SAS/GRAPH® 101

Mike Zdeb, University@Albany School of Public Health, Rensselaer, NY
Robert Allison, SAS Institute Inc., Cary, NC

ABSTRACT

SAS/GRAPH shares a trait with other parts of the SAS system. Upon first glance, the sheer volume of documentation makes it seem as if the learning curve might require a significant investment of time. In reality, very little knowledge of the 'bells and whistles' of SAS/GRAPH is needed before you can produce presentation-quality graphics. The following instructions start with very simple SAS code - where SAS makes the decisions as to the look of your graphics - and work their way gradually toward more customized output by adding more user control. PROC GPLOT is used throughout, though the same approach can be taken with any of the other SAS/GRAPH procedures. Hopefully, the knowledge you gain here can be used in experimenting with other procedures.

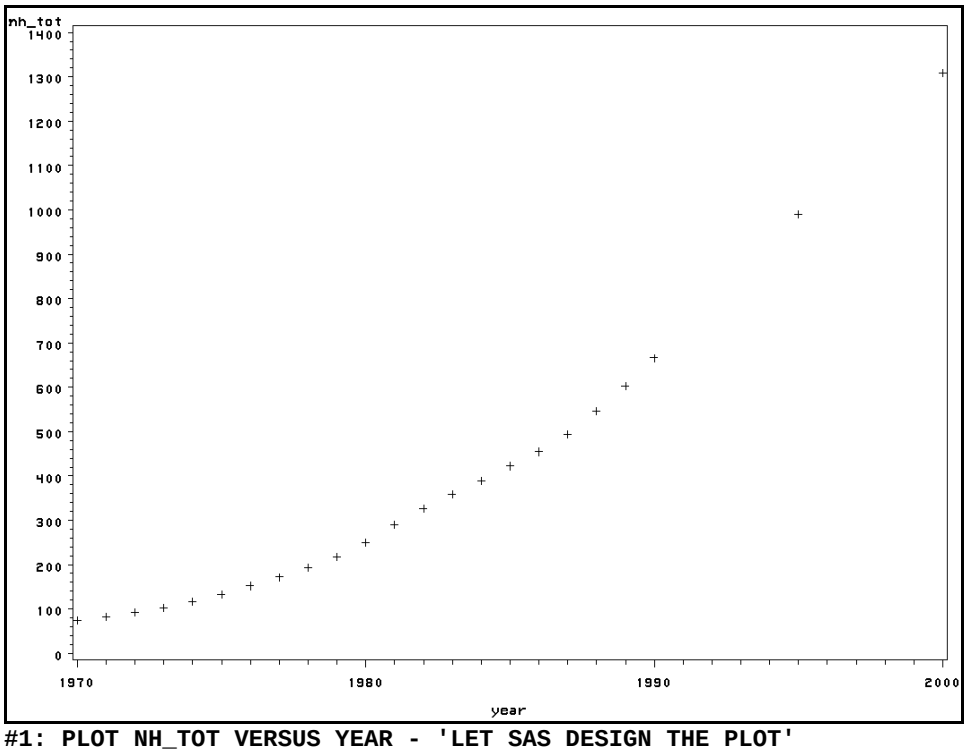
INTRODUCTION

The data sets to be used in this presentation comprise health care expenditures in the United States from 1970 through 2000 and in several other countries from 1970 through 1990. All data are shown in the appendix. We will start with data set #1 (named t112) and produce a series of plots that shows the health expenditures in the United States from 1970 through 2000.

BUILDING A PLOT STEP-BY-STEP

Plot #1 is 'minimalist' SAS/GRAPH, or 'here are my data, show me a picture.' PROC GPLOT is used with no options.

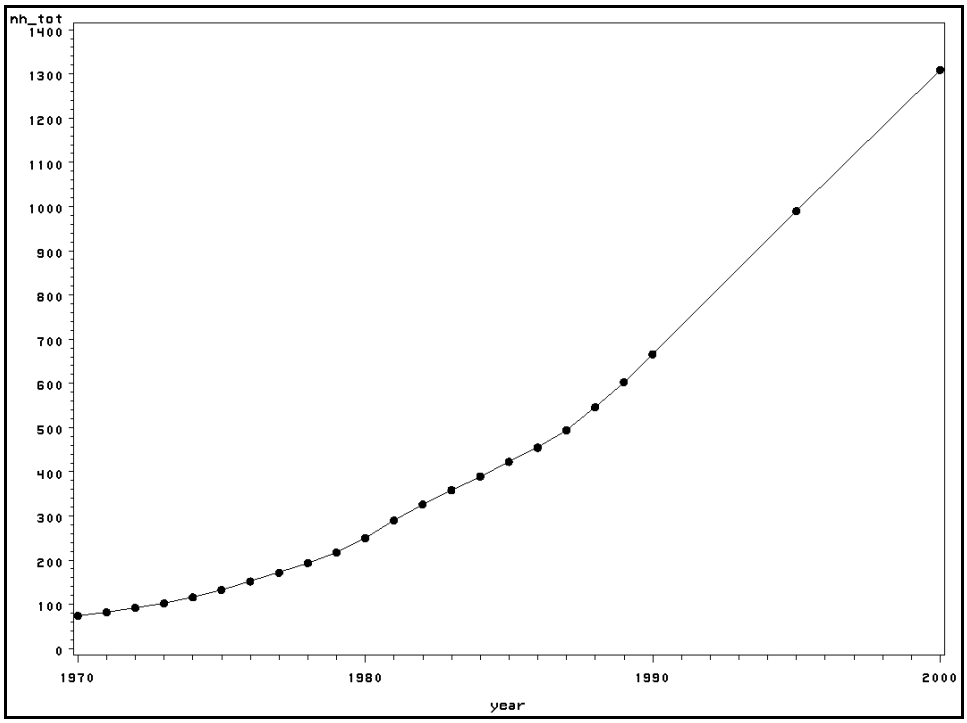
```
#1: PLOT NH_TOT VERSUS YEAR - 'LET SAS DESIGN THE PLOT'
proc gplot data=t112;
   plot nh_tot*year;
run;
```



#1: PLOT NH_TOT VERSUS YEAR - 'LET SAS DESIGN THE PLOT'

SAS/GRAPH has made a few decisions - among them are: the scales on both the X and Y axes; the font used for text (plus other text attributes - e.g., size, and color if you have an output device that supports color); the symbol to use for points on the graph (and not to connect them). You will also notice that the variable used first in the PLOT statement ends up as the Y-variable. Now that SAS has shown you a basic plot, you are ready to start taking control. The first change to be made is to the symbol that is used to show the points on the graph. The SYMBOL statement allows you not only to select a symbol to be plotted, but also to state whether or not to join the points, and if they are joined, how they're joined. We will stick to basics and select a 'dot' for the points, and merely 'join' them in a 'connect the dots' fashion.

```
#2: CHANGE THE PLOTTING SYMBOL AND JOIN THE POINTS
symbol1 value=dot interpol=join;

proc gplot data=t112;
   plot nh_tot*year;
run;
```



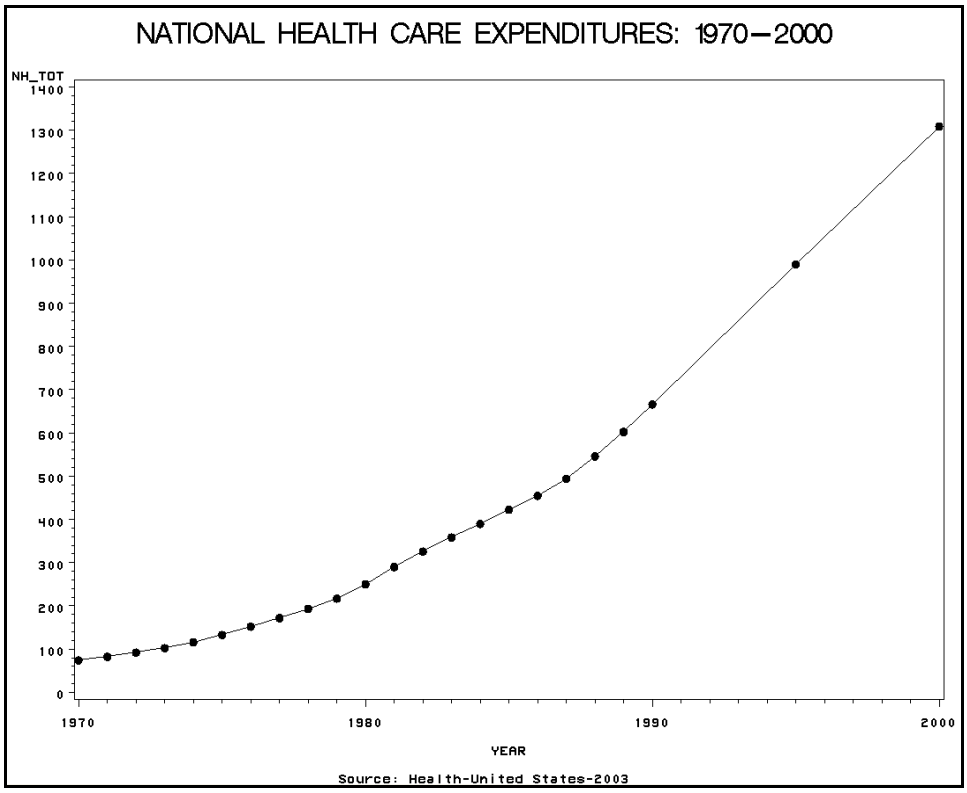
#2: CHANGE THE PLOTTING SYMBOL AND JOIN THE POINTS

It is now time to add a title (to describe the plot) and a footnote (to tell people where the data came from). Notice that you can shorten the names of the options used in the SYMBOL statement.

```
#3: ADD A TITLE AND FOOTNOTE
symbol1 v=dot i=join;

title "NATIONAL HEALTH CARE EXPENDITURES: 1970-2000";
footnote "Source: Health-United States-2003";

proc gplot data=t112;
   plot nh_tot*year;
run;
```

