Getting Started With SAS/GRAPH® the Easy Way

Hany AbouTaleb / Genevieve Kearse, Hoechst Roussel Pharmaceuticals Inc., Somerville, N.J.

ABSTRACT
No single statistical tool is as powerful as a graph. Using graphs we can display a large amount of information, look for patterns and relationships, support or disprove a proposed hypothesis and discover new ideas. This article will describe and illustrate a graphical methodology for data analysis using basic rules and principles. The following dossier of statistical graphs illustrates the methods and principle elements of graphing data using SAS/GRAPH software. The SAS system and SAS/GRAPH software provide basic tools for statistical analysis and graphical display.

Introduction:
This article is divided into several sections and designed to outline the principles and methodology for graphical data analysis, describe the SAS/GRAPH environment, and illustrate the various graphical elements using the SAS system and SAS/GRAPH software. Graphical data analysis is a powerful communication tool. The graphical form and quantitative information can be enhanced by using some basic rules for graphic presentation:

1. Clear Data Presentation:
   - Make the data stand out by using a graphical element that is visually projected. Avoid extra elements of the graph to interfere with the data.
   - Connecting plotting symbols by lines is useful to track the movement of the values. Avoid obscuring the symbol with the lines or overlapping symbols so all data can be seen.
   - Use a pair of scale lines for the horizontal and vertical scales and position the data region within the rectangle. Use a reference line when there is an important value that must be seen across the entire graph. Place tick markers outside of the data region. Avoid clutter in the data region, and minimize the number of tick marks.
   - Do not graph too much in a single graph. Different graphical elements in the data region obscure one another.
   - Do not allow data labels, notes, keys, and markers in the data region to interfere with the data or to clutter the graph. Put keys and markers outside the data region, and put notes in the legend or footnote text.
   - Plotting symbols must be visually distinguishable and clarity preserved under reduction and reproduction so that all the data can be seen.

2. Clear Understanding:
   - Put major conclusions into graphical form. Make legends comprehensive and informative. They should describe everything that is graphed and draw attention to the important features of the data.
   - Error bars should be clearly explained and communicate meaningful information relating to the data. Avoid ambiguous descriptions.
   - Use logarithms of a variable when percent change is important. The scale label should correspond to the tick mark labels.
   - Graphs should be carefully checked for errors.
   - Graphs designed for clarity describe everything that is graphed, enable important data to stand out, and communicate conclusions drawn from the data.

3. Scales / Ranges:
   - Choose the range of the tick marks to include or nearly include the range of the data.
   - Choose the scales so that the data fill up as much of the data region as possible.
   - Do not insist that zero be included on a scale showing magnitude, if it ruins the resolution of the data on the graph.
   - Use a logarithmic scale when it is important to understand percent change or multiplicative factors. Showing data on a logarithmic scale can improve resolution.
   - Use a scale break only when necessary. If a break cannot be avoided, use a full scale break. Do not connect numerical values on two sides of a break.

4. General Strategy:
   - A large amount of quantitative information can be packed into a small region.
   - Graphing data should be an iterative, experimental process.
   - Graph data two or more times when it is needed to examine options.
   - Many useful graphs require careful, detailed study.
Producing Graphics Output In the SAS System:

The SAS system and SAS/GRAPH software provide the basic tools for statistical analysis and graphical display. When we use SAS/GRAPH software, we run programs to produce graphics. Once they are created we can:

- Display graphics on the monitor.
- Print graphics on a printer or plotter.
- Store graphics in host file.
- Store graphics in SAS Catalog.

Once graphics are displayed, we can:
- Display graphics on the monitor.
- Print graphics on a printer or plotter.
- Store graphics in host file.
- Store graphics in SAS Catalog.

To display the graphs on a monitor, you must first select a SAS/GRAPH device driver to match the monitor in use.

1. Begin a new SAS session.
2. Invoke the GDEVICE procedure by entering the following, then pressing the SUBMIT key.

```
proc gdevice;
run;
```

3. Scan the displayed list and choose the device drivers that match your monitor and printer (example: monitor - win, printer - winplot).

