

ABSTRACT

Getting information from a SAS data set to a graph is relatively simple. Gchart and Gplot Procedures can be done in only 4 or 5 lines of code. Making the graph look nice and displaying other information for users, well that's a completely different matter. Symbol, Pattern and Axis statements, Annotate data sets, Device drivers, Reference Lines and User defined templates are all available to the programmer when they create graphics via the medium of the SAS System. This paper will look at these options and more in detail and demonstrate the way in which SAS can produce better-looking graphics.

INTRODUCTION

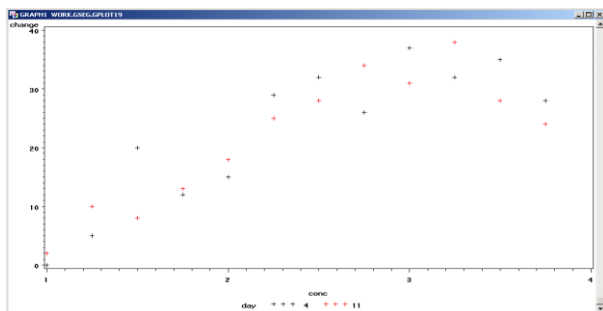
Many SAS programmers shy away from the area of SAS/GRAPH as the code can appear quite involved and convoluted. This paper will (hopefully) make these options and statements easier to understand. Graphics can be produced from SAS data sets in a matter of 3 or 4 lines of code. In order to make the graphics more presentable to users/managers a number of options are available to the programmer. Default symbol settings and pattern settings are easily overwritten and can make the graph look a lot more professional. Annotation can be added and default graphic sizes can be altered to name but a few of the programming approaches available to us, the programmer.

SIMPLE GRAPHICS

As alluded to earlier producing simple graphics is a very easy task within the SAS System. Two procedures available within the SAS/GRAPH software (GLOT and GCHART) allow the programmer to produce either a plot or a chart in 4 statements.

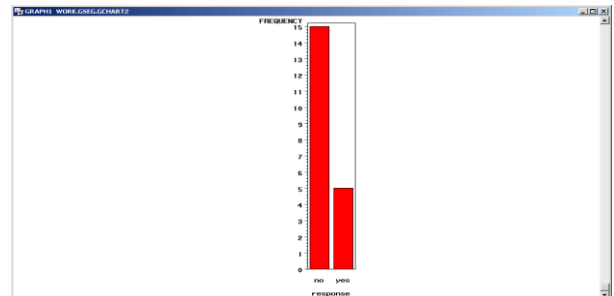
GLOT

```
proc gplot data=views.plot ;  
  plot change*conc=day ;  
run;  
quit;
```



GCHART

```
proc gchart data=views.response ;  
  vbar response / discrete ;  
run;  
quit;
```



As can be seen from these two outputs SAS/GRAPH provides default settings when producing graphics. Often a user has a particular preference as to the settings he/she prefers. Maybe the individual who originally requested the output has stipulated a particular style.

How can we as programmers alter these styles?

We shall introduce all the available options in turn and demonstrate their usage and scope.

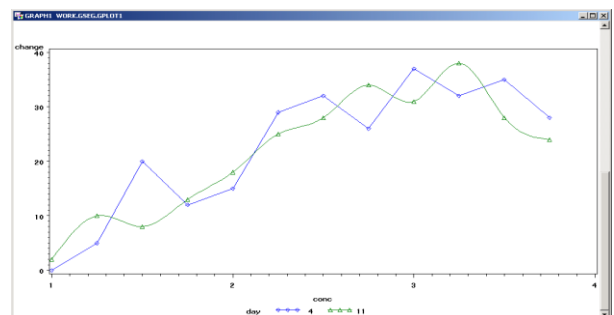
SYMBOL STATEMENTS

When a graph is produced via the SAS System there is a list of default styles that are applied. The styles control aspects such as the colour of the plotted line, the data representation graphic and the line style. The programmer, via the use of symbol statements, can control these aspects and more.

