW=5; KEYWINDOW=0; \
             TITLE='Clipping of 5 red observations'] y1; X=x1; PEN=1
CALCULATE    x2,y2 = GRNORMAL(10;10;1)
DGRAPH       [WINDOW=5; KEYWINDOW=0; SCREEN=keep] y2; X=x2; PEN=2
DGRAPH       [WINDOW=6; KEYWINDOW=0; SCREEN=keep; \
             TITLE='No clipping with SCREEN=resize'] y1; X=x1; PEN=1
DGRAPH       [WINDOW=6; KEYWINDOW=0; SCREEN=resize] y2; X=x2; PEN=2
STOP
```



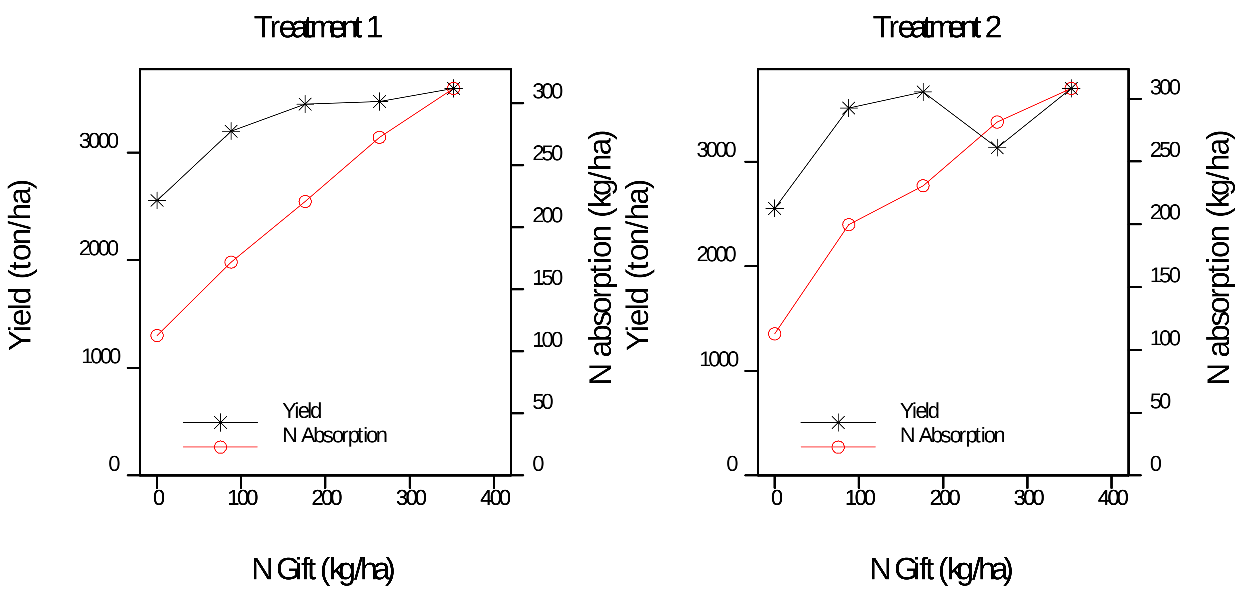
Example 23

```
JOB    'Example 24
       Typesetting within strings. See Appendix 3'
XAXIS  1; ACTION=hide; LOWER= 0; UPPER= 12
YAXIS  1; ACTION=hide; LOWER= 0; UPPER= 12
DSTART [TITLE='Typesetting within strings.~{break}See Appendix 3']
PEN    1; SYMBOLS= 0; \
       LABELS= '~{alpha}  ~{beta}  ~{gamma}  ~{delta}  ~{epsilon}'; SIZE=2
DGRAPH [WINDOW= 1; KEYWINDOW= 0] 11; 1
PEN    1; SYMBOLS= 0; SIZE=2 ; \
       LABELS= '~{phi} ~{varphi}   ~{sigma} ~{varsigma}   ~{theta} ~{vartheta}'
DGRAPH [WINDOW= 1; KEYWINDOW= 0; SCREEN= keep] 10; 1
PEN    1; SYMBOLS= 0; \
       LABELS= '~{Alpha}  ~{Beta}  ~{Gamma}  ~{Delta}  ~{Epsilon}'; SIZE=2
DGRAPH [WINDOW= 1; KEYWINDOW= 0; SCREEN= keep] 9; 1
PEN    1; SYMBOLS= 0; SIZE=2 ; \
       LABELS= '~{Phi} ~{Varphi}   ~{Sigma} ~{Varsigma}   ~{Theta} ~{Vartheta}'
DGRAPH [WINDOW= 1; KEYWINDOW= 0; SCREEN= keep] 8; 1
PEN    1; SYMBOLS= 0; LABELS= 'x~^{2n}'; SIZE=2
DGRAPH [WINDOW= 1; KEYWINDOW= 0; SCREEN= keep] 7; 1
PEN    1; SYMBOLS= 0; LABELS= '~i{x~_{i,j}}'; SIZE=2
DGRAPH [WINDOW= 1; KEYWINDOW= 0; SCREEN= keep] 6; 1
PEN    1; SYMBOLS= 0; LABELS= '~i{x~_{i,j}}~^2'; SIZE=2
DGRAPH [WINDOW= 1; KEYWINDOW= 0; SCREEN= keep] 5; 1
PEN    1; SYMBOLS= 0; LABELS= 'x~_{i} - y~_{j}'; SIZE=2
DGRAPH [WINDOW= 1; KEYWINDOW= 0; SCREEN= keep] 4; 1
PEN    1; SYMBOLS= 0; LABELS= '~b{X}~i{~_{i,j}}~^2'; SIZE=2
DGRAPH [WINDOW= 1; KEYWINDOW= 0; SCREEN= keep] 3; 1
PEN    1; SYMBOLS= 0; LABELS= 'a~^p + b~^q'; SIZE=2
DGRAPH [WINDOW= 1; KEYWINDOW= 0; SCREEN= keep] 2; 1
PEN    1; SYMBOLS= 0; LABELS= 'y = ~{alpha} + ~{beta} e~^{~{kappa}x}'; \
       SIZE=2
DGRAPH [WINDOW= 1; KEYWINDOW= 0; SCREEN= keep] 1; 1
STOP
```



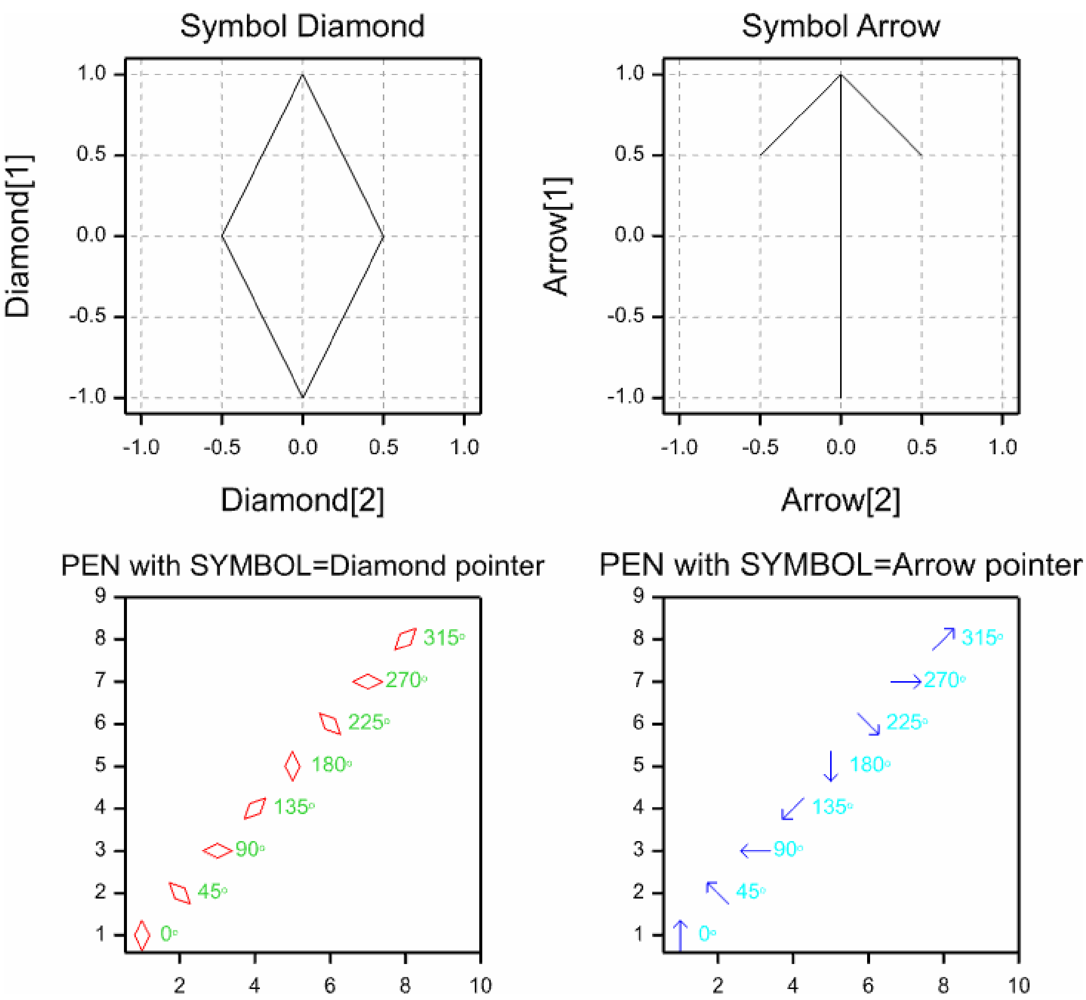
Example 24

```
JOB     'Example 25
         Directive YAXIS: parameter XORIGIN'
FACTOR Treat
READ    Treat, Yld,  TotN, Ngift, Nres
 1     2554    112.8      0    9.2       2     2554    112.8      0    9.2
 1     3198    171.9     88    6.1       2     3513    199.8     88    5.3
 1     3448    220.7    176   11.5       2     3666    230.8    176   11.4
 1     3473    272.4    264   13.4       2     3134    281.5    264   26.3
 1     3595    311.9    352   51.1       2     3701    308.3    352   59.0 :
FRAME   1...4; YLOWER= 0.5; YUPPER= 1; XLOWER= 2(0, 0.5); XUPPER= 2(0.5, 1); \
         XMUPPER= 0.10; XMLOWER= 0.10
FRAME   5...8; YLOWER= 0.55; YUPPER= 0.67, 0.65; XLOWER= 2(0.11, 0.61); \
         XUPPER= 2(0.5, 1); BOXKEY=omit
XAXIS   1...4; LOWER= -20; UPPER= 420; TITLE= 'N Gift (kg/ha)'; YORIGIN= 0; \
         MARKS= 100
YAXIS   1...4; MPOSITION= outside; LPOSITION= outside; \
         LOWER= 0; XORIGIN= (-20, 420)2; MARKS= (1000, 50)2; \
         TITLE= 'Yield (ton/ha)', 'N absorption (kg/ha)'
PEN     1, 2; METHOD= line; SYMBOLS= 9, 2
FOR     ii= 1, 2; jj= 1, 3; kk= 2, 4; ll= 5, 7; mm= 6, 8
  TXCONSTR  [TEXT=title] 'Treatment ', ii
  RESTRICT  Yld, Ngift, TotN; Treat.EQ.ii
  DGRAPH    [WINDOW= jj; KEYWINDOW= ll; SCREEN= keep; \
            TITLE= title] Yld; Ngift; PEN= 1; \
            DESCRIPTION= 'Yield'
  &         [WINDOW= kk; SCREEN= keep; KEYWINDOW= mm] TotN; Ngift; PEN= 2; \
            DESCRIPTION= 'N Absorption'
  RESTRICT  Opbr, Ngift, TotN
ENDFOR
STOP
```



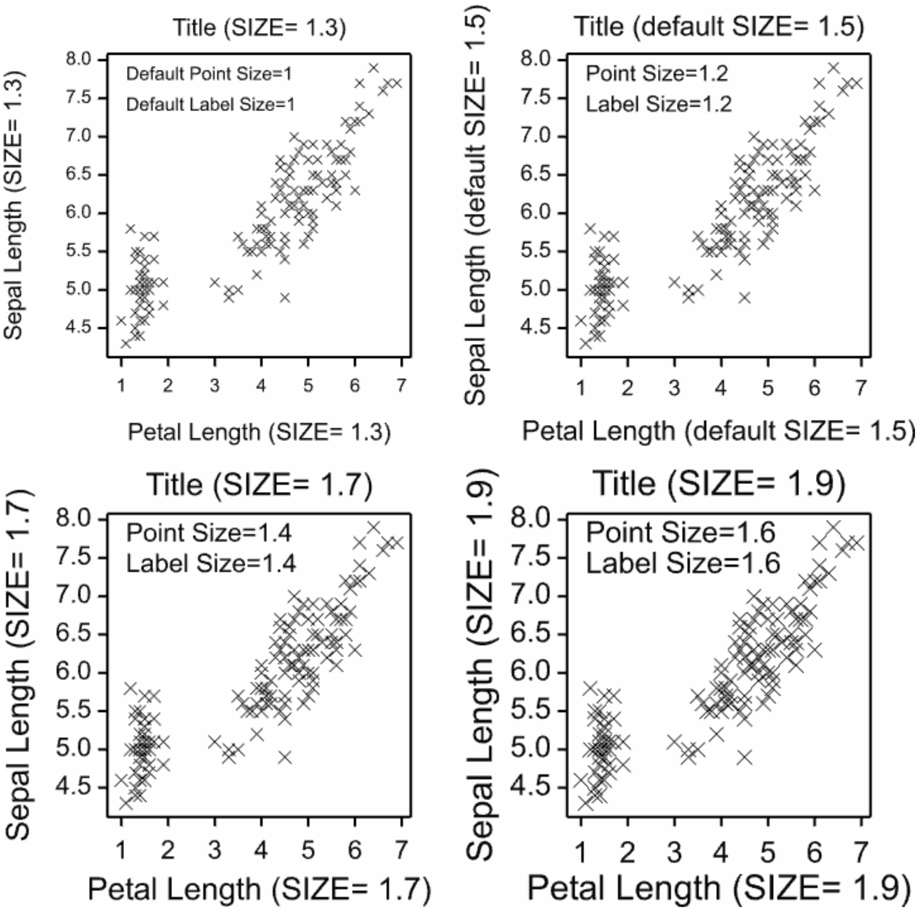
Example 25

```
JOB      'Example 26
         Directive Pen: parameters SYMBOLS=pointer and ROTATION'
FRAME    [GRID=xy,yx] 5, 6
XAXIS    5, 6; LOWER= -1.1; UPPER= 1.1; TITLE= 'Diamond[2]', 'Arrow[2]'; \
         MARKS=0.5
YAXIS    5, 6; LOWER= -1.1; UPPER= 1.1; TITLE= 'Diamond[1]', 'Arrow[1]'; \
         MARKS=0.5
PEN      -4; LINESTYLE= 7 " Default pen number for drawing the grid "
VARIATE  [VALUES= -1,  0,    1,   0,    -1] Diamond[1]
&        [VALUES=  0, -0.5, 0,    0.5,   0] Diamond[2]

VARIATE  [VALUES= -1,  1,   0.5, *, 1, 0.5] Arrow[1]
&        [VALUES=  0,  0,  -0.5, *, 0, 0.5] Arrow[2]
YAXIS    7...8; UPPER=  9
XAXIS    7...8; UPPER= 10
PEN      1; METHOD= line; SYMBOLS= 0; JOIN= given
PEN      2, 4; SYMBOLS= Diamond, Arrow; SIZE= 2; \
         ROTATION= !(0, 45, 90, 135, 180, 225, 270, 315)
PEN      3, 5; SYMBOLS=0; LABELS= !T('   0~^o', '   45~^o', '   90~^o', \
         '   135~^o', '   180~^o', '   225~^o', '   270~^o', '   315~^o')
DGRAPH   [WINDOW= 5; KEYWINDOW= 0; TITLE= 'Symbol Diamond'] \
         Diamond[1]; Diamond[2]
&        [WINDOW= 6; SCREEN= keep; TITLE= 'Symbol Arrow'] \
         Arrow[1]; Arrow[2]
&        [WINDOW= 7; TITLE= 'PEN with SYMBOL=Diamond pointer'] \
         2(!(1...8)); !(1...8); PEN= 2, 3
&        [WINDOW= 8; TITLE= 'PEN with SYMBOL=Arrow pointer'] \
         2(!(1...8)); !(1...8); PEN= 4, 5
STOP
```



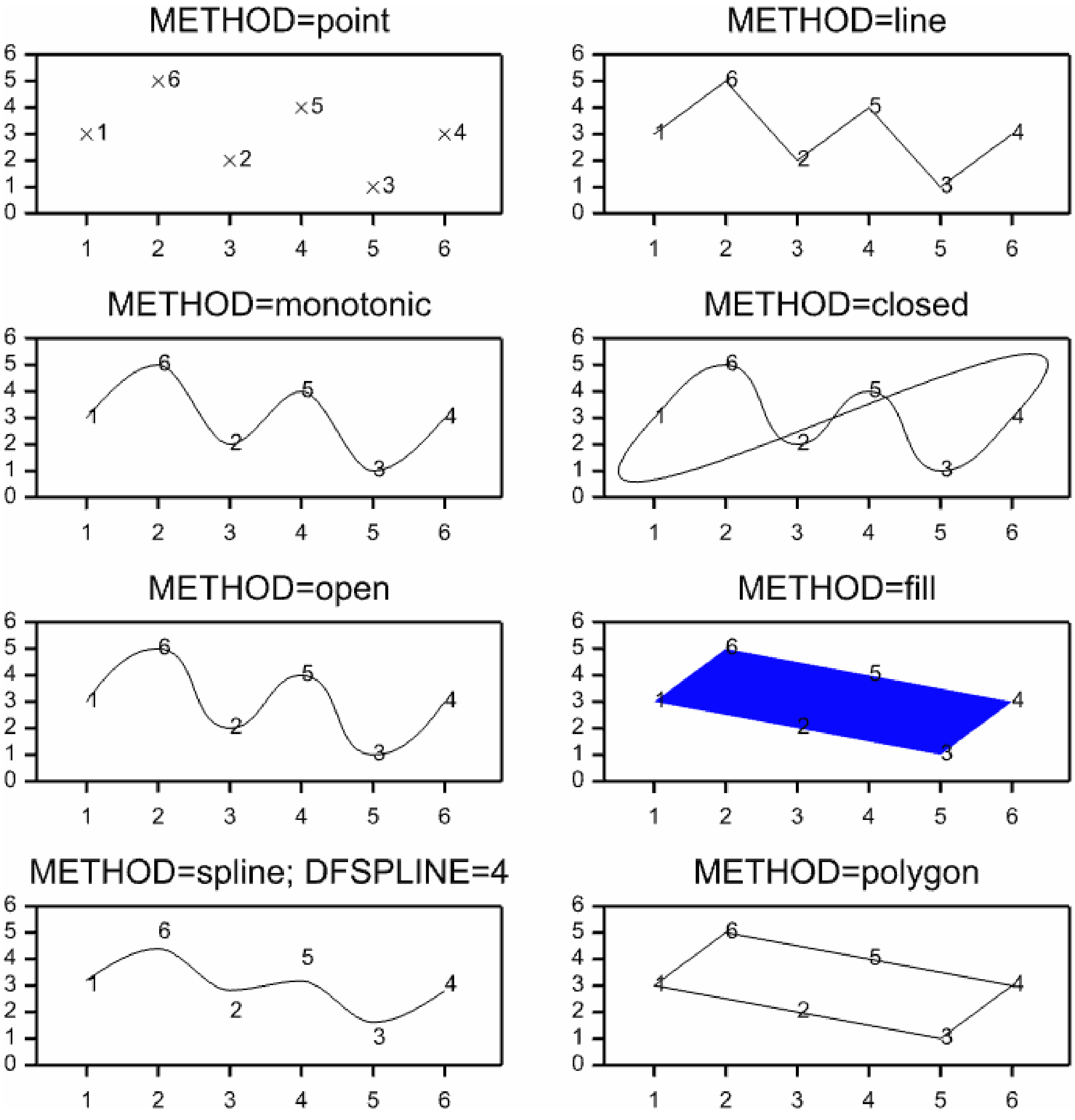
Example 26

```
JOB     'Example 27
        Directive PEN:  parameter SIZE
  NOTE: Sizes of Titles and Labels are decreased if they don''t fit.
        See for example Examples 13.2, 15.1, 15.4 and 19.4'
SPLOAD  'C:/Program Files/Gen15ed/Data/Iris.gsh'
XAXIS   5...8; TITLE= 'Petal Length (SIZE= 1.3)', \
        'Petal Length (default SIZE= 1.5)', 'Petal Length (SIZE= 1.7)', \
        'Petal Length (SIZE= 1.9)' ; PENLABELS= 5...8 ; PENTITLE= 9...12
YAXIS   5...8; TITLE= 'Sepal Length (SIZE= 1.3)', \
        'Sepal Length (default SIZE= 1.5)', 'Sepal Length (SIZE= 1.7)', \
        'Sepal Length (SIZE= 1.9)' ; PENLABELS= 5...8 ; PENTITLE= 9...12
FRAME   5...8 ; PENTITLE=13...16
PEN     NUMBER= 1...16; LINESTYLE= 1; COLOUR= 1; SYMBOLS= 1; \
        SIZE= (1, 1.2, 1.4, 1.6)2,(1.3, 1.5, 1.7, 1.9)2
DGRAPH  [WINDOW= 5; KEYWINDOW= 0; TITLE= 'Title (SIZE= 1.3)'] \
        Y=Sepal_Length ; X=Petal_Length ; PEN= 1
DTEXT   [WINDOW=5] Y=7.8,7.4; X=1.1; TEXT= 'Default Point Size=1',\
        'Default Label Size=1'; PEN=1
DGRAPH  [WINDOW= 6; KEYWINDOW= 0; SCREEN= keep; \
        TITLE= 'Title (default SIZE= 1.5)'] Y=Sepal_Length ; \
        X=Petal_Length ; PEN= 2
DTEXT   [WINDOW=6] Y=7.8,7.4; X=1.1; TEXT= 'Point Size=1.2',\
        'Label Size=1.2'; PEN=2
DGRAPH  [WINDOW= 7; KEYWINDOW= 0; SCREEN= keep; TITLE= 'Title (SIZE= 1.7)'] \
        Y=Sepal_Length ; X=Petal_Length ; PEN= 3
DTEXT   [WINDOW=7] Y=7.8,7.4; X=1.1; TEXT= 'Point Size=1.4',\
        'Label Size=1.4'; PEN=3
DGRAPH  [WINDOW= 8; KEYWINDOW= 0; SCREEN= keep; TITLE= 'Title (SIZE= 1.9)'] \
        Y=Sepal_Length ; X=Petal_Length ; PEN= 4
DTEXT   [WINDOW=8] Y=7.8,7.4; X=1.1; TEXT= 'Point Size=1.6',\
        'Label Size=1.6'; PEN=4
STOP
```



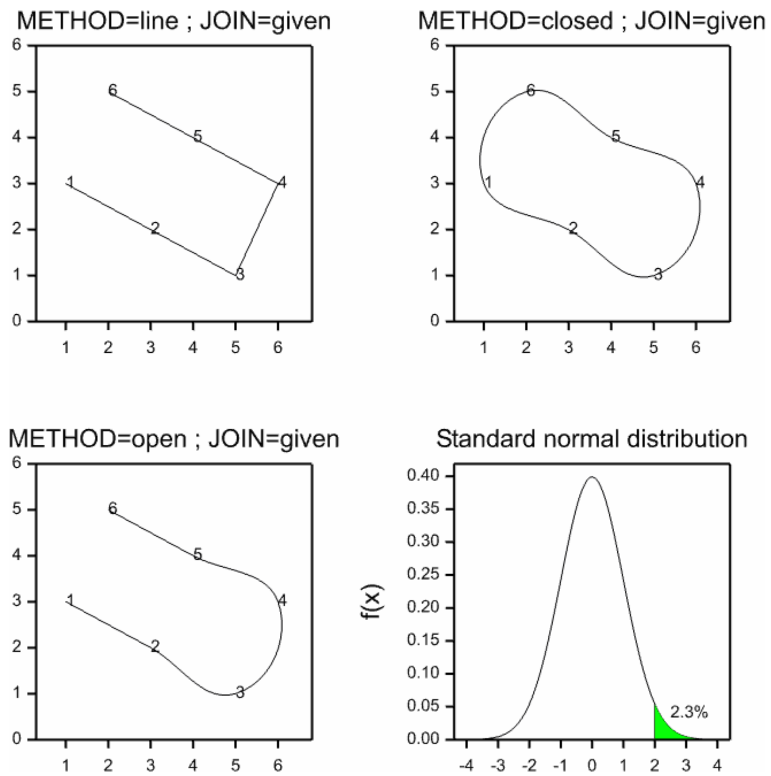
Example 27

```
JOB      'Example 28
         Directive PEN: parameters METHOD, DFSPLINE and CAREA'
VARIATE  [VALUES= 3, 2, 1, 3, 4, 5] y
&        [VALUES= 1, 3, 5, 6, 4, 2] x
FACTOR   [LEVELS= 7; VALUES= 1...6] factname
FRAME    WINDOW=1...8; YLOWER=2(0, 0.25, 0.5, 0.75); \
         YUPPER=2(0.25, 0.5, 0.75, 1); XLOWER=(0, 0.5)4; XUPPER=(0.5, 1)4; \
         XMLOWER= 0.04; YMLOWER= 0.04
YAXIS    WINDOW= 1...8; LOWER=0; UPPER=6
XAXIS    1...8; LOWER= 0.3; UPPER= 6.8
PEN      1...8; METHOD= point, line, monotonic, closed, open, fill, spline, \
         polygon; SYMBOLS= 1,7(0); LABELS=factname; LINESTYLE=1; COLOUR=0
PEN      6; CAREA= 'blue'
PEN      7; DFSPLINE=4  "Default"
DGRAPH   [WINDOW= 7; TITLE= 'METHOD=point'; KEYWINDOW= 0] \
         Y= y; X= x ; PEN= 1
&        [WINDOW= 8; TITLE= 'METHOD=line'; SCREEN= keep] Y= y; X= x; PEN=2
&        [WINDOW= 5; TITLE= 'METHOD=monotonic'] Y= y; X= x; PEN= 3
&        [WINDOW= 6; TITLE= 'METHOD=closed'] Y= y; X= x; PEN= 4
&        [WINDOW= 3; TITLE= 'METHOD=open'] Y= y; X= x; PEN= 5
&        [WINDOW= 4; TITLE= 'METHOD=fill'] Y= y; X= x ; PEN= 6
&        [WINDOW= 1; TITLE= 'METHOD=spline; DFSPLINE=4'] Y= y; X= x; PEN= 7
&        [WINDOW= 2; TITLE= 'METHOD=polygon'] Y= y; X= x; PEN= 8
STOP
```



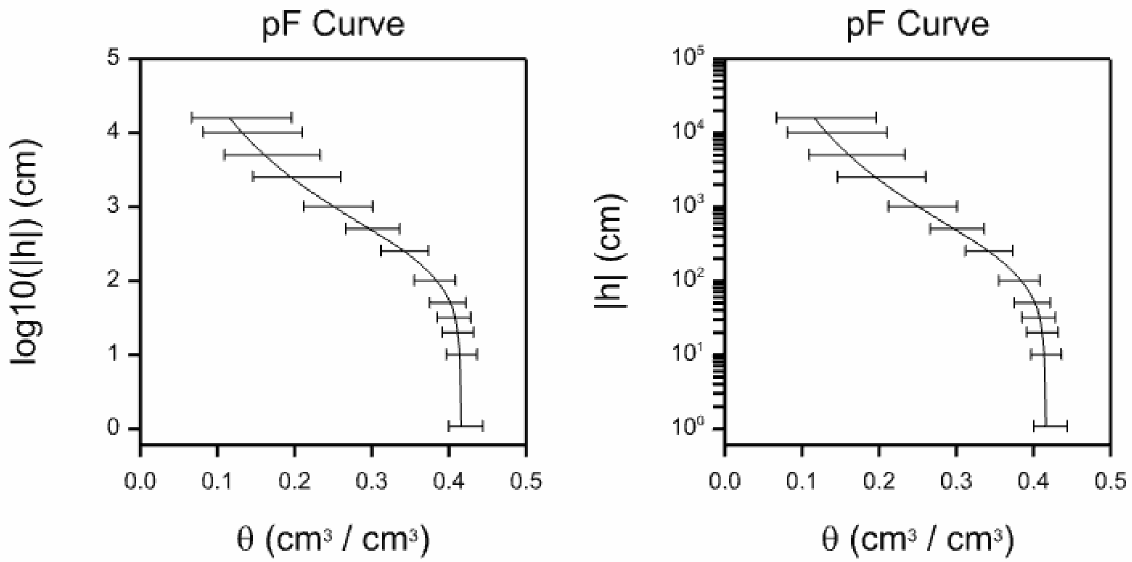
Example 28

```
JOB       'Example 29
          Directive PEN: parameters METHOD and JOIN'
VARIATE   [VALUES= 3, 2, 1, 3, 4, 5] y
&         [VALUES= 1, 3, 5, 6, 4, 2] x
FACTOR    [LEVELS= 6; VALUES= 1...6] factname
YAXIS     5...7; LOWER=0; UPPER=6
XAXIS     5...7; LOWER= 0.3; UPPER= 6.8
PEN       1...3; METHOD= line, closed, open; SYMBOLS= 3(0); LABELS= factname; \
          LINESTYLE= 1; COLOUR= 1; JOIN=given
"NOTE: METHOD=point, monotonic, spline, fiil and polygon ignore JOINT=given"
DGRAPH    [WINDOW= 5; TITLE= 'METHOD=line ; JOIN=given'; KEYWINDOW= 0] \
          Y= y; X= x ; PEN= 1
&         [WINDOW= 6; SCREEN=keep ; TITLE= 'METHOD=closed ; JOIN=given'] \
          Y= y; X= x; PEN= 2
&         [WINDOW= 7; TITLE= 'METHOD=open ; JOIN=given'] Y= y; X= x; PEN= 3

SCALAR    u; VALUE=2  "Modifying this value gives another percentage"
CALCULATE u_1 = u + 0.1
VARIATE   [VALUES= -4, -3.9 ... 4] x
&         [VALUES=  u, u, u_1 ... 4] xfill
CALCULATE z, fu, zfill = 1 / SQRT( 2 * C( 'pi')) * EXP(-0.5 * x, u, xfill ** 2)
&         zfill$[1] = 0
YAXIS     8; LOWER= 0; TITLE= 'f(x)'
PEN       [RESET=y] 1...3; METHOD= monotonic, line, fill; \
          COLOUR= *, 0 "black",'lime'; SYMBOLS= 0; LINESTYLE= 1; \
          JOIN= ascending, (given)2
SCALAR    %GT2; DECIMALS=1
CALCULATE %GT2 = 100*CUNORMAL(u;0;1)
TXCONSTR  [TEXT=label] %GT2,'%'
PEN       4; LABELS= label; SYMBOLS= 0; COLOUR= 0
DGRAPH    [WINDOW=8; KEYWINDOW= 0; SCREEN=keep; \
          TITLE= 'Standard normal distribution'] \
          Y= z, !(0, fu), zfill; X= x, !(u, u), xfill; PEN= 1...3
&         [SCREEN= keep] 0.04; X=2.5; PEN= 4
STOP
```



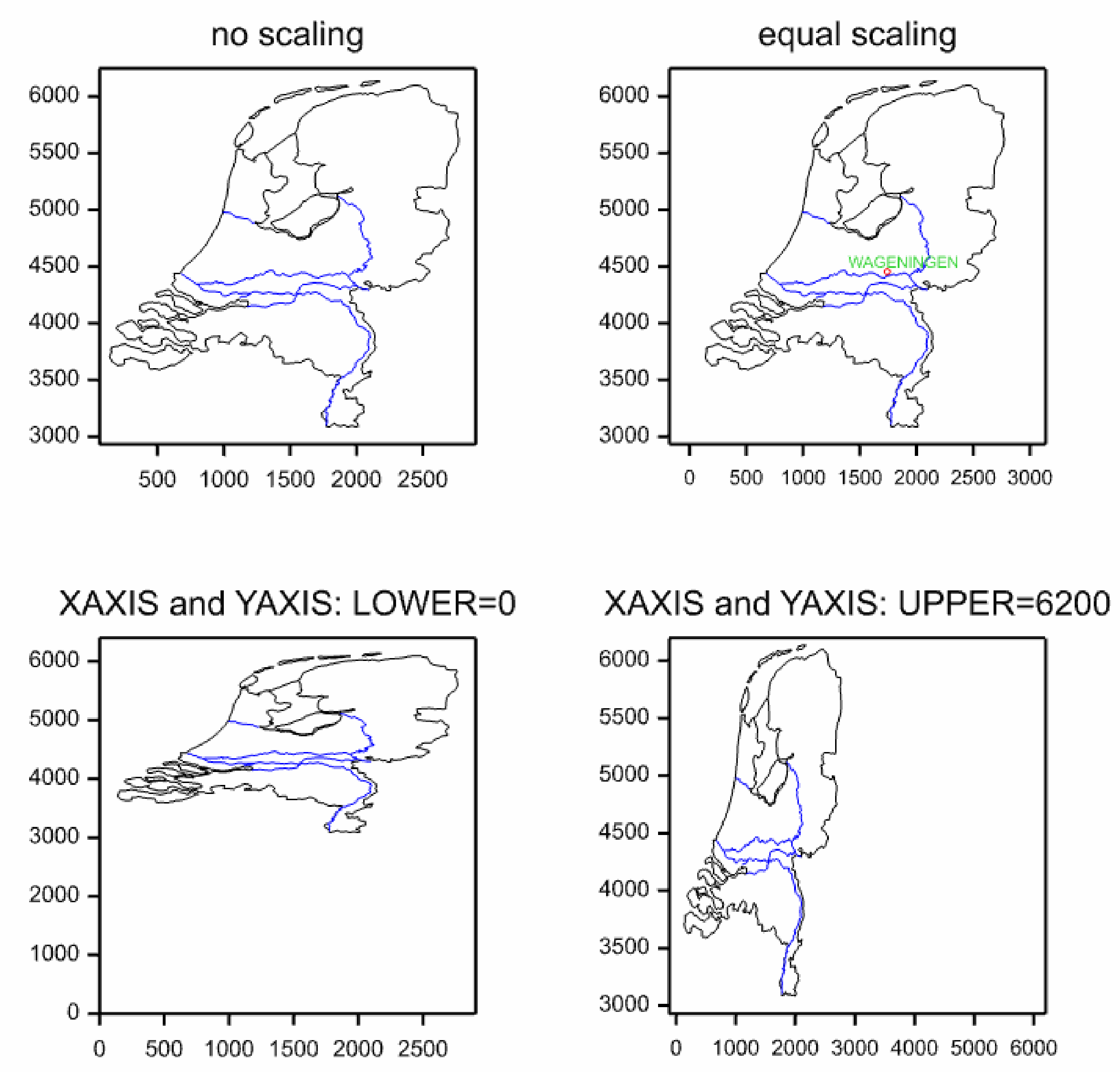
Example 29

```
JOB       'Example 30
           Directive YAXIS: parameter TRANSFORM in the right window'
READ       Log_h, Lower, Theta, Upper
  0.03   0.400   0.416   0.444      1.00   0.397   0.414   0.436
  1.30   0.391   0.411   0.432      1.50   0.385   0.408   0.428
  1.70   0.375   0.402   0.422      2.00   0.355   0.384   0.408
  2.40   0.312   0.342   0.373      2.70   0.266   0.297   0.336
  3.00   0.212   0.251   0.301      3.40   0.146   0.195   0.260
  3.70   0.109   0.161   0.233      4.00   0.081   0.132   0.210
  4.20   0.067   0.116   0.196 :

CALCULATE  h = 10**Log_h

PEN        1; METHOD= monotonic; SYMBOLS= 0

XAXIS      5,6; TITLE= '~{theta} (cm~^3 / cm~^3)'; LOWER= 0; UPPER= 0.5; \
           MARKS= !(0, 0.1 ... 0.5)
YAXIS      5; TITLE= 'log10(|h|) (cm)'; UPPER=5
DGRAPH     [TITLE= 'pF Curve'; WINDOW= 5; KEYWINDOW= 0] \
           Log_h; Theta; XLOWER= Lower; XUPPER= Upper

VARIATE    [VALUES= 1...10,20,30 ... 100,200,300 ... 1000,\
           2000,3000 ... 10000,20000,30000 ... 100000] Marks
YAXIS      6; TITLE= '|h| (cm)'; ; UPPER=10**5; \
           TRANSFORM=log10; MARKS= Marks; LABELS=!t('10~^0',8(''), \
           '10~^1',8(''),'10~^2',8(''),'10~^3',8(''),'10~^4',8(''),'10~^5')
DGRAPH     [TITLE= 'pF Curve'; WINDOW= 6; KEYWINDOW= 0; SCREEN=keep] \
           h; Theta; XLOWER= Lower; XUPPER= Upper
STOP
```



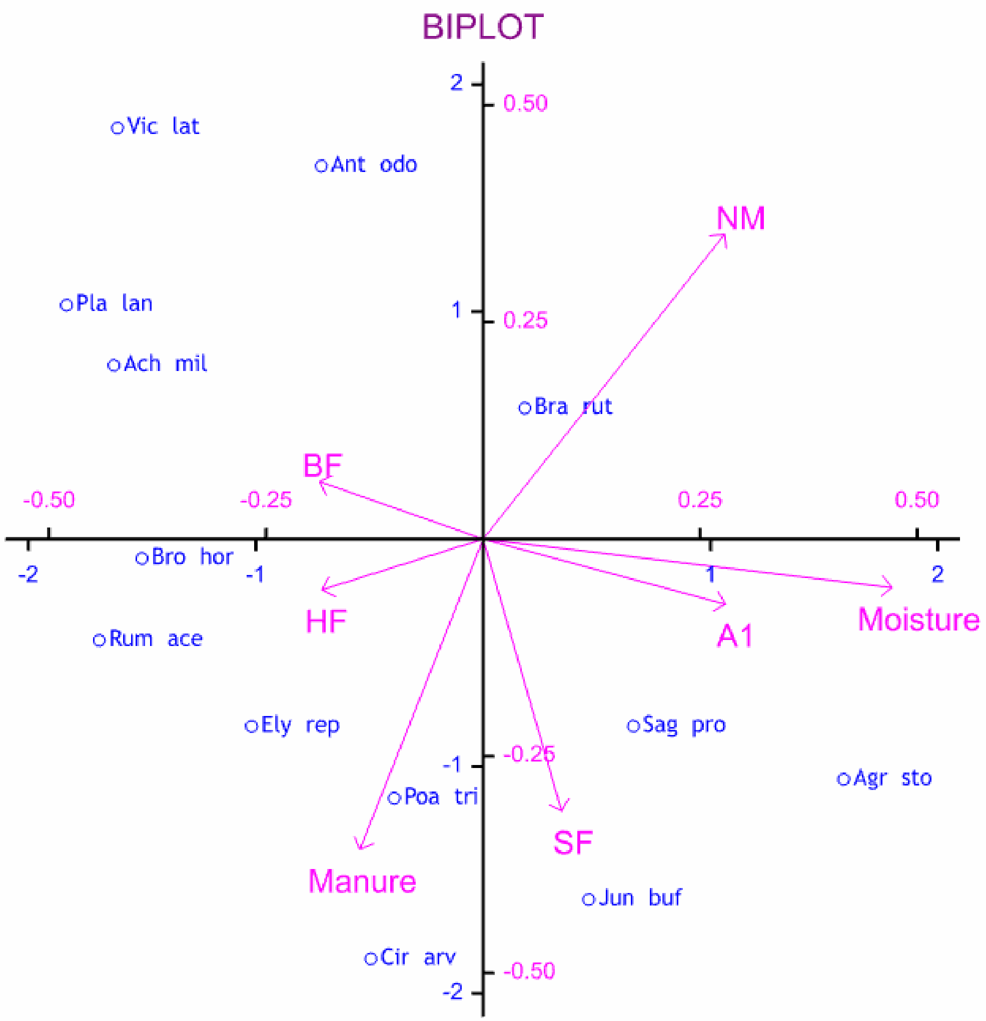
Example 30

