

A Visual Revolution – Statistical Graphics in SAS 9.2

Bob Newman, Amadeus Software Limited, Oxfordshire, United Kingdom

ABSTRACT

This paper gives an introduction to the ODS Statistical Graphics facilities of SAS 9.2, emphasising ease of use. The facilities are compared and contrasted with those of traditional SAS graphics. The emphasis is on what can be produced, rather than the code required to produce it.

This paper will be relevant for Programmers and Statisticians responsible for producing figures for clinical output.

INTRODUCTION

ODS Statistical Graphics was introduced in SAS 9.2, and enhanced in SAS 9.2 Phase 2. For many types of plot, ODS Statistical Graphics (which will be referred to in this paper as “SG”) is much quicker and easier to use than traditional SAS graphics (“TG”). This is particularly true for some commonly-required types of composite plot, such as matrix plots and trellis plots. This paper exhibits a selection of the plots that can be produced using SG, with brief notes on how to go about it.

SG provides three methods for creating user-defined plots:

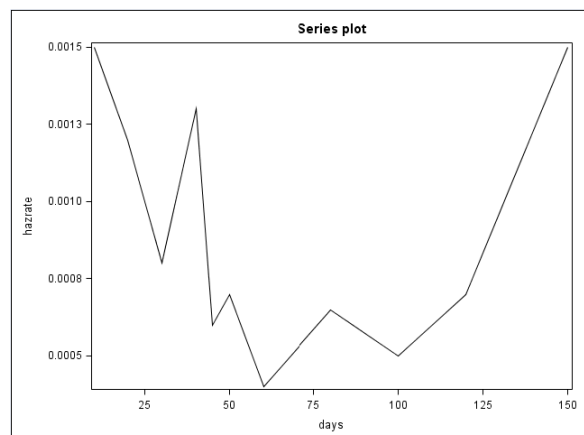
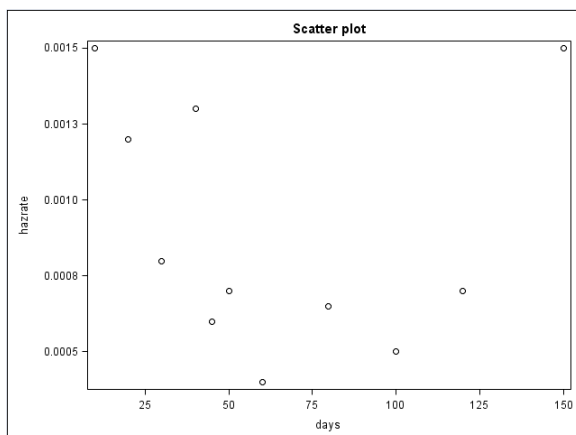
- New SG procedures such as Proc SGPLOT
- A new Graphics Template Language (GTL)
- In SAS 9.2 Phase 2, a new utility called the ODS Graphics Designer. This is by far the easiest method to use.

Where similar plots can be produced by either SG or TG, it will almost always be preferable to use SG. There are however some kinds of plot that can be produced using TG, but not (yet) using SG. The two types of graphics can be expected to co-exist for some time. The techniques they use are quite different.

BASIC GRAPHS USING PROC SGPLOT

SIMPLE PLOTS

Here are a couple of simple plots produced from the same data set using Proc SGPLOT.



They look similar to the equivalent plots that could be produced using TG. The code required to produce them is straightforward.

```
title "Scatter plot";
proc sgplot data=haz1;
  scatter x=days y=hazrate;
run;
```

