ABSTRACT

This tutorial presents interesting and unusual uses of PROC GREPLAY that go beyond the examples supplied in the SAS/GRAPH® User's Guide, Version 5 Edition. Examples of the following types of applications will be discussed:

* Creative template designs
* Combining multiple graphs
* Effective and unusual presentation of data
* Combining text and graphics
* Zooming and rotating with templates
* Adapting color to different media

This is not meant to be an exhaustive summary of the capabilities of PROC GREPLAY, but rather an illustration of a few of its nonstandard uses.

CREATIVE TEMPLATE DESIGNS

In most PROC GREPLAY applications using templates, template panels are rectangular. However, a panel can take any shape using three or four sides. For example, you can use the following statements to design a template with nonrectangular panels that give the perspective of looking into a box:

```sas
PROC GREPLAY NOFS;
TC MYTEMP;
TDEF BOX 1/LLX=20 LLY=0 ULX=20 ULY=40;
URX=20 URY=90 URX=100 URY=90
URX=100 URY=100 URX=50 URY=80
5/LLX=20 LLY=90 ULX=20 ULY=90;
URX=80 URY=90 URX=80 ULY=40;
```

(Line mode PROC GREPLAY statements are used for illustration in this paper. However, everything that is demonstrated here can also be done in full-screen mode.)

If you take five graphs and replay them inside this template, the output in Figure 1 is obtained:

![Figure 1: Financial Data for XYZ Company](image1)

A second example illustrates the use of non-rectangular panels and clipping to create a template that looks like an open book. Use the following statements to create the template:

```sas
PROC GREPLAY NOFS;
TC TEMPLATE;
TDEF BOOK 1/LLX=25 LLY=20 ULX=25 ULY=94;
URX=50 URY=80 LRX=50 LRY=05 CLIP;
2/LLX=15 LLY=15 ULX=15 ULY=88;
URX=50 URY=80 LRX=50 LRY=05 CLIP;
3/LLX=75 LLY=94 ULX=75 ULY=20 CLIP;
URX=85 URY=88 LRX=85 LRY=15 CLIP;
4/LLX=50 LLY=95 ULX=50 ULY=80;
URX=85 URY=82 LRX=85 LRY=10;
```

If you then replay several graphs in the template, the output in Figure 2 is generated:

![Figure 2: Financial Data for XYZ Company](image2)
Note the use of clipping in the design of the template. This causes some of the pages to "cover" pages in back of them.

COMBINING MULTIPLE GRAPHS

By using templates creatively, you can generate a single graph that is made of several components, each of which is a separate graph. For example, you can combine pie charts and bar charts to produce an interesting and informative effect. First, generate a pie chart, using statements such as these:

PROC GCHART DATA=SALES GOUT=PIEBAR;
PIE DEPT / SUMVAR=SALES DISCRETE
VALUE=NONE CTEXT=BLACK SLICE=NONE
COUTLINE=BLACK NOHEADING;
TITLE1 F=XSWISS H=8 C=BLACK
'ELECTRONICS REVENUE';
TITLE2 F=XSWISS H=8 C=BLACK
'QUARTERLY SALES BY DEPARTMENT';
PATTERN1 C=BLACK V=PIN45;
PATTERN2 C=BLACK V=PIN45;
PATTERN3 C=BLACK V=PIN45;
PATTERN4 C=BLACK V=PIN45;
RUN;

This produces the graph illustrated in Figure 3.

ELECTRONICS REVENUE
QUARTERLY SALES BY DEPARTMENT

Next generate a series of bar charts, using the following code:

PROC GCHART DATA=TV GOUT=PIEBAR;
VBAR QUARTER / SUMVAR=SALES DISCRETE
CAXIS=WHITE RAXIS=AXIS1 MAXIS=AXIS2 FRAME;
AXIS1 VALUE=(H=3 F=XSWISS) LABEL=NONE
ORDER=0 TO 40000 BY 10000;
AXIS2 VALUE=(H=3 F=XSWISS) LABEL=NONE;
TITLE1 F=XSWISS H=8 'TV';
TITLE2 F=XSWISS H=8 '$100,000';
RUN;

These statements produce the graph in Figure 4.