```
JOB      'Example 31
         Directive FRAME: parameter SCALING
         Directives XAXIS and YAXIS: parameters LOWER and UPPER'
OPEN     'Netherlands.dat'; CHANNEL= 2
FRAME    5...6 ; SCALING=*, xy
XAXIS    7; LOWER= 0; UPPER= *
YAXIS    7; LOWER= 0; UPPER= *
XAXIS    8; LOWER= *; UPPER= 6200
YAXIS    8; LOWER= *; UPPER= 6200
PEN      1; SYMBOLS= 0; METHOD= OPEN; JOIN= GIVEN
FOR      ii= 'black', 'black', 6('blue'), 11('black')
  PEN      1; COLOUR= ii
  READ     [CHANNEL= 2; SETNVALUES= yes] x, y
  DGRAPH   [WINDOW= 5; KEYWINDOW= 0; SCREEN= keep; TITLE= 'no scaling'] y; x
  &        [WINDOW= 6; TITLE= 'equal scaling'] y; x
  &        [WINDOW= 7; TITLE= 'XAXIS and YAXIS: LOWER=0'] y; x
  &        [WINDOW= 8; TITLE= 'XAXIS and YAXIS: UPPER=6200'] y; x
ENDFOR
PEN      2; SIZE= 0.5; SYMBOL= 2
DGRAPH   [WINDOW= 6; KEYWINDOW= 0; SCREEN= KEEP] 4430; 1741; PEN= 2
PEN      1; LABELS= 'WAGENINGEN'; SYMBOLS= 0; SIZE= 0.7
DGRAPH   [WINDOW= 6; KEYWINDOW= 0; SCREEN= KEEP] 4480; 1550; PEN= 1
STOP
```



Example 31

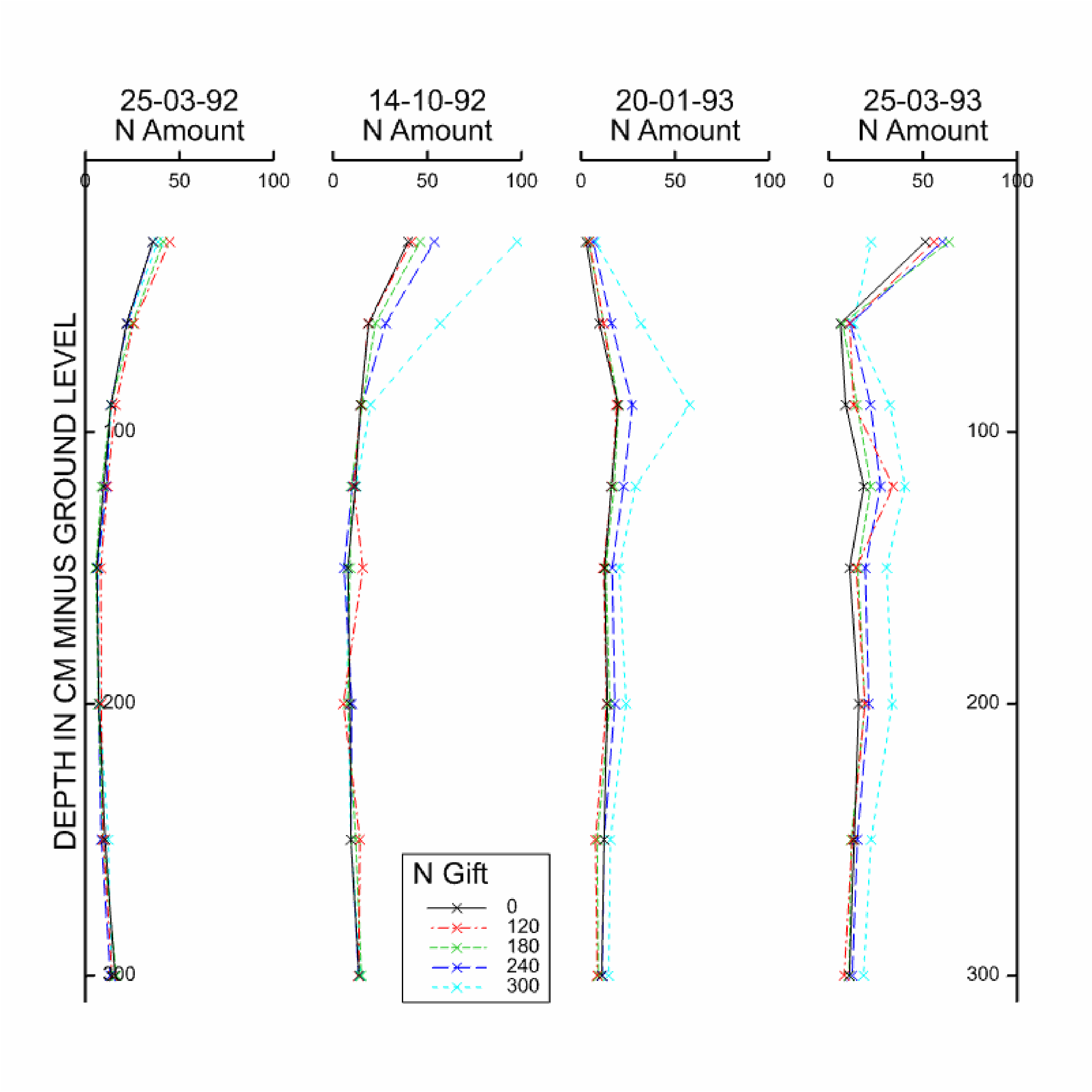
```
JOB       'Example 32
          Directive PEN: parameter SYMBOLS=pointer (arrowhead)
                         parameter ROTATION
          2 different scales along y- and x-axis by using 2 DGRAPH statements'
TEXT      uname, vname
READ      uunit, uname, uas[1, 2]
    1  'Ach  mil'     -1.6241     .7642   2  'Agr  sto'      1.5860   -1.0573
    5  'Ant  odo'      -.7126    1.6420   7  'Bro  hor'     -1.5005    -.0856
    9  'Cir  arv'      -.4940   -1.8476  11  'Ely  rep'     -1.0186    -.8249
   15  'Jun  buf'       .4646   -1.5862  18  'Pla  lan'     -1.8330    1.0295
   20  'Poa  tri'      -.3905   -1.1402  23  'Rum  ace'     -1.6882    -.4459
   24  'Sag  pro'       .6608    -.8236  28  'Vic  lat'     -1.6083    1.8091
   29  'Bra  rut'       .1830     .5759  :
READ      vunit, vname, vas[1, 2]
    1   A1            .2789    -.0751     2   Moisture      .4709    -.0560
    3   Manure       -.1420    -.3577     7   SF            .0904    -.3137
    8   BF           -.1883     .0661     9   HF           -.1853    -.0584
   10   NM            .2781     .3513 :
CALCULATE phi = ARCCOS ( vas[2] / SQRT( vas[1]**2 + vas[2]**2)) * \
          180 / C('pi') * -vas[1] / ABS( vas[1])
&         kwadrant = 1 * (phi .lt. 0 .and. phi .ge. -90) + \
          2 * (phi .ge.   0 .and. phi .lt.   90) + \
          3 * (phi .ge.  90 .and. phi .lt.  180) + \
          4 * (phi .lt. -90 .and. phi .ge. -180)
&         nvar = NVAL( vunit)
&         nvar1, nvar2 = nvar, nvar1 + 1
VARIATE   [NVAL= 2] (vary, varx)[1...nvar]
MATRIX    [ROWS= nvar; COLUMNS= 4] mat
&         [COLUMNS= 2] mleft, mright
CALCULATE mat$[*; 1...4] = 0, vas[1], 0, vas[2]
&         mleft, mright = mat$[*; !(1,2), !(3,4)]
EQUATE    mleft, mright; varx, vary
FRAME     1, 2; YLOWER= 0; YUPPER= 1; XLOWER= 0; XUPPER= 1; BOX=omit
XAXIS     1, 2; LOWER= -2.1, -.55; UPPER= 2.1, .55; YORIGIN= 0; \
          MARKS= !(2, 1, -1, -2), !(-.5, -.25, .25, .5); \
          MPOSITION= outside, inside; LPOSITION= outside, inside
YAXIS     1, 2; LOWER= -2.1, -.55; UPPER= 2.1, .55; XORIGIN= 0; \
          MARKS= !(2, 1, -1, -2), !(-.5, -.25, .25, .5); \
          MPOSITION= outside, inside; LPOSITION= outside, inside
PEN       1; LABELS= uname; SYMBOLS= 2; FONT= 23; COLOUR='Blue'
PEN       -3,-5; COLOUR='Blue','Purple'
DGRAPH    [KEYWINDOW= 0; TITLE= 'BIPLOT'] uas[2]; uas[1]
VARIATE   [VALUES= -1, 0, -1] arrowhead[1]
&         [VALUES= -1, 0,  1] arrowhead[2]
SCALAR    size, vmax; 1.4, .5
PEN       1...nvar; LABELS= *; LINESTYLE= 1; METHOD= line; COLOUR='Fuchsia'; \
          SYMBOLS=0
PEN       nvar1; SYMBOLS= arrowhead; SIZE= 1.3; ROTATION= phi; COLOUR='Fuchsia'
PEN       nvar2; SYMBOLS= 0; SIZE= size; LABELS= vname; COLOUR='Fuchsia'
CALCULATE length = CHARACTERS( vname)
&         vasnew[1] = vas[1] - 0.01 * length * vmax * kwadrant .IN. !(1, 4) - \
          0.02 * length * vmax * kwadrant .IN. !(2, 3)
&         vasnew[2] = vas[2] + 0.03 * vmax * kwadrant .IN. !(1, 2) - \
          0.08 * vmax * kwadrant .IN. !(3, 4)
PEN       -3; COLOUR='Fuchsia'
DGRAPH    [WINDOW= 2; SCREEN= keep; KEYWINDOW= 0] vary[], vas[2], vasnew[2]; \
          varx[], vas[1], vasnew[1]; PEN= 1...nvar2
STOP
```



BIPLOT
Vic lat
Ant odo
NM
Pla lan
Ach mil
Bra rut
BF
Bro hor
HF
A1
Moisture
Rum ace
Ely rep
Sag pro
Agr sto
Poa tri
SF
Manure
Jun buf
Cir arv
-0.50
-0.25
0.25
0.50
-2
-1
1
2

Example 32

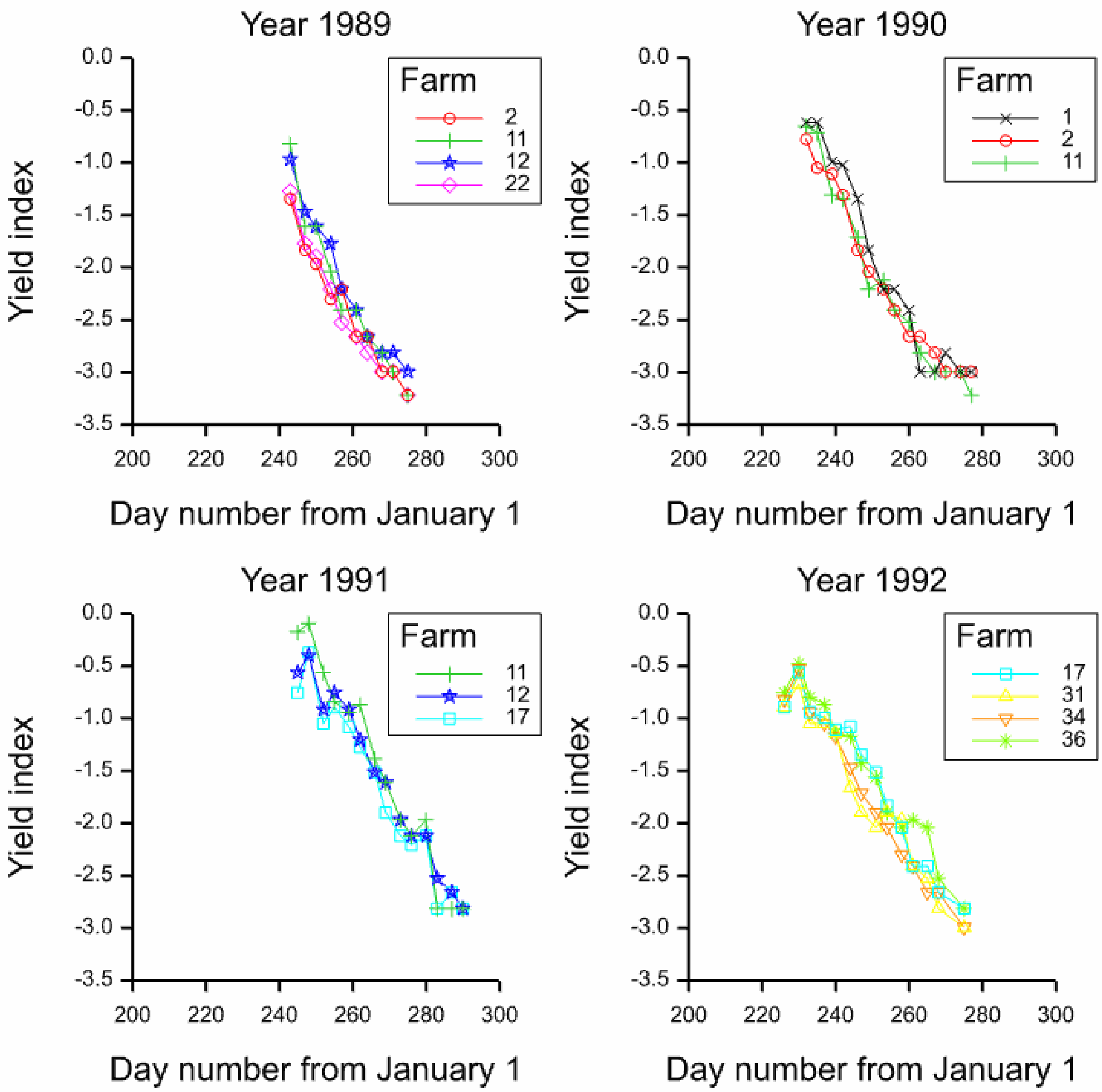
```
JOB    'Example 33
       Directive YAXIS: parameters REVERSE, XORIGIN, MPOSITION, LPOSITION
       Directive DGRAPH: parameter DESCRIPTION if parameter PEN=factor'
FACT   Ngift
READ   Ngift, Depth, m[1...4]; FREP= labels
   0   30  35.85  39.90   2.85  51.30    0   60  21.90  18.90   9.75   6.30
   0   90  13.65  14.55  19.80   9.00    0  120   9.75  11.85  16.20  18.65
   0  150   6.15   7.80  12.60  11.10    0  200   7.25   9.00  14.00  16.00
   0  250  10.50   9.50  12.33  13.50    0  300  15.75  13.75  11.00  11.00
 120   30  44.85  41.70   4.50  55.95  120   60  25.65  18.60  12.00  11.10
 120   90  16.05  14.85  18.90  13.35  120  120  11.70  10.80  16.40  34.20
 120  150   8.25  15.75  12.00  14.55  120  200   8.50   5.50  13.67  19.25
 120  250   9.50  14.25   7.67  12.50  120  300  14.25  13.50   8.67   8.25
 180   30  41.40  46.20   4.05  63.90  180   60  25.05  22.50  12.00   7.35
 180   90  13.50  15.45  20.10  15.15  180  120   8.70   9.60  17.60  22.35
 180  150   5.55   9.00  13.20  15.45  180  200   7.00   7.75  15.33  19.25
 180  250   9.25  12.00   8.67  12.00  180  300  16.50  15.00   9.33  10.75
 240   30  36.00  53.70   6.45  60.45  240   60  22.35  28.05  16.20  11.55
 240   90  13.35  15.45  27.30  22.20  240  120  11.10  10.20  22.60  27.60
 240  150   6.45   5.85  16.60  19.35  240  200   7.00  10.00  18.00  21.25
 240  250   8.50   9.50  12.33  15.25  240  300  13.25  13.50  11.33  12.50
 300   30  38.55  97.75   7.65  22.50  300   60  22.35  56.85  31.95  13.05
 300   90  13.80  19.95  57.75  32.55  300  120  11.40  12.15  29.00  40.50
 300  150   6.75   7.65  20.40  30.75  300  200   7.50   7.75  24.00  33.75
 300  250  12.25   9.50  15.67  22.50  300  300  14.00  14.50  14.67  18.50  :
FRAME  1...4; XLOWER= 0, 0.25 ... 0.75; XUPPER= 0.25, 0.5 ... 1; \
       YLOWER= 0; YUPPER= 1; XMLOWER= 0.04; \
       XMUPPER= 0.02; YMUPPER= 0.07; YMLOWER= 0.08; BOX=omit
XAXIS  1...4; TITLE= '25-03-92', '14-10-92', '20-01-93', '25-03-93'; \
       LOWER= 0; UPPER= 100; MARKS= !(0, 50, 100); YORIGIN= 0; \
       MPOSITION= outside; LPOSITION= inside
YAXIS  1...4; TITLE = 'DEPTH IN CM MINUS GROUND LEVEL', *, *, *; \
       ACTION= display, 2(hide), display; \
       LOWER= 0; UPPER=310; MARKS= !(100, 200, 300); XORIGIN= 0, 0, 0, 100; \
       MPOSITION= inside, *, *, inside; LPOSITION= inside, *, *, inside; \
       REVERSE=yes
PEN    1...5; METH= line; JOIN= given; LINESTYLE= 1, 4, 5, 6, 7
GETATT [labels] Ngift; SAVE= SaveN
FRAME  13; XLOWER= 0.36; YUPPER= 0.23
 " Because the levels of Ngift are equal to the numbers 1 till 5 pens 1...5
   are used for drawing the lines. In the legend the first string of the text
   structure specified by DESCRIPTION is placed behind the symbol of pen 1, the
   second string behind the symbol of pen 2, etc.
   Is the length of the text structure smaller than the number of levels of the
   factor, then the strings are recycled. "
DGRAPH [KEYWINDOW= 13; TITLE= 'N Amount'; KEYDESCRIPTION= 'N Gift'] \
       Depth; m[1]; PEN= Ngift; DESCRIPTION= SaveN['labels']
&      [WINDOW= 2; KEYWINDOW= 0; SCREEN= keep] Depth; m[2]; PEN= Ngift
&      [WINDOW= 3] Depth; m[3]; PEN= Ngift
&      [WINDOW= 4] Depth; m[4]; PEN= Ngift
STOP
```



25-03-92
N Amount
14-10-92
N Amount
20-01-93
N Amount
25-03-93
N Amount
0
50
100
DEPTH IN CM MINUS GROUND LEVEL
100
200
300
N Gift
0
120
180
240
300

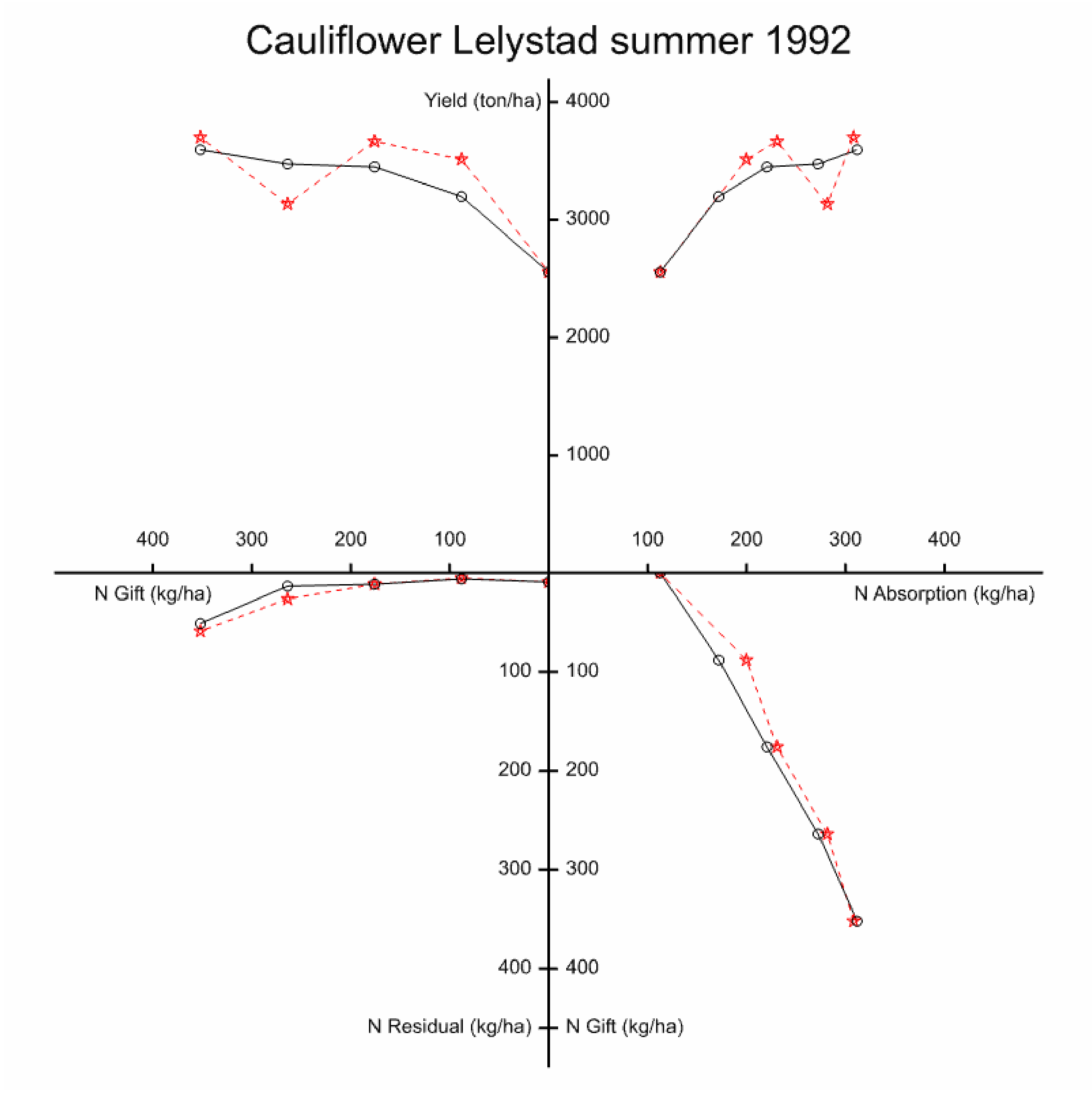
Example 33

```
JOB          'Example 34
             Directive DGRAPH: parameters PEN=factor and DESCRIPTION'
FACTOR       Year, Farm
READ         Year, Daynr, Farm, Index
89 243  2 .26  89 243 11 .44  89 243 12 .38  89 243 22 .28  89 247  2 .16
89 247 11 .20  89 247 12 .23  89 247 22 .17  89 250  2 .14  89 250 11 .20
89 250 12 .20  89 250 22 .15  89 254  2 .10  89 254 11 .13  89 254 12 .17
89 254 22 .11  89 257  2 .11  89 257 11 .09  89 257 12 .11  89 257 22 .08
89 261  2 .07  89 261 11 .09  89 261 12 .09  89 261 22 .07  89 264  2 .07
89 264 11 .07  89 264 12 .07  89 264 22 .06  89 268  2 .05  89 268 11 .06
89 268 12 .06  89 268 22 .05  89 271  2 .05  89 271 11 .05  89 271 12 .06
89 271 22 .05  89 275  2 .04  89 275 11 .04  89 275 12 .05  89 275 22 .04
90 232  1 .54  90 232  2 .46  90 232 11 .52  90 235  1 .54  90 235  2 .35
90 235 11 .49  90 239  1 .37  90 239  2 .33  90 239 11 .27  90 242  1 .36
90 242  2 .27  90 242 11 .26  90 246  1 .26  90 246  2 .16  90 246 11 .18
90 249  1 .16  90 249  2 .13  90 249 11 .11  90 253  1 .11  90 253  2 .11
90 253 11 .12  90 256  1 .11  90 256  2 .09  90 256 11 .09  90 260  1 .09
90 260  2 .07  90 260 11 .08  90 263  1 .05  90 263  2 .07  90 263 11 .06
90 267  1 .05  90 267  2 .06  90 267 11 .05  90 270  1 .06  90 270  2 .05
90 270 11 .05  90 274  1 .05  90 274  2 .05  90 274 11 .05  90 277  1 .05
90 277  2 .05  90 277 11 .04  91 245 11 .84  91 245 12 .57  91 245 17 .47
91 248 11 .91  91 248 12 .67  91 248 17 .69  91 252 11 .57  91 252 12 .40
91 252 17 .35  91 255 11 .43  91 255 12 .47  91 255 17 .41  91 259 11 .39
91 259 12 .40  91 259 17 .34  91 262 11 .42  91 262 12 .30  91 262 17 .28
91 266 11 .25  91 266 12 .22  91 266 17 .22  91 269 11 .20  91 269 12 .20
91 269 17 .15  91 273 11 .14  91 273 12 .14  91 273 17 .12  91 276 11 .12
91 276 12 .12  91 276 17 .11  91 280 11 .14  91 280 12 .12  91 280 17 .12
91 283 11 .06  91 283 12 .08  91 283 17 .06  91 287 11 .06  91 287 12 .07
91 287 17 .07  91 290 11 .06  91 290 12 .06  91 290 17 .06  92 226 17 .41
92 226 31 .41  92 226 34 .44  92 226 36 .47  92 230 17 .57  92 230 31 .51
92 230 34 .59  92 230 36 .62  92 233 17 .39  92 233 31 .35  92 233 34 .39
92 233 36 .45  92 237 17 .37  92 237 31 .35  92 237 34 .35  92 237 36 .42
92 240 17 .33  92 240 31 .32  92 240 34 .31  92 240 36 .32  92 244 17 .34
92 244 31 .19  92 244 34 .23  92 244 36 .31  92 247 17 .26  92 247 31 .15
92 247 34 .18  92 247 36 .24  92 251 17 .22  92 251 31 .13  92 251 34 .15
92 251 36 .21  92 254 17 .16  92 254 31 .15  92 254 34 .13  92 254 36 .15
92 258 17 .13  92 258 31 .14  92 258 34 .10  92 258 36 .13  92 261 17 .09
92 261 31 .09  92 261 34 .09  92 261 36 .14  92 265 17 .09  92 265 31 .08
92 265 34 .07  92 265 36 .13  92 268 17 .07  92 268 31 .06  92 268 34 .07
92 268 36 .08  92 275 17 .06  92 275 31 .05  92 275 34 .05  92 275 36 .06 :
GETATTRIBUTE [levels] Farm; SAVE= SFarm
FTEXT        SFarm[] ; FarmsLabels
CALCULATE    nlevels = NLEV( Farm)
CALCULATE    lnIndex= LOG( Index)
FRAME        1...4; YLOWER= 2(0.6, 0.1); YUPPER= 2(0.93, 0.43); \
             XLOWER= (0.35, 0.85)2; XUPPER= (0.5, 1)2
FRAME        5...8; BOX=omit
XAXIS        5...8; TITLE = 'Day number from January 1'; LOWER= 200; UPPER= 300
YAXIS        5...8; TITLE = 'Yield index'; LOWER= -3.5; UPPER= 0
PEN          1...nlevels; METHOD= line ; SYMBOLS=1...nlevels
FOR          jj= 89...92; \
             tt= 'Year 1989', 'Year 1990', 'Year 1991', 'Year 1992'; \
             ww= 5...8; kk= 1...4
  RESTRICT     lnIndex, Daynr; Year .EQ. jj
  DGRAPH       [TITLE= tt; WINDOW= ww; KEYWINDOW= kk; SCREEN= keep; \
               KEYDESCRIPTION= 'Farm'] \
               lnIndex; Daynr; PEN= Farm; DESCRIPTION= FarmsLabels
  RESTRICT     lnIndex, Daynr
ENDFOR
STOP
```


Year 1989
Farm
2
11
12
22
Yield index
Day number from January 1
Year 1990
Farm
1
2
11
Yield index
Day number from January 1
Year 1991
Farm
11
12
17
Yield index
Day number from January 1
Year 1992
Farm
17
31
34
36
Yield index
Day number from January 1

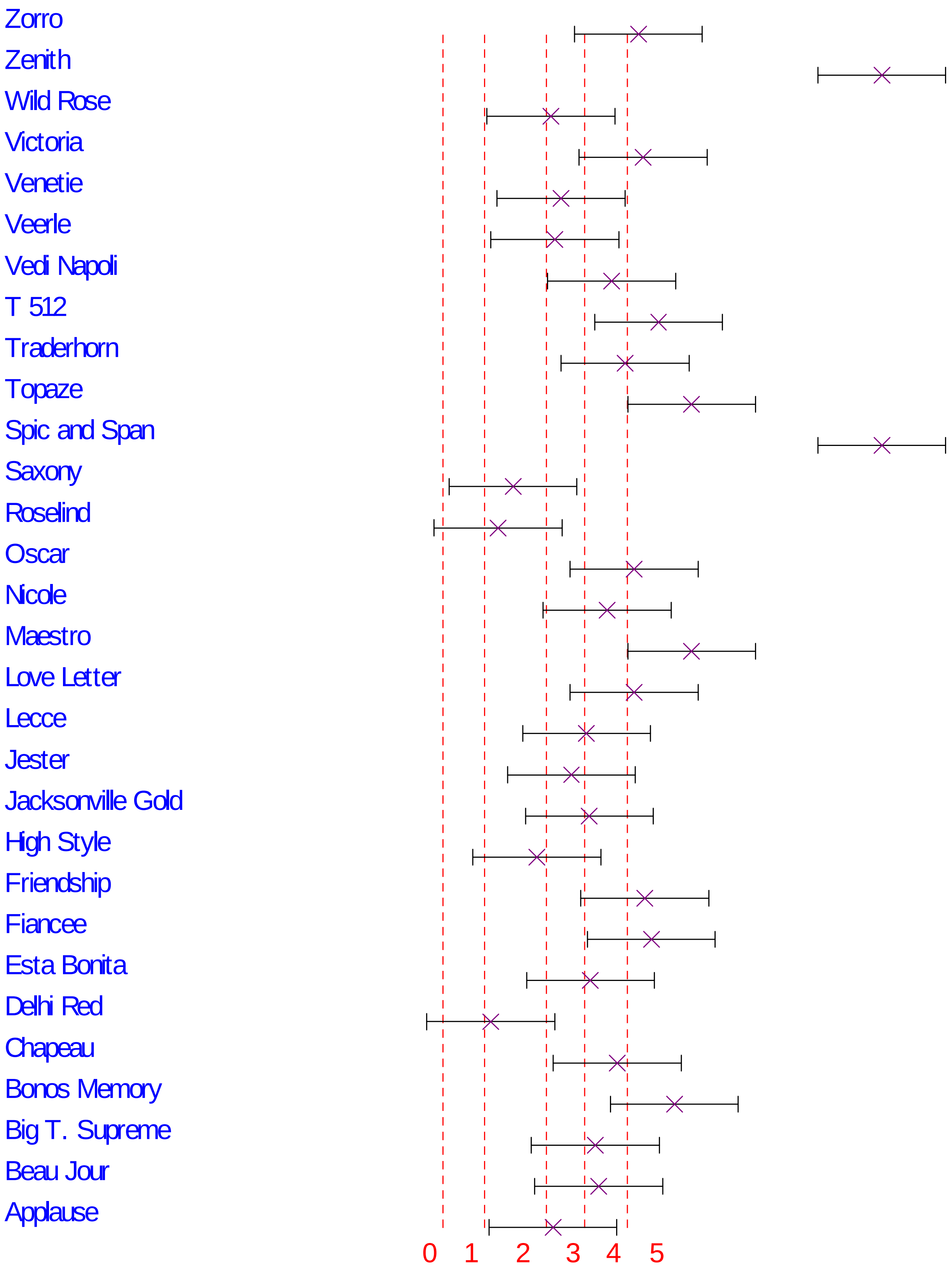
Example 34

```
JOB    'Example 35
       Directive DSTART: option TITLE
       Directives YAXIS and XAXIS: parameters MPOSITION, LPOSITION'
FACTOR Treat
READ   Treat, Yld,  TotN, Ngift, Nres
 1    2554   112.8     0   9.2      2    2554   112.8     0   9.2
 1    3198   171.9    88   6.1      2    3513   199.8    88   5.3
 1    3448   220.7   176  11.5      2    3666   230.8   176  11.4
 1    3473   272.4   264  13.4      2    3134   281.5   264  26.3
 1    3595   311.9   352  51.1      2    3701   308.3   352  59.0 :
FRAME  5...8; XMLOWER= 0; XMUPPER= 0; YMLOWER= 2(0, 0); YMUPPER= 2(0,0); \
       BOX=omit
XAXIS  5,7; LOWER= 0, 0; UPPER= 500, 500; YORIGIN= 0; \
       MARKS= !(100, 200, 300, 400),!(400); \
       LABELS= *, 'N Gift (kg/ha)'; \
       MPOSITION= inside, *; LPOSITION= inside, inside; REVERSE=yes
XAXIS  6,8; LOWER= 0, 0; UPPER= 500, 500; YORIGIN= 0; \
       MARKS= !(100, 200, 300, 400),!(400); \
       LABELS= *, 'N Absorption (kg/ha)'; \
       MPOSITION= inside, *; LPOSITION= inside, inside
YAXIS  5...8; LOWER= 0, 0, 0, 0; UPPER= 4200, 4200, 500, 500; XORIGIN= 0; \
       MARKS= 2( !(1000, 2000, 3000, 4000), !(100, 200, 300, 400, 460) ); \
       LABELS= !T( 3(''), 'Yield (ton/ha)'), *,\
       !T( '100', '200', '300', '400', 'N Residual (kg/ha)'), \
       !T( '100', '200', '300', '400', 'N Gift (kg/ha)'); \
       MPOSITION=      *, inside, inside, inside; \
       LPOSITION= inside, inside, inside, inside; REVERSE=2(no,yes)

PEN    1, 2; METHOD= line; LINESTYLE= 1, 7; SYMBOLS= 2, 4
DSTART [TITLE='Cauliflower Lelystad summer 1992']
DGRAPH [WINDOW= 5; KEYWINDOW= 0] Yld; Ngift; PEN= Treat
&      [WINDOW= 6; SCREEN= keep] Yld; TotN; PEN= Treat
&      [WINDOW= 7] Nres; Ngift; PEN= Treat
&      [WINDOW= 8] Ngift; TotN; PEN= Treat
STOP
```


Cauliflower Lelystad summer 1992
Yield (ton/ha)
4000
3000
2000
1000
400
300
200
100
100
200
300
400
N Gift (kg/ha)
N Absorption (kg/ha)
100
100
200
200
300
300
400
400
N Residual (kg/ha)
N Gift (kg/ha)

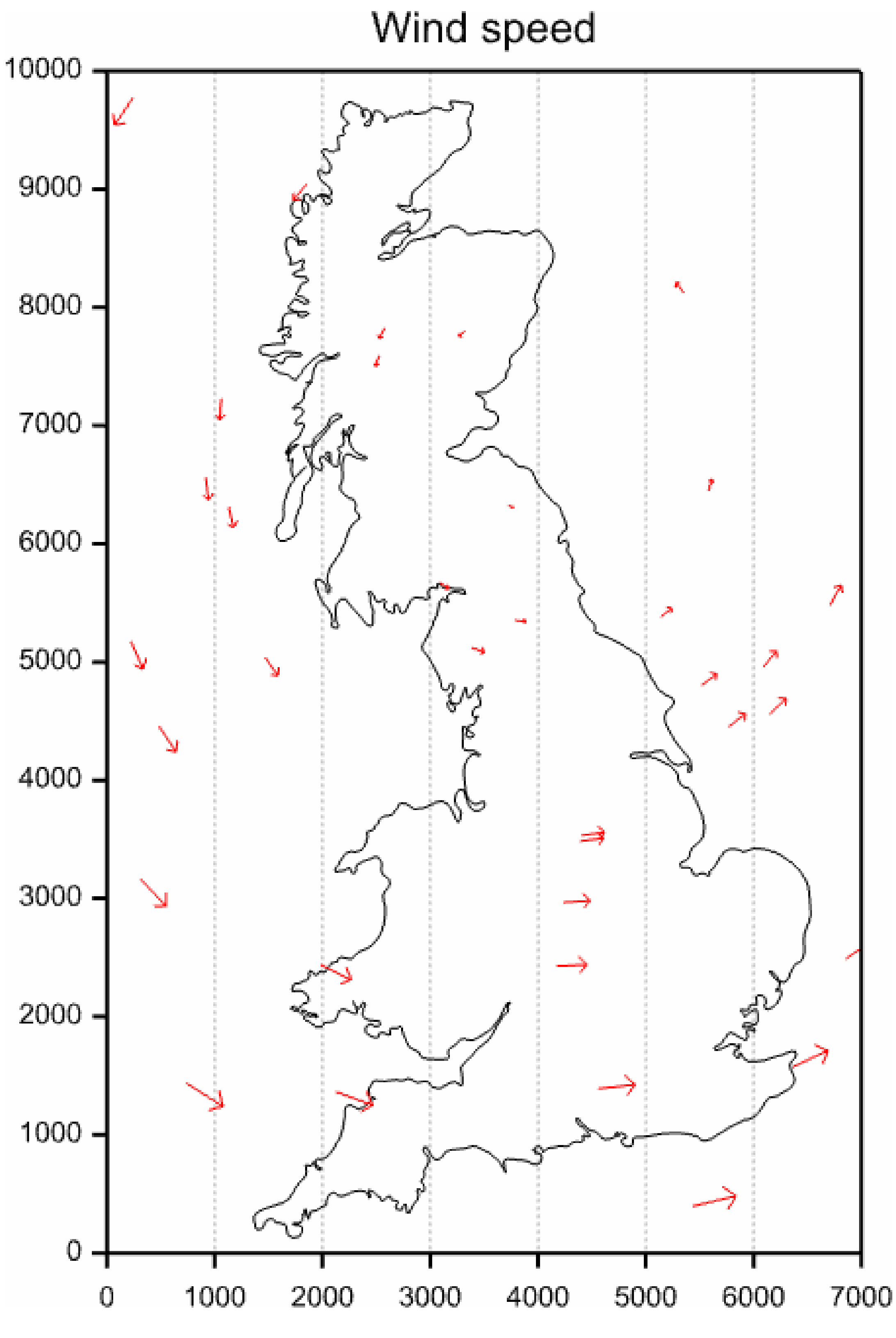
Example 35