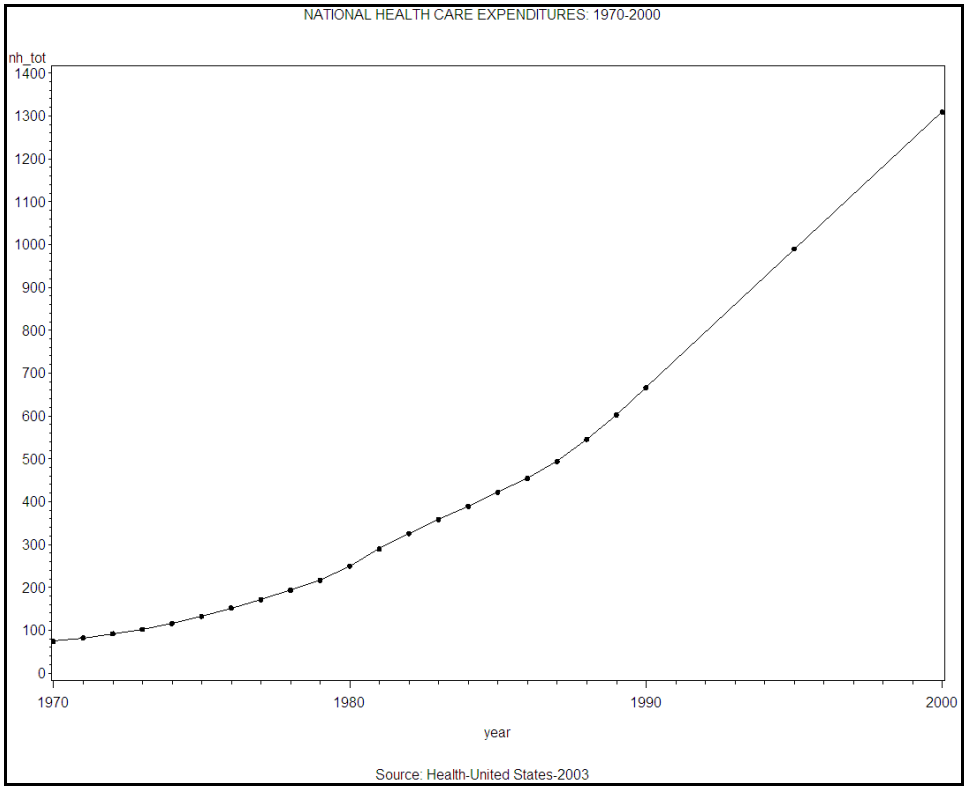
SAS has made more decisions. Fonts have been chosen and both the title and the footnote have been centered on the plot. It is time to choose a font for the text in the plot. We can either use a GOPTIONS statement and declare a font to be used for all text in the plot, or we can specify a font every time we tell SAS about text to be placed on the plot. The GOPTIONS method will be used, and a text height for all text in the plot will be specified.

```
#4: USE A GOPTIONS STATEMENT TO CONTROL THE  TEXT FONT AND HEIGHT
goptions ftext='Arial' htext=2 gunit=pct;

symbol1 v=dot i=join;

title "NATIONAL HEALTH CARE EXPENDITURES: 1970-2000";
footnote "Source: Health-United States-2003";

proc gplot data=t112;
   plot nh_tot*year;
run;
```



#3: ADD A TITLE AND FOOTNOTE

Note, the font chosen in #4 is a TrueType font (Arial) and is enclosed in quotes. There are also a number of SAS-supplied fonts, e.g. SWISS (not quoted when used) that is similar to Arial The SAS-supplied fonts are installed as part of SAS and will work on all output devices. TrueType fonts are not part of SAS. They must be present on the device where SAS is running and they are limited to certain output devices and also to certain operating systems.



#4: USE A GOPTIONS STATEMENT TO CONTROL THE TEXT FONT AND HEIGHT

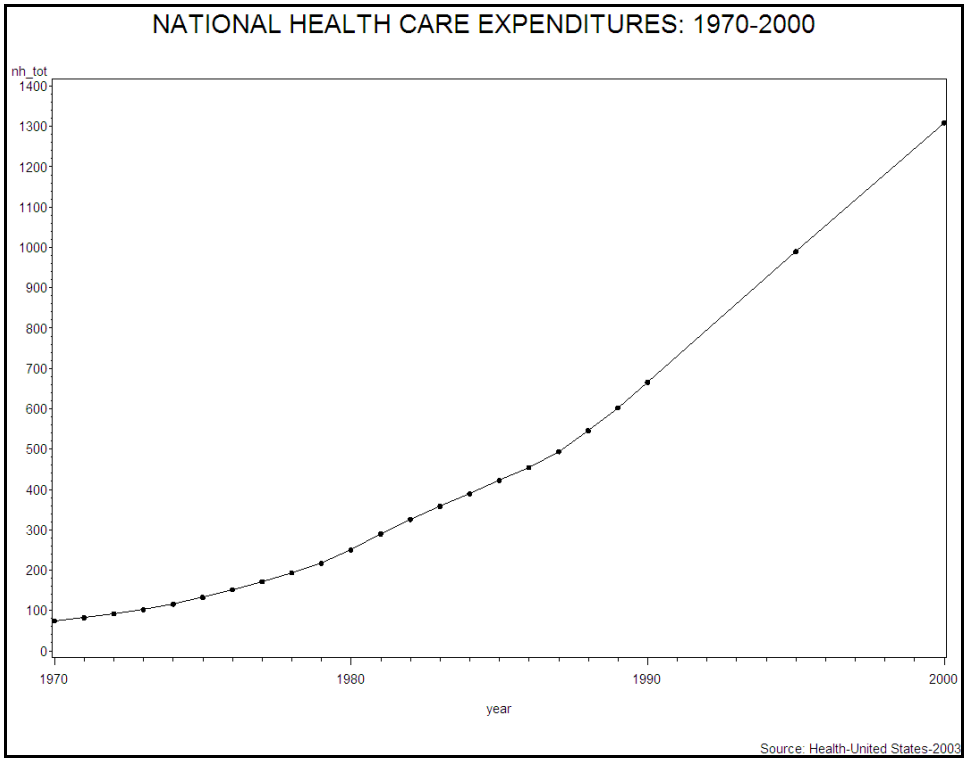
What is the extra option that is specified, GUNIT? An explanation requires a slight digression from our path of plot construction. The height of text (HTEXT) is specified as 2. The *question* is '2 WHAT' - inches, centimeters, points? The *answer* is, if no UNITS are specified, they are character cells, which should lead to the next question, what is a character cell? SAS/GRAPH divides each output device it can use (printer, plotter, computer screen) into a grid, a grid composed of character cells. The number of these cells varies from device to device. Before this discussion of cells goes too far, the easiest way to insure that graphics you design on one device (e.g., the monitor on your PC) will look similar if you change output devices (e.g., to a laser printer) is to specify that all UNITS used in your SAS/GRAPH job will be a percentage of the screen, paper, etc. This is done by specifying GUNIT=PCT (e.g., if your paper is 8 inches high, HTEXT=2 means text will be .16 inches high, 2% of 8 inches). That is more than enough for now about units - back to plot construction.

The GOPTIONS has changed all of the text height (including the symbol, also considered as text). The title should stand out, so we will override the GOPTIONS HTEXT=2 with a text height specified for the title. The footnote will also be moved to the lower right (right-justified) using the 'JUSTIFY=' option (also available for titles, but we will leave the title centered). Note, you can also specify unique attributes for the first title in the GOPTIONS statement using FTITLE and HTITLE.

#5: CHANGE THE HEIGHT OF THE TITLE - RIGHT JUSTIFY THE FOOTNOTE

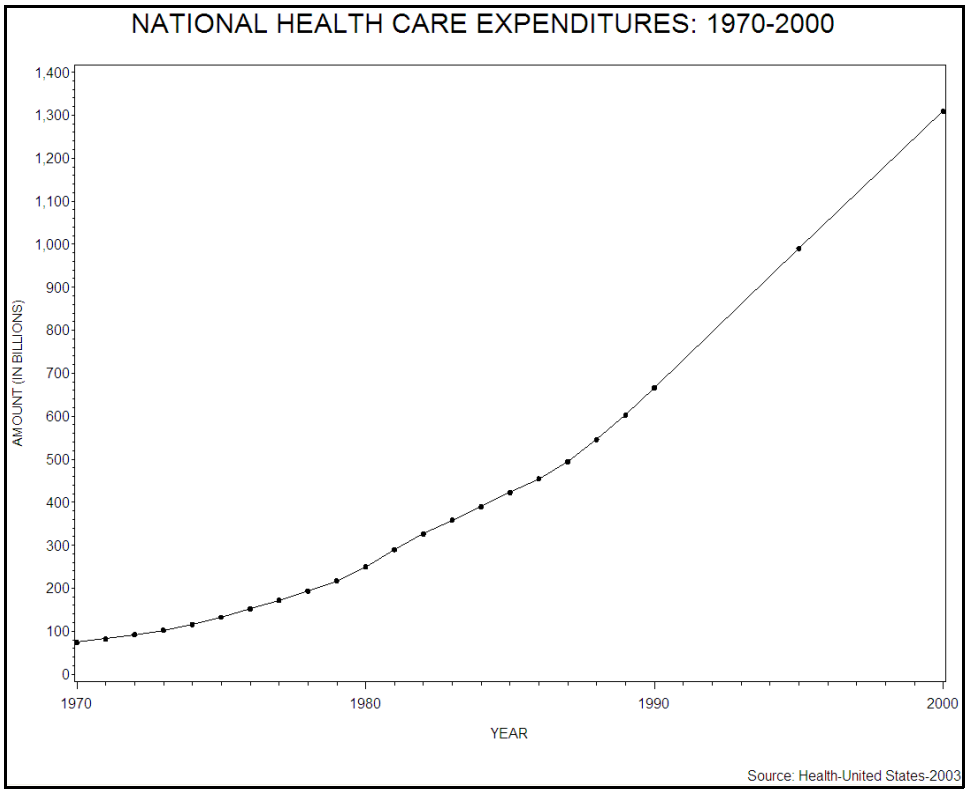
```
goptions ftext='Arial' htext=2 gunit=pct;

symbol1 v=dot i=join;

title height=4 "NATIONAL HEALTH CARE EXPENDITURES: 1970-2000";
footnote justify=right "Source: Health-United States-2003";

proc gplot data=t112;
   plot nh_tot*year;
run;
```

Now that the line, title, and footnote look OK, it is time to work on the appearance of the X and Y axes. We can control just about any aspect of the axes. The first thing we will do is change the labels. The X-axis label of 'year' would look better in upper case and the Y-axis might look better if it were labeled 'AMOUNT (IN BILLIONS)' - actually we would all be better off it said '(IN THOUSANDS),' but that is a problem that is not solved by any SAS/GRAPH option or procedure. We will have to settle for the changes we can make with an AXIS statement. Note, if variables used in the PLOT statement are labeled, the labels are used in the plot.



