Introduction

SAS/GRAPH can be a challenge when you want to use it for producing charts and plots for everyday use. This hands-on workshop is for users who want to produce high-quality presentation graphics without a lot of time and energy spent on learning the entire SAS System for Graphics.

You will learn how to set up the graphics environment, create basic charts and plots, enhance them, and manage the graphics output. Using examples provided, participants will learn how to use SAS/GRAPH, including syntax conventions for using SAS on MVS, CMS, UNIX, OS/2, Windows, and DOS. Although most examples will use LIBNAME's and FILENAME's for the PC environments, all other examples will be host independent.

Scope of the Workshop

We will work through two examples of taking data from a conceptual, "raw" form to final form. We will also show an example of taking a graphic produced with SAS and saving it in a form that can then be used with other software packages.

SAS/Graph Software Under Various Operating Systems

This workshop was designed using SAS for Windows 3.1. Most of the conventions described in the lessons below, will apply to versions of SAS at version 6.04 and higher.
Overview of the Capabilities of SAS/GRAPH

The SAS System for Graphics (specifically, SAS/GRAPH) contains numerous procedures allowing for a wide range of graphics capabilities within SAS Software. New users of SAS/GRAPH often find that taking output from SAS procedures and manually creating graphs and charts with other software packages is easier than spending the time to learn SAS/GRAPH. Advantages of SAS/GRAPH include the ability to process large amounts of graphics, programmatically. Other benefits include the ability to take output from a SAS DATA step and SAS PROC steps and mold data into a fine, presentation-quality graphic.

Even though SAS/GRAPH can be intimidating at first, the objective is to show first-time users that they can produce some great looking graphics by learning just a few essential components.

To take some of the mysteries out of SAS/GRAPH, let's divide the task into a few component parts:

Understanding the Graphics Environment

1. Device Management - How to output your graph to your monitor or printer
2. GOPTIONS and other G-Statements

Generating Simple Charts and Graphs

1. Producing Bar and Pie Charts
2. Producing Scatter Plots

Enhancing your Output

1. Enhancing Graphics Output - How to change or enhance individual graphs
2. Global Options - How to specify options that affect how all of your graphics output will look

Managing Your Output

1. Output Management - How to store your SAS graphics and get them into other software packages
Understanding the Graphics Environment:

Initial questions that arise for first-time users of SAS/GRAPH often involve the options specified on the GOPTIONS statement. The GOPTIONS statement basically identifies things about your environment that can enhance the appearance of your graphic as well as controlling some basic environment issues. It handles where you want your graphic to be displayed (DEVICE= statement), things that control the overall appearance of the graph and the attributes specific to the graphic itself.

Device Management

Graphics devices are a necessary evil in SAS/GRAPH. The devices are important because they communicate to SAS which type of graphics output we would like SAS/GRAPH to produce. Perhaps one of the more common questions that arise is How do I know what device I need? A device simply means where do you want this graphic to go: a monitor, a printer, a camera, a plotter, etc. Once you know where you want the graph/chart/plot to go, then you can reference the appropriate device driver on the GOPTIONS statement. The GOPTIONS statement is really what is meant by your graphics environment. The advantage to using SAS for Windows for the Hands-On Session here at SUGI is the fact that we can accept many of the defaults that SAS gives us. For example, if we don't specify a DEVICE= on our GOPTIONS statement, the default is to display the graph on the monitor (DEVICE=WIN for Windows 3.1, OS2 for OS/2, VGA for most DOS environments, TEK4010 or TEK4014 for VT100 connections to 3270 mainframe environments).

