

The Basics of Creating Graphs with SAS/GRAPH® Software

Jeff Cartier, SAS Institute Inc., Cary, NC

ABSTRACT

SAS/GRAPH software is a very powerful tool for creating a wide range of business and scientific graphs. This presentation provides an overview the types of graphs that can be produced with SAS/GRAPH software and the basic procedure syntax for creating some of them, including bar charts and plots. You will understand how to identify the appropriate variables of your data for the graph you want and how to customize its appearance. You will also see how to create image files for your graphs as well as how to use the Output Delivery System to create PDF, RTF, or HTML files with graphical content. In conclusion, you will see an interactive application included with SAS/GRAPH software that can be used to create many types of graphs and graphical output formats, including the generation of SAS/GRAPH programs.

OVERVIEW

SAS/GRAPH software consists of a collection of procedures, SAS data sets containing map data, and other utility procedures and applications that help manage graphical data and output. There are five main procedures that produce specific types of graphs:

GCHART	bar, pie, block, donut, star charts
GPLOT	scatter, bubble, line, area, box, regression plots
G3D	3-D scatter and surface plots
GCONTOUR	contour plots
GMAP	maps with user-defined data

CREATING BAR CHARTS

Each procedure listed above has one or more modifying statements (called *action* statements) that narrow down the type of graph produced. For example,

```
PROC GCHART DATA=SAS-data-set;
  VBAR variable ;    vertical bar chart
  HBAR variable ;    horizontal bar chart
  PIE variable ;     pie bar chart
```

Let's look at an actual program that creates a vertical bar chart based the data set SASHELP.CLASS. Its first eight observations:

Name	Sex	Age	Height	Weight
Alfred	M	14	69	112.5
Alice	F	13	56.5	84
Barbara	F	13	65.3	98
Carol	F	14	62.8	102.5
Henry	M	14	63.5	102.5
James	M	12	57.3	83
Jane	F	12	59.8	84.5
Janet	F	15	62.5	112.5

Figure 1 Partial display of SASHELP.CLASS

Let's say we would like the bar chart to show the distribution of student ages as a percentage. Here's the complete program:

```
proc gchart data=sashelp.class;
  vbar age / type=percent;
run;
quit;
```

The VBAR statement declares the charting variable AGE. The slash indicates the beginning of optional syntax. The TYPE= option specifies the statistic to be computed. (The default statistic is FREQ, a frequency count of observations). Notice that the summarization of the data is done by this procedure -- we did not have to code separate steps to either sort the data or pre-compute the percentages.

Where does the output go? Unless you indicate otherwise, graphics procedures sent output to the WORK .GSEG catalog. You can view this output at anytime from the GRAPH window (View->Graph). Until you have all the desired features for your graph, it is a good idea to let your output go there. You can think of this a "preview" of your final output.

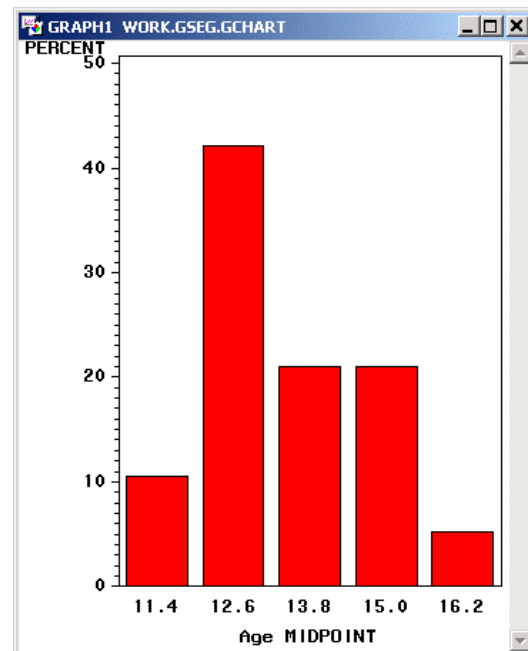


Figure 2 Output Displayed in Graph Window

The axis values for AGE may not be what you expected to see. This is because GHART treats a numeric chart variable as having continuous values by default. To determine how many bars to display, the procedure divides the range of values into some number of equal size intervals and then assigns each observation to one of the intervals. The values you see under each bar are the *midpoints* of the intervals. This representation is often called a *histogram*. The VBAR and HBAR statements have options that let you control how many intervals to use and whether to display interval range instead of interval midpoint values.

Creating a histogram would be more appropriate representation for variables like HEIGHT or WEIGHT.

We want to treat the AGE values as if they were *categories*. This requires adding the DISCRETE option to the VBAR statement:

```
vbar age / discrete type=percent;
```

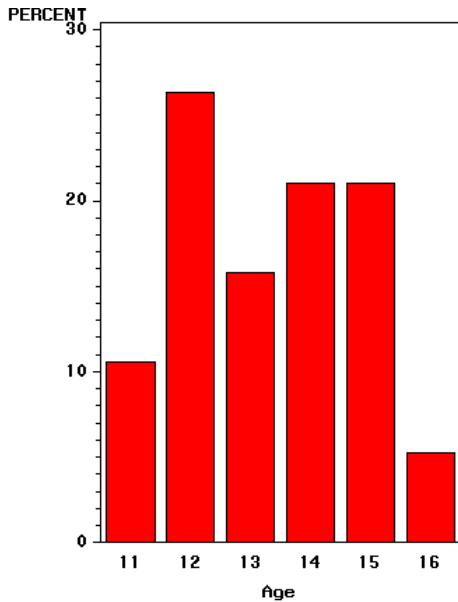


Figure 3 AGE Categories

RESPONSE VARIABLE

Sometimes you want to compute a summary statistic for a numeric variable within the categories or intervals. For example, you may be interested having the bars represent the average height within age values. In this scenario, HEIGHT is used as a *response variable*.