#5: **CHANGE THE HEIGHT OF THE TITLE - RIGHT JUSTIFY THE FOOTNOTE**

```
#6: CHANGE THE Y-AXIS LABEL - BOTH THE CONTENT AND ANGLE
options validvarname=upcase;
goptions ftext='Arial' htext=2 gunit=pct;

symbol1 v=dot i=join;

axis1 label=(angle=90 "AMOUNT (IN BILLIONS)");

title h=4 "NATIONAL HEALTH CARE EXPENDITURES: 1970-2000";
footnote j=right "Source: Health-United States-2003";

proc gplot data=t112;
   plot nh_tot*year/vaxis=axis1;
   format nh_tot comma.;
run;
```



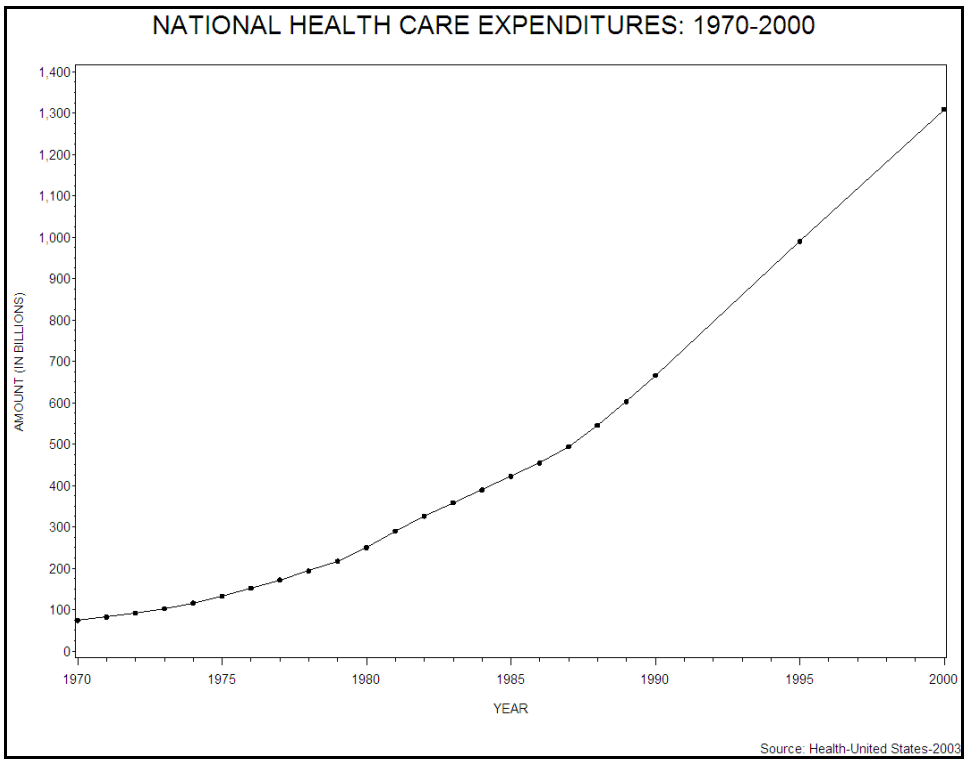
#6: CHANGE THE Y-AXIS LABEL - BOTH THE CONTENT AND ANGLE

Simply adding an AXIS statement to our SAS code will not change either the X or Y axis. The PROC must be told to use the AXIS definition, in this case for the vertical (or Y) axis. We have changed the label text and its angle. The ANGLE option determines the angle of the baseline on which the text rests. Note that an OPTIONS statement is used that changes the case of the X-label from 'year' to 'YEAR.' The FORMAT statement within PROC GPLOT inserts commas in the values on the Y-axis. The JUSTIFY option in the FOOTNOTE statement was shortened to J and the HEIGHT option in the TITLE statement was shortened to H.

We will now change another attribute of the axes, the number of tick marks and number of labels printed for axis values.

```
#7: ADDITIONAL CHANGES OF BOTH THE X AND Y AXES
options validvarname=upcase;
goptions ftext='Arial' htext=2 gunit=pct;

symbol1 v=dot i=join;

axis1 label=(angle=90 "AMOUNT (IN BILLIONS)") minor=(n=3);
axis2 order=(1970 to 2000 by 5) minor=(n=4);

title h=4 "NATIONAL HEALTH CARE EXPENDITURES: 1970-2000";
footnote j=right "Source: Health-United States-2003";
```

```
proc gplot data=t112;
   plot nh_tot*year/vaxis=axis1 haxis=axis2;
   format nh_tot comma.;
run;
```



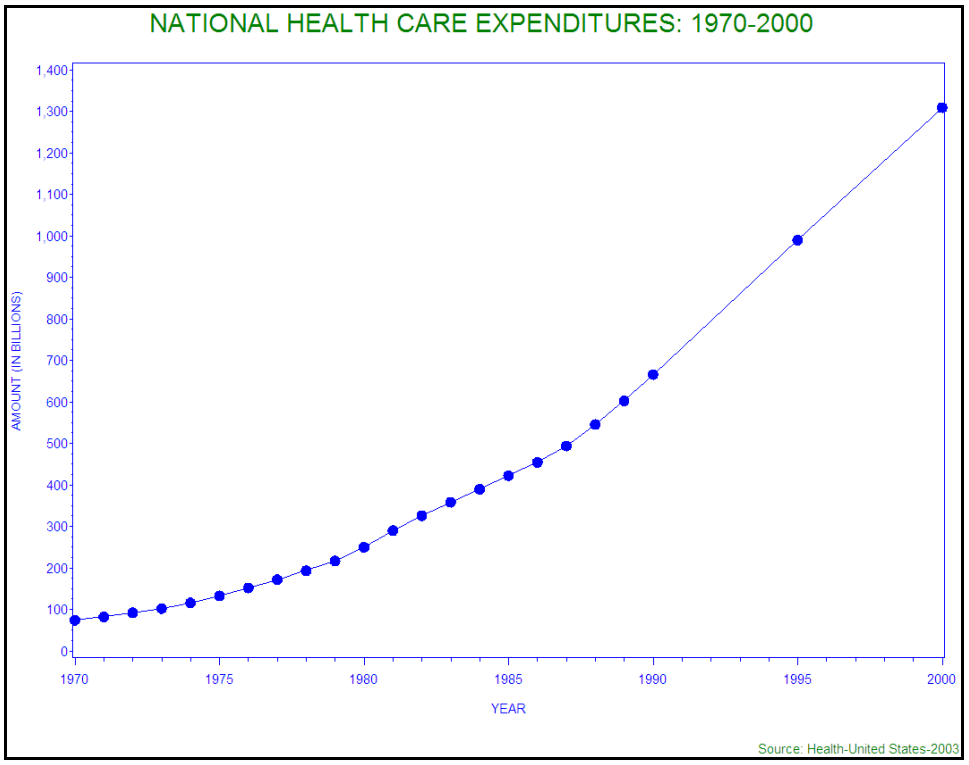
#7: ADDITIONAL CHANGES OF BOTH THE X AND Y AXES

On the vertical axis, we add a minor tick mark at every 25 billion dollars. On the horizontal axis, we change the number of years that are printed on the plot, every fifth year is labeled and the intervening years are noted by minor tick marks. The plot has come a long way, now we will add some color. Color is another attribute that can be controlled either within the GOPTIONS statement or within each statement (titles, footnotes, axes, etc.).

#8: ADD SOME COLOR USING GOPTIONS AND AXIS STATEMENTS

```
options validvarname=upcase;
goptions ftext='Arial' htext=2 gunit=pct ctext=green;

symbol1 v=dot i=join c=blue h=2.5;

axis1 label=(angle=90 "AMOUNT (IN BILLIONS)") minor=(n=3) color=blue;
axis2 order=(1970 to 2000 by 5) minor=(n=4) color=blue;

title1 h=4 "NATIONAL HEALTH CARE EXPENDITURES: 1970-2000";
footnote1 j=right "Source: Health-United States-2003";

proc gplot data=t112;
   plot nh_tot*year/vaxis=axis1 haxis=axis2;
   format nh_tot comma.;
run;
```

We have used the CTEXT option with the GOPTIONS statement to set the text color to be used in the entire plot. In the SYMBOL statement, we have specified a color and size (you can see that the symbol color affects both the dots and the line connecting the dots). What happened to the color of the axis labels and labels for axis values? They are now blue since the CAXIS option in an AXIS statement affects both the axis and any text associated with the axis. We can make the axes and tick marks blue, but leave all text associated with the axes green by moving the color specification for the axes from the AXIS statement to the PLOT statement in PROC GPLOT.



#8: ADD SOME COLOR USING GOPTIONS AND AXIS STATEMENTS

#9: CHANGE THE COLOR OF THE AXES, LEAVE ALL AXES TEXT GREEN

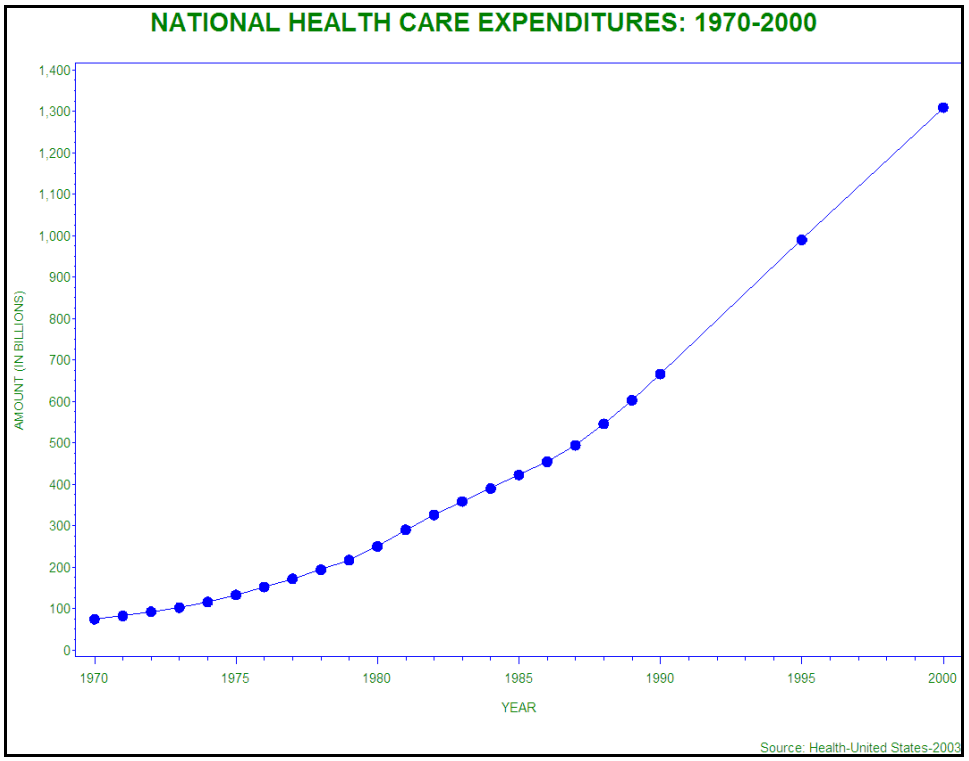
```
options validvarname=upcase;
goptions ftext='Arial' htext=2 gunit=pct ctext=green;

symbol1 v=dot i=join c=blue h=2.5;

axis1 label=(angle=90 rotate=0 "AMOUNT (IN BILLIONS)") minor=(n=3) ;
axis2 order=(1970 to 2000 by 5) minor=(n=4) offset=(2,2);

title h=4 font='Arial/bo' "NATIONAL HEALTH CARE EXPENDITURES: 1970-2000";
footnote j=right "Source: Health-United States-2003";

proc gplot data=t112;
   plot nh_tot*year/vaxis=axis1 haxis=axis2 caxis=blue;
   format nh_tot comma.;
run;
```

Another way to change the color of both axes is to use the CAXIS option within PROC GPLOT. This color selection only changes the color of the axes themselves and leaves the text green (as chosen with the CTEXT option in the GOPTIONS statement). Note, a different font is used for the title, 'Arial/bo' or Arial bold. Either or both the attributes bold and italic can be added to TrueType fonts using slash followed by 'bo' or 'it'. For example Arial bold italic is specified as 'Arial/bo/it'. There are numerous other font attributes that can be changed in a similar manner, but we can leave that to a look at on-line help. One of the AXIS statements was also modified, AXIS2 that controls the appearance of the X-axis. The OFFSET option leaves some space on both ends of the axis between the tick marks and the graph borders,

In addition to total national health care expenditures, the data set also contains information as to how much money the federal government has spent on health care from 1970 through 2000. It would be interesting to see how the total amount spent on health has changed relative to government expenditures. We will leave all our options (symbols, colors, axes, etc.) unchanged and add another line to the plot ('...the plot thickens...').