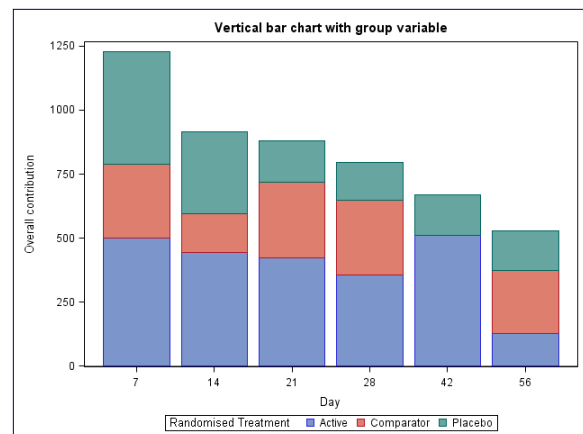
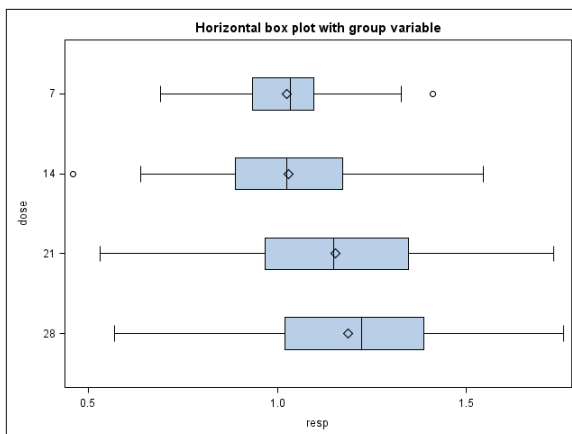
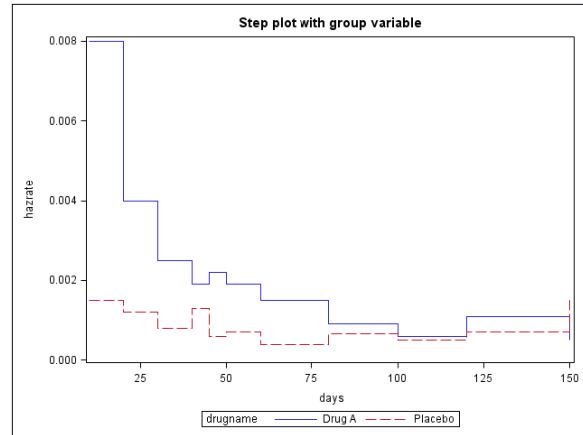
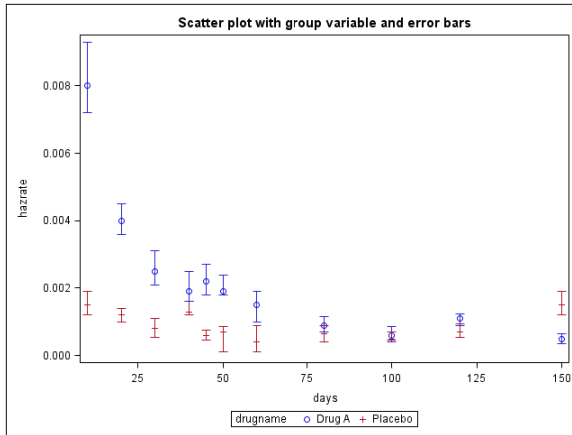
```
title "Series plot";
proc sgplot data=haz1;
  series x=days y=hazrate;
run;
```

PhUSE 2009

Notice that only one word changes. A scatter plot is produced using a SCATTER statement, and a series plot using a SERIES statement with identical syntax. (Compare TG, where it would have been necessary to change the interpolation option on a SYMBOL statement.) If you use the ODS Graphics Designer instead, you don't have to write any code at all.

GROUPED PLOTS

Here are some plots, again produced using Proc SGPLOT, that use a group variable.



Each of these uses the default colours, line types, markers etc as defined by the default style, and legends are generated automatically. The amount of code required is still small. For the box plot, for example:

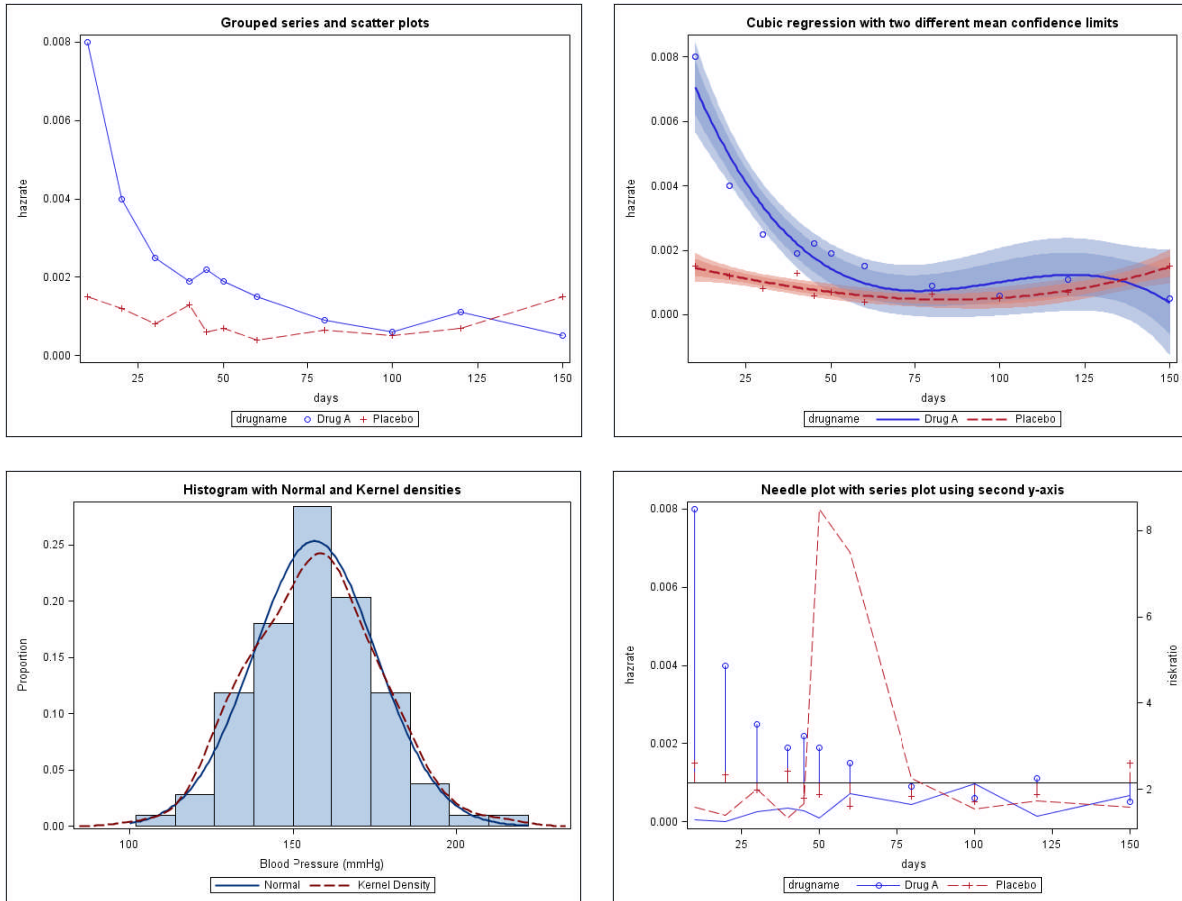
```
title 'Horizontal box plot with group variable';
proc sgplot data=amadeus.response;
  hbox resp / category=dose;
run;
```

Notice that SG's "Grouped Bar Chart" is what was known to TG as a "Stacked Bar Chart". We will see some more variations on the bar chart theme later.

SUPERIMPOSING PLOTS

Plots can be superimposed simply by specifying more than one plot in the same call of Proc SGPLOT. Here are some examples.

PhUSE 2009



Not all types of plot can be superimposed in this way – Proc SGPLOT regards some combinations as incompatible. Note also that all the plots produced in a single call to Proc SGPLOT will necessarily be using the same input data set, though they can use different variables from it.

The regression fit above is one type of “Fit and Confidence” plot. Loess plots, spline plots and confidence ellipses are also available. This example superimposes two separate regression fits for different confidence levels. Transparency is specified so that, where red and blue areas overlap, both can be seen.

Notice that the final example uses a second y-axis (as for the PLOT2 statement in TG). In SG, it is simply a matter of specifying the Y2AXIS option. The SG facility is more general, in that it is also possible to use a second X-axis at the top of the graph.

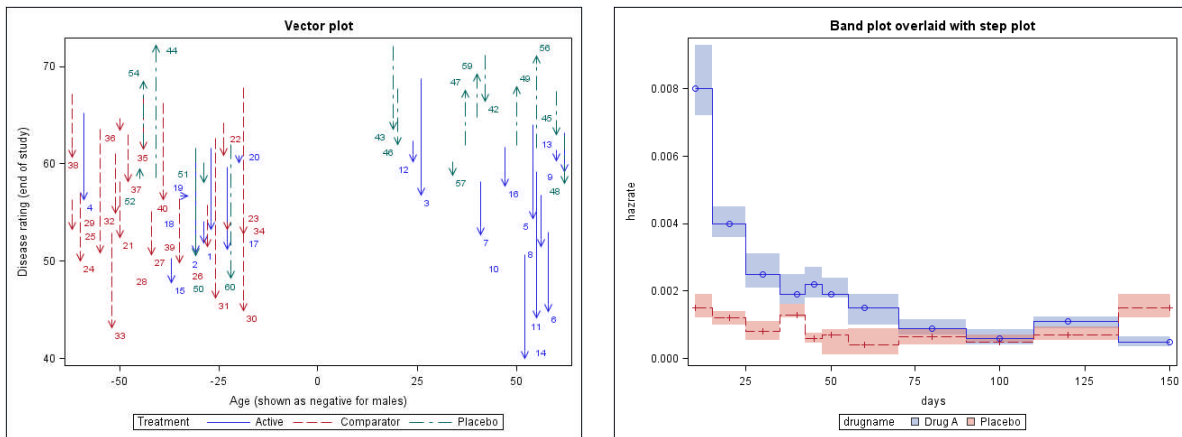
Notice also that in every case where grouped plots are superimposed, they are colour co-ordinated e.g. in every plot, the “Placebo” data is coloured red. SAS handles this automatically.