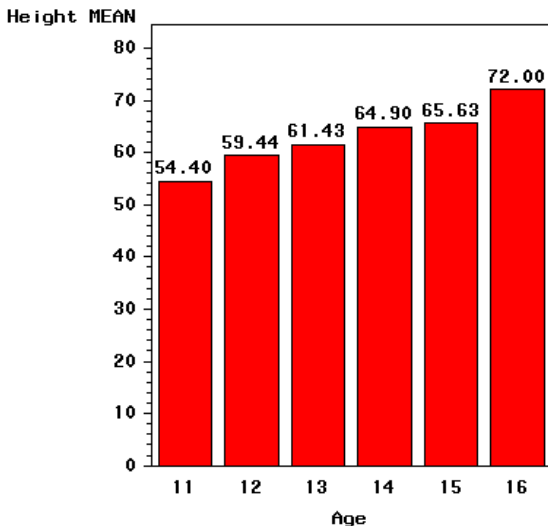


Figure 4 HEIGHT as Response Variable

To declare a response variable, use the SUMVAR= option. When using this option, you must also change the chart statistic to SUM or MEAN.

```
proc chart data=sashelp.class;
  vbar age / discrete type=mean
              sumvar=height mean;
run;
quit;
```

We have also added the MEAN option to force the display the value of the mean at the end of each bar.

GROUP AND SUBGROUP VARIABLES

The VBAR and HBAR statements can also produce a grouped (side-by-side) bar chart using the GROUP= option.

```
hbar age / discrete type=mean
           sumvar=height mean
           group=sex;
```

Here we have changed the chart orientation by using the HBAR statement to avoid crowding the bars too much and to make the bar values easier to read.

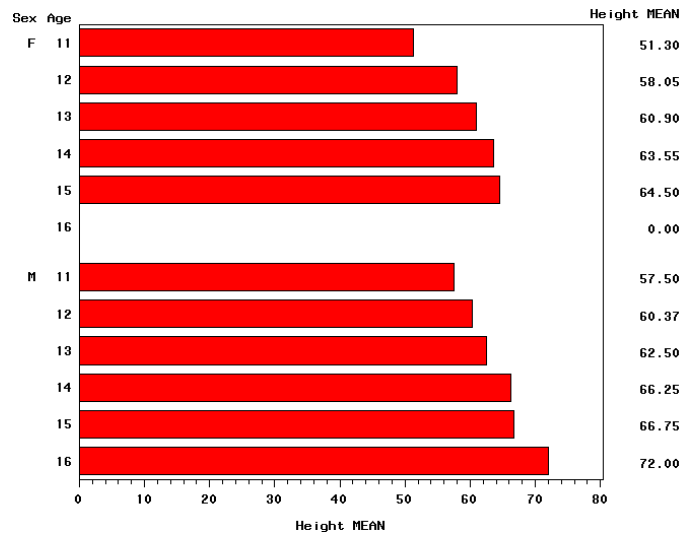


Figure 5 SEX as Group Variable

You can create a subgrouped (stacked) bar chart using the SUBGROUP= option.

```
vbar age / discrete type=percent
           subgroup=sex;
```

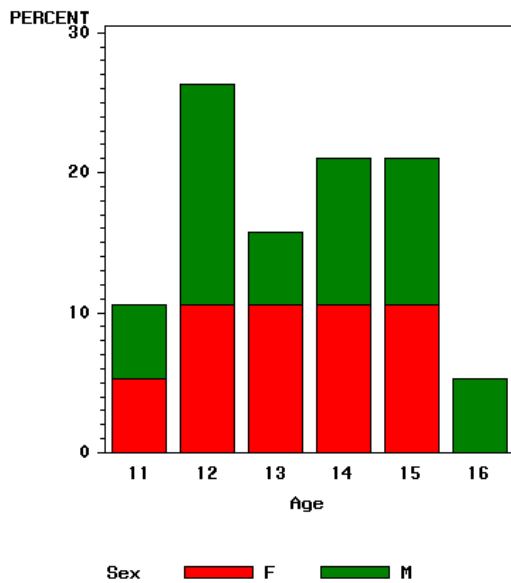


Figure 6 SEX as Subgroup Variable

A legend is automatically created.

To make your bar charts understandable, the variables you choose for the SUBGROUP= and GROUP= options, like the chart variable itself, should not have a large number of distinct values.

There are a large number of additional options that can be used with the action statements to control reference lines, appearance features such as bar width and bar spacing, and colors for all aspects of the chart.

TITLE and FOOTNOTE STATEMENTS

The five graphics procedures mentioned earlier support TITLE and FOOTNOTE statements. As you add titles and footnotes to the graphics area, the vertical size of the chart decreases to compensate for the additional text.