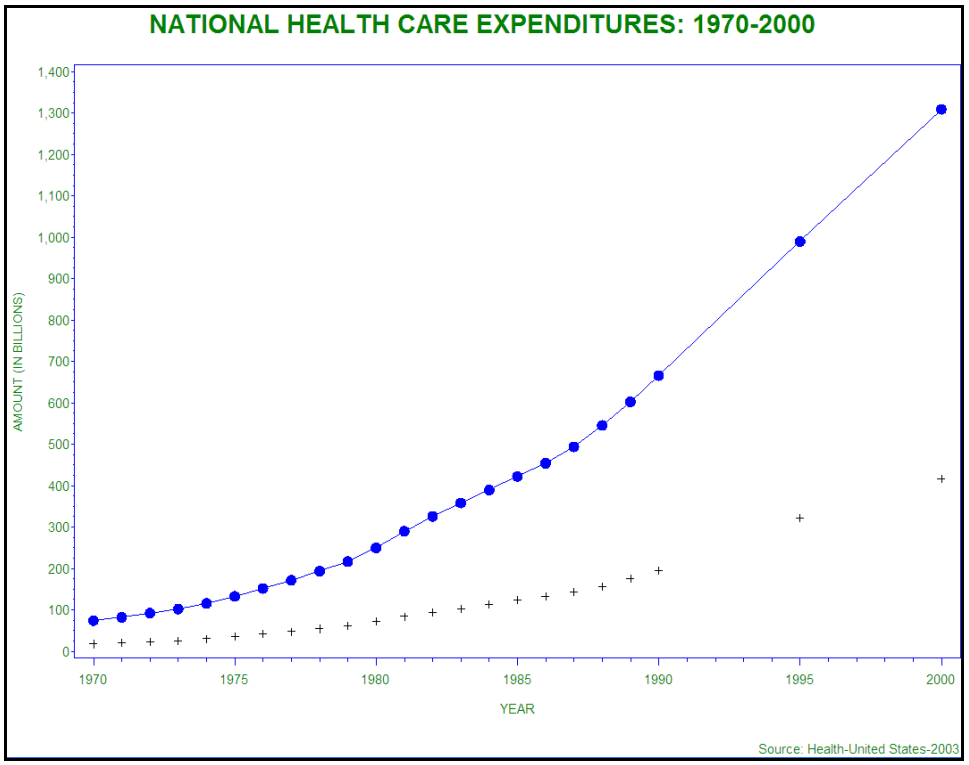
#10: LEAVE ALL OPTIONS SET AS BEFORE-ADD ANOTHER LINE TO THE PLOT

```
options validvarname=upcase;
goptions ftext='Arial' htext=2 gunit=pct ctext=green;

symbol1 v=dot i=join c=blue h=2.5;

axis1 label=(angle=90 rotate=0 "AMOUNT (IN BILLIONS)") minor=(n=3) ;
axis2 order=(1970 to 2000 by 5) minor=(n=4) offset=(2,2);
```



#9: CHANGE THE COLOR OF THE AXES, LEAVE ALL AXES TEXT GREEN

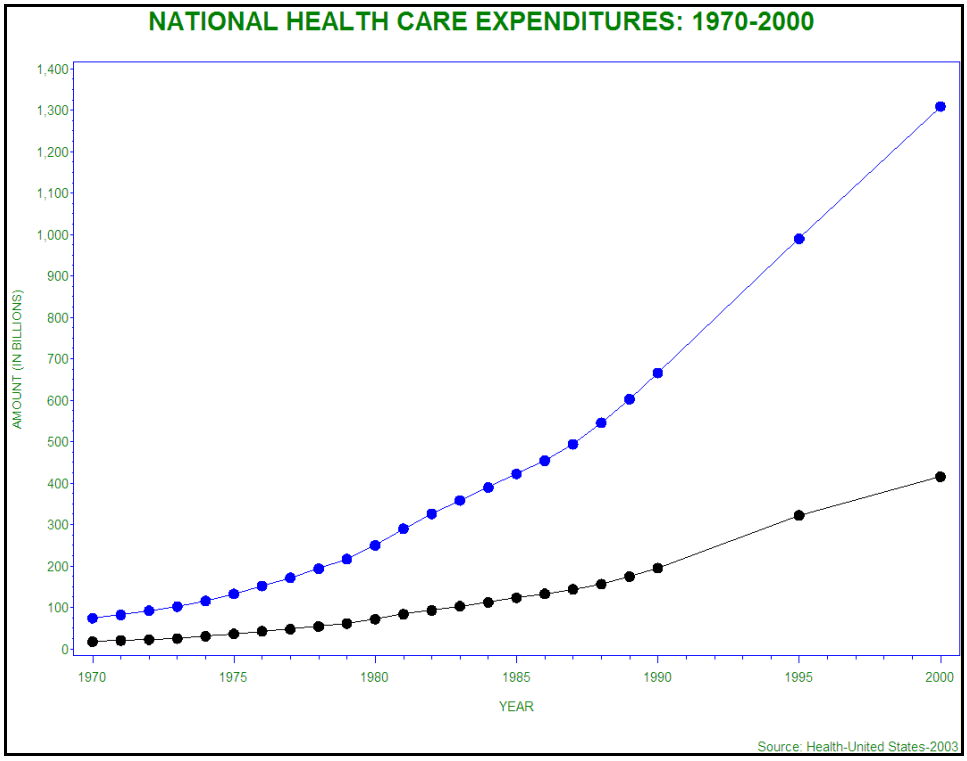
```
title h=4 f='Arial/bo' "NATIONAL HEALTH CARE EXPENDITURES: 1970-2000";
footnote j=right "Source: Health-United States-2003";

proc gplot data=t112;
   plot (nh_tot fg_tot)*year/overlay vaxis=axis1 haxis=axis2 caxis=blue;
   format nh_tot comma.;
run;
```



#10: LEAVE ALL OPTIONS SET AS BEFORE-ADD ANOTHER LINE TO THE PLOT

We can add another line merely by adding another Y-variable to the plot statement and by adding the OVERLAY option. Without the OVERLAY option, two Y-variables would produce two separate plots. Just as when we started building the plot, SAS/GRAPH has chosen its own symbol for the second line, and has chosen not to join the points (the default symbol). We need another SYMBOL statement to control the appearance of the second line. Note, we shortened FONT in the TITLE statement to F.

```
#11: ADD ANOTHER SYMBOL STATEMENT
options validvarname=upcase;
goptions ftext='Arial' htext=2 gunit=pct ctext=green;

symbol1 v=dot i=join c=blue h=2.5 l=1;
symbol2 v=dot i=join h=2.5 l=3;

axis1 label=(angle=90 rotate=0 "AMOUNT (IN BILLIONS)") minor=(n=3);
axis2 order=(1970 to 2000 by 5) minor=(n=4) offset=(2,2);

title h=4 f='Arial/bo' "NATIONAL HEALTH CARE EXPENDITURES: 1970-2000";
footnote j=right "Source: Health-United States-2003";

proc gplot data=t112;
   plot (nh_tot fg_tot)*year/overlay vaxis=axis1 haxis=axis2 caxis=blue;
   format nh_tot comma.;
run;
```

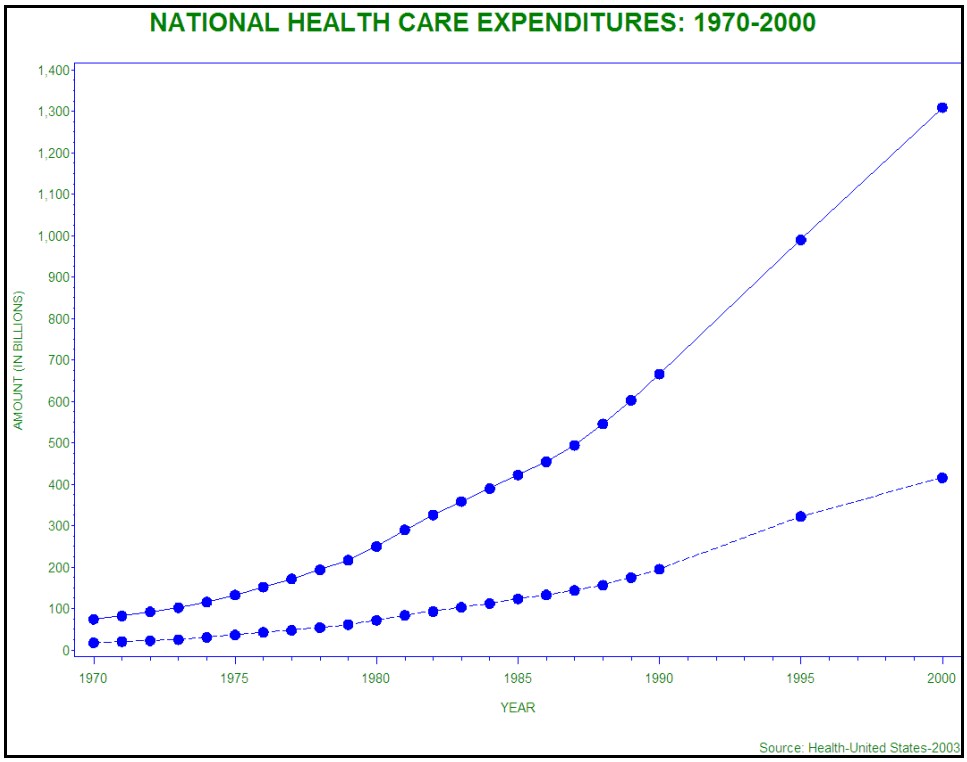


#11: ADD ANOTHER SYMBOL STATEMENT

There are a number of different ways to differentiate the lines on a plot. One way is to change the symbol VALUE (V=). Another way is to change the type of line used to connect the points by using the LINE (L=) option. SAS/GRAPH provides forty-six different line types (though a plot where you would need more that 3 or 4 of these is probably doomed from the start). We have chosen lines 1 and 3 (solid and dashed). The COLOR attribute was purposely left off the second SYMBOL statement to demonstrate a point. Once you define a symbol, SAS/GRAPH will cycle through a color list before using any subsequent SYMBOL statement. The second line in the plot has all the attributes of the first line, but it is a different color. If you plan to use different line types or different symbols (dots, squares, triangles) in a plot, you can specify a color for all symbols in the GOPTIONS statement using the CSYMBOL option. That prevents the color cycling that occurred in plot #11.