The coding required here is still straightforward. For the histogram and density plot, for example:

```
title 'Histogram with Normal and Kernel densities';
proc sgplot data=amadeus.bpsys;
  histogram supsys /scale=proportion;
  density supsys / scale=proportion;
  density supsys / scale=proportion type=kernel;
run;
```

NOVEL PLOT TYPES

The Vector plot and the Band plot do not have exact equivalents in TG.



In a Vector plot, arrows are drawn between pairs of points. This plot type became available in Proc SGPLOT in SAS 9.2 Phase 2.

In a Band plot, the area between two lines is shaded. This is similar to a "filled line plot" in TG. In this example, the band is between variables containing upper and lower estimates; this is overlaid on a step plot of the hazrate variable itself.

There are some other plot types, such as the Block plot and Fringe plot, that are supported by GTL and by the ODS Graphics Designer, but not available in Proc SGPLOT. Examples will be given later.

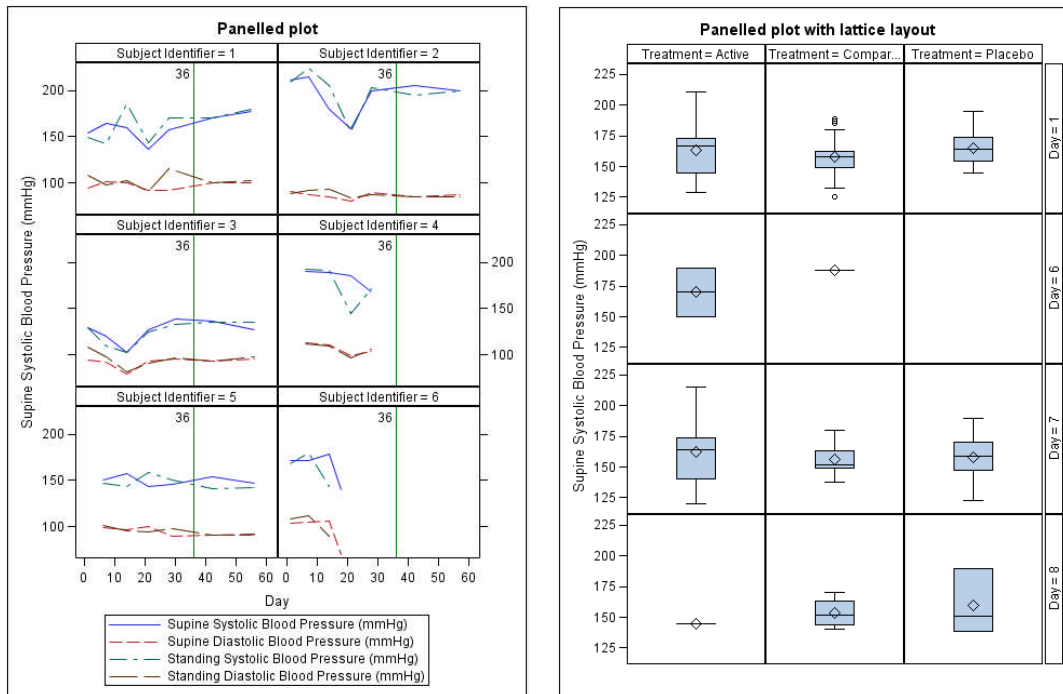
PANELLED GRAPHS USING PROC SGPANEL AND PROC SGSCATTER

SGPANEL

Proc SGPANEL enables multiple similar plots to be displayed on the same page. Using SGPANEL is similar to using Proc SGPLOT with BY-group processing (which is also supported), except that SGPANEL uses its own statement PANELBY and puts all the plots onto the same page.

Roughly speaking, any plot that can be produced using Proc SGPLOT can be produced in a panelled form using Proc SGPANEL. However there are a few restrictions, the main one being that the second X-axis and second Y-axis are not supported (X2AXIS and Y2AXIS options).

Here are two examples of panelled plots:



In the first example, each individual plot relates to a different subject, and contains four series plots and a reference line. Notice the common axis labels and common legend. To produce this kind of plot in TG, using Proc GREPLAY,

would not have been easy.

