Producing Publication Quality Business Graphs with SAS/GRAPH® Software
Andrew Kopras, Parliamentary Library, Canberra A.C.T., Australia

INTRODUCTION

Modern colour graphic output devices like ink jet or colour wax transfer printers allow us to produce high quality full colour output unheard of at the time when pen plotters were at the pinnacle of graphic technology.

Part of our service at the Parliamentary Library's Statistics Group is to provide our clients (Senators and Members of Australian Parliament) with graphic output of the required statistical data. In some cases, these graphs can be quite simple but on many occasions higher quality graphics are required, especially when these are used during presentations, meetings and the like. In producing these graphs, we have devised several methods of making them more presentable. Some of these methods are discussed in this paper, in particular -

- Producing fully shaded 3 dimensional bar graphs.
- Producing 3 dimensional pie charts.
- Use of coloured backgrounds and in particular, shaded backgrounds.
- Selecting most appropriate colour schemes with user-defined colours.
- Using GREPLAY templates to produce some interesting output.

FUNCTIONS OF THE PARLIAMENTARY LIBRARY

The primary function of the Department of the Parliamentary Library is to make available to Senators and Members, upon request or in anticipation of general interest, information and research materials and services.

The Parliamentary Library is divided into two main parts:

- Library, Reference and Information Service which deals with normal library functions as well as -
  - General Reference
  - Subject Reference
  - Current Information Service
  - Media monitoring, etc.
- Parliamentary Research Service which provides Senators and Members with more complex research on a variety of subjects. The Research Service is divided into seven specialist groups -
  - Defence
  - Economics and Commerce
  - Education and Welfare
  - Foreign Affairs
  - Law and Government
  - Science, Technology and Environment
  - Statistics

The function of the Statistics Group is to provide up-to-date statistics on all subjects and provide advice on the validity and use of statistics generally.

LIBRARY'S USE OF THE SAS® SYSTEM

Contrary to what may be expected, SAS is not used in the Parliamentary Library to catalogue books. It's use is solely limited to the Research Services's Statistics Group. Some of the things that we use the SAS system for are -

- Simple and complex statistical data analysis.
- Choropleth mapping, especially of Electoral Divisions.
- Producing text charts for presentations, etc.
- Business graphics.

3 DIMENSIONAL BAR CHARTS

In SAS Color Graphics II course there is an example how the annotate facility can be used to add a 3 dimensional effect to a bar graph. This is done by using the MOVE and DRAW functions to draw a 3D extension behind each bar. The height of each bar is read in from the data for the chart and the width of the bars must be fixed in the VBAR statement. The SAS code is as follows -

DATA BARS;
  LENGTH FUNCTION COLOR STYLE $ 8;
  SET HSMEDIAN;
  WHEN='A';
  XSYS='2'; YSYS='2';
  X=YEAR; Y=PRICE;
  FUNCTION='MOVE'; OUTPUT;
  XSYS='C'; YSYS='C';
  X=-1; Y=0; FUNCTION='MOVE'; OUTPUT;
  X=+1.25; Y=0.5; FUNCTION='DRAW'; OUTPUT;
  X=2; Y=0; OUTPUT;
  X=-1.25; Y=-0.5; OUTPUT;
  X=1.25; Y=0.5; FUNCTION='MOVE'; OUTPUT;
  X=0; YSYS='1'; Y=0; FUNCTION='DRAW';
  OUTPUT;
PROC GCHART DATA=HSMEDIAN;
  VBAR YEAR / SUMVAR=PRICE DISCRETE WIDTH=2
  SPACE=2.2 ANNOTATE=BARS;
An output generated by the above code is shown in Example 1. Notice that the width of the bars is set at 2 window cells and the code for the annotate data set is based on this width.