#12: USE GOPTIONS AND THE CYSMBOL OPTION

```
options validvarname=upcase;
goptions ftext='Arial' htext=2 gunit=pct ctext=green csymbol=blue;

symbol1 v=dot i=join h=2.5 l=1;
symbol2 v=dot i=join h=2.5 l=3;

axis1 label=(angle=90 rotate=0 "AMOUNT (IN BILLIONS)") minor=(n=3);
axis2 order=(1970 to 2000 by 5) minor=(n=4) offset=(2,2);

title h=4 f='Arial/bo' "NATIONAL HEALTH CARE EXPENDITURES: 1970-2000";
footnote j=right "Source: Health-United States-2003";

proc gplot data=t112;
   plot (nh_tot fg_tot)*year/overlay vaxis=axis1 haxis=axis2 caxis=blue;
   format nh_tot comma.;
run;
```



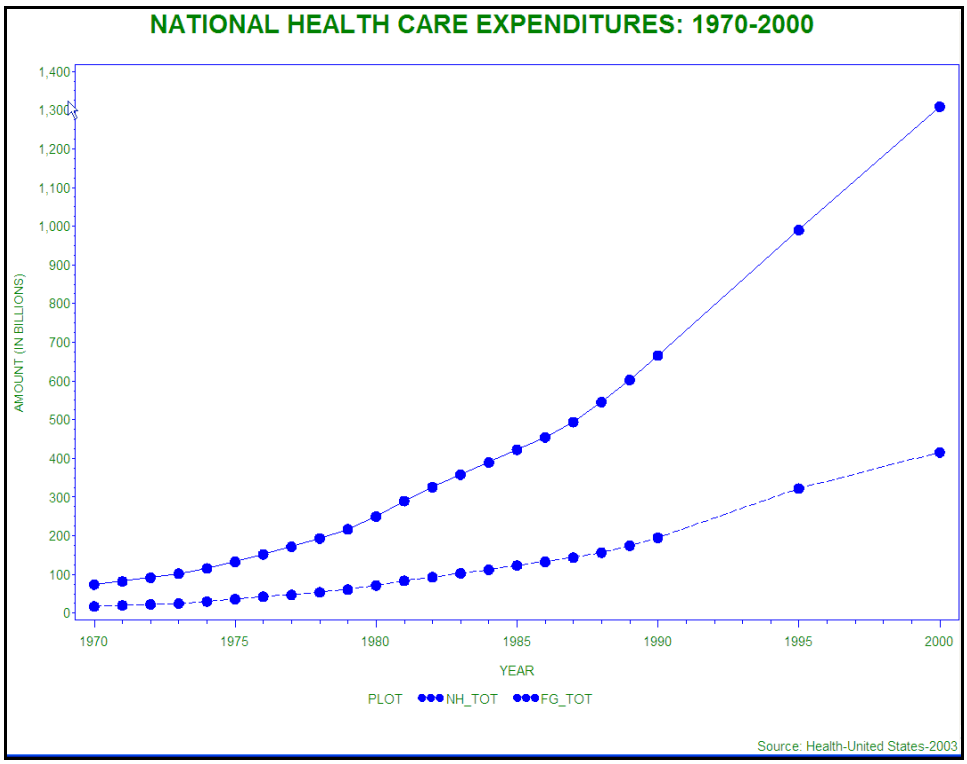
#12: USE GOPTIONS AND THE CYSMBOL OPTION

There is still something missing from the plot, a legend that identifies each line. The legend is not supplied automatically even though there is more than one line on the plot. To add the legend, a LEGEND option is added to the PLOT statement in PROC GPLOT.

#13: ADD A LEGEND

```
options validvarname=upcase;
goptions ftext='Arial' htext=2 gunit=pct ctext=green csymbol=blue;

symbol1 v=dot i=join h=2.5 l=1;
symbol2 v=dot i=join h=2.5 l=3;

axis1 label=(angle=90 rotate=0 "AMOUNT (IN BILLIONS)") minor=(n=3);
axis2 order=(1970 to 2000 by 5) minor=(n=4) offset=(2,2);

title h=4 f='Arial/bo' "NATIONAL HEALTH CARE EXPENDITURES: 1970-2000";
footnote j=right "Source: Health-United States-2003";
```

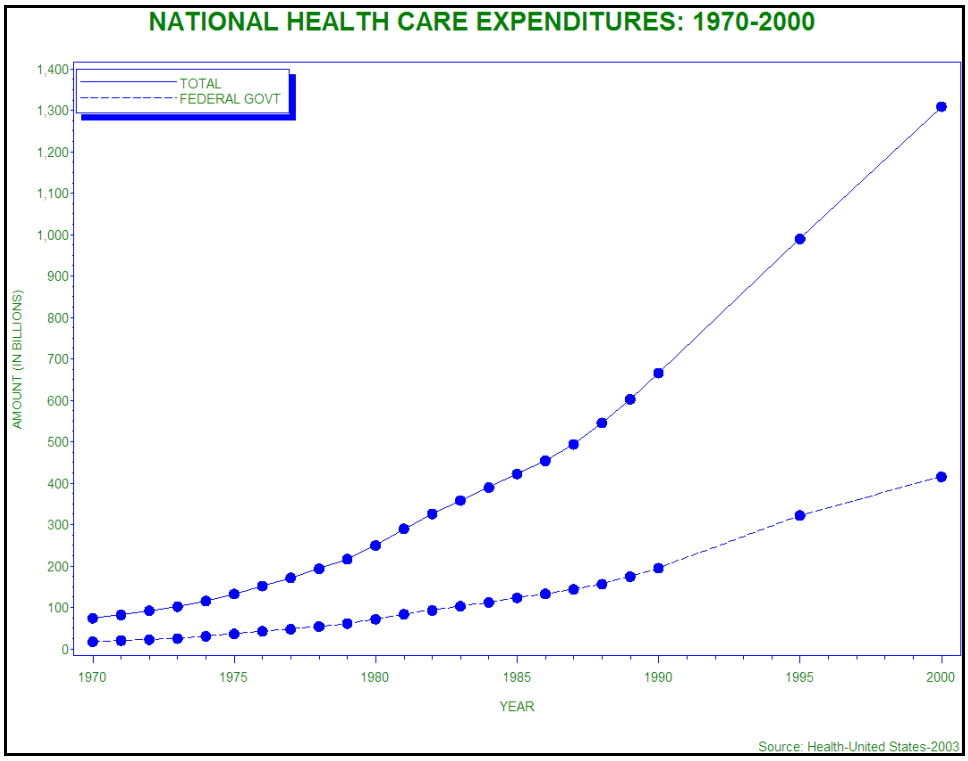
```
proc gplot data=t112;
   plot (nh_tot fg_tot)*year/overlay vaxis=axis1 haxis=axis2 caxis=blue legend;
   format nh_tot comma.;
run;
```



#13: ADD A LEGEND

The default legend is placed in the bottom center of the plot with the label 'PLOT'. The Y-variable names are used to label the two sets of lines/symbols. However, it is difficult to identify which line is which in the plot. Just as we modified the symbol and the axes, we can modify the legend with a LEGEND statement.

```
#14: MODIFY THE DEFAULT LEGEND
options validvarname=upcase;
goptions ftext='Arial' htext=2 gunit=pct ctext=green csymbol=blue;

symbol1 v=dot i=join h=2.5 l=1;
symbol2 v=dot i=join h=2.5 l=3;

axis1 label=(angle=90 rotate=0 "AMOUNT (IN BILLIONS)") minor=(n=3);
axis2 order=(1970 to 2000 by 5) minor=(n=4) offset=(2,2);

legend1 label=none value=(j=left "TOTAL" j=left "FEDERAL GOVT")
        mode=protect position=(top inside left)
        cborder=blue cshadow=blue
        across=1 shape=line(10);

title h=4 f='Arial/bo' "NATIONAL HEALTH CARE EXPENDITURES: 1970-2000";
footnote j=right "Source: Health-United States-2003";

proc gplot data=t112;
   plot (nh_tot fg_tot)*year/overlay vaxis=axis1 haxis=axis2 caxis=blue legend=legend1;
   format nh_tot comma.;
run;
```



#14: MODIFY THE DEFAULT LEGEND

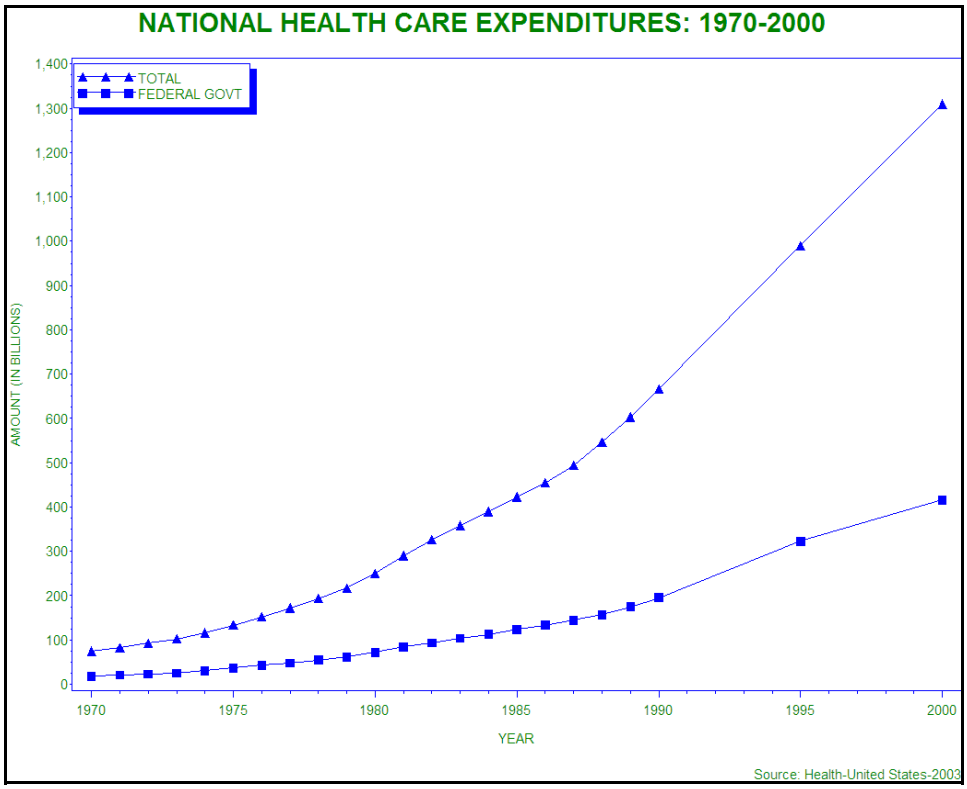
A number of options are used in the LEGEND statement. No label is needed since the title adequately explains what the plot is attempting to show (LABEL=NONE). The names of the variables are replaced using the VALUE option. Each time that a new JUSTIFICATION (J=) option is used with the VALUE, the text is associated with a new portion of the legend (in this case, a different line). The POSITION option moves the legend inside the plot and the MODE option insures that any line that might cross the legend box will be drawn behind the box. If you know something about your data (in this case, we all know that a plot of health care cost increase over time will start in the lower left and continue toward the upper right), you might not need to 'protect' the legend if you place it correctly. The last two options cause the values to be stacked (ACROSS=1) and to show long lines rather than symbols (SHAPE=).