```
JOB       'Example 36
          Display of the results of a threshold model.
          This figure can also be produced by Biometris procedure DORDINAL'
TEXT      tname
READ      tname, aantal[0...5] ; FREPRESENTATION=labels
         'Applause' 0  1  6 11  2  0     'Beau Jour' 0  0  1  7  7  5
   'Big T. Supreme' 0  1  2  5  7  5  'Bonos Memory' 0  0  0  0  2 17
          'Chapeau' 0  0  0  3 10  7     'Delhi Red' 5  4  7  1  1  2
      'Esta Bonita' 0  0  0 10  7  3       'Fiancee' 0  0  0  1  4 15
       'Friendship' 0  0  0  2  4 14    'High Style' 0  0 13  6  0  1
'Jacksonville Gold' 0  1  2  4 12  1        'Jester' 0  2  3  7  7  1
            'Lecce' 0  1  1  6 12  0   'Love Letter' 0  0  0  3  5 12
          'Maestro' 0  0  0  0  1 19        'Nicole' 0  0  0  5 10  5
            'Oscar' 0  0  0  3  5 12      'Roselind' 0  6 14  0  0  0
           'Saxony' 0  3 15  1  1  0 'Spic and Span' 0  0  0  0  0 20
           'Topaze' 0  0  0  2  9  9    'Traderhorn' 0  0  0  0  4 16
            'T 512' 0  0  0  0  1 19   'Vedi Napoli' 0  0  2  4  6  8
           'Veerle' 0  1 10  1  7  1       'Venetie' 0  0  7  8  4  1
         'Victoria' 0  0  0  0  7 13     'Wild Rose' 0  0  9  8  3  0
           'Zenith' 0  0  0  0  0 20         'Zorro' 0  0  0  3  4 13 :
GROUPS    tname ; name
MODEL     [DISTRIBUTION= multinomial ; YRELATION=cum] aantal[]
TERMS     [FULL=yes] name
FIT       [NOMESSAGE= dispersion] name
RKEEP     ESTIMATES= esti
VARIATE   [NVALUES= 5] bound
&         [NVALUES= 30] estimate
EQUATE    esti; !P( bound, estimate)
CALCULATE diff = DIFF( bound; 1)
&         diff$[1] = diff$[2]
&         xpos = bound - diff/2
SCALAR    L
CALCULATE L = bound$[5] + diff$[5] / 2
VARIATE   [VALUES= #xpos, L] xpos
TEXT      [NVALUES=6 ; VALUES='0', '1', '2', '3', '4', '5'] namebound
CALC      nlines= NVAL( estimate)
&         nlines1 = nlines + 1
&         minbound = MIN( esti)
&         maxbound = MAX( esti)
&         xlower = minbound - (maxbound - minbound)
&         xupper = maxbound + (maxbound - minbound)
&         lower, upper = estimate - 1, -1 * 2.9444
VARIATE   [VAL= (1, #nlines)5] y
VARIATE   [VAL= 2(#bound)] x
FACTOR    [LEV= !(1...5); VAL= 2(1...5)] factor
FRAME     3; BOX=omit
XAXIS     3; ACTION=hide; LOWER= xlower; UPPER= xupper
YAXIS     3; ACTION=hide; LOWER= -1; UPPER= nlines1
PEN       1...5; METHOD=line; JOIN=given; SYMBOLS=0; LINESTYLE=7; COLOUR='red'
PEN       6; SYMBOLS=0; LABELS=tname; COLOUR='blue'
PEN       7; LINESTYLE= 1; SYMBOLS= 1; COLOUR= 'purple'
PEN       8; SYMBOLS=0; LABELS=namebound; COLOUR='red'
PRIN      (factor)2; FREP=ord,lev ; DEC=0
DGRAPH    [WINDOW= 3; KEYWINDOW= 0] !(1...nlines); X=!(#nlines(xlower)); PEN=6
&         [SCREEN= keep] y; X=x; PEN=factor
&         !(1...nlines); X=estimate; PEN=7; XLOWER= lower; XUPPER= upper
&         !(6(0)); X=xpos; PEN=8

"DORDINAL [XLGAP=0.2; LTITLE='Procedure DORDINAL' ; LXTITLE=namebound ; \
          PRINT=plot,line] name
"
STOP
```


Zorro
Zenith
Wild Rose
Victoria
Venetie
Veerle
Vedi Napoli
T 512
Traderhorn
Topaze
Spic and Span
Saxony
Roselind
Oscar
Nicole
Maestro
Love Letter
Lecce
Jester
Jacksonville Gold
High Style
Friendship
Fiancee
Esta Bonita
Delhi Red
Chapeau
Bonos Memory
Big T. Supreme
Beau Jour
Applause
0 1 2 3 4 5

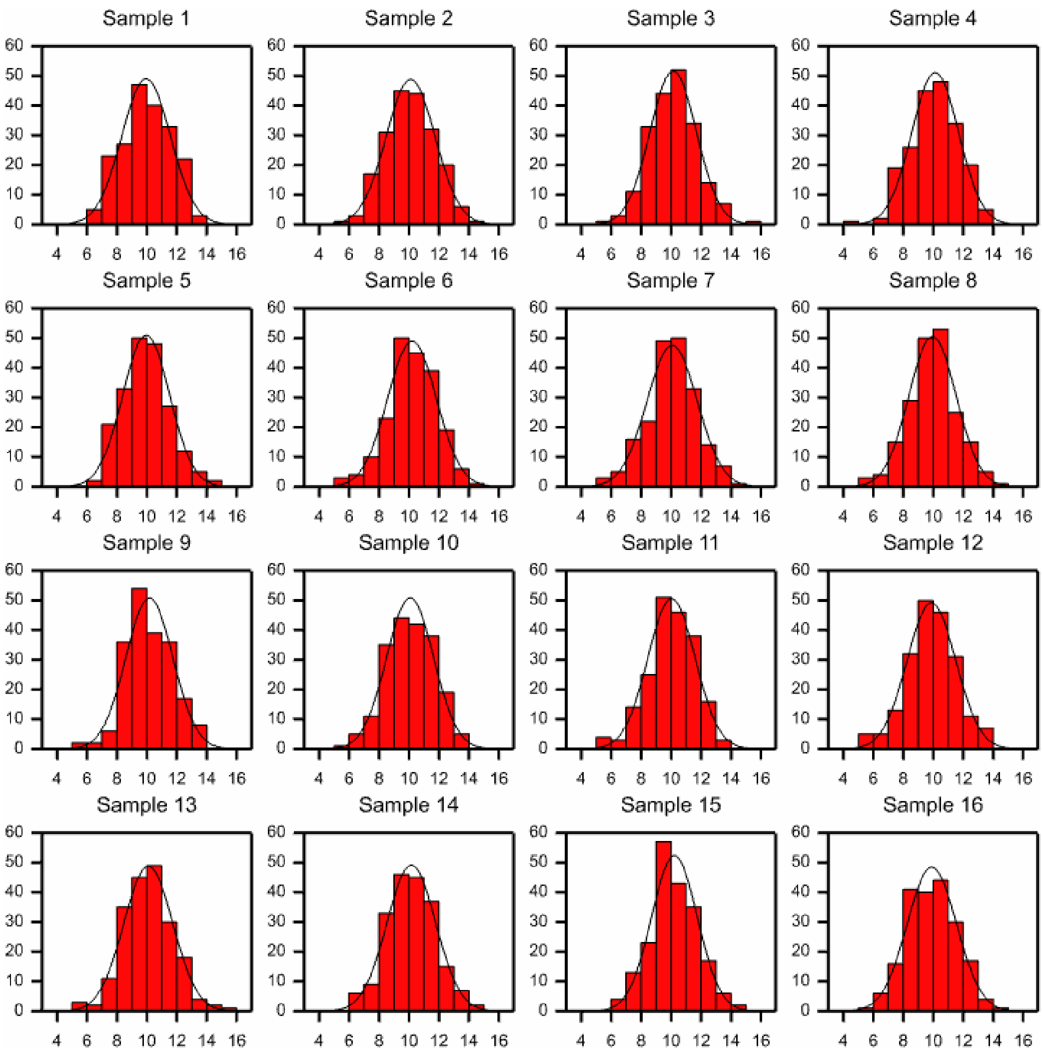
Example 36

```
JOB      'Example 37
         Directive PEN: parameter JOIN=given'
" Example DGRA-7: Drawing a weather map
  This example draws a map of Great Britain with wind speed and
  direction indicated at various points."
FILEREAD [NAME='%gendir%/examples/DGRA-7.DAT'] X,Y
XAXIS    1; LOWER=0; UPPER=7000; MARK=!(0,1000...7000)
YAXIS    1; LOWER=0; UPPER=10000; MARK=!(0,1000...10000)
FRAME    [GRID=xy] 1; YLOWER=0; YUPPER=1; XLOWER=0; XUPPER=0.7
PEN      1; SYMBOL=0; METHOD=open; JOIN=given; LINESTYLE=1
DGRAPH   [KEYWINDOW=0; TITLE='Wind speed'] Y; X
" Add arrows of different sizes and angles at different places"
DELETE   [REDEFINE=yes] Y,X
SCALAR   N,P; 35,5     " no. units, no. points in shapes "
MATRIX   [ROWS=P; COLUMNS=N] XX,YY
DIAGONAL [ROWS=N] Size
POINTER  [NVALUES=!T(X,Y)] At,Shape
MATRIX   [ROWS=1; COLUMNS=N] Angle,At[]
MATRIX   [ROWS=P; COLUMNS=1] Onep,Shape[]; VALUES=!((1)#P),(*)2
READ     Shape[1,2]
   0.0  0.0
   0.0  1.0
  -0.25 0.75
   0.0  1.0
   0.25 0.75 :
OPEN     '%gendir%/examples/DGRA-7A.DAT'; CHANNEL=data7
READ     [CHANNEL=data7] At[],Size,Angle
CLOSE    data7
CALC     Rad = ARCCOS(0.5)/60.0
CALC     XX,YY = (Shape['X','Y']*+COS(Angle*Rad)+    \
         (-1,+1)*Shape['Y','X']*+SIN(Angle*Rad))*+(Size)*6    \
         + Onep*+At['X','Y']
VARIATE  [NVAL=P] Xg[1...N],Yg[1...N]
CALC     Xg[1...N] = XX$[*;1...N]
CALC     Yg[1...N] = YY$[*;1...N]
"Each Xg[] and Yg[] variate contains 5 points on the coordinates scale
 to draw an arrow.
 This is different from example 26 where arrows are drawn with the
 SYMBOL=pointer parameter of the PEN directive."
PEN      1; LINESTYLE=1; SYMBOL=0; METHOD=line; JOIN=given; CLINE='red'
DGRAPH   [SCREEN=keep; KEYWINDOW=0] Yg[]; Xg[]; PEN=1
STOP
```


Wind speed
10000
9000
8000
7000
6000
5000
4000
3000
2000
1000
0
0
1000
2000
3000
4000
5000
6000
7000

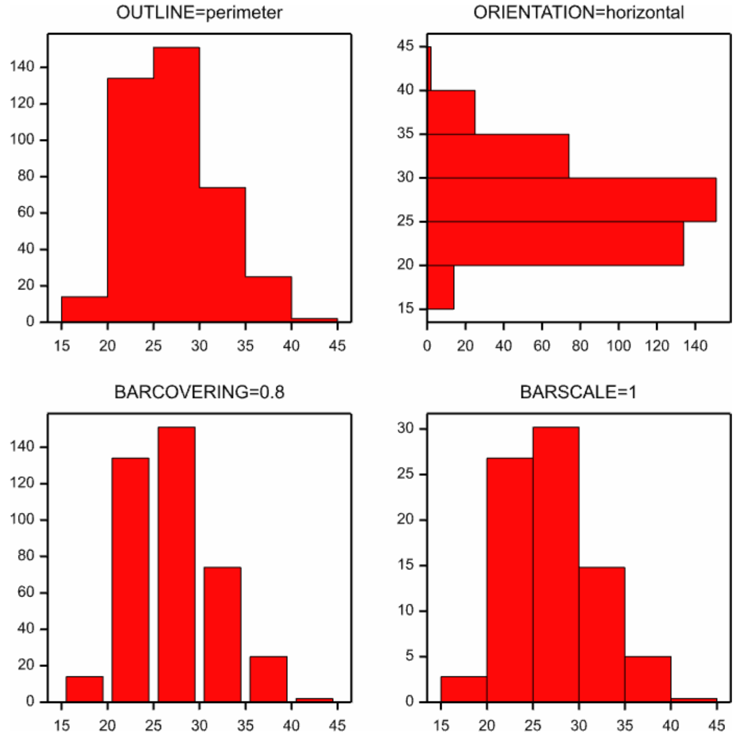
Example 37

```
JOB           'Example 38
              DHISTOGRAM and DGRAPH in the same window.'
SCALAR        Nval; 200
FRAME         1...16; YLOWER= 4(0.75, 0.50, 0.25, 0); \
              YUPPER= 4( 1, 0.75, 0.50, 0.25); XLOWER= (0, 0.25, 0.5, 0.75)4; \
              XUPPER= (0.25, 0.5, 0.75, 1)4; YMLOWER= 0.04; YMUPPER= 0.04; \
              XMLOWER= 0.05; XMUPPER= 0
XAXIS         1...16; LOWER=3; UPPER= 17
YAXIS         1...16; LOWER= 0; UPPER= 60; MARKS= 10
PEN           -5,-3 ; SIZE=1, 0.8
PEN           1; METHOD= monotonic; SYMBOLS= 0
VARIATE       [VALUES= 4, 4.1 ... 16] x
FOR           kk= 1...16; cc= 'clear', ('keep')15; ss= 815967, 15(0)
  PRINT         [CHANNEL= Titel; IPRINT= *; SQUASH= yes] \
                'Sample', kk; JUSTIFICATION= left; DECIMALS= 0; FIELD= 1; \
                SKIP= 0, 1
  GRANDOM       [NVALUES= Nval; MEAN= 10; VARIANCE= 2.5; SEED= ss] normal
  DHISTOGRAM [WINDOW= kk; KEYWINDOW= 0; TITLE= Titel; LIMITS= !(4...16); \
                LABELS= !T('4','','6','','8','','10','','12','','14','','16',''); \
                SCREEN= #cc] normal
  CALCULATE   Mu = MEAN( normal)
  &             Sigma = SQRT( VAR( normal))
  &             y = Nval / Sigma / SQRT( 2 * C( 'pi')) * \
                EXP( - 0.5 * ((x - Mu) / Sigma ) ** 2)
  DGRAPH        [WINDOW= kk; KEYWINDOW= 0; SCREEN= keep] Y= y; X= x
ENDFOR
STOP
```



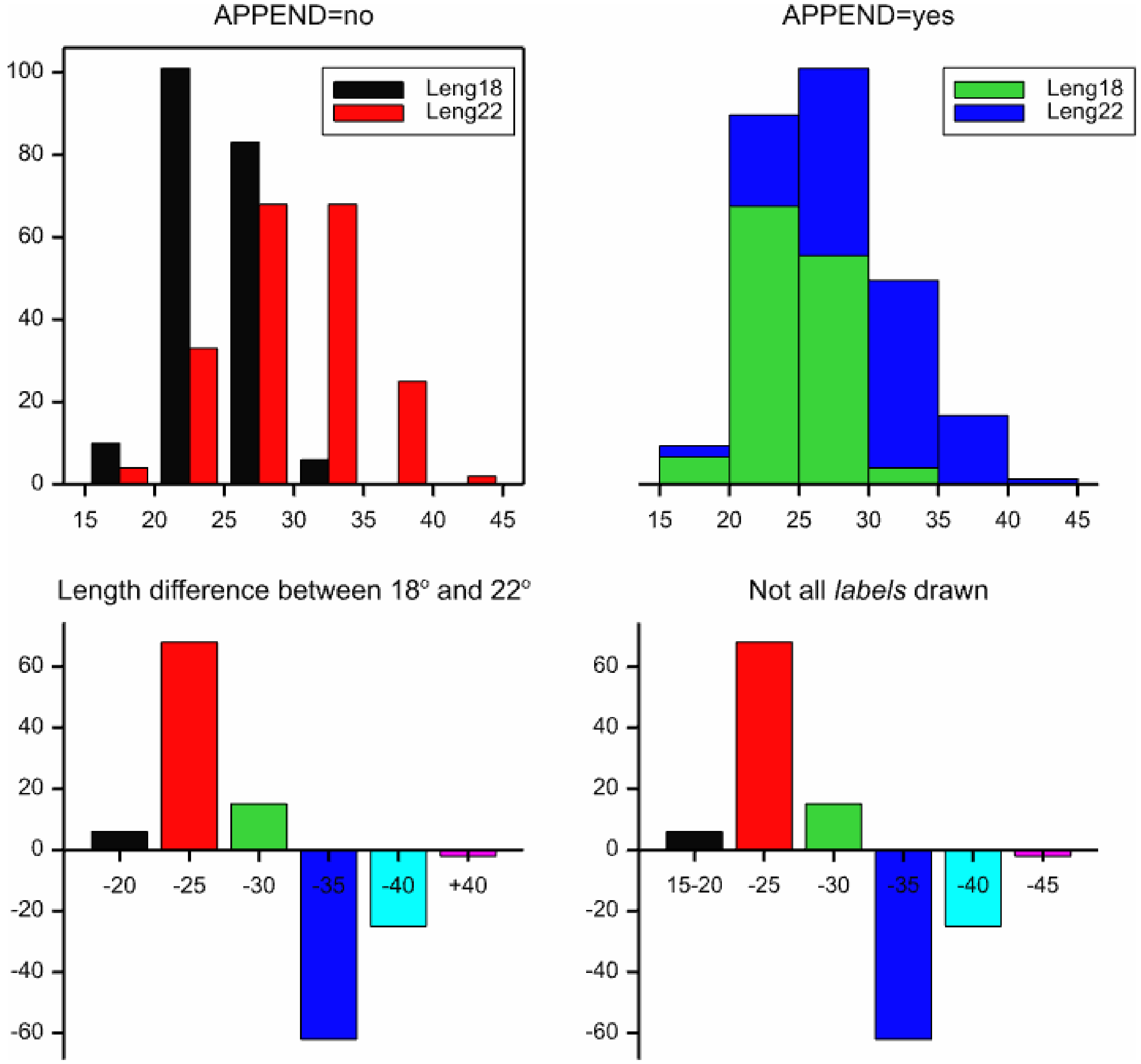
Example 38

```
JOB         'Example 39
            Directive DHISTOGRAM: options ORIENTATION, OUTLINE, BARCOVERING,
                                                  BARSCALE'
READ        Length
24 28 22 22 26 24 29 26 29 25 28 26 24 31 23 30 24 26 30 26
25 31 27 21 30 20 21 29 29 28 21 26 21 22 26 29 20 28 30 30
26 30 23 23 27 25 30 24 32 24 26 23 24 23 30 25 24 26 26 24
22 27 23 26 26 24 21 29 25 25 30 23 25 21 25 26 20 23 27 21
28 27 23 27 30 21 31 27 28 23 23 24 31 29 23 25 26 27 27 28
25 22 30 29 27 29 24 25 19 25 26 20 23 22 26 25 24 23 28 27
24 22 29 25 24 27 26 28 30 25 22 22 27 28 24 23 24 28 25 23
27 27 24 22 26 29 23 22 26 25 26 23 26 26 25 21 24 20 25 23
23 27 26 25 30 27 21 27 19 25 21 25 24 21 23 20 24 21 25 30
25 20 21 27 23 24 25 28 21 28 34 24 23 23 24 30 19 22 27 25
29 34 28 23 35 32 30 28 31 23 36 24 24 31 28 19 35 24 26 39
34 32 41 24 25 34 29 34 23 23 29 25 30 31 31 33 21 24 31 33
33 31 24 31 24 40 33 41 24 29 35 31 28 26 32 31 29 28 27 28
31 30 32 28 25 29 30 28 36 29 34 30 28 32 30 29 30 26 39 37
30 24 36 29 32 28 34 25 31 37 34 33 35 24 28 28 27 23 31 39
36 28 38 31 20 26 23 29 32 37 25 34 29 26 32 28 35 28 26 27
26 36 36 27 39 33 29 34 28 22 26 37 28 36 38 33 24 29 28 34
36 25 34 20 31 28 32 33 26 22 38 27 31 25 31 33 26 26 31 31
24 31 29 37 32 30 23 27 35 30 32 22 25 39 28 34 33 32 31 31
33 16 34 31 32 29 36 35 27 36 30 23 34 29 22 30 27 33 36 28:

FRAME       5...8; YMLOWER= 0.05; XMLOWER= 0.05
PEN         -5 ; SIZE=1.2
VARIATE     [VALUES= 20, 25 ... 40] Limits
TEXT        [VALUES= '-20', '-25', '-30', '-35', '-40', '+40'] Labels
DHISTOGRAM  [WINDOW= 5 ; KEYWINDOW= 0; OUTLINE=perimeter; \
            LIMITS=Limits; TITLE= 'OUTLINE=perimeter'] Length
DHISTOGRAM  [WINDOW= 6; KEYWINDOW= 0; SCREEN= keep; ORIENTATION=horizontal; \
            LIMITS=Limits; TITLE= 'ORIENTATION=horizontal'] Length
DHISTOGRAM  [WINDOW= 7; SCREEN= keep; KEYWINDOW= 0; BARCOVERING=0.8; \
            LIMITS=Limits; TITLE= 'BARCOVERING=0.8'] Length
DHISTOGRAM  [WINDOW= 8; SCREEN= keep; KEYWINDOW= 0; BARSCALE=1; \
            LIMITS=Limits; TITLE= 'BARSCALE=1'] Length
STOP
```



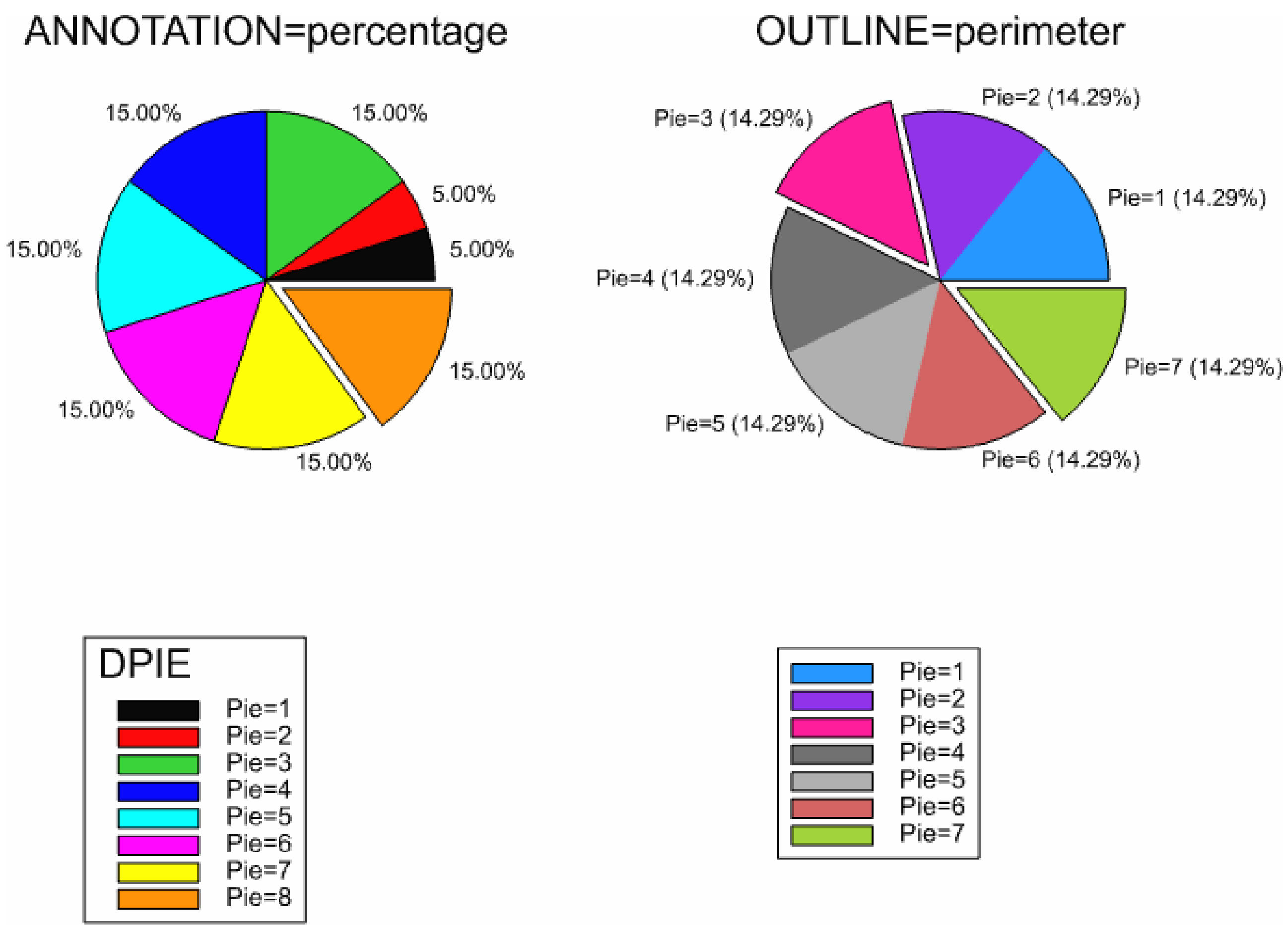
Example 39

```
JOB         'Example 40
            Directive DHISTOGRAM: options APPEND, LIMITS and LABELS'
READ        Length
24 28 22 22 26 24 29 26 29 25 28 26 24 31 23 30 24 26 30 26
25 31 27 21 30 20 21 29 29 28 21 26 21 22 26 29 20 28 30 30
26 30 23 23 27 25 30 24 32 24 26 23 24 23 30 25 24 26 26 24
22 27 23 26 26 24 21 29 25 25 30 23 25 21 25 26 20 23 27 21
28 27 23 27 30 21 31 27 28 23 23 24 31 29 23 25 26 27 27 28
25 22 30 29 27 29 24 25 19 25 26 20 23 22 26 25 24 23 28 27
24 22 29 25 24 27 26 28 30 25 22 22 27 28 24 23 24 28 25 23
27 27 24 22 26 29 23 22 26 25 26 23 26 26 25 21 24 20 25 23
23 27 26 25 30 27 21 27 19 25 21 25 24 21 23 20 24 21 25 30
25 20 21 27 23 24 25 28 21 28 34 24 23 23 24 30 19 22 27 25
29 34 28 23 35 32 30 28 31 23 36 24 24 31 28 19 35 24 26 39
34 32 41 24 25 34 29 34 23 23 29 25 30 31 31 33 21 24 31 33
33 31 24 31 24 40 33 41 24 29 35 31 28 26 32 31 29 28 27 28
31 30 32 28 25 29 30 28 36 29 34 30 28 32 30 29 30 26 39 37
30 24 36 29 32 28 34 25 31 37 34 33 35 24 28 28 27 23 31 39
36 28 38 31 20 26 23 29 32 37 25 34 29 26 32 28 35 28 26 27
26 36 36 27 39 33 29 34 28 22 26 37 28 36 38 33 24 29 28 34
36 25 34 20 31 28 32 33 26 22 38 27 31 25 31 33 26 26 31 31
24 31 29 37 32 30 23 27 35 30 32 22 25 39 28 34 33 32 31 31
33 16 34 31 32 29 36 35 27 36 30 23 34 29 22 30 27 33 36 28:
FACTOR      [LEVELS= !(18,22); VALUES= 200(18, 22)] Temp
VARIATE     Leng18, Leng22; VALUES= Length
RESTRICT    Leng18, Leng22; Temp .EQ. 18, 22
" Windows 1 and 2 are the keywindows of the histograms in windows 5 and 6"
FRAME       WINDOW= 1, 2; YLOWER= 0.7; YUPPER= 0.92; XLOWER= 0.26, 0.80; \
            XUPPER= 0.5, 1; XMLOWER= 0; XMUPPER= 0
FRAME       5...8; YMLOWER= 0.05; XMLOWER= 0.05; BOX= include, 3(omit)
PEN         -5 ; SIZE=1.2
YAXIS       WINDOW= 6; ACTION=hide
VARIATE     [VALUES= 20, 25 ... 40] Limits
TEXT        [VALUES= '-20', '-25', '-30', '-35', '-40', '+40'] Labels
DHISTOGRAM  [WINDOW= 5; KEYWINDOW= 1; LIMITS= Limits; TITLE= 'APPEND=no'; \
            LABELS= Labels] Leng18, Leng22; PEN= 1, 2
&           [WINDOW= 6; KEYWINDOW= 2; SCREEN= keep; APPEND= yes; \
            TITLE= 'APPEND=yes'] Leng18, Leng22; PEN= 3, 4; \
            NOBSERVATIONS= NObs18, NObs22
VARIATE     VObs18, VObs22; VALUES= NObs18, NObs22
CALCULATE   Diff = VObs18 - VObs22
FACTOR      [LABELS= Labels] FDiff
TABLE       [CLASSIFICATION= FDiff; VALUES= #Diff] TabDiff
DHISTOGRAM  [WINDOW= 7; SCREEN= keep; KEYWINDOW= 0; \
            TITLE= 'Length difference between 18~^o and 22~^o'] TabDiff
&           [WINDOW= 8; TITLE= 'Not all ~i{labels} drawn'; \
            LABELS= !T('15-20', '-25','-30','-35','-40','-45')] TabDiff
STOP
```


APPEND=no
100
80
60
40
20
0
15
20
25
30
35
40
45
Leng18
Leng22
APPEND=yes
15
20
25
30
35
40
45
Leng18
Leng22
Length difference between 18° and 22°
60
40
20
0
-20
-40
-60
-20
-25
-30
-35
-40
+40
Not all *labels* drawn
60
40
20
0
-20
-40
-60
15-20
-25
-30
-35
-40
-45

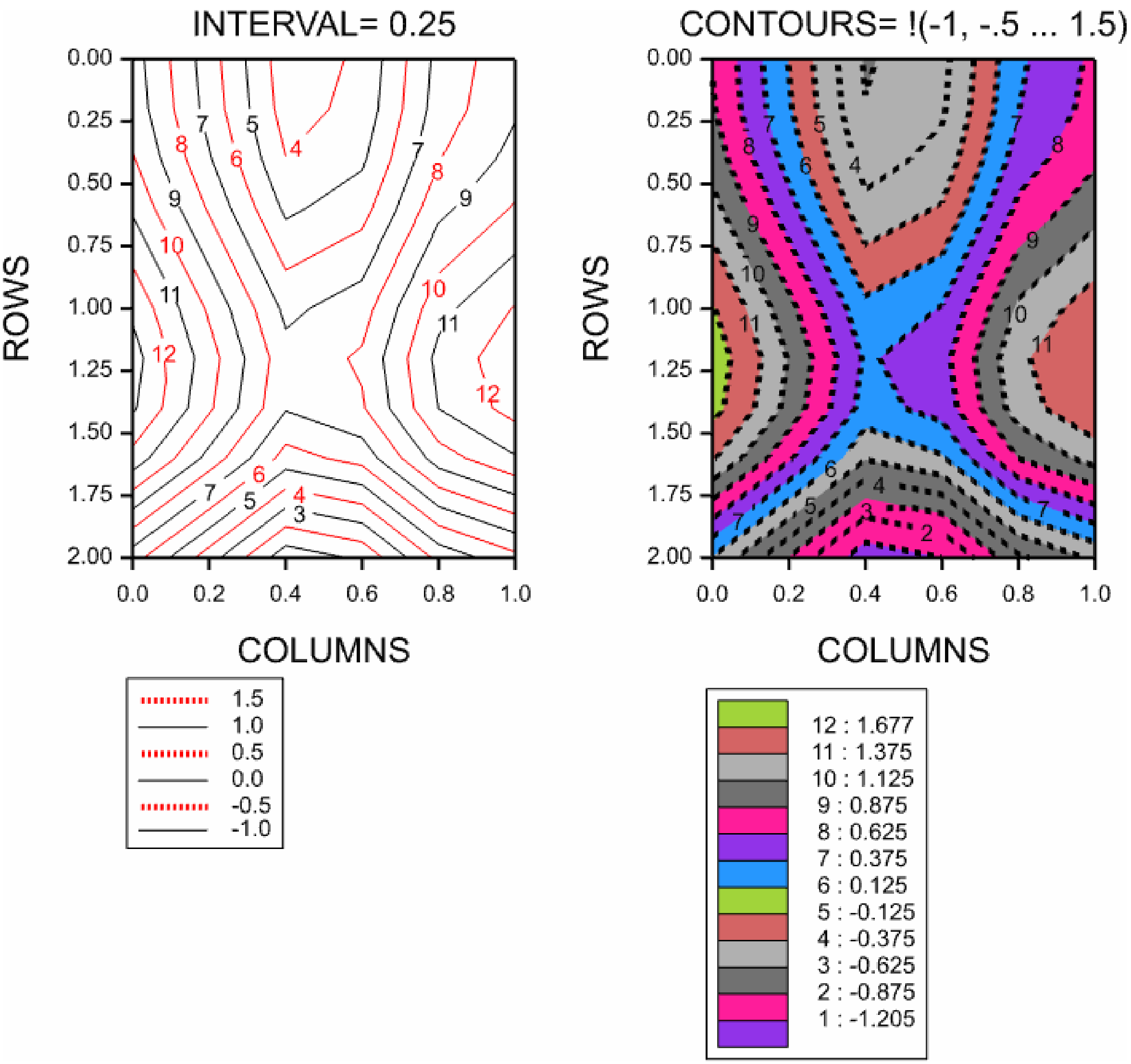
Example 40

```
JOB     'Example 41
        Directive PEN: options ANNOTATION, OUTLINE, KEYDESCRIPTION
                       parameters SLICE= negative scalar, DESCRIPTION'
FRAME   WINDOW= 1...4; YLOWER= 2(0.25, 0); YUPPER= 2(0.75, 0.25); \
        XLOWER= 0, 0.50, 0.15, 0.65; XUPPER= (0.5, 1)2
PEN     -3 ; SIZE=0.85
PEN     -6 ; SIZE=0.95
PEN     -5 ; SIZE=1.50
setdevice 'Exa41.png'; number=nr
DPIE    [WINDOW=1; KEYWINDOW= 3; KEYDESCRIPTION='DPIE'; ANNOTATION=perc; \
        TITLE= 'ANNOTATION=percentage'] \
        SLICE= 2(5), 5(15), -15; \
        DESCRIPTION= 'Pie=1', 'Pie=2', 'Pie=3', 'Pie=4', \
        'Pie=5', 'Pie=6', 'Pie=7', 'Pie=8'
DPIE    [WINDOW= 2; KEYWINDOW= 4; SCREEN= keep; OUTLINE=perimeter; \
        TITLE= 'OUTLINE=perimeter'] \
        SLICE= 2(1), -1, 3(1), -1; PEN=11...17; \
        DESCRIPTION= 'Pie=1', 'Pie=2', 'Pie=3', 'Pie=4', \
        'Pie=5', 'Pie=6', 'Pie=7'
STOP
```



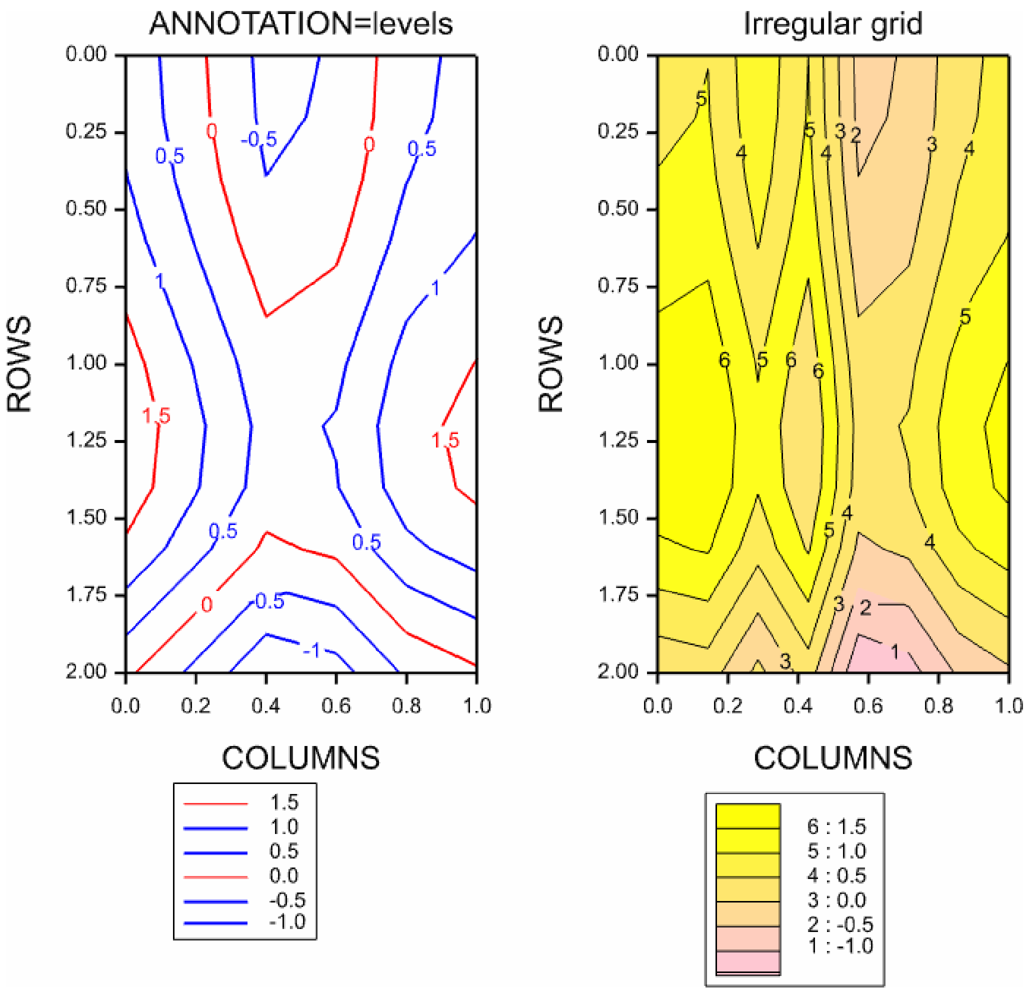
Example 41

```
JOB        'Example 42
           Directive DCONTOUR: parameters INTERVAL and CONTOURS
           parameters PENHIGHLIGHT, PENFILL and HIGHLIGHTFREQUENCY'
VARIATE    [VALUES= 0, 0.2 ... 2] Rows
&          [VALUES= 0, 0.2 ... 1] Columns
MATRIX     [ROWS= Rows; COLUMNS= Columns] X, Y, Function
&          [VALUES= (#Columns)11] X
CALCULATE Y $ [*; 1...6] = Rows
&          Function = COS( 1 / (X + 0.1) ** 2) + SIN( Y ** 2) "min=-1.4,max=1.8"
FRAME      1...4; YLOWER= (0.4, 0)2; YUPPER= (1, 0.4)2; \
           XLOWER= 0, 0.1, 0.5, 0.6; XUPPER= 2(0.5, 1)
XAXIS      1, 3; TITLE= 'COLUMNS'
YAXIS      1, 3; TITLE= 'ROWS'
PEN        2; LINESTYLE= 2; COLOUR='red'; THICKNESS= 4
DCONTOUR   [TITLE= 'INTERVAL= 0.25'; WINDOW= 1; KEYWINDOW= 2] \
           Function; INTERVAL= 0.25; PENFILL=0; \
           PENHIGHLIGHT=2; HIGHLIGHTFREQUENCY=2
PEN        1; LINESTYLE= 7; COLOUR='black'; THICKNESS= 4; METHOD=open
DCONTOUR   [TITLE= 'CONTOURS= !!(-1, -.5 ... 1.5)'; \
           SCREEN= keep; WINDOW= 3; KEYWINDOW= 4] Function ; \
           CONTOURS= !(-1, -.75 ... 1.5); PENCONTOUR=1 ; \
           DESCRIPTION=!t(A,B,C,D,E,F,G); PENFILL=!(11...17)
STOP
```



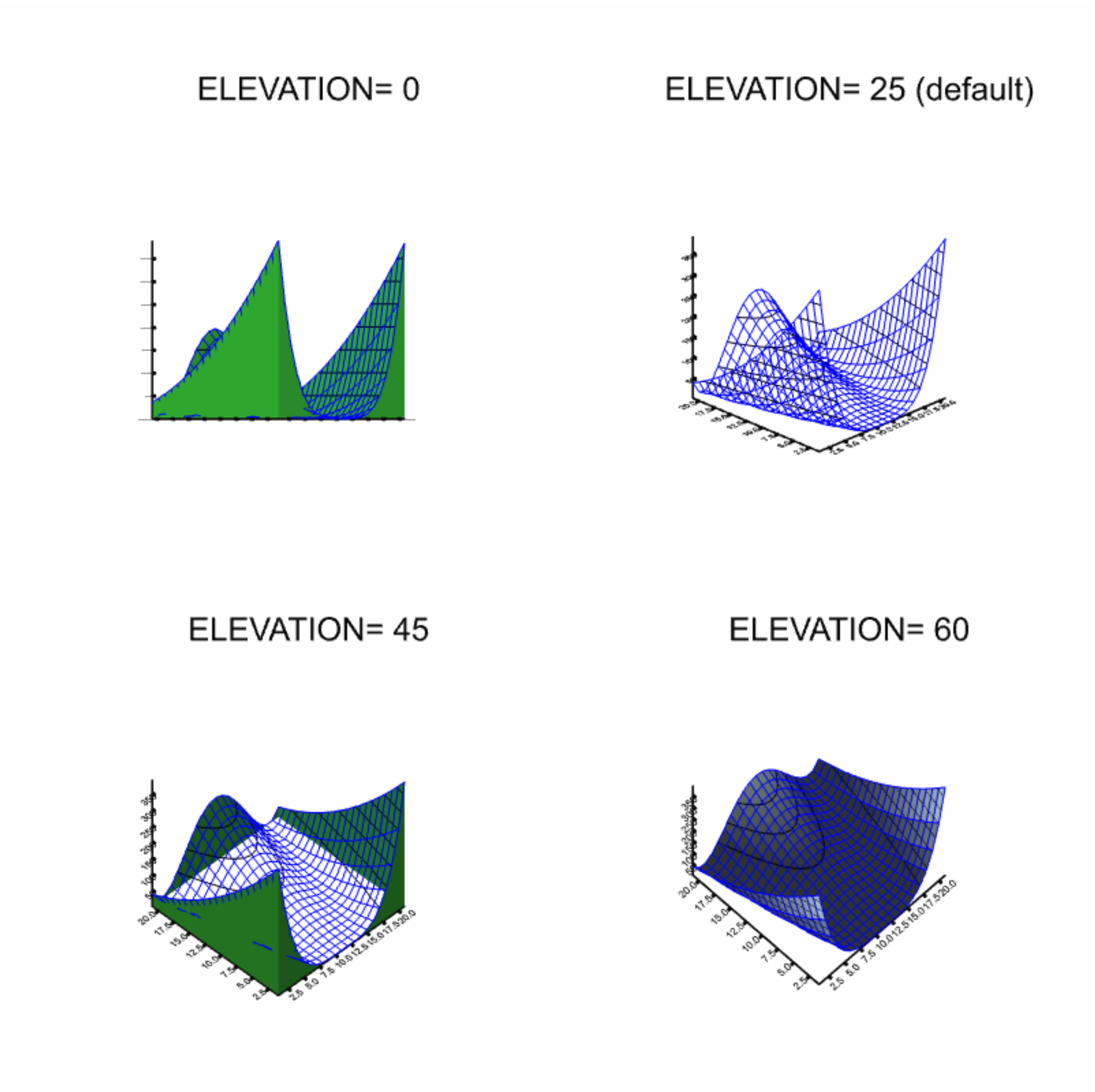
Example 42

