Paper 262-30

An Introduction to the Simplicity and Power of SAS/Graph®

Dan Bruns, Chattanooga, TN

IN THE BEGINNING

In today's hectic and ever increasing demand for more information faster, the graphics tools for SAS are a great way to get lots of information conveyed quickly. I have seen tremendous enhancements and improvements in the product since it was first introduced (and first learned it) in the late 1970s. But the basic, underlying, data-driven principals (like for most SAS procedures) are still there and easy to learn. In this tutorial we will explore how a little understanding and knowledge of the basic concepts and options of SAS/Graph will get you on your way to producing all kinds of impressive and informative graphs that can be printed, imbedded in other docs, or made available to web sites.

SOME BASICS

There are several procedures and statements that come with the SAS/GRAPH product, but we will only cover a few of the most powerful procedures and all the global statements in order to give you a flavor of what can be done. Within each of the procedures and statements we cover there will be *dozens* of options, but we will cover just a few of the most powerful.

SAS/GRAPH Procedures

GANNO -- displays graphs created by Annotate data sets.
GAREABAR -- enables you to produce an area bar chart showing the magnitudes of *two* variables for each category of data.
GBARLINE -- produces bar line charts. Bar line charts are vertical bar charts with a plot overlay.
GCHART -- produces six types of charts: block charts, horizontal and vertical bar charts, pie and donut charts, and star charts. These charts graphically represent the value of a statistic calculated for one or more variables in an input SAS data set.
GCONTOUR -- produces plots that represent three-dimensional relationships. The colors, contours, or surface areas of a contour plot represent the values of a contour variable at each point in a plane that is formed by a dependent and an independent variable.
GDEVICE -- is a tool for examining and changing the parameters of the graphics device driver catalog entries used with SAS/GRAPH software.
GFONT -- displays new or existing fonts and creates user-generated fonts for use in SAS/GRAPH programs.
GIMPORT -- enables you to import into SAS/GRAPH software graphics output that is produced with other software applications, graphics output that is produced by SAS/GRAPH software, or graphics output that is produced on other machines.
GMAP -- produces two-dimensional (choropleth) or three-dimensional (block, prism, and surface) color maps that show variations of a variable value with respect to an area.
GOPTIONS -- provides information about the values of graphics options and the global statement definitions that are currently in effect in your session.
GPLOT -- plots the values of two or more variables on a set of coordinate axes (X and Y). The coordinates of each point on the plot correspond to two variable values in an observation of the input data set.
GPRINT -- converts a text file into graphics output that can be displayed or printed on a graphics output device.
GPROJECT -- processes traditional map data sets by converting spherical coordinates (longitude and latitude) into Cartesian coordinates for use by the GMAP procedure.
GMAP -- produces two-dimensional (choropleth) or three-dimensional (block, prism, and surface) color maps that show variations of a variable value with respect to an area.
GRADAR -- creates radar (or star) charts that show the relative frequency of data measures in quality control or market research problems. (The chart statistics are displayed along spokes that radiate from the center of the chart, hence the term "star" charts).
GREDUCE -- processes map data sets so that they can draw simpler maps with fewer boundary points.
GREMOVE -- combines unit areas defined in a map data set into larger unit areas by removing shared borders between the original unit areas.
GREPLAY -- displays and manages graphics output that is stored in SAS catalogs. The GREPLAY procedure also creates templates and color maps that you can use when you replay your graphics output.
GSLIDE -- is useful for creating text slides for presentations. You can overlay text slides on other graphics output with

the GREPLAY procedure.
GTESTIT -- is a diagnostic tool for testing the installation of SAS/GRAPH software and the configuration of your device.
G3D -- produces three-dimensional graphs that plot one vertical variable (z) for a position on a plane that is specified by two horizontal variables (x and y).
MAPIMPORT -- enables you to import ESRI shapefiles (spatial data formats) and process the SHP files into SAS/GRAPH traditional map data sets.

SAS/GRAPH Global Statements

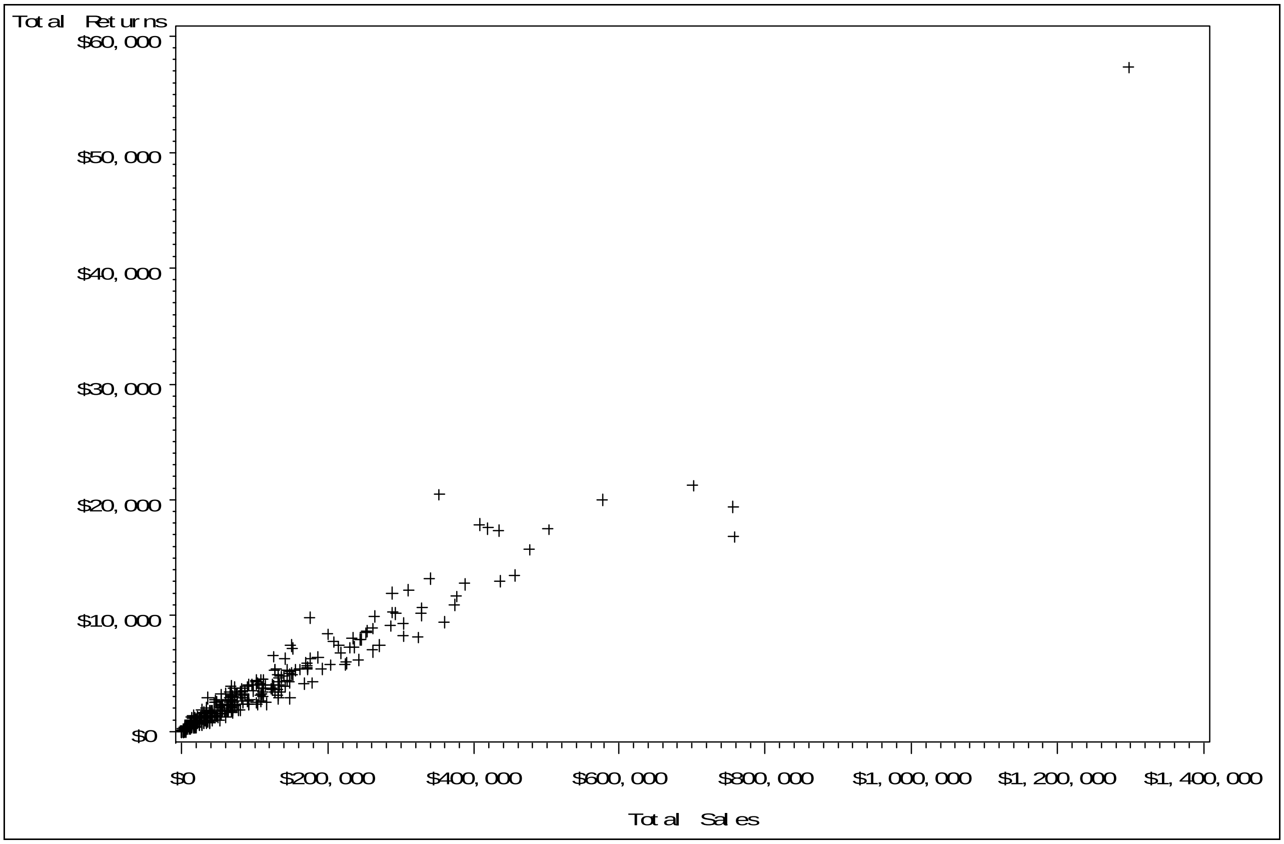
AXIS -- modifies the appearance, position, and range of values of axes in charts and plots.
GOPTIONS -- specifies graphics options that control the appearance of graphics elements by specifying characteristics such as default colors, fill patterns, fonts, or text height. Graphics options can also temporarily change device settings.
LEGEND -- modifies the appearance and position of legends generated by procedures that produce charts, plots, and maps.
PATTERN -- controls the color and fill of patterns that are assigned to areas in charts, maps, and plots.
SYMBOL -- specifies the shape and color of plot symbols as well the interpolation method for plot data. It also controls the appearance of lines in contour plots.
TITLE and **FOOTNOTE** -- add titles and footnotes to graphics output.