To list all of the device drivers available to the version of SAS you are running, use the GDEVICE procedure. In interactive mode submit the following program:

```
Proc Gdevice;
run;
```

If you are not using SAS in interactive mode (Display Manager), use the following program to list all of the possible device drivers in your SAS log.

```
Proc Gdevice nofs;
   list _all_;
run;
```
GOPTIONS and other G-statements

Although there are numerous options that can be specified on the GOPTIONS statement, don't feel overwhelmed. Although these represent my own idiosyncracies (a special thanks to Ken Hardy for his input on these as well!), I have included some of the options that handle about 90% of my graphics needs. If you really want find out all of the options that you have control over use the following program:

```sas
Proc Goptions symbol pattern title nolog; run;
```

**Selected GOPTIONS for Managing the Environment**

Generally, once you understand what options are available, I find it useful to save a list of them as a template which will handle most of your graphics applications and then use comments to able/disable them as needed. At that point, then use `%INCLUDE` or an AUTOEXEC.SAS to set those graphics options without having to bring the code into every graphics job you run.

**Common GOPTIONS for Controlling the Environment as a Whole (use for hardcopy)**

<table>
<thead>
<tr>
<th>GOPTIONS</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESET=</td>
<td>global</td>
<td>Resets any options that might have been changed</td>
</tr>
<tr>
<td>DEVICE=</td>
<td>WIN</td>
<td>Sets the device driver to Windows 3.1</td>
</tr>
<tr>
<td>COLORS=</td>
<td>(white yellow cyan green)</td>
<td>Orders the colors used</td>
</tr>
<tr>
<td>CBORDER=</td>
<td>white</td>
<td>Sets the background color</td>
</tr>
<tr>
<td>BORDER=</td>
<td></td>
<td>Draws a border around the graphic</td>
</tr>
<tr>
<td>GUNIT=PCT</td>
<td></td>
<td>Keeps the text in proportion regardless of output device</td>
</tr>
<tr>
<td>HTITLE=</td>
<td>6</td>
<td>Sets the height of TITLE1 to 6%</td>
</tr>
<tr>
<td>HTEXT=</td>
<td>3</td>
<td>Sets the height of all other text 5%</td>
</tr>
<tr>
<td>FTITLE=</td>
<td>ZapfBold</td>
<td>Sets the font for TITLE1 to Zapf Bold - Jim Goodnight's favorite according to him at the Sugi 18 opening ceremony</td>
</tr>
<tr>
<td>FTEXT=</td>
<td>Zapf</td>
<td>Sets the font for the rest of the text to Zapf Regular</td>
</tr>
<tr>
<td>ROTATE=</td>
<td>landscape</td>
<td>Useful when printing the graphic - orients the page on its side - not necessary for on-screen</td>
</tr>
<tr>
<td>NSIZE=</td>
<td>5 IN</td>
<td>Determine the horizontal space the graphic will occupy - ideal of your exporting the graphic to another software package like WordPerfect</td>
</tr>
<tr>
<td>VSIZE=</td>
<td>3 IN</td>
<td>Determines the vertical space</td>
</tr>
<tr>
<td>NOPROMPT=</td>
<td></td>
<td>SAS won't prompt for the next graph</td>
</tr>
<tr>
<td>CTITLE=</td>
<td>yellow</td>
<td>Default color for TITLES and FOOTNOTES</td>
</tr>
<tr>
<td>CTEXT=</td>
<td>white</td>
<td>Default color for graph text</td>
</tr>
</tbody>
</table>

Other graphics options for outputting SAS/GRAPH to CGM, PostScript and other proprietary graphics formats will be discussed briefly later.
I. Creating Basic Graphs

Once you have identified where you want your graph, plot or chart to go - either in printed form or in hardcopy - you need to decide on which type of graphic you would like to produce. SAS/GRAPH has a wide range of graphics capabilities and just a few will be discussed here. For an idea of charts, plots, graphs and maps that can be produced, refer to the SAS/GRAPH manuals listed in the bibliography.

Creating Bar and Pie Charts

You can use bar charts or pie charts to summarize your data graphically. For example, you could use charts to display

- **Bar charts, horizontal and vertical** - to represent statistics based on the values of one or more variables
- **Pie charts** - to compare the contribution of a variable to the whole
- **Star charts** - to graphically show out-of-balance data
- **Block charts** - to represent three-dimensional values of statistics

We will discuss two of these charts: bar and pie charts. Bar charts are used to represent the magnitude of data ranges. Pie charts are ideal for representing the relative contribution of each part to the whole.