```
JOB        'Example 43
           Directive DCONTOUR: parameters INTERVAL and CONTOURS
           parameters PENHIGHLIGHT, PENFILL and HIGHLIGHTFREQUENCY'
VARIATE    [VALUES= 0, 0.2 ... 2] Rows
&          [VALUES= 0, 0.2 ... 1] Columns
MATRIX     [ROWS= Rows; COLUMNS= Columns] X, Y, Function
&          [VALUES= (#Columns)11] X
CALCULATE  Y $ [*; 1...6] = Rows
&          Function = COS( 1 / (X + 0.1) ** 2) + SIN( Y ** 2)
FRAME      1, 3; YLOWER= 0.25; YUPPER= 1; XLOWER= 0, 0.5; XUPPER= 0.5, 1
&          2, 4; YLOWER= 0; YUPPER= 0.25; XLOWER= 0.15, 0.65; XUPPER= 0.5, 1
XAXIS      1, 3; TITLE= 'COLUMNS'
YAXIS      1, 3; TITLE= 'ROWS'
PEN        4; LINESTYLE= 1; COLOUR= 'blue'; THICKNESS= 2
DCONTOUR   [TITLE= 'ANNOTATION=levels' ; ANNOTATION=levels] Function ; \
           PENCONTOUR=4; PENFILL=0; PENHIGHLIGHT=2; HIGHLIGHT=3
VARIATE    [VALUES= 0, 0.1 ... 0.4, 0.6, 0.8, 1] Irr_Cols
MATRIX     [ROWS= Rows; COLUMNS= Irr_Cols] X, Y, Function
&          [VALUES= (#Irr_Cols)11] X
CALCULATE  Y $ [*; 1...8] = Rows
&          Function = COS( 1 / (X + 0.1) ** 2) + SIN( Y ** 2)
PRINT      Function
PEN        5,6; COLOUR= 'pink', 'yellow'
DCONTOUR   [TITLE= 'Irregular grid'; WINDOW= 3; KEYWINDOW= 4; SCREEN= keep]\
           Function; PENFILL=!(5,6)
STOP
```



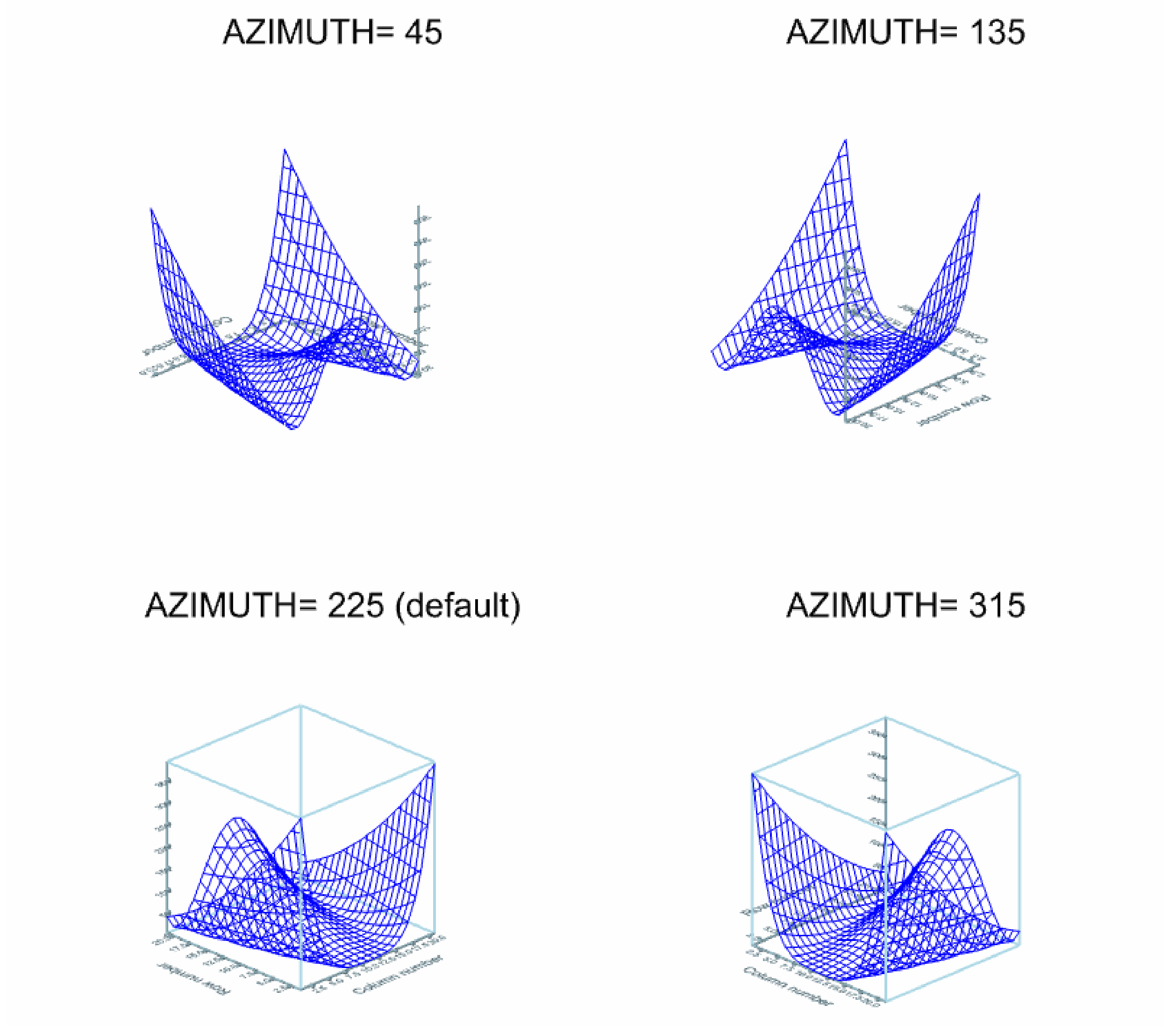
Example 43

```
JOB         'Example 44
            Directive DSURFACE: option ELEVATION
                                parameters PENFILL, PENMESH, PENSIDE '
MATRIX      [ROWS= 21; COLUMNS= 21; VALUES= (-1.4, -1.26 ... 1.4)21] p1
&           [VALUES= 21(0, 0.07 ... 1.4)] p2
MATRIX      Rbrock
CALCULATE   Rbrock = 100 * (p2 - p1 * p1) ** 2 + (1 - p1) ** 2
FRAME       5...8; YMLOWER= 0; XMLOWER= 0; XMUPPER= 0
PEN         2,4 ; COLOUR= 'lightcyan','blue'
DSURFACE    [WINDOW= 5; KEYWINDOW= 0; ELEVATION= 0; TITLE='ELEVATION= 0'] \
            Rbrock ; PENFILL= 0 ; PENMESH=4 ; PENSIDE=3
&           [WINDOW= 6; SCREEN= keep; ELEVATION= 25; \
            TITLE= 'ELEVATION= 25 (default)'] Rbrock ; PENFILL= 0 ; PENMESH=4
&           [WINDOW= 7; ELEVATION= 45; TITLE= 'ELEVATION= 45'] Rbrock ; \
            PENFILL= 0 ; PENMESH=4 ; PENSIDE=3
&           [WINDOW= 8; ELEVATION= 60; TITLE= 'ELEVATION= 60'] Rbrock ; \
            PENFILL= 2 ; PENMESH=4
STOP
```



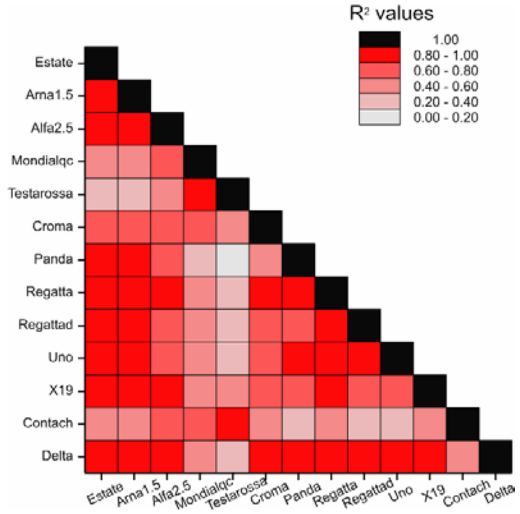
Example 44

```
JOB         'Example 45
            Directive DSURFACE: : option AZIMUTH'
MATRIX      [ROWS= 21; COLUMNS= 21; VALUES= (-1.4, -1.26 ... 1.4)21] p1
&           [VALUES=21(0, 0.07 ... 1.4)] p2
MATRIX      Rbrock
CALCULATE   Rbrock = 100 * (p2 - p1 * p1) ** 2 + (1 - p1) ** 2
FRAME       5...8 ; BOXSURFACE=bounded,omit,full,full
XAXIS       5...8; TITLE= 'Column number' ; ZORIGIN= 0
YAXIS       5...8; TITLE= 'Row number' ; ZORIGIN= 0
PEN         1 ; COLOUR= 'blue'
PEN         -1,-2,-3 ; COLOUR= 'lightblue'
DSURFACE    [WINDOW= 5; KEYWINDOW=0; AZIMUTH= 45; TITLE= 'AZIMUTH= 45'] \
            Rbrock ; PENFILL=0
&           [WINDOW= 6; SCREEN= keep; AZIMUTH= 135; \
            TITLE= 'AZIMUTH= 135'] Rbrock ; PENFILL=0
&           [WINDOW= 7; AZIMUTH= 225; TITLE= 'AZIMUTH= 225 (default)'] \
            Rbrock ; PENFILL=0
&           [WINDOW= 8; AZIMUTH= 315; TITLE= 'AZIMUTH= 315'] Rbrock ; PENFILL=0
STOP
```



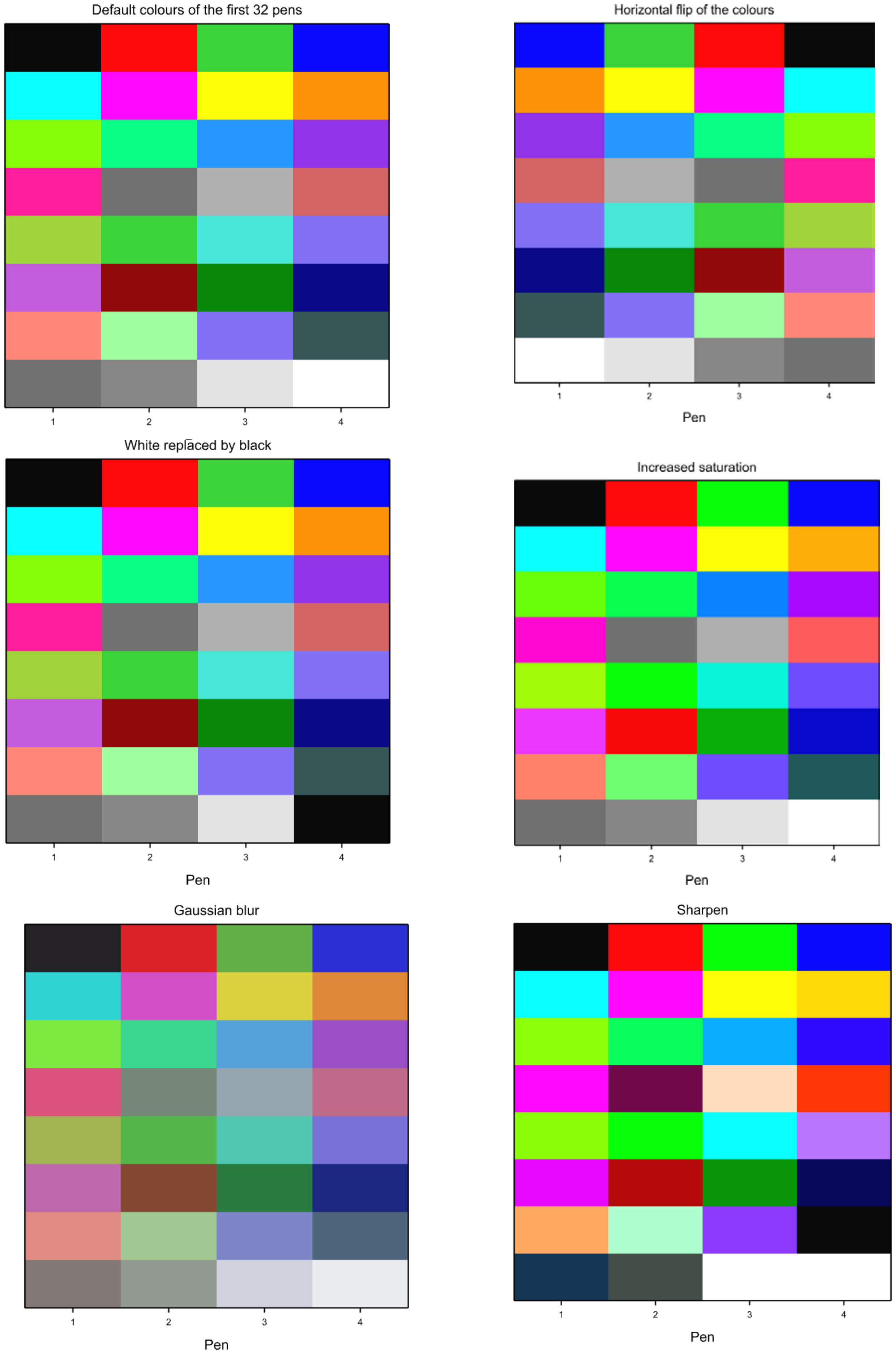
Example 45

```
JOB          'Example 46
             Directive DSHADE: parameters LIMITS and DESCRIPTION
                               option GRID=correlation matrix
             Procedure DCOLOURS'
TEXT         [VALUES=Estate,'Arna1.5','Alfa2.5',Mondialqc,Testarossa,Croma,\
             Panda,Regatta,Regattad,Uno,X19,Contach,Delta] Car
CALCULATE    nCar = NVALUES(Car)
SYMMETRIC    [ROWS=Car] Carsim
READ         Carsim
 1.000
 0.976 1.000
 0.815 0.800 1.000
 0.576 0.546 0.762 1.000
 0.389 0.355 0.561 0.827 1.000
 0.794 0.776 0.762 0.768 0.567 1.000
 0.820 0.855 0.618 0.296 0.103 0.546 1.000
 0.981 0.969 0.823 0.589 0.394 0.820 0.801 1.000
 0.839 0.822 0.675 0.525 0.329 0.756 0.756 0.844 1.000
 0.884 0.909 0.693 0.409 0.211 0.650 0.960 0.866 0.815 1.000
 0.870 0.858 0.828 0.578 0.425 0.602 0.758 0.836 0.700 0.787 1.000
 0.462 0.432 0.618 0.709 0.885 0.440 0.215 0.458 0.309 0.302 0.531 1.000
 0.959 0.951 0.837 0.585 0.393 0.814 0.813 0.959 0.806 0.871 0.826 0.471 1.000
:
VARIATE        [VALUES=0.2, 0.4, 0.6, 0.8, 1] limits
CALCULATE      nlim = NVAL(limits)
VARIATE        [VALUES=0,#limits] lim0
VARIATE        [NVAL=nlim] lim1
EQUATE         lim0 ; NEWSTR=lim1
TXCONSTRUCT    [TEXT=hdesc] lim1, ' - ',limits ; DECIMALS=2
TEXT           [VALUES=#hdesc, '     1.00    '] desc
CALCULATE      ncolours = NVAL(limits) + 1
CALCULATE      ncol_1 = ncolours - 1
VARIATE        pen ; VALUES = 10 + !(1...ncolours)
CALCULATE      revpen  = REVERSE(pen)
TEXT           lcol ; VALUES=!t(red,black)
VARIATE        [VALUES=ncol_1,1] vncolours
DCOLOURS       [METHOD=LINEAR] START='lightgrey'; END=lcol; \
               NCOLOURS=vncolours ; GAMMA=0.75; WINDOW=0; RGB=rgb
PEN            #pen; CAREA=#rgb
FRAME          1; BOX=omit
XAXIS          1; MARKS=!(1...nCar); LABELS=Car ; LROTATION= 25
YAXIS          1; MARKS=!(1...nCar); LABELS=Car
FRAME          1; YLOWER=0.1; YUPPER=0.9; XLOWER=0.1; XUPPER=0.9; BOXKEY=omit
FRAME          2; YLOWER=0.4; YUPPER=0.9; XLOWER=0.6; XUPPER=1  ; BOXKEY=omit
DSHADE         [KEYDESCRIPTION= 'R~^{2} values'] Carsim; LIMITS=limits; \
               PEN=pen ; DESCRIPTION=desc
STOP
```



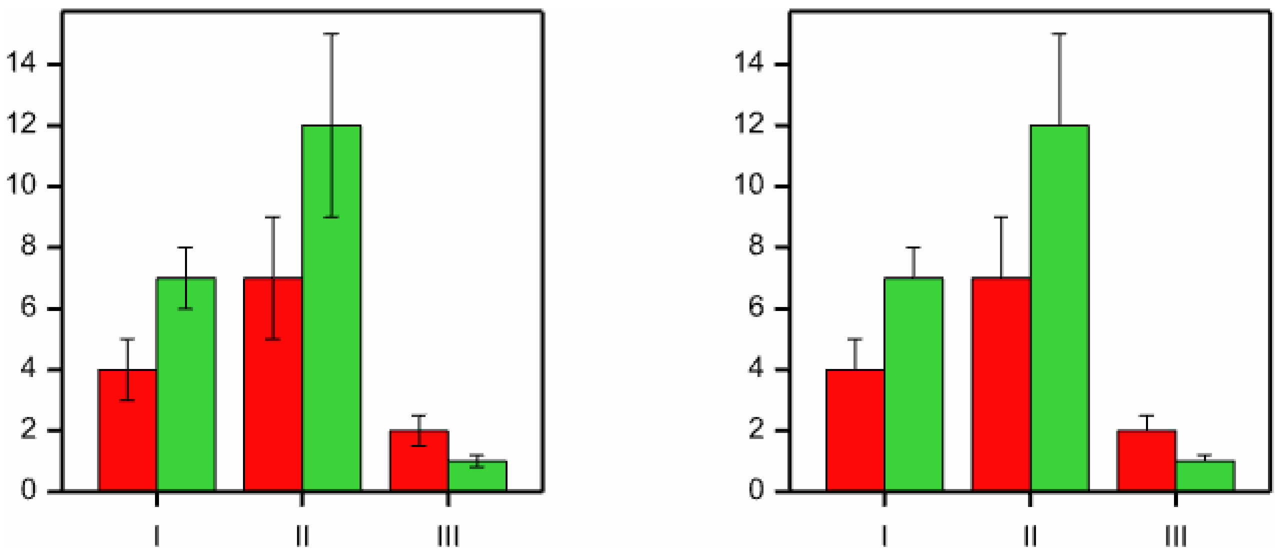
Example 46

```
JOB          'Example 47
              Directive DBITMAP: use of image functions'
VARIATE      [VALUES=1...32] Ncolour
GETRGB       Ncolour; RGB=RGB; NAME=Name
MATRIX       [ROWS=8 ; COLUMNS=4] matRGB ; VALUES=RGB
YAXIS        3 ; MARKS=!(-1) ; LOWER=1 ; UPPER=50
XAXIS        3 ; TITLE= 'Pen' ; MARKS=1
"Example 47.1"
SETDEVICE    'Exa47a.png'; NUMBER=nr
DBITMAP      [WINDOW=3; TITLE= 'Default colours of the first 32 pens'] matRGB
"Example 47.2"
CALCULATE    flip = IMHFLIP(matRGB)
CLOSE        nr; FILETYPE=graphics
SETDEVICE    'Exa47b.png'; NUMBER=nr
DBITMAP      [WINDOW=3; TITLE= 'Horizontal flip of the colours'] flip
"Example 47.3"
CALCULATE    repl = IMCREPLACE(matRGB ; RGB(255;255;255) ; RGB(0;0;0) ; 0)
CLOSE        nr; FILETYPE=graphics
SETDEVICE    'Exa47c.png'; NUMBER=nr
DBITMAP      [WINDOW=3; TITLE= 'White replaced by black'] repl
"Example 47.4"
CALCULATE    satu = IMSATURATE(matRGB ; 1.8)
CLOSE        nr; FILETYPE=graphics
SETDEVICE    'Exa47d.png'; NUMBER=nr
DBITMAP      [WINDOW=3; TITLE= 'Increased saturation'] satu
"Example 47.5"
CALCULATE    blur = IMGBLUR(matRGB ; 0.5)
CLOSE        nr; FILETYPE=graphics
SETDEVICE    'Exa47e.png'; NUMBER=nr
DBITMAP      [WINDOW=3; TITLE= 'Gaussian blur'] blur
"Example 47.6"
CALCULATE    sharpen = IMSHARPEN(matRGB ; 50)
CLOSE        nr; FILETYPE=graphics
SETDEVICE    'Exa47f.png'; NUMBER=nr
DBITMAP      [WINDOW=3; TITLE= 'Sharpen'] sharpen
STOP
```


Default colours of the first 32 pens
1
2
3
4
Horizontal flip of the colours
1
2
3
4
Pen
White replaced by black
1
2
3
4
Pen
Increased saturation
1
2
3
4
Pen
Gaussian blur
1
2
3
4
Pen
Sharpen
1
2
3
4
Pen

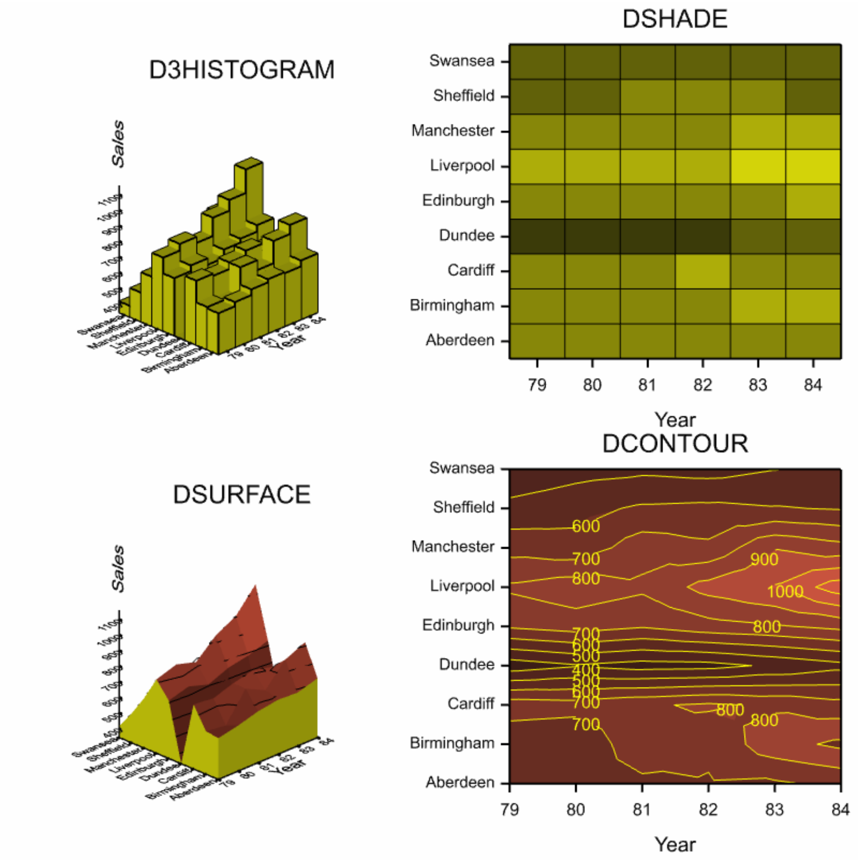
Example 47

```
JOB         'Example 48
            Directive BARCHART: parameters ERRORBAR and LOWERERRORBAR'
FACTOR      [LABELS= !T(A, B)] f1
FACTOR      [LABELS= !T(I, II, III)] f2
TABLE       [CLASS= f1, f2; VALUES= 4, 7, 2, 7, 12, 1] tab
TABLE       [CLASS= f1, f2; VALUES= 1,2,0.5,1,3,0.2] error
TABLE       [CLASS= f1, f2; VALUES= 6(0)] lower
calc        x=grnormal(10;10;1)
setdevice    'Exa48.png'; number=nr
BARCHART    [WINDOW=5; KEYWINDOW=0] tab; ERRORBAR=error
BARCHART    [WINDOW=6; KEYWINDOW=0; SCREEN=keep] \
            tab; ERRORBAR=error; LOWERERRORBAR=lower
STOP
```



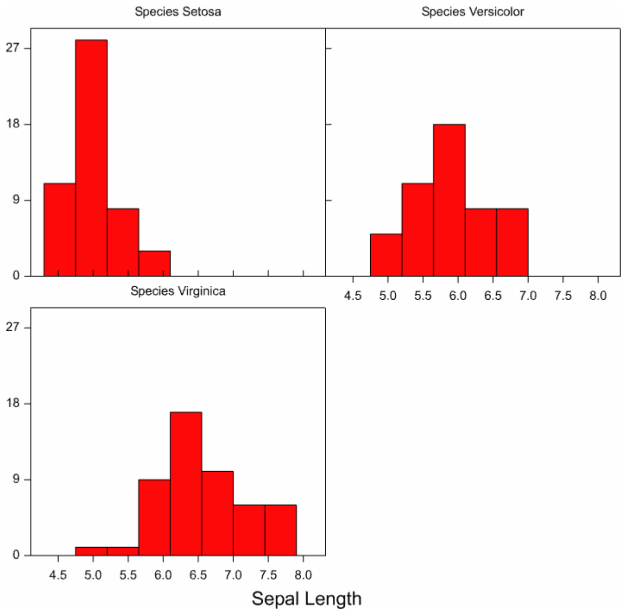
Example 48

```
JOB          'Example 49
             Plotting the same table with 4 different directives'
TEXT         [VAL= Aberdeen, Birmingham, Cardiff, Dundee, Edinburgh, \
             Liverpool, Manchester, Sheffield, Swansea] Town
VARIATE      [VAL= 79 ... 84] Year; DECIMALS= 0
FACTOR       [NVALUES=54 ; LABELS= Town] Factor1
FACTOR       [NVALUES=54 ; LEVELS= Year] Factor2
TABLE        [CLASSIFICATION= Factor1, Factor2] TabSales
READ         TabSales
    608 635 672 692  685  723       618 601 784 720  863  921
    757 743 785 816  783  737       343 391 358 366  418  470
    714 751 710 763  788  830       816 859 820 938 1007 1158
    662 632 758 721  893  837       531 569 615 624  607  593
    416 461 478 462  497  520 :
VARIATE      vTabSales ; VALUES=TabSales
GENERATE     Factor1, Factor2
FFRAME       [ROWS= 2; COLUMNS= 2; MARGIN= ytitle, xtitle]
XAXIS        1...4; TITLE= 'Year'; MARKS= Year
YAXIS        1...4; MARKS= !(1...9); LABELS= Town; \
             LDIRECTION=perpendicular
ZAXIS        1...4; TITLE= 'Sales'; TPOSITION=end
PEN          -1,-3; SIZE=2.5,1.7
PEN          5; COLOUR= 'tomato'
D3HISTOGRAM  [WINDOW= 1; TITLE= 'D3HISTOGRAM'] TabSales ; PEN=7
PEN          -1,-3; SIZE=1.2,1
DSHADE       [WINDOW= 2; KEYWINDOW=0; SCREEN= keep; TITLE= 'DSHADE'; \
             YORIENTATION=normal] TabSales; NGROUPS=9; PEN=7
PEN          -1,-3; SIZE=2.5,1.7
DSURFACE     [WINDOW= 3; KEYWINDOW=0; SCREEN= keep; TITLE= 'DSURFACE'] \
             TabSales; PENFILL=5; PENMESH=0; PENSIDE=7
PEN          -1,-3; SIZE=1.2,1
DCONTOUR     [WINDOW= 4; KEYWINDOW=0; SCREEN= keep; TITLE= 'DCONTOUR'; \
             YORIENTATION=normal] TabSales ; PENCONTOUR=7 ; PENFILL=5
STOP
```

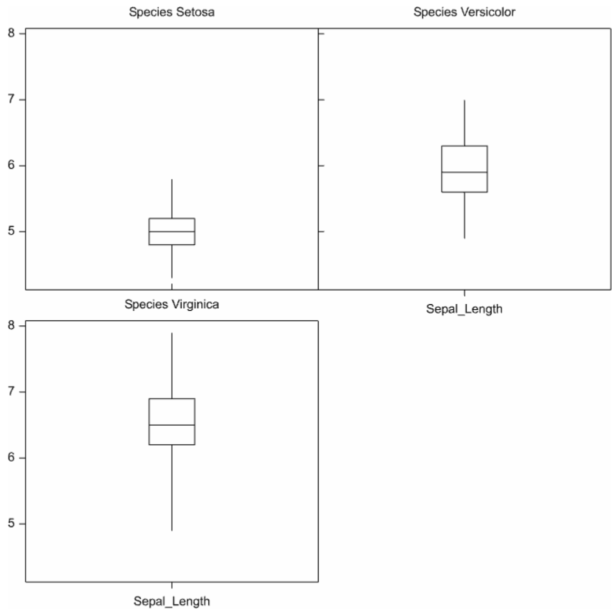


Example 49

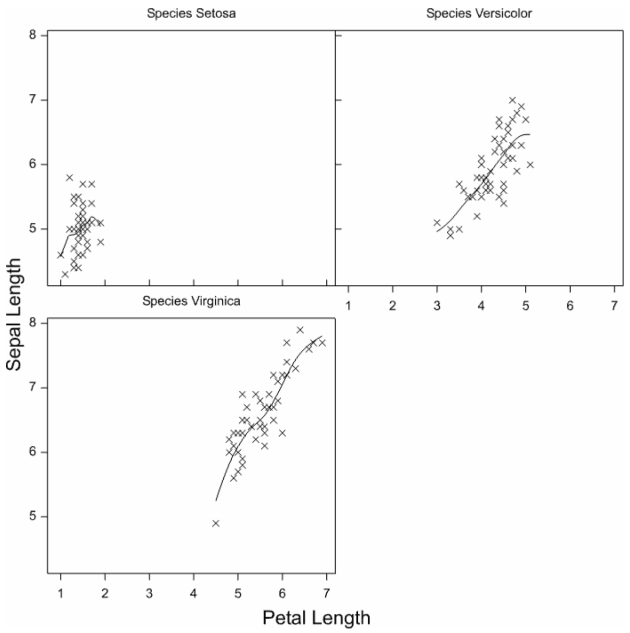
```
JOB         'Example 50
            Procedure TRELLIS: option METHOD'
SPLOAD      '%GENDIR%/Data/Iris.gsh'
XAXIS       1; TITLE= 'Sepal Length'
TABULATE    [CLASS=Species] Sepal_Length; MEAN=Tab
"Example 50.1"
TRELLIS     [GROUPS=Species; XTITLE= 'Sepal Length'] Sepal_Length ; \
            METHOD=histogram
"Example 50.2"
TRELLIS     [GROUPS=Species] Sepal_Length ; METHOD=boxplot
"Example 50.3"
TRELLIS     [GROUPS=Species ; YTITLE= 'Sepal Length'; XTITLE= 'Petal Length'; \
            KEYHEIGHT=0] 2(Sepal_Length) ; X=2(Petal_Length); \
            METHOD=spline,point
"Example 50.4"
TRELLIS     [GROUPS=Species ; YTITLE= 'Sepal Length'; XTITLE= 'Petal Length'; \
            KEYHEIGHT=0] 2(Sepal_Length) ; X=2(Petal_Length); \
            METHOD=mean,point
STOP
```



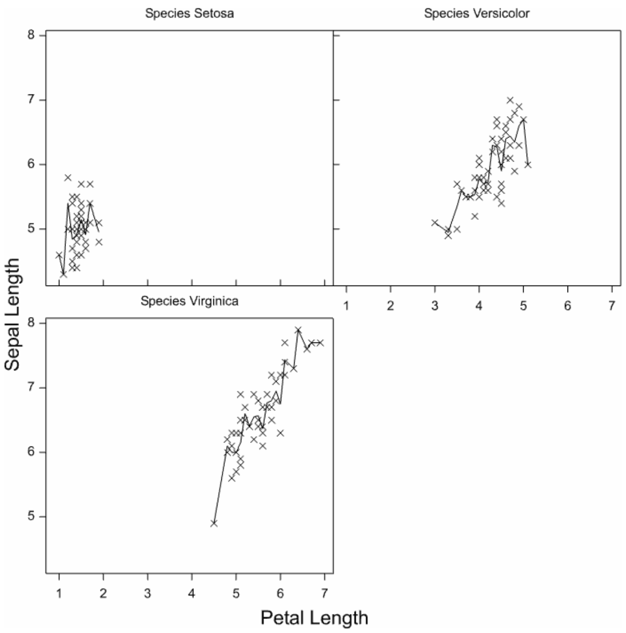
Example 50.1



Example 50.2

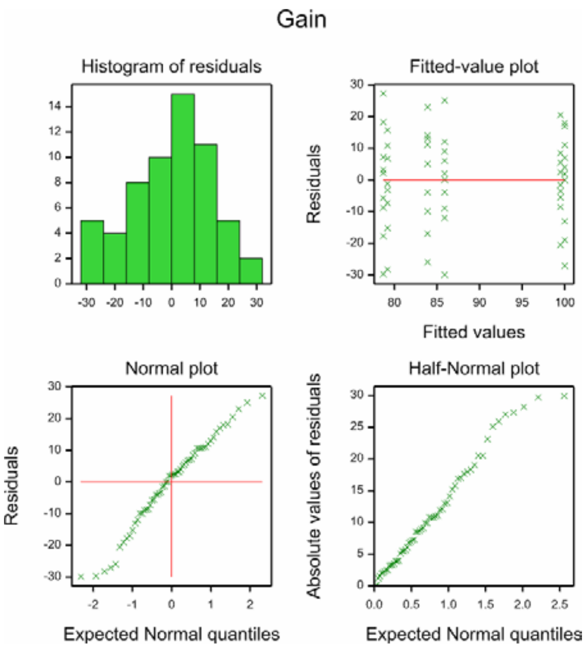


Example 50.3

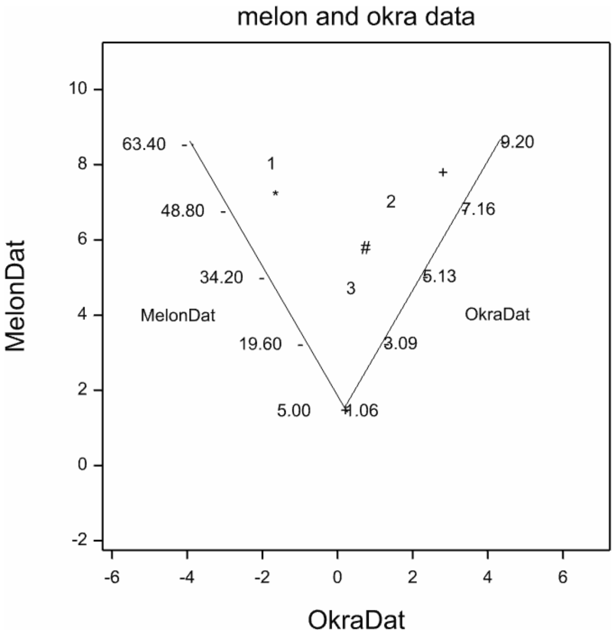


Example 50.4

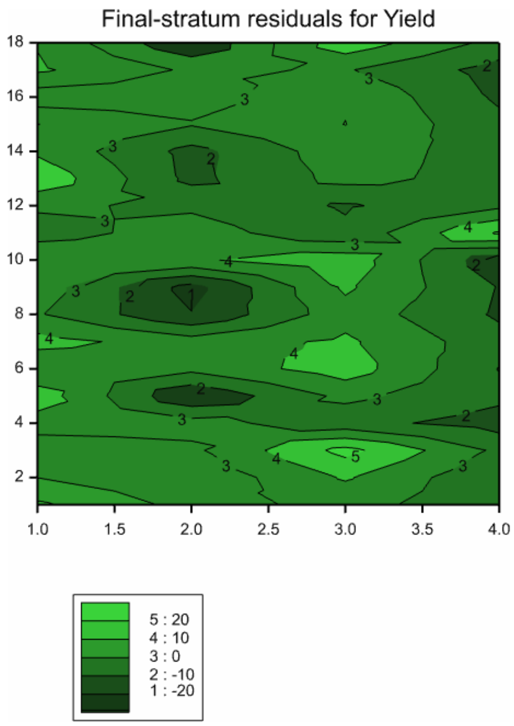
```
JOB         'Example 51
            4 examples of ANOVA Procedures (see also Appendix 2)
     You can also run these examples with Help → Examples → Commands...
     and then choose the procedure or directive'
EXAMPLE     [EXECUTE=yes] 'plib'; EXNAME='APLOT'; SOURCE=APLO : ##APLO
EXAMPLE     [EXECUTE=yes] 'plib'; EXNAME='ABIVARIATE'; SOURCE=ABIV : ##ABIV
EXAMPLE     [EXECUTE=yes] 'plib'; EXNAME='AFIELDRESIDUALS'; SOURCE=AFIE : ##AFIE
DELETE      [LIST=excl; REDE=y]
EXAMPLE     [EXECUTE=yes] 'plib'; EXNAME='AMCOMPARISON'; SOURCE=AMCO : ##AMCO
STOP
```



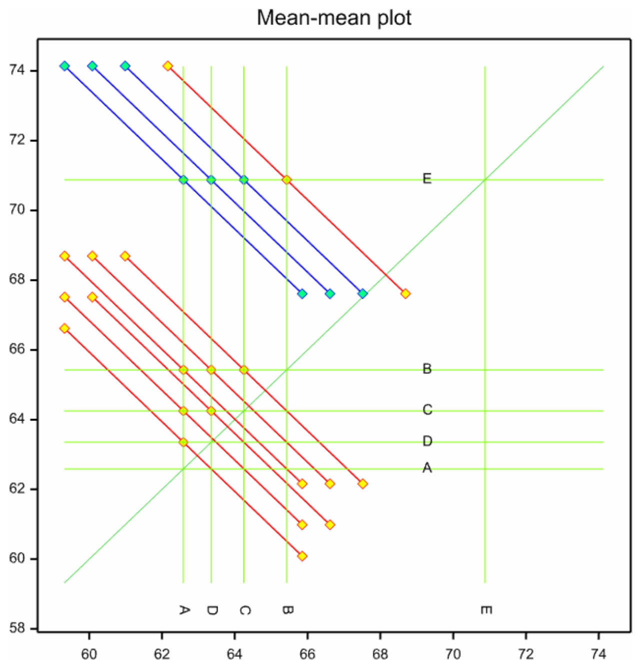
Example of APLOT



Example of ABIVARIATE

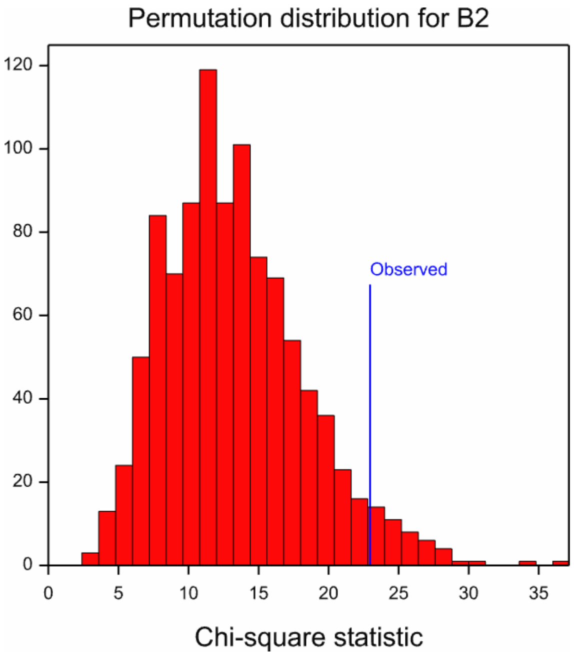


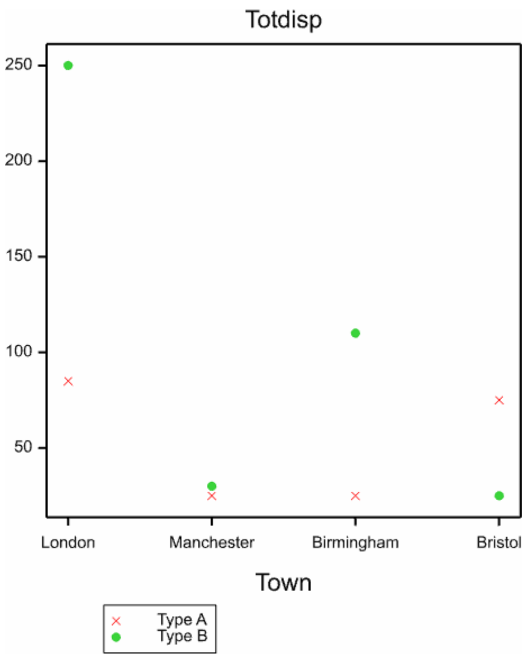
Example of AFIELDRESIDUALS



Example of AMCOMPARISON

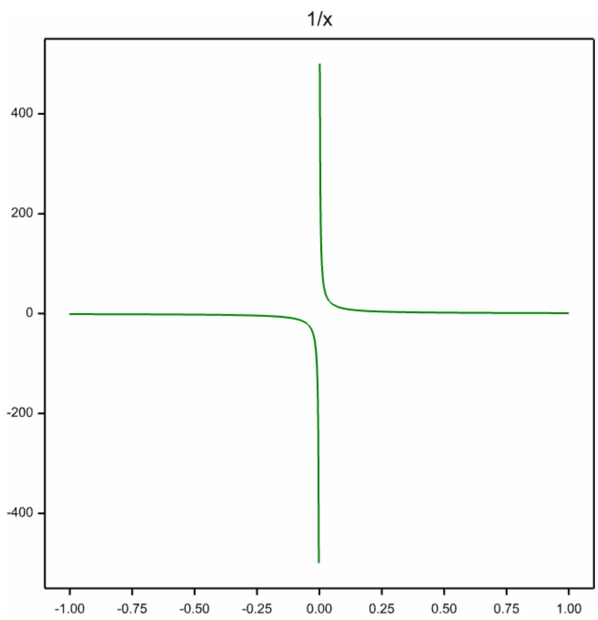
```
JOB         'Example 52
            4 examples of BASIC Procedures (see also Appendix 2)
     You can also run these examples with Help → Examples → Commands...
     and then choose the procedure or directive'
EXAMPLE     [EXECUTE=yes] 'plib'; EXNAME='CHIPERMTEST'; SOURCE=CHIP : ##CHIP
EXAMPLE     [EXECUTE=yes] 'plib'; EXNAME='DTABLE'; SOURCE=DTAB : ##DTAB
EXAMPLE     [EXECUTE=yes] 'plib'; EXNAME='DFUNCTION'; SOURCE=DFUN : ##DFUN
EXAMPLE     [EXECUTE=yes] 'plib'; EXNAME='RUGPLOT'; SOURCE=RUGP : ##RUGP
                                         STOP
```

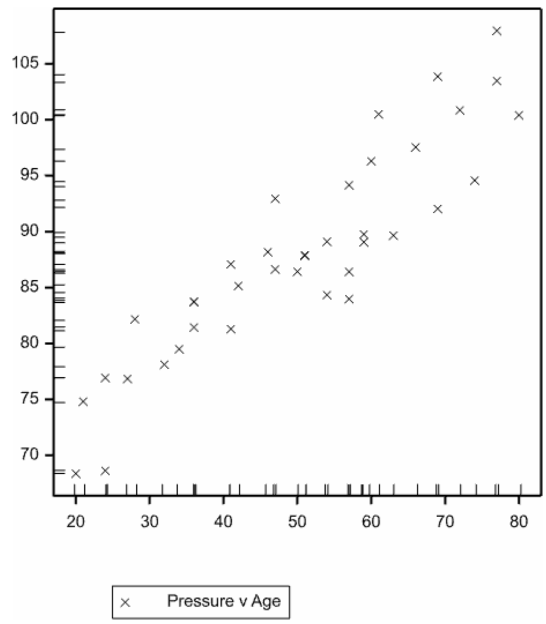




Example of CHIPERMTEST

Example of DTABLE



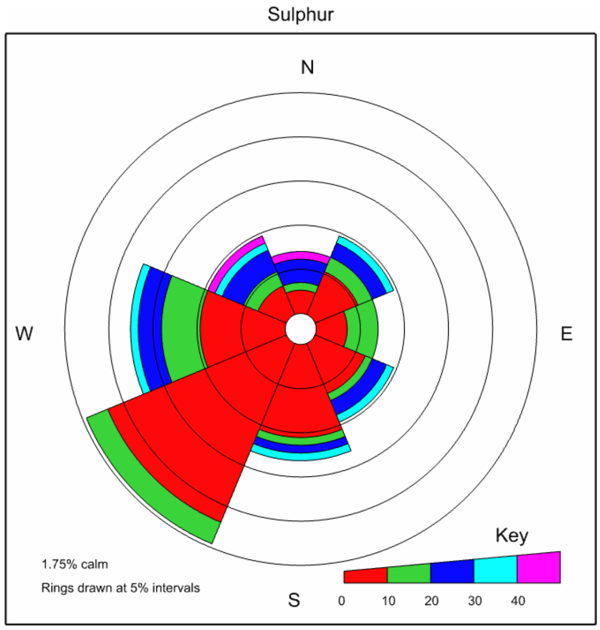


Example of DFUNCTION

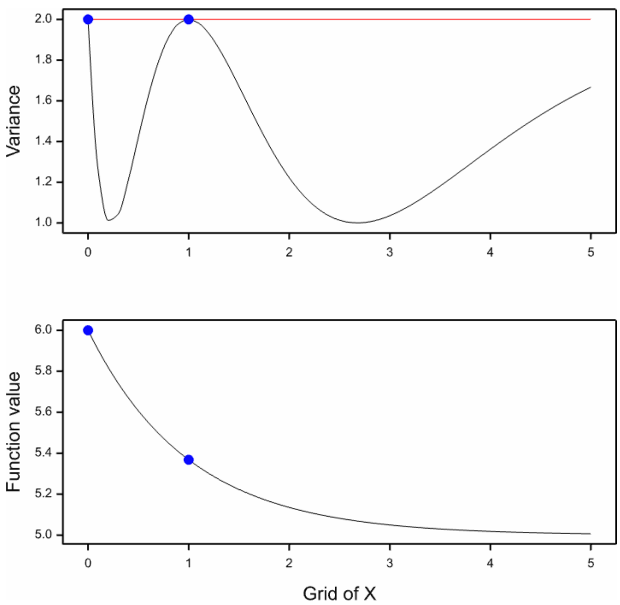
Example of RUGPLOT