GPLOT PROCEDURE

One of the really unique features of most SAS procedures is that they are driven by the data. And that becomes even more apparent when you see the output of the SAS/GRAPH procedures. We are going to begin with the GPLOT procedure because everyone has plotted data points a graph at some point. And that is all SAS is going to do….plot a point (with an x-value and a y-value) on the graph for each observation.

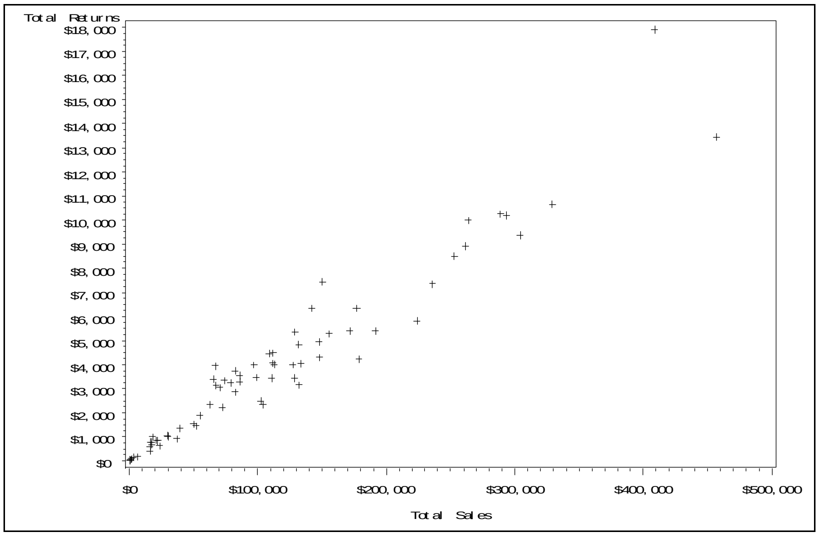
With the PLOT statement you request plots to be generated with sets of **Y-var * X-var** pairs. GPLOT and several of the graph procedures support run-group processing.

```
proc gplot data=sashelp.shoes;
   plot   Returns * Sales   ;
   run;
```



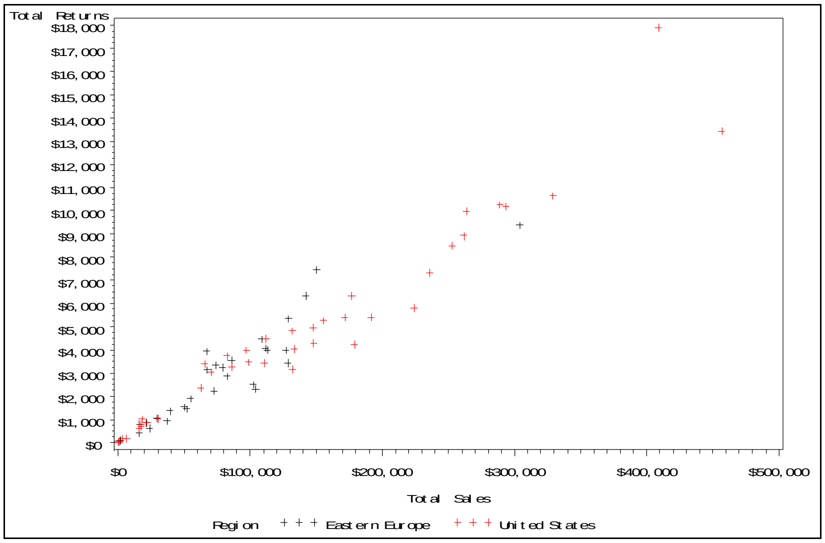
```
where Region
       in("United States",
       "Eastern Europe");
plot  Returns * Sales;
run;
```

Notice the default plot symbol is a plus sign and black is the default color for most everything. But rest assured that and virtually everything else about the graph you are able to control. In the second plot, can you tell which Region each point is from?



```
where Region
   in("United States",
   "Eastern Europe");
plot Returns * Sales = Region;
run;
```

Using the Y*X=variable notation tells SAS to cycle through the colors from the devices color list for each unique value of Region. If there are more values than colors in the list, another symbol is chosen and the colors cycle again. The windows display is known as the WIN device and has 11 colors (black, red, green, blue, cyan, magenta, grey, pink, orange, brown, and yellow).