Symbol statements are held in memory and hence all subsequent graphics within that SAS session will use the same settings for the symbol statements.

Symbol statements must be run before the graphic is produced.

```
symbol1 c=blue v=diamond i=join;  
symbol2 c=green v=triangle i=spline;
```



The symbol statements demonstrated above have resulted in the first data series being coloured blue with diamonds as data point representation and the points being joined together via straight lines. The second data series is coloured green within triangles as data point representation and the line being joined with a smoothed line of best fit.

Available data representation graphics are: -

VALUE=	Plot Symbol	VALUE=	Plot Symbol
PLUS	+	% (percent)	⌘
X	×	& (ampersand)	⌘
STAR	*	' (single quote)	⌘
SQUARE	□	= (equals)	☆
DIAMOND	◇	- (hyphen)	⊙
TRIANGLE	△	@ (at)	♀
HASH	#	* (asterisk)	♀
Y	Y	+ (plus)	⊕
Z	Z	> (greater than)	♂
PAW	⋯	. (period)	℥
POINT	.	< (less than)	℥
DOT	●	, (comma)	⊕
CIRCLE	○	/ (slash)	ψ
_ (underscore)	□	? (question mark)	℥
" (double quote)	♠	((left parenthesis)	℥
# (pound sign)	♥	) (right parenthesis)	♂
\$ (dollar sign)	◇	: (colon)	⌘

Source: SAS System Help

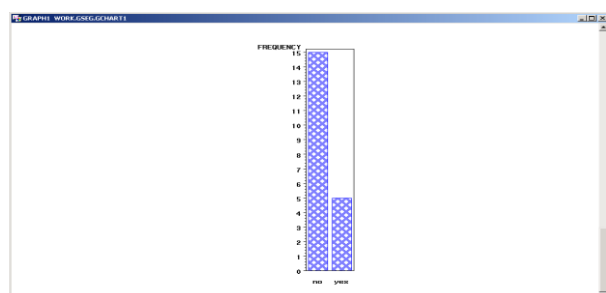
PATTERN STATEMENTS

Pattern statements are stored and used in a similar fashion to the symbol statements within the SAS System.

Whereas symbol statements are generally used within the GPLOT procedure pattern statements are generally used within the GCHART procedure.

Pattern statements can control the colour and representation of the bars within the graphics resulting from a GCHART procedure.

```
pattern1 c=blue v=x4;
```



The pattern statement above has resulted in the bars being displayed in a cross diagonal in blue ink.

Available data representation options are: -

E - empty
S - solid
Li - left diagonal lines
Ri - right diagonal lines
Xi - cross diagonal lines

Where "i" is a value from 1 to 5 representing the intensity of the line.

AXIS STATEMENTS

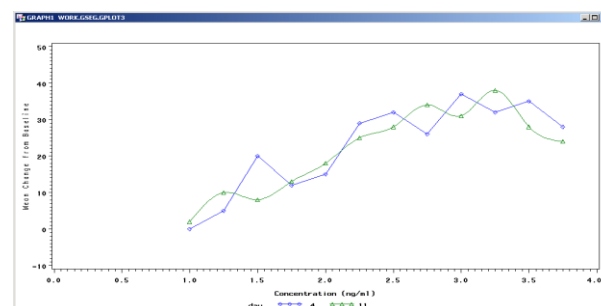
Again, Axis statements are stored in a similar fashion to both Symbol and Pattern statements.

However, in order to use axis statements the individual procedures must request their use on the plot or chart statements.

Axis statements control the appearance of the axis on graphics produced via procedures within the SAS System.

The whole appearance of the axes within the SAS System can be controlled via a suite of axis statements.

```
axis1 order=-10 to 50 by 10
      label=(a=90 'Mean Change from
      Baseline');
axis2 order=0 to 4 by 0.5
      label=(a=0 'Concentration (ng/ml)');
```



The axis statements above control the labels used, the minor and major tick points and the angle of the text.

Further attributes can also be controlled via the axis statements; full information on axis statements, which is to numerous to list here, is available via the SAS System Help.