To create bar and pie charts, you can use the GCHART procedure. The general form of the GCHART procedure is:

```sas
PROC GCHART DATA= SAS-data-set;
   HBAR chart-variable | options;
   VBAR chart-variable | options;
   PIE chart-variable | options;
RUN;
```
Example: GEORGIA

The data we will use for this example is included in the Appendix. It has the following structure:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Length</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>product</td>
<td>$7</td>
<td>One of Georgia's three main staples (from a Northern Californian's perspective)</td>
</tr>
<tr>
<td>city</td>
<td>$8</td>
<td>One of three popular Georgia cities</td>
</tr>
<tr>
<td>quarter</td>
<td>8</td>
<td>One of the yearly quarters</td>
</tr>
<tr>
<td>consumed</td>
<td>8</td>
<td>Amount of product that was consumed in these cities</td>
</tr>
</tbody>
</table>

Step #1 Specifying the Chart Variable

Our first task is to produce a simple vertical bar chart that shows us the relationship between the three products: peaches, peanuts and BBQ.

```plaintext
proc gchart data=georgia;
  vbar product /sumvar=consumed;
run;
quit;
```

Step #2 Changing the chart type

Similarly, let's produce a pie chart showing the same information.

```plaintext
proc gchart data=georgia;
  pie product /sumvar=consumed;
run;
quit;
```
Step #3 Change the chart variable to CITY, with PRODUCT stacked

Next, let's change the focus from product, to city where they were consumed.

```
proc gchart data=georgia;
  vbar city / sumvar=consumed
               subgroup=product;
run;
quit;
```

Step #4 Group the chart by quarter and hide the separate products

Then group the variables by quarter.

```
proc gchart data=georgia;
  vbar city / sumvar=consumed
               subgroup=city
               group=quarter;
run;
quit;
```

Step #5 Add titles and footnotes, clean up the legend, axis and patterns used for the bars

```
/* define title and footnote for chart */
title "Georgia's Top 3 Staples";
title3 "by City/Quarter";
footnote justify=right '(BARCHART.SAS)'
         justify=left 'Produced with SAS/GRAPH'
         move=(+0,+.5) '02'x move=(+0,-.5) 'Software';
```

```
/* AXIS statements allow you to supply information on how your vertical and horizontal axes will appear on the graph. */
axis1 color=gray
       value=none
       label=none;
axis2 color=gray
       label=('Units Consumed' justify=right '(in thousands)') width=3;
```
/*-----------------------------*/
| PATTERN and SYMBOL statements affect the area within the |
| plotting area. e.g., bars, lines, plot symbols.          |
/*-----------------------------*/

/* define pattern characteristics */
pattern1 value=solid color=bigg;
pattern2 value=solid color=vigb;
pattern3 value=solid color=rgr;
pattern4 value=solid color=gray;

/*-----------------------------*/
| This section produces the actual plot and any options that |
| directly relate to the data and the axis area.            |
/*-----------------------------*/

/* define legend characteristics */
legend frame label=none;

/* produce the graphic */
proc gchart data=georgia;
  format quarter roman.;
  vbar city / frame
    sumvar=consumed
    subgroup=city
    group=quarter
    legend=legend
    maxis=axis1
    raxis=axis2;
run;
quit;

Creating Basic Plots

A plot in its simplest form shows one variable against another within a set of coordinate axes. Generally, a plot is useful for showing the relationship between two variables. Examples of these kinds of plots include:

- **Simple line plots** - shows the relationship of one variable to another with a line

- **Two-dimensional scatter plots** - shows the relationship of one variable to another with symbols scattered in the two-dimensional space

- **Hi-low plots** - shows how several values of one variable relate to one value of another

- **Bubble plots** - shows the relative magnitude of one variable in relation to two other variables
To create a plot, you can use the GPLOT procedure. The PROC GPLOT statement takes on the general form:

```
PROC GPLOT DATA= SAS-data-set;
PLOT vertical-variable*horizontal-variable | options;
RUN;
```