```
JOB        'Example 53
         2 examples of CIRCULAR and 2 of DESIGN Procedures (see also Appendix 2)
     You can also run these examples with Help → Examples → Commands...
     and then choose the procedure or directive'
EXAMPLE     [EXECUTE=yes] 'plib'; EXNAME='DCIRCULAR'; SOURCE=DCIR : ##DCIR
EXAMPLE     [EXECUTE=yes] 'plib'; EXNAME='WINDROSE'; SOURCE=WIND : ##WIND
EXAMPLE     [EXECUTE=yes] 'plib'; EXNAME='AFNONLINEAR'; SOURCE=AFNO : ##AFNO
EXAMPLE     [EXECUTE=yes] 'plib'; EXNAME='DDESIGN'; SOURCE=DDES : ##DDES
STOP
```

Example of DCIRCULAR



Example of WINDROSE

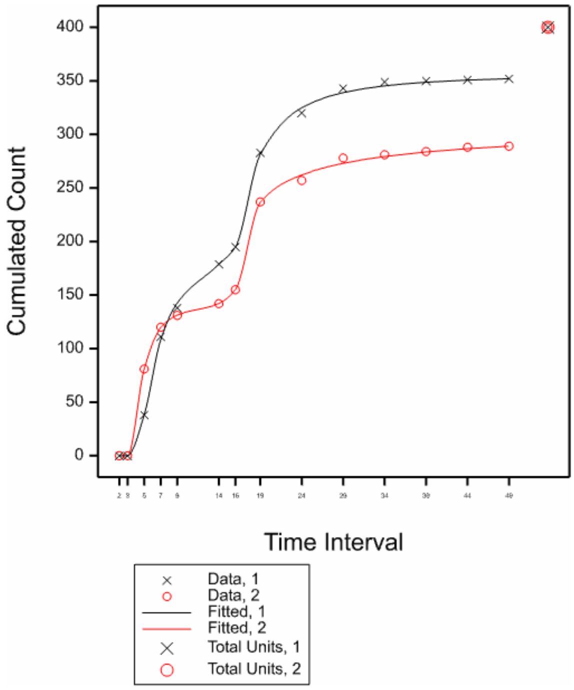


Example of AFNONLINEAR

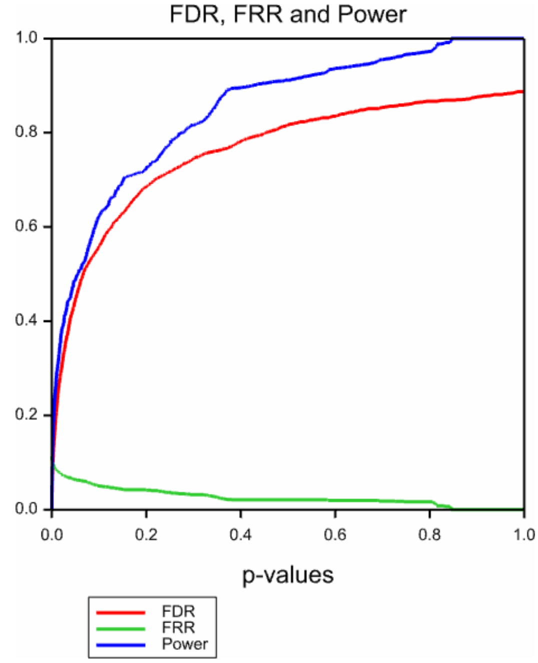
Marvellous 0.6 cwt	Marvellous 0.4 cwt	Marvellous 0.2 cwt	Marvellous 0 cwt
Victory 0 cwt	Victory 0.2 cwt	Victory 0.6 cwt	Victory 0.4 cwt
Golden rain 0 cwt	Golden rain 0.2 cwt	Golden rain 0.4 cwt	Golden rain 0.6 cwt
Marvellous 0.4 cwt	Marvellous 0 cwt	Marvellous 0.2 cwt	Marvellous 0.6 cwt
Victory 0.6 cwt	Victory 0 cwt	Victory 0.2 cwt	Victory 0.4 cwt
Golden rain 0.2 cwt	Golden rain 0 cwt	Golden rain 0.4 cwt	Golden rain 0.6 cwt
Golden rain 0.2 cwt	Golden rain 0.4 cwt	Golden rain 0.6 cwt	Golden rain 0 cwt
Marvellous 0.6 cwt	Marvellous 0.2 cwt	Marvellous 0.4 cwt	Marvellous 0 cwt
Victory 0 cwt	Victory 0.6 cwt	Victory 0.2 cwt	Victory 0.4 cwt
Marvellous 0.4 cwt	Marvellous 0.6 cwt	Marvellous 0 cwt	Marvellous 0.2 cwt
Golden rain 0 cwt	Golden rain 0.4 cwt	Golden rain 0.6 cwt	Golden rain 0.2 cwt
Victory 0.2 cwt	Victory 0.4 cwt	Victory 0.6 cwt	Victory 0 cwt
Golden rain 0.6 cwt	Golden rain 0 cwt	Golden rain 0.4 cwt	Golden rain 0.2 cwt
Victory 0.4 cwt	Victory 0.6 cwt	Victory 0 cwt	Victory 0.2 cwt
Marvellous 0.4 cwt	Marvellous 0.6 cwt	Marvellous 0.2 cwt	Marvellous 0 cwt
Victory 0.4 cwt	Victory 0 cwt	Victory 0.6 cwt	Victory 0.2 cwt
Golden rain 0.6 cwt	Golden rain 0.4 cwt	Golden rain 0 cwt	Golden rain 0.2 cwt
Marvellous 0 cwt	Marvellous 0.2 cwt	Marvellous 0.4 cwt	Marvellous 0.6 cwt

Example of DDESIGN

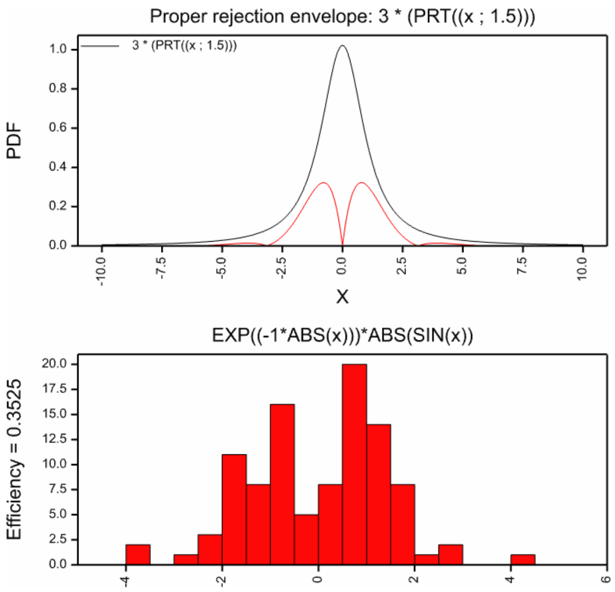
```
JOB         'Example 54
            4 examples of DISTRIBUTION Procedures (see also Appendix 2)
     You can also run these examples with Help → Examples → Commands...
     and then choose the procedure or directive'
EXAMPLE     [EXECUTE=yes] 'plib'; EXNAME='CUMDISTRIBUTION'; SOURCE=CUMD : ##CUMD
EXAMPLE     [EXECUTE=yes] 'plib'; EXNAME='FDRBONFERRONI'; SOURCE=FDRB : ##FDRB
DELETE      [LIST=excl; REDE=y]
EXAMPLE     [EXECUTE=yes] 'plib'; EXNAME='GREJECTIONSAMPLE'; SOURCE=GREJ: ##GREJ
EXAMPLE     [EXECUTE=yes] 'plib'; EXNAME='KERNELDENSITY'; SOURCE=KERN : ##KERN
STOP
```



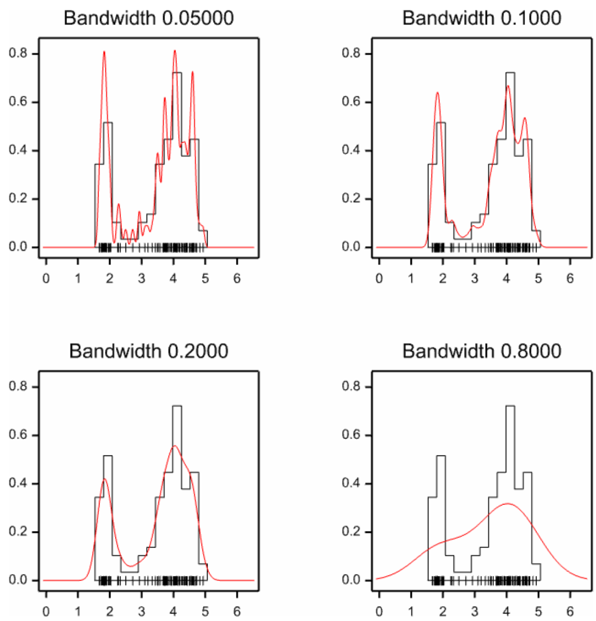
Example of CUMDISTRIBUTION



Example of FDRBONFERRONI

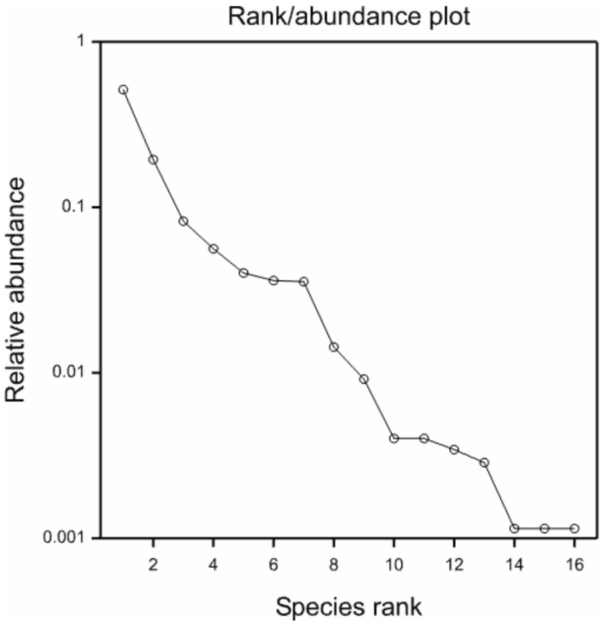


Example of GREJECTIONSAMPLE

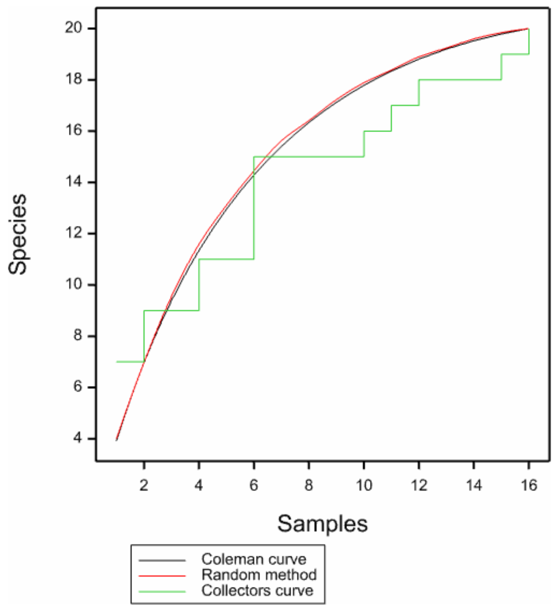


Example of KERNELDENSITY

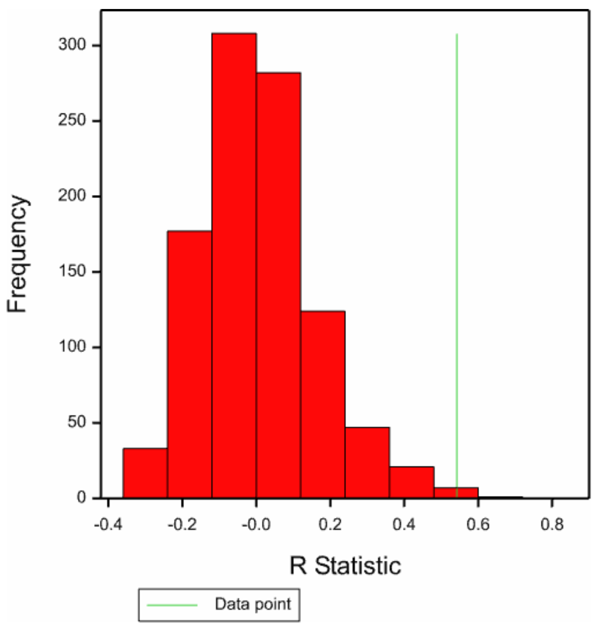
```
JOB         'Example 55
            4 examples of ECOLOGY Procedures (see also Appendix 2)
     You can also run these examples with Help → Examples → Commands...
     and then choose the procedure or directive'
EXAMPLE     [EXECUTE=yes] 'plib'; EXNAME='ECABUNDANCEPLOT'; SOURCE=ECAB : ##ECAB
EXAMPLE     [EXECUTE=yes] 'plib'; EXNAME='ECACCUMULATION'; SOURCE=ECAC : ##ECAC
EXAMPLE     [EXECUTE=yes] 'plib'; EXNAME='ECANOSIM'; SOURCE=ECAN : ##ECAN
EXAMPLE     [EXECUTE=yes] 'plib'; EXNAME='ECRAREFACTION'; SOURCE=ECRA : ##ECRA
STOP
```



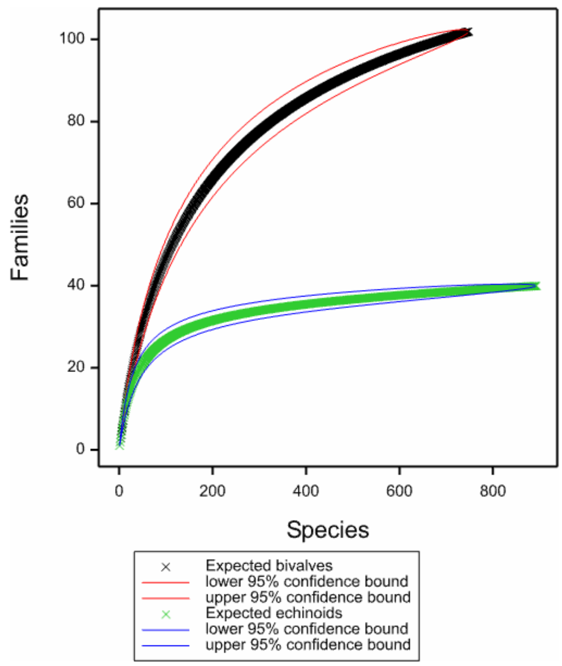
Example of ECABUNDANCEPLOT



Example of ECACCUMULATION

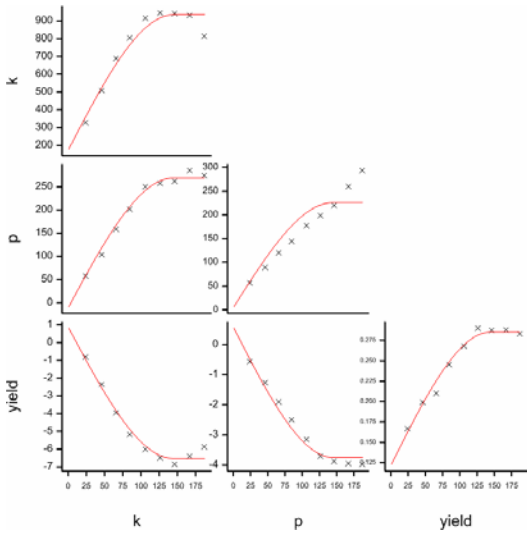


Example of ECANOSIM

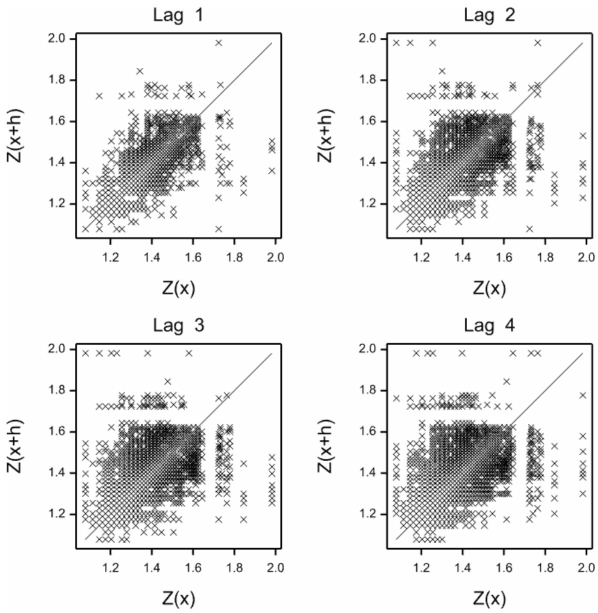


Example of ECRAREFACTION

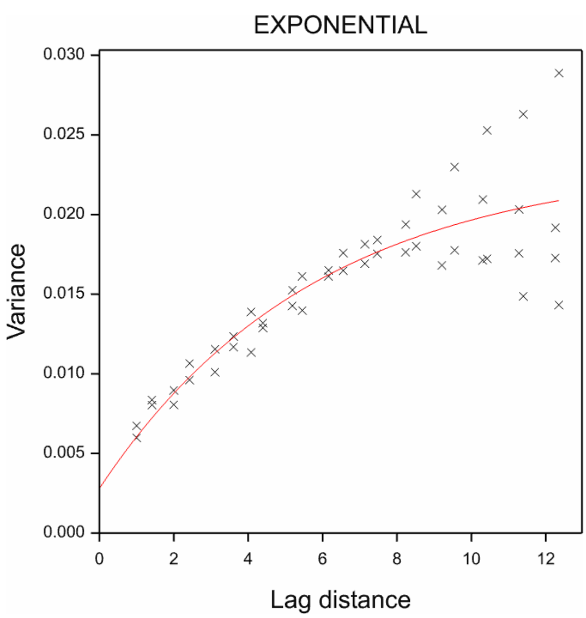
```
JOB         'Example 56
            4 examples of GEOSTATISTICS Procedures (see also Appendix 2)
     You can also run these examples with Help → Examples → Commands...
     and then choose the procedure or directive'
EXAMPLE     [EXECUTE=yes] 'plib'; EXNAME='DCOVARIOGRAM'; SOURCE=DCOV : ##DCOV
EXAMPLE     [EXECUTE=yes] 'plib'; EXNAME='DHSCATTERGRAM'; SOURCE=DHSC : ##DHSC
EXAMPLE     [EXECUTE=yes] 'plib'; EXNAME='DVARIOGRAM'; SOURCE=DVAR : ##DVAR
EXAMPLE     [EXECUTE=yes] 'plib'; EXNAME='KCROSSVALIDATION'; SOURCE=KCRO: ##KCRO
KCROSSVALIDATION [PRINT=stat,corr; PLOT=scatter; \
                 Y=y; X=x; RADIUS=5; MINP=7; MAXP=20]\
                 DATA=logk; MODEL=spherical; NUGGET=0.00466; SILL=0.01515;\
                 RANGE=10.8; STAT=statsSTOP
```



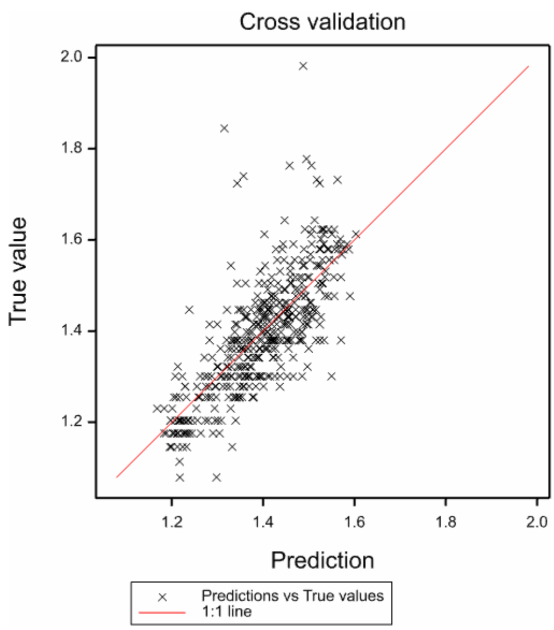
Example of DCOVARIOGRAM



Example of DHSCATTERGRAM

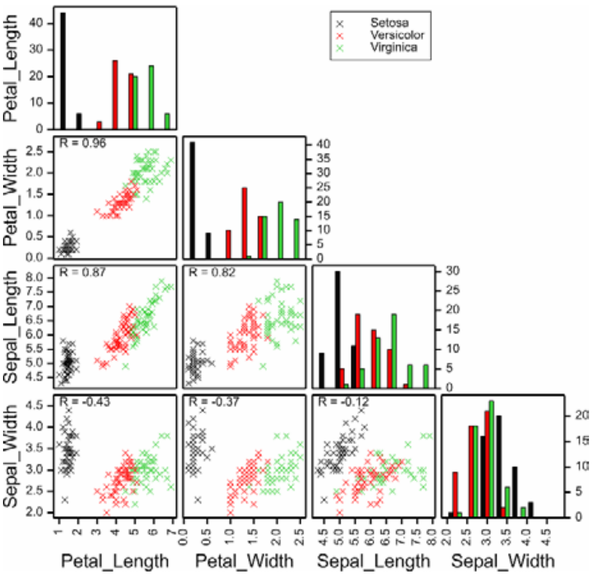


Example of DVARIOGRAM

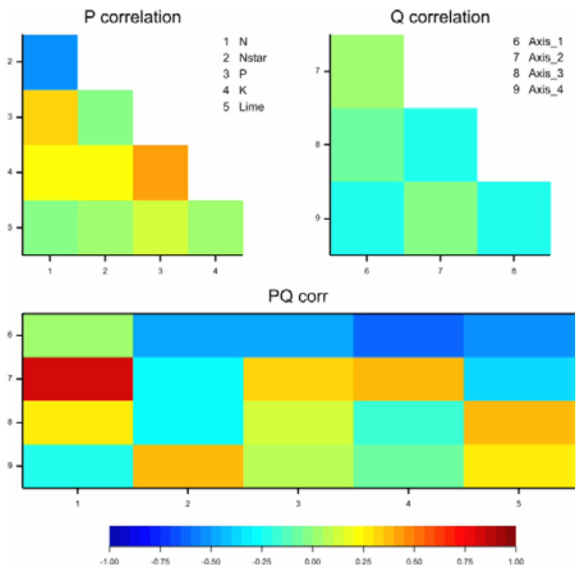


Example of KCROSSVALIDATION

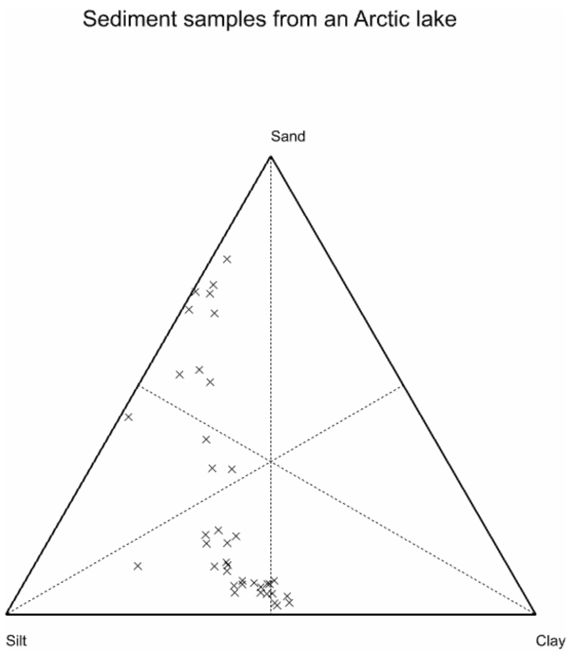
```
JOB         'Example 57
            4 examples of GRAPHICS Procedures (see also Appendix 2)
     You can also run these examples with Help → Examples → Commands...
     and then choose the procedure or directive'
EXAMPLE     [EXECUTE=yes] 'plib'; EXNAME='DMSCATTER'; SOURCE=DMSC : ##DMSC
DELETE      [REDE=yes] Species
EXAMPLE     [EXECUTE=yes] 'plib'; EXNAME='DCORRELATION'; SOURCE=DCOR : ##DCOR
EXAMPLE     [EXECUTE=yes] 'plib'; EXNAME='DCOMPOSITIONAL'; SOURCE=DCOM : ##DCOM
DELETE      [REDE=yes] Species
EXAMPLE     [EXECUTE=yes] 'plib'; EXNAME='DOTHISTOGRAM'; SOURCE=DOTH : ##DOTH
STOP
```



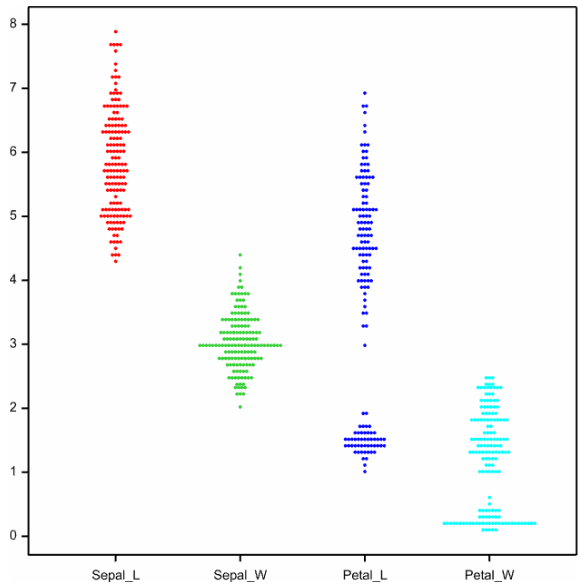
Example of DMSCATTER



Example of DCORRELATION

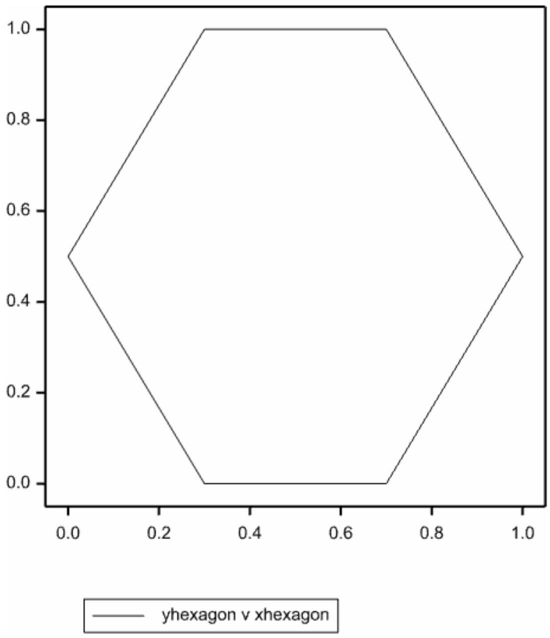


Example of DCOMPOSITIONAL

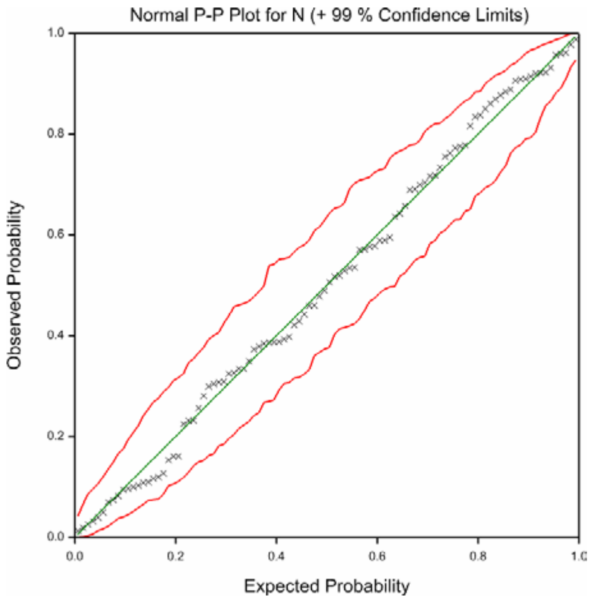


Example of DOTHISTOGRAM

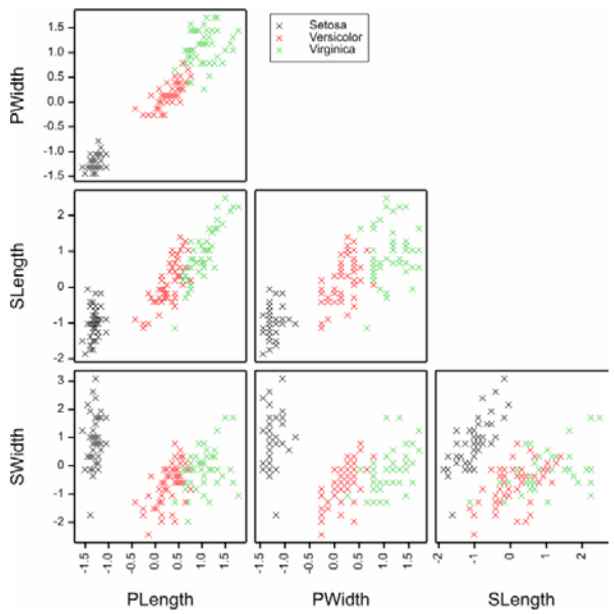
```
JOB          'Example 58
             4 examples of GRAPHICS Procedures (see also Appendix 2)
     You can also run these examples with Help → Examples → Commands...
     and then choose the procedure or directive'
EXAMPLE      [EXECUTE=yes] 'plib'; EXNAME='DPOLYGON'; SOURCE=DPOL : ##DPOL
EXAMPLE      [EXECUTE=yes] 'plib'; EXNAME='DPROBABILITY'; SOURCE=DPRO : ##DPRO
EXAMPLE      [EXECUTE=yes] 'plib'; EXNAME='DSCATTER'; SOURCE=DSCA : ##DSCA
EXAMPLE      [EXECUTE=yes] 'plib'; EXNAME='DXYGRAPH'; SOURCE=DXYG : ##DXYG
STOP
```



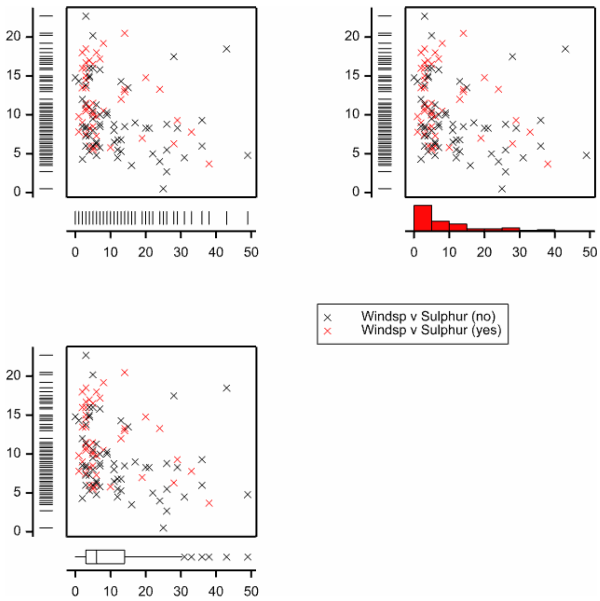
Example of DPOLYGON



Example of DPROBABILITY

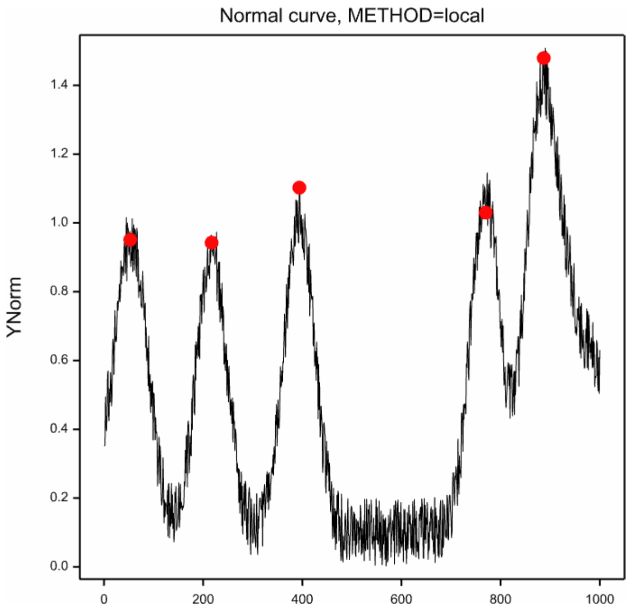


Example of DSCATTER

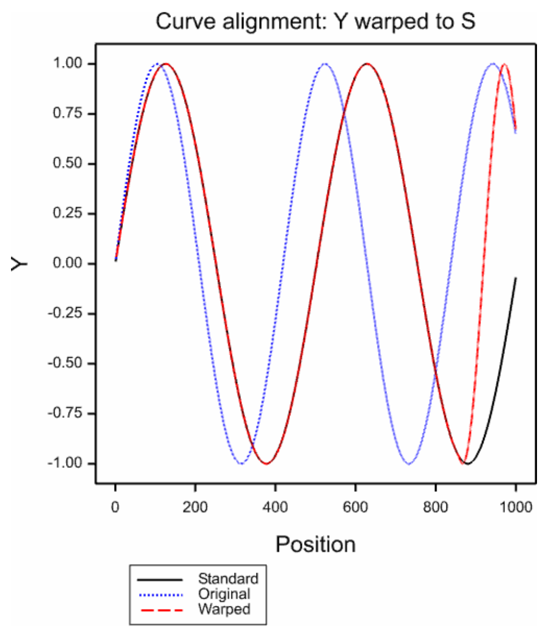


Example of DXYGRAPH

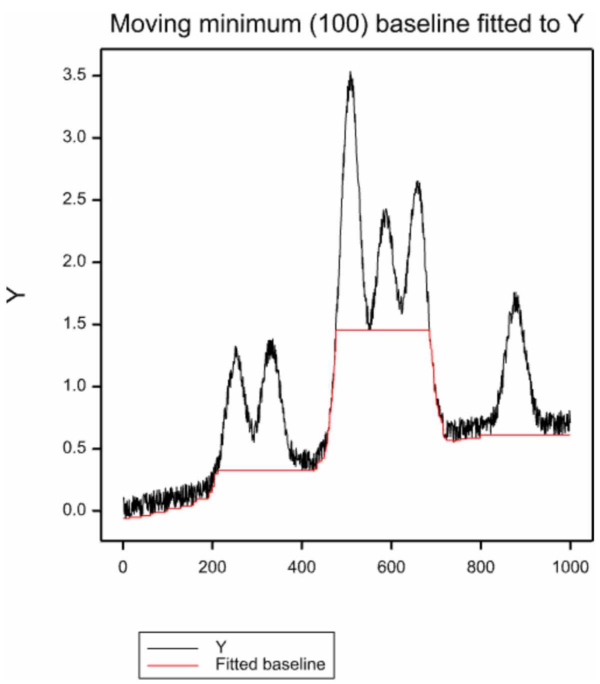
```
JOB         'Example 59
            4 examples of MANIPULATION Procedures (see also Appendix 2)
     You can also run these examples with Help → Examples → Commands...
     and then choose the procedure or directive'
EXAMPLE     [EXECUTE=yes] 'plib'; EXNAME='PEAKFINDER'; SOURCE=PEAK : ##PEAK
DELETE      [LIST=excl;REDE=y]
EXAMPLE     [EXECUTE=yes] 'plib'; EXNAME='ALIGNCURVE'; SOURCE=ALIG : ##ALIG
EXAMPLE     [EXECUTE=yes] 'plib'; EXNAME='BASELINE'; SOURCE=BASE : ##BASE
EXAMPLE     [EXECUTE=yes] 'plib'; EXNAME='RLASSO'; SOURCE=RLAS : ##RLAS
STOP
```



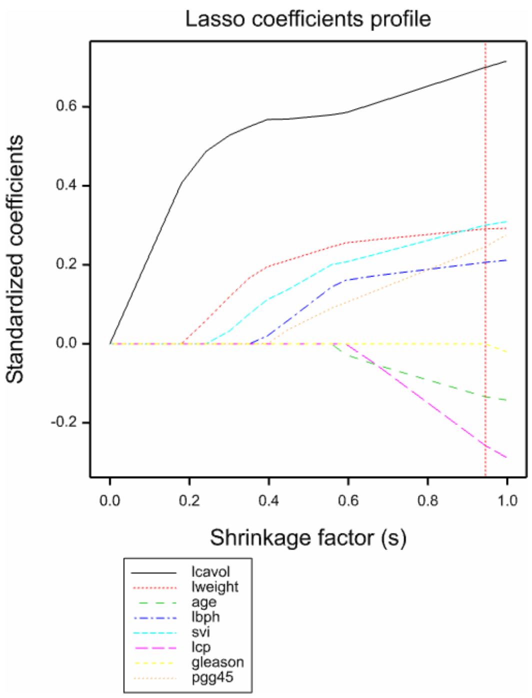
Example of PEAKFINDER



Example of ALIGNCURVE

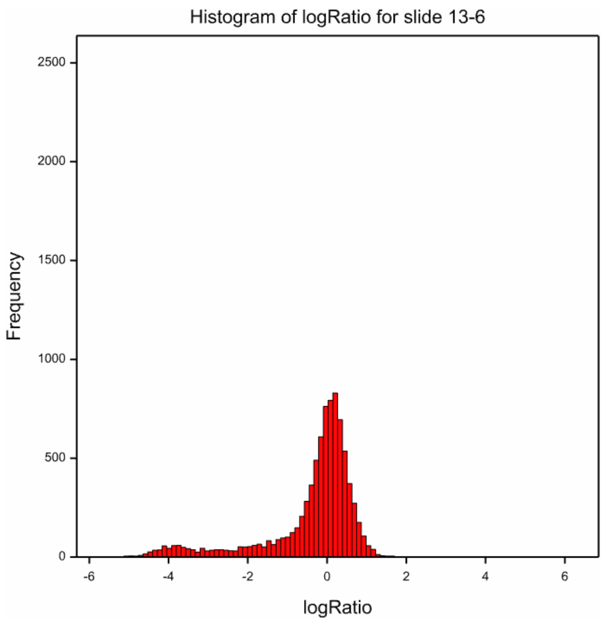


Example of BASELINE

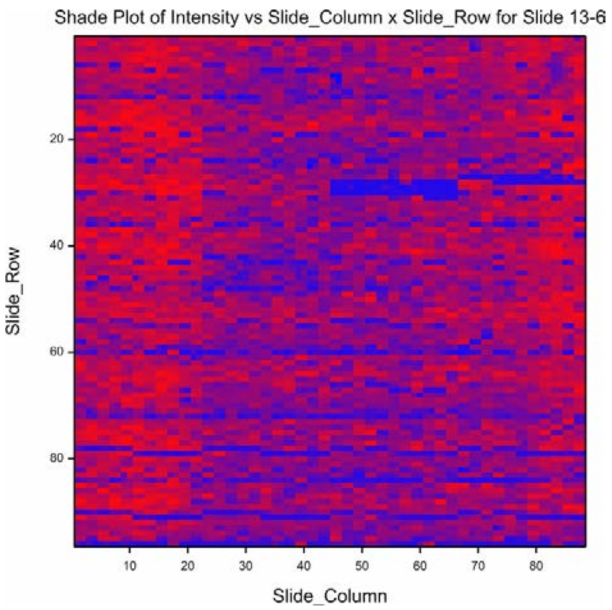


Example of RLASSO

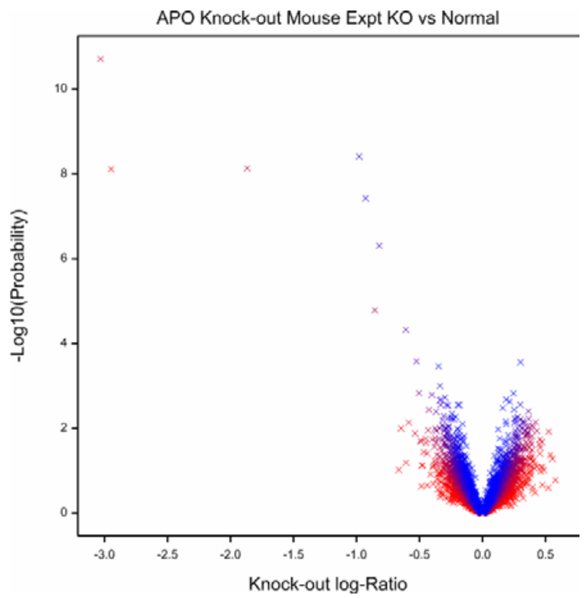
```
JOB         'Example 60
            4 examples of MICROARRAYS Procedures (see also Appendix 2)
     You can also run these examples with Help → Examples → Commands...
     and then choose the procedure or directive'
EXAMPLE     [EXECUTE=yes] 'plib'; EXNAME='MAHISTOGRAM'; SOURCE=MAHI : ##MAHI
EXAMPLE     [EXECUTE=yes] 'plib'; EXNAME='MASHADE'; SOURCE=MASH : ##MASH
EXAMPLE     [EXECUTE=yes] 'plib'; EXNAME='MAVOLCANO'; SOURCE=MAVO : ##MAVO
EXAMPLE     [EXECUTE=yes] 'plib'; EXNAME='MNORMALIZE'; SOURCE=MNOR : ##MNOR
STOP
```



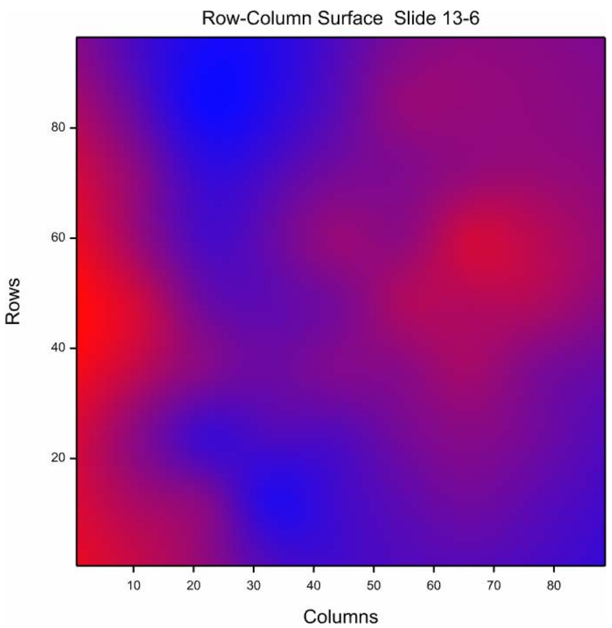
Example of MAHISTOGRAM



Example of MASHADE

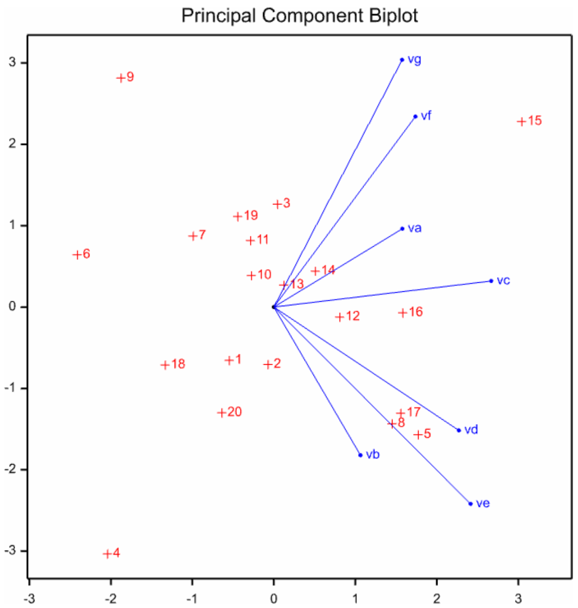


Example of MAVOLCANO

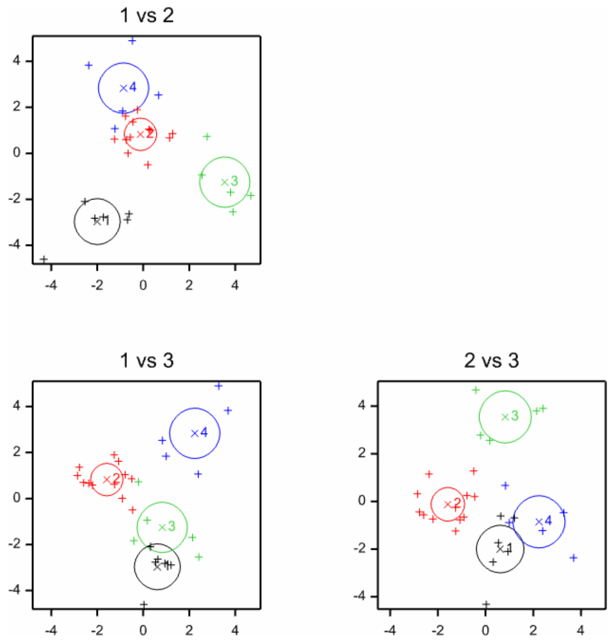


Example of MNORMALIZE

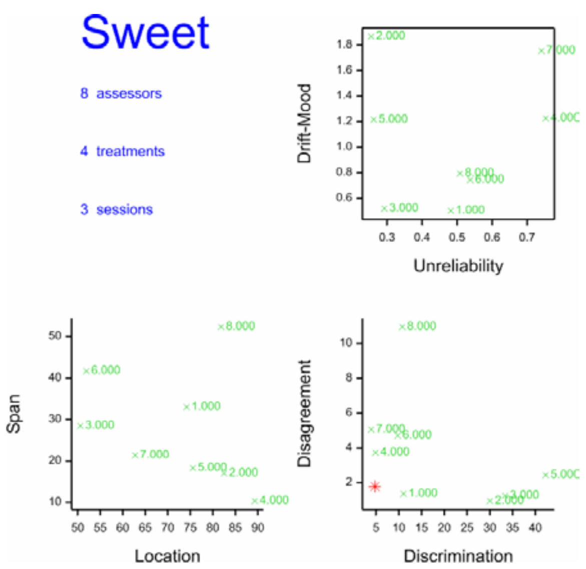
```
JOB         'Example 61
            4 examples of MVA Procedures (see also Appendix 2)
     You can also run these examples with Help → Examples → Commands...
     and then choose the procedure or directive'
EXAMPLE     [EXECUTE=yes] 'plib'; EXNAME='BIPLOT'; SOURCE=BIPL : ##BIPL
EXAMPLE     [EXECUTE=yes] 'plib'; EXNAME='CVAPLOT'; SOURCE=CVAP : ##CVAP
EXAMPLE     [EXECUTE=yes] 'plib'; EXNAME='SAGRAPES'; SOURCE=SAGR : ##SAGR
EXAMPLE     [EXECUTE=yes] 'plib'; EXNAME='SDISCRIMINATE'; SOURCE=SDIS : ##SDIS
STOP
```

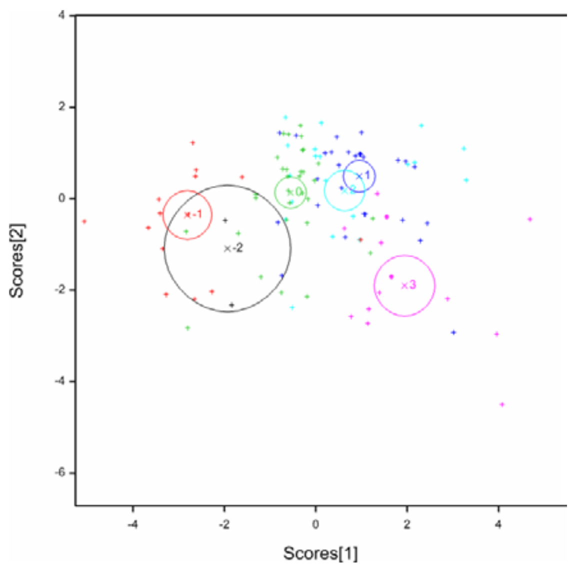


Example of BIPLOT



Example of CVAPLOT

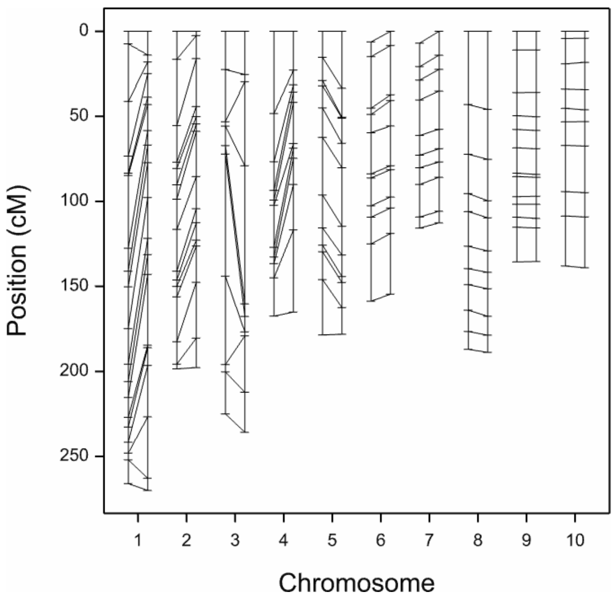




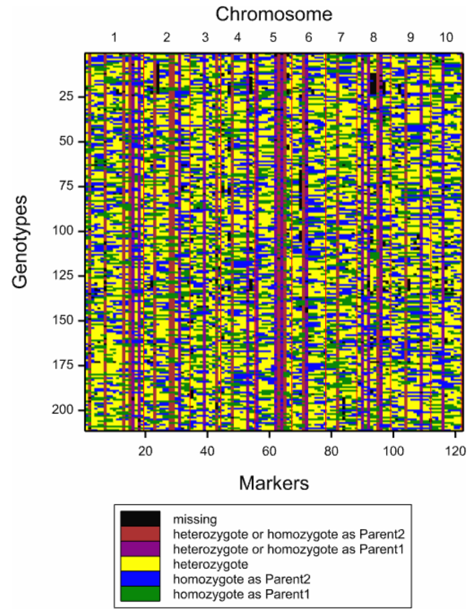
Example of SAGRAPES

Example of SDISCRIMINATE

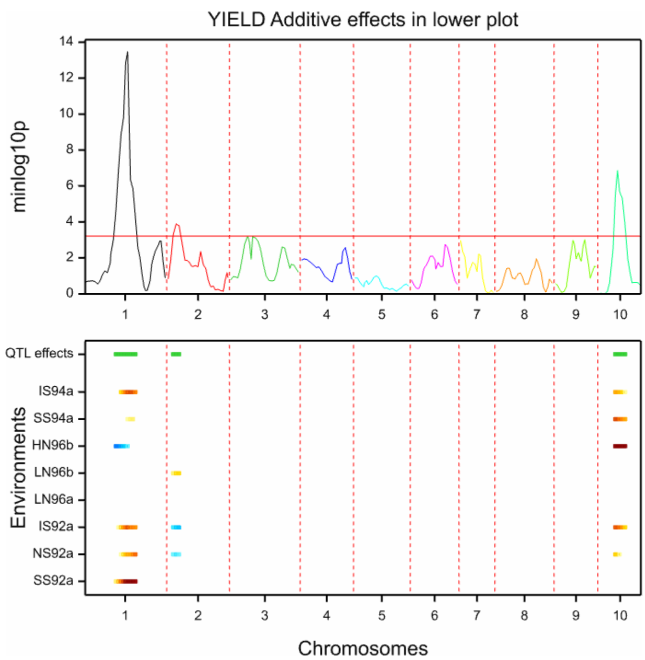
```
JOB          'Example 62
             4 examples of QTL Procedures (see also Appendix 2)
     You can also run these examples with Help → Examples → Commands...
     and then choose the procedure or directive'
EXAMPLE      [EXECUTE=yes] 'plib'; EXNAME='DQMAP'; SOURCE=DQMAP : ##DQMAP
EXAMPLE      [EXECUTE=yes] 'plib'; EXNAME='DQMKSCORES'; SOURCE=DQMK : ##DQMK
EXAMPLE      [EXECUTE=yes] 'plib'; EXNAME='DQMQTLSCAN'; SOURCE=DQMQ : ##DQMQ
EXAMPLE      [EXECUTE=yes] 'plib'; EXNAME='DQRECOMBINATIONS';SOURCE=DQRE : ##DQRE
STOP
```



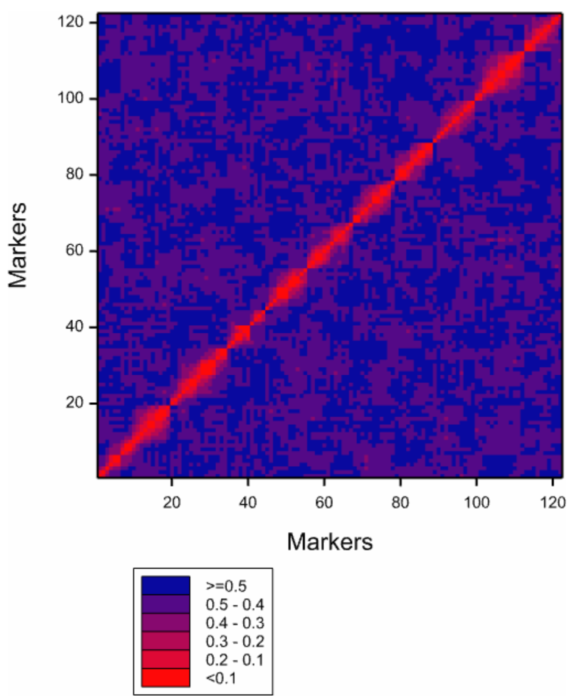
Example of DQMAP



Example of DQMKSCORES

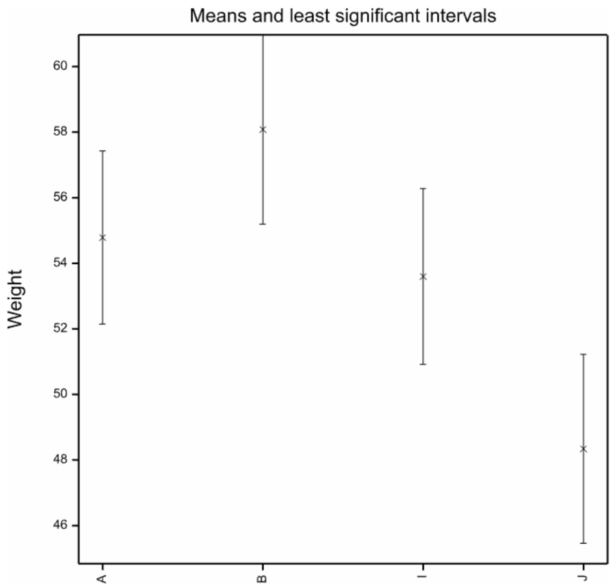


