An Introduction to SAS/GRAPH® Software via the GPLOT Procedure

Michael S. Zdeb
New York State Department of Health

INTRODUCTION

SAS/GRAPH shares a trait with other parts of the SAS system, i.e., 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 at the bottom - where SAS makes the decisions as to the look of your graphics - and work their way gradually toward the top 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 your experimenting with other procedures. (Some debt is due the version 6 *SAS/GRAPH Usage* manual, a wonderful addition to the SAS documentation series, where the same approach is taken, albeit in a more comprehensive fashion than is possible in a single paper.)

BUILDING A CHART

Since one topic that seems to be on many people's minds this year is the amount of money spent on health care, the datasets to be used in this presentation all deal with health care expenditures both in the United States and in several other countries. The data are shown in the appendix. We will start with dataset #1 and produce a plot that shows the total national health expenditure in the United States from 1970 through 1990. Plot #1 is 'minimalist' SAS/GRAPH, or 'here are my data, show me a picture.'

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

The results this job (and each subsequent job) are shown at the end of the paper. SAS/GRAPH has made a few decisions - among them are: the scales on both the X and Y axes; the font to use 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'll 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're ready to start taking control. The first change to be made is to the SYMBOL that's 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'll 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;
```

It's now time to add a title (to describe the plot) and a footnote (to tell people where the data came from).

```
* #3: ADD A TITLE AND FOOTNOTE;
title1
"NATIONAL HEALTH CARE EXPENDITURES: 1970-1990";
footnote1
"Source:  Health-United States-1990";
```

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

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

What's this extra option that's specified - GUNIT? An explanation requires a slight digression from our path of chart construction. The height of text (HTEXT) is specified as 2. The QUESTION is '2 WHAT' - inches, centimeters, etc.? The ANSWER is, if no UNITS are specified, '...they are character cells...' - which should lead to the next question, 'What's a character cell?' SAS/GRAPH divides each output device it can use (printers, plotters, terminals, etc.) into a grid - a grid composed of character cells. The number of these cells varies from device to device. Before this 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. - i.e., 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's more than enough for now about units - back to chart construction.

The GOPTIONS has changed all of the text height - 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 let's leave the title centered).

```
* #5:  CHANGE THE HEIGHT OF THE TITLE - RIGHT
JUSTIFY THE FOOTNOTE;
title1 h=4
"NATIONAL HEALTH CARE EXPENDITURES:  1970-1990";
footnote1 justify=right
"Source:  Health-United States-1990";
```

Now that the line, title, and footnote look OK, it's 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'll do is change the labels. The X-axis label of 'YEAR' is fine, but the Y-axis might look better if it were labeled 'AMOUNT (IN BILLIONS)' - actually we'd all be better off it said '(IN THOUSANDS),' but

that's a whole other problem that's not solved by any SAS/GRAPH option or procedure. We'll have to settle for the changes we can make with an AXIS statement.

```
* #6:  CHANGE THE Y-AXIS LABEL - BOTH THE
CONTENT AND ANGLE;
axis1 label=(angle=90 rotate=0
             "AMOUNT (IN BILLIONS)");
proc gplot data=t112;
    plot nh_tot*year/vaxis=axis1;
run;
```

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've changed the label text and its angle. The ANGLE option determines the angle of the baseline on which the text rests, while the ROTATE option determines the angle of the text relative to the baseline. We will now change another attribute of the axes, i.e., the number of tick marks and number of labels printed for axis values.

```
* #7:  ADDITIONAL CHANGES OF BOTH THE X AND Y
AXES;
axis1 label=(angle=90 rotate=0
             "AMOUNT (IN BILLIONS)")
             minor=(n=3);

axis2 order=(1970 to 1990 by 5)
      minor=(n=4);
proc gplot data=t112;
    plot nh_tot*year/vaxis=axis1
                     haxis=axis2;
run;
```

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 chart - every fifth year is labeled - the intervening years are noted by minor tick marks. The chart has come a long way, now we'll add some color. Color (you'll have to use a little bit of imagination in looking at the attached output) is another attribute that can be controlled either within the GOPTIONS statement or within each statement (titles, footnotes, axes, etc.).

```
* #8:  ADD SOME COLOR VIA GOPTIONS AND AXES
STATEMENTS;
goptions ftext=swiss htext=2 gunit=pct
         ctext=green csymbol=yellow;
axis1 label=(angle=90 rotate=0
             "AMOUNT (IN BILLIONS)")
      minor=(n=3)
      color=yellow;
axis2 order=(1970 to 1990 by 5)
      minor=(n=4)
      color=yellow;
```

We have used the 'CTEXT=' and 'CSYMBOL=' options with the GOPTIONS statement to set the text and symbol colors to be used in the entire chart. You can see (REMEMBER, use your imagination) that the symbol color affects both the dots and the line connecting the dots since both are chosen in the SYMBOL statement. What happened to the color of the axis labels and labels for axis values? They are now yellow - the 'CAXIS=' option in an AXIS statement affects both the axis and any text associated with the axis. Let's make the axes and tick marks yellow, but leave all text associated with the axes green.

```
* #9:  CHANGE THE COLOR OF THE AXES, LEAVE ALL
AXES TEXT GREEN, MAKE THE SYMBOL LARGER WITH THE
'H=' OPTION;
axis1 label=(angle=90 rotate=0
                "AMOUNT (IN BILLIONS)")
       minor=(n=3) ;
axis2 order=(1970 to 1990 by 5)
      minor=(n=4);
symbol1 value=dot h=2.5 interpol=join;
proc gplot data=t112;
    plot nh_tot*year/vaxis=axis1
                     haxis=axis2
                     caxis=yellow;
run;
```

Another way to change the color of both axes is to use the 'CAXIS=' option within the PROC. 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). The 'H=' SYMBOL option uses the same units as text (remember 'GUNIT=PCT'). We'll declare the single-line chart completed now and move on - to placing two lines on the chart.

In addition to total national health care expenditures, the dataset also contains information as to how much money the federal government has spent on health care from 1970 through 1990. 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 chart ('...the plot thickens...').