LABEL AND FORMAT STATEMENTS

By default, SAS/GRAPH procedures use existing variable labels and formats to label axes and legends. If no variable labels or formats exist, variable names and default formats are used. You can create temporary variable labels and formats with LABEL and FORMAT statements.

```
proc chart data=sashelp.class;
  title1 'Increase of Height with Age';
  title2 '(Average Height in Inches)';
  format height 5.;
  label age='Age to Nearest Year'
        height='Height';
  vbar age / discrete type=mean mean
        sumvar=height;
run;
quit;
```

Increase of Height with Age
(Average Height in Inches)

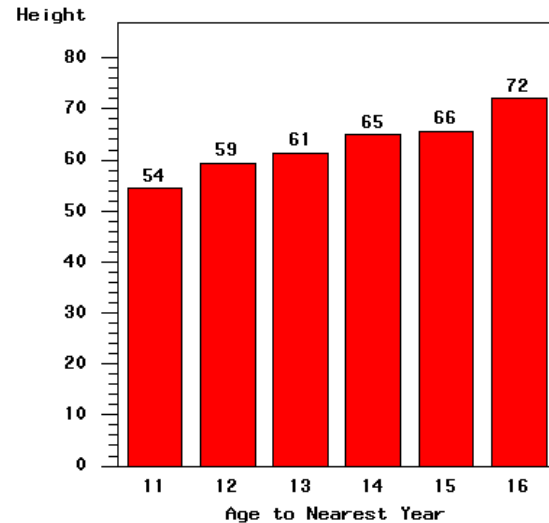


Figure 6 Titles, Variable Labels, and Formats

RUN-GROUP PROCESSING

You may have wondered why the examples programs have concluded with RUN and QUIT statements. The five graphics procedures mentioned so far support RUN-group processing.

What this means is that you can create multiple graphs from one procedure step. Whenever a RUN statement is encountered, a graph is produced. The QUIT statement formally terminates the procedure.

Here is an example of producing two bar charts:

```
proc chart data=sashelp.class;
  title1 'Increase of Height with Age';
  title2 '(Average Height in Inches)';
  format height weight 5.;
  label age='Age to Nearest Year'
        height='Height' weight='Weight';
  vbar age / discrete type=mean mean
        sumvar=height;

  run;
  title1 'Increase of Weight with Age';
  title2 '(Average Weight in Pounds)';
  vbar age / discrete type=mean mean
        sumvar=weight;

  run;
quit;
```

Note that FORMAT and LABEL statements stay in effect across RUN groups. You can have as many RUN groups as you wish and use any one of the procedure's action statements per RUN group.

BY-GROUP PROCESSING

The same five procedures support BY-group processing. When you specify a BY statement, a separate graph is produced for each distinct value of the BY variable(s).

The usual restrictions apply – your data must be presorted by the BY variable(s) for this to work.

```
proc sort data=sashelp.class out=class;
  by sex;
run;
```

```
proc chart data=class;
  vbar age / discrete type=freq;
  by sex;
run;
quit;
```

This program produces one chart per distinct value of the SEX variable. Each chart is automatically labeled with the BY-group value.

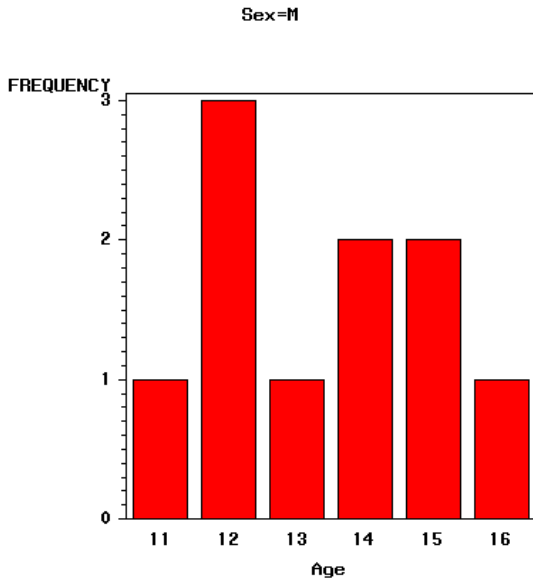


Figure 7 SEX as BY Variable (second of two graphs)

WHERE PROCESSING

Graphics procedures support WHERE processing. No sorting or other preprocessing is required for WHERE processing.

```
proc chart data=sashelp.class;
  title1 'Frequency of Males by Age';
  where sex = 'M';
  vbar age / discrete type=freq;
run;
quit;
```

It is a good idea to add a title or footnote to indicate the portion of the data contributing to the graph.

CREATING PLOTS

In its simplest form, the GLOT procedure uses the PLOT statement to create a two-dimensional scatter plot. Notice that two variables are required and the Y (vertical) variable is specified first:

```
PROC GLOT DATA=SAS-data-set ;
  PLOT Y-variable *X-variable ; plot using left vertical axis
```

For example,

```
proc gplot data=sashelp.class;
  title 'Scatter Plot of Height and Weight';
  plot height* weight;
run;
quit;
```

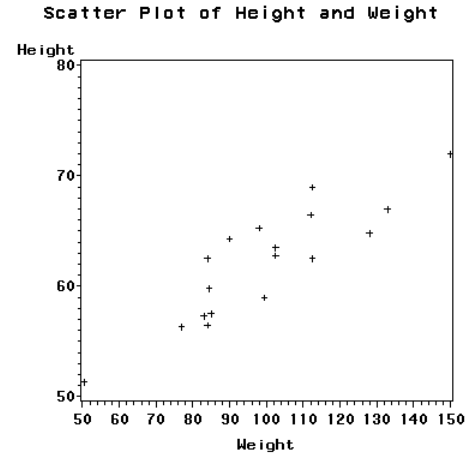


Figure 8 Simple Scatter Plot

SUBGROUPS

The PLOT request supports an optional third variable that lets you identify subgroups of the data. Each value of the subgroup variable will be shown in a different color or marker symbol.