ALTERNATIVE METHOD - TWO LINE PLOT

There is an alternative method for producing a two line (or multi-line) plot with PROC GPLOT. The data must be in a different format. There still must be X and Y variables. But a third variable is required that indicates what line on which to place each particular X-Y point. The data used for all the previous plots can be rearranged (see data set #2 in the Appendix) so the new data set contains three variables: YEAR; EXP, expenditure; EXPSRC; expenditure source (1=Total, 2=Federal Government). These data are used as follows.

#15: ALTERNATIVE METHOD - TWO LINE PLOT

```
options validvarname=upcase;
goptions ftext='Arial' htext=2 gunit=pct ctext=green csymbol=blue;

symbol1 f=marker v='C' i=join h=1.25;
symbol2 f=marker v='U' i=join h=1.25;

axis1 label=(angle=90 rotate=0 "AMOUNT (IN BILLIONS)") minor=(n=3);
axis2 order=(1970 to 2000 by 5) minor=(n=4) offset=(2,2);

legend1 label=none value=(j=left "TOTAL" j=left "FEDERAL GOVT")
        mode=protect position=(top inside left)
        cborder=blue cshadow=blue across=1 shape=symbol(6,1.25);

title h=4 f='Arial/bo' "NATIONAL HEALTH CARE EXPENDITURES: 1970-2000";
footnote j=right "Source: Health-United States-2003";

proc gplot data=t112new;
   plot exp*year=expsrc/vaxis=axis1 haxis=axis2 caxis=blue legend=legend1;
   format exp comma.;
run;
```



#15: ALTERNATIVE METHOD - TWO LINE PLOT

The PLOT statement within PROC GPLOT now specifies three variables. The variable used after the equals sign (EXPSRC) differentiates X-Y points between the two lines. It has two values, 1 and 2, so there are two lines. If it had three values, there would be three lines, etc. The SYMBOL statements now use characters from the MARKER font. The character 'C' produces a triangle, while 'U' produces a square. The SHAPE option within the LEGEND statement specifies that symbols be shown rather than lines as used previously. The numbers in the parentheses, 6 and 1.25, control the X-spacing and Y-height of the symbols in the legend.

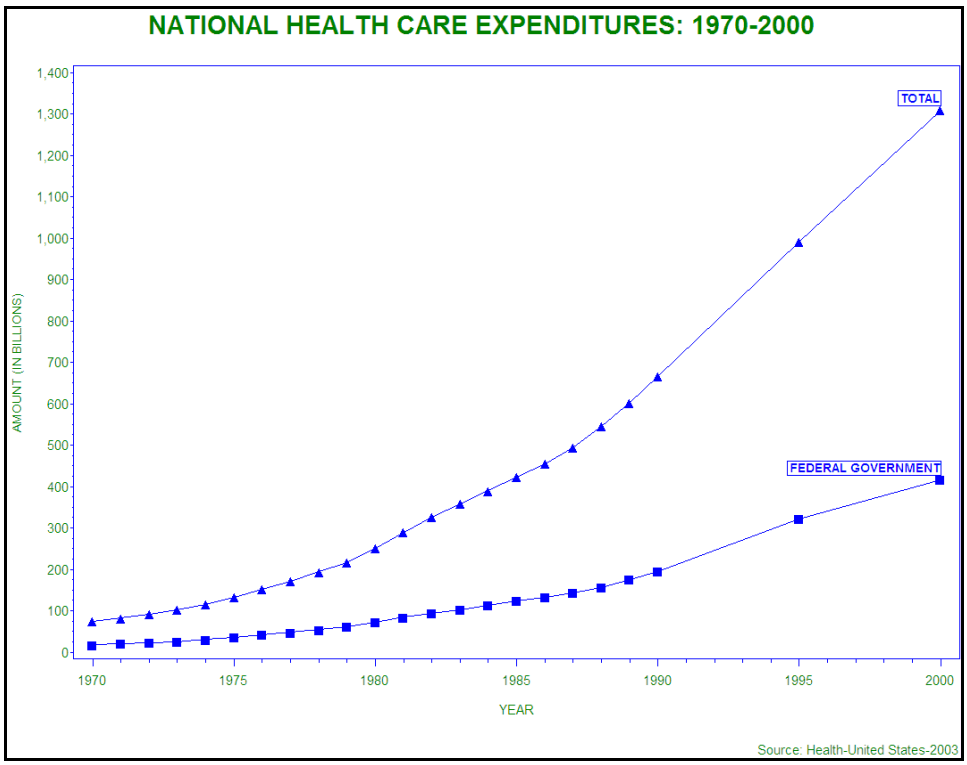
ANNOTATION

An introduction to SAS/GRAPH would not be complete without at least a mention of the ANNOTATE facility. You may have seen some very fancy charts that were produced by SAS/GRAPH and wondered 'How did he(she) ever get that bar chart overlaid on a map of Europe?' Well, even if you have not seen or wondered about that, you should at least know what ANNOTATE is. The ANNOTATE facility within SAS/GRAPH allow you to take a chart produced by one of the standard procedures and add anything you want anywhere on the chart. You can also construct charts entirely with ANNOTATE functions if you want to display data in some way that just is not possible with one of the SAS/GRAPH procedures. One way to think of the ANNOTATE facility is that it a text description of what you might do if you manually edited a graphic in a program such as Adobe Illustrator.

We will use ANNOTATE to add some text to one of our plots. When we produced a plot with two lines, we added a legend to tell anyone looking at our plot what data each line represented. There is a another way to distinguish between the two lines, label each line using ANNOTATE. If you are really new to SAS/GRAPH (or SAS in general), the following may seem a little obtuse. Do not worry, it's not just you, ANNOTATE is obtuse.

```
#16 USE ANNOTATION TO LABEL THE LINES
data my_labels;
   retain xsys ysys '2' function 'label'
          position '1' style "'Arial/bo'"
          color 'blue' cborder 'blue';
   set t112 end=last;
   if last then do;
      text=' FEDERAL GOVERNMENT '; x=year; y=fg_tot; output;
      text=' TOTAL '             ; x=year; y=nh_tot; output;
   end;
run;
```

```
goptions ftext='Arial' htext=2 gunit=pct ctext=green csymbol=blue;
options validvarname=upcase;

symbol1 f=marker v='C' i=join h=1.25;
symbol2 f=marker v='U' i=join h=1.25;

axis1 label=(angle=90 rotate=0 "AMOUNT (IN BILLIONS)") minor=(n=3);
axis2 order=(1970 to 2000 by 5) minor=(n=4) offset=(2,2);

title h=4 f='Arial/bo' "NATIONAL HEALTH CARE EXPENDITURES: 1970-2000";
footnote j=right "Source: Health-United States-2003";

proc gplot data=t112;
   plot (nh_tot fg_tot)*year/overlay vaxis=axis1 haxis=axis2 caxis=blue annotate=my_labels;
   format nh_tot comma.;
run;
```



#16 USE ANNOTATION TO LABEL THE LINES

It is difficult with only one example for you to understand the power (and intricacies) of the ANNOTATE facility. This example is just meant to show you one application. The first data step creates an annotate data set, MY_LABELS. That data set contains two observations and each observation contains enough information to place a label on the plot. If you are familiar with CNTLIN data sets, they must contain variables with specified names in order to be understood by PROC FORMAT. The same is true for ANNOTATE data sets. There are only certain variable names that make sense to SAS/GRAPH procedures when an annotate data set is used to enhance procedure output.

The first line after DATA MY_LABELS is a RETAIN statement. All the variables in that statement have allowable annotate data set names. They are assigned values that determine the placement and appearance of text on our plot. The easiest one to understand is FUNCTION (we are adding labels). Another easy one is COLOR. Notice that the we no longer use FTEXT or FONT. In their place we use STYLE to specify a font and also notice that when a TrueType font is used, it must appear within two sets of quotes. The variable POSITION determines the location where, around a given point, the text is placed. There are 15 allowable values and '1' places text to the upper-left of a point on the plot. The variable CBORDER causes a box to be drawn around the text in the given color. That explains everything in the RETAIN statement except for XSYS and YSYS. Those two variables define a coordinate system. For now, it is enough to understand that assigning both those variables a value of 2 allows you to use your data values (values of the X,Y variables in the plot) to place text on the plot.

You can probably ascertain from the data step SAS code that we used the year as the X-coordinate and two different values for expenditures as the Y-coordinates for the labels. We only used the last observation in the data set (where the year was 2000) to provide the X-Y information. We also provided text to be placed at each point. The ANNOTATE option is used on the PLOT statement in PROC GPLOT to add the labels to the lines.

At this point, you may be saying that this seems like a lot of effort to add some labels to one chart. It is time for another small sermon. You can add the labels to one, two, three, etc. charts with the SAS/GRAPH graphics editor, or you can learn how to export the chart to another program that allows you to edit graphics, e.g. PowerPoint or Adobe Illustrator. However, if you have a lot of charts to produce with extra information to be added to each chart, you can edit all those charts. Better yet, you can write some ANNOTATE code and let SAS take care of the customization for you.

REDIRECTING GRAPHICS OUTPUT TO FILES

All the graphics output thus far has been directed to the graphics output window, using SAS in display manager mode on a PC. Graphics output can also be directed to a number of different types of output devices (printers, plotters) or to files for later inclusion in a document or for posting on a web site. Two common file types for graphics output are GIF and PDF.