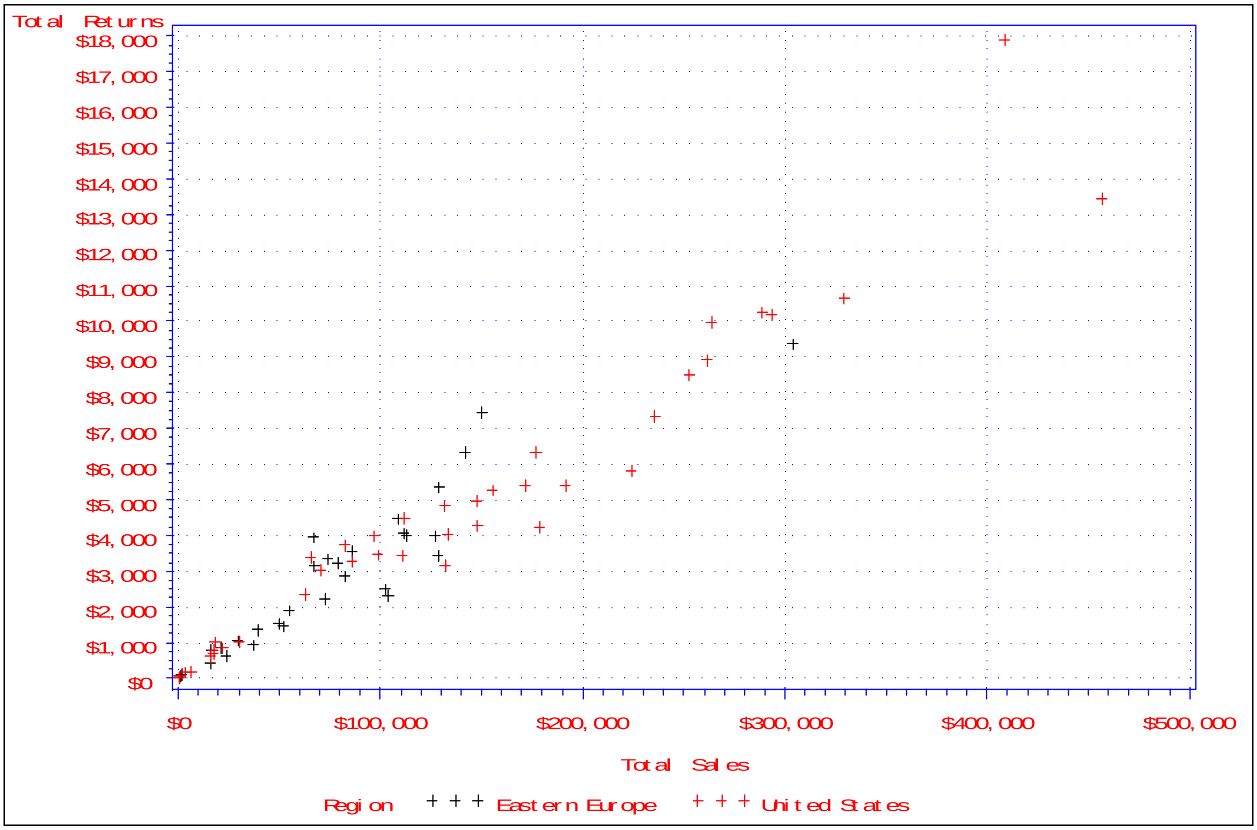
In nearly all the graphics procedures, the statement that requests a certain kind of graphic will also allow you to specify options to control and enhance the graph. To specify those options on the PLOT statement, simply follow the plot request(s) with a slash followed by the **plot options** you want applied to the requested plots. There are far too many to cover all of them in this tutorial, so we will mention a few of the more powerful ones.

You will notice that most options in SAS/GRAPH that begin with the letter "C" will refer to a **color** specification, those that begin with the letter "H" will refer to a **height** specification, those that begin with the letter "F" will refer to a **font** specification, and those that begin with the letter "L" will refer to a **line-type** specification.

When in doubt about any specification, such as a color or font, always refer the Reference Manual. Especially line-types since they are a coded value of 1 to 46 (1 being solid; 2-46 some form of dashes). Over time you will get more use to them and develop your favorites.

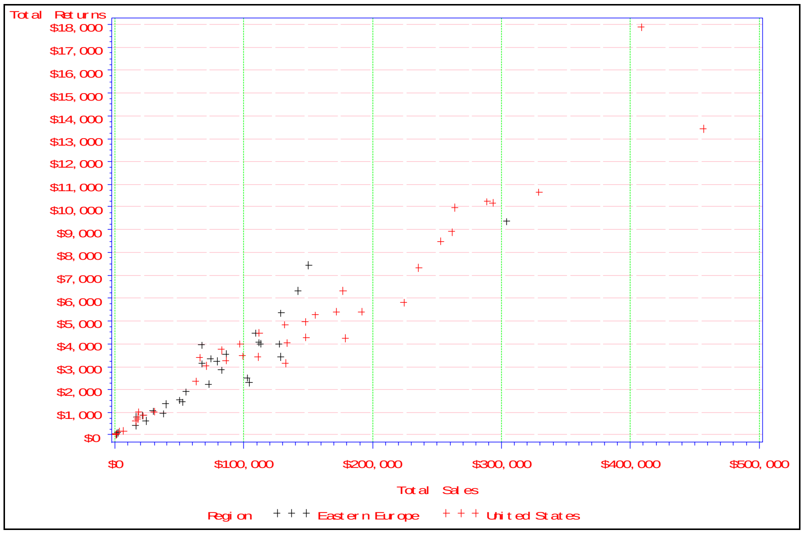
```
where Region
   in("Unites States","Eastern Europe");
plot Returns * Sales=Region /
       caxis=blue
       ctext=red
       grid ;
       run;
```

The CAXIS defines the color for all the axis and CTEXT defines the color for all the text on the plot. By specifying the GRID option you get what are called **reference lines** at the major tick marks on the vertical and horizontal axis and will be a dashed line and the same color as the axis.