You can control exactly what markers are used by defining one or more SYMBOL statements. The marker color and size can also be specified:

```
symbol1 color=blue value=square height=2;
symbol2 color=red value=triangle height=2;
```

```
proc gplot data=sashelp.class;
  title 'Scatter Plot of Height and Weight';
  plot height* weight = sex;
run;
quit;
```

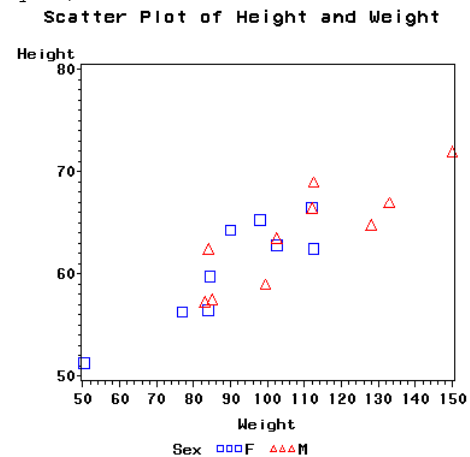


Figure 9 Subgrouped Scatter Plot

PLOTS WITH INTERPOLATIONS

The SYMBOL statement is also used to specify an *interpolation* among the data points. There are many values for INTERPOL= that are statistical in nature. Some draw a line of best fit (e.g. regression, spline) and others draw distributions of the data (e.g. box and whisker, standard deviation).

```

title1
  'Linear Regression of Height and Weight';
title2 '(with 95% Confidence Limits)';

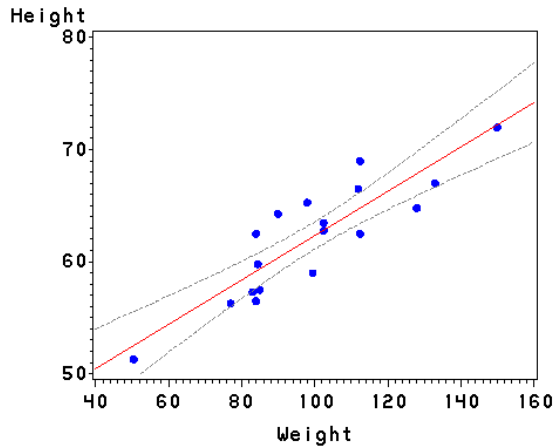
symbol ci=red cv=blue co=gray value=dot
       interpol=rlclm95 ;

proc gplot data=sashelp.class;
  plot height*weight / regeqn;
run; quit;

```

INTERPOL=RLCLM95 requests a linear regression with bands drawn at the 95% confidence limits of the mean predicted values. The REGEQN option causes the derived equation to be displayed.

Linear Regression of Height and Weight (with 95% Confidence Limits)



Regression Equation:
Height = 42.57014 + 0.197615*Weight

Figure 10 Regression Plot

The simplest interpolation is JOIN which draws line segments connecting the data points. Join plots are often used to show trends in your data.

Here we use some stock market data with one observation per year. There are variables with various measurements such as total shares traded and average share price.

```

symbol1 color=blue interpol=join;
proc gplot data=sasuser.nyse;
  plot shares*year / hminor=0;
run;
quit;

```

The HMINOR=0 option was added to get rid of the minor tick marks on the horizontal axis. We could have displayed markers but choose not to.

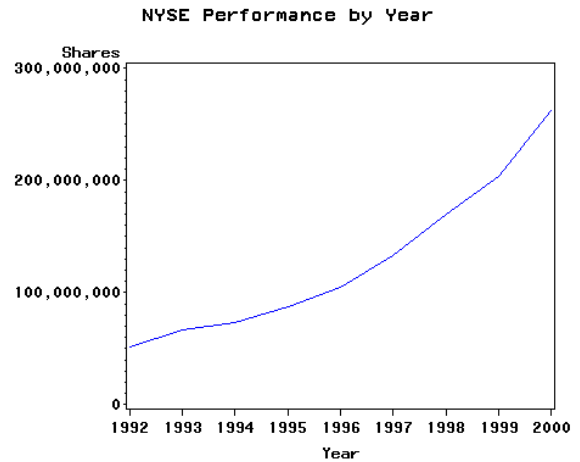


Figure 11 Simple Join Plot

OVERLAY PLOTS

The PLOT statement can draw any number of plots overlaid on top of each other. The syntax for requesting this is to separate each plot request with a blank and to specify the OVERLAY option. Each plot request must use the same X variable:

```
PLOT Y1*X Y2*X Y3*X / OVERLAY;
```

(If you omit the OVERLAY option, separate non-overlaid plots will be produced.)

When there is a large variation in the range of values (or different units of measurement) overlaying plots on the same vertical axis can obscure what you are trying to show. In such cases, you can use a PLOT2 statement to create one or more overlays on a right axis. The syntax for PLOT2 is basically the same as PLOT, the only difference is that the overlays are drawn on the right vertical axis. Once again, the horizontal variable must be the same.

```

symbol1 color=blue interpol=join;
symbol2 color=red interpol=join;

```

```

proc gplot data=sasuser.nyse;
  plot shares*year / caxis=blue hminor=0
    noframe;
  plot2 price*year / caxis=red;
run;
quit;

```

This program also turned off the border that is normally enclosed the axis space (NOFRAME) and colored the axes so it was easy to match the join line and its axis (CAXIS=).