```
* #10:  LEAVE ALL OPTIONS SET AS BEFORE - ADD
ANOTHER LINE TO THE CHART;
proc gplot data=t112;
    plot nh_tot*year
         fg_tot*year/vaxis=axis1
                     haxis=axis2
                     caxis=yellow
                     overlay;
run;
```

We can add another line merely by adding another set of 'Y*X' variables to the plot statement, and by adding the 'OVERLAY' option. Just as when we started building the chart, SAS/GRAPH has chosen its own symbol for the second line, and has chosen not to join the points. We need another SYMBOL statement to control the appearance of the second line.

```
* #11:  CHANGE THE SYMBOL OPTIONS FOR BOTH LINES
ON THE CHART;
symbol1 value=dot h=2.5 interpol=join line=1;
symbol2 value=dot h=2.5 interpol=join line=3;
```

There are a number of different ways to differentiate the lines on a plot. One way is to change the symbol VALUE - there are dots, triangles, squares, etc. Another way is to change the color (not an option is your printer only allows black-and-white). A third way is to change the type of line used to connect the points by using the 'LINE=' option. SAS/GRAPH provides forty-six different line types (though a chart where you would need all these probably is doomed from the start). We have chosen lines 1 and 3 (solid and dashed). We should probably change the title, but even if we change the title, something else is missing. It would be nice to have a LEGEND to tell those looking at the chart what each line represents.

Prior to version 6.07 of SAS/GRAPH, the production of a legend in PROC GPLOT required you to use a PLOT statement of the form 'Y*X=Z' where Z is a third variable that differentiates among various groups in the data. In our case, the X-variable would remain as the year and the Y-variable would be expenditures (either national total or federal government). The Z variable will be set to a value of 1 if the expenditure is national, and 2 if the expenditure is federal government (see the appendix for SAS code to restructure the data and for a listing of the restructured dataset, #2).

As of version 6.07, a legend can be added to an overlay style plot by adding a legend option to the plot statement.

```
* #12:  A NEW TWO-LINE PLOT WITH A NEW TITLE AND
A LEGEND;
goptions reset=symbol;
symbol1 value=dot    h=2.5 interpol=join;
symbol2 value=square h=2.5 interpol=join;
title1 h=4
"HEALTH CARE EXPENDITURES IN THE UNITED STATES:
1970-1990";
proc gplot data=t112;
    plot nh_tot*year
         fg_tot*year/vaxis=axis1
                     haxis=axis2
                     caxis=yellow
                     overlay
                                  legend;
run;
```

The label on the legend - 'PLOT' - is not very descriptive. Neither are the labels - 'NH_TOT' and 'FG_TOT' - on the two lines. Also notice that we have changed the SYMBOLS. We are no longer using line type to differentiate the data, but the symbol itself (a dot versus a square). You probably also noticed that there is a 'GOPTIONS RESET=SYMBOL' statement in the code. If this was not included, SAS/GRAPH would have remembered that the line type for the second symbol was dashed. SYMBOL statements are GLOBAL, i.e., they retain their values for the duration of a SAS session until you change them with another SYMBOL statement or reset them within a GOPTIONS statement. Since we had set the line type and do not have a 'LINE=' option in the new SYMBOL statement, the line would have remained dashed for SYMBOL2. SYMBOLS are not the only GLOBAL statements - TITLES, FOOTNOTES, LEGENDS, AXES are all GLOBAL. Enough about global issues, back to SAS/GRAPH

.

We can improve the appearance of the legend and make it informative - we'll change the LEGEND labeling, relocate it, and 'dress it up' a bit.

```
* #13:  IMPROVE THE LEGEND - MAKE IT USEFUL;
legend1 label=none
        value=(j=left "TOTAL" j=left
               "FEDERAL GOVT")
        mode=protect
        position=(top inside left)
        cborder=yellow
        cshadow=yellow;
proc gplot data=t112;
    plot nh_tot*year
         fg_tot*year/vaxis=axis1
                     haxis=axis2
                     caxis=yellow
                     overlay
                     legend;
run;
```

We don't really need a label on our LEGEND since the title and Y axis label do a good job of informing anyone looking at our chart as to what we are trying to portray. The LEGEND now has real labels that identify the two lines. We've relocated it on the chart using the POSITION option. What does MODE=PROTECT do? Just in case the lines in the chart try to occupy the same area as the legend, setting mode to protect assures that the LEGEND will be in the foreground, in front of the lines. Specifying 'CBORDER=' tells SAS/GRAPH that we want a FRAME around the LEGEND (we could have specified FRAME, but FRAME alone would default to another color). The 'CSHADOW=' option gives us the drop-shadow behind the legend. We replaced line labels in the LEGEND by using a VALUE option, with 'J=LEFT' left-justifying the words. One last item to notice about the chart is that moving the legend away from the bottom of the plot gives us more room for the plot - the Y axis is a little longer. It's time to declare this chart complete. The complete code used to produce both the final one-line and two-line charts is in the appendix.

PRODUCING MANY PLOTS

One of the best features of SAS/GRAPH is its ability to produce many plots with a minimum of user intervention. 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 appendix contains another dataset (#3) that we will use to show how 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, AXES, and a LEGEND. Let's first construct our SAS code to produce only one plot - expenditures in the United States versus England.

```
* #1:  PRODUCE ONE PLOT - SEE IF IT LOOKS
CORRECT;
goptions ftext=swiss htext=2 gunit=pct
         csymbol=yellow;

symbol1 value=dot    interpol=join;
symbol2 value=square interpol=join;

axis1 label=(angle=90 rotate=0
             "AMOUNT")
      minor=(n=3) ;
axis2 order=(1960 to 1990 by 5)
      minor=(n=4);

footnote1 justify=right
"Source:  Health-United States-1990";

legend1 label=none
        value=(j=l "USA" j=l "ENGLAND")
        mode=protect
        position=(top inside left)
        cborder=yellow
        cshadow=yellow;

title1 h=3
"PER CAPITA HEALTH CARE EXPENDITURES-USA vs
ENGLAND:  1960-1990";
proc gplot data=t113;
    plot usa*year
         eng*year/vaxis=axis1
                  haxis=axis2
                  caxis=yellow
                  overlay
                  legend=legend1;
run;
```

What would we have to change to produce another plot of the United States data versus some other country? Within the data step, we would have to specify another country (e.g., 'GAR' instead of 'ENG') . Then we would also have to change the title and the legend (e.g., from 'ENGLAND' to 'GERMANY'). All we do is turn these changeable items into macro variables. The SYMBOLS, AXES, and FOOTNOTE never change, so they can stay out of the macro.