Repeat the bar chart three more times, once for each division in the company. Then create a template that will place the pie chart in the middle of the graph, and one bar chart in each corner. To do this, use the following code:

PROC GREPLAY NOPR;
TC TEMPLATE;
TDEF PBAR
1/LLX= 0 LLY= 0 ULX= 0 ULY=100
URX=100 URY=100 LRX=100 LRY= 0
2/LLX= 0 LLY= 45 ULX= 0 ULY= 79
URX= 34 URY= 79 LRX= 34 LRY= 45
3/LLX= 66 LLY= 45 ULX= 66 ULY= 79
URX=100 URY= 79 LRX=100 LRY= 45
4/LLX= 0 LLY= 0 ULX= 0 ULY= 34
URX= 34 URY= 34 LRX= 34 LRY= 0
5/LLX= 66 LLY= 0 ULX= 66 ULY= 34
URX=100 URY= 34 LRX=100 LRY= 0;

These statements produce the template illustrated in Figure 5.
Then place the graphs in the appropriate panels:

\[ \text{PLOT PIEBAR;} \]
\[ \text{TEMPLATE PBAR;} \]
\[ \text{TRYPLAY 1:1 2:2 3:3 4:4 5:5;} \]

and the graph in Figure 6 is produced.

**ELECTRONICS REVENUE**

**QUARTERLY SALES BY DEPARTMENT**

- **TV** $100,000
- **STEREO** $65,000
- **VCR** $85,000
- **COMPUTER** $89,500

**EFFECTIVE AND UNUSUAL PRESENTATION OF DATA**

An interesting way to represent multiple variables in a single plot is by using a scatterplot matrix. This type of graph, which permits the display of multiple bivariate relationships in a single display, is illustrated below in Figure 7.

To produce this plot, first run the GPLOT procedure for each pair of variables to be analyzed. For example, you can obtain a plot of radiation by temperature with the following statements:

```
PROC GPLOT DATA=Ol GOUT=SCATTER;
PLOT T*R / VAXIS=AXIS1 HAXIS=AXIS2 FRAME;
AXIS1 VALUE=(F=XSWISS H=3) ORDER=60 TO 100 BY 20 MINOR=NONE LABEL=NONE LENGTH=70 PCT OFFSET=(5 PCT,5 PCT);
AXIS2 VALUE=(F=XSWISS H=3) ORDER=0 TO 200 BY 100 MINOR=NONE LABEL=NONE LENGTH=83 PCT OFFSET=(5 PCT,5 PCT);
SYMBOL1 F=XSWISS V='.' H=2 C=BLACK;
RUN;
```

These statements produce the plot in Figure 8.

Next, generate a plot of radiation level by ozone level. This plot does not have tick marks or labels on the axes:

```
PROC GPLOT DATA=D1 DATA=SCATTER;
PLOT R*O / VAXIS=AXIS1 HAXIS=AXIS2 FRAME;
AXIS1 MAJOR=NONE MINOR=NONE LABEL=NONE VALUE=NONE LENGTH=70 PCT;
AXIS2 MAJOR=NONE MINOR=NONE LABEL=NONE VALUE=NONE LENGTH=83 PCT;
SYMBOL1 F=XSWISS V='.' H=2 C=BLACK;
```

This produces the plot in Figure 9.

Finally, create a "plot" containing the variable "names" for the 1st, 5th, and 9th panels. To produce one of these, run a PROC GPLOT with no marker symbols, and use the ANNOTATE option to place the words "RADIATION," "OZONE," and "TEMPERATURE" in the appropriate plot areas. For example,

```
DATA ANNO;
FUNCTION='LABEL'; COLOR='BLACK'; STYLE='XSWISS'; XSYS='1'; YSYS='1'; HSYS='1'; X=50; Y=50; SIZE=22;
POSITION='5'; TEXT='RADIATION';
RUN;
```

```
PROC GPLOT DATA=D1 DATA=ANNO GOUT=SCATTER;
PLOT R*R / VAXIS=AXIS1 HAXIS=AXIS2 FRAME;
AXIS1 VALUE=(F=XSWISS H=3) ORDER=0 TO 200 BY 100 MINOR=NONE LABEL=NONE VALUE=NONE LENGTH=83 PCT OFFSET=(5 PCT,5 PCT);
AXIS2 MAJOR=NONE MINOR=NONE;
RUN;
```

987
This produces the plot in Figure 10.

Next, create a template containing nine boxes as follows:

```
PROC GREPLAY NOPR;  
TO TEMPLATE;   
TDEF SCAT  
1/LLX= 0 LLY= 60 ULX= 0 ULY=100  
2/LLX= 32 LLY= 60 ULX= 32 ULY=100  
3/LLX= 65 LLY= 60 ULX= 65 ULY=100  
4/LLX= 0 LLY= 32 ULX= 0 ULY= 68  
5/LLX= 32 LLY= 32 ULX= 32 ULY= 68  
6/LLX= 65 LLY= 32 ULX= 65 ULY= 68  
7/LLX= 100 LLY= 68 ULX=100 ULY= 68  
8/LLX= 32 LLY= 38 ULX= 32 ULY= 38  
9/LLX= 65 LLY= 0 ULX= 65 ULY= 0  
URX= 100 URY= 100 LRY= 60  
LLY= 60 URY=100 LRY= 60  
ULX= 0 LRX= 34 ULX= 32 LRX= 67  
ULX= 65 LRX= 100 ULY=100 URY=100  
ULY=100 LRY= 60  
```

Finally, after placing the nine graphs in the template, the graph in Figure 11 is generated.