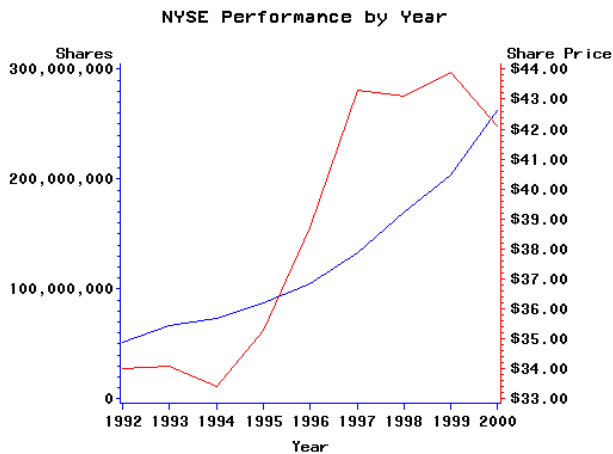


Figure 12 Plot with Left and Right Axes

This paper does not show examples of other very useful SAS/GRAPH statements such as AXIS and LEGEND which are used to fine-tune axis labels, axis scales, tick marks, and legend contents for many procedures. As you have seen, the default axes and legends are always appropriate, but sometimes you want more precise control of the formatting and layout. For example, you can request logarithmic axis scales.

CONTROLLING FINAL OUTPUT

Your goal of creating a specific graph also includes outputting it to a desired medium - a printer, an image file that can be inserted in a document or HTML page, or some other proprietary format like RTF or PDF. To do this, you must designate a graphics device driver that will convert your working graph into a format appropriate for the output medium. You will also make adjustments to the program so the graph is sized as you want, the desired fonts are used, and the output file is specified. Many these details are addressed by setting appropriate graphics options via the GOPTIONS statement.

The recommendations that follow apply to any chart, plot, or graph you can create. We will look at three common issues that need to be addressed for any output format: the device driver, the graph size, and fonts used.

DEVICE DRIVERS

Device drivers are components of SAS/GRAPH software that translate device-independent output from procedures (the working graphs shown so far) into commands that produce similar output on a particular output device.

Let's say that we wish to create an image file for our graph. SAS/Graph supplies device drivers for most commonly used image formats including GIF, JPEG, PNG, BMP, etc. To set the device driver, use the DEVICE= option:

```
options device=gif;

/* your SAS/GRAPH program */
```

When you specify a device driver, you may notice that the output graph changes size or the default font used.

SIZING THE GRAPH

The next issue to address is output size. The default size for the GIF driver is 8.42" by 6.31" (800 by 600 pixels), which may be too large if you want to insert the graph into some document. Let's say that a 4" by 3" graph would be more appropriate. You use the HSIZE= and VSIZE= options to set the horizontal and

vertical dimensions of the output.

```
options device=gif
        hsize=4in vsize=3in
        border;

/* your SAS/GRAPH program */
```

Notice that sizes you select affect the aspect ratio - you could create a narrower graph by setting HSIZE=3in and VSIZE=4in. Size dimensions can be specified in inches (IN), centimeters (CM), or points (PT).

The BORDER goption is very handy for checking graph size. It draws a frame around the graphics area and provides visual confirmation for the size you set. Whether you want this frame displayed on your final output is up to you. Figure 14 shows a border.

TIP: The GRAPH window has a feature called AUTOSIZE which is set to ON by default. This causes the window contents to grow and shrink as you change window size. This is not helpful when you are trying to create a graph of a certain size. Issue the AUTOSIZE OFF command and then the WSAVE command to make graphs maintain their size and aspect ratio in this window.

FONT AND FONT SIZE

SAS/GRAPH uses fonts supplied with SAS software by default. However, SAS/GRAPH can use any font installed on your system for display purposes, as long as you request it. To learn what fonts are available on your system, you can issue the FONTLIST command. This displays a dialog from which you can list all installed fonts and available sizes, along with font samples.

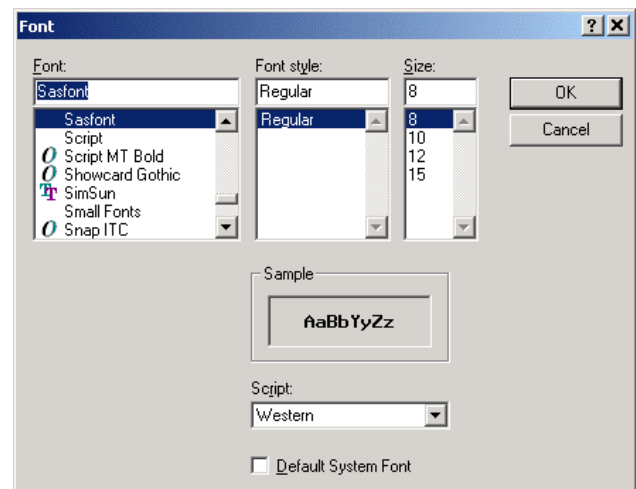


Figure 13 Showing Installed Fonts

If you use a font name that is not in the installed list, a substitution will be made. The FTEXT= goption sets the font for all text in the graph, including titles and footnotes. Many fonts will automatically scale up or down in size based on the graph size. To assure that the text size is appropriate, use HTEXT= goption to force the font size.

```
options device=gif
        hsize=4in vsize=3in
        border
        ftext='Helvetica' htext=12pt;

/* your SAS/GRAPH program */
```