```
* #2:  TAKE THE REPETITIVE STEPS AND PUT THEM IN
A MACRO;
%macro plotit(cou1,cou2);
legend1 label=none
        value=(j=l "USA" j=l "&cou2")
        mode=protect
        position=(top inside left)
        cborder=yellow
        cshadow=yellow;
title1 h=3
"PER CAPITA HEALTH CARE EXPENDITURES-USA vs
&cou2:  1960-1990";

proc gplot data=t113;
    plot usa*year
         &cou*year/vaxis=axis1
                   haxis=axis2
                   caxis=yellow
                   overlay
                   legend=legend1;
run;
%mend;

%plotit(gar,GERMANY);
%plotit(can,CANADA);
%plotit(jap,JAPAN);
%plotit(swe,SWEDEN);
```

Now it's easy to produce many plots with a minimum of SAS code. This is one of the great advantages of SAS/GRAPH over some of the other plotting packages that you may use - many packages force you to produce one plot at a time. Your labor in producing one plot is lost for production of the second, third etc. This may not be too much of a hardship with a few plots, but it becomes a real detriment to productivity when the number of plots becomes large - end of sermon. Two plots are shown at the end of the paper.

ANNOTATION

Your introduction to SAS/GRAPH would not be complete without at least a mention of the ANNOTATE function. 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 overlayed on a map of Europe?' Well, even if you haven't seen or wondered about that, you should at least know what ANNOTATE is. The ANNOTATE functions within SAS/GRAPH allow you to take a chart produced by one of the standard PROCs and add anything you want (text, lines, etc.) anywhere on the chart. You can also construct charts entirely with ANNOTATE functions if you want to display data in some way that just isn't possible with one of the SAS/GRAPH procedures. We will stick to the first use and show how ANNOTATE can be used to add some text to one of our plots.

When we produced a plot with two lines, we saw that there were two ways to produce a legend that would tell anyone looking at our plot what data each line represented. There is a another to distinguish between the two lines, i.e. to label each line using ANNOTATE. If you are really new to SAS/GRAPH (or SAS in general), the following way seem a little obtuse - don't worry, it's not just you - ANNOTATE is obtuse. We will take dataset #3 and plot the per capita health expenditures in the United States and England. Instead of using a legend, we'll label each line with ANNOTATE.

```
* #1:  CREATE AN ANNOTATE DATA SET (CALL IT
'LABELS') FROM DATASET #3;
data labels;
retain xsys ysys '2' function 'label' when 'A'
position '1';
set t113 end=last;
if last then do;
   text='UNITED STATES';
   x=year; y=usa; output;
   text='ENGLAND';
   x=year; y=eng; output;
end;
run;

* #2:  SET UP GOPTIONS, SYMBOLS, AXES, TITLE,
AND FOOTNOTE;
goptions ftext=swiss ctext=green htext=2
         gunit=pct csymbol=yellow;
symbol1 value=dot    h=2.5 interpol=join;
symbol2 value=square h=2.5 interpol=join;
axis1 label=(angle=90 rotate=0
             "AMOUNT")
      minor=(n=3) ;
axis2 order=(1960 to 1990 by 5)
      minor=(n=4);
title1 h=4
"PER CAPITA HEALTH CARE EXPENDITURES-USA vs
ENGLAND:  1960-1990";
footnote1 justify=right
"Source:  Health-United States-1990";

* #3:  CREATE THE CHART - USE THE ANNOTATE
DATASET LABELS;
proc gplot data=t113;
   plot usa*year
        eng*year/vaxis=axis1
                 haxis=axis2
                 caxis=yellow
                 overlay
                 annotate=labels;
run;
```

The annotate data set 'LABELS' contains two observations. Each observation contains enough information to place a label on the chart. It would be 'overkill' at this point to go into a lot of detail as to the various items that were specified in order to get the labels in the correct spot - this is just meant to show you one application of ANNOTATE. You can probably ascertain from the data step SAS code that we used the year as the X-coordinate and the per capita expenditure as the Y-coordinate for the labels. We only used the last observation in the dataset (where the year was 1990) to provide the X-Y information. We also specified: a coordinate system (XSYS, YSYS); a function (LABEL); a time (WHEN, in this case 'A" means place the label on the chart AFTER the PROC is done with its work); a location around the X-Y point (POSITION, in this case '1' means to the upper-left of the point). Voila! The annotated chart is shown at the end of the paper.

You may be saying that this seems like a lot of effort to add some labels to one chart. It's time for another small sermon. You can add the labels to one, two, three, etc. charts with the SAS/GRAPH graphics editor. Or, if you become an advanced student, you can export the chart to another program that allows you to edit graphics (e.g., Freelance or Harvard Graphics). However, what if you have a lot of charts to produce with extra

information to be added to each chart. Sure, you can edit all those charts. Better yet, you can write some ANNOTATE code and let SAS take care of it for you.

PRODUCING MANY PIE-CHARTS

If you're bored with all those plots, you can produce another type of chart. Now that you understand how to use a macro to produce repetitive plots, you can try out your knowledge on pie charts. The dataset to be used (#4) is in the appendix. The data show how the source of funds for health expenditures in the United States has changed over the period 1950 through 1990. We want to produce a pie chart for each of the 10-year intervals (5 pie charts)..

We learned in our production of plots that it was sometimes necessary to rearrange our data - this is also true for this example. With plots, we started at the bottom. In this final example we are going to jump right in at the end so you can see a finished product.