Example of DQMQTLSCAN

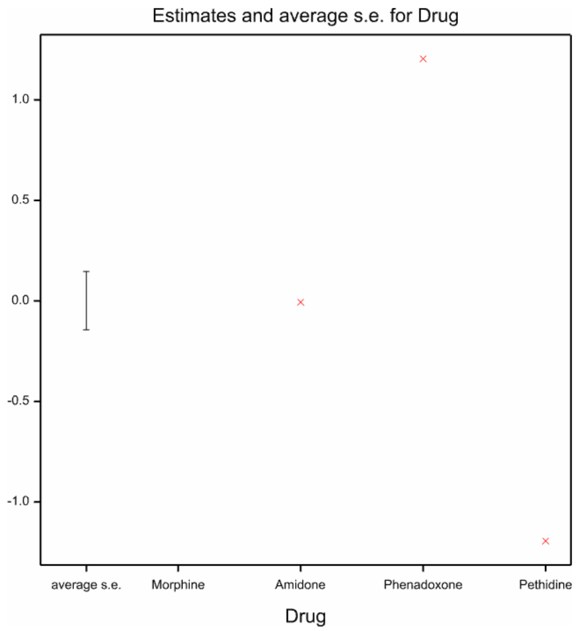


Example of DQRECOMBINATIONS

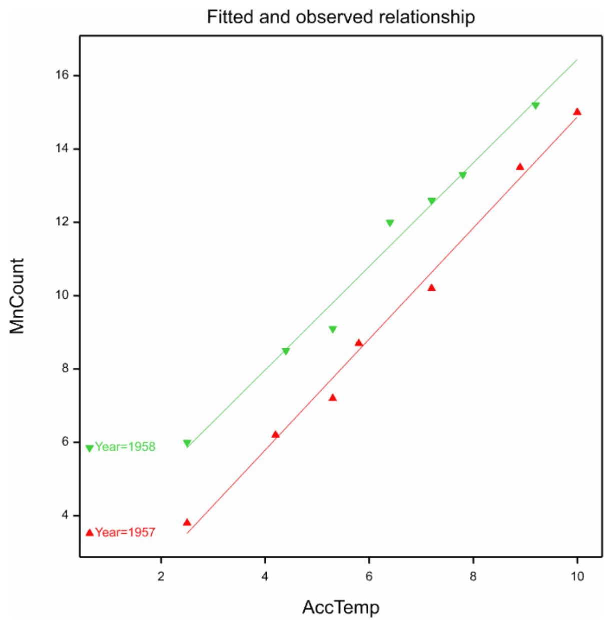
```
JOB         'Example 63
            4 examples of REGRESSION Procedures (see also Appendix 2)
     You can also run these examples with Help → Examples → Commands...
     and then choose the procedure or directive'
EXAMPLE     [EXECUTE=yes] 'plib'; EXNAME='LSIPLOT'; SOURCE=LSIP : ##LSIP
EXAMPLE     [EXECUTE=yes] 'plib'; EXNAME='RDESTIMATES'; SOURCE=RDES : ##RDES
EXAMPLE     [EXECUTE=yes] 'plib'; EXNAME='RGRAPH'; SOURCE=RGRA : ##RGRA
EXAMPLE     [EXECUTE=yes] 'plib'; EXNAME='RQNONLINEAR'; SOURCE=RQNO : ##RQNO
STOP
```



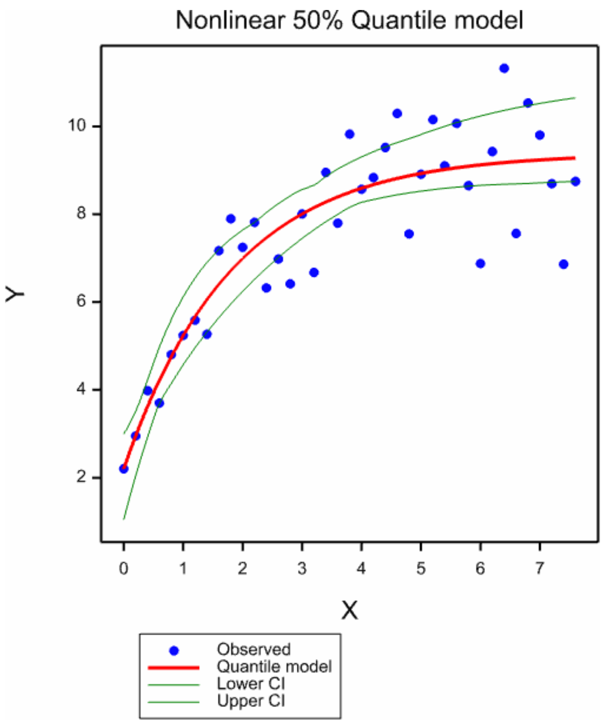
Example of LSIPLOT



Example of RDESTIMATES

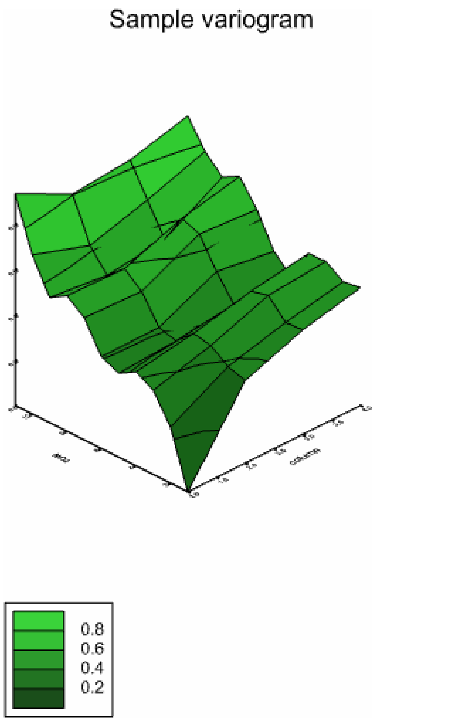


Example of RGRAPH

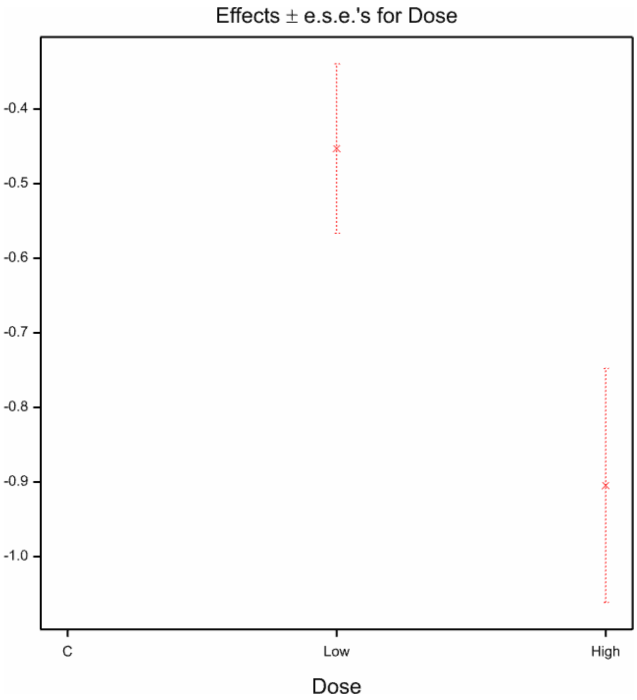


Example of RQNONLINEAR

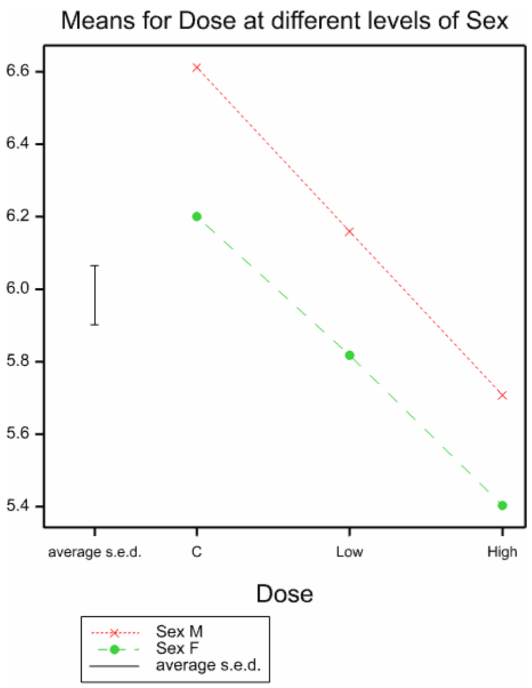
```
JOB         'Example 64
            4 examples of REML Procedures (see also Appendix 2)
     You can also run these examples with Help → Examples → Commands...
     and then choose the procedure or directive'
EXAMPLE     [EXECUTE=yes] 'plib'; EXNAME='F2DRESIDUALVARIOGRAM'; SOURCE=F2DR
##F2DR
EXAMPLE     [EXECUTE=yes] 'plib'; EXNAME='VDEFFECTS'; SOURCE=VDEF : ##VDEF
EXAMPLE     [EXECUTE=yes] 'plib'; EXNAME='VGRAPH'; SOURCE=VGRA : ##VGRA
EXAMPLE     [EXECUTE=yes] 'plib'; EXNAME='VMCOMPARISON'; SOURCE=VMCO : ##VMCO
VMCOM       [PRINT=plot] Diet
STOP
```

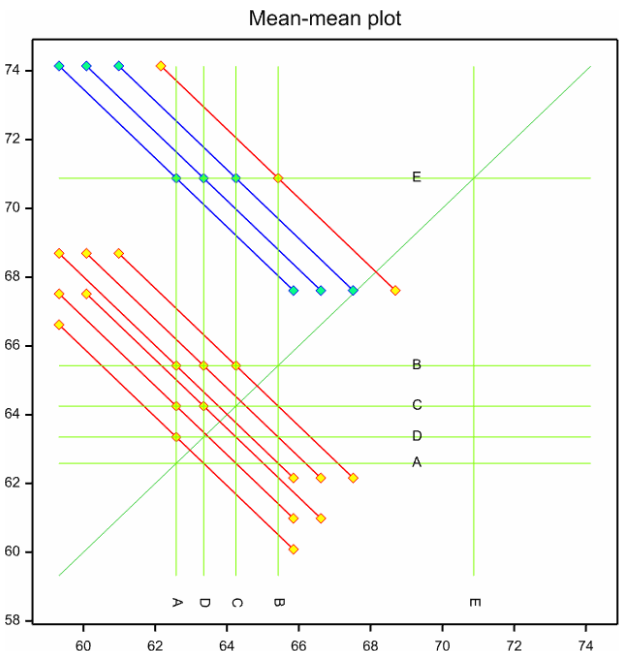


Example of F2DRESIDUALVARIOGRAM



Example of VDEFFECTS

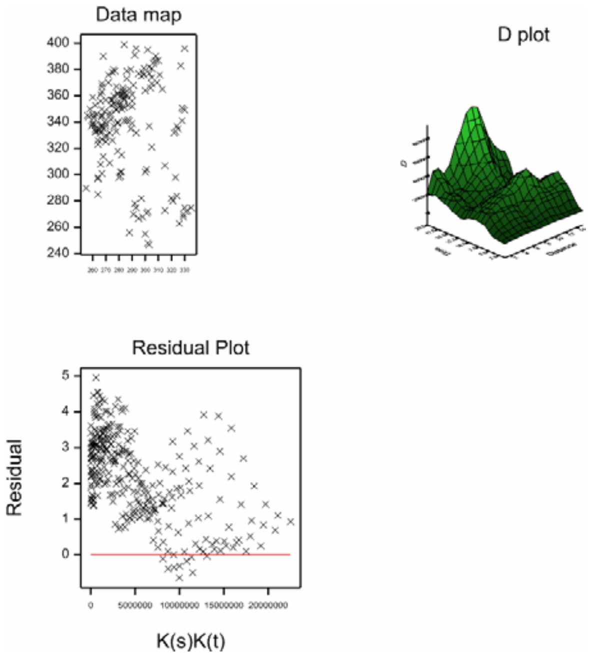




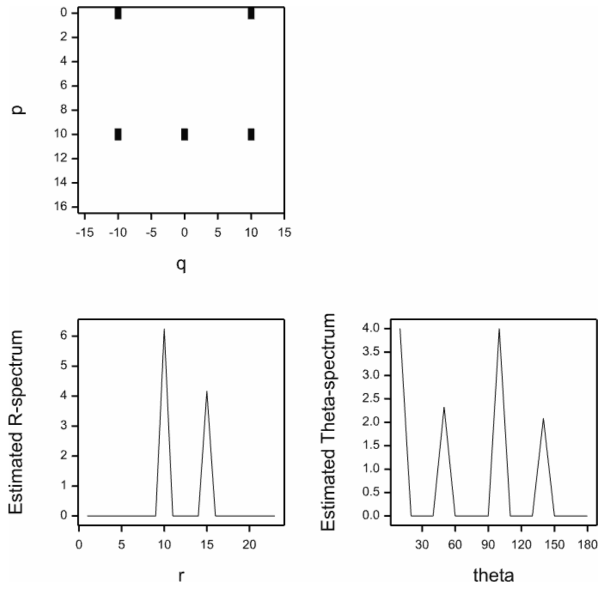
Example of VGRAPH

Example of VMCOMPARISON

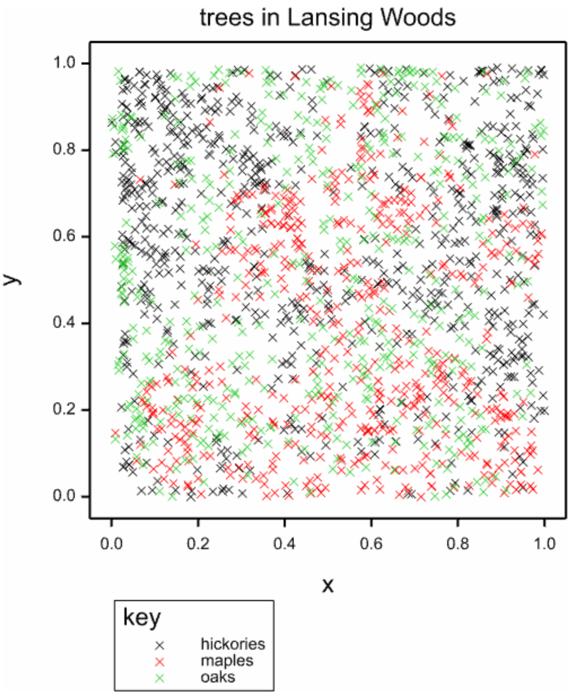
```
JOB          'Example 65
             4 examples of SPATIALSTATISTICS Procedures (see also Appendix 2)
     You can also run these examples with Help → Examples → Commands...
     and then choose the procedure or directive'
EXAMPLE      [EXECUTE=yes] 'plib'; EXNAME='DKSTPLOT'; SOURCE=DKST : ##DKST
EXAMPLE      [EXECUTE=yes] 'plib'; EXNAME='DPSPECTRALPLOT'; SOURCE=DPSP : ##DPSP
EXAMPLE      [EXECUTE=yes] 'plib'; EXNAME='DPTMAP'; SOURCE=DPTM : ##DPTM
EXAMPLE      [EXECUTE=yes] 'plib'; EXNAME='KSTMCTEST'; SOURCE=KSTM : ##KSTM
STOP
```



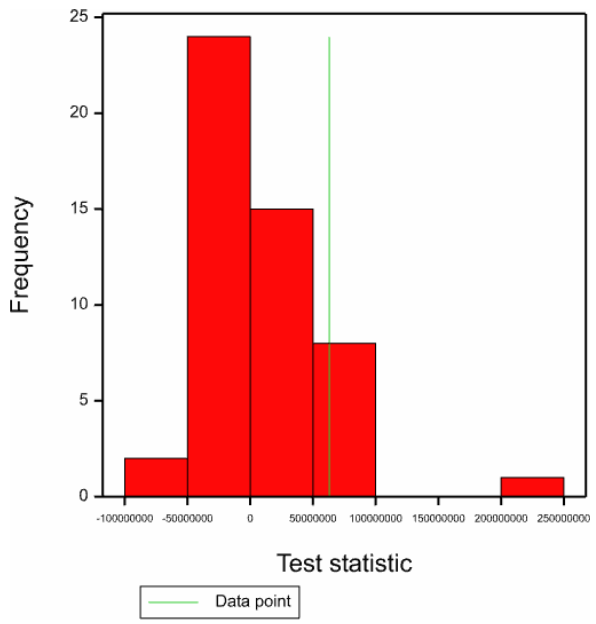
Example of DKSTPLOT



Example of DSPECTRALPLOT

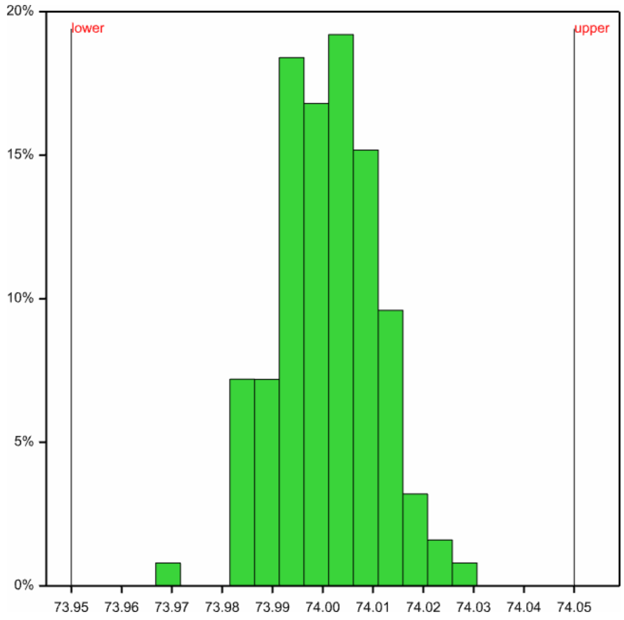


Example of DPTMAP

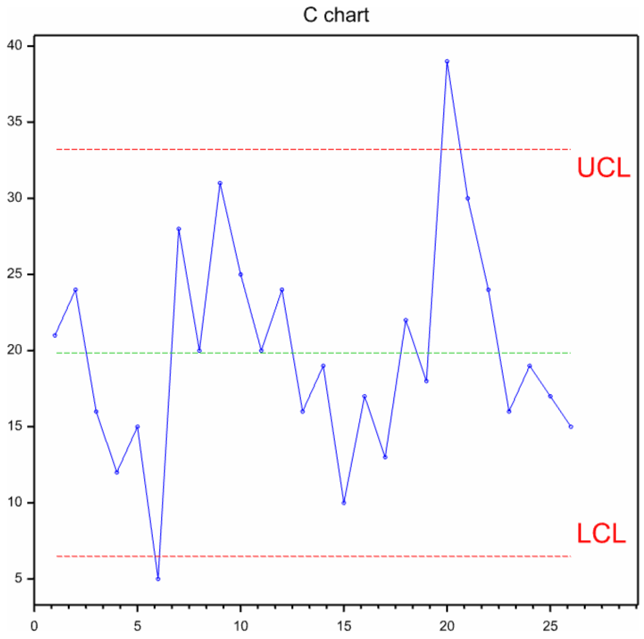


Example of KSTMCTEST

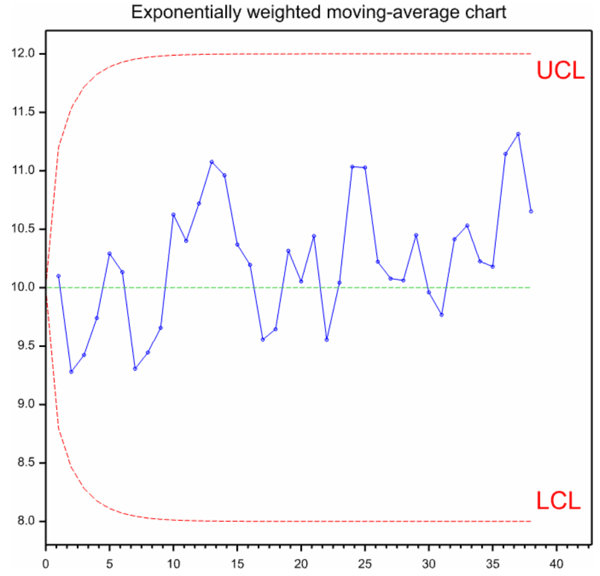
```
JOB          'Example 66
             4 examples of SPC Procedures (see also Appendix 2)
     You can also run these examples with Help → Examples → Commands...
     and then choose the procedure or directive'
EXAMPLE      [EXECUTE=yes] 'plib'; EXNAME='SPCAPABILITY'; SOURCE=SPCA : ##SPCA
SPCAPABILITY [PRINT=hist] Diameter; SAMPLES=5; LOWERLIMIT=73.95; \
             UPPERLIMIT=74.05
EXAMPLE      [EXECUTE=yes] 'plib'; EXNAME='SPCCHART'; SOURCE=SPCC : ##SPCC
EXAMPLE      [EXECUTE=yes] 'plib'; EXNAME='SPEWMA'; SOURCE=SPEW : ##SPEW
EXAMPLE      [EXECUTE=yes] 'plib'; EXNAME='SPSHEWART'; SOURCE=SPSH : ##SPSH
STOP
```



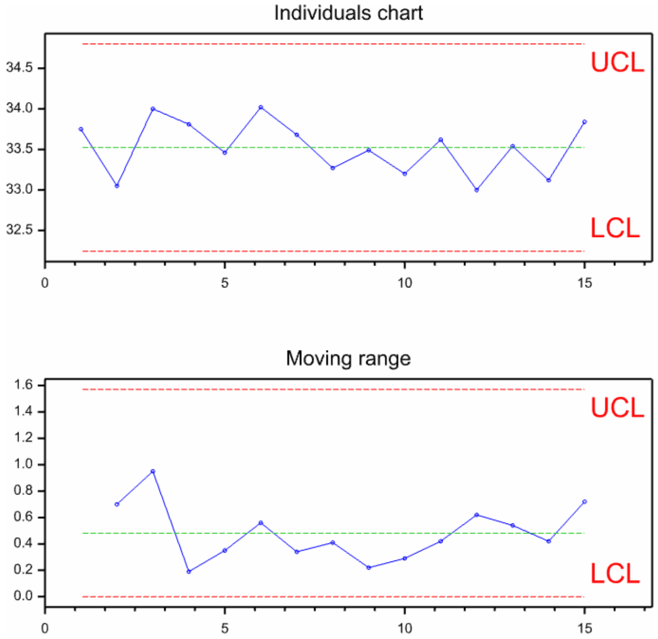
Example of SPCAPABILITY



Example of SPCCHART



Example of SPEWMA



Example of SPSHEWART

APPENDIX 1: DEFAULT SETTINGS OF FRAME, AXES AND PEN

DHELP possible, current

Possible graphical settings

FRAME: Window 1 ... 256

AXES: Window 1 ... 256

Pentitle	1 ... 256
Penaxes	1 ... 256
Pengrid	1 ... 256

PEN: Number 1 ... 256

Colour	0 ... 16777216, or -1 for 'background'
Linestyle	0 ... 32
Symbol	0 ... 22 or symbol name (see PEN)
Brush	1 ... 16
Font	1 ... 25

Font	Mapping
1	Default font (see Tools|Options|Fonts in Graphics Viewer)
2	Arial
3	Arial Black
4	Arial Narrow
5	Book Antiqua
6	Bookman Old Style
7	Century Gothic
8	Comic Sans MS
9	Courier New
10	Franklin Gothic Medi
11	Garamond
12	Georgia
13	Haettenschweiler
14	Impact
15	Lucida Console
16	Lucida Sans Unicode
17	Microsoft Sans Serif
18	Monotype Corsiva
19	Palatino Linotype
20	Sylfaen
21	Tahoma
22	Times New Roman
23	Trebuchet MS
24	Verdana
25	Arial

DEVICE:

*	1	Graphics Window
	2	PostScript
	3	Encapsulated PostScript
	4	HPGL
	5	Encapsulated PostScript File (*.eps)
	6	Windows Enhanced Metafile (*.emf)
	7	JPEG Files (*.jpg, *.jpeg)
	8	TIFF Files (*.tif, *.tiff)
	9	Portable Network Graphics Files (*.png)
	10	GenStat Metafile (*.gmf)
	11	Windows Bitmap File (*.bmp)
	12	Windows Default Printer

* marks the current device

Current graphical settings

FRAME directive

Window	Ylower	Yupper	Xlower	Xupper	Ymlower	Ymupper	Xmlower	Xmupper	PenTitle	PenKey	PenGrid
1	0.250	1.000	0.000	0.750	0.100	0.070	0.120	0.050	-5	-6	-4
2	0.000	0.250	0.150	0.670	0.100	0.070	0.120	0.050	-5	-6	-4
3	0.000	1.000	0.000	1.000	0.100	0.070	0.120	0.050	-5	-6	-4
4	0.000	1.000	0.000	1.000	0.100	0.070	0.120	0.050	-5	-6	-4
5	0.500	1.000	0.000	0.500	0.100	0.070	0.120	0.050	-5	-6	-4
6	0.500	1.000	0.500	1.000	0.100	0.070	0.120	0.050	-5	-6	-4
7	0.000	0.500	0.000	0.500	0.100	0.070	0.120	0.050	-5	-6	-4
8	0.000	0.500	0.500	1.000	0.100	0.070	0.120	0.050	-5	-6	-4
9	0.750	1.000	0.000	1.000	0.100	0.070	0.120	0.050	-5	-6	-4
10	0.500	0.750	0.000	1.000	0.100	0.070	0.120	0.050	-5	-6	-4
11	0.250	0.500	0.000	1.000	0.100	0.070	0.120	0.050	-5	-6	-4
12	0.000	0.250	0.000	1.000	0.100	0.070	0.120	0.050	-5	-6	-4
...											
256	0.000	1.000	0.000	1.000	0.100	0.070	0.120	0.050	-5	-6	-4

XAXIS, YAXIS, ZAXIS directives

Window	Xlower	Xupper	Ylower	Yupper	Zlower	Zupper
1	*	*	*	*	*	*
...						
256	*	*	*	*	*	*

Window	Xtitle	Ytitle	Ztitle	Xlabels	Ylabels	Zlabels
1	unset	unset	unset	unset	unset	unset
...						
256	unset	unset	unset	unset	unset	unset

PEN directive (For Colour Names see: 4.9.9.1)

	Colour	Linestyle	Method	Join	Brush	Font	Thickness	Size
1	0	*	point	ascending	*	1	1	1
2	16711680	*	point	ascending	*	1	1	1
3	3329330	*	point	ascending	*	1	1	1
4	255	*	point	ascending	*	1	1	1
5	65535	*	point	ascending	*	1	1	1
6	16711935	*	point	ascending	*	1	1	1
7	16776960	*	point	ascending	*	1	1	1
8	16747520	*	point	ascending	*	1	1	1
9	8388352	*	point	ascending	*	1	1	1
10	65407	*	point	ascending	*	1	1	1
11	2003199	*	point	ascending	*	1	1	1
12	9055202	*	point	ascending	*	1	1	1
13	16716947	*	point	ascending	*	1	1	1
14	6908265	*	point	ascending	*	1	1	1
15	11119017	*	point	ascending	*	1	1	1
16	13458524	*	point	ascending	*	1	1	1
17	10145074	*	point	ascending	*	1	1	1
18	3329330	*	point	ascending	*	1	1	1
19	4251856	*	point	ascending	*	1	1	1
20	8087790	*	point	ascending	*	1	1	1
21	12211667	*	point	ascending	*	1	1	1
22	9109504	*	point	ascending	*	1	1	1
23	32768	*	point	ascending	*	1	1	1
24	128	*	point	ascending	*	1	1	1
25	16416882	*	point	ascending	*	1	1	1
26	10025880	*	point	ascending	*	1	1	1
27	8087790	*	point	ascending	*	1	1	1
28	3100495	*	point	ascending	*	1	1	1
29	6908265	*	point	ascending	*	1	1	1
30	8421504	*	point	ascending	*	1	1	1
31	14474460	*	point	ascending	*	1	1	1
32	16777215	*	point	ascending	*	1	1	1
33	0	*	point	ascending	*	1	1	1
...								
256	0	*	point	ascending	*	1	1	1

Pens used for initial defaults

	Colour	Linestyle	Method	Join	Brush	Font	Thickness	Size	Purpose
-1	0	1	point	ascending	*	1	1	1.5	Axis Title Pen
-2	0	1	line	ascending	*	1	2	1	Axis Pen
-3	0	1	point	ascending	*	1	1	1	Axis Labels Pen
-4	11119017	2	line	ascending	*	1	1	1	Grid Pen
-5	0	1	point	ascending	*	1	1	1.5	Title Pen
-6	0	1	point	ascending	*	1	1	1	Key Pen
-7	0	1	point	ascending	*	1	1	1	DSHADE grid
-8	0	1	point	ascending	*	1	1	1	Histogram Outlines
-9	0	1	point	ascending	*	1	1	1	Barchart Outlines
-10	0	1	point	ascending	*	1	1	1	Pie Outlines
-11	0	1	line	ascending	*	1	1	1	Error Bars

For further details of current settings, you can use the SAVE parameter of the corresponding FRAME, XAXIS, YAXIS, ZAXIS, or PEN directive

Tip

The purpose of the negative pen numbers, the default numbers of specific pen related options and parameters of different directives, is further explained in the next table.

Directive	Option/Parameter	Default PEN	What to draw
XAXIS	Parameter PENTITLE	-1	Title of x-axis
XAXIS	Parameter PENAXIS	-2	x-axis
XAXIS	Parameter PENLABELS	-3	Labels of x-axis
YAXIS	Parameter PENTITLE	-1	Title of y-axis
YAXIS	Parameter PENAXIS	-2	y-axis
YAXIS	Parameter PENLABELS	-3	Labels of y-axis
ZAXIS	Parameter PENTITLE	-1	Title of z-axis
ZAXIS	Parameter PENAXIS	-2	z-axis
ZAXIS	Parameter PENLABELS	-3	Labels of z-axis
FRAME	Parameter PENGRID	-4	Grid lines
FRAME	Parameter PENTITLE	-5	Title of plot
FRAME	Parameter PENKEY	-6	Description of key
DGRAPH	Parameter YBARPEN	-11	Error bars in y-direction
DGRAPH	Parameter XBARPEN	-11	Error bars in x-direction
DHISTOGRAM	Option PENOUTLINE	-8	Outline of bars or perimeter
BARCHART	Option PENOUTLINE	-9	Outline of bars or perimeter
BARCHART	Parameter PENERRORBARS	-11	Error bars
DSHADE	Option PENGRID	-7	Grid lines
DPIE	Option PENOUTLINE	-10	Outline of slices or perimeter
DSTART*	Option TITLE	(-12)	Overall title of the plots
All high-quality plot directives*	Option KEYDESCRIPTION	(-5)	Overall description of the key

*The difference between the blue and black coloured lines in the table is that the overall title and overall description can be modified only by modifying the settings of the options and parameters of respectively pen -12 and pen -5, while the black coloured options and parameters in column 2 can be set by other pen numbers of which the settings of options and parameters can be modified.

For example to give the title of a plot a red colour the command