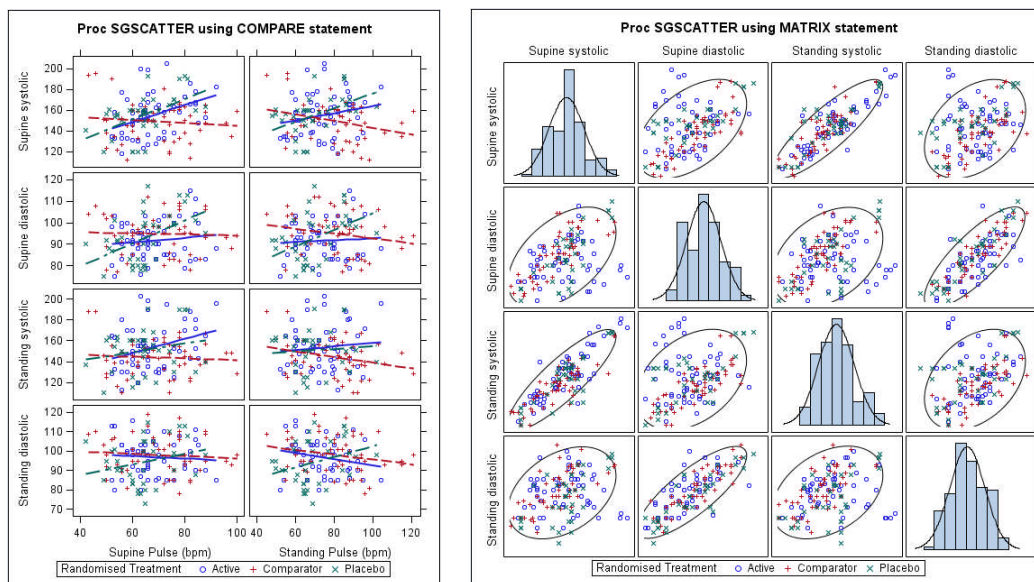
The second example has a box plot for each treatment and each day, arranged in a lattice layout with treatments running horizontally and days vertically.

The amount of code required to produce these rather complex plots is still surprisingly small, for example:

```
title 'Panelled plot with lattice layout';
proc sgpanel data=amadeus.bloodpressure;
  panelby trtcd day / layout=lattice rows=4;
  label trtcd='Treatment';
  vbox supsys;
run;
```

SGSCATTER

As its name suggests, Proc SGSCATTER is concerned mainly with scatter plots, in varying combinations. It has three main plotting statements (with a fourth introduced in SAS 9.2 Phase 2) enabling combinations of variables to be expressed in various ways.



The COMPARE statement produces scatter plots of each of several X-axis variables against each of several Y-axis variables. In the example, linear regression is performed for each value of the group variable "Randomised treatment".

The MATRIX statement produces scatter plots of all possible pairs of variables from a given list. Optionally (as here), distribution plots of the individual variables are shown along the diagonal of the matrix. This example also includes confidence ellipses. The code to produce even this last example is not extensive:

```
title 'Proc SGSCATTER using MATRIX statement';
proc sgscatter data=bp2;
  matrix supsys supdia stdsys stddia
    / group=trtcd ellipse=(alpha=0.10) diagonal=(histogram normal);
run;
```

COMPOSITE AND NON-STANDARD GRAPHS

So far we have seen different types of plot superimposed, and panelled graphs containing multiple similar plots. Is it also possible to produce panelled graphs containing more than one different type of plot – for example, a scatter plot and a bar chart side by side? Yes it is – although note that all the plots within the graph must use the same input data set. There are also a number of plot types which are not (yet) implemented in Proc SGPLOT, but are available in other ways.

There are two main ways of exploiting these features:

THE HARD WAY

You can get your hands dirty and write your own templates in GTL. Learning the whole of GTL would be a major