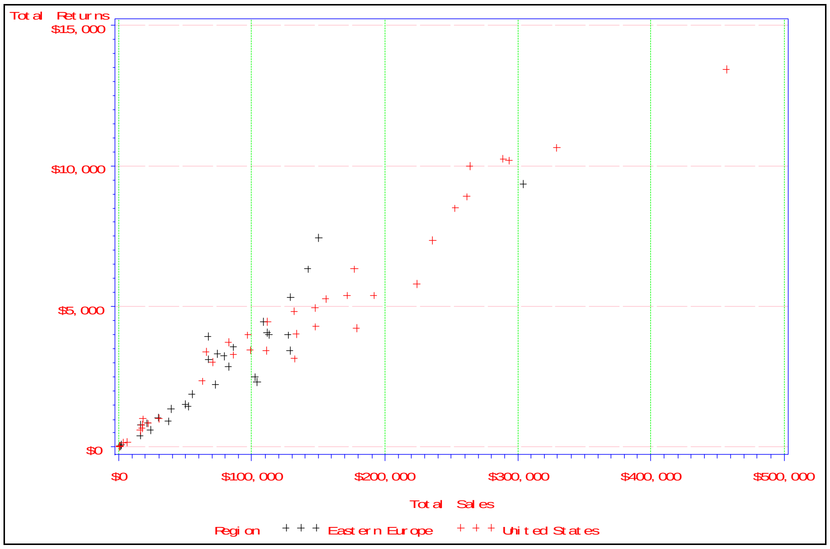
If you need more control of the reference lines, use the AUTOHREF and AUTOVREF options to have SAS automatically generate reference lines. Then use the LHREF (line-type for horizontal ref line), LVREF (line-type for vertical ref line), CHREF (color for horizontal ref line), and CVREF (color for vertical ref line) to control the line-type and color of the reference lines.

```
where Region
   in("United States",
   "Eastern Europe");
plot Returns * Sales=Region /
       autohref  lhref=2
       chref=lime
       autovref  lvref=5
       cvref=pink
       caxis=blue
       ctext=red ;
       run;
```



Another powerful option is VAXIS (and HAXIS) to specify the major tick marks of the vertical axis. If any point lies beyond what you specify, it is NOT plotted—so be careful.

```
where Region
      in("United States",
      "Eastern Europe");
plot Returns * Sales=Region /
   vaxis=0 to 15000 by 5000
   autohref  lhref=2
   chref=lime
   autovref  lvref=5
   cvref=pink
   caxis=blue
   ctext=red ;
run;
```



GLOBAL STATEMENTS

Titles

Using global statements gives you even MORE control of certain aspects of the graphs along with more options to further enhance the graph. Let's take a look at the TITLE statement.

```
title1 c=darkblue h=2.5 f=swissb  "SAS/Graph "
         c=darkred h=3.0 f=swissbi "GPLOT Example";
```

Can you guess what the C=, H=, and F= are specifying?
That's right! – Color, Height, and Font. The quoted text following the options will take on those specifications.

Axis

Instead of using the VAXIS option to define the vertical axis, let's code an AXIS statement (which is much more powerful) and point to it in the plot option.

```
axis1
     label=(c=darkorange h=1.5 f=zapfbi
          j=r "Total Returns")
     offset=(0.2 in )
     order=(0 to 15000 by 5000)
     value=(c=darkorange f=swissl);
```

Since you can (and usually do) have more than one axis statement, you need to number them (from 1 – 99) just like a title statement.

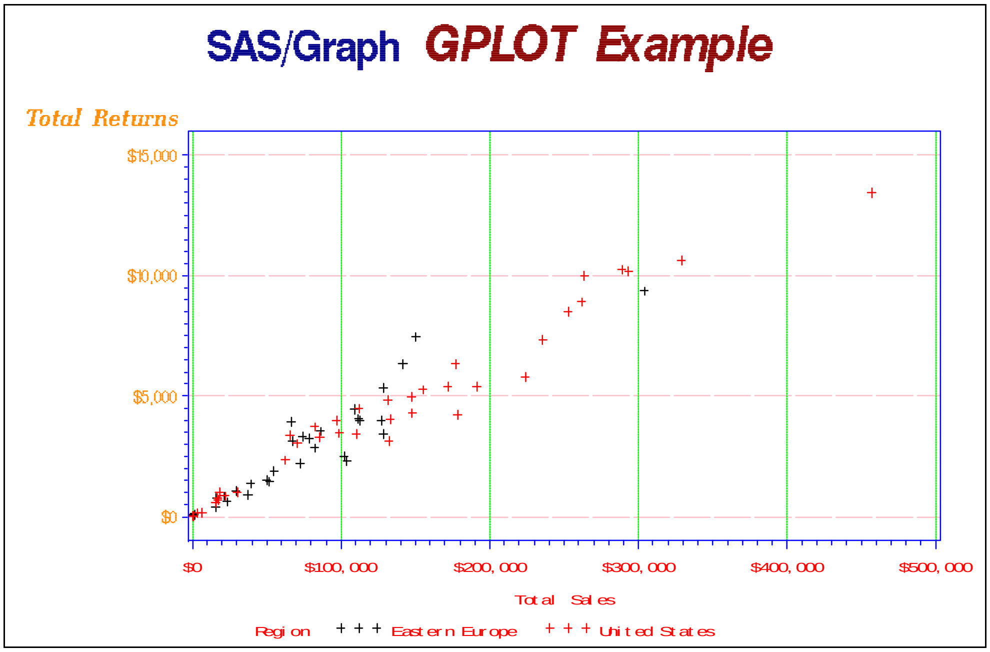
The LABEL= option defines the options for just the label of the axis. The J= option in the justification of the text – left(L), center(C), right(R).

The ORDER= defines the range of values and major tick marks.

The VALUE= define the attributes of the labeling of the axis and the tick marks.

The OFFSET= defines how far from the lower left corner of the graph to start the first tick mark.

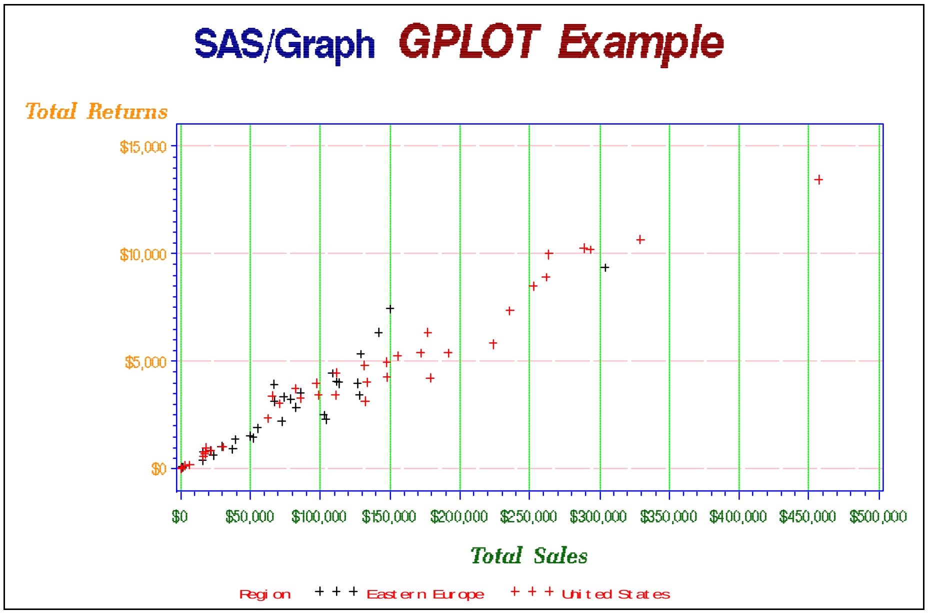
```
proc gplot data=sashelp.shoes;
       where Region in("United States", "Eastern Europe");
       plot Returns * Sales=Region /
              vaxis=axis1  /*vaxis=0 to 15000 by 5000*/
              autohref  lhref=2 chref=lime
              autovref  lvref=5 cvref=pink
              caxis=blue   ctext=red ;
       run;
```