4. End the gdevice procedure and then set the device driver in SAS session by submitting the following Goptions statement:

```
Goptions DEVICE=win
TARGETDEVICE=winplot
```

Where DEVICE=win selects the Microsoft windows display device, and TARGETDEVICE=winplot selects the Microsoft hard copy (color) as the target device.

A partial list of the devices drivers is shown in Output.

Specifying a target device ensures that the appearance of graphics on the display will be very close to that of the hard copy. Using TARGETDEVICE= option sacrifices some hardware capabilities on some devices; you can remove the target device by specifying GOPTIONS TARGETDEVICE=.

REST=GLOBAL
Cancels all currently defined AXIS, FOOTNOTE, LEGEND, PATTERN, SYMBOL, and TITLE definitions.

It is useful to rest GOptions every time you write a program to display a graph.

GUNIT=PCT
Specifies that options in SAS/GRAPH procedure and global statement use units of percent of the graphics output area unless other units are used.

ROTATE=LANDSCAPE
Sets the orientation of the graphs to landscape orientation.
CBACK=WHITE
Sets the background color of the monitor to white.

HTITLE=4
Sets the text height for the first title to 4 (in units of percent of the display height).

HTEXT=3
Sets the text height for all text in graphs to 3 (in units of percent of the display height).

FTEXT=duplex
Sets the text font to duplex for all text on graphs.

HPOS=80, VPOS=40
Specifies the graphs to use the 80 rows and 40 columns. For more options see the GRAPHICS OPTIONS AND DEVICE PARAMETERS chapter.

Macro Language:
Use SAS macro language so that the programs are flexible enough to accept changes in the device or the data. The efficiency of your program can be improved by adding macros. For more details, Please refer to the SAS MACRO Reference book.

Annotate Facility:
The annotate facility enables one or the programmer to program graphics by using variables in SAS data sets. Use this facility to enhance graphics by adding text or some elements to the graphics output. This facility can also be used to construct custom graphics output by itself. For more options, see the ANNOTATE chapter vol. 1.

HOW SAS/GRAPH fits into the SAS System:

The SAS/GRAPH produces graphs by using data from SAS data sets, or with PROC step runs on existing data and produces graphics output which can then be displayed in the graph window, written to a file or printed.

A SAS/GRAPH program can be made up of these parts:

1. FILENAME statements: for external files used as input and output.
2. LIBNAME statements: for SAS libraries used as input or output.
3. GOPTIONS statements: to set up global characteristics of a graph and control where the output is sent.
4. Global statements: to set up the titles, footnote, legend, axes, symbols and patterns.
5. Procedure statements: determine what data sets are used, what catalog is used for catalog output and what kind of graph is produced.

With the global statements we can apply the basic rules and principles for producing graphs as follows:

Titles and Footnotes: can appear anywhere in the program and once they are defined, the titles and footnotes remain in effect until they are canceled or the SAS session is ended. Up to ten titles and ten footnotes can be used in each graph. To change a title or footnote that has already been defined, issue another TITLE statement that has the same number as the title you want to change.
The null Title cancels the title and footnote. You can control the font and the height of the characters by FTEXT= and HTEXT= options, defined in the GOPTIONS statement but by using certain options within TITLE and FOOTNOTE statements, you can override default.

Example: title1 font=duplex height=3 pct'SAS/GRAPH';
For more options see the TITLE and FOOTNOTE chapter 11.

AXIS: statements give the user ability to control the location, values, and appearance of the axes of the charts. You can control the order of the data, the locations and appearances of the axis line and the tick marks, and the text appearance of major tickmark. The AXIS definitions are used only when they are explicitly assigned by an option in a procedure that produces a chart.

For more options see AXIS Statement chapter 11.

For more options see PATTERN Statement chapter 15.