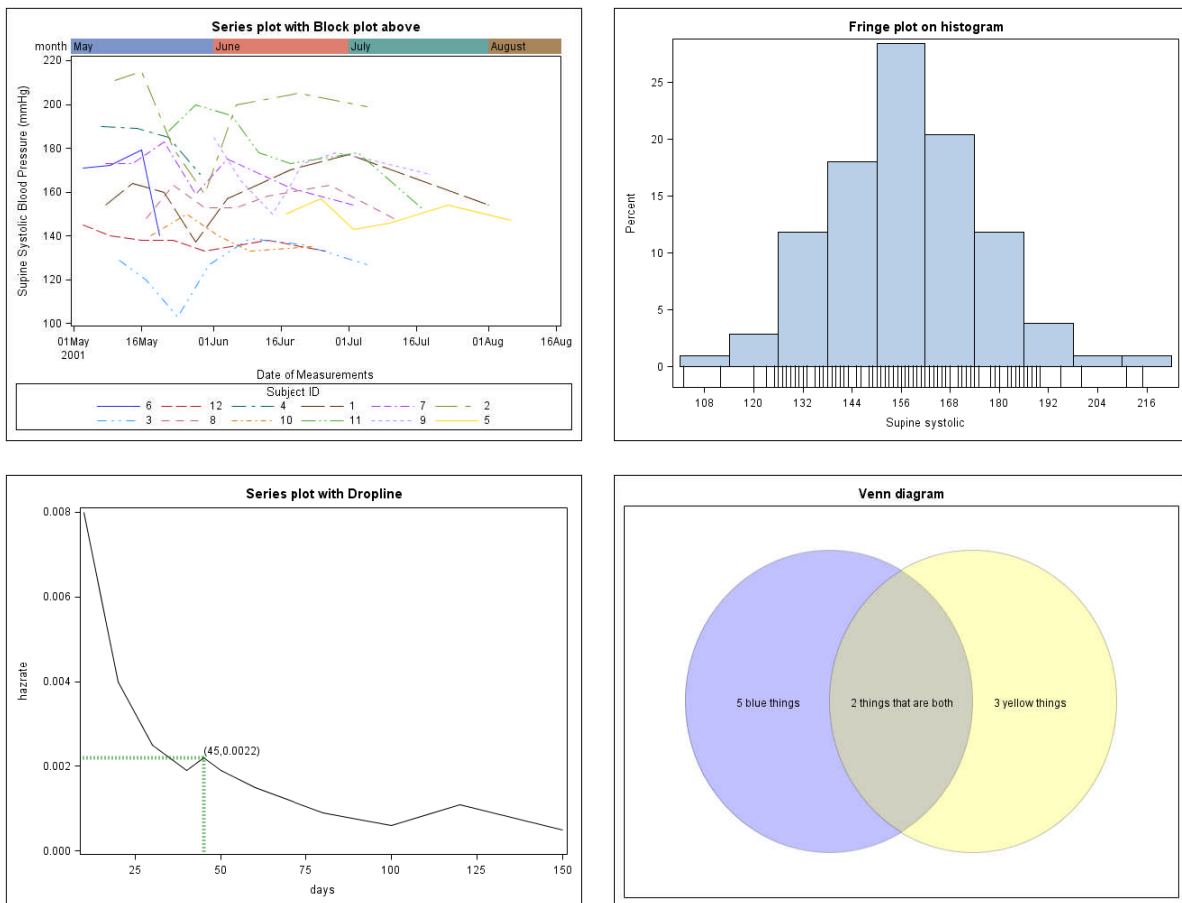
PhUSE 2009

undertaking, but it is relatively easy to learn enough to be able to do something useful. Examples of GTL code are readily available. Each time SGPLOT, SGPanel or SGSCATTER produces a graph, it does so by first producing the GTL to generate a graphics template. You can specify an option TMPLOUT to any of these procedures which will cause them to output this GTL code to a text file. Modifying existing code is always less daunting than producing new code from scratch.

Having defined a graphics template by running your GTL code, you can then produce a graph in one of two ways:

- Using Proc SGRENDER, which needs as inputs a template and a data set.
- Using a special form of data step with ODS FILE PRINT. This does exactly the same thing.

These examples were all produced using custom-written GTL code. All of them use GTL features that are not available in procedures such as Proc SGPLOT.



The first example may look like a single plot to the naked eye, but in fact it is two plots one above the other. The Block plot is the coloured panel that contains the month names.

The Fringe plot below the histogram has a whisker for each individual data point.