Let’s do the same thing for the horizontal axis.

```
axis2
label=(c=darkgreen h=1.5
       f=zapfbi)
order=(0 to 500000 by 50000)
value=(f=swissl
       c=darkgreen);
```

```
proc gplot data=sashelp.shoes;
where Region in("United States","Eastern Europe");
plot Returns * Sales=Region /
       vaxis=axis1  haxis=axis2
       autohref  lhref=2 chref=lime
       autovref  lvref=5 cvref=pink
       caxis=blue   ctext=red ;
       run;
```



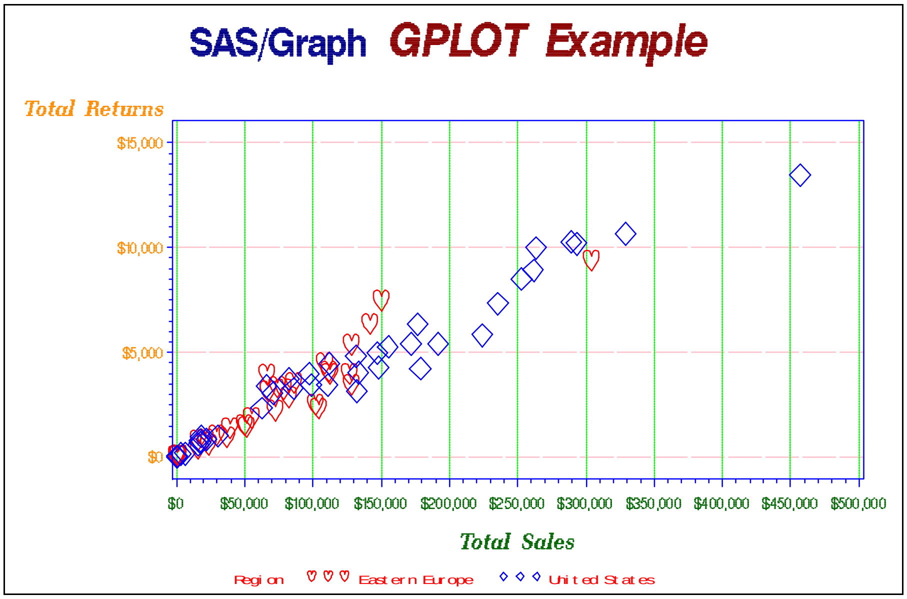
Symbol

Now let's see what we can do with those symbols (points) on the plot. By using the SYMBOL statement you can decide how it is going to look and how to join the point. By default, the points are not joined. Let's define two symbol statements for the two regions.

```
symbol1  c=red  h=2  v=# /*heart*/;
symbol2  c=blue h=3  v=diamond  ;
```

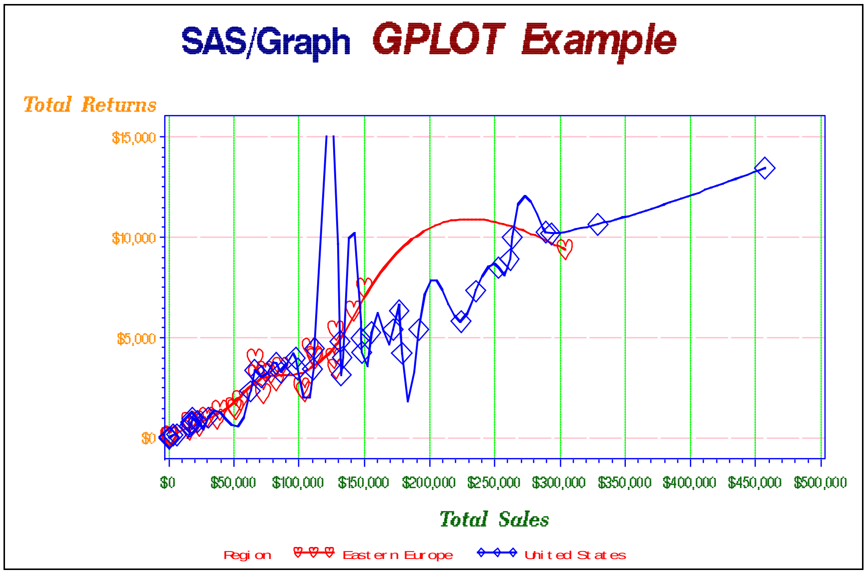
I am sure you know what the C= and H= define. The VALUE=(V=) is used, instead of say the F=, to define what symbol we want. Again, see the Reference Manual for a complete list of possible option values.

If we just rerun the code from earlier, GPLOT will see the new symbol definitions and use them. Simply, if the graph needs 2 symbols it will look for a symbol1 and symbol2 and use them if found. If not, defaults are used.



Let's add an INTERPOLATION=(I=) option to the symbol statements. See the Reference Manual for a complete list of possible interpolation techniques to decide which best fits your needs (if any). The WIDTH=(W=) option defines the width the line drawn through the points.

```
symbol1  c=red  h=2  v=# /*heart*/  i=sm50s   w=2;
symbol2  c=blue h=3  v=diamond      i=splines w=2.5;
```

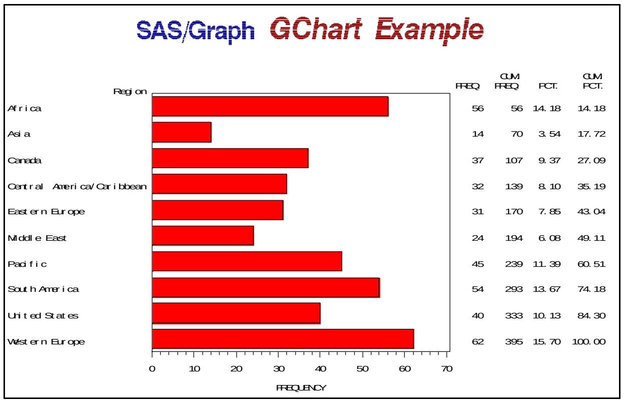


GCHART PROCEDURE

In my humble opinion, GCHART is one of the versatile graphics procedures. It can generate several different kinds of charts which are very popular in print or on the web. Here is an example of each type using just the defaults.