```
* PRODUCE A GROUP OF PIE CHARTS USING A MACRO;
goptions ftext=swiss htext=2 ctext=green
         gunit=pct;

pattern1 value=psolid color=red;
pattern2 value=psolid color=green;
pattern3 value=psolid color=blue;
pattern4 value=psolid color=yellow;

title1 h=3
"PERCENT DISTRIBUTION OF HEALTH CARE
EXPENDITURES";
footnote1 justify=right
"Source:  Health-United State-1990";

%macro piechart(yr);
data t121new (keep=source pct);
   set t121;
   if year eq &yr;
   source='PRIVATE HLTH INS'; pct=phi; output;
   source='OUT-OF-POCKET';    pct=oop; output;
   source='OTHER PRIV FUNDS'; pct=opf; output;
   source='GOVERNMENT';       pct=gov; output;
run;
title2 h=3 "BY SOURCE OF FUNDS-UNITED STATES:
&yr";
proc gchart data=t121new;
   pie source/sumvar=pct
              noheading
              slice=arrow
              coutline=cyan;
run;
%mend;

%piechart(1950);
%piechart(1960);
%piechart(1970);
%piechart(1980);
%piechart(1990);
```

Our GOPTIONS statement looks the same as when we produced plots. Instead of SYMBOLS we now have PATTERNS to specify how chart slices are to be filled. A data step is used to both select a year of data and to rearrange the data for PROC GCHART. The only variable that changes from chart-to-chart is the year - our macro variable. Two pie charts are shown at the end of the paper (they really look 'round' if you don't print them the way I did - using PROC GREPLAY, but that's a whole other adventure).

FINAL WORDS

If you are a 'SAS/GRAPH neophyte,' you shouldn't be intimidated by documentation. The approach taken here in building a plot from the bottom up is the same approach taken in the new version 6 *SAS/GRAPH Software: Usage* manual for all SAS/GRAPH procedures. Now that you know how to build a plot, start with *Usage* and work your way through the other procedures - in a lot more detail. Once you've conquered *USAGE*, you'll be ready for the *REFERENCE* manuals!

APPENDIX

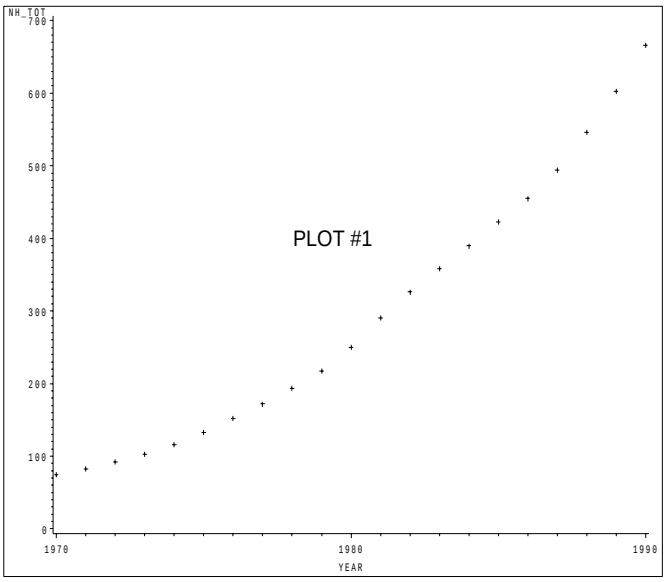
```
*DATASET #1 - HEALTH CARE EXPENDITURES IN THE
UNITED STATES (IN BILLIONS) NH_TOT->NATIONAL
TOTAL/FG_TOT->FEDERAL GOVERNMENT;
data t112;
   input year nh_tot fg_tot @@;
cards;
1970  74.4  17.7  1971  82.3  20.4
1972  92.3  22.9  1973 102.5  25.2
1974 116.1  30.5  1975 132.9  36.4
1976 152.2  42.9  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
;
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;

proc print data=t112new;
run;
```

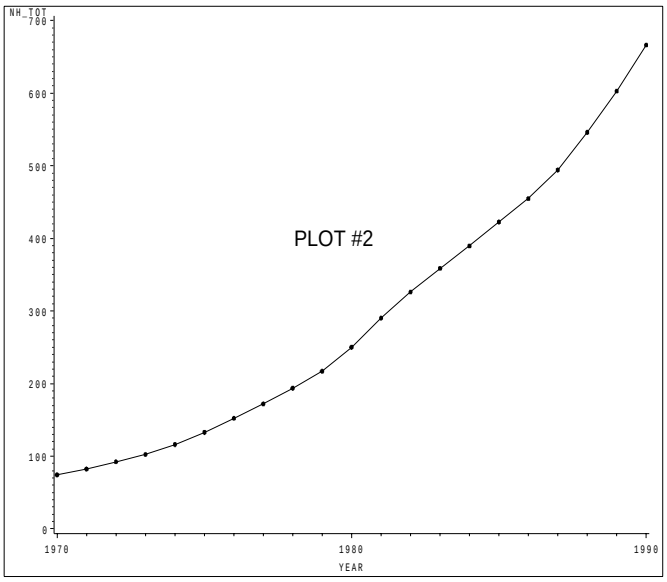
OBS	YEAR	EXP	EXPSRC
1	1970	74.4	1
2	1970	17.7	2
3	1971	82.3	1

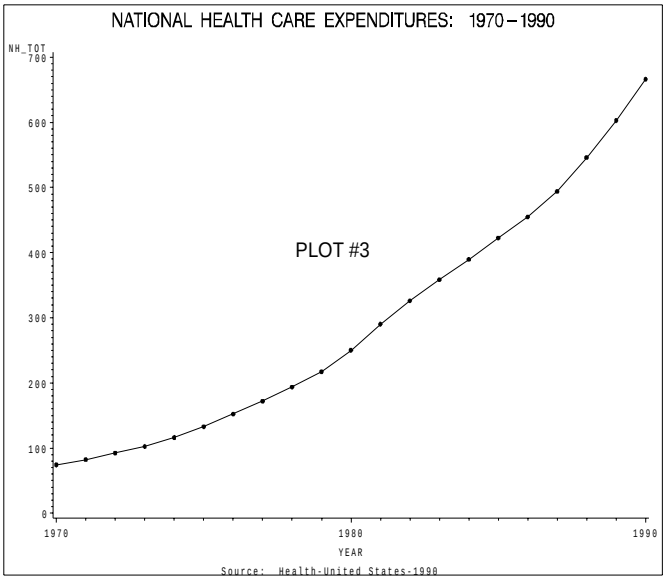
39	1989	602.8	1
40	1989	175.0	2
41	1990	666.2	1
42	1990	195.4	2

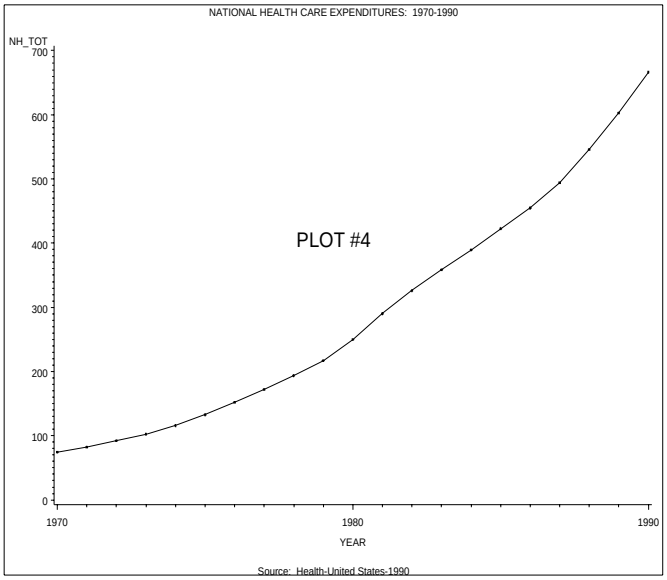
```
*DATASET #3 - PER CAPITA HEALTH CARE
EXPENDITURES CAN->CANADA/GAR->GERMANY/JAP->JAPAN
SWE->SWEDEN/ENG->ENGLAND/USA->UNITED STATES;
data t113;
   input year can gar jap swe eng usa;
cards;
1960   117    86    26    92    76   142
1965   165   119    63   145    98   206
1970   274   199   126   274   146   346
1975   478   420   252   475   272   592
1980   806   739   515   842   454  1063
1985  1315  1046   785  1125   662  1711
1988  1585  1243   978  1258   817  2145
1989  1683  1224  1060  1344   864  2346
1990  1795  1287  1145  1421   932  2566
;
run;