If you wish to differentiate fonts, font size, and font style in your titles or footnotes, use the FONT= and HEIGHT= options on each TITLE or FOOTNOTE statement. Notice that you can specify font style attributes by including /BOLD or /ITALIC as part of the font name.

```

title1 font="helvetica/bold"
      'Increase of Height with Age';
title2 font='helvetica/italic' height=10pt
      '(Average Height in Inches)';

/* your SAS/GRAPH program */

```

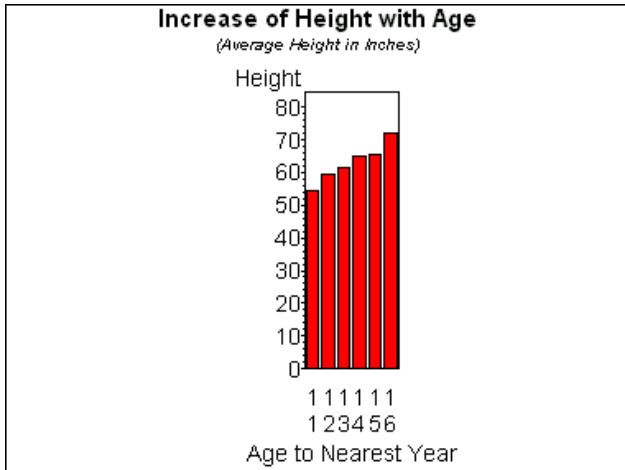


Figure 14 GIF Output with Fonts and Image Size Assigned

The size and fonts look OK, but the chart is too narrow, causing the horizontal axis values to rotate and the bar values not to be displayed. To control the width of the chart within the graphics area, use the HPOS= option. The value you assign to HPOS changes the number of drawing units (*cells*) for the chart. Generally, a number between 100 and 30 is used. The larger the HPOS value, the narrower the chart. Making HPOS too small will cause the chart to extend beyond the graphics area and be clipped.

```

goptions device=gif
      hsize=4in vsize=3in
      border
      ftext='Helvetica' htext=12pt
      hpos=40;

/* your SAS/GRAPH program */

```

An appropriate HPOS value depends on many factors including the number of bars, so a "good" HPOS value for one chart is not necessarily appropriate for another. Plots, maps and other chart types often don't require any HPOS adjustment.

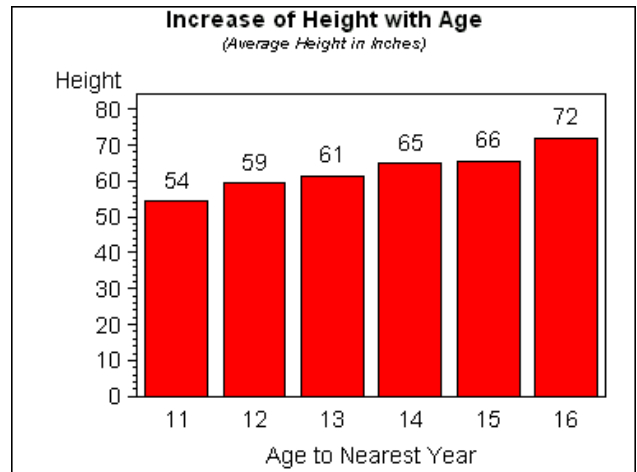


Figure 15 GIF Output with Adjusted Chart Width

CREATING THE OUTPUT FILE

So far we have been using the GRAPH window to preview output. Once the graph looks OK, you need to tell SAS/GRAPH to create an output file and to provide the path and filename for it.

To establish the file path, use a FILENAME statement to point to the directory where the output is to be written. Any fileref can be used. Then include the GSFNAME= option with this fileref to make SAS/GRAPH write output to this directory.

This program creates two graphs and two output files. We have added the NAME= option to the VBAR action statements to specify the name of each output file. Because the device driver is GIF, the two output files will be named as follows:

```

C:\myGraphs\Height.gif
C:\myGraphs\Weight.gif

```

If you don't name your output, each SAS/GRAPH procedure will make up names for you such as GCHART.GIF and GCHART1.GIF

```

filename out 'C:\myGraphs';

goptions device=gif
      hsize=4in vsize=3in
      border
      ftext='Helvetica' htext=12pt
      hpos=40
      gsfname=out;

proc chart data=sashelp.class;
  title1 font="helvetica/bold"
    'Increase of Height with Age';
  title2 font='helvetica/italic' height=10pt
    '(Average Height in Inches)';

  format height 5.;
  label age='Age to Nearest Year'
    height='Height';
  vbar age / discrete type=mean mean
    sumvar=height name='Height';
run;

```



```

title1 font="helvetica/bold"
  'Increase of Weight with Age';
title2 font='helvetica/italic' height=10pt
  '(Average Weight in Pounds)';

format weight 5.;
vbar age / discrete type=mean mean
  sumvar=weight name='Weight';

run;
quit;

filename out;

```

TIP: As you develop your SAS/GRAPH programs, every time you create output the temporary catalog WORK.GSEG keeps growing in size and assigning unique names to each output graph. To assure your output names are used, it is recommended that you remove your temporary output files before the final run. Use this program to remove the entire WORK.GSEG catalog (or delete it from a SAS Explorer window).

```

proc catalog c=work.gseg kill;
run;

```

After submitting this, the next SAS/GRAPH program that creates output will automatically create a new WORK.GSEG catalog.