Assuming that the output graphics device contains 28 rows and 80 columns, you will want each text panel to contain 14 rows and 40 columns of text. To do this, the input text file to PROC GPRINT should have a maximum line width of 40 and a 'page eject' carriage-control character in column 1 of every 14th line. When PROC GPRINT encounters the carriage-control character, it will begin a new graph. Once the text file is created, assign the file to the appropriate fileref (FT20FOOl in this example), and run PROC GPRINT as follows:

```
GOPTIONS HPOS=40 VPOS=14;  
PROC GPRINT DDNAME=FT20FOOl GOUT=TXTGRAF;   
```

Assuming that the raw input file to PROC GPRINT contains 42 lines with a carriage-control character every 14 lines, the procedure places three graphs in the output catalog. Figure 13 illustrates one of the possible graphs.

This example demonstrates the ability to combine text and graphics on a single page using PROC GREPLAY. The first step is to create a text file. If the text file is to contain output from a non-graphics procedure, you can use PROC PRINTTO to generate the file. You can then go in and add text to the file if you wish.

In this example we have edited the file so that there is a carriage control character every 14th line. The result
The next step is to generate the 'graphics' part of the output. In this case, you can use PROC GCHART to generate a bar chart in the output catalog:

```
PROC GCHART DATA=DD GOUT=TXTGRAF;
VBAR YEAR / SUMVAR=SALES NOLEGEND DISCRETE;
TITLE 'SALES BY YEAR';
```

Then replay the three PROC GPRINT graphs along with the bar chart in the template that was defined above. The resulting graph is shown in Figure 14.

```
DATA US;
SET MAPS.US;
IF STATE=2 OR STATE=15 THEN DELETE;
PROC GMAP DATA=US MAP=US GOUT=ZOOMAP;
ID STATE;
CHORD STATE=10 LEGEND LEVELS=1;
PATTERN1 V=E C=BLACK;
RUN;
```

Next, define a template such that the 0-100 range of X and Y coordinates is in the lower right corner. When the map is displayed in this template, the only part displayed on the screen or page will be the lower right corner, or the coordinates in the 0-100 range. This is illustrated in Figure 16.

```
PROC GREPLAY NOFS;
TC TEMPLATE;
TDEF ZOOM
(-100,200) (100,200)
(-100,0) (0,0)
ULX= 100 URY=200 ULX= 100 URY=200
URX= 100 URY=200 LRX= 100 LRY=- 100;
```

Then display the map in the template with the following statements:

```
TEMPLATE ZOOM;
GOUT ZOOMAP;
TREPLAY 1:1;
```

and obtain a close-up of the southeastern part of the country, as illustrated in Figure 17.

The box in the lower right corner represents the area that is displayed on the graphics device. The larger rectangle represents the entire area taken up by the picture. If the template is defined with the coordinates of the larger rectangle, the graph itself will take up that area, but the only part displayed on the screen or page is that inside the smaller box.

As a specific example of using the zooming feature, assume you want to produce a map of the continental United States and then zoom in on the southeastern part of the country. To do this, first generate a full size map:

```
DATA US;
SET MAPS.US;
IF STATE=2 OR STATE=15 THEN DELETE;
PROC GMAP DATA=US MAP=US GOUT=ZOOMAP;
ID STATE;
CHORD STATE=10 LEGEND LEVELS=1;
PATTERN1 V=E C=BLACK;
RUN;
```

By defining template panel coordinates outside the range of 0-100, you can use PROC GREPLAY to "zoom in" on portions of a graph. When you view a graph with PROC GREPLAY, the screen or page is the coordinate area between 0 and 100. If you define a panel that is bounded by the coordinates (0,0), (0,50), (50,50), and (50,0) and display a graph in it, you are still viewing the area between 0 and 100 on the screen or page, but the graph itself is displayed in the area bounded by the coordinates above. However, if a panel is defined with coordinates below 0 and above 100, the graph will take up the entire panel, but the part you see will be between 0 and 100 in each direction. This is illustrated in Figure 15.
When templates are rotated, the graphs inside them rotate as well. This means that you can use templates to rotate graphs, even on devices that do not support the ROTATE graphics option. To illustrate this, first generate two graphs as follows:

```sas
PROC GSLIDE GOUT=ROT;
  TITLE1 H=30 PCT ' UNROTATED ';
  TITLE2 J=C F=XSWISS H=25 PCT ' UNROTATED ';
RUN;
PROC GSLIDE GOUT=ROT;
  TITLE1 H=30 PCT ' UNROTATED ';
  TITLE2 J=C F=XSWISS H=25 PCT ' ROTATED ';
RUN;
```