*DATASET #4 - PERCENT DISTRIBUTION OF HEALTH
CARE EXPENSES BY SOURCE OF FUNDS - UNITED STATES
OOP->OUT-OF-POCKET/PHI->PRIVATE HEALTH INS
OPF->OTHER PRIVATE FUNDS/GOV->GOVERNMENT;
```

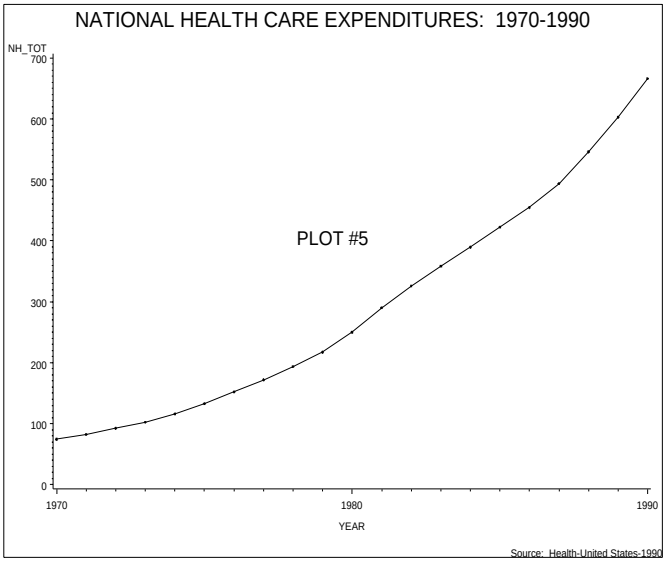
```
data t121;
   input year oop phi opf gov;
cards;
1950       65.5    9.1   2.9    22.4
1960       55.9   21.0   1.7    21.4
1970       39.5   23.4   2.6    34.6
1980       27.1   29.7   3.5    39.7
1990       23.3   31.8   3.6    41.3
;
run;

************************************
COMPLETE CODE FOR THE FINAL 1-LINE PLOT
************************************;
goptions ftext=swiss htext=2 gunit=pct
         ctext=green csymbol=yellow;

title1 h=4
"NATIONAL HEALTH CARE EXPENDITURES:  1970-1990";
footnote1 justify=right
"Source:  Health-United State-1990";

axis1 label=(angle=90 rotate=0
             "AMOUNT (IN BILLIONS)")
      minor=(n=3);
axis2 order=(1970 to 1990 by 5)
      minor=(n=4);

symbol1 value=dot h=2.5 interpol=join;

proc gplot data=t112;
  plot nh_tot*year/vaxis=axis1
                   haxis=axis2
                   caxis=yellow;
run;


************************************
COMPLETE CODE FOR THE FINAL 2-LINE PLOT
************************************;
goptions ftext=swiss htext=2 gunit=pct
         ctext=green csymbol=yellow;

title1 h=4
"HEALTH CARE EXPENDITURES-UNITED STATES:
1970-1990";
footnote1 justify=right
"Source:  Health-United State-1990";

axis1 label=(angle=90 rotate=0
             "AMOUNT (IN BILLIONS)")
      minor=(n=3) ;
axis2 order=(1970 to 1990 by 5)
      minor=(n=4) ;

symbol1 value=dot    h=2.5 interpol=join;
symbol2 value=square h=2.5 interpol=join;

legend1 label=none
        value=(j=l "TOTAL" j=l "FEDERAL GOVT")
        mode=protect
        position=(top inside left)
        cborder=yellow
        cshadow=yellow;