We had decided that this method can be improved. We drew the annotated 3D additions to the bars using functions POLY and POLYCONT. This was made somewhat more difficult by the fact that POLY and POLYCONT do not update the XLAST and YLAST variables. However after many tries we finally succeeded. In addition we had improved the code by putting the width of the bar into a variable, so that if we wish to vary the width, the code does not have to be altered. The code we arrived at was as follows:

```plaintext
DATA BARS;
  LENGTH FUNCTION COLOR STYLE $ 8;
  RETAIN XBAR 2 YINC 1.75 YHI 210 YLO 0;
  DROP YINC XBAR YREL XREL YHI YLO;
  SET HSMEDIAN;
  LINE=1;
  WHEN='A';
  XSYS='2'; YSYS='2';
  X=YEAR;
  Y=PRICE;
  FUNCTION='MOVE'; OUTPUT;
  XSYS='C'; YSYS='7';
  STYLE='EMPTY'; COLOR='WHITE';
  XREL=-(XBAR/2); YREL=0;
  X=XREL; Y=YREL;
  FUNCTION='POLY'; OUTPUT;
  XREL=XREL+(1.25*XBAR/2); YREL=YREL+YINC;
  X=XREL; Y=YREL;
  FUNCTION='POLYCONT';
  COLOR='BLACK'; OUTPUT;
  XREL=XREL+(2*XBAR/2); YREL=YREL+0;
  X=XREL; Y=YREL;
  X=YEAR; Y=0;
  FUNCTION='MOVE'; OUTPUT;
  XSYS='C'; YSYS='7';
  STYLE='EMPTY'; COLOR='WHITE';
  XREL=2.25*(XBAR/2); YREL=YINC;
  X=XREL; Y=YREL;
  FUNCTION='POLY'; OUTPUT;
  XREL=XREL+(1.25*XBAR/2); YREL=YREL+YINC;
  X=XREL; Y=YREL;
  FUNCTION='POLYCONT';
  COLOR='BLACK'; OUTPUT;
  XREL=XREL+(2*XBAR/2); YREL=YREL-0;
  X=XREL; Y=YREL;
  X=YEAR; Y=0;
  FUNCTION='MOVE'; OUTPUT;
  XSYS='C'; YSYS='7';
  STYLE='EMPTY'; COLOR='WHITE';
  XREL=XREL+(1.25*XBAR/2); YREL=YREL+YINC;
  X=XREL; Y=YREL;
  FUNCTION='POLY'; OUTPUT;
  XREL=XREL+(1.25*XBAR/2); YREL=YREL+0;
  X=XREL; Y=YREL;
  X=YEAR; Y=0;
  FUNCTION='MOVE'; OUTPUT;
  XSYS='C'; YSYS='7';
  STYLE='EMPTY'; COLOR='WHITE';
  XREL=XREL+(1.25*XBAR/2); YREL=YREL+YINC;
  X=XREL; Y=YREL;
  FUNCTION='POLY'; OUTPUT;
  XREL=XREL+(1.25*XBAR/2); YREL=YREL+YINC;
  X=XREL; Y=YREL;
  PROC GCHART DATA=HSMEDIAN;
  VBAR YEAR / SUMVAR=PRICE DISCRETE WIDTH=2
  SPACE=2.2
  AXIS=O TO 210 BY 30
  COULINE=BLACK;
  ANNOTATE=BARS;
```

In addition to specifying the width of the bars, this method also requires the range of the Y-axis (ie. 0 TO 210) to be fixed. The output from the above is shown in Example 2. At first sight, the only difference is the improved handling of the bottom of the bars. One may wonder, why bother? However, the annotated 3D extensions on the back of the bars are now polygons! As such they can be filled with patterns, solid colours or shades of gray. In fact there are two polygons behind each bar so each may be filled with a different colour (or a different shade of gray for black and white publishing). A shades of gray annotated 3D bar output is shown in Example 3.

3 DIMENSIONAL PIE CHARTS

Those familiar with the annotate facility will know that the following code will draw one segment of a pie chart:

```plaintext
DATA ANNDAT;
  XSYS='3'; YSYS='3';
  STYLE='SOLID'; FUNCTION='PIE';
  SIZE=30; X=55; Y=48;
  ANGLE=32; ROTATE=50;
  COLOR='GRAYSA';
  OUTPUT;
PROC GANNO ANNO=ANNDAT;
```

If we are using a raster device (eg. ink jet, colour wax or laser printer) and we draw the same segment say 20 times, each time increasing the Y value slightly, the end result is a 3 dimensional looking pie slice. Top it off, we draw the last slice in a different colour. The code for such a 3D slice is not much more than that for a simple slice, we just add the bolded code:

```plaintext
DATA ANNDAT;
  XSYS='3'; YSYS='3';
  STYLE='SOLID'; FUNCTION='PIE';
  SIZE=30; X=55; Y=48;
  ANGLE=32; ROTATE=50;
  COLOR='GRAY79';
  DO I=1 TO 20;
    OUTPUT;
    Y=Y + 0.5;
  END;
  COLOR='GRAYSA';
  OUTPUT;
PROC GANNO ANNO=ANNDAT;
```

Using this technique, we can now draw a complete pie chart in 3D. The only thing we have to be wary of is to start drawing the back slices first otherwise the pie will not look right. A complete pie chart drawn using this technique is shown in Example 4.