#17: PRODUCE A GIF FILE

```
options validvarname=upcase;
goptions device=gif gsfname=gout xpixels=1024 ypixels=768
         ftext='Arial' htext=2 gunit=pct ctext=green csymbol=blue;

symbol1 f=marker v='C' i=join h=1.25;
symbol2 f=marker v='U' i=join h=1.25;

axis1 label=(angle=90 rotate=0 "AMOUNT (IN BILLIONS)") minor=(n=3);
axis2 order=(1970 to 2000 by 5) minor=(n=4) offset=(2,2);

legend1 label=none value=(j=left "TOTAL" j=left "FEDERAL GOVT")
        mode=protect position=(top inside left)
        cborder=blue cshadow=blue
        across=1 shape=symbol(6,1.25);

title h=4 f='Arial/bo' "NATIONAL HEALTH CARE EXPENDITURES: 1970-2000";
footnote j=right "Source: Health-United States-2003";

filename gout 'z:\healthexp.gif';

proc gplot data=t112new;
   plot exp*year=expsrc/vaxis=axis1 haxis=axis2 caxis=blue legend=legend1;
   format exp comma.;
run;
```

Several new options are used in the GOPTIONS statement: the DEVICE option specifies that a GIF file will be created; GSFNAME specifies a FILEREF for a file that will contain the GIF output; XPIXELS and YPIXELS control the resolution of the GIF file. Just above PROC GPLOT, a FILENAME statement associates a file name (Z:\HEALTHEXP.GIF) with the FILEREF GOUT that was used in the GOPTIONS statement.

The output delivery system (ODS) is used to produce PDF output.

#18: PRODUCE A PDF FILE

```
options orientation=landscape validvarname=upcase;;
goptions ftext='Helvetica' htext=2 gunit=pct ctext=green csymbol=blue rotate=landscape;

symbol1 f=marker v='C' i=join h=1.25;
symbol2 f=marker v='U' i=join h=1.25;

axis1 label=(angle=90 rotate=0 "AMOUNT (IN BILLIONS)") minor=(n=3);
axis2 order=(1970 to 2000 by 5) minor=(n=4) offset=(2,2);

legend1 label=none value=(j=left "TOTAL" j=left "FEDERAL GOVT")
        mode=protect position=(top inside left)
        cborder=blue cshadow=blue
        across=1 shape=symbol(6,1.25);

title h=4 f='Helvetica/bo' "NATIONAL HEALTH CARE EXPENDITURES: 1970-2000";
footnote j=right "Source: Health-United States-2003";

ods listing close;
ods pdf file='z:\healthexp.pdf' notoc;
```

```
proc gplot data=t112new;
   plot exp*year=expsrc/vaxis=axis1 haxis=axis2 caxis=blue legend=legend1;
   format exp comma.;
run;

ods pdf close;
ods listing;
```

This time, no device is specified in the GOPTIONS statement. In anticipation of producing PDF output, the font is changed from a TrueType (Arial) to a Postscript (Helvetica) font. Just above PROC GPLOT, three ODS statements are used. The first two specify that we do not want any graphics results created in either the graphics output or results windows. The ODS PDF statement directs output to a file (Z:\HEALTHEXP.PDF). The NOTOC (no table of contents) option eliminates the default bookmark that is added to PDF output.

PRODUCING MANY PLOTS

One of the best features of SAS/GRAPH is its ability to produce many plots with a minimum of user intervention. You can use a BY statement within PROC GPLOT and create a plot for each level of the BY-VARIABLE. Or, since the plots are based on SAS code, you can use the power of a MACRO to automate the production of a series of plots. The macro method allows more control and flexibility than BY-VARIABLE processing and the macro method is used in the following examples.

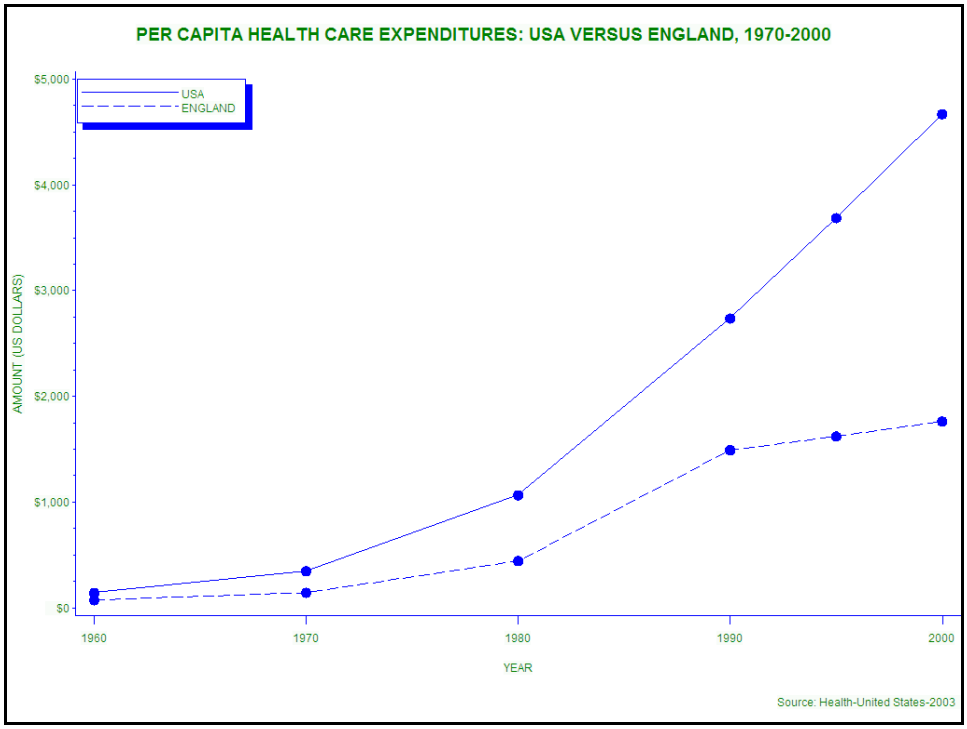
The Appendix contains another data set (#3) that we will use to produce a series of plots showing how per capita health expenditures in the United States have changed over time versus changes in several other countries. We will use many of the things that we learned in building a plot: SYMBOLS; AXIS; LEGEND. First, we write SAS code to produce only one plot, per capita expenditures in the United States versus England.

```
#19: ONE PLOT - DOES IT LOOK OK
options validvarname=upcase;
goptions device=gif gsfname=gout xpixels=1024 ypixels=768
ftext='Arial' htext=2 gunit=pct ctext=green csymbol=blue;

symbol1 v=dot i=join h=2.5 l=1;
symbol2 v=dot i=join h=2.5 l=3;

axis1 label=(angle=90 rotate=0 "AMOUNT (US DOLLARS)") minor=(n=3);
axis2 order=(1960 to 2000 by 10) minor=none offset=(2,2);

legend1 label=none value=(j=left "USA" j=left "ENGLAND")
        mode=protect position=(top inside left)
        cborder=blue cshadow=blue
        across=1 shape=line(10);
title1;
title2 h=3 f='Arial/bo' "PER CAPITA HEALTH CARE EXPENDITURES: USA VERSUS ENGLAND, 1970-2000";

footnote1 j=right "Source: Health-United States-2003   ";
footnote2 ' ';

filename gout 'z:\us_england.gif';

proc gplot data=t113;
   plot (usa eng)*year/overlay vaxis=axis1 haxis=axis2 caxis=blue legend=legend1 noframe;
   format usa dollar.;
run;
```

A GIF file is produced and it does look OK. One extra option was used on the PLOT statement in PROC GPLOT. NOFRAME specifies that we only want X-Y axes, not a complete box around the plot. Also, notice that a blank title (TITLE1) and a blank footnote (FOOTNOTE2) are used to move the text away from the upper and lower plot borders. Several blank spaces are added to FOOTNOTE1 to move the text away from the right border.

To produce a plot of the United States versus another country, all we need do is change the country name in the title and legend, specify a new file name, and change a variable name in the PLOT statement. The SAS code is rearranged, with all SAS code that does not change from plot-to-plot moved outside of the macro.

```
#20G: MANYPLOTS (GIF FILES)
options validvarname=upcase;
goptions device=gif gsfname=gout xpixels=1024 ypixels=768
         ftext='Arial' htext=2 gunit=pct ctext=green csymbol=blue;

symbol1 v=dot i=join h=2.5 l=1;
symbol2 v=dot i=join h=2.5 l=3;
```



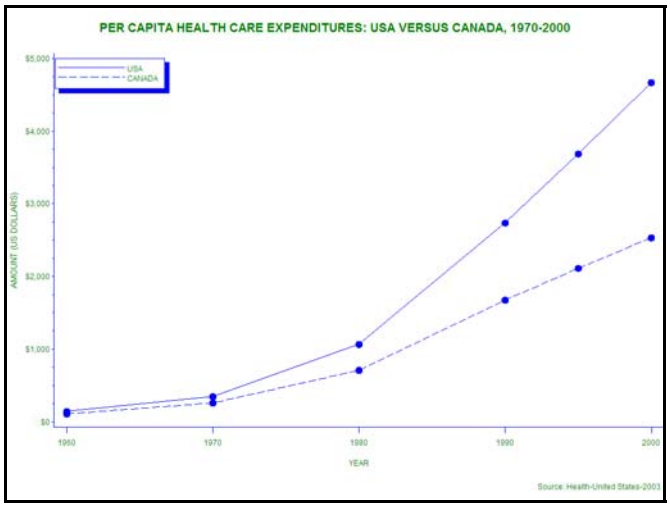
#19: ONE PLOT - DOES IT LOOK OK

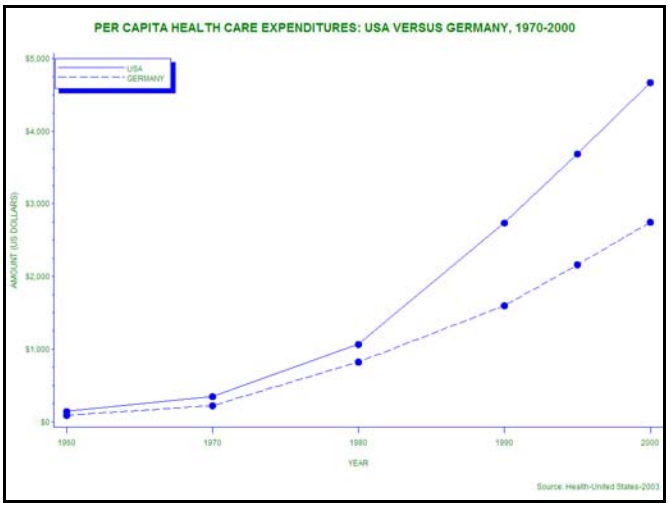
```
axis1 label=(angle=90 rotate=0 "AMOUNT (US DOLLARS)") minor=(n=3);
axis2 order=(1960 to 2000 by 10) minor=none offset=(2,2);
title1;
footnote1 j=right "Source: Health-United States-2003   ";
footnote2 ' ';

%macro makegif(var1,var2);
legend1 label=none value=(j=left "USA" j=left "&var1")
        mode=protect position=(top inside left)
        cborder=blue cshadow=blue
        across=1 shape=line(10);

title2 h=3 f='Arial/bo' "PER CAPITA HEALTH CARE EXPENDITURES: USA VERSUS &var1., 1970-2000";

filename gout "z:\US_&var1..gif";

proc gplot data=t113;
   plot (usa &var2)*year/overlay vaxis=axis1 haxis=axis2 caxis=blue legend=legend1 noframe;
   format usa dollar.;
run;
%mend;

%makegif(CANADA,can);
%makegif(GERMANY,ger);
%makegif(JAPAN,jap);
%makegif(NORWAY,nor);
%makegif(ENGLAND,eng);
```