proc gplot data=t112;
   plot nh_tot*year
        fg_tot*year/vaxis=axis1
                    haxis=axis2
                    caxis=yellow
                    overlay
                    legend=legend1;
run;
```

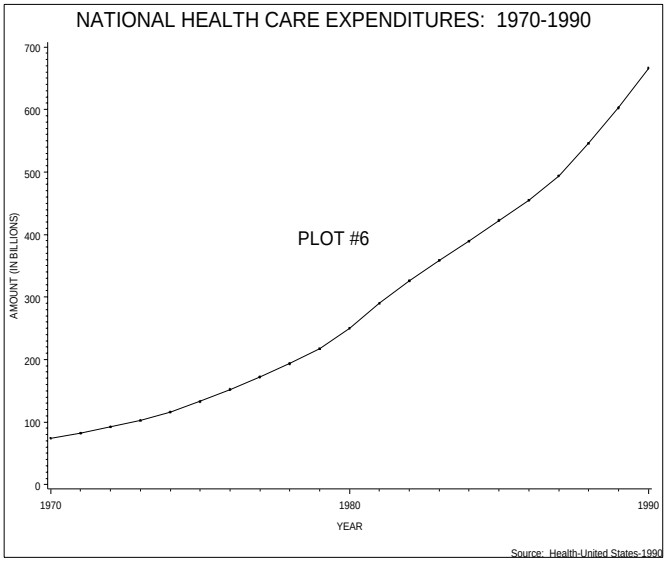

NH_TOT
700
600
500
400
300
200
100
0
PLOT #1
1970
1980
1990
YEAR

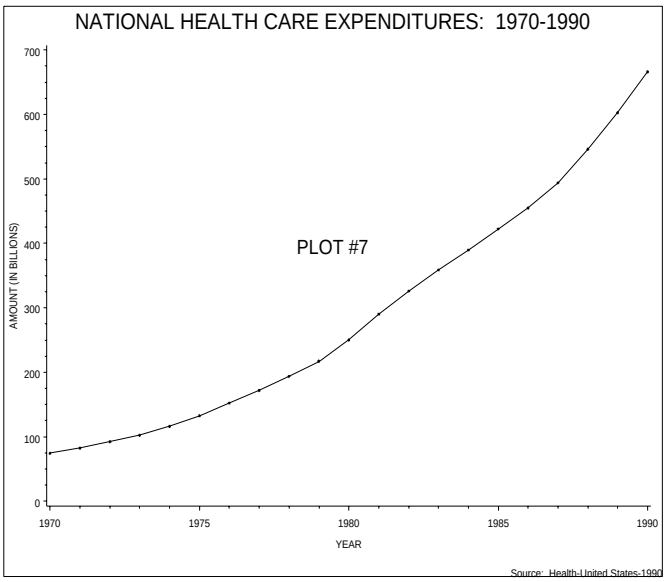

NH_TOT
700
600
500
400
300
200
100
0
PLOT #2
1970
1980
1990
YEAR

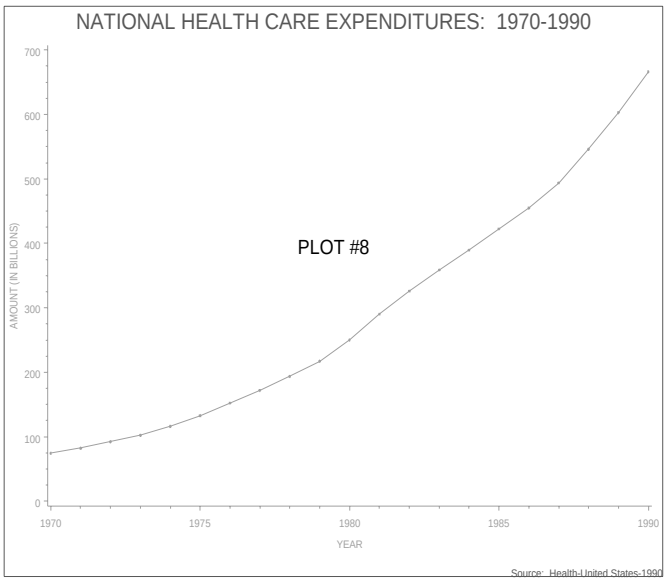

NATIONAL HEALTH CARE EXPENDITURES: 1970-1990
NH_TOT
700
600
500
400
300
200
100
0
PLOT #3
1970
1980
1990
YEAR
Source: Health-United States-1990

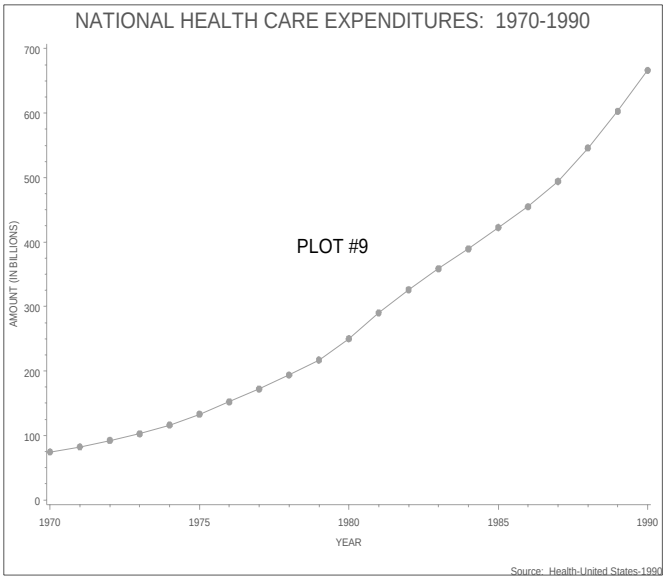

NATIONAL HEALTH CARE EXPENDITURES: 1970-1990
NH_TOT
700
600
500
400
300
200
100
0
PLOT #4
1970
1980
1990
YEAR
Source: Health-United States-1990

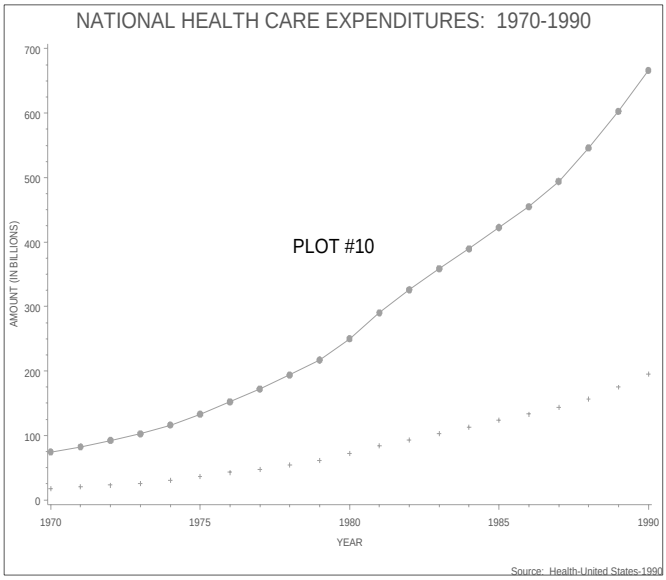

NATIONAL HEALTH CARE EXPENDITURES: 1970-1990
NH_TOT
700
600
500
400
300
200
100
0
PLOT #5
1970
1980
1990
YEAR
Source: Health-United States-1990

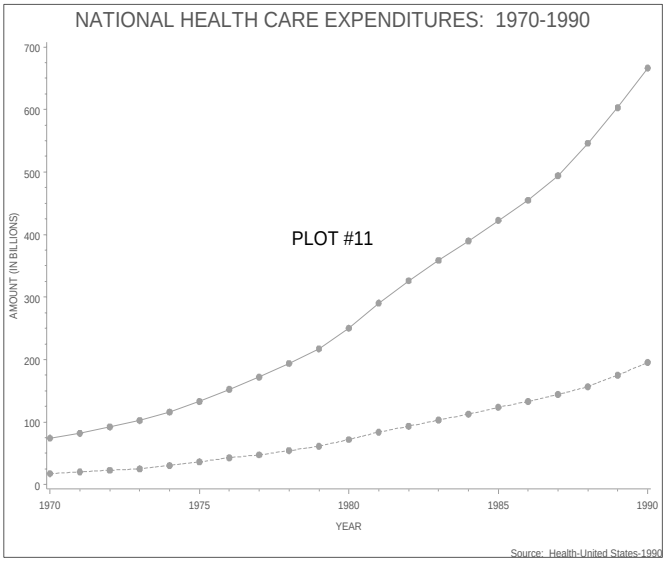