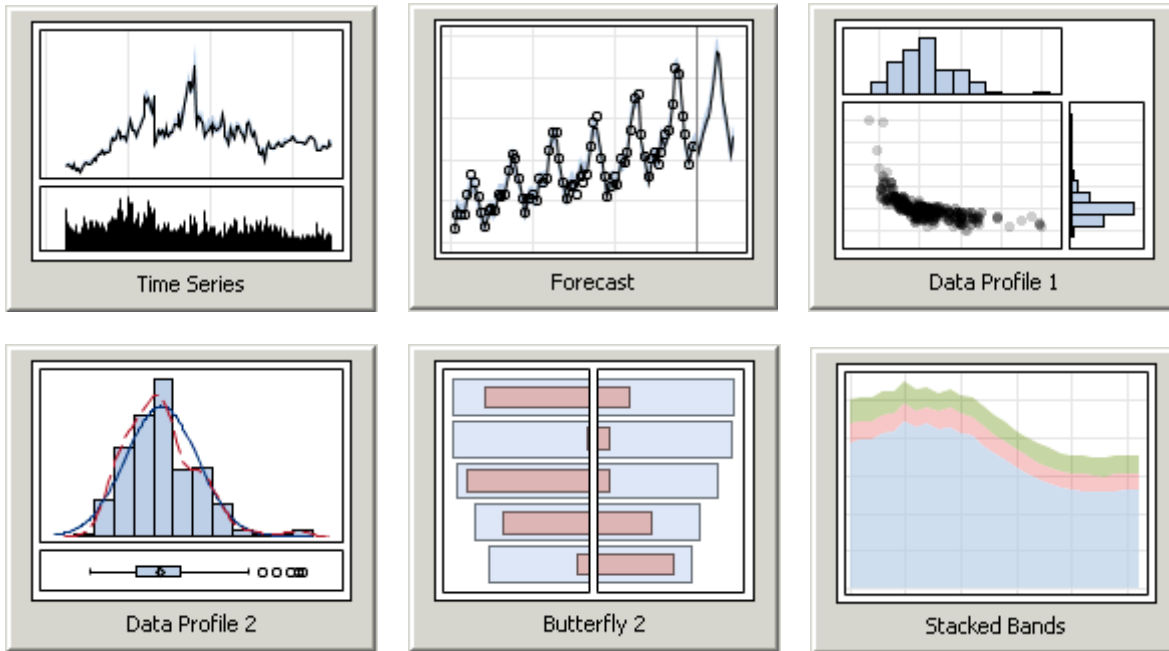
The Dropline in SG is a form of plot that can be overlaid on other plots. In TG, it would have been regarded as a form of annotation.

The Venn diagram example shows creative use of the ELLIPSEPARM statement.

THE EASY WAY

If you have SAS 9.2 Phase 2, the ODS Graphics Designer will come to your rescue. You can define your own composite graphs, using components which include all of the SGPLOT plot types, as well as contour plots, block plots, fringe plots, droplines etc. A number of types of composite graph are available as standard, for example:

PhUSE 2009



These are some of the options as they appear in the Graph Gallery within the ODS Graphics Designer.

Notice especially the Butterfly chart, a form of bar chart which is standard in SG. In TG terms, this is equivalent to two horizontal grouped bar charts back-to-back. However, the bars within a group are not side by side; rather they have a common centre and different widths.

Having designed your graph, you can save the definition in a .SGD file and subsequently regenerate the graph using Proc SGDESIGN.

MAKING GRAPHS GENERIC

In its basic form, a graphics template is, to some extent, specific to a particular data set. (Actually, to be precise, it is not the data set name that is fixed, but the names of the variables to be plotted. It would be possible to re-use a template with a different data set, provided that it contained variables with the same names).

It is possible to parameterise a graphics template so that the names of the variables to be plotted can be specified (typically to Proc SGRENDER) at run time. This is normally done by modifying the GTL to use dynamic variables, whose values can then be specified in the Proc SGRENDER code.

A similar facility, known as "Shared Variable" (SV) graphs is planned for the ODS Graphics Designer, but this has not been included in the initial release of SAS 9.2 Phase 2. The idea is that the names of the shared variables can be specified when replaying the graph using an .SGD file and Proc SGDESIGN.

EDITING AND ANNOTATING GRAPHS

SG's annotation facilities are less extensive than those of TG.

There are number of annotation-like facilities available within the SG procedures and/or GTL, for example:

- INSET boxes to contain text
- Reference lines and droplines
- DATALABEL option enabling individual points to be labelled with their values (like POINTLABEL in TG)
- GTL statement LINE, to plot a line through a specified point with a specified gradient
- GTL statement ENTRY to write a text string. This can include subscripts, superscripts, and/or Unicode characters, and (using the PAD option) can be positioned anywhere on the plot.

The above can all be included in the plot, and are not regarded as annotations in SG.

There is no equivalent in SG to the Annotate facility of TG.

The ODS Graphics Editor in SG (definitely *not* to be confused with the ODS Graphics Designer) is broadly equivalent to the Graphics Editor in TG. It is not programmable. It only allows one-off changes to be made manually to a particular image file.

In the ODS Graphics Editor, *editing* and *annotation* are two distinct sets of facilities. Editing facilities include resizing, selecting a different ODS style, and adding or removing titles and footnotes. Annotation facilities include the addition, modification or removal of text, lines, circles and images.

An editable image (and an image is only editable if it was specifically created to be editable) can be both edited and annotated.