USING THE OUTPUT DELIVERY SYSTEM (ODS)

For printing graphs or creating PDF, PDF, or HTML files it is recommended that you use the simple syntax of ODS. One major advantage of ODS is that the output of multiple SAS procedures (SAS/GRAPH and other products) can be printed or included in one output file.

Use the ODS PRINTER destination to print your output:

```

ods printer printer='printer-name';
/* your SAS procedures */
ods printer close;

```

If you omit the PRINTER= option, the default printer is used.

To create output files from ODS use:

```

ods destination file='filename';
/* your SAS procedures */
ods destination close;

```

where **destination** can be PDF, RTF, HTML.

Output destinations such as PDF and HTML can have a table of contents. Here is a PDF example using the GIF driver. It uses additional ODS statements and SAS/GRAPH features to improve the table of contents:

```

ods pdf file='C:\mypdf.pdf'
  style=sansprinter;

options nodate nonumber;
goptions device=gif ftext='helvetica'
htext=12pt hpos=40;
ods proclabel='Charts';
proc gchart data=sashelp.class;
label height='Height' weight='Weight';
format height weight 5.1;
title1 font='helvetica/bold' height=16pt
  'Increase of Height with Age';
title2 font='helvetica/italic' height=12pt
  '(Average Height in Inches)';

```

```

vbar age / discrete type=mean
  sumvar=height mean
  des='Mean Height by Age';

run;
title1 font='helvetica/bold' height=16pt
  'Increase of Weight with Age';
title2 font='helvetica/bold' height=12pt
  '(Average Weight in Pounds)';
vbar age / discrete type=mean
  sumvar=weight mean
  des='Mean Weight by Age';

run;
quit;
ods noproctitle;
ods proclabel='Statistics Table';
title font='helvetica' bold height=16pt
  'Statistics Table';
proc means data=sashelp.class maxdec=1;
class age;
var height weight;
run;

ods pdf close;

```

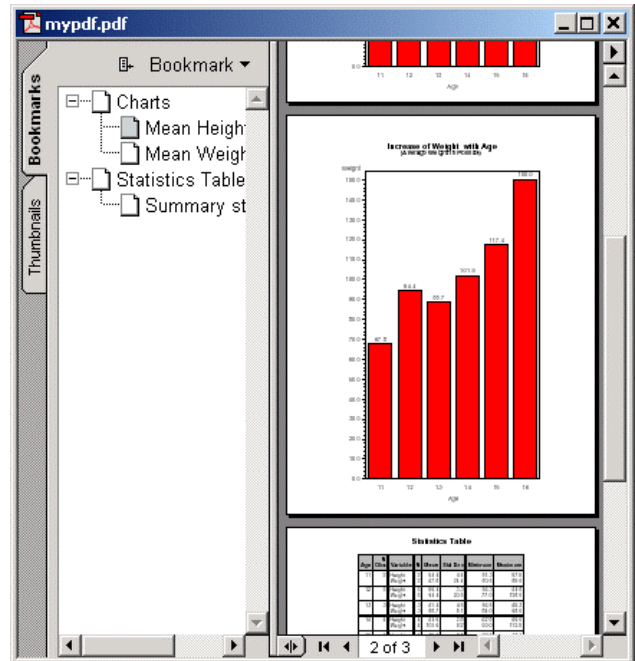


Figure 15 PDF Output

When using any ODS destination for output, you can set your graphics device driver to GIF, or use a *client* device driver of ACTIVEX or JAVA. Client drivers are used only in conjunction with ODS:

```

ods html path='C:\' style=statdoc
  file='myhtml.html'
  contents='mycontents.html'
  frame='myframe.html'
  (title='My Output');

goptions reset=all device=activex
  ftext='helvetica' htext=12pt
  hsize=5in vsize=3in;

/* same program used with PDF example*/

ods html close;

```

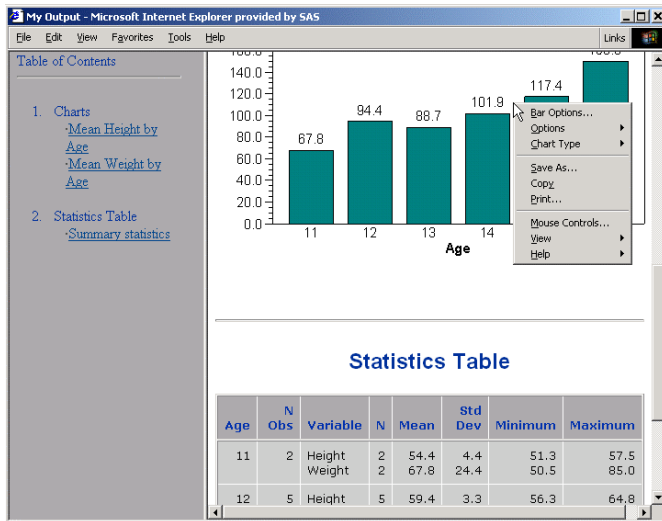



Figure 16 HTML Output Using ACTIVEV Driver

A client device driver uses its own technology to render the graph. Behind the scenes your SAS/GRAPH procedure code is translated into instructions specific to JAVA or ACTIVEV for drawing equivalent charts, plots, and maps. There are no additional image files as would be necessary with the GIF driver.