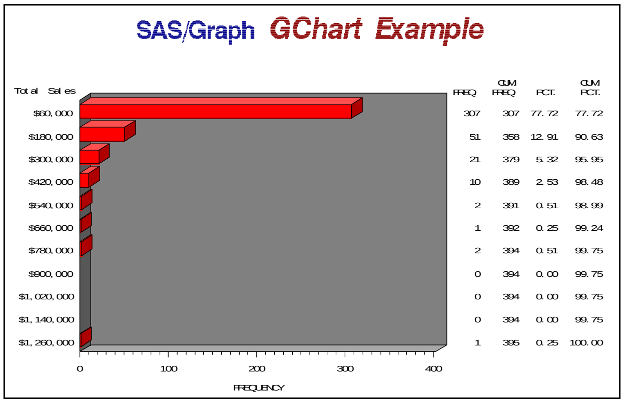
Horizontal Bar Chart (HBAR)

`hbar Region ;`



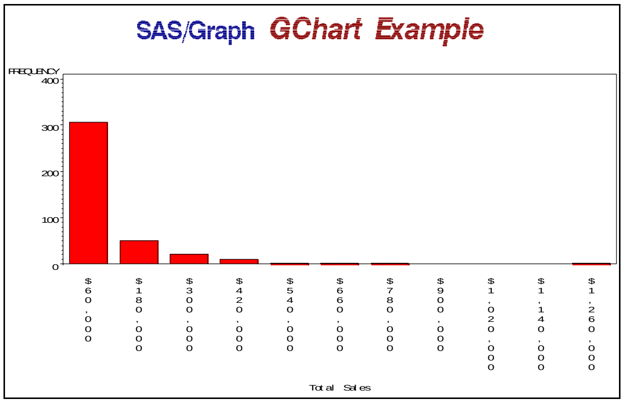
3D Horizontal Bar Chart (HBAR3D)

`hbar3d Sales;`



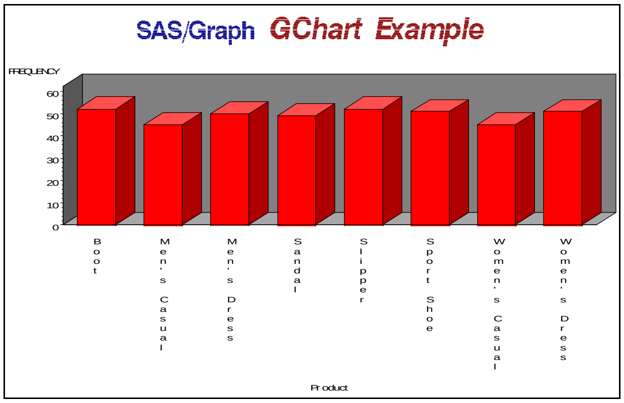
Vertical Bar Chart (VBAR)

`vbar Sales;`



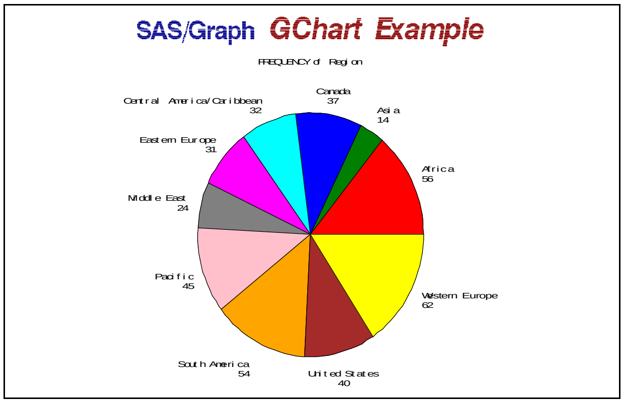
3D Vertical Bar Chart (VBAR3D)

`vbar3d Product ;`



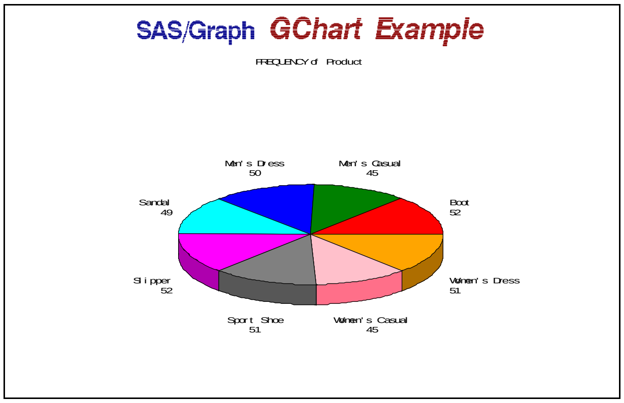
Pie Chart (PIE)

`pie Region ;`



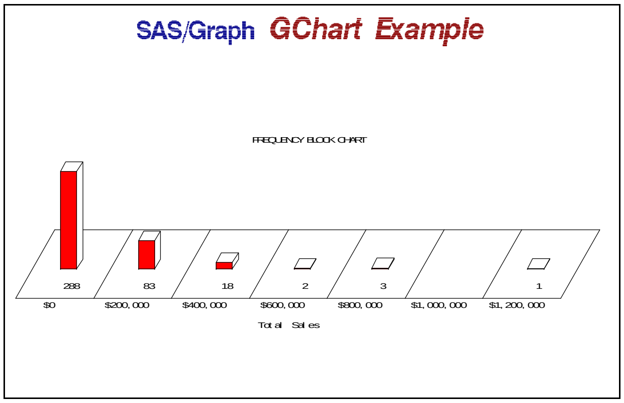
3D Pie Chart (PIE3D)