**Example: EMPLOYEE**

The data we will use for this example is included in the Appendix. It has the following structure:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Length</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>$8</td>
<td>Employee's name</td>
</tr>
<tr>
<td>height</td>
<td>8</td>
<td>Employee's height in inches</td>
</tr>
<tr>
<td>weight</td>
<td>8</td>
<td>Employee's weight in pounds</td>
</tr>
<tr>
<td>age</td>
<td>8</td>
<td>Employee's age in years</td>
</tr>
</tbody>
</table>

**Step #1 Create a simple plot of WEIGHT by HEIGHT**

```
proc gplot data=employee ;
  plot WEIGHT * HEIGHT;
run;
quit;
```

**Step #2 Change the origin of the axis to (0,0) and change the symbol used for marking the plots**

```
symbol1 color=yellow
  value=star
  height=2;
axis1
  order=(0 to 80 by 20 )
  color=gray
  width=2.0
;
axis2
  order=(0 to 160 by 20)
```
Step #3  Revert back to the original axes and add a 95% confidence interval around the regression plot

axis1
   color=gray
   width=2.0
;
axis2
   color=gray
   width=2.0
;
symbol1 color=orange value=star
   height=2
   cv=yellow
   interpol=rlclm95;

proc gplot data=employee
   plot WEIGHT * HEIGHT / haxis=axis1 vaxis=axis2
; run;
quit;

Step #4  Add some advanced features and make it look nicer

proc means data=employee vardef=DF MEAN
   var HEIGHT WEIGHT AGE;
output out=OUTMEAN MEAN=MEAN1-MEAN3
; run;

Data _null_
   set outmean;
   call symput('HMEAN',MEAN1);
   call symput('WMEAN',MEAN2);
run;

382
/* define title and footnote for chart */
title "Plot of Weight and Height";
footnote justify=right '(PLOT.SAS)'
   justify=left 'Produced with SAS/GRAPH'
   move=(+0,+0.5) '02' move=(+0,-.5) ' Software';

/* define legend characteristics */
legend frame label=none;

/* produce the graphic */
proc gplot data=employee;
   plot WEIGHT * HEIGHT / haxis=axis1
                       vaxis=axis2
                       hminor=3
                       vminor=1
                       vref=&WMEAN
                       lvref=2
                       cvref=cyan
                       caxis=gray
                       ctext=gray
; run;
quit;

Enhancing your Output

So far we have only touched the surface as to the total capability of SAS/GRAPH. We have also seen how we can add titles and footnotes as well as options on the GOPTIONS statement to make the graph more legible. Other SAS statements such as SYMBOL, AXIS, PATTERN affect the area that is plotted. The SAS manuals listed in the reference section are a must if you want to explore the various options that can be specified to enhance your plots and charts. Many of the examples in this paper followed guidelines outlined by Betsy Coming in her Observations series cited in the bibliography.

Managing Your Output

Most of the focus during this hands-on session has been getting SAS to display your graphic back to your monitor. Output from SAS/GRAPH is automatically stored in a temporary catalog within the SAS System (called WORK.GSEG). You can also store your graphics permanently and use procedures such as PROC GREPLAY to view them at a later time. The next example shows you how to permanently save your output into a catalog.
**SAS Graphics Catalogs**

To create a permanent version of your graph in a SAS catalog, you must first specify a place for the catalog to be stored, then run a SAS/GRAPH procedure with the GOUT= option. SAS (Version 6) stores its catalogs in libraries specified on a LIBNAME statement. In a directory based system like DOS, Windows or OS/2, we specify the libname by pointing it to a directory. For example:

LIBNAME mylib 'c:\mygraphs';

Then we can run our procedures as we did previously with the GOUT= option pointing it to this library.

proc gplot data=employee gout=mylib.mygraf;
  plot WEIGHT * HEIGHT;
run;

Our plot is now stored in a permanent library called MYLIB in a catalog called MYGRAF.