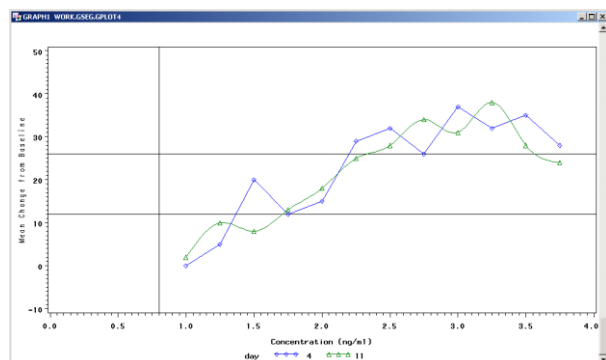
REFERENCE LINES

Reference Lines can only be requested on a procedure-by-procedure basis, these are not held in memory like the statements we have seen so far.

Reference lines are vertical or horizontal solid lines that can be placed on to a graph to highlight certain areas or points of interest.

Upon either the plot or chart statement a reference line can be added as an option as follows:

```
Vref=12 26 ;
Href=0.8;
```



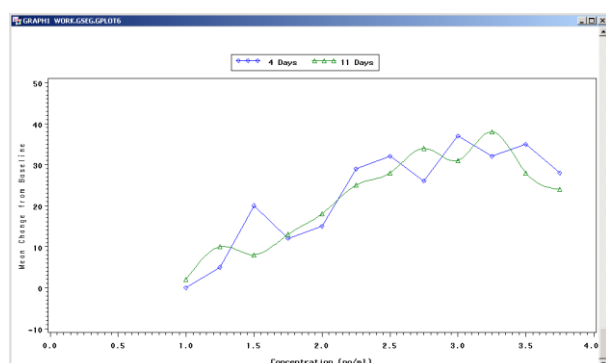
This results in two horizontal (vref) reference lines at the points of 12 and 26 on the vertical axis and one vertical (href) line at the point 0.8 on the horizontal axis.

LEGENDS

By default the SAS System automatically produces a legend to help explain the produced graphic. As with other areas of the SAS System we as programmers are able to alter the default settings and produce our own legends.

Many aspects of the legend can be altered; the main areas altered by the programmer are the labels and location of the default SAS legend.

```
legend1 label=none
value=(t=1 '4 Days' t=2 '11 Days')
frame position=top;
```



The legend statement above removes the default label, specifies that there is to be 2 tick marks with the labels of "4 Days" and "11 Days" respectively. Requests a frame for the legend and re-positions the legend at the top of the graphic area.

GRAPHIC DEVICES

When a SAS graphic is produced the default display is within the graph window. It is possible to produce graphics in a number of different formats depending on the requirements.

The formats available can be split into three sections: -

- Screen
- Printer
- File

The following SAS code will allow users to establish which graphic devices are currently available to them:

```
proc gdevice;
run;
```

This code will display all the available drivers to the programmer.

Alternatively a SAS programmer can look within the SASHELP.DEVICES catalog, which will detail the same information.

Once a device has been identified it can be utilised within the SAS System via a goptions statement.

```
filename outfile 'c:\views2003_example.jpg';
goptions device=imgjpeg gsfname=outfile;
```

The code above will result in the following graphical output being sent to a jpeg file.

ANNOTATING GRAPHICS

When graphics are produced within the SAS System there is no information displayed upon the actual plot area regarding the data values.

Via the use of annotation datasets the programmer can produce graphics containing additional information that will enhance the graphic for users.

Annotation can be added via one of 2 main methods: -

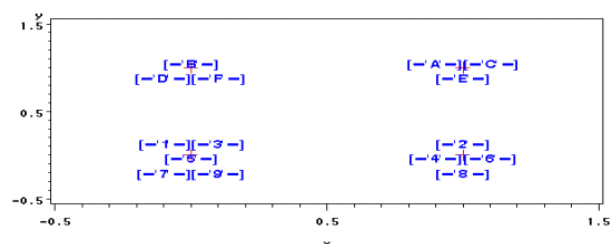
- User created datasets
- Dataset created with the use of the SAS System provided Annotate macros

The annotate dataset, created via either method, is a dataset with certain key variable names that, when interpreted via SAS/GRAPH procedures, will add information to the graphical output.