`pie3d Product ;`



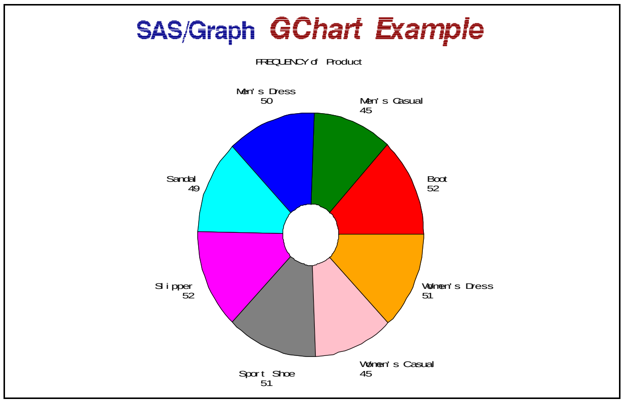
Block Chart (BLOCK)

`block  Sales ;`



Donut Chart (DONUT)

`donut Product;`

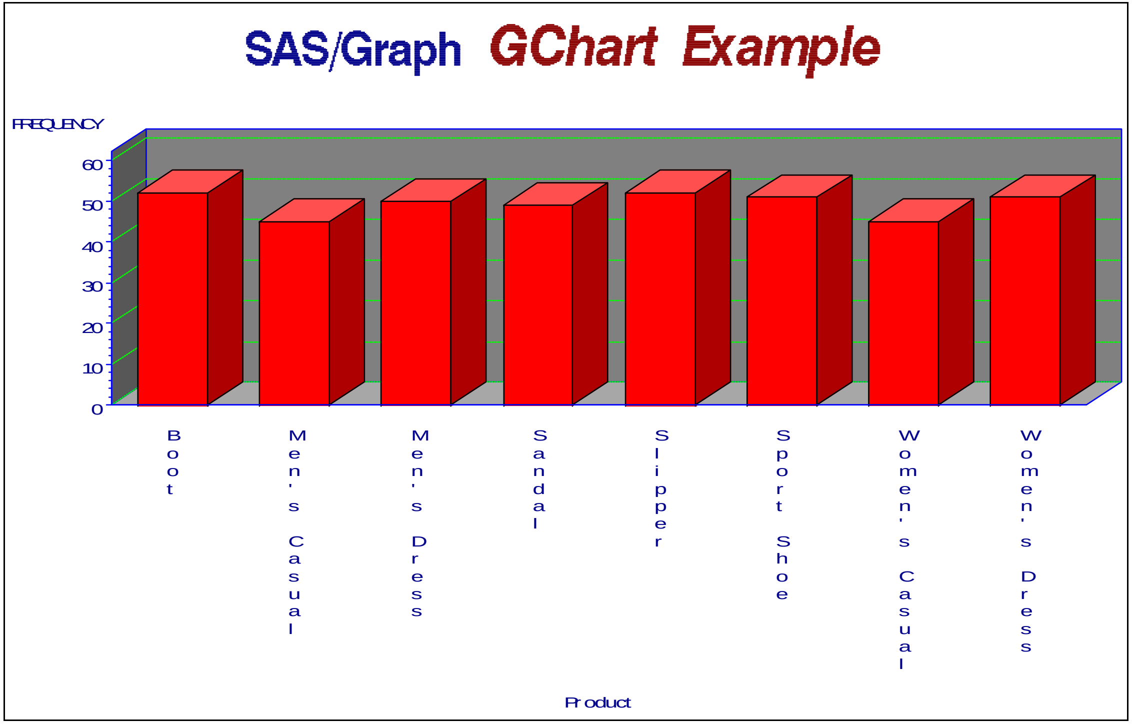


Notice how SAS chooses a ***midpoint*** for each bar, slice, or block. If it is a character variable, it is easy—one for each unique value of the chart variable AND in alphabetical order. BUT, if the chart variable is numeric, SAS computes the midpoints based upon the highest and lowest values of the chart variable then **normally** divides the ***range equally over six or seven bars.*** Generally speaking, if I refer to a bar, the same would generally apply to a pie slice or a block. In the case of our SHOES data there was one observation with a Sales value of almost double the next highest value. But the algorithm used to compute the midpoints is smart enough to account for such outliers and adjust the number of bars accordingly. We'll discuss what the size of the bars mean a little later.

We will concentrate on the bar charts to illustrate some of the options, many of which can be applied to all the types. (And because my personal favorite is the VBAR3D.)

Several of the options we used for GPLOT can be used as chart options. Which one below changed slightly?

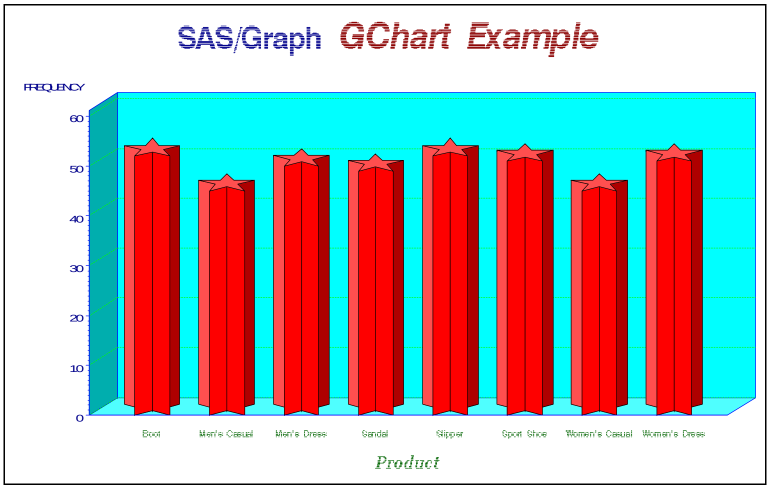
```
title1 c=darkblue h=2.5 f=swissb  "SAS/Graph "
       c=darkred h=3.0 f=swissbi "GChart Example";
proc gchart data=sashelp.shoes;
       vbar3d Product /
              caxis=blue
              ctext=darkblue
              autoref  lref=2 cref=lime;
       run;
```



Answer: The H and V have been dropped in the **AUTOREF** options because there can be on **one** reference line.