CHART BACKGROUNDS

Adding a coloured background to a chart can greatly improve it's appearance. These can be of different complexity, ranging from a simple solid colour behind the chart, to a number of different multicoloured backgrounds, behind different areas of the chart.

Placing a simple coloured background behind a chart is a simple procedure. You simply add the following GOPTIONS statement to your code:

```plaintext
GOPTIONS CBACK=CYAN;
```

This method is good for placing backgrounds behind simple graphs like pie or text charts. For line or bar charts, we
considered a different colour in the data area contrasting with the background for the rest of the chart to be even better. This can be done by the use of the CFRAME statement inside the PLOT, VBAR etc. commands, for example:

GOPTIONS CBACK=GREEN;

PROC GCHART DATA=HOUSES;
   VBAR YEAR/SUMVAR=PRICE
   CFRAME=CYAN;

The best results, however, were achieved by the use of shaded colour backgrounds. We did this with the help of the user-defined colours facility in SAS/GRAPH. The colour definition system we used was the Hue/Lightness/Saturation (HLS) method. The colour is specified in the format Hhhhiiss where hhh is the hue of the required colour expressed in hexadecimal notation ranging from 0 (Hex 000) to 360 (Hex 168) and ii and ss are the values for the required colour’s Lightness and Saturation respectively, ranging from 0 to 255 (Hex FF). We then used the annotate facility to draw 50 solid colour bars across the page, keeping the hue and saturation of the bar’s colour constant and varying the lightness slightly for each bar. The SAS code to place such a background behind the whole chart is:

DATA ANNO;
   LENGTH COLOR $8.
   SYSVAR='3': STHUE=304: INCR=1;
   HUE=STHUE; WHEN='B'; LITE=107;
   DO 1=1 TO 50;
      XSYS=SYSVAR; YSYS=SYSVAR;
      FUNCTION='BAR';
      X=100; Y=(50 - I) * la0/ 50;
   OUTPUT;
   LINE=0;
      FUNCTION='BAR';
      X=100; Y=(50 - I) * 100/ 50;
   OUTPUT;
END;

The same code can be used to draw a shaded background behind the data box by changing the value of the SYSVAR variable to ‘1’. The direction of the shade can be changed from darkest/lightest to lightest/darkest by changing the value of the INCR variable to -1. Thus an excellent result can be achieved for say a bar chart by placing a darkest/lightest background behind the complete chart and then a lightest/darkest background behind the data box.

Equally attractive background can be achieved by keeping the colour’s lightness and saturation constant and varying the hue down the page. With this method, it is handy to determine how the output device we are using is displaying different hues, especially how they blend together when we vary the hue component of our HLS colour definition. We did this by borrowing an example from an excellent paper on the annotate facility given at the SUGI’12 Conference. Basically, this example draws a pie chart colouring each subsequent 2 degree slice with a hue 2 units higher than that for the previous slice. The code was as follows:

DATA PIES;
   LENGTH FUNCTION COLOR STYLE $ 8;
   RETAIN XSYS YSYS '2' HSYS '3'
   STYLE 'SOLID' X Y SIZE 45
   DO A=0 TO 358 BY 2;
      COLOR='H' H PUT(A,HEX3.) H '84FF';
   OUTPUT;
END;

PROC GANNQ ANNO=PIES;

From the resulting pie it is easy to select a suitable 50 degree range of hues for our background. The values for the hue codes can be determined by specifying the angle at which the colour is found (ie blue at 0 degrees, magenta at 60, red at 120, yellow at 180, green at 240 and cyan at 300).

SPECIAL EFFECTS WITH GREPLAY TEMPLATES

On many occasions we needed to put more than one graph on the page. This is easily done with the PROC GREPLAY facility. First we produce each required graphs separately and store them in a GREPLAY catalogue. We then set up with the template facility the position of each graph on the output page. Finally we replay the stored graphs according to these templates.