The key variables that can be used within the annotate area of SAS Graphics are: -

- | | |
|------------|------------|
| - ANGLE | - STYLE |
| - CBORDER | - SUBGROUP |
| - CBOX | - TEXT |
| - COLOR | - WHEN |
| - GROUP | - X |
| - HSYS | - XC |
| - IMGPATH | - XSYS |
| - LINE | - Y |
| - MIDPOINT | - YC |
| - POSITION | - YSYS |
| - ROTATE | - Z |
| - SIZE | - ZSYS |

The positioning variables within the annotation datasets can have a value from 1-9 or A – F. These dictate the location of the written information relative to the plotted value.



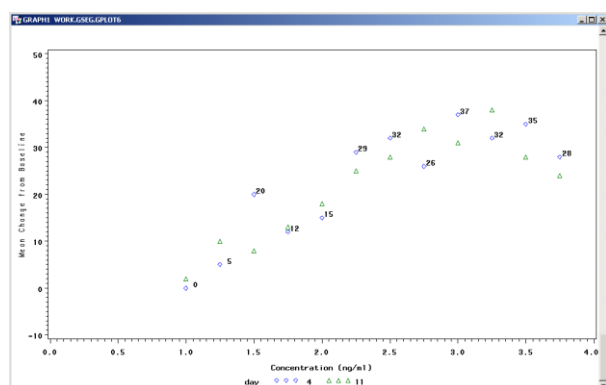
A number of functions can also be utilised within the annotation facility. Namely: -

- | | |
|------------|------------|
| - BAR | - PIECNTR |
| - CNTL2TXT | - PIEXY |
| - COMMENT | - POINT |
| - DEBUG | - POLY |
| - DRAW | - POLYCONT |
| - DRAW2TXT | - POP |
| - FRAME | - PUSH |
| - IMAGE | - SWAP |
| - LABEL | - SYMBOL |
| - MOVE | - TXT2CNTL |
| - PIE | |

Customised annotation datasets are generally based on the dataset that is then subsequently used within the graphic. The example program below demonstrates this.

```
data anno_ds ;
  set views.plot (rename=(conc=x change=y)
                  where=(day=4));
  retain xsys '2' ysys '2' ;
  text=put(y,$5.);
  y=y+1;
run;

proc gplot data=views.plot anno=anno_ds;
  plot change*conc=day / vaxis=axis1
  haxis=axis2;
run;
quit;
```



This program takes the PLOT dataset and creates annotation dataset from it (ANNO_DS). The annotation dataset contains the key variable names of X, Y, XSYS, YSYS and text. This stipulates that the SAS System should right the text (in effect the value of y) at the point (x, y+1) on the plot. The xsys and ysys values indicate that the SAS System should use the real data values when plotting the information (see above for other definitions of xsys and ysys).

It is also possible to use annotation macros that are provided as part of the SAS/GRAPH area of the SAS system. These are: -

- | | |
|-------------|-------------|
| - %ANNOMAC | - %POLY |
| - %BAR | - %POLY2 |
| - %BAR2 | - %POLYCONT |
| - %CIRCLE | - %POP |
| - %CNTL2TXT | - %PUSH |
| - %COMMENT | - %RECT |
| - %DCLANNO | - %SCALE |
| - %DRAW | - %SCALET |
| - %DRAW2TXT | - %SEQUENCE |
| - %FRAME | - %SLICE |
| - %LABEL | - %SWAP |
| - %LINE | - %SYSTEM |
| - %MOVE | - %TXT2CNTL |
| - %PIEXY | |

The annotation macros allow the user to produce the required key variables within the annotation dataset without having to be aware of the variable names. The macros are a pseudo interface to creating the annotation dataset.