NATIONAL HEALTH CARE EXPENDITURES: 1970-1990
AMOUNT (IN BILLIONS)
700
600
500
400
300
200
100
0
PLOT #6
1970
1980
1990
YEAR
Source: Health-United States-1990

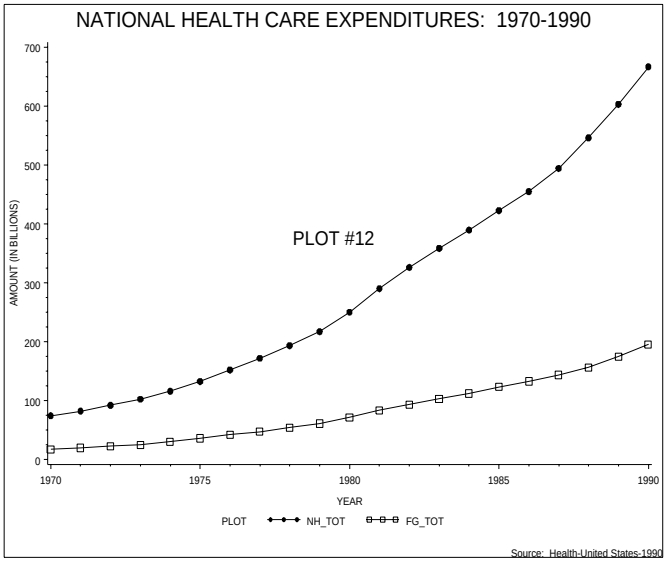

NATIONAL HEALTH CARE EXPENDITURES: 1970-1990
AMOUNT (IN BILLIONS)
YEAR
PLOT #7
Source: Health-United States-1990

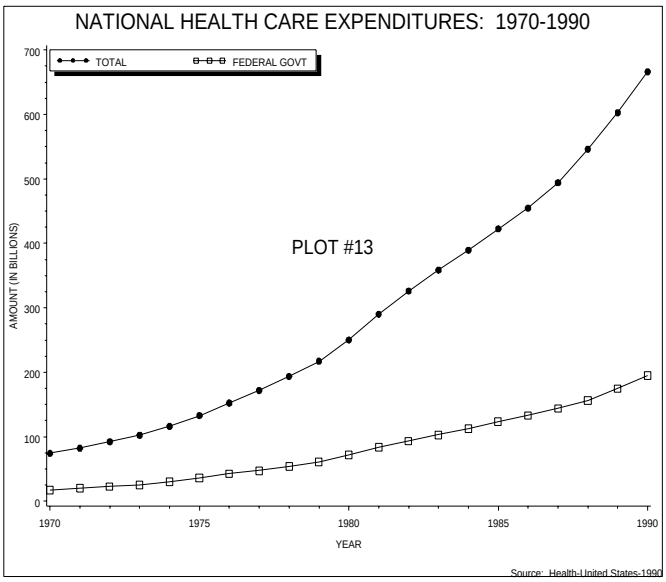

NATIONAL HEALTH CARE EXPENDITURES: 1970-1990
AMOUNT (IN BILLIONS)
YEAR
PLOT #8
Source: Health-United States-1990

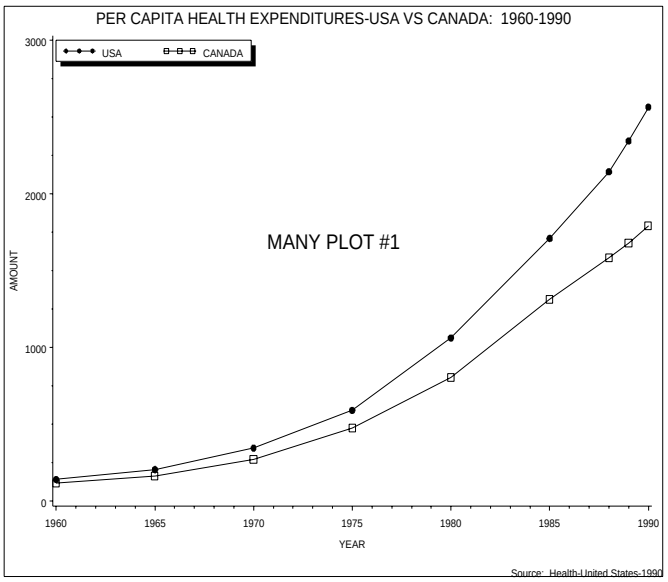

NATIONAL HEALTH CARE EXPENDITURES: 1970-1990
AMOUNT (IN BILLIONS)
YEAR
PLOT #9
Source: Health-United States-1990

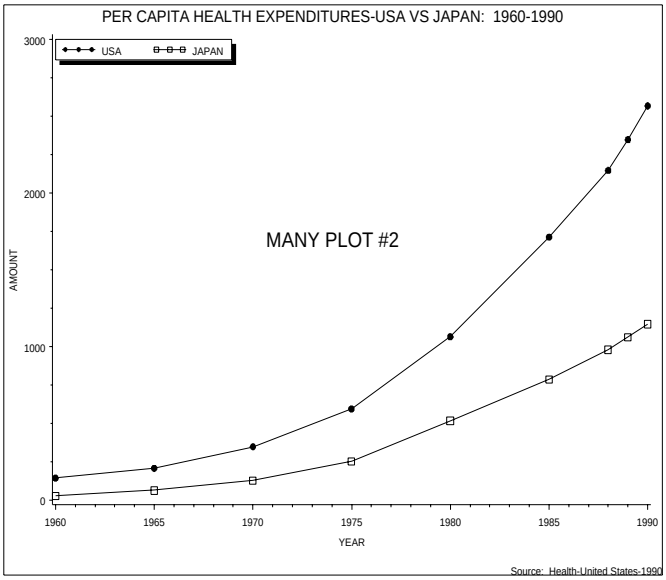

NATIONAL HEALTH CARE EXPENDITURES: 1970-1990
AMOUNT (IN BILLIONS)
YEAR
PLOT #10
Source: Health-United States-1990

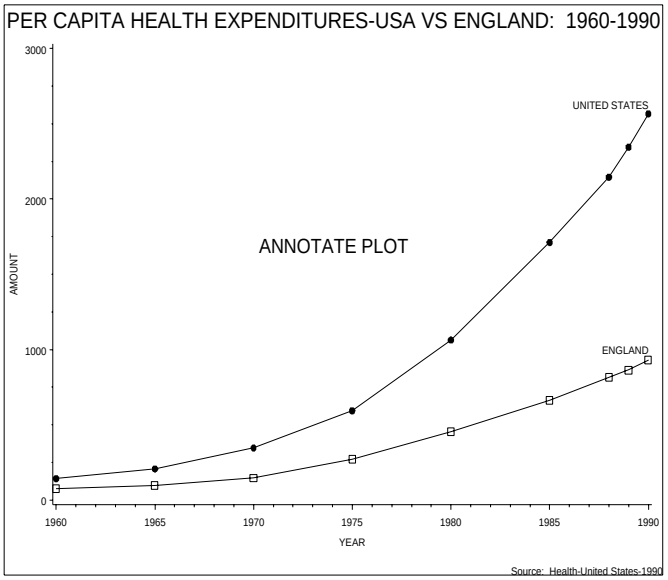

NATIONAL HEALTH CARE EXPENDITURES: 1970-1990
AMOUNT (IN BILLIONS)
YEAR
PLOT #11
Source: Health-United States-1990

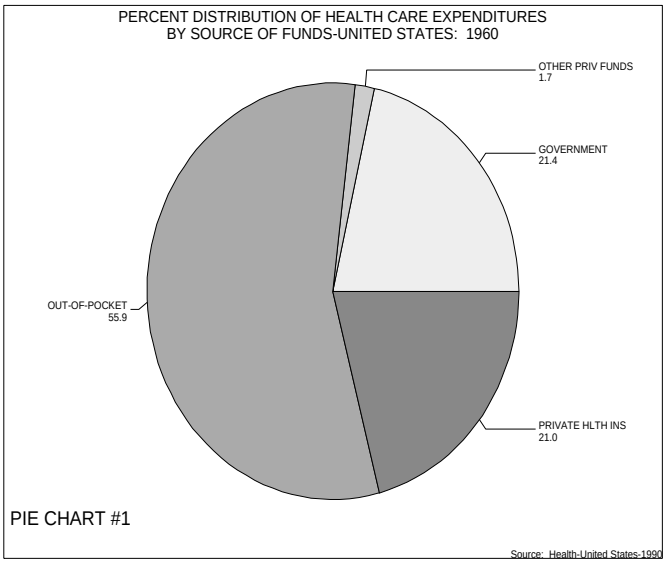

NATIONAL HEALTH CARE EXPENDITURES: 1970-1990
AMOUNT (IN BILLIONS)
YEAR
PLOT #12
PLOT NH_TOT FG_TOT
Source: Health-United States-1990

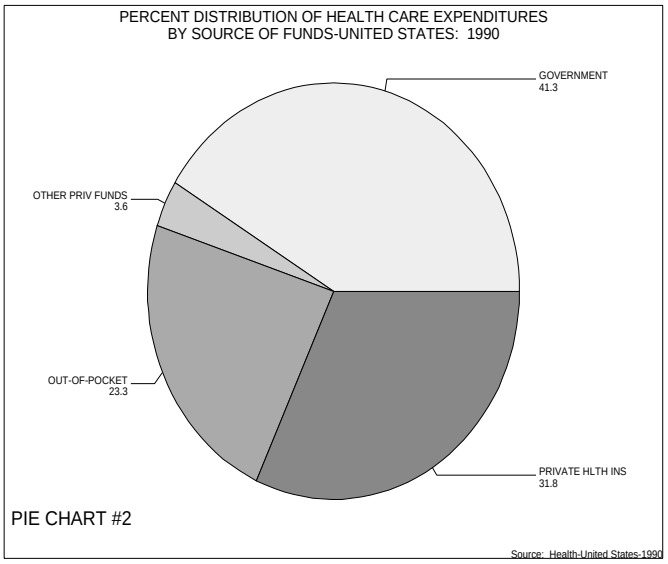

NATIONAL HEALTH CARE EXPENDITURES: 1970-1990
TOTAL
FEDERAL GOVT
AMOUNT (IN BILLIONS)
PLOT #13
YEAR
Source: Health-United States-1990


PER CAPITA HEALTH EXPENDITURES-USA VS CANADA: 1960-1990
USA
CANADA
AMOUNT
MANY PLOT #1
YEAR
Source: Health-United States-1990


PER CAPITA HEALTH EXPENDITURES-USA VS JAPAN: 1960-1990
USA
JAPAN
AMOUNT
MANY PLOT #2
YEAR
Source: Health-United States-1990


PER CAPITA HEALTH EXPENDITURES-USA VS ENGLAND: 1960-1990
UNITED STATES
ENGLAND
AMOUNT
ANNOTATE PLOT
YEAR
Source: Health-United States-1990


PERCENT DISTRIBUTION OF HEALTH CARE EXPENDITURES
BY SOURCE OF FUNDS-UNITED STATES: 1960
OTHER PRIV FUNDS
1.7
GOVERNMENT
21.4
OUT-OF-POCKET
55.9
PRIVATE HLTH INS
21.0
PIE CHART #1
Source: Health-United States-1990


PERCENT DISTRIBUTION OF HEALTH CARE EXPENDITURES
BY SOURCE OF FUNDS-UNITED STATES: 1990
GOVERNMENT
41.3
OTHER PRIV FUNDS
3.6
OUT-OF-POCKET
23.3
PRIVATE HLTH INS
31.8
PIE CHART #2
Source: Health-United States-1990