SYMBOL: statements define the characteristics of the symbols that display the data plotted by GPLOT procedure. These characteristics are the appearance of plot symbols and plot lines including bars, boxes, confidence limit lines, and area fill; interpolation methods; how plots handle data out of range; and the number of patterns. For more options, see SYMBOL Statement chapter 16.

Procedure Statement: to draw graphs, within a procedure certain statements and options are used. For more detail see SAS/GRAPH software, vols. 1 and 2.

LEGEND: statements give the user the ability to control the location and appearance of legends on a chart. You can have up to 99 legends; you can control the type of the pattern for the graph, the color of the pattern, or repeat the pattern. These are three different types of patterns: bar and block patterns, map and plot patterns, and pie and star pattern. The procedure uses the pattern you create or can generate them as needed and assigns them to the graph by default. Times a definition is used. You can have up to 99 symbols. For more options, see SYMBOL Statement chapter 16.

data homerun;
input hitter$ homerun @@;
run;

data homerun;set homerun;
by hitter;
if first.hitter then year=0;
year+1;
" set selected graphics options for the examples */
goptions reset=global gunit=pct cback=white colors=(black blue green red) htitle=6 htext=4 ftext=zapf border;
" define title and footnote */
title1 e=black h=4 'Comparison of Home run hitter of all time, Ruth or Maris';
" modify symbol characteristics */
symbol1 interpol=join value=none color=red height=3;
symbol2 interpol=join value=dot color=blue height=3;
" modify horizontal axis */
axis1 minor none label=('Years') offset=(3,3) length=70;

1010
/* modify vertical axis */
axis2 minor=(number=1 height=1) major=(height=1.5)
   label=(a=90 r=0 'Home Runs');
/* modify the legend */
LEGEND1 LABEL =(POSITION=TOP J=L F=SIMPLEX H=4 C=BLACK 'Legend')
   MODE = RESERVE
   ACROSS=1
   SHAPE = SYMBOL(6,3)
   VALUE =(H=4 F=SIMPLEX J=L C=BLACK)

Comparison of Home run hitter of all time, Ruth or Maris

/* set selected graphics options for the examples */
goptions reset=global gunit=pct device=win
targetdevice=ps cback=white colors=(black blue green red)
   ROTATE=LANDSCAPE
   htitle=4 htext=3 ftext=duplex border;
/* define title and footnote */
title1 e=black h=4 'Comparison of Home run hitter of all time, Ruth or Maris';
/* modify symbol characteristics */
symbol1 interpol=box10 cv=red mode=include value=
   co=blue height=5;
/* modify horizontal axis */
axis1 label=('Hitter') minor=none offset=(3,3);
/* modify vertical axis */
axis2 minor=(height=1) major=(height=1.5)
   order=(0 to 70 by 10)
   label=(a=90 r=0 'Home Run');
axis3 minor=(number=1 height=1) major=(height=1.5)
   order=(0 to 15 by 5)
   label=(a=90 r=0 'Years');
/* produce a plot for boxplot */
proc gplot data=homerun;
   plot homerun*hitter frame
   hminor=3 vminor=1
   haxis=axis1 vaxis=axis2;
   plot year*hitter frame
   hminor=1 vminor=3
   haxis=axis3 vaxis=axis1;
   run;quit;