More complicated procedure is required if we wish to have a major title and/or footnotes on our page. What we did was to produce a title graph containing the required titles and a footnote and store it in our GREPLAY catalogue. We then set up an additional template which consists of the entire page. So if for example we had previously stored graphs named BAR, LINE, and PIE and a title graph named HEAD in our GREPLAY catalogue, we can construct our output with this code:

PROC GREPLAY NOFS TC=SUGA90.TEMPLATE ;
   TDEF SHOW DES='LINE, BAR & PIE' /
      1 / LLX=1 LLY=3 ULX=75 ULY=75 URX=50
         URY=75 LRX=50 LRY=3
      2 / LLX=51 LLY=51 ULX=51 ULY=95 URX=99
         URY=95 LRX=99 LRY=51
      3 / LLX=51 LLY=3 ULX=51 ULY=49 URX=99
         URY=49 LRX=99 LRY=3
      4 / LLX=0 LLY=0 ULX=0 ULY=100 URX=100
         URY=100 LRX=100 LRY=0 ;
   IGOUT SUGA90.REPLAY ;
   TC SUGA90.TEMPLATE ;
   TEMPLATE SHOW ;
   TPLAY 1:BAR 2:LINE 3:PIE 4:HEAD ;

The output from the above is shown in Example 5. Coincidently, we had entered a colour version of this graph in a graphics competition held in conjunction with SUGI90, under the “Best presentation of data” category, and we have been awarded second prize.

Little known feature of the GREPLAY facility is that the replayed graphs can be at different angles to each other. For example an
output page can be constructed in such a way that when folded in four it becomes a greeting card. What we did was to create three graphs, one each for the front, back and the middle of our card and then used the following code to produce our card:

```plaintext
PROC GREPLAY NOPS TC=SUGA90.TEMPLATE;
TDEF CARDLAND
DES='Complete card in landscape'
 1 / LLX=100 LLY=50 ULX=50.5 ULY=50 URX=50.5
    URY=100 LRX=100 LRY=100
 2 / LLX=100 LLY=0 ULX=50.5 ULY=0 URX=50.5
    URY=49.5 LRX=100 LRY=49.5
 3 / LLX=0 LLY=100 ULX=49.5 ULY=100 URX=49.5
    URY=0 LRX=0 LRY=0
IGOUT SUGA90.REPLAY;
TC SUGA90.TEMPLATE;
TEMPLATE CARDLAND;
TPLAY 1:FRONT 2:BACK 3:INSIDE;
```

The resulting card is shown in Example 6.

Colour wax transfer printers offer special media which can be used to iron on images on to T-shirts, etc. The problem is that the output has to be printed on to this media as a mirror-image of itself. We can however construct a template that can do this for us:

```plaintext
PROC GREPLAY NOPS TC=SUGA90.TEMPLATE;
TDEF REVERSE DES='Mirror-image template'
 1 / LLX=100 LLY=0 ULX=100 ULY=100 URX=0
    URY=100 LRX=0 LRY=0
IGOUT SUGA90.REPLAY;
TC SUGA90.TEMPLATE;
TEMPLATE REVERSE;
TPLAY 1:TSHIRT;
```

This code can display any graph stored in a GREPLAY catalogue as a mirror-image of itself.

**CONCLUSION**

SAS/GRAPH contains many features which allow us to create very high quality output. All that is needed is a little extra effort. The additional SAS code that is needed may seem cumbersome at first, but it must be remembered that once written the same code may be reused many times with just minimal changes. Judging from the response we have had from our clients and others to our graphs, we feel that the extra effort was worthwhile.

The views expressed in this paper are those of the author and do not necessarily reflect those of the Parliamentary Research Service, or the Department of the Parliamentary Library.

SAS and SAS/GRAPH are the registered trademarks of SAS Institute Inc, Cary, NC, USA.
EXAMPLE 1

MEDIAN PRICE FOR ESTABLISHED DWELLINGS – SYDNEY

EXAMPLE 2

MEDIAN PRICE FOR ESTABLISHED DWELLINGS – SYDNEY
EXAMPLE 3

**MEDIAN PRICE FOR ESTABLISHED DWELLINGS – SYDNEY**

<table>
<thead>
<tr>
<th>Year</th>
<th>$000s</th>
</tr>
</thead>
<tbody>
<tr>
<td>79</td>
<td>0</td>
</tr>
<tr>
<td>80</td>
<td>30</td>
</tr>
<tr>
<td>81</td>
<td>60</td>
</tr>
<tr>
<td>82</td>
<td>90</td>
</tr>
<tr>
<td>83</td>
<td>120</td>
</tr>
<tr>
<td>84</td>
<td>150</td>
</tr>
<tr>
<td>85</td>
<td>180</td>
</tr>
<tr>
<td>86</td>
<td>210</td>
</tr>
</tbody>
</table>

EXAMPLE 4

**Australian Exports by Destination, 1988 – 89**
HOUSE PRICES
Sydney: more expensive, more volatile.

REAL PRICE CHANGES
Percentage Change in Real Price Since Same Month of Previous Year

EXAMPLE 6

From all of us

Merry Christmas
and a Happy New Year