BUILDING ON THE DEFAULT LINE: STURDY SAS/GRAPH® CONSTRUCTION

Susan J. Kenny
Quintiles Inc.
Research Triangle Park, NC

Introduction

To the new user, learning to use SAS/GRAPH software can be confusing. This is because SAS/GRAPH has many features and options that make it powerful yet at the same time can make it overwhelming. This paper is designed to help the new user understand some of the basic features of SAS/GRAPH and how to avoid common errors.

Basic Building Tools

GOPTIONS

Similar to system options, SAS/GRAPH has graphic options that can be used to control the graphic environment and graphic output. These options are specified in a GOPTIONS statement. Among the many graphic options that are available, the following are perhaps the most important for the beginning user.

DEVICE= device-driver-name

This graphic options specifies which SAS supplied device driver to use to produce the graph. The device driver name that is used in the statement must be a valid device name that SAS can find in a device catalog. By default, this device catalog is found in theSASHELP.DEVICES catalog. To determine the device driver name for your device, use the producedure GDEVICE and scan the descriptions to locate your device. For example, DEVICE=PS specifies the device driver name if you will be using a postscript device for hardcopy production. If you are working in Microsoft Windows® and wish to view your graph on the screen, you would use DEVICE=WIN.

TARGETDEVICE=device-driver-name

If you are working in an interactive windowing environment, you will most likely wish to see your graph on the screen as it will appear in hard copy (WYSIWYG). The device driver name that is supplied to this goption, therefore, is the name for the device where the graph will be ultimately produced. For example, if you are working under Windows but eventually will be producing your hard copy graph with a Hewlet Packard Laser Jet Series III printer, you would use the statement

```
 GOPTIONS DEVICE=WINTARGETDEVICE=HPLJS3;
```

ROTATE=landscape | portrait

This options specifies the orientation of the graphic display. Each device has a default orientation so this option is only needed if you wish to change the orientation. Even though it may be redundant, explicitly specifying the orientation for each device will serve as a reminder for the desired orientation.

DISPLAY/NODISPLAY

This option is used in an interactive environment and indicates whether or not to display the graph on the screen. The NODISPLAY options is useful when a series of graphs is being produced and you do not wish to view each graph as it is being created.

RESET=all | global | statement name

It is important to remember that GOPTIONS stay in effect until they are changed or reset. There may be times where you will want to reset one or more GOPTIONS within the body of a program and this statement will allow you to do this. The RESET=all resets all GOPTIONS to their default values while the RESET=global will reset only the global statements unique to SAS/GRAPH (ie, AXIS, FOOTNOTE, LEGEND, NOTE, PATTERN, SYMBOL, AND TITLE statements). If you wish to reset only certain statements, then use the RESET=statement name format.

Unlike system options, the order which you specify the graphic options is important. Most importantly is the position of the DEVICE= option. Once this option is specified, the values for other options, such as ROTATE=, are set to the device default. Therefore, if you wish to override a default device option, you must specify that option after you specify the device. For example, the statement

```
   GOPTIONS DEVICE=PS ROTATE=LANDSCAPE ;
```

will produce a landscape display on a postscript printer but the statement

```
   GOPTIONS ROTATE=LANDSCAPE DEVICE=PS;
```

will produce a portrait display since the default orientation for the postscript device is portrait.

DEVICE DRIVERS

You may wonder why you have to use the TARGETDEVICE goption in order to see the graph on a screen as it would appear in hard copy. This is because the capabilities of each device driver are different. Each printer and its associated device driver can reference a certain number of dots per inch (DPI) or pixels on a page. Some devices, such as high resolution color printers, can reference many pixels, which creates a finer graphic image, while other devices have fewer pixels, such as a dot matix printer.

For each device driver, SAS also defines a grid of cells, each cell containing a defined number of pixels. By default, SAS uses cells as the unit of measurement with height specifications. Therefore, the statement

```
TITLE1 H=2 'MY TEXT';
```

will create text that is 2 cells high but the physical height of this text will be dependent on the device driver.

FINDING INFORMATION

To find out more information about your graphic devices, use the GDEVICE procedure. This will provide information on the number of cells (HPOS and VPOS), the total number of pixels, the maximum size of paper the device can support, the number of colors available, just to name a few. The procedure GTESTIT can be used as a diagnostic tool for testing the configuration of your device and will provide some of the current settings of device parameters and graphic options.

THE SAS/GRAPH BLUEPRINT

The basic elements of any graph are titles, footnotes, and the graphic image. To prevent these elements from overlaying each other, SAS/GRAPH reserves space for each element in a defined order. First, space is allotted for text specified in the TITLE statements. Next, space is reserved for text specified in FOOTNOTE statements. If a LEGEND statement is specified, by default space is reserved below the axis area of the graph and above the footnote area. Lastly, whatever space is left over after the title, footnotes and legend space is reserved is then used for the graphic image, which produced by one of the graphic procedures. This area is thus called the procedure output area. If the title, footnote, and legend areas are too large, there may not be enough room to produce the graph. If this is the case, you will get a error message to this effect and no output is produced.

AVOIDING DESIGN FLAWS

FLOATING Y-AXIS TEXT

Have you ever produced a graph and ended up with the y-axis text appearing at an undesireable distance away from the axis? No amount of respecifying the axis text will alleviate this situation. This is because SAS/GRAPH will leave space for the axis values appearing next to the major tick marks according to the size of the format of the y-axis variable, regardless of whether any numeric value actually utilizes the full size of the format. For example, suppose a numeric variable, YAXIS, has a format of 12. yet actual values of the variable range from 0 to 100. When using this variable on the y-axis, SAS will allow enough space to accommodate numeric values with up to 12 significant digits. Thus, the axis label is pushed out this distance, which is much greater than what is needed to present the maximum value of 100. Often this situation will occur when using data that was produced via a spreadsheet application export, such as with Excel®. By default, these applications will use a 12. format for any numeric value. To correct this situation within SAS, determine the maximum and minimum values of the y-axis variable and then specify a more fitting format either within a data step or in the graphic procedure.

AXES THAT DO NOT APPEAR

Every SAS/GRAPH user will sooner or later become familiar with the error message that 'the axis cannot be fit as specified...'. Generally, this error results from one of two situations. If you are trying to specify an axis order with many increments, it is possible that SAS/GRAPH will not be able to produce all of the axis values because there are too few cells to accommodate all of the text at the given height. For example, if you specify

```
AXIS1 ORDER=(1 TO 100 BY 1) VALUE=(H=1);,
```

SAS/GRAPH may not have 100+ cells within which to produce the axis values. Rather than automatically reduce the axis order or the height, SAS will instead issue the error message and no graph will be produced. To solve this problem, select fewer increments and/or reduce the height of the axis values. Keep in mind that it is more advantageous to have fewer, readable axis values than many difficult to read values.

The second situation that might yield this error is one where the procedure area for the graphic image is too small. Generally, this will happen if there are many and/or large text specified in titles, footnotes, or legend statements that take too much space away from the procedure area. Alternately, if you are specifying a fixed length for your axis, the length may be too long for the space available. Therefore, to fix this problem, reduce the number or size of the text statements or reduce the size of the length of the axis.

STATISTICS THAT ARE NOT ACCURATE

Unfortunately, the potential for graphically presenting incorrect statistical results without realizing it is all too possible. A few safeguards and a good understanding of how SAS/GRAPH internally calculates statistics can help prevent this undesirable situation. SAS/GRAPH can automatically present a variety of descriptive statistics. Generally, these statistics are obtained by specifying the interpolation (I=) option within a SYMBOL statement. For example, using I=BOX will produce box and whisker plots for the distribution of the y-axis variable at each level of the x-axis variable. Similarly, I=RL will include a regression line on an y*x plot. Other options include I=STD which is used to create standard deviation (or optionally standard error of the mean) bars on either side of the mean.

SAS/GRAPH will calculate the desired statistics using the observations within the data set specified in the graph procedure *provided the data values fall within the specified axis range*!! By default, SAS/GRAPH uses the SYMBOL option MODE=EXCLUDE to exclude any data values from the internal statistical calculations which occur outside the axis range. If you wish to restrict an axis range to be less than the minimum or maximum data values, then use MODE=INCLUDE in your SYMBOL statements. When using SAS to internally calculate any statistics, it is always a good idea to also run the statistical procedure, such as MEAN, UNIVARIATE, or REG, against the data that is being plotted. Then visually compare your graph results with the output from the statistical procedure. Although this may be tedious, it is better than running the risk of having an inaccurate graph. Be cautious when formatting numeric values that will be used for statistical calculations. If the format specified in a SAS/GRAPH procedure is too small for the variable, rounding will occur. This will also produce slightly inaccurate results.

SYMBOLS THAT ARE NOT AS EXPECTED

The misspecification of the SYMBOL statement is probably one of the more frequent causes of graph that do not look as expected. It is not always obvious how many symbol statements are needed and how they are used. The first important thing to recognize is that if you are using a color graphic device, there is a colors list that is used by SAS/GRAPH. This colors list provides the number of available colors. If the color option (C=) is not specified in a SYMBOL statement, the SYMBOL definition will cycle through each color in the colors lists before the next symbol definition is used. For example, suppose you are using a device that has 4 available colors. If you use the statements

```
SYMBOL1 V=STAR;
SYMBOL2 V=SQUARE;
```

SAS will internally generate 8 symbol definitions, the first 4 with the star symbol and the next 4 with the square symbol. To avoid this situation, use the C= option within each SYMBOL statement. Once a color is specified for a definition, SAS will not cycle through the internal colors list. This concept also applies when using the PATTERN statement.

Understanding how many symbol definitions are needed is sometimes difficult. In general, you will need a SYMBOL statement for each time SAS/GRAPH plots a Y by X value, assuming that you wish to have a different symbol value for each of the plots. For example, if you are using the statement

```
PLOT Y*X=Z;,
```

SAS/GRAPH will be producing ‘Z’ plots, all plotted within the same axes. Therefore, you will need ‘Z’ SYMBOL statements. If you are using the overlay option with the statement

```
PLOT (A*X) (B*X) (C*X)/OVERLAY; ,
```

then you will need 3 SYMBOL statements to adequately distinguish between the A, B and C variable values.

TRICKS OF THE TRADE

Albert Einstein once said that ‘imagination is more important than knowledge’. Keep this in mind when using SAS/GRAPH. Often, you may wish to do something that is not found on any page of the manual. If this is the case, use your imagination and most often, you will find that the SAS software will accommodate your desires. Here are a few examples to get you started.

Creating Graph Margins Without Using a Template

Suppose you wish to add a 2 inch margins around all sides of your graph. You look at the manual under GREPLAY and read about templates and soon you begin to feel faint. As a quick alternative to using a template to run your graph through consider using 3 title statements and 1 footnote statement to place blank or ‘dummy’ text of the desired heights. Here, the trick is to use the angle option (A=) on 2 of the title statements to angle them 90 degrees clockwise and counterclockwise. With the addition to the other title statement being placed at the top of the page and the footnote statement being the last footnote specified, you will be essentially adding blank space all around the margins of your graph. Keep in mind that SAS/GRAPH supplied device drivers generally allows for a ½ inch margin on all 4 sides. So if you wish a total of 2 inches, then your text statements will need to be specified as 1.5inches. The following statements will do the trick:

```
TITLE1 H=1.5IN ‘     ‘;
TITLE2 A=-90 H=1.5IN ‘    ‘;
TITLE3 A=90 H=1.5IN ‘     ‘;
FOOTNOTEX H=1.5IN  ‘      ‘;
(where x is the last footnote specified)
```

Keeping Graphs Uniform Across a Series

Suppose you have 10 graphs and you wish for all of them to be similar in appearance. However, some graphs have more titles or footnotes than other graphs and as such, the procedure area for the graph will change size. This will create non-uniformity in the look of the graphs. Here, you have at least 2 options to solve this problem. First, you could specify the same number of title or footnote statements for each graph and use dummy statements as place holders when there is no actual text to present. Alternately, you could use the ORIGIN= option on the AXIS statement. This will indicate to SAS where to place the origin of the graph and is specified as an (X,Y) coordinate in units of either percent of the page or in actual physical measurement on the page (in inches or centimeters). This option will anchor the origin of the graph to this location and so each graph will appear at the same place. Keep in mind that you must specify the origin such that there is enough room to produce the graph after space is allocated for the title and footnote statements.

Avoiding Crowding of Axis Values

As noted above, if you specify too many increments on the axis, there is the risk of running out of room and not having your graph be created. As an alternative to reducing the number of increments and values, you might consider using the angle option (A=) on the VALUE statement of the axis definition. By angling the axis value text, more text will be able to fit along the axis.

Using By Values/Variables in Text Statements

There may be times that you wish to place values of BY variables or BY variable names in footnotes or titles. This can be done with macro variables but an easier solution is to use a new option of the SAS software that is available for releases 6.07 and later. To place the current value of the specified BY variable in a text string, insert the text '#BYVALn' in your text statement, where 'n' specifies the position of the variable in the BY statement. Alternately, you can insert '#BYVAL(variable-name)' in the text statement. For example, suppose you are doing a series of graphs with the statement BY DOSE SEX;. The title statement

```
   TITLE1 'DOSE RESPONSE RELATIONSHIP FOR
   DOSE GROUP OF #BYVAL(DOSE) FOR
   #BYVAL(SEX)';
```

would equate to 'DOSE RESPONSE RELATIONSHIP FOR DOSE GROUP OF 10 MG/DL FOR FEMALES' in production, for values of '10 mg/dl' and 'females' for the DOSE and SEX variables, respectively. Similar syntax would be used with the #BYVAR option.

SUMMARY

Perhaps the best tool to use when beginning to work with SAS/GRAPH is persistence. At first try, many things may not seem obvious or even logical. But with persistence comes experience and soon you will be able to create graphs with ease. Don't overlook the many sources of help for SAS/GRAPH. In addition to the SAS/GRAPH manuals, searching through the past SUGI proceedings can often yield papers that address the topic that you are working with. Additionally, consider looking through the several books by users that have been written specifically for SAS/GRAPH. And finally, don't forget to have fun!

Trademarks

The author can be contacted at:

Susan J. Kenny, Ph.D.
Statistical Programming Systems
Quintiles, Inc.
PO Box 13979
Research Triangle Park, NC 27709
(919) 941-1463
skenny@quintiles.com