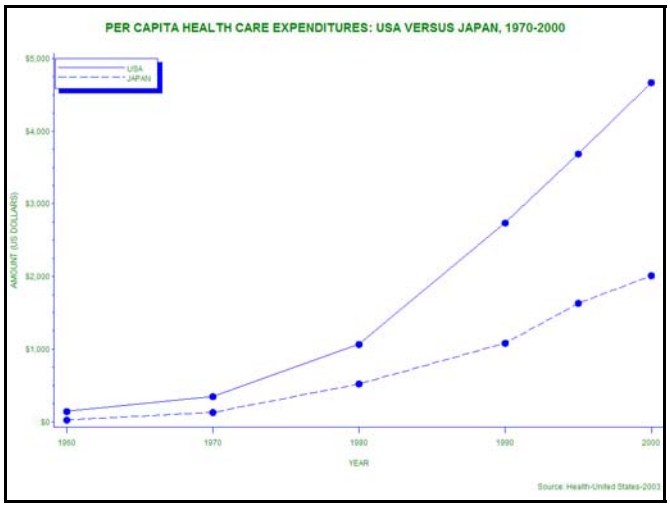
The macro is used to produce five plots. The four new plots (other than England, already shown above) are shown on the next page. Raster file output alternatives to GIF include PNG (DEVICE=PNG) and JPEG (DEVICE=JPEG). Rather than producing five GIF files, all the plots can be placed in one PDF file.

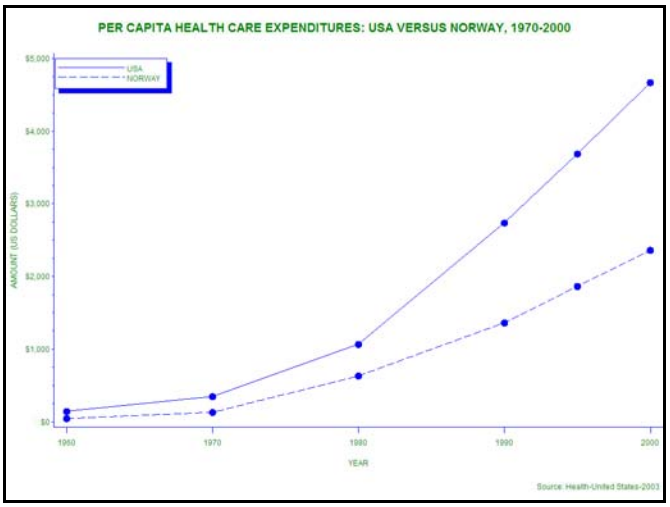
#20P: MANYPLOTS (ONE PDF FILE)

```
options orientation=landscape validvarname=upcase;
goptions ftext='Helvetica' htext=2 gunit=pct ctext=green csymbol=blue rotate=landscape;

symbol1 v=dot i=join h=2.5 l=1;
symbol2 v=dot i=join h=2.5 l=3;
axis1 label=(angle=90 rotate=0 "AMOUNT (US DOLLARS)") minor=(n=3);
```










```
axis2 order=(1960 to 2000 by 10) minor=none offset=(2,2);
footnote j=right "Source: Health-United States-2003";

%macro makeplot(var1,var2);
legend1 label=none value=(j=left "USA" j=left "&var1")
        mode=protect position=(top inside left)
        cborder=blue cshadow=blue across=1 shape=line(10);

title h=3 f='Helvetica/bo'
      "PER CAPITA HEALTH CARE EXPENDITURES: USA VERSUS &var1., 1970-2000";

proc gplot data=t113;
   plot (usa &var2)*year/overlay vaxis=axis1 haxis=axis2 caxis=blue legend=legend1;
   format usa dollar.;
run;
quit;
%mend;

ods listing close;
ods pdf file='z:\many_plots.pdf' notoc;

%makeplot(CANADA,can);
%makeplot(GERMANY,ger);
%makeplot(JAPAN,jap);
%makeplot(NORWAY,nor);
%makeplot(ENGLAND,eng);

ods pdf close;
ods listing;
```

REFERENCES (ALL HYPERLINKED)

There are a number of useful links on the SAS web site to help you with using SAS/GRAPH. The SAS/GRAPH manual is the starting point for basic information on SAS/GRAPH procedures and on the Annotate facility. The manual is available on-line.... ***Documentation for SAS Products and Solutions*** (http://support.sas.com/documentation/onlinedoc/index.html)

Other excellent tutorial-style papers are...

Cartier, Jeff. 2003. "It's All in the Presentation." *Proceedings of the Twenty-Eight Annual SAS Users Group International Conference*. Seattle, WA.

Cartier, Jeff. 2002. "The Basics of Creating Graphs with SAS/GRAPH® Software." *Proceedings of the Twenty-Seventh Annual SAS Users Group International Conference*. Orlando, FL.

There are numerous SAS/GRAPH examples, including the SAS code used to produce them on-line at...
SAS/GRAPH Sample Programs (http://support.sas.com/techsup/sample/sample_graph.html)

and more SAS/GRAPH examples at... ***SAS Graphing Components*** (http://support.sas.com/rnd/datavisualization/)

Another source of SAS/GRAPH examples and the SAS code used to produce them is...

SAS Institute Inc. (1991), *SAS/GRAPH Software Usage, Version 6, First Edition*, Cary, NC: SAS Institute Inc.

If you would like to learn more about using SAS/GRAPH output in documents or on the web, the following are very good...

Exporting SAS/GRAPH Output to PDF Files from Release 8.2 and higher
(http://support.sas.com/techsup/technote/ts659/ts659.pdf)

An Introduction to Exporting SAS/Graph Output to Microsoft Office SAS Release 8.2 and higher
(http://support.sas.com/techsup/technote/ts674/ts674.html)

Schrader, Dawn. 2003. "Exporting SAS/GRAPH® Output: Concepts and Ideas." *Proceedings of the Pharmaceutical Industry SAS Users Group*. Miami, FL.

If you want to search through all the technical support sites and papers for graphics-related information...

SAS Technical Support Documents (http://support.sas.com/techsup/tnote/tnote_index.html)

A paper that shows the graphical capabilities of PROC UNIVARIATE and PROC BOXPLOT is...

Are Histograms Giving You Fits? New SAS Software for Analyzing Distributions
(http://support.sas.com/rnd/app/papers/distributionanalysis.pdf)

while...

Rodriguez, Robert. 2004. "An Introduction to ODS for Statistical Graphics in SAS 9.1." *Proceedings of the Twenty-Ninth Annual SAS Users Group International Conference*, Montreal, QC.

shows how to produce graphics with PROC CORR, SAS/STAT, and SAS/ETS procedures.

If you are interested in statistical graphics, Michael Friendly's web site and book are good sources of ideas and SAS code...

Michael Friendly's Home Page (http://www.math.yorku.ca/SCS/friendly.html)

Friendly, Michael. 1991. *SAS System for Statistical Graphics, First Edition*. Cary, NC: SAS Institute Inc.

If you want to start slowly and learn more about SAS/GRAPH, the following books are well-written and are part of the SAS books-by-users program...

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

Carpenter, Art. 1999. *Annotate: Simply the Basics*. Cary, NC: SAS Institute Inc.

Finally, if you would like to look at some 'exotic' uses of SAS/GRAPH, there are web sites...

SAS Code Samples (http://www.devenezia.com/downloads/sas/samples/index.html)

SAS/GRAPH Software Samples (http://support.sas.com/rnd/samples/graph/index.html)

and a paper...

Zdeb, Mike, and Allison, Robert. 2005. "Stretching the Bounds of SAS/GRAPH® Software." *Proceedings of the Thirtieth Annual SAS Users Group International Conference*. Philadelphia, PA.

TRADEMARK CITATION

CONTACT INFORMATION (BOTH HYPERLINKED)

Either of the authors can be contacted using e-mail...

Mike Zdeb msz03@albany.edu
Robert Allison Robert.Allison@sas.com

APPENDIX

The data used in this paper are produced by the following three data steps.

```
*DATASET #1 - HEALTH CARE EXPENDITURES IN THE UNITED STATES (IN BILLIONS) NH_TOT->NATIONAL
              TOTAL/FG_TOT->FEDERAL GOVERNMENT;
data t112;
input year nh_tot fg_tot @@;
cards;
1970   74.4  17.7 1971  82.3  20.4
1972   92.3  22.9 1973 102.5  25.2
1974  116.1  30.5 1975 132.9  36.4
1976  152.2  42.9 1977 172.0  47.6
1978  193.7  54.3 1979 217.2  61.4
1980  250.1  72.0 1981 290.2  84.0
1982  326.1  93.3 1983 358.6 103.2
1984  389.6 112.6 1985 422.6 123.6
1986  454.8 133.1 1987 494.1 144.0
1988  546.0 156.7 1989 602.8 175.0
1990  666.2 195.4 1995 990.2 322.4
2000 1309.4 416.0
;
run;

*DATASET #2 - REARRANGE THE DATA EXP->EXPENDITURES/EXPSRC->EXPENDITURE SOURCE;
data t112new (keep=year exp expsrc);
set t112;
exp=nh_tot; expsrc=1; output;
exp=fg_tot; expsrc=2; output;
run;

*DATASET #3 - PER CAPITA HEALTH CARE EXPENDITURES
              CAN->CANADA/GE211R->GERMANY/JAP->JAPAN/NOR->NORWAY/
              ENG->ENGLAND/USA->UNITED STATES;
data t113;
input year can ger jap nor eng usa;
cards;
1960 109 90 26 46 74 143
1970 260 223 130 131 144 348
1980 710 824 522 632 444 1067
1990 1676 1600 1083 1363 1492 2738
1995 2114 2164 1631 1865 1622 3688
2000 2535 2748 2012 2362 1763 4672
;
run;
```