A PNG image (even if not created editable) can still be annotated.

PhUSE 2009

The annotation facilities, when available, are broadly similar to those of TG, and in some respects go beyond anything that is available in the SG procedures or GTL e.g. rotation of text strings through any angle.

STATISTICAL GRAPHICS AND TRADITIONAL GRAPHICS

WHY DO WE NEED BOTH?

Where SG and TG can produce equivalent plots, it will almost always be preferable to use SG. And many non-SAS/GRAPH procedures have already been modified so that they are now capable of producing SG graphics. But there are several reasons why TG will not be going away just yet:

- There are numerous TG procedures (for example those relating to 3D plots) for which there is no SG equivalent.
- Even where there is an equivalent SG procedure, not all features of the TG procedure may be reproduced (e.g. bubble plots, pie charts, and certain kinds of regression plot supported by TG are not currently available in SG).
- TG's facilities for annotating plots are better than those of SG.
- Some wholly new TG procedures (GTILE and GKPI) were introduced in SAS 9.2, at the same time as SG.

The two different graphics systems do not mix easily. Putting together a composite plot in which some parts are produced using SG and others using TG can be a challenge. One possible approach (if HTML output is required) is to use the HTMLPANEL tagset, which can be downloaded from the SAS Technical Support site.

DIFFERENCES BETWEEN THE TWO SYSTEMS

The following table summarises the major differences.

ODS Statistical Graphics	Traditional SAS Graphics
Template-based	Device-based
Easy to code, with relatively few options. Usually does what you want by default. If you want something unusual, it may be difficult or impossible.	Harder to code, with many options and interactions between them. Almost any result is achievable, if you are prepared to work hard enough.
Always creates an image file (by default in PNG format). Does not use catalogues and GRSEG entries.	Always creates a GRSEG ('graphics segment') entry in a SAS catalogue. Optionally, an image file may also be created.
Composite plots created by using a panelled layout (e.g. using Proc SGPANEL).	Composite plots created from GRSEG entries using Proc GREPLAY.
Graphics options specified using ODS GRAPHICS statement.	Graphics options specified using GOPTIONS statement.
Properties of axes, legends, markers, fills etc are determined by options and parameters within the SG procedures and/or GTL, and by the ODS style.	Many properties are specified by global statements AXIS, LEGEND, SYMBOL, PATTERN, NOTE.
TITLE and FOOTNOTE global statements are supported, except for certain options.	TITLE and FOOTNOTE global statements are fully supported.
For text, only system fonts (such as Arial) are supported.	For text, both system fonts (such as Arial) and SAS/GRAPH fonts (such as SWISSB) are supported.
A pre-determined set of 24 marker symbols are (for almost all purposes) the only ones available.	There is a pre-determined set of standard marker symbols, but other fonts can be used instead e.g. MARKER, ORFONT.
Area fills can use only solid colour. Hatchings are not supported. Transparency is available for fills, and for many other graphics elements.	Fills can use solid colour or a range of hatchings. Transparency is not supported.
Scaling and anti-aliasing are both enabled by default.	Neither scaling nor anti-aliasing is supported.
ODS styles are supported throughout SG, and this includes some new style elements – but not all of the style elements supported by TG (e.g. drop shadows, background images for plots).	New SG-specific style elements are not supported.
Relatively limited annotation facilities – although the plot is less likely to need any non-standard annotations.	Comprehensive annotation facilities.

CONCLUSION

ODS Statistical Graphics really has revolutionised the world of SAS graphics – although there will remain a place for traditional SAS graphics for some time to come.

PhUSE 2009

REFERENCES

Delwiche, Lora and Slaughter, Susan. 2008. "Using PROC SGPLOT for Quick High Quality Graphs".
Schwartz, Susan. 2008. "Butterflies, Heat Maps, and More: Explore the New Power of SAS/GRAPH".
Heath, Dan. 2009. "Secrets of the SG Procedures".
Cartier, Jeff. 2008. "The Power of the Graphics Template Language".
Matange, Sanjay. 2009. "ODS Graphics Designer - An Interactive Tool for Creating Batchable Graphs".

ACKNOWLEDGEMENTS

Peer Reviewed: Ian Amaranayake Principal Consultant, Amadeus Software Limited.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Author Name:	Bob Newman
Company:	Amadeus Software Limited
Address:	Orchard Farm, Witney Lane, Leafield, Oxon OX29 9PG
Work Phone:	01993 878287
Fax:	01993 878042
Email:	bob.newman@amadeus.co.uk
Web:	www.amadeus.co.uk

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.