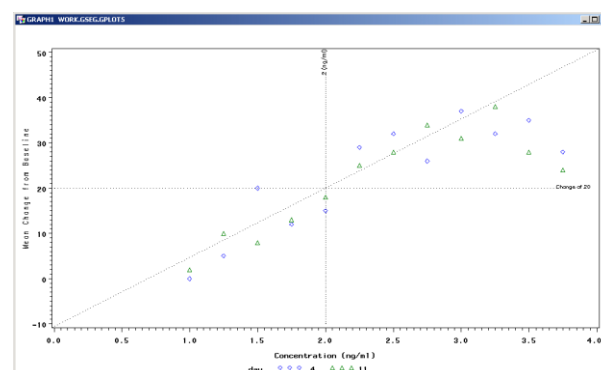
In order to use the annotation macros they must first be compiled, the macros are provided with the SAS/GRAPH software. The macros can be compiled by running the simple code below:

```
%annomac;
```

Once the macros have been compiled they can be utilised within the data step as shown below.

```
data anno_ds ;
  %system(1,1,1);
  %line(0,0,100,100,black,33,0.1);
  * Diagonal Line ;
  %system(2,2,1);
  %label(3.7,20,'Change of
20',black,0,0,2,simplex,C);
  %line(0,20,3.7,20,black,33,0.1);
  * Diagonal Line ;
  %system(2,2,1);
  %label(2,45,'2
(ng/ml)',black,90,0,2,simplex,C);
  %line(2,-10,2,45,black,33,0.1);
  * Diagonal Line ;
run;
```

```
proc gplot data=views.plot anno=anno_ds;
  plot change*conc=day / vaxis=axis1
  haxis=axis2;
run;
quit;
```



The annotation dataset, regardless of the method used, is specified as an option on the procedure statement.

TEMPLATES

When a graphic is produced with the use of the SAS System it is usually displayed in its own window. This can cause a problem when the user wishes to display two or more graphics at the same time.

Once again there are methods available to allow the programmer to do just this. The concept of Templating with the SAS/GRAPH area allows the programmer to pre-define a graphical output area and then insert graphics into the different areas.

As with the area of annotation it is possible to produce your own customised template or use the template provided with the SAS System.

Templates, customised or SAS provided, are simply areas on the graphical output screen in which the programmer requests graphics to be displayed, via the procedure GREPLAY.

Users can use any of the templates from the SASHELP.TEMPLT catalog for which information is shown below: -

Name	Description
H2	1 BOX LEFT, 1 BOX RIGHT
H2s	1 BOX LEFT, 1 BOX RIGHT (WITH SPACE)
H3	3 BOXES ACROSS (HORIZONTALLY)
H3s	3 BOXES ACROSS (WITH SPACE)
H4	4 BOXES ACROSS (HORIZONTALLY)
H4s	4 BOXES ACROSS (WITH SPACE)
L1r2	1 BOX LEFT, 2 BOXES RIGHT
L1r2s	1 BOX LEFT, 2 BOXES RIGHT (WITH SPACE)
L2r1	2 BOXES LEFT, 1 BOX RIGHT
L2r1s	2 BOXES LEFT, 1 BOX RIGHT (WITH SPACE)
L2r2	2 BOXES LEFT, 2 BOXES RIGHT
L2r2s	2 BOXES LEFT, 2 BOXES RIGHT (WITH SPACE)
U1d2	1 BOX UP, 2 BOXES DOWN
U1d2s	1 BOX UP, 2 BOXES DOWN (WITH SPACE)
U2d1	2 BOXES UP, 1 BOX DOWN
U2d1s	2 BOXES UP, 1 BOX DOWN (WITH SPACE)
V2	1 BOX UP, 1 BOX DOWN
V2s	1 BOX UP, 1 BOX DOWN (WITH SPACE)
V3	3 BOXES STACKED VERTICALLY
V3s	3 BOXES STACKED VERTICALLY (WITH SPACE)
Whole	ENTIRE SCREEN TEMPLATE