**Exporting a SAS Graphic to Word Perfect for Windows (Version 5.2)**

In addition to printing and saving your output, you may at some time want to take those graphics and output, or export them into another software package. The SAS Institute has written a series of Technical Support (TS) documents which take the user through exporting SAS/GRAPH output to a variety of software packages. The TS-252 series can be obtained through SAS Technical Support or via their Anonymous FTP. The following example takes you through exporting one of the plots created above to a CGM file so we can import it into Word Perfect 5.2 for Windows. If you want to create a CGM or postscript file on a MVS or VM system, you’ll also need a GPROTOCOL= statement (e.g., sasgpasc’or gddmfam4).

Filename GSASFILE ‘a:\import.me’;

Goptions GSFLEN=4096
  GACCESS=GSASFILE
  GSFNAME=GSASFILE
  GSFMODE=replace,

Goptions
  DEV=CGMWPWA
  NODISPLAY;

proc gplot data=employee;
  plot WEIGHT * HEIGHT / haxis=axis1 vaxis=axis2
    hminor=3 vminor=1
    vref=&WMEAN lvref=2
    cvref=cyan caxis=gray
    ctext=gray;
run; quit;
Once you have created the file, you can import it into WordPerfect for Windows 5.2. If you are already in WordPerfect, use the following:

1. Choose the Graphics menu
2. Select Figure...
3. Select Retrieve...
4. Change the directory to the A: drive and select IMPORT.ME. The file will be brought into the current document at your cursor location.

**Summary**

In this hands-on session, we have learned how to create some basic charts and graphs with SAS/GRAPH software. By learning just a few components, hopefully, we were able to take some of the mystery out of the process.

Although these examples were simple, we hope that you can see the power of the software and you ought to feel great that you now have some experience under your belt.

Now you're ready to do "real work" with SAS/GRAPH.

**References**


SAS/GRAPH Software: Usage, Version 6, First Edition

SAS/GRAPH Software: Usage, Version 6, First Edition


SAS/GRAPH Software: Examples, Version 6, First Edition


Appendix A: Data used for the Graph and Charts

/* create data set GEORGIA */
data georgia;
    length product $ 7 city $ 8; input name $ height weight age;
consumed; cards;
    Peaches Savannah 1 97
    Peaches Atlanta 1 26
    Peaches Athens 1 97
    Peanuts Savannah 1 82
    Peanuts Atlanta 1 21
    Peanuts Athens 1 98
    BBQ Savannah 1 35
    BBQ Atlanta 1 18
    BBQ Athens 1 70
    Peaches Savannah 2 44
    Peaches Atlanta 2 07
    Peaches Athens 2 29
    Peanuts Savannah 2 31
    Peanuts Atlanta 2 59
    Peanuts Athens 2 56
    BBQ Savannah 2 17
    BBQ Atlanta 2 99
    BBQ Athens 2 34
    Peaches Savannah 3 96
    Peaches Atlanta 3 91
    Peaches Athens 3 85
    Peanuts Savannah 3 46
    Peanuts Atlanta 3 66
    Peanuts Athens 3 38
    BBQ Savannah 3 10
    BBQ Atlanta 3 52
    BBQ Athens 3 96
    Peaches Savannah 4 57
    Peaches Atlanta 4 51
    Peaches Athens 4 08
    Peanuts Savannah 4 99
    Peanuts Atlanta 4 30
    Peanuts Athens 4 37
    BBQ Savannah 4 74
    BBQ Atlanta 4 19
    BBQ Athens 4 25
;
run;

/* create data set EMPLOYEE */
data employee;
    input name $ height weight age;
cards;
    Margaret 59.0 99.5 12
    Joe 56.5 84.0 13
    Mike 57.3 83.0 12
    Glenn 59.8 84.5 12
    Walter 62.5 84.0 13
    Bob 64.8 128.0 12
    Carolyn 51.3 50.5 11
    Dorinda 64.3 90.0 14
    Andrew 56.3 77.0 12
    David 66.5 112.0 15
    Diane 62.8 102.5 14
    Michael 63.5 102.5 14
    Cletus 67.0 133.0 15
    Rex 62.5 112.5 15
    Mike 57.5 85.0 11
    Harold 72.0 150.0 16
    Bill 66.5 112.0 15
    Greg 69.0 112.5 14
    George 65.3 98.0 13
;
run;