```
PEN     -5 ; COLOUR='red'
```

is the same as

```
PEN     10 ; COLOUR='red'
FRAME   1 ; PENTITLE=10
```

However, the consequence of the first statement is that the colour of the overall description of the key is also red. But if a red colour for the title is specified in the second way the colour of the overall description of the key remains black while the default colour of pen -5 is black.

See also the graph in Example 22.

APPENDIX 2: GRAPHICAL PROCEDURES PER MODULE

Module AOV (Analysis Of Variance)

A2DISPLAY	provides further output following an analysis of variance by A2WAY.
A2PLOT	plots effects from two-level designs with robust s.e. estimates.
A2WAY	performs analysis of variance of a balanced or unbalanced design with up to two treatment factors.
ABIVARIATE	produces graphs and statistics for bivariate analysis of variance.
ADPOLYNOMIAL	plots single-factor polynomial contrasts fitted by ANOVA.
AFIELDRESIDUALS	display residuals in field layout.
AGRAPH	plots one-or two-way tables of means from ANOVA.
AMCOMPARISON	performs pairwise multiple comparison tests for ANOVA means.
AMMI	allows exploratory analysis of genotype*environment interactions.
AONEWAY	performs one-way analysis of variance.
AOVANYHOW	performs analysis of variance using ANOVA, regression or REML as appropriate.
AOVDISPLAY	provides further output from an analysis by AOVANYHOW.
APERMTEST	does random permutation tests for analysis-of-variance tables.
APLOT	plots residuals from an ANOVA analysis.
AUDISPLAY	produces further output for an unbalanced design (after AUNBALANCED).
AUGRAPH	plots tables of means from AUNBALANCED.
AUMCOMPARISON	performs pairwise multiple comparison tests for means from an unbalanced analysis of variance, performed previously by AUNBALANCED.
AUNBALANCED	performs analysis of variance for unbalanced designs.
CINTERACTION	clusters rows and columns of a two-way interaction table.
LSIPLOT	plots least significant intervals, saved from SEDLSI.
SEDLSI	calculates least significant intervals.

Module BASIC

A2DISPLAY	provides further output following an analysis of variance by A2WAY.
A2WAY	performs analysis of variance of a balanced or unbalanced design with up to two treatment factors.
AONEWAY	performs one-way analysis of variance.
BOXPLOT	draws box-and-whisker diagrams or schematic plots.
CHIPERMTEST	performs a random permutation test for a two-dimensional contingency table.
CONVEXHULL	finds the points of a single or a full peel of convex hulls.
DFUNCTION	plots a function.
DTABLE	plots tables.
RUGPLOT	draws "rugplots" to display the distribution of one or more samples.
TALLY	forms a simple tally table of the distinct values in a vector.

Module BAYES

BGPLOT	produces plots for output and diagnostics from MCMC simulations.

Module BIOASSAY

CUMDISTRIBUTION	fits frequency distributions to accumulated counts.

Module CIRCULARDATA

DCIRCULAR	plots circular data.
WINDROSE	plots rose diagrams of circular data like wind speeds.

Module DATAMINING

BCLASSIFICATION	constructs a classification tree.
BRDISPLAY	displays a regression key.

BREGRESSION	constructs a regression tree.
SOMESTIMATE	estimates the weights for self-organizing maps.

Module DESIGN

AFIELDRESIDUALS	display residuals in field layout.
AFNONLINEAR	forms D-optimal designs to estimate the parameters of a nonlinear or generalized linear model.
DDESIGN	plots the plan of an experimental design.

Module DISTRIBUTIONS

CUMDISTRIBUTION	fits frequency distributions to accumulated counts.
FDRBONFERRONI	estimates false discovery rates by a Bonferroni-type procedure.
FDRMIXTURE	estimates false discovery rates using mixture distributions.
GEV	fits a Generalized Extreme Value distribution to a variate.
GPARETO	fits a Generalized Pareto distribution to the observations in a variate above a given threshold.
GREJECTIONSAMPLE	generates random samples using rejection sampling.
KERNELDENSITY	uses kernel density estimation to estimate a sample density.
TALLY	forms a simple tally table of the distinct values in a vector.

Module ECOLOGY

ECABUNDANCEPLOT	produces rank/abundance,ABC and k-dominance plots.
ECACCUMULATION	plots species accumulation curves for samples or individuals.
ECANOSIM	performs an analysis of similarities (ANOSIM)
ECFIT	fits models to species abundance data
ECNICHE	generates relative abundance of species for niche-based models
ECRAREFACTION	calculates individual or sample-based rarefaction
LORENZ	plots the Lorenz curve and calculates the Gini and asymmetry coefficients.

Module ECONOMICS

LORENZ	plots the Lorenz curve and calculates the Gini and asymmetry coefficients.

Module GEOSTATISTICS

DCOVARIOGRAM	plots 2-dimensional auto- and cross-variograms.
DHSCATTERGRAM	plots an h-scattergram.
DVARIOGRAM	plots fitted models to an experimental variogram.
KCROSSVALIDATION	computes cross validation statistics for punctual kriging.
MVARIOGRAM	fits models to an experimental variogram.

Module GLM (Generalized Linear Model)

DORDINAL (Biometris)	plots and displays the results of a simple ordinal logistic regression model
HGGRAPH	draws a graph to display the fit of an HGLM or DHGLM analysis.
HGPLOT	produces model-checking plots for a hierarchical or double hierarchical generalized linear model.
RCHECK	checks the fit of a linear or generalized linear regression.
RDESTIMATES	plots one- or two-way tables of regression estimates.
RGRAPH	draws a graph to display the fit of a regression model.
RJOINT	does modified joint regression analysis for variety-by-environment data.
RQUADRATIC	fits a quadratic surface and estimates its stationary point.
RSURVIVAL	models survival times of exponential, Weibull, extreme-value, log-logistic or lognormal distributions.

Module GRAPHICS

A2PLOT	plots effects from two-level designs with robust s.e. estimates.
ABIVARIATE	produces graphs and statistics for bivariate analysis of variance.

ADPOLYNOMIAL	plots single-factor polynomial contrasts fitted by ANOVA.
AFIELDRESIDUALS	display residuals in field layout.
AKAIKEHISTOGRAM	prints histograms with improved definition of groups
APLOT	plots residuals from an ANOVA analysis.
BANK	calculates the optimum aspect ratio for a graph.
BGPLOT	produces plots for output and diagnostics from MCMC simulations.
BIPLOT	produces a biplot from a set of variates.
BOXPLOT	draws box-and-whisker diagrams or schematic plots.
CABIPLOT	plots results from correspondence analysis or multiple correspondence analysis.
CRBIPLOT	plots correlation or distance biplots after RDA, or ranking biplots after CCA.
CRTRIPLOT	plots ordination biplots or triplots after CCA or RDA.
CVAPLOT	plots the mean and unit scores from a canonical variates analysis.
DBARCHART	produces bar charts for one or two-way tables.
DBIPLOT	plots a biplot from an analysis by PCP, CVA or PCO.
DCIRCULAR	plots circular data.
DCOLOURS	forms a band of graduated colours for graphics.
DCOMPOSITIONAL	plots 3-part compositional data within a barycentric triangle.
DCORRELATION	plots a correlation matrix.
DCOVARIOGRAM	plots 2-dimensional auto- and cross-variograms.
DDENDROGRAM	draws dendrograms with control over structure and style.
DDESIGN	plots the plan of an experimental design.
DFOURIER	performs a harmonic analysis of a univariate time series.
DFRTEXT	adds text to a graphics frame.
DFUNCTION	plots a function.
DHELP	provides information about GenStat graphics (no graphical output).
DHSCATTERGRAM	plots an h-scattergram.
DKALMAN	plots results from an analysis by KALMAN.
DMADENSITY	plots the empirical CDF or PDF (kernel smoothed) by groups.
DMASS	plots discrete data like mass spectra, discrete probability functions.
DMSCATTER	produces a scatter-plot matrix for one or two sets of variables.
DMST	gives a high resolution plot of an ordination with minimum spanning tree.
DOTHISTOGRAM	plots dot histograms.
DOTPLOT	produces a dot-plot using line-printer or high-resolution graphics.
DPARALLEL	displays multivariate data using parallel coordinates.
DPOLYGON	draws polygons using high-resolution graphics.
DPROBABILITY	plots probability distributions, and estimates their parameters.
DPSPECTRALPLOT	calculates an estimate of the spectrum of a spatial point pattern.
DPTMAP	draws maps for spatial point patterns using high-resolution graphics.
DQMAP	displays a genetic map.
DQMKSCORES	plots a grid of marker scores for genotypes and indicates missing data.
DQMQTLSCAN	plots the results of a genome-wide scan for QTL effects in multi-environment trials.
DQRECOMBINATIONS	plots a matrix of recombination frequencies between markers.
DQSQTLSCAN	plots the results of a genome-wide scan for QTL effects in single-environment trials.
DREPMEASURES	plots profiles and differences of profiles for repeated measures data.
DRESIDUALS	plots residuals.
DSCATTER	produces a scatter-plot matrix using high-resolution graphics.
DTABLE	plots tables.
DTEXT	adds text to a graph.
DTIMEPLOT	produces horizontal bars displaying a continuous time record.
DVARIOGRAM	plots fitted models to an experimental variogram.
DXYGRAPH	draws two-dimensional graphs with marginal distribution plots alongside the y- and x-axes.

F2DRESIDUALVARIOGRAM	calculates and plots a 2-dimensional variogram from a 2-dimensional array of residuals.
FFRAME	forms multiple windows in a plot-matrix for high-resolution graphics (no graphical output).
GETRGB	gets the RGB values of the standard graphics colours.
GGEBIPLOT	plots displays to assess genotype+genotype-by-environment variation.
INSIDE	determines whether points lie within a specified polygon.
LORENZ	plots the Lorenz curve and calculates the Gini and asymmetry coefficients.
LSIPLOT	plots least significant intervals, saved from SEDLSI.
MA2CLUSTER	performs a two-way clustering of microarray data by probes (or genes) and slides.
MAHISTOGRAM	plots histograms of microarray data.
MAPCLUSTER	clusters probes or genes with microarray data.
MAPLOT	produces two-dimensional plots of microarray data.
MASCLUSTER	clusters microarray slides.
MASHADE	produces shade plots to display spatial variation of microarray data.
MAVOLCANO	produces volcano plots of microarray data.
RCHECK	checks the fit of a linear or generalized linear regression.
RUGPLOT	draws "rugplots" to display the distribution of one or more samples.
SETDEVICE	opens a graphical file and specifies the device number on basis of its extension.
TRELLIS	does a trellis plot.
VPLOT	plots residuals from a REML analysis.
WINDROSE	plots rose diagrams of circular data like wind speeds.

Module INTERACTIONS

CINTERACTION	clusters rows and columns of a two-way interaction table.
MEDIANTETRAD	gives robust identification of multiple outliers in 2-way tables.
KSTMCTEST	performs a Monte-Carlo test for space-time interaction.

Module MANIPULATION

ALIGNCURVE	forms an optimal warping to align an observed series of observations with a standard series.
BASELINE	estimates a baseline for a series of numbers whose minimum value is drifting.
GETRGB	gets the RGB values of the standard graphics colours (no plots).
GREJECTIONSAMPLE	generates random samples using rejection sampling.
MIN1DIMENSION	finds the minimum of a function in one dimension.
PEAKFINDER	finds the locations of peaks in an observed series.
QNORMALIZE	performs quantile normalization.
RLASSO	performs lasso using iteratively reweighted least-squares.
YTRANSFORM	estimates the parameter lambda of a single parameter transformation.

Module METAANALYSIS

META	combines estimates from individual trials.

Module MICROARRAYS

DMADENSITY	plots the empirical CDF or PDF (kernel smoothed) by groups.
FDRBONFERRONI	estimates false discovery rates by a Bonferroni-type procedure.
FDRMIXTURE	estimates false discovery rates using mixture distributions.
MA2CLUSTER	performs a two-way clustering of microarray data by probes (or genes) and slides.
MAEBAYES	modifies t-values by an empirical Bayes method.
MAHISTOGRAM	plots histograms of microarray data.
MAPCLUSTER	clusters probes or genes with microarray data.
MAPLOT	produces two-dimensional plots of microarray data.

MASCLUSTER	clusters microarray slides.
MASHADE	produces shade plots to display spatial variation of microarray data.
MAVOLCANO	produces volcano plots of microarray data.
MNORMALIZE	normalizes two-colour microarray data.
QNORMALIZE	performs quantile normalization.

Module MVA (MultiVariate Analysis)

AMMI	allows exploratory analysis of genotype × environment interactions.
BCDISPLAY	displays a classification tree.
BCFDISPLAY	displays information about a random classification forest.
BCLASSIFICATION	constructs a classification tree.
BGRAPH	plots a tree.
BIPLOT	produces a biplot from a set of variates.
BKDISPLAY	displays an identification key.
BKEY	constructs an identification key.
BPRUNE	prunes a tree using minimal cost complexity.
BRDISPLAY	displays a regression key.
BREGRESSION	constructs a regression tree.
CABIPLOT	plots results from correspondence analysis or multiple correspondence analysis.
CONVEXHULL	finds the points of a single or a full peel of convex hulls.
CRBIPLOT	plots correlation or distance biplots after RDA, or ranking biplots after CCA.
CRTRIPLOT	plots ordination biplots or triplots after CCA or RDA.
CVAPLOT	plots the mean and unit scores from a canonical variates analysis.
CVASCORES	calculates scores for individual units in canonical variates analysis.
DBBIPLOT (Biometris)	produces a high-resolution graphical biplot.
DBIPLOT	plots a biplot from an analysis by PCP, CVA or PCO.
DDENDROGRAM	draws dendrograms with control over structure and style.
DISCRIMINATE	performs discriminant analysis.
DMST	gives a high resolution plot of an ordination with minimum spanning tree.
DPARALLEL	displays multivariate data using parallel coordinates.
GENPROCRUSTES	performs a generalized Procrustes analysis.
LRVSCREE	prints a scree diagram and/or a difference table of latent roots.
MCORANALYSIS	does multiple correspondence analysis.
RIDGE	produces ridge regression and principal component regression analyses.
SAGRAPES	produces statistics and graphs for checking sensory panel performance.
SDISCRIMINATE	selects the best set of variates to discriminate between groups.

Module NONPARAMETRIC

KAPLANMEIER	calculates the Kaplan-Meier estimate of the survivor function.

Module OPTIMIZATION

MIN1DIMENSION	finds the minimum of a function in one dimension.

Module OUTLIERS

MEDIANTETRAD	gives robust identification of multiple outliers in 2-way tables.

Module Quality Control

OCATTRIBUTES (Biometris)

Calculates operating characteristic curves for single and multiple acceptance sampling plans for attributes.

Module QTL

DQMAP	displays a genetic map.
DQMKSCORES	plots a grid of marker scores for genotypes and indicates missing data.

DQMQTLSCAN	plots the results of a genome-wide scan for QTL effects in multi-environment trials.
DQRECOMBINATIONS	plots a matrix of recombination frequencies between markers.
DQSQTLSCAN	plots the results of a genome-wide scan for QTL effects in single-environment trials.
QEIGENANALYSIS	uses principal components analysis and the Tracy-Widom statistic to find the number of significant principal components to represent a set of variables.
QLDDECAY	estimates linkage disequilibrium (LD) decay along a chromosome.
QMAP	constructs genetic linkage maps using marker data from experimental populations.
QMASSOCIATION	performs multi-environment marker-trait association analysis in a genetically diverse population using bi-allelic and multi-allelic markers.
QMKDIAGNOSTICS	generates descriptive statistics and diagnostic plots of molecular marker data.
QMQTLSCAN	performs a genome-wide scan for QTL effects (Simple and Composite Interval Mapping) in multi-environment trials or multiple populations.
QRECOMBINATIONS	calculates the expected numbers of recombinations and the recombination frequencies between markers.
QSASSOCIATION	performs marker-trait association analysis in a genetically diverse population using bi-allelic and multi-allelic markers.
QSQTLSCAN	performs a genome-wide scan for QTL effects (Simple and Composite Interval Mapping) in single-environment trials.

Module REGRESSION

AOVANYHOW	performs analysis of variance using ANOVA, regression or REML as appropriate.
AOVDISPLAY	provides further output from an analysis by AOVANYHOW.
HGGRAPH	draws a graph to display the fit of an HGLM or DHGLM analysis.
HGPLOT	produces model-checking plots for a hierarchical or double hierarchical generalized linear model.
LRIDGE	does logistic ridge regression.
LSIPLOT	plots least significant intervals, saved from SEDLSI.
R2LINES	fits two-straight-line (broken-stick) models to data.
RCHECK	checks the fit of a linear or generalized linear regression.
RDESTIMATES	plots one- or two-way tables of regression estimates.
RGRAPH	draws a graph to display the fit of a regression model.
RIDGE	produces ridge regression and principal component regression analyses.
RJOINT	does modified joint regression analysis for variety-by-environment data.
RLASSO	performs lasso using iteratively reweighted least-squares.
RLFUNCTIONAL	fits a linear functional relationship model.
RQLINEAR	fits and plots quantile regressions for linear models.
RQNONLINEAR	fits and plots quantile regressions for nonlinear models.
RQSMOOTH	fits and plots quantile regressions for loess or spline models.
RQUADRATIC	fits a quadratic surface and estimates its stationary point.
RUNCERTAINTY (Biometris)	calculates contributions of model inputs to the variance of a model output.
SEDLSI	calculates least significant intervals.

Module REML Restricted Maximum Likelihood (Unbalanced ANOVA)

AOVANYHOW	performs analysis of variance using ANOVA, regression or REML as appropriate.
AOVDISPLAY	provides further output from an analysis by AOVANYHOW.
F2DRESIDUALVARIOGRAM	calculates and plots a 2-dimensional variogram from a 2-dimensional array of residuals.
VDEFFECTS	plots one- or two-way tables of effects estimated in a REML analysis.
VGRAPH	plots tables of means from REML.

VMCOMPARISON	performs pairwise comparisons between REML means.
VPLOT	plots residuals from a REML analysis.

Module REPEATEDMEASURES

DREPMEASURES	plots profiles and differences of profiles for repeated measures data.
DTIMEPLOT	produces horizontal bars displaying a continuous time record.

Module RESAMPLE

BOOTSTRAP	produces bootstrapped estimates, standard errors and distributions.
APERMTEST	does random permutation tests for analysis-of-variance tables.

Module SAMPLESIZE

TPOWER (Biometris)	calculates the power of Student's t-test and plots power curves.

Module SENSORYANALYSIS

SAGRAPES	produces statistics and graphs for checking sensory panel performance.

Module SPATIALSTATISTICS

DHSCATTERGRAM	plots an h-scattergram.
DKSTPLOT	produces diagnostic plots for space-time clustering.
DPOLYGON	draws polygons using high-resolution graphics.
DPSPECTRALPLOT	calculates an estimate of the spectrum of a spatial point pattern.
DPTMAP	draws maps for spatial point patterns using high-resolution graphics.
KCROSSVALIDATION	computes cross validation statistics for punctual kriging.
KSTMCTEST	performs a Monte-Carlo test for space-time interaction.
MVARIOGRAM	fits models to an experimental variogram.
PTDESCRIBE	gives summary and second order statistics for a point process.

Module SPC (Statistical Process Control)

SPCAPABILITY	calculates capability statistics.
SPCCHART	plots c or u charts representing numbers of defective items.
SPEWMA	plots exponentially weighted moving-average control charts.
SPPCHART	plots p or np charts for binomial testing for defective items.
SPSHEWHART	plots control charts for mean and standard deviation or range.

Module SURVEY

SVCALIBRATE	performs generalized calibration of survey data.
SVHOTDECK	performs hot-deck and model-based imputation for survey data.
SVSTRATIFIED	analyses stratified random surveys by expansion or ratio raising.
SVTABULATE	tabulates data from random surveys, including multistage surveys and surveys with unequal probabilities of selection.
SVWEIGHT	forms survey weights.

Module SURVIVAL

KAPLANMEIER	calculates the Kaplan-Meier estimate of the survivor function.
RLIFETABLE	calculates the life-table estimate of the survivor function.
RSURVIVAL	models survival times of exponential, Weibull, extreme-value, log-logistic or lognormal distributions.

Module TABULATION (Survey analysis)

DBARCHART	produces barcharts for one or two-way tables.
DTABLE	plots tables.
SVCALIBRATE	performs generalized calibration of survey data.

SVHOTDECK	performs hot-deck and model-based imputation for survey data.
SVSTRATIFIED	analyses stratified random surveys by expansion or ratio raising.
SVTABULATE	tabulates data from random surveys, including multistage surveys and surveys with unequal probabilities of selection.
SVWEIGHT	forms survey weights.

Module TIMESERIES

BJESTIMATE	fits an ARIMA model, with forecast and residual checks.
BJFORECAST	plots forecasts of a time series using a previously fitted ARIMA.
BJIDENTIFY	displays time series statistics useful for ARIMA model selection.
DFOURIER	performs a harmonic analysis of a univariate time series.
DKALMAN	plots results from an analysis by KALMAN.
MCROSSSPECTRUM	performs a spectral analysis of a multiple time series.
MOVINGAVERAGE	calculates and plots the moving average of a time series.
PTDESCRIBE	gives summary and second order statistics for a point process.
REPPERIODOGRAM	gives periodogram-based analyses for replicated time series.
SMOOTHSPECTRUM	forms smoothed spectrum estimates for univariate time series.
TSUMMARIZE (directive)	displays characteristics of time series models.

Module TREES

BCDISPLAY	displays a classification tree.
BCLASSIFICATION	constructs a classification tree.
BGRAPH	plots a tree.
BKDISPLAY	displays an identification key.
BKEY	constructs an identification key.
BPRUNE	prunes a tree using minimal cost complexity.
BRDISPLAY	displays a regression key.
BREGRESSION	constructs a regression tree.

APPENDIX 3: TYPESETTING WITHIN STRINGS

Strings can contain typesetting commands to represent Greek and mathematical symbols, see next page. The commands are converted automatically by GenStat to match the style of output (HTML, LaTeX, plain-text or RTF). They are all introduced by the character tilde (~).

The style of font can be changed to **bold or italic**.

~bold or ~b	introduces a sequence of bold characters; these must be placed within curly brackets and any spaces between ~bold and the opening curly bracket are ignored e.g. ~bold {Please note:}
~italic or ~i	introduces a sequence of italic characters; these must be placed within curly brackets and any spaces between ~italic and the opening curly bracket are ignored e.g. ~italic {Passer domesticus}

You can also produce output in **the same style as GenStat** uses when it echoes commands in the output.

~genstat or ~g	introduces some output in the style that GenStat uses to echo commands; it must be placed within curly brackets and any spaces between ~genstat and the opening curly bracket are ignored.

You can define **subscripts and superscripts** (for example to define equations).

~_	introduces a subscript; if the subscript is a single character it can be placed immediately after _, otherwise it must be placed within curly brackets; any spaces between ~_ and the opening curly bracket are ignored.
~^	introduces a superscript; if the superscript is a single character it can be placed immediately after ^, otherwise it must be placed within curly brackets; any spaces between ~^ and the opening curly bracket are ignored.

You can use **special characters in subscripts or superscripts**, but fonts must be specified outside the subscript or superscript. For example:

~i {x~_{i,j}}	defines $x_{i,j}$
x~^ {2n}	defines x^{2n}
~i{x~_{i,j}}~^2	defines $x_{i,j}{}^{2}$
~b{X}~i{~_{i,j}}~^2	defines $\mathbf{X}_{i,j}{}^{2}$

For additional flexibility, you can specify output information in either HTML, LaTeX or RTF. This will be inserted only into output constructed by GenStat in the same style. You can also supply information to be included only in plain-text output (which may, for example, be your translation of the HTML, LaTeX or RTF information).

~html or ~h	introduces a sequence of information in HTML; the information must be placed within curly brackets and any spaces between ~html and the opening curly bracket are ignored.
~latex or ~l	introduces a sequence of information in LaTeX; the information must be placed within curly brackets and any spaces between ~latex and the opening curly bracket are ignored. The information may itself contain curly brackets. These are assumed to be paired according to the usual rules of LaTeX, except that any curly brackets preceded by the LaTeX escape character \ are ignored.
~plain or ~p	introduces a sequence of information to be inserted only in plain-text output; the information must be placed within curly brackets and any spaces between ~plain and the opening curly bracket are ignored.
~rtf or ~r	introduces a sequence of information in RTF; the output must be placed within curly brackets and any spaces between ~rtf and the opening curly bracket are ignored. The information may itself contain curly brackets. These are assumed to be paired according to the usual rules of RTF, except that curly brackets preceded by the RTF escape character \ are ignored.

The following commands define Greek characters and various special and mathematical symbols. **Greek characters in capital letters** can be obtained by beginning the name of the character with a capital letter, for example ~{Sigma}; subsequent capital letters are irrelevant. See the right column on the next page.

The character definitions (within the curly brackets) can be abbreviated. GenStat checks through the possibilities, in the order defined below, until it finds the first match.

~	~{~}	tilde symbol; also see ~{tilde}		
α	~{alpha}	Greek character alpha	Α	~{Alpha}
β	~{beta}	Greek character beta	Β	~{Beta}
γ	~{gamma}	Greek character gamma	Γ	~{Gamma}
δ	~{delta}	Greek character delta	Δ	~{Delta}
ε	~{epsilon}	Greek character epsilon	Ε	~{Epsilon}
ε	~{varepsilon}	Greek character epsilon (variant)	Ε	~{Varepsilon}
ζ	~{zeta}	Greek character zeta	Ζ	~{Zeta}
η	~{eta}	Greek character eta	Η	~{Eta}
θ	~{theta}	Greek character theta	Θ	~{Theta}
ϑ	~{vartheta}	Greek character theta (variant)	Θ	~{Vartheta}
ι	~{iota}	Greek character iota	Ι	~{Iota}
κ	~{kappa}	Greek character kappa	Κ	~{Kappa}
λ	~{lambda}	Greek character lambda	Λ	~{Lambda}
μ	~{mu}	Greek character mu	Μ	~{Mu}
ν	~{nu}	Greek character nu	Ν	~{Nu}
ξ	~{xi}	Greek character xi	Ξ	~{Xi}
ο	~{omicron}	Greek character omicron	Ο	~{Omicron}
π	~{pi}	Greek character pi	Π	~{Pi}
ϖ	~{varpi}	Greek character pi (variant)	Π	~{Varpi}
ρ	~{rho}	Greek character rho	Ρ	~{Rho}
ρ	~{varrho}	Greek character rho (variant)	Ρ	~{Varrho}
σ	~{sigma}	Greek character sigma	Σ	~{Sigma}
ς	~{varsigma}	Greek character sigma (terminal version)	Σ	~{Varsigma}
τ	~{tau}	Greek character tau	Τ	~{Tau}
υ	~{upsilon}	Greek character upsilon	Υ	~{Upsilon}
φ	~{phi}	Greek character phi	Φ	~{Phi}
φ	~{varphi}	Greek character phi (variant)	Φ	~{Varphi}
χ	~{chi}	Greek character chi	Χ	~{Chi}
ψ	~{psi}	Greek character psi	Ψ	~{Psi}
ω	~{omega}	Greek character omega	Ω	~{Omega}
•	~{bull} or ~{bullet}	bullet		
·	~{cdot}	decimal point; also see ~{middot}		
÷	~{div} or ~{divide}	divide symbol		
>>	~{gg}	">>" symbol; also see ~{raquo}		
<<	~{laquo}	"<<" symbol; also see ~{ll}		
<<	~{ll}	"<<" symbol; also see ~{laquo}		
·	~{middot}	alternative way of specifying a decimal point; also see ~{cdot}		
-	~{minus}	minus symbol		
±	~{plusminus}	"+ or minus" symbol; also see ~{pm}		
±	~{pm}	"+ or minus" symbol; also see ~{plusminus}		
>>	~{raquo}	">>" symbol; also see ~{gg}		
√	~{sqrt}	square-root symbol		

⊕	~{oplus}	+ within circle
⊖	~{ominus}	minus symbol within circle. **At this moment round brackets instead of circle.**
⊗	~{otimes}	multiply symbol within circle
Ø	~{oslash}	slash symbol within circle
⊙	~{odot}	dot within circle. **At this moment round brackets instead of circle.**
~	~{tilde}	tilde symbol; also see ~{~}
×	~{times}	multiply symbol
	~{break}	**starts a new line in captions**

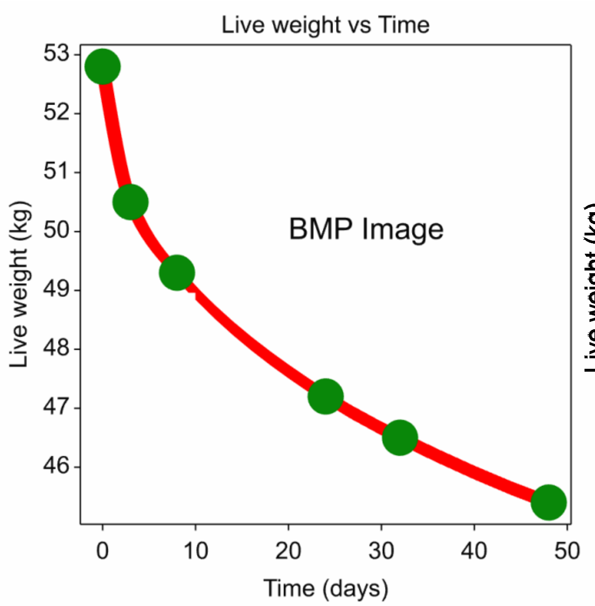
Some other symbols can be inserted into text by using extended ASCII codes. **For these symbols no tilde is needed.** Extended ASCII codes can be inserted by holding the Alt button pressed and then typing 3 digits on the right of your keyboard. In the figure below an overview is given of the possibilities. The dot in for example Alt13. should be replaced by one of the digits 0 ... 9. For example Alt130 gives an é and Alt211 an Ë.

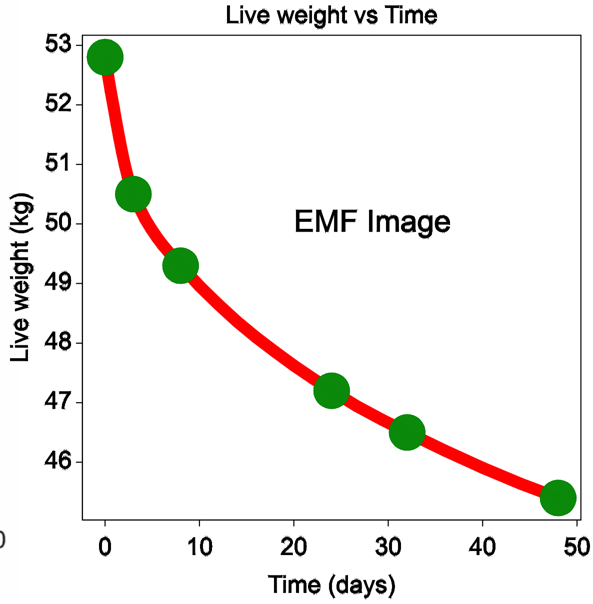
	0	1	2	3	4	5	6	7	8	
Alt12.									Ç	ü
Alt13.	é	â	ä	à	å	ç	ê	ë	è	ï
Alt14.	î	ì	Ä	Å	É	æ	Æ	ô	ö	ò
Alt15.	û	ù	ÿ	Ö	Ü	ø	£	Ø	×	ƒ
Alt16.	á	í	ó	ú	ñ	Ñ	ª	º	¿	®
Alt17.	¬	½	¼	¡	«	»	¦	¦	¦	¦
Alt18.	¦	Á	Â	À	©	¦	¦	+	+	¢
Alt19.	¥	+	+	-	-	+	-	+	ã	Ã
Alt20.	+	+	-	-	¦	-	+	¤	ð	Ð
Alt21.	Ê	Ë	È	i	Í	Î	Ï	+	+	¦
Alt22.	_	¦	Ì	¯	Ó	ß	Ô	Ò	õ	Õ
Alt23.	µ	þ	Þ	Ú	Û	Ù	ý	Ý	¯	´
Alt24.	-	±	=	¾	¶	§	÷	¸	°	¨
Alt25.	·	¹	³	²	¦					

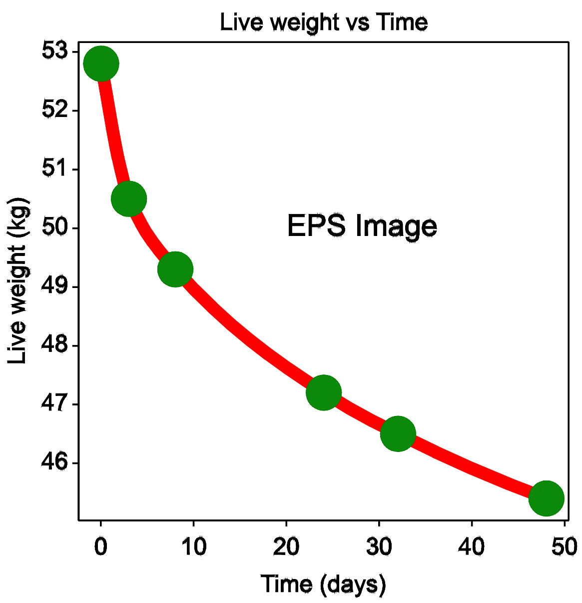
APPENDIX 4: GRAPH IN WORD 2010

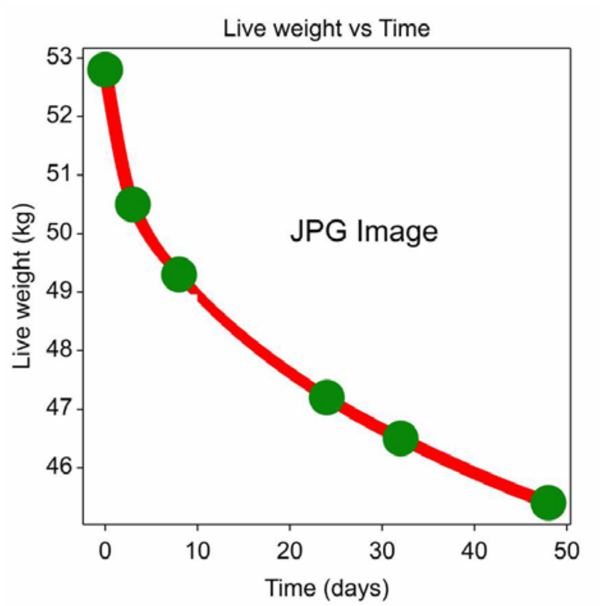
To insert saved images into Word, use the Insert →Picture menu in Word and select the image you want to insert. If you have used `DEVICE 1` the image is displayed on the screen. In that case you can right click on the image and select Copy. With Paste you can insert the image in Word then.

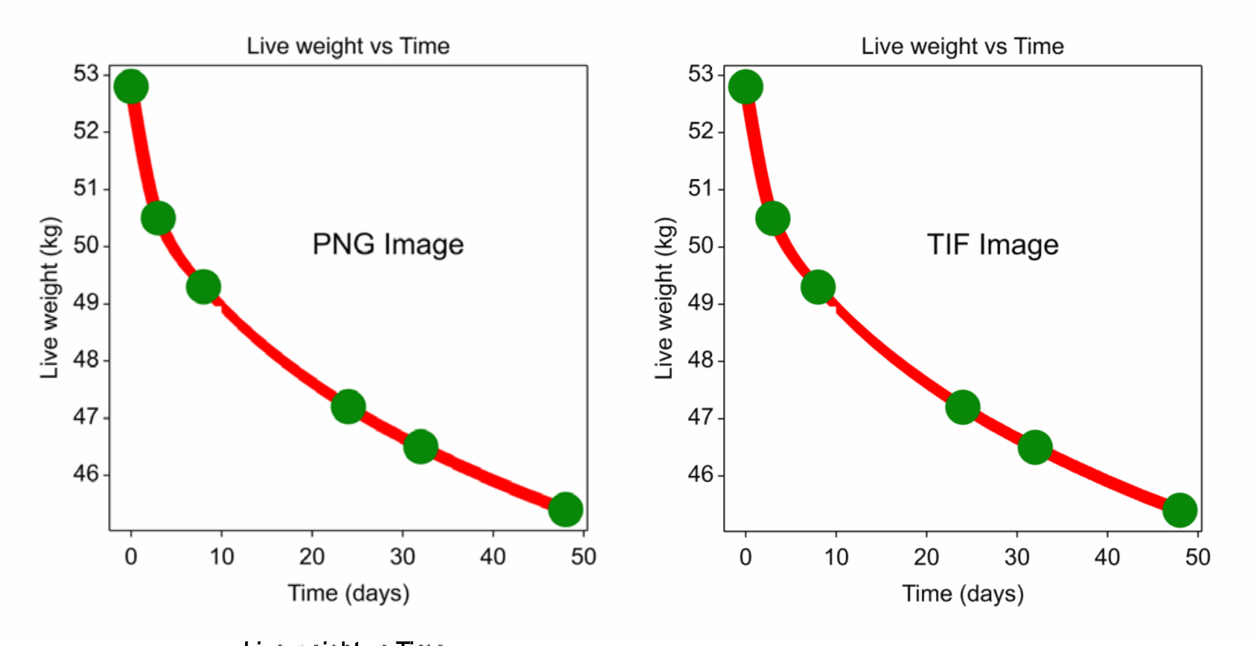
The image will be positioned in Word at the cursor position when you gave the Insert command. The GMF file format cannot be inserted in Word. Here are 6 different formats inserted into this document plus the Copy and Paste method.

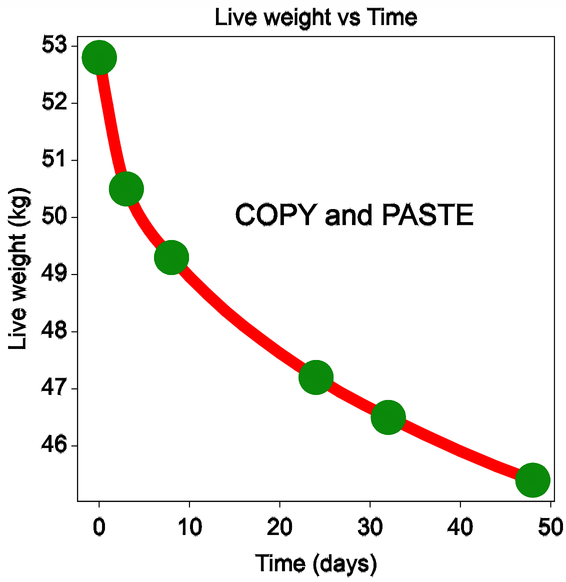












In the Tools menu of the Tools bar of the Graphics Viewer you can click on Options as shown in FigureA4.1

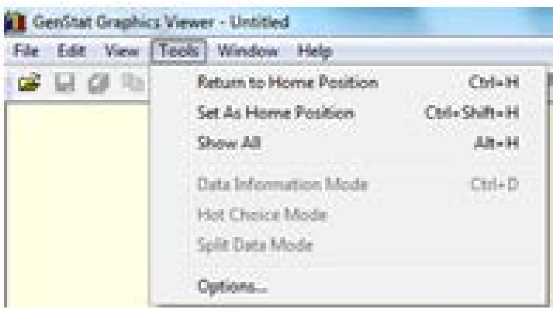


Figure A4.1

This gives the Options Layout in Figure A4.2. In the Saving Files tab you can specify the resolution. The 7 images here are produced given the current resolution (the default), but you can specify also: Save at predetermined resolution, and give your own height.

Another setting can be specified in the Advanced tab (see Figure A4.3 and **Appendix 5**) with respect to Quality. (This is equivalent to the `RESOLUTION` parameter of the `DEVICE` directive.) Here also the default setting is used for the 7 images.

Tip

Do not tick the Use bitmap fonts for copying and setting as Enhanced Metafile (.emf) box if you want to Copy and Paste the figure in Word. The quality of the picture is much lower then.

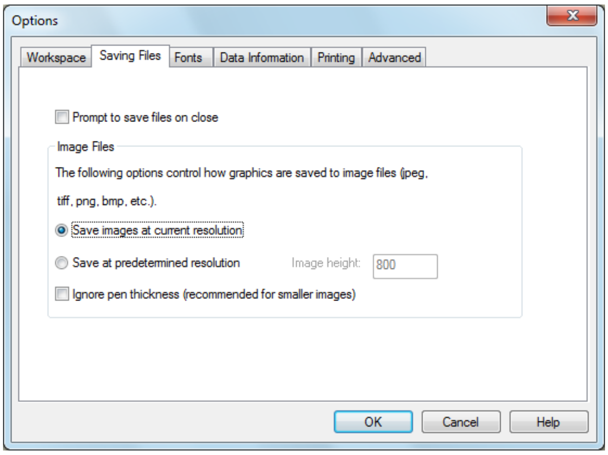


Figure A4.2

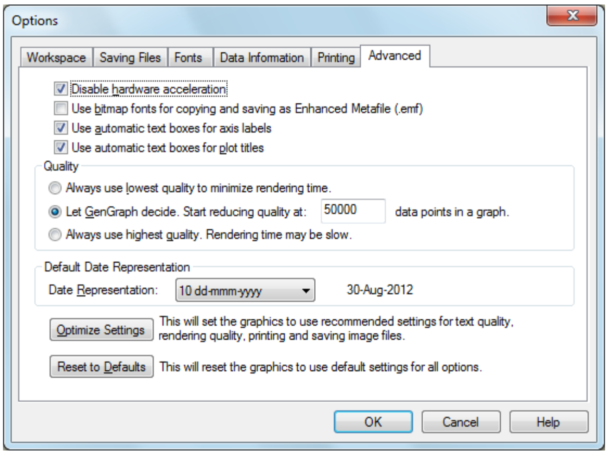


Figure A4.3

To move the image around in Word you will need to change the Picture Layout. To do this right click the picture and choose Size and Position... This will open the Layout dialog shown in Figure A4.4. To position the image where you want, select the Text Wrapping tab and choose a Wrapping style: options Square or Tight will flow the text around the picture. The image can then be dragged to anywhere in the text, and resized by selecting the picture and clicking on one of the corner boxes and dragging inwards to reduce the size, or outwards to increase it.

For some formats, like BMP, JPG, PNG and TIF, the margins around the image are too large. Click on the picture and a Picture Tools Format button appears on the Tool bar of Word. Click on Format and the ribbon shown in Figure A4.5 appears.

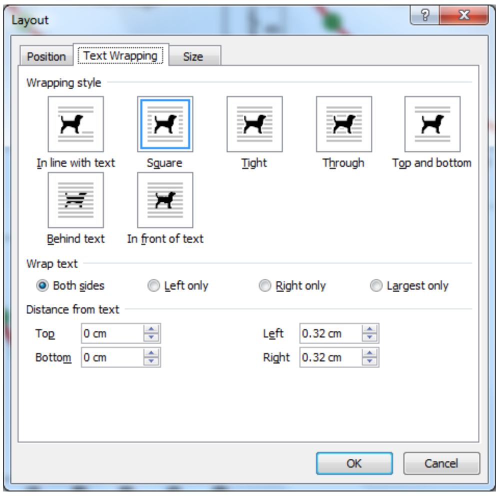


Figure A4.4

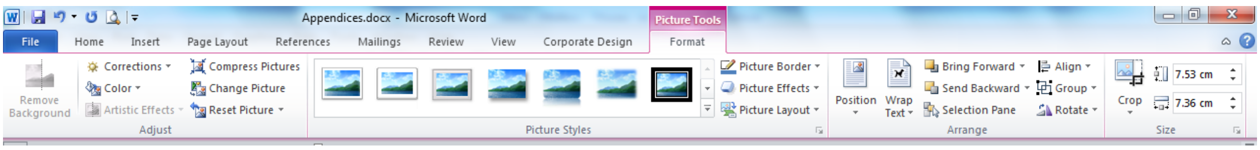
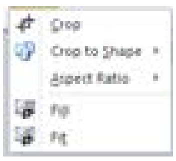


Figure A4.5

At the right of this ribbon there is a Crop menu

with 5 options:



Click on the Crop option and selection markers appear at the sides and corners of the image. Dragging them inwards removes (blank) parts of the image.

Tip

Given the images in this Appendix the formats EMF and EPS and the Copy and Paste method seem to give the best quality **printed on paper**. They also have minimal margins. However, after inserted them into Word, the view on the screen is too bold.

Example A4 on the next page produces the 7 images. This gives the opportunity to find out which formats and settings are best for your own configuration.