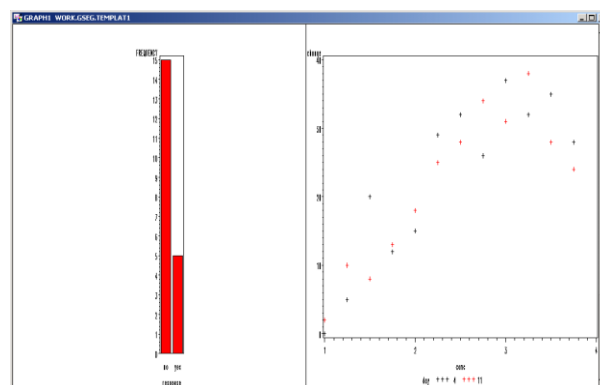
Source: The SAS system catalog SASHELP.TEMPLT

Upon selection of a template the SAS procedure GREPLAY can be used to display graphics within:

```
proc greplay nofs igout=gseg
tc=sashelp.templt;
template h2 ;
treplay 1:2 2:4 ;
quit;
```

The code above requests to use template h2; this contains 2 boxes, 1 on the left and 1 on the right. Within this the programmer has requested that the graphic output that is stored second in the catalog work.gseg (igout value) is displayed within the first pane of the template (1:2). Within the second pane of the template the fourth entry in the catalog work.gseg is displayed (2:4).

This would result in a display similar to: -



If the programmer wishes they can create their own template rather than using when of the pre-defined SAS templates.

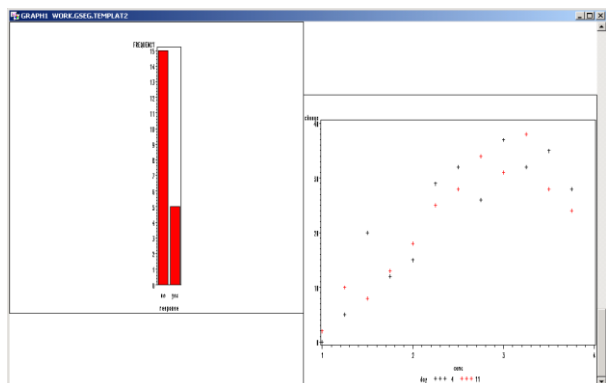
Programmers can again use the procedure GREPLAY to define their own graphical templates.

```
proc greplay nofs igout=gseg tc=views.temps
;
tdef tl_br
1/llx=0 lly=20
ulx=0 uly=100
lrx=50 lry=20
color=black;
tdef tl_br
2/llx=50 lly=0
ulx=50 uly=80
lrx=100 lry=80
lrx=100 lry=0
color=black;
template tl_br ;
treplay 1:2 2:4 ;
run;
quit;
```

Users must provide a template name (e.g.: tl_br) and then for each pane (e.g.: 2) must provide the co-ordinates between 0 and 100 for the following points: -

Llx	Lower Left x
Lly	Lower Left y
Ulx	Upper Left x
Uly	Upper Left Y
Lrx	Lower Right x
Lry	Lower Right Y
Urx	Upper Right X
Ury	Upper Right Y

Again this program then requests that the same two graphics be displayed in the first and second panes.



CONCLUSION

Graphics. Easy to produce; hard to make presentable. Well maybe not. Within this paper we have seen the areas of symbol, pattern and axis statements. Legends, annotation and templating.

All of these statements or areas allow us as SAS programmers to enhance and fully customise our graphical output. The options and combinations available for self-customisation are almost limitless.

Hopefully the world of SAS/GRAPH is no longer such a minefield to you.

Happy Graphing!!

REFERENCES

SAS System Help
SAS System Catalogs

CONTACT INFORMATION

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

Author Name	Simon Williams
Company:	Amadeus Software Limited
Address:	Orchard Farm, Witney Lane, Leafield, Oxon OX29 9PG
Work Phone:	01993 878287
Fax:	01993 878042
Email:	simon.williams@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.