One of the distinguishing characteristics of using client devices is that their graphs are interactive when used with ODS HTML or ODS RTF. This means the graphs display bar or point values with a simple mouse-over action. There are built-in context menus to let you change chart types and appearance options.

USING GRAPH-N-GO TO CREATE GRAPHS

Beginning with Version 8, Graph-N-Go is a production application integrated into SAS Display Manager as part of SAS/Graph Software.

The purpose of this application to provide a simple, interactive environment to create and modify a variety of charts and plots based on SAS data sources. You use point and click and drag and drop to create your graphs, instead of writing an SAS programs.

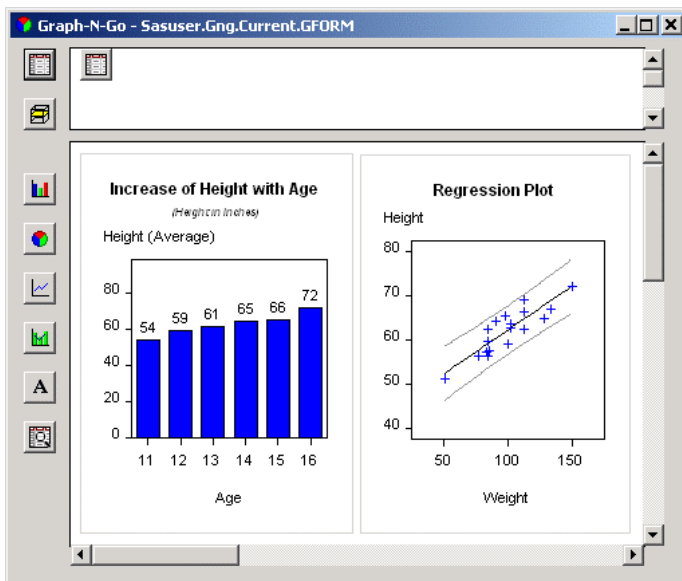


Figure 17 Graph-N-Go Window

To start Graph-N-Go choose

Solutions-> Reporting-> Graph-N-Go
from Display Manager (or type GNG on any command line).

This application does not create the kinds of graphs that the G3D, CONTOUR, or GMAP procedures do, but it does create most of the bar and pie charts from GCHART and most of the plots from GPLOT. Version 9 will offer support for histograms, block charts, and overlay plots. There is very extensive help available within the application.

The graphs you create in Graph-N-Go can be printed, or exported to image files, HTML files, or source files. Although GNG does not use SAS/GRAPH to render its graphs, it can create a SAS/Graph program that will produce very similar output.

Here is portion of program automatically generated for the bar chart shown in Figure 17. There are many comments added to the program including documentation references and example ODS statements.

```

/* Set the SAS/Graph options */
options reset=all hpos=40
      ctext=CX000000 ftext="Arial"
      colors=(CX0000FF CXFF0000
              CX008080 CX00FF00
              CXFF00FF CXFFFF00
              CX00FFFF CX800000
              CX008000 CX800080
              CX000080 CX808000
              CXFFFFFF CX808080
              CXC0C0C0 CX000000);

/* Set the Titles/Footnotes */
title1 justify=center color=CX000000
      font="Arial" height=12 pt
      "Increase of Height with Age";
title2 justify=center color=CX000000
      font="Arial" height=12 pt
      "(Height in Inches)";

/* Set the SAS/Graph device driver */
* goptions device=JAVA xpixels=252
          ypixels=332;

/* AXIS1 for Category variable AGE */
/* AXIS2 for Response variable HEIGHT and
statistic MEAN */
axis1 minor=none label=("Age") ;
axis2 minor=none label=("Height (Average)")
      order=(0 to 80 by 20) ;

proc gchart data=SASHELP.CLASS;
  vbar AGE /
    type=MEAN sumvar=HEIGHT maxis=axis1
    discrete frame cframe=CXFFFFFF
    outside=MEAN caxis=CX000000
    woutline=1 coutline=CX000000
    raxis=axis2;
run;
quit;




```

CONCLUSION

In such a short amount of space it is not possible to show or explain all the features of SAS/GRAPH software. Hopefully, you have a better idea of what the software can do and understand some of its basic programming constructs. Because graphs can be created from programs, you can set up batch jobs that automatically produce graphs from production data sets. This is a major plus for people creating large scale systems with SAS. For scientists and statisticians, the analytical capabilities of GPLOT are in high demand.

FOR MORE INFORMATION

The online help for SAS/GRAPH software is very complete:

- [-]  SAS/GRAPH Software
 -  What's New in Release 8.2
 -  What's New in Release 8.1
 - [+]  SAS/GRAPH Procedures
 - [+]  SAS/GRAPH Statements
 -  Graphics Options and Device Parameters Dictionary
 -  Annotate Dictionary
 -  Using Graphics Devices
 -  Generating Graphs for the Web
 - [+]  Working with Images
 -  Using the Graphics Editor
 - [+]  Other Ways to Create Graphs

You can find a copy of this paper and the others contributed by staff members of the Data Visualization Division by visiting SAS Institute's web site:

<http://www.sas.com/rnd/datavisualization/papers/index.htm>