```
XAXIS        3 ; TITLE= 'Time (days)'
YAXIS        3 ; TITLE= 'Live weight (kg)'
VARIATE      [VALUES=52.8, 50.5, 49.3, 47.2, 46.5, 45.4] weight
VARIATE      [VALUES=   0,    3,    8,   24,   32,   48] time
PEN          1,2 ; COLOUR= 'red','green' ; METHOD=monotonic,point ; \
             SMSYMBOL=*,6 ; SYMBOL=0,2 ; THICKNESS=20,* ; CFILL=*,'green'
PEN          -1,-2,-3,-5 ; SIZE=2
FRAME        3 ; YMLOWER=0.13 ; XMLOWER=0.13
FOR          dd= '','bmp','emf','eps','gmf','jpg','png','tif'
  TXCONSTRUCT [TEXT=file]  'Liveweight.',dd
  TXCONSTRUCT [TEXT=ddu ; CASE=upper] dd
  TXCONSTRUCT [TEXT=dtext] ddu,' Image'
  IF         dd.NI.''
    SETDEVICE    file ; NUMBER=device
    DGRAPH       [WINDOW=3 ; KEYWINDOW=0 ; \
                 TITLE= 'Live weight vs Time'] \
                 2(weight) ; X=2(time) ; PEN=1,2
    PEN          4 ; SIZE=2.5 ; COLOUR= 'black'
    DTEXT        [WINDOW=3] 50 ; 20 ; TEXT= dtext ; PEN=4
    CLOSE        device ; FILETYPE=graphics
  ELSE
    DEVICE       1
    DGRAPH       [WINDOW=3 ; KEYWINDOW=0 ; \
                 TITLE= 'Live weight vs Time'] \
                 2(weight) ; X=2(time) ; PEN=1,2
    PEN          4 ; SIZE=2.5 ; COLOUR= 'black'
    DTEXT        [WINDOW=3] 50 ; 15 ; TEXT=  'COPY and PASTE' ; PEN=4
  ENDIF
ENDFOR
STOP
```

APPENDIX 5: ADVANCED OPTIONS of the GRAPHICS VIEWER

These options are provided to override certain default settings in cases where problems arise because of incompatibilities with Windows, other software or device drivers. It is recommended that you do not change these options unless problems occur. **In Appendix 4 you can find a screen shot in Figure A4.3.**

- Use Bitmap Fonts

GenStat graphics provides two kinds of font support: rendered fonts and bitmap fonts. The former are more suitable for on-screen viewing, as they will automatically resize when the image is zoomed in or out, whereas the bitmap fonts usually produce best results when printing or copying graphs.
Disabling bitmap fonts for saving as Enhanced Metafiles (.emf) and copying may help solve specific problems with text alignment or pitch, but usually at the cost of reduced quality and increased file size.

- Disable Hardware Acceleration

To display its graphical output, GenStat uses the OpenGL system. Many graphics cards provide on-board support for OpenGL, accessed via proprietary device drivers, in order to obtain the best performance. GenStat will try to use any hardware support that is available, but on some occasions problems may arise because of incompatibilities in the device driver. These can usually be corrected by downloading an updated driver (from the graphics card manufacturer's website). Alternatively, you can disable hardware acceleration, which should result in GenStat selecting a screen mode that uses software OpenGL support.
Symptoms arising from hardware incompatibilities can vary a great deal, from simple problems with colours appearing or disappearing, to unexplained crashes. In the worst case, the GenStat graphics viewer fails during initialisation of the display, so the option is also provided on the main GenStat options menu, on the Graphics tab.
Changes to this setting will not take effect until the GenStat graphics viewer is restarted.

- Use Automatic Text Boxes For Axis Labels

Selecting this option forces axis labels to fit into text boxes. The width of the text boxes is based on the distance between tick marks on the relevant axis. The text boxes can only contain one line of text. Turning off this option may result in axis labels overlapping.

- Use Automatic Text Boxes For Plot Titles

Selecting this option forces the plot title to fit into a text box based on the width of the plot.

- Quality

To smooth the appearance of the plot it is rendered several times, each in a slightly different position. By default GenGraph will reduce the number of times the plot is rendered based on the number of data points in the plot. This has the effect of reducing the amount of time taken to render the plot at the expense of a less smooth appearance. You can choose the number of data points at which GenGraph begins to reduce the quality. The quality is reduced further each time the number of data points in a plot exceeds a multiple of the the number chosen.
If you experience problems with slow rendering you can turn off this feature by choosing the option: Always use lowest quality to minimize rendering time. You can also choose to alway use the highest quality but this may result in slow rendering, especially if the plot contains many data points.
This only affects on-screen rendering.

- Default Date Representation

This sets the default date representation used by GenStat for axis labels.

- Optimize Settings

This will set the graphics to use recommended settings for text quality, rendering quality, printing and saving image files. Try using this if you are experiencing problems creating image files or printing. This can cause hardware acceleration to be disabled which will not take effect until the GenStat graphics viewer is restarted.

- Reset to Defaults

This will set the graphics to use default settings for all options including workspace, datainfo, saving files, fonts, printing and advanced options. This can cause hardware acceleration to be disabled which will not take effect until the GenStat graphics viewer is restarted.

- Default Font

This specifies the default font used by GenStat for textual output. This includes titles, axis labels, and strings plotted within the graph.

APPENDIX 6: IMAGE FUNCTIONS

These functions operate on images, represented as matrices of RGB values (see PEN for more details). **Example 47 demonstrates some functions.**

The functions are based on algorithms in the ImgSource library supplied by Smaller Animals Software, Inc. For more information, see http://www.smalleranimals.com/

`IM3CONVOLUTION`

Apply 3×3 convolution filter

`IM3CONVOLUTION`(r;f;i;cr;cg;cb;d) applies the convolution filter in the 3×3 matrix f to the RGB image in matrix r; scalar i (default 1) defines the intensity parameter; scalars cr, cg and cb contain 0 or 1 (default) according to whether the red, green and blue channels, respectively, are to be modified. If the "feedback" defined by scalar d is 0 (default), the new value at each point is i multiplied by the sum of the values at the point and nearby points multiplied by the convolution matrix. Alternatively, if d=i (default), the new value at each point is calculated by taking (1-i) multiplied by the current value at the point, and then subtracting i multiplied by the sum of the values at the point and nearby points multiplied by the values in the convolution matrix.

`IMBEQUALIZE`

Histogram equalization of brightness

`IMBEQUALIZE`(r;l;u) performs a histogram equalization of the brightness of the RGB image in matrix r; scalar l specifies the lower threshold and scalar h specifies the upper threshold.

`IMBLUR`

Blur

`IMBLUR`(r;b) blurs the RGB image in matrix r by the amount specified in scalar b (0<b<100; default 2).

`IMBRIGHTNESS`

Modify brightness

`IMBRIGHTNESS`(r;l;h;m) modifies the brightness of the RGB image in matrix r, setting pixels in each channel with brightness less than l (default 0) to 0 and those brighter than h (default 255) to 255; m defines the mode of adjustment (default 0 stretches brightness and 1 distributes brightness evenly across the range).

`IMBSTRETCH`

Histogram stretch of brightness

`IMBSTRETCH`(r;l;u;m) performs a histogram stretch of the brightness in the RGB image in matrix r; scalar l (default 0) specifies the percentage of pixels to set to 0 (i.e. black), scalar h (default 0) specifies the percentage of pixels to set to white, and scalar m (0≤m≤255; default 128) specifies the colour value in each channel to be set to the middle intensity.

`IMCEQUALIZE`

Histogram equalization of individual colours

`IMCEQUALIZE`(r;l;u) performs an independent histogram equalization of the colours in the RGB image in matrix r; scalar l specifies the lower threshold and scalar h specifies the upper threshold.

`IMCONTRAST`

Modify contrast and brightness

`IMCONTRAST`(r;c;b) modifies contrast and brightness of the RGB image in matrix r; c (-1≤c≤1; default 0 i.e. no adjustment) defines the adjustment to the contrast, and b (-1≤b≤1; default 0 i.e. no adjustment) defines the adjustment to the brightmess.

`IMCREPLACE`

Replace one colour with another

`IMCREPLACE`(r;c;d;t) replaces colour c in the RGB image in matrix r with colour d, using tolerance t (default

0).

IMCSTRETCH

Histogram stretch of individual colours

IMCSTRETCH(r;l;h;m) performs a histogram stretch of the individual colours in the RGB image in matrix r; scalar l (default 0) specifies the percentage of pixels to set to 0 (i.e. black), scalar h (default 0) specifies the percentage of pixels to set to white, and scalar m (0≤m≤255; default 128) specifies the colour value in each channel to be set to the middle intensity.

IMDESPECKLE

Despeckle

IMDESPECKLE(r) despeckles the RGB image in matrix r.

IMELLIPSE

Draw ellipse

IMELLIPSE(r;cx;cy;hr;vr;c;cf;p) draws an ellipse with centre (cx, cy), horizontal radius hr (default 40), vertical radius vr (default 40), colour cl, fill colour cf (default 0) and opacity p (0≤p≤1, where 0 is transparent and the default of 1 is solid) on the RGB image in matrix r.

IMEMBOSS

Emboss

IMEMBOSS(r;b;t;a;e;d) embosses the RGB image in matrix r; matrix b specifies a "bump map" defining the peaks and valleys in the output image (typically this is a grey scale version of r); matrix t defines the texture to apply to the input matrix; scalar a gives the angle of the light source in radians; scalar e is the elevation of the light source in radians; scalar d should be set to either 0 or 1 to select between the two available blending algorithms (1 produces a better effect with textured images).

IMGAMMA

Apply gamma correction to brightness

IMGAMMA(r;g) applies gamma correction g (g≥0; default 1.5) to the brightness of the RGB image in matrix r; g<1 decreases brightness, and g>1 increases brightness.

IMGBLUR

Gaussian blur

IMGBLUR(r;s) applies a Gaussian blur with standard deviation s to the RGB image in matrix r.

IMGRAYSCALE or IMGREYSCALE

Convert to grey scale

IMGREYSCALE(r) or IMGRAYSCALE(r) convert the RGB image in matrix r to grey scale

IMHFLIP

Horizontal flip

IMHFLIP(r) performs a horizontal flip on the RGB image in matrix r.

IMLINE

Draw line

IMLINE(r;x1;y1;x2;y2;c) draws a line from point (x1, y1) to (x2, y2) in colour c on the RGB image in matrix r.

IMMCONVOLUTION

Apply convolution filter

IMMCONVOLUTION(r;f;i;cr;cg;cb;m) applies the convolution filter in matrix f to the RGB image in matrix r ; scalar i (default 1) defines the intensity parameter; scalars cr, cg and cb contain 0 or 1 (default) according to whether the red, green and blue channels, respectively, are to be modified. If the mode defined by scalar m is 0 (default), the new value at each point is i multiplied by the sum of the values at the point and nearby points multiplied by the convolution matrix. Alternatively, if m=1 (default), the new value at each point is the current value at the point minus i multiplied by the sum of the values at the point and nearby points multiplied by the values in the convolution matrix.

IMMEDIANFILTER

Median filter

IMMEDIANFILTER(r) performs a median filter on the RGB image in a matrix r.

IMOVERLAY

Overlay images

IMOVERLAY(rt;rb;m;mp;p;x;y) overlays the RGB image in matrix rt over the RGB image in matrix rb; m controls how images are blended (0 = fast blend, 1 = slower, more accurate blend, 2 = pixels combined with logical AND, 3 = pixels combined with logical OR, 4 = pixels combined with logical XOR, 5 = output pixel is maximum of top and bottom as in Photoshop "Lighten", 6 = output pixel is minimum of top and bottom as in Photoshop "Darken", 7 = output pixel is sum of top and bottom, 8 = output pixel is difference of top and bottom, 9 = if top > mp, output top, 10 = if top < mp, output top, 11 = absolute value of the difference of top and bottom, 12 = take top × bottom / maximum component, 13 = take top × bottom × ModeParameter / maxComponent, 14 = screen, 15 = define bottom to be bottom + top - mp, 16 = define bottom to bottom - top - mp, 17 = pixels combined with logical NAND, 18 = pixels combined with logical NOR, 19 = pixels combined with logical NXOR/XNOR, 20 = color dodge, 21 = color burn, 22 = soft dodge, 23 = soft burn, 24 = Photoshop "overlay", 25 = soft light, 26 = hard light, 27 = XFader reflect, 28 = XFader glow, 29 = XFader freeze, 30 = XFader heat); p defines the opacity of the blended image; and (x, y) specifies the position of bottom left-hand corner of the top image on the bottom image.

IMPUSH

Point-to-point warp

IMPUSH(r;x1;y1;x2;y2) applies a point-to-point warp on the RGB image in matrix r, "pushing" point (x1, y1) to (x2, y2).

IMRECTANGLE

Draw coloured rectangle

IMRECTANGLE(r;x1;y1;x2;y2;c) colours the rectangle with bottom left corner (x1, y1) and top right corner (x2, y2) in the RGB image in matrix r to be colour c.

IMROTATE

Rotate

IMROTATE(r;a;b) rotates the RGB image in matrix r; a is the angle in radians (default π/2); b is the background colour to put into the (blank) corners.

IMSATURATE

Adjust saturation

IMSATURATE(r;s) adjusts the saturation level of the RGB image in matrix r according to the value of scalar s (default 1.1): when s>1 the saturation is increased, when 0<s<1 saturation is decreased, and when s<0 photo-negative is generated.

IMSHARPEN

Sharpen

IMSHARPEN(r;s) sharpens the RGB image in matrix r by the amount specified in scalar s (0<s<100; default 2).

IMSIZE

Change size

IMSIZE(r;w;h;m) changes the sizes of the RGB image in matrix r to have width w and height h; m selects the algorithm to use to assign colours in the new image: 0 = box filter, 1 = triangle filter, 2 = Hamming filter, 3 = Gaussian filter, 4 = bell filter, 5 = B-spline filter, 6 = cubic 1 filter, 7 = cubic 2 filter, 8 = Lanczos3 filter, 9 = Mitchell filter, 10 = sinc filter, 11 = Hermite filter, 12 = Hanning filter, 13 = Catrom filter, 14 = fast area-average, 15 = area-average, 16 = bi-linear interpolation, 17 (default) = bi-cubic interpolation, 18 = nearest neighbour.

IMSSTRETCH

Histogram stretch of saturation

IMSSTRETCH(r;l;h;m) performs a histogram stretch of the saturation in the RGB image in matrix r; scalar l (default 0) specifies the percentage of pixels to set to 0 (i.e. black), scalar h (default 0) specifies the percentage of pixels to set to white, and scalar m ($0 \le m \le 255$; default 128) specifies the colour value in each channel to be set to the middle intensity.

IMSTEXT

Draw smooth text

IMSTEXT(r;st;c;fh;y1;x1;y2;x2;ft;tr;sm) draws the text in string st with height fh, font ft, colour c, transparency tr and smoothness sm (sm=1 for none, or 2 or 4) within the bounding rectangle with top left corner at (x1, y1) and bottom right corner at (x2, y2) on the RGB image in matrix r.

IMTEXT

Draw text

IMTEXT(r;st;c;fh;y1;x1;y2;x2;ft) draws the text in string st with height fh, font ft and colour c within the bounding rectangle with top left corner at (x1, y1) and bottom right corner at (x2, y2) on the RGB image in matrix r.

IMUNSHARPEN

Apply unsharp mask

IMUNSHARPEN(r;t;a;s) applies an unsharp mask to the RGB image in matrix r: this first applies a Gaussian blur with standard deviation s; it then finds the difference between pixels in the blurred image and in the original and, if this is greater than t in each channel, it adds the amount specified by scalar a multiplied by the difference from the original value.

IMVFLIP

Vertical flip

IMVFLIP(r) performs a vertical flip on the RGB image in a matrix r.

IMXSHEAR

Shear in x direction

IMXSHEAR(r;x;b) shears the RGB image in matrix r by moving the top of the image |x| pixels to the right (x>0) or left (x<0); the blank parts of the new image are given (background) colour b.

IMYSHEAR

Shear in y direction

IMYSHEAR(r;y;b) shears the RGB image in matrix r by moving the right-hand side of the image |y| pixels up or down; the blank parts of the new image are given (background) colour b.

APPENDIX 7: BARS, PANES AND WINDOWS

Bars

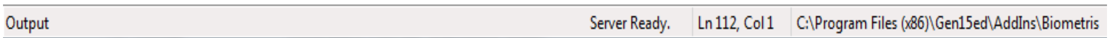
Entirely at the top of GenStat for Windows is the Title bar.
If, for example, the Input Log Window is active, the Title bar looks like:

Title bar 

The next 3 bars are called:
Menu bar
Standard toolbar
Spreadsheet toolbar

The Standard toolbar and the Spreadsheet toolbar can be adjusted with:
Tools (in the Menu bar) → Customize toolbar … → Toolbar: Standard
or → Toolbar: Spreadsheet

Entirely at the bottom is the Status bar.
If, for example, the Output window is active the Status bar looks like:

Status bar 

Task bar of Windows 7:
Task bar 

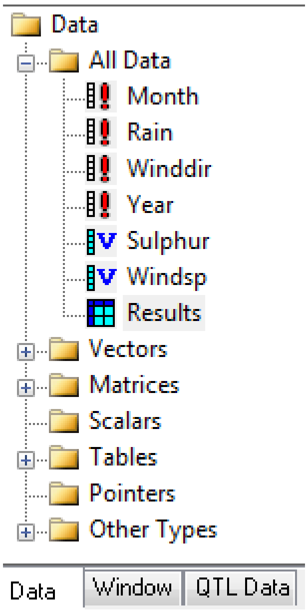
Tip

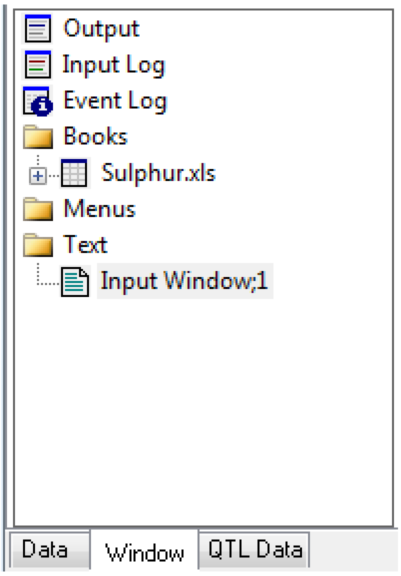
Both toolbars can also be customized by right clicking in the open space of the bar. In the standard toolbar Working directory, Restart Server and Clear Output and data are useful extra buttons.

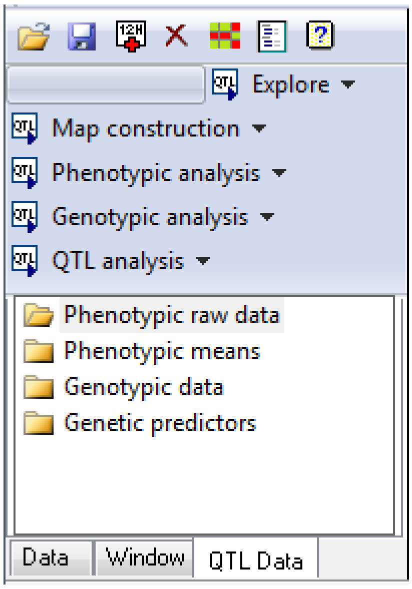
Panes

At the left of the screen there is a **docked Pane** with 3 Tabs: the **Data View** , the **Window navigator** and the **QTL Data View**.
If the Data tab is clicked the Data View Pane is visible. Clicking on the Window Navigator tab or the QTL Data View tab opens one of the other 2 panes:

Data View: **Window navigator**: **QTL Data View**:







In the Window Navigator the **Output**, **Input Log** and **Event Log** windows are always present.
Books are: spreadsheets, **Menus**: windows opened via the menus and **Text**: Input windows with GenStat commands.

All 3 tabs can be hidden by clicking with the right Mouse button on the Tab and then clicking with the left Mouse button on Hide. You get the tabs of the pane back by clicking:
View → Data View or View → Window Navigator or View → QTL Data View

Windows

As you can see in the Window navigator pane, there are 6 types of windows. The Output, Input Log and Event Log are always present. The Books are the open spreadsheets, the Menus the open menus, and the Text the different open windows.

When you open a text file, usually a GenStat program, an Input Window is opened with the name of that file. If you run an example program from the Help, or if you click on the New Text Window button on the Standard Tool bar an Input Window is opened with the name Input Window;n. If it is the first Input Window n equals 1; if there are more, n is one larger than the largest n till then.
You can switch between all windows with the Previous and Next Window buttons on the Standard Tool bar: