Extreme Graphics Make Over: Using SAS/GRAPH® to Get the Graphical Output You Need

Miriam Cisternas, Ovation Research Group, San Francisco, CA
Art Carpenter, California Occidental Consultants, Oceanside, CA

ABSTRACT
Back in the days of the mainframe, creating presentation quality graphs was a challenge, but with the current versions of SAS/GRAPH® customizing your graphical output can be quite easy. Easy that is, once you know which of the many graphical statements and options to use.

This workshop will introduce you to those statements and options. Through a series of exercises we will show you how you can dress up your GPLOT graphs. You will quickly learn how to customize titles, footnotes, axes, and plotting symbols, as well as how to add reference lines, legends and notes to highlight specific messages in your data. You will also gain an understanding of how specific GOPTIONS and global statement options interact to create your finished product.

KEYWORDS
SAS/GRAPH, GPLOT, AXIS, GOPTIONS

INTRODUCTION
This workshop is intended for SAS programmers with a working knowledge of BASE SAS® but who are new to SAS/GRAPH. SAS/GRAPH is a collection of procedures to produce graphics output, including graphs, charts, and maps, and allows for fine-tuned customization of text appearance, colors, and virtually every element of the graphic output created. Output can be viewed on screen, saved in a number of file formats, or printed. But the flexibility and power of SAS/GRAPH can be overwhelming to the new user, in part because there are a lot of statements and options to learn and also due to some global statements having different behavior than their BASE SAS equivalents. The goal of this workshop is to gently introduce you to SAS/GRAPH options and SAS global statements, and to provide you the necessary tools to explore further after the conference.

STRUCTURE OF SAS/GRAPH PROGRAMS
- SAS/GRAPH programs that include any sort of customization will usually follow a structure that includes the following elements:

- A GOPTIONS statement that sets graphics options throughout the SAS session or until another GOPTIONS statement is specified. In the absence of a GOPTIONS statement, default settings are used.

- Global statements to define titles and footnotes, as well as graphic elements such as legends, plotting symbols and axes. As in BASE SAS, these statements can be specified before or within a procedure and remain in effect throughout the SAS session or until reset or cancelled by a GOPTIONS statement.

- SAS/GRAPH procedure(s) to produce graphics output. These may include options that override those specified in the GOPTIONS statement.

SYNTAX OF GENERAL STATEMENTS YOU SHOULD LEARN FOR SAS/GRAPH
The global GOPTIONS statement includes over 150 options. The options used in this workshop include:

Option	Explanation
RESET=all | statement_name	Resets all GOPTIONS or a specific GOPTION to its default value.
DEVICE=*devicename*	The *devicename* chosen controls how the graph is constructed, including graph size, orientation, and fonts used. Examples include EMF (Enhanced Meta File, often the driver of choice for Windows-based systems), GIF, and PNG. Devices can be created and modified with PROC GDEVICE. For more information about device drivers, search for *About Device Drivers* in the online documentation.
TARGETDEVICE=*devicename*	Displays the output in the graphics window as it would appear in *devicename*. To avoid confusion, it is helpful to have both DEVICE and TARGETDEVICE point to the same

Option	Explanation
	devicename.
GSFNAME=*fileref*	Directs the graphics output to a specific *fileref* created earlier.
FTEXT=font_name	Sets font for all text in the graph (except TITLE1, which is specified with the FTITLE=*font_name* option). For further information about setting font options, see *Specifying Fonts in SAS/GRAPH Programs* in the SAS on-line documentation.
HTEXT=size <units>	Sets the height for all text in the graph (except TITLE1). By default, units are specified in cells, which are in turn specified by your device driver (but can be overwritten by the HPOS option and VPOS options, see the online documentation for details).
CTEXT=*color*	Specifies the text and border color (except for titles). There are many ways to specify colors; we will stick to the predefined SAS color names for this workshop. Available colors are determined by your *devicename*. Search for *Color-naming Schemes* in the online documentation for more information.
BORDER|NOBORDER	Tells SAS whether to draw a border around the graph

The global TITLE and FOOTNOTE statements include more options than are available in BASE SAS. These options are also applicable to the NOTE statement (used to specify text within a graphics area) as well. These options are applied immediately and can be reset even within a single TITLE, FOOTNOTE, or NOTE statement. Options include:

COLOR=*color*

FONT= font_name

HEIGHT= *size <units>:* CELLS are the default units, but CM (centimeters), IN (inches) PT (points) and PCT (percentage of graph area) are other options

JUSTIFY=CENTER | RIGHT | LEFT. CENTER is the default.

Other general SAS/GRAPH statements include the following global statements:

SYMBOL statements controls plotting symbols

AXIS statements create customized axis objects which can be referenced within a SAS/GRAPH procedure.

The LEGEND statement creates customized legend objects.

There is another very useful statement, NOTE, that writes text at a specified location within a procedure. This is not a global statement and must appear within the SAS/GRAPH PROC step in it is to be applied

PROC GPLOT SYNTAX
The basic syntax of PROC GPLOT is as follows:

PROC GPLOT *options*;

PLOT x * y /*options*;

RUN;

<additional PLOT statements>

<RUN;>

<QUIT;>

PROC PLOT includes RUN-group processing, which, during interactive sessions, allows for PLOT statement(s) to be run by submitting a RUN statement. The procedure is retained in memory for more RUN processing, or can be terminated by a QUIT statement or another PROC statement.

A LOOK AT OUR DATA
This fictitious data provides mean resting heart rate values for patients treated by a cardiovascular medication vs. those administered placebo. Each group contains mean values over six days.

```
Obs     treat     hr      date

  1       c        78     12346
  2       c        79     12347
  3       c        78     12348
  4       c        79     12349
  5       c        78     12350
  6       c        79     12351
  7       t        78     12346
  8       t        78     12347
  9       t        71     12348
 10       t        70     12349
 11       t        72     12350
 12       t        70     12351
```

ACTIVITY 1: SET UP OPTIONS AND CREATE INITIAL PLOT

The object of this activity is to create the data to be graphed, set up global options to chose a device driver, specify a graph size, and allow overwriting of graphs in the catalog, and to plot all data points. This initial plot should distinguish the average heart-rate values of the treatment patients from those of the control patients. The SAS code for this exercise appears below:

```
* Create the dataset and print it out;
<code omitted>

goptions reset=all❶ border❷
         device=png❸ target=png❹
         ;

FILENAME fileref1 "&path\png\gplot01.png";  ─────┐
FILENAME fileref2 "&path\png\gplot02.png";
FILENAME fileref3 "&path\png\gplot03.png";
FILENAME fileref4 "&path\png\gplot04.png";
FILENAME fileref5 "&path\png\gplot05.png";        ❺
FILENAME fileref6 "&path\png\gplot06.png";
FILENAME fileref7 "&path\png\gplot07.png";
FILENAME fileref8 "&path\png\gplot08.png";
FILENAME fileref9 "&path\png\gplot09.png";  ─────┘

***Create a graph that distinguishes points by the value of treat
    (control vs. treatment;
goptions gsfname=fileref1; ❻
proc gplot data=hr;
    plot hr*date=treat; ❼
    title1 'Study Heart Rate Through Time';
run;
quit;
```

❶ Reset all options to their default values.

❷ Put a border around the graphics area.

❸ Use the PNG (Portable Network Graphics format) device to create the graph.

❹ Use the PNG (Portable Network Graphics format) device to view the graph on screen within SAS.

❺ Set up *fileref*s for each exercise, so output can be routed there (in addition to being viewed on screen).

❻ Send the output of the next plot to *fileref1*.

❼ Plot **hr*****date** for each value of **treat**, overlaying on same graph.

As with any SAS program, it is prudent to examine the LOG before reviewing the results. The following NOTE after the RUN statement of our GPLOT procedure step assures us that the graph was written to the external file we specified:

```
   NOTE:  RECORDS WRITTEN TO c:\workshop\ws133\png\gplot01.png
```

The following NOTE informs us that the twelve observations in the source dataset were used to construct the graph:

```
   NOTE: There were 12 observations read from the data set WORK.HR.
```

In order to view the graph within SAS, position your mouse to the left of the **Gplot: Study Heart Rate Through Time** item in the **Results** tab. That will expand the entry to include all of the plots produced by the **RUN** group associated with the TITLE1 Study Heart Rate Through Time, as shown in Figure 1, below. Then click on the only plot we created in that RUN group, **Plot of hr * date = treat**. The GRAPH window will appear as displayed in Figure 2, below:
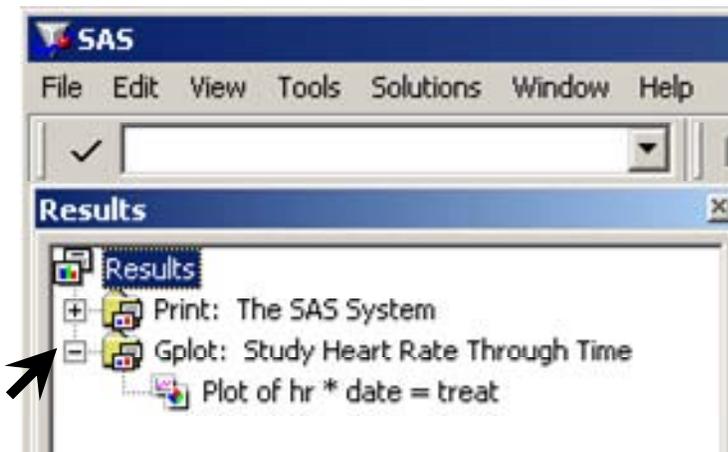
Figure 1: Expanding a GPLOT item to list all plots produced by a single RUN group
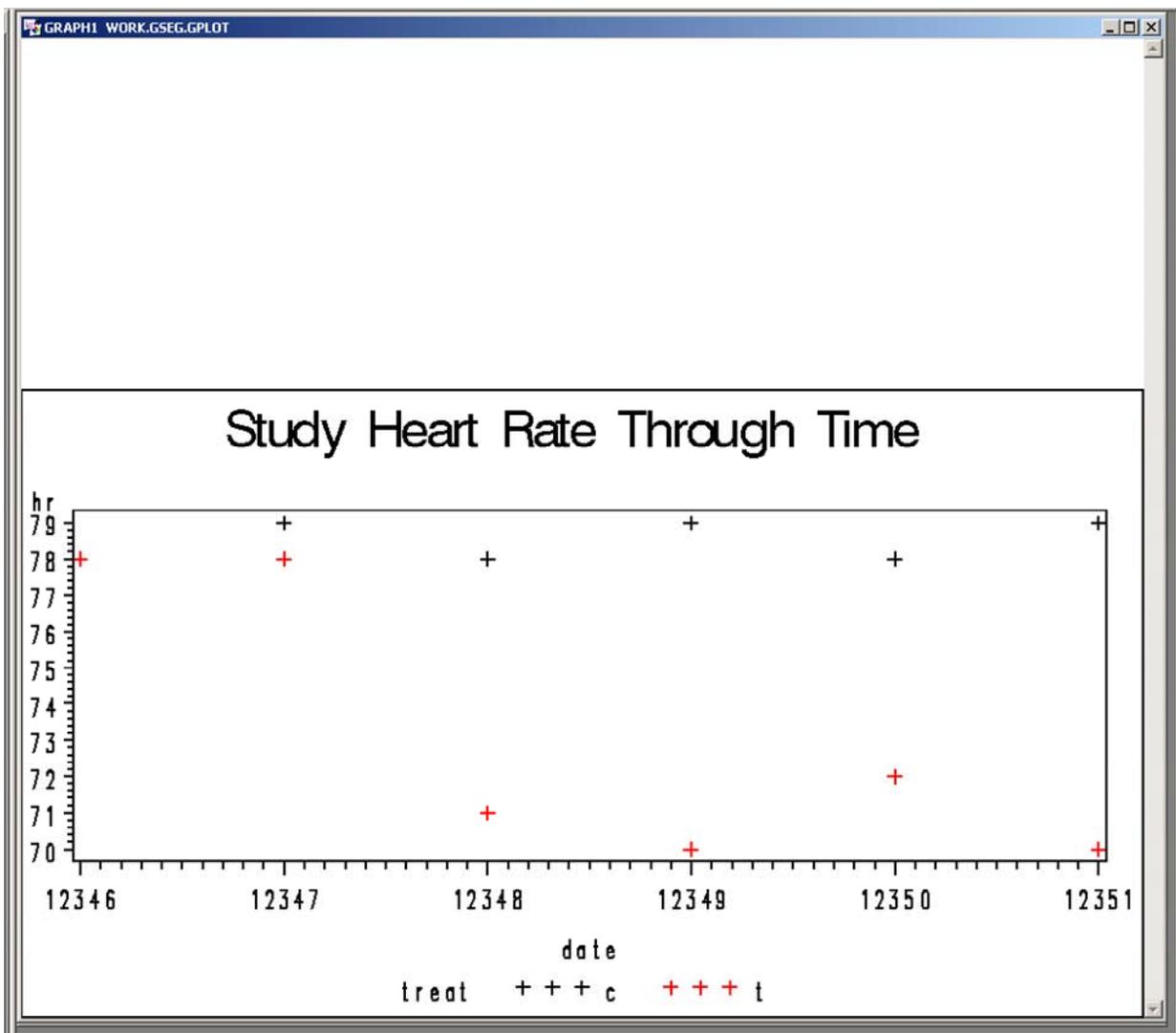

Figure 2: Viewing results in the GRAPH window

You will notice that the plot does not take up the entire graphics area. This is due to the maximum size of the driver we are using (PNG plotting size is 6.474 X 3.631). The graph itself is quite basic and could be modified for improved readability. The following exercises will accomplish just that.

ACTIVITY 2: ADDING SYMBOL AND FOOTNOTE STATEMENTS
Let the extreme makeover begin! First of all, in looking at our graph, we observe that it will be easier to distinguish between the treatment and control data if we plot data points for the control patient with blue dots and connected them with a blue lines, and plot treatment points with red squares connected by red lines. A footnote documenting source of these data would also be helpful. Following is the SAS code that accomplishes this, with the new statements displayed in bold.

```
symbol1 color=blue❶ line=1❷ value=dot❸    interpol=join❹;
symbol2 color=red❶  line=2❷ value=square❸ interpol=join❹;



proc gplot data=hr;
   plot hr*date=treat;
   title1 'Study Heart Rate Through Time';
   footnote1 'Protocol XYZ/2';
   run;
quit;
```

❶ Sets the color for both the symbol and any lines used to connect them.

❷ Sets the style of line used to connect the symbols. Documentation on line types is included in the version 8 online documentation under *the SYMBOL Statement*. . By default, all line values = 1(straight line).

❸ Sets the plotting symbol. Symbols values are also explained under the SYMBOL statement chapter in the online documentation

❹ Indicates how lines are to be joined, in this case with straight lines. Points are joined in the order they appear in the dataset, so it is wise to sort your X-axis variable before plotting.

The resulting graph appears in Figure 3, below. The curves are now nicely distinguished and our footnote appears, but we still have more dressing up to do. The SYMBOL1 statement options were applied to the control (treat='c') patients because 'c' precedes 't' in the sorting sequence.
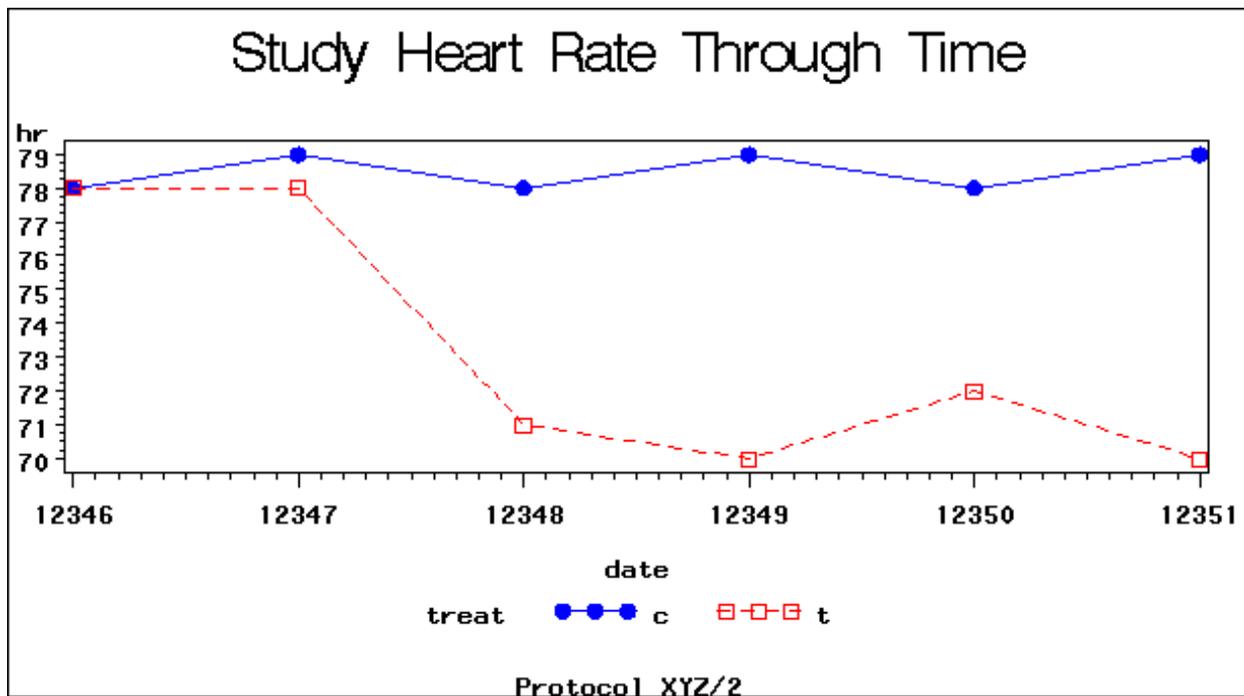

Figure 3. Activity 2 Results

ACTIVITY 3: CUSTOMIZING TEXT IN TITLE AND FOOTNOTE STATEMENT

Let's assume that our company's style conventions for figure titles dictate using the Times New Roman TrueType® font (as opposed to the swiss bold default) and that we are partial to the color blue. And let us further specify that the footnote should be left-justified and use the SAS/GRAPH simplex font, and that the name of the protocol (XYZ/2) should be 50% larger than the word protocol. The following code illustrates how this can be accomplished.

```
title1 f='Times New Roman'❶
       c=blue❷
       h=2❸ 'Study Heart Rate Through Time';

footnote1 f=simplex❹
          j=l❺ 'Protocol' h=1.5❻ 'XYZ/2';

proc gplot data=hr;
   plot hr*date=treat;
   title1 'Study Heart Rate Through Time';
run;
quit;
```

❶ Sets the font to Times New Roman TrueType font. If that font is not installed on the system, the SIMULATE SASFONT will be used instead.

❷ Set font color to blue (predefined SAS color).

❸ Sets the the height of title1 to 2 cells (cells are the default unit).

❹ Use the SAS simplex font for the text that follows.

❺ Left-justify the text that follows.

❻ Set size of the following text to 1.5 cells. Since the default footnote size is 1 cell, the following text will be 50% taller than that which preceded it.

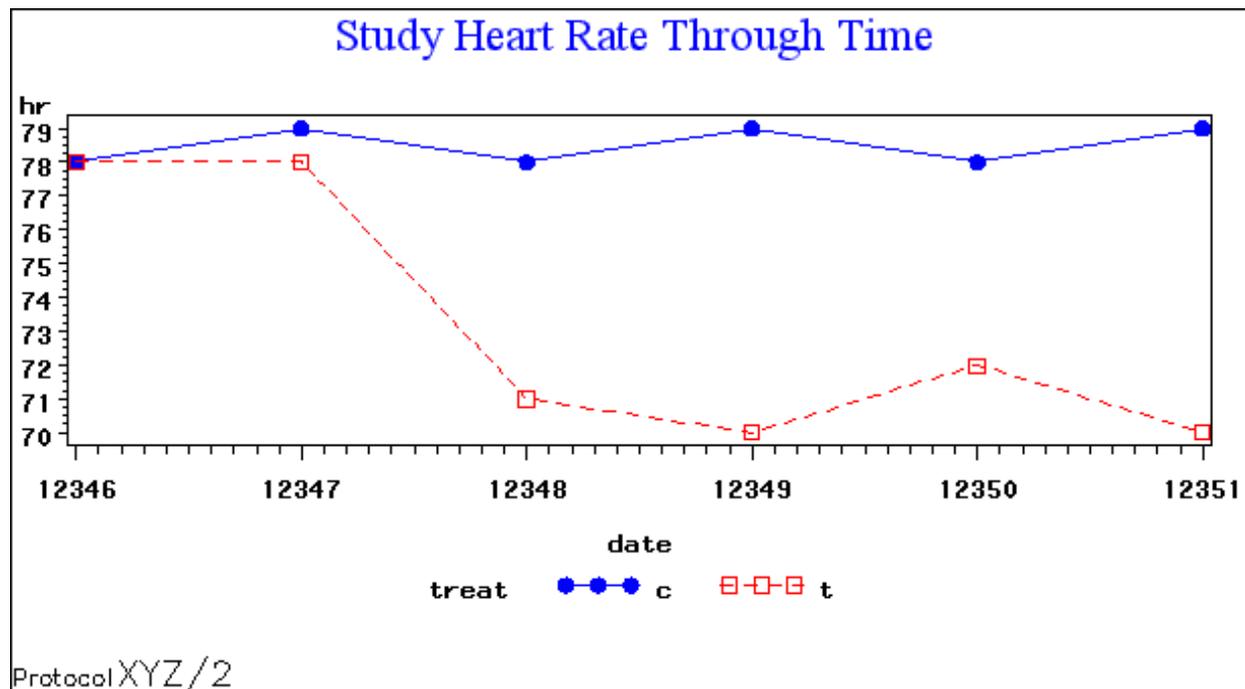Results for Activity 3 appear in Figure 4, below:

Figure 4. Activity 3 Results

ACTIVITY 4: SPECIFYING START AND STOP OF THE HORIZONTAL AXIS USING HAXIS= AND FORMAT DATE VALUES

The starting and stopping points as well as the major tick marks of the axes can be specified within PROC GPLOT using the HAXIS= and VAXIS= options, and SAS dates can be formatted by use of the FORMAT statement, as follows:

```
proc gplot data=hr;
    plot hr*date=treat /haxis= '19oct93'd to '25oct93'd;
    format date date9.;
run;
quit;
```

The resulting graph appears in Figure 5, below. Note that whereas we originally had 6 labels on the graph, we now have an additional label (19Oct1993) at the origin which has no data associated with it. The AXIS= option allows you to customize the scale of the graph to include values not present in your data or to hide values by not including them in the axis range.
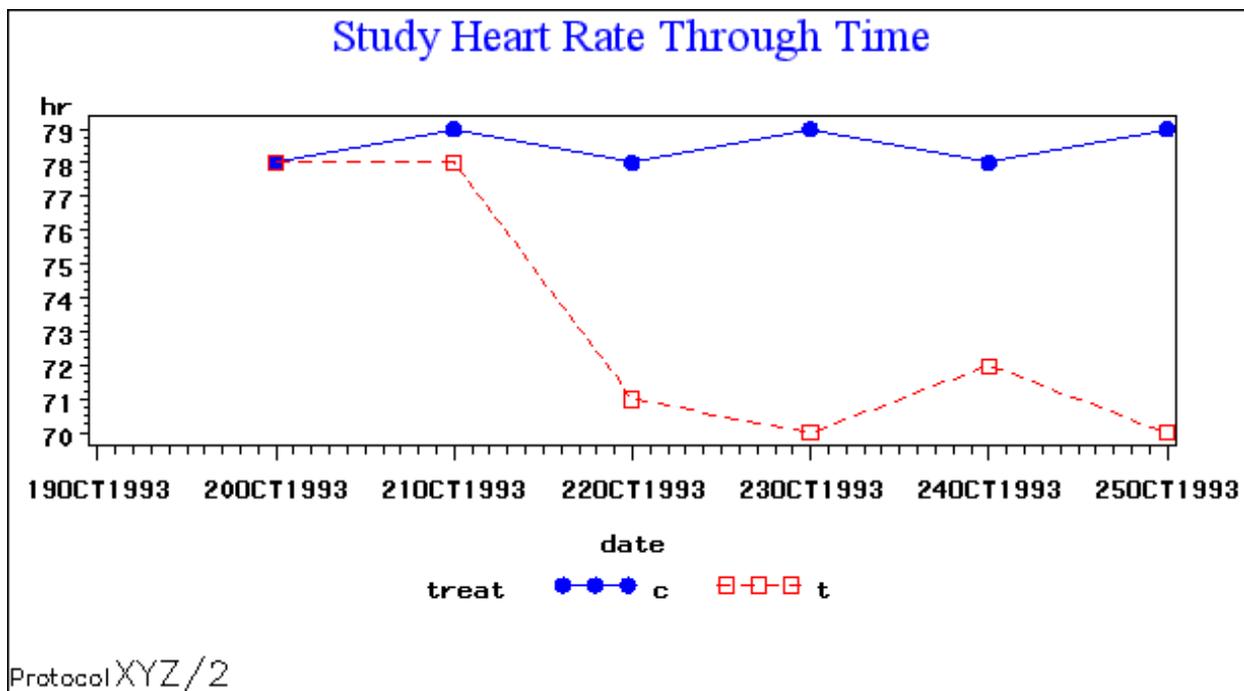

Figure 5. Activity 4 Results

ACTIVITY 5: USING AXIS STATEMENTS TO FURTHER CUSTOMIZE AXIS APPEARANCE
The graph produced by activity #4 is an improvement, but it would be even better to get more control over text in the
axis labels, include only one minor tick mark in the X-axis, and dispense with the minor tick marks between each of
the dates in the horizontal axis. This level of control is possible by using AXIS statements which can format virtually
any part of the AXIS. Code to create the axis objects and reference them in the PLOT statement of PROC GPLOT
appears below:

```
* Horizontal axis;
axis1 order=('20oct93'd to '25oct93'd)❶
      label=('visit date' h=1 f=swiss)❷
      minor=NONE❸
      value=(angle=45 h=.75 f=simplex)❹
;

***vertical axis;
axis2 order=(70 to 80 by 2)❺
      minor=(n=1)❻
      label=(a=90 j=c h=1.75 f=simplex 'heart rate' );❼
*-----------------------------------------------------------------------------;
proc gplot data=hr;
    plot hr*date=treat/haxis=axis1 vaxis=axis2; ❽
    format date date9.;
    run;
quit;
```

❶ Axis values should commence with 20oct93 and end with 25oct93, with every date in between being
displayed.

❷ Value of the axis label including height and font specifications.

❸ No minor tick marks should be displayed.

❹ Axis values should be rotated 45 degrees and have a height of .75 of a cell and be displayed in the
simplex font.

❺ Values should start at 70 and end at 80, with only the even values being displayed.

❻ Include one minor tick mark.

❼ Rotate the label 90 degrees (to line up with the axis), center it, provide it a cell height of 1.75, and specify
the font and label text.

❽ Associate each axis object with the appropriate axis in the PLOT statement.

The resulting graph appears in Figure 6, below. You will note that the vertical axis has been compressed, due to the
amount of space taken up by the horizontal axis labels. You can also observe that the font associated with the
vertical axis values has not changed, since no font was specified for them in the AXIS2 statement.
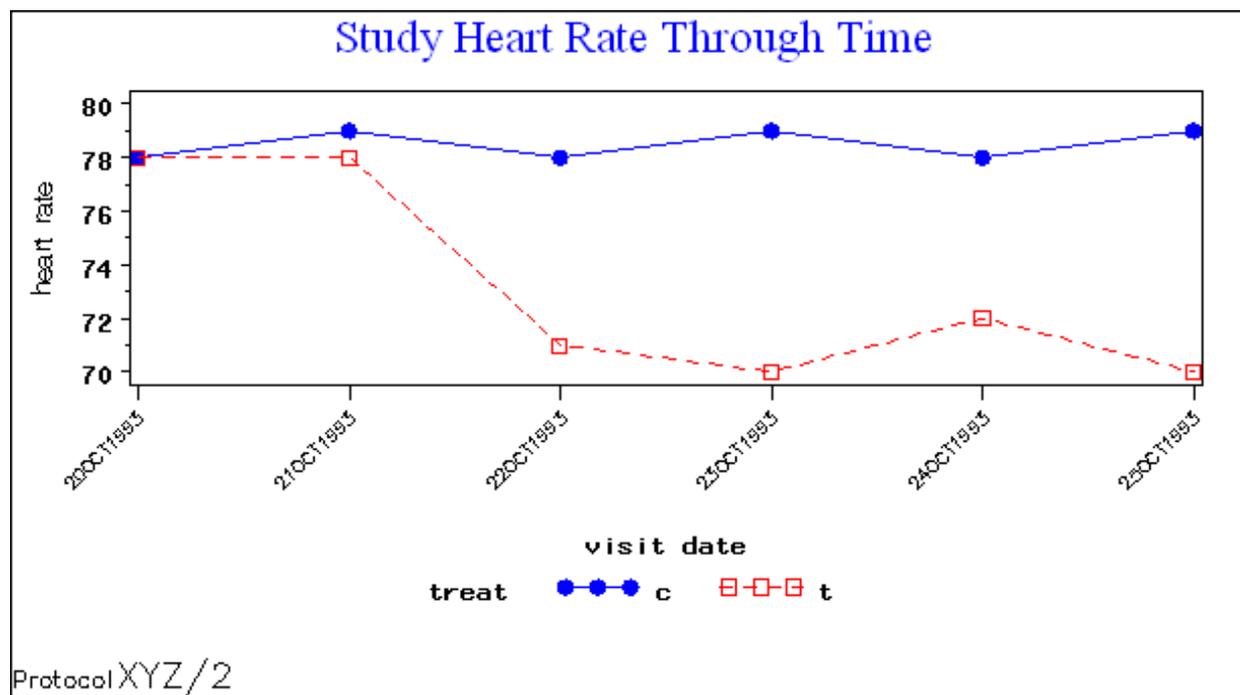
Figure 6. Activity 5 Results

ACTIVITY 6: CREATE A FRAMED LEGEND
Although a default legend appears for our graph, its appearance could be improved by framing the legend and positioning it to the side of the graph (to avoid any further compression of the vertical axis). We could also provide a descriptive label for the legend. This is easily accomplished with the text in bold, below:

```
legend1 position = (top right outside)❶ mode=reserve❷
        across=1❸ frame❹
        label=(f=swiss h=1 c=green 'Treatment Group' position=top
               justify=center)❺
        value=(t=1 c=blue 'Control' h=.75
               t=2 c=red '50 mg xyz' h=.75);❻


proc gplot data=hr;
    plot hr*date=treat/haxis=axis1 vaxis=axis2
                      legend=legend1;❼
    format date date9.;
run;
quit;
```

❶ The legend will be placed to the top right, outside the axis area.

❷ Reserve space on the page for the legend, thereby decreasing the size the X and Y axes.

❸ Only one entry should be printed per legend row.

❹ Frame the legend.

❺ Provide and format a label to document the legend.

❻ Provide and format explanatory text for each legend entry. The default entry order is determined by the sort order of the values of the variable TREAT.

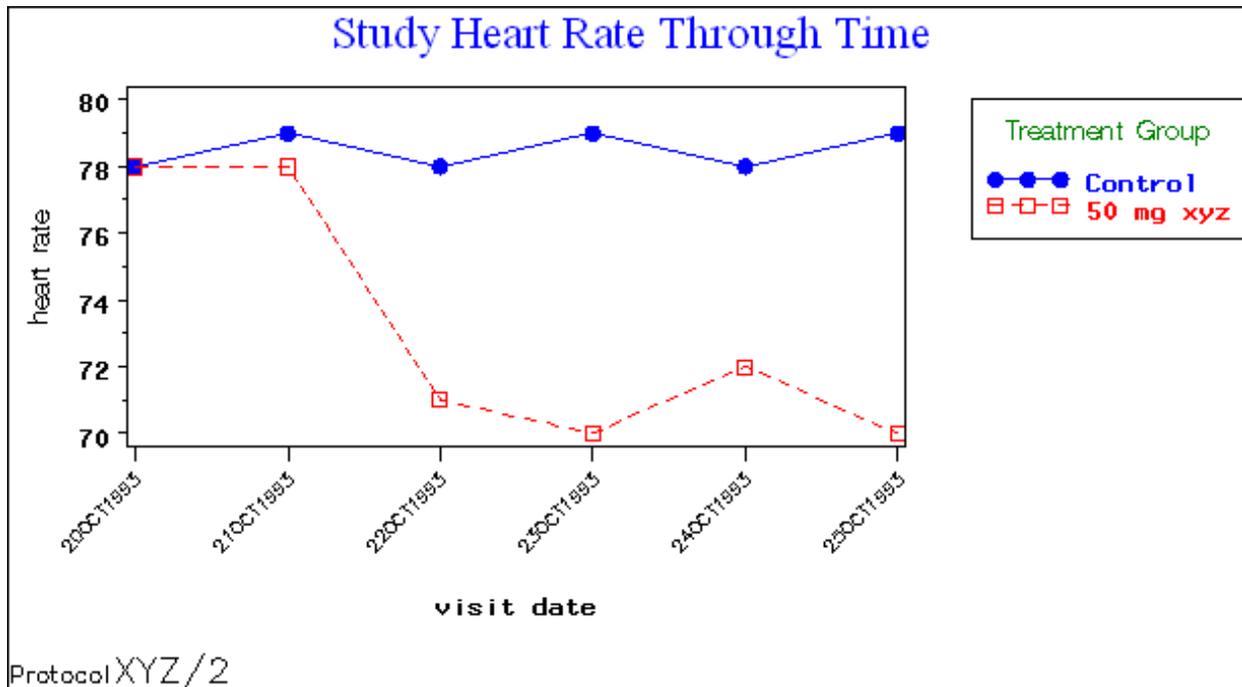❼ Associate the legend object with the PLOT legend= option.

Figure 7. Activity 6 Results

ACTIVITY 7: ADD A DASHED REFERENCE LINE AND LABEL IT.
Since this graph displays results for a clinical trial, it would be helpful to indicate when randomization occurred. This can easily be done by the addition of a reference line. The reference line itself is added as an option on the PLOT statement, but specific formatting of the label is performed in the AXIS statement.

```
axis1 order=('20oct93'd to '25oct93'd)
      label=('visit date' h=1 f=swiss)
      minor=NONE
      value=( angle=45 h=.75)
      reflabel=(h=.75 f=swiss 'Randomization');❶

proc gplot data=hr;
   plot hr*date=treat/haxis=axis1 vaxis=axis2
                      legend=legend1
                      href='21OCT93'D❷ lhref=4;❸
   format date date9.;
   run;
quit;
```

❶ Provide a label for the horizontal reference line and format it.

❷ Place the reference line on a specific value.

❸ Choose a line type for the reference line. (The same line types are available as for the L= option of the SYMBOL statement.)
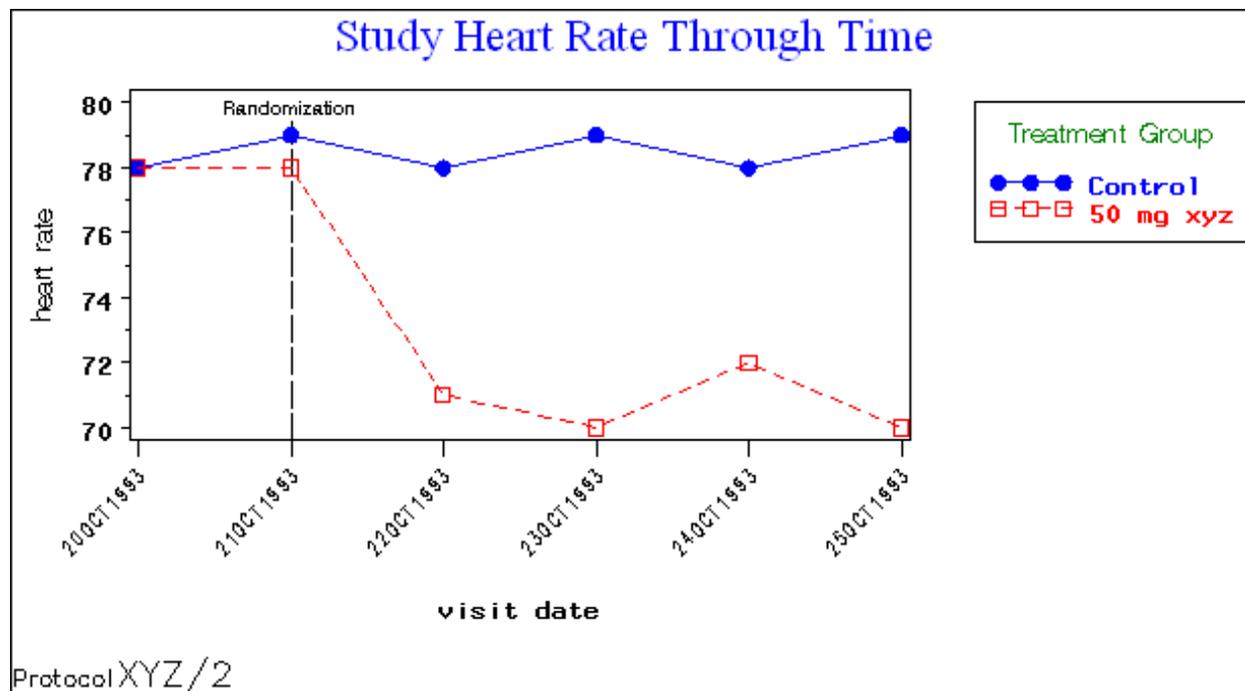

Figure 8. Activity 7 Results

ACTIVITY 8: ADD A NOTE TO DRAW ATTENTION TO A SPECIFIC REFERENCE POINT

The investigator of this study wants you to draw attention to the fact that immediately after randomization, there is a marked decrease in the heart rate of the group on medication. This can be accomplished with the NOTE statement. The tricky part of the note statement is getting used to the **m**ove= option, which specifies placement of the text you specify after it. This can be done relatively (in relationship to the last position written) or absolutely, in which the (x,y) coordinates reference the lower left corner of the graphics area. By default, the (x,y) coordinates reference cells, but can be customized to the other units discussed previously. More information on the NOTE statement can be found in the TITLE, FOOTNOTE, and NOTE Statements section of the SAS/GRAPH Software Reference online documentation.

```
proc gplot data=hr;
   plot hr*date=treat/haxis=axis1 vaxis=axis2
                      legend=legend1 href='21OCT93'D lhref=4;
   format date date9.;
   note f='Times New Roman / it' m=(26,13) h=.75 'Note immediate decrease in';
   note f='Times New Roman / it' m=(27,12) h=.75 'tx group after randomization.';
   run;
quit;
```

More information on the NOTE statement can be found in the TITLE*, FOOTNOTE, and NOTE Statements* section of the SAS/GRAPH Software Reference online documentation.
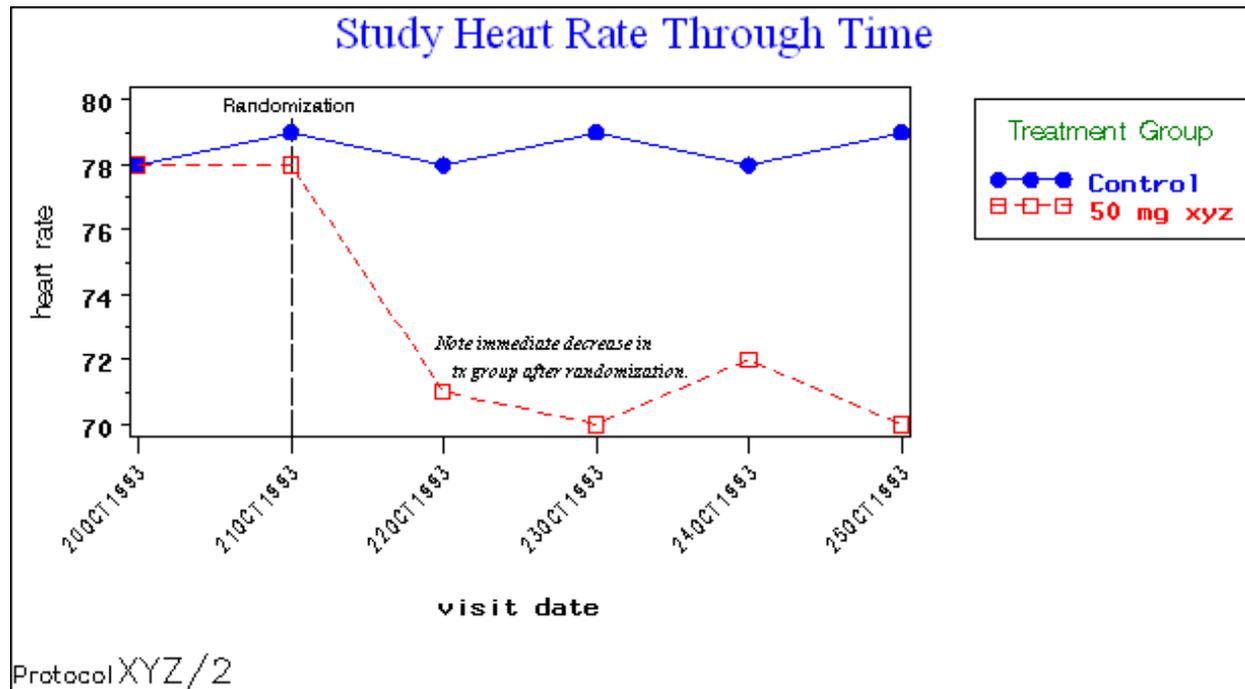

Figure 9. Activity 8 Results

ACTIVITY 9: SET GLOBAL OPTIONS FOR FTEXT, HTEXT, AND CTEXT, AND RUN THE LAST PLOT AGAIN

The point of this exercise is to illustrate how global options interact with those set by the graphics device driver and those set within the procedure.

```
goptions ftext='swiss' htext=1 ctext=black;


proc gplot data=hr;
   plot hr*date=treat/haxis=axis1 vaxis=axis2
                      legend=legend1 href='21OCT93'D lhref=4;
   format date date9.;
   note f='Times New Roman / it' m=(26,13) h=.75 'Note immediate decrease in';
   note f='Times New Roman / it' m=(27,12) h=.75 'tx group after randomization.';
   run;
quit;
```

Figure 10, below illustrates that all values with fonts that were not specified in the AXIS or LEGEND statements or within the GPLOT procedure have been changed to swiss with a height of 1. These include:

- Values of Y-axis

- X-axis label

- Values of the legend entries

Note that using a True-Type font TrueType font for the FTEXT= option on a GOPTIONS statement is not recommended for SAS 8.2. See http://support.sas.com/techsup/unotes/SN/003/003536.html for details.
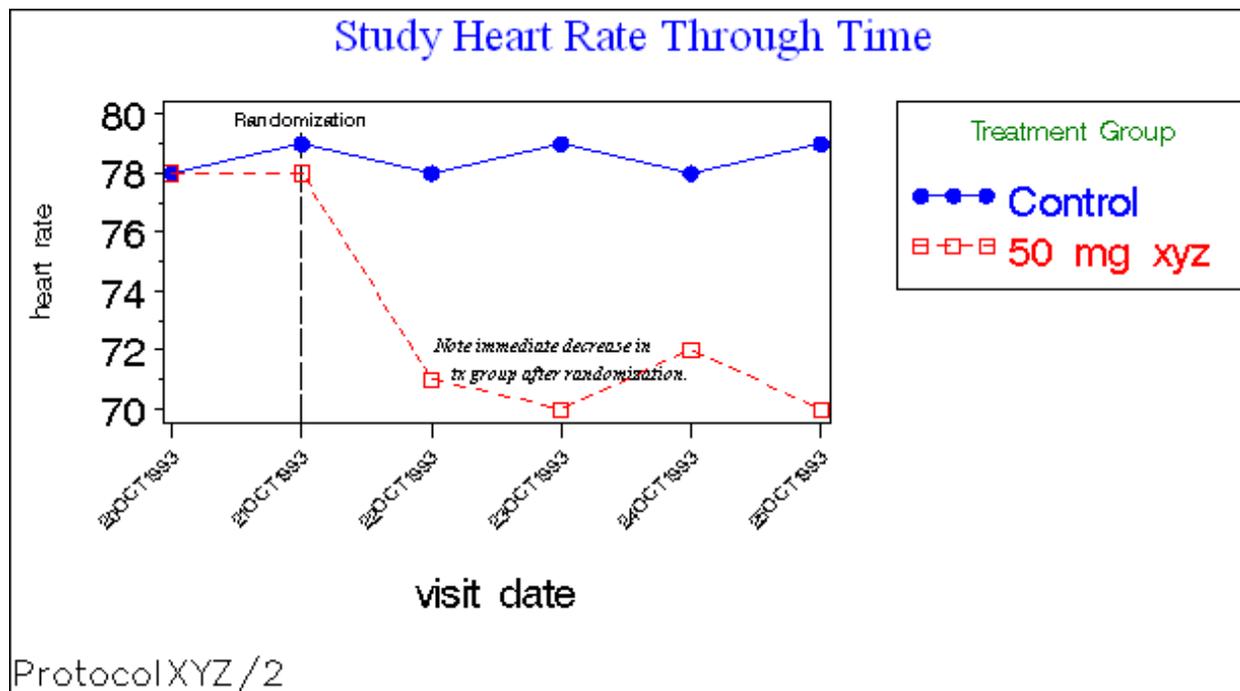

Figure 10. Activity 9 Results

ACTIVITY 10: BRINGING IN PNG GRAPH FILES TO MS POWERPOINT (EXTRA CREDIT!)
There are certain files created by SAS/GRAPH devices (PNG, EMF, BMP, GIF, etc.) that can be easily imported into
Windows applications. And while ODS destinations can help write directly to some Windows applications, MS
PowerPoint slides require another approach. Let's use the third graph we created, gplot03.png, to illustrate this
technique.

After opening MS PowerPoint, place your cursor in the body of a slide and go to Insert-→ Picture → From File, as in
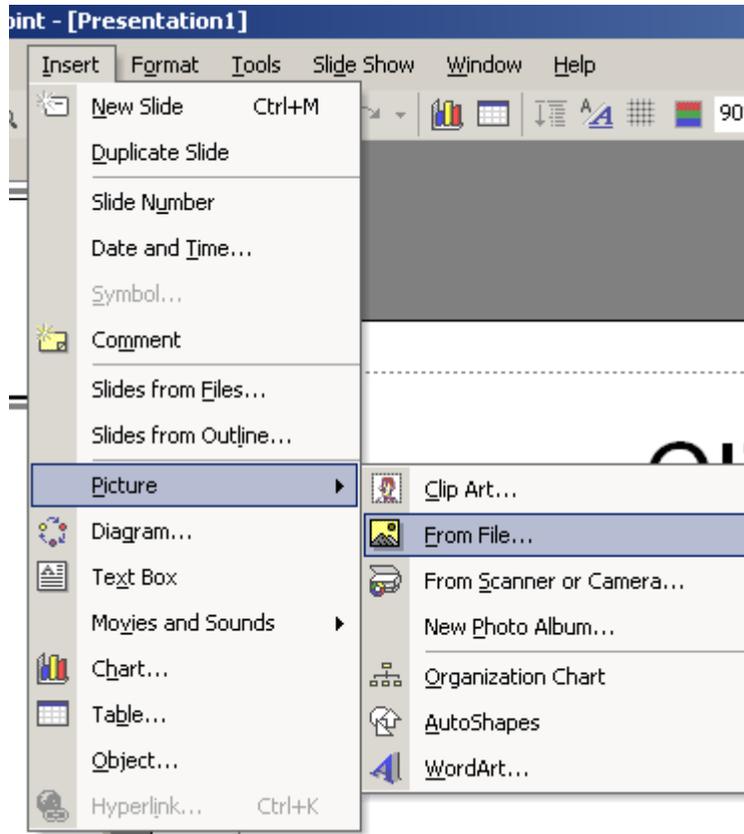Figure 11, below:

Figure 11. Inserting a picture create by SAS/GRAPH into MS PowerPoint

You will then be presented with the Insert Picture dialog box. Navigate to **C:\workshop\ws133\png** (the location
where we saved the files) and choose **gplot03.png**.

CONCLUSIONS
While even a full-day course does not provide enough time to delve into all the customizing options of SAS/GRAPH,
this workshop and paper has provided basic tools for customizing SAS/GRAPH output. The best way to improve
your skills is to put them to work right away, so feel free to install the workshop files on your computer and begin
exploring more options in the SAS/GRAPH documentation.

REFERENCES
SAS Online Doc Version 8. Cary, NC: SAS Institute Inc., 1999.

Carpenter, Arthur L and Shipp, Charles E. *Quick Results with SAS/GRAPH Software,* Cary, NC: SAS Institute Inc.,
1995.

An Introduction to Exporting SAS/GRAPH Output to Microsoft Office (TS-674).
http://support.sas.com/techsup/technote/ts674/ts674.html.

CONTACT INFORMATION
Your comments and questions are valued and encouraged. Contact the authors at:

Miriam Cisternas
Ovation Research Group
5051 Millay Court
Carlsbad, CA 92008
(760) 804-1946
mcisternas@ovation.org

Arthur L. Carpenter
California Occidental Consultants
P.O. Box 430
Oceanside, CA 92085-0430
(760) 945-0613
art@caloxy.com
www.caloxy.com

SAS and SAS/GRAPH are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries.
® indicates USA registration.

Other brand and product names are registered trademarks or trademarks of their respective companies.