/* Author: HANY ABOUTALEB */
/* Title: The Greatest home run hitter of all time */
/* Purpose: To be used in the SAS paper */
/*---------------------------*/
data homerun;
   input hitter$ homerun @@;
cards;
Ruth 54 Maris 8 Ruth 59 Maris 13 Ruth 35 Maris 14
Ruth 41 Maris 16 Ruth 46 Maris 23 Ruth 25 Maris 26
Ruth 47 Maris 28 Ruth 60 Maris 33 Ruth 54 Maris 39
Ruth 46 Maris 61 Ruth 49 Ruth 46 Ruth 41 Ruth 34
Ruth 22;
/* produce a plot with a legend */
proc gplot data=homerun;
   plot homerun*year=hitter frame
   haxis=axis1 vaxis=axis2 legend=legend1;
data homerun;set homerun;
   by hitter;
   if first.hitter then year=0;
   year+1;
/* set selected graphics options for the examples */
goptions reset=global gunit=pct device=win
targetdevice=ps cback=white colors=(black blue green red)
   ROTATE=LANDSCAPE
   htitle=4 htext=3 ftext=duplex border;
/* define title and footnote */
title1 e=black h=4 'Comparison of Home run hitter of all time, Ruth or Maris';
/* modify symbol characteristics */
symbol1 interpol=box10 cv=red mode=include value=
   co=blue height=5;
/* modify horizontal axis */
axis1 label=('Hitter') minor=none offset=(3,3);
/* modify vertical axis */
axis2 minor=(height=1) major=(height=1.5)
   order=(0 to 70 by 10)
   label=(a=90 r=0 'Home Run');
axis3 minor=(number=1 height=1) major=(height=1.5)
   order=(0 to 15 by 5)
   label=(a=90 r=0 'Years');
/* produce a plot for boxplot */
proc gplot data=homerun;
   plot homerun*hitter frame
   hminor=3 vminor=1
   haxis=axis1 vaxis=axis2;
   plot year*hitter frame
   hminor=1 vminor=3
   haxis=axis3 vaxis=axis1;
   run;quit;
**Comparison of Home run hitter of all time, Ruth or Maris**

```
%macro create_homerun_data;set homerun
%do hitter=Ruth Maris %by %; if %eq %hitter% Ruth then hit=1; else hit=2;
%end; %by hitter;
%run;
```

```sas
data homerun;set homerun;
by hitter;
if hitter='Ruth' then hit=1; else hit=2;
if first.hitter then year=0;
year+1;
proc print; title 'The Greatest home run hitter of all time';
set selected graphics options for the examples
%do axis=1 to 2 %by ;
  %let out=.03 %if %eq %axis% 1 %then cback=white colors=(black blue green red) %else cback=white colors=(black blue green red) %end;
%end;
%define title and footnote
%do symbol=1 to 2 %by ;
  %let out=3 %if %eq %symbol% 1 %then interpol=join value=dot color=red height=3 %else interpol=join value=dot color=red height=3 %end;
%end;
%modify horizontal axis
axis1 minor=none label=('Years') offset=(3,3) width=3;
/* modify vertical axis */
axis2 minor=(number=1 height=1) major=(height=1.5)
  label=(a=90 r=0 'Home Runs hit');
proc format; value hitter 1='Ruth' 2='Maris';
/* produce a bubble plot */
proc gplot data=homerun;
  bubble homerun*year=hit
    / frame haxis=axis1 vaxis=axis2
    hminor=O BSIZE=12 BCOLOR=RED BLABEL
    caxis=blue blfont=swissi
    CFRAME=WHITE;
  format hit hitter.;
  run;
quit;
```

**Conclusion:**

Graphics, in general, are powerful tools for the visualization. The SAS System and SAS/GRAPH software provide basic tools for statistical analysis and graphical data display. It can be enhanced by using basic rules for graphics.

**References**


**ACKNOWLEDGMENTS:**

The authors like to thank Dorothy Huang, Manager of the scientific programmer group at Hoechst for her encouragement and support, Steven Dawls from the medical writing group at Hoechst for his time to enhance this document, Prof. Judy Coomes at William Paterson College and Dr. Thomas F. Devlin at Montclair State University for their suggestions and advice.
Contact:

Hany AbouTaleb
Hoechst Roussel Pharmaceuticals Inc.
Route 202-206
P.O. Box 2500
Somerville, NJ 08876-1258
Office: (908) 231-3486
FAX: (908) 231-4779
e-mail: abouta31@crd01.hcc.com

Genevieve Kearse
Hoechst Roussel Pharmaceuticals Inc.
Route 202-206
P.O. Box 2500
Somerville, NJ 08876-1258
Office: (908) 231-3944
FAX: (908) 231-4779
e-mail: kearse31@crd01.hcc.com