Next define a template with a single panel on the left side:

```sas
PROC GREPLAY NOFS;
  TC TEMPLATE,
    TDEF ROTATE
    L/LX= 0 LLY=25 ULX= 0 ULY=75
    URX=50 URY=75 LRX=50 LRY=25:
NOW display the first graph in that panel to illustrate the unrotated orientation:

```sas
IGOUT ROT;
  TEMPLATE ROTATE;
  TREPLAY 1:1 2:2;
```

These statements produce the graph in Figure 18.

When templates are rotated, the graphs inside them rotate as well. This means that you can use templates to rotate graphs, even on devices that do not support the ROTATE graphics option. To illustrate this, first generate two graphs as follows:

```sas
PROC GSLIDE GOUT=ROT;
  TITLE1 H=30 PCT ' UNROTATED ';
  TITLE2 J=C F=XSWISS H=25 PCT ' UNROTATED ';
RUN;
PROC GSLIDE GOUT=ROT;
  TITLE1 H=30 PCT ' UNROTATED ';
  TITLE2 J=C F=XSWISS H=25 PCT ' ROTATED ';
RUN;
```

Next define a template with a single panel on the left side:

```sas
PROC GREPLAY NOFS;
  TC TEMPLATE;
  TDEF ROTATE
  L/LX= 0 LLY=25 ULX= 0 ULY=75
  URX=50 URY=75 LRX=50 LRY=25:
NOW display the first graph in that panel to illustrate the unrotated orientation:

```sas
IGOUT ROT;
  TEMPLATE ROTATE;
  TREPLAY 1:1 2:2;
```

Now display both graphs in the template using the following statements:

```sas
TEMPLATE ROTATE;
  TREPLAY 1:1;
```

and note that the second graph (Figure 19) has been rotated 90 degrees.

**Unrotated**

**Rotated**

### Adapting Color to Different Media

Often you will want to display the same graph on different types of devices, and tailor the colors of the graph to the characteristics of the device. For example, if you wanted to generate a graph in both slide and hard copy form, you would normally want to use bright colors for the slide and darker colors for the hard copy. You can take advantage of both the color naming conventions and color mapping capabilities to make this task easier. To do this, generate a graph, and instead of using "real" color names, use names such as COLOR1, COLOR2, and so on. For example,

```sas
PROC GSLIDE GOUT=COLORS;
  TITLE1 C=COLOR1 'THIS IS COLOR 1';
  TITLE2 C=COLOR2 'THIS IS COLOR 2';
  TITLE3 C=COLOR3 'THIS IS COLOR 3';
  TITLE4 C=COLOR4 'THIS IS COLOR 4';
RUN;
```

Note that you can use any legal SAS name as a color name, and those names are stored with the picture. Actual color names are resolved when the graph is replayed. If you want to send the graph to a slide maker, you can define a color map to be used with PROC GREPLAY as follows:

```sas
PROC GREPLAY NOFS;
  DEVICE Ui200035:
  IGOUT COLORS
    CC COLORCAT;
  CDEF SLIDE
    1/COLOR1:WHITE
    2/COLOR2:YELLOW
    3/COLOR3:CYAN
    4/COLOR4:PINK;
  CMAP SLIDE;
  REPLAY 1;
/* SLIDE MAKER */
```

Use of this color map causes the colors COLOR1, COLOR2, COLOR3, and COLOR4 to be mapped to bright colors before the picture is sent to the slide maker. If you want to send the picture to
a plotter, you can define a color map with colors more appropriate to that medium:

CDEF PLOTTER
1/COLOR1:BLACK
2/COLOR2:RED
3/COLOR3:GREEN
4/COLOR4:BLUE;
DEVICE ZETA887; /* PLOTTER */
CMAP PLOTTER;
REPLAY 1;

You can also use color maps to create palettes for desired effects. For example you can define a color map that generates various shades of green and yellow:

CDEF GY
1/COLOR1:LIOL /* Light Olive */
2/COLOR2:MOGY /* Moderate Greenish Yellow */
3/COLOR3:LGY /* Light Greenish Yellow */
4/COLOR4:LITY /* Light Yellow */

CONCLUSION

This tutorial has illustrated only a few of the many creative approaches that can be used with PROC GREPLAY to generate new and interesting types of graphics output. Many other possibilities are available, limited only by your imagination.

SAS/GRAPH is a registered trademark of SAS Institute Inc., Cary, NC, USA. A footnote should accompany the first use of each registered trademark or trademark and should state that the referenced trademark is used to identify products or services of SAS Institute Inc.