As before, to control the axis code an AXIS statement and specify which axis to use it for in the chart options. **MAXIS stands for midpoint axis** whether it be on the horizontal for VBAR or on the vertical for HBAR. The **SHAPE** option lets you define the shape of the bar. As always, check the Reference Manual for a complete list of possible values. **CFRAME** lets you control the color of the frame or background behind the bars.

```
axis1
   label=(c=darkgreen
       h=1.5 f=zapfbi)
   value=(f=swissb
       c=darkgreen  h=0.75);
proc gchart data=sashelp.shoes;
vbar3d Product /
       caxis=blue
       ctext=darkblue
       autoref  lref=2 cref=lime
       maxis=axis1
       shape=star
       cframe=cyan;
       run;
```



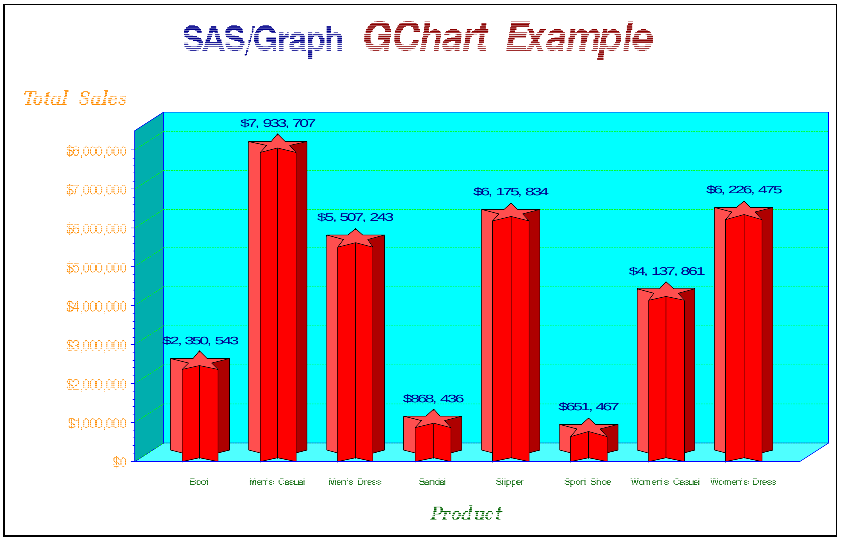
So what do you think the chart option for the **other axis** is called?   Answer: **RAXIS** which stand for response axis.

Thus far we have taken the default for the response axis which is Frequency – a count of observations that fall within the bars midpoint. For our data, that does not seem very useful. What ***might*** you want the height of the bar to represent?

Well, you have several possibilities – FREQ (the default), CFREQ (cumulative frequency), PERCENT, CPERCENT (cumulative percent), SUM, and MEAN. Simply code one of these as a chart option. If you use the SUM or MEAN option, you will also need to code the SUMVAR option to specify which variables value to take the SUM or MEAN of.

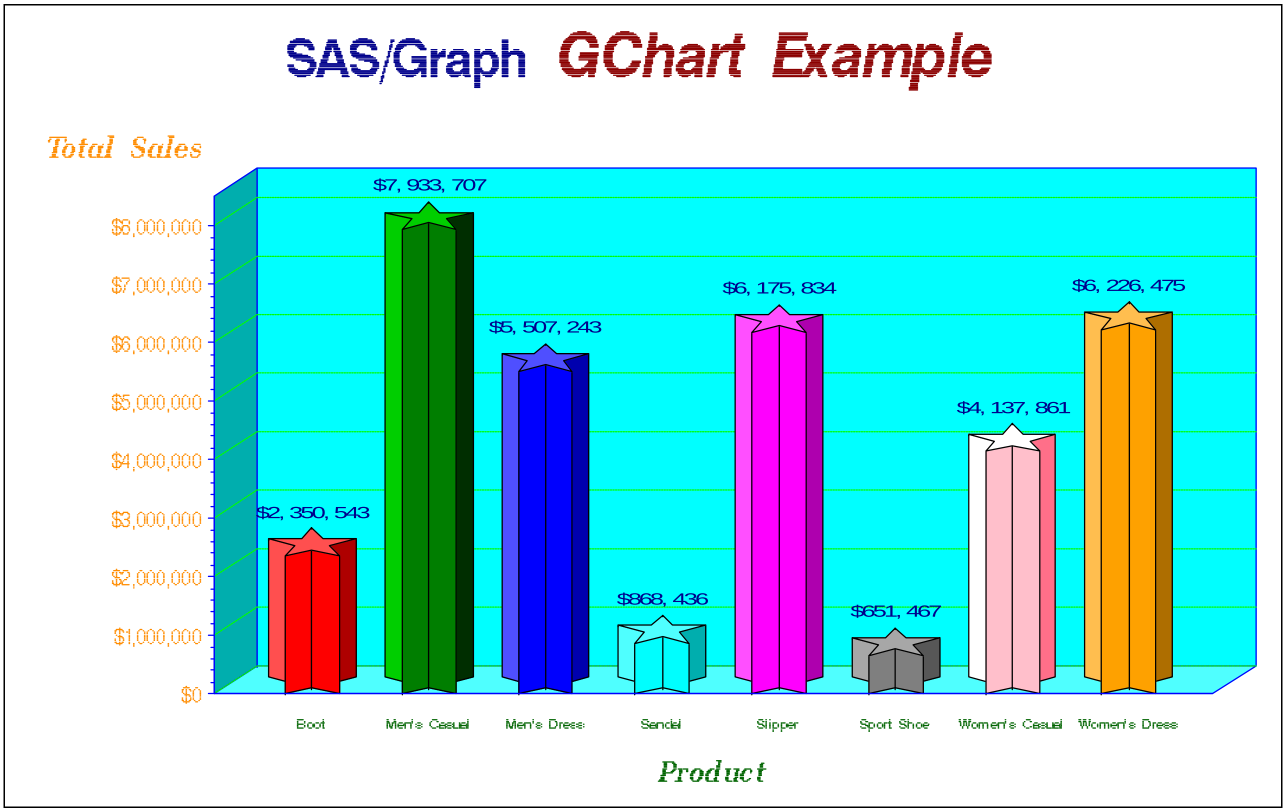
```
axis1  label=(c=darkgreen
             h=1.5 f=zapfbi)
      value=(f=swissb
             c=darkgreen h=0.70);
axis2  label=(c=darkorange
             h=1.5 f=zapfbi)
      value=(c=darkorange
             f=swissl)    ;
```

```
proc gchart data=sashelp.shoes;
vbar3d Product /
       caxis=blue   ctext=darkblue
       autoref  lref=2 cref=lime
       sum   sumvar=sales
       maxis=axis1  raxis=axis2
       shape=star   cframe=cyan;
       run;
```



One graph option you may find useful (I do) is the **patternid**. If you do not specify the patternid option, all the bars are the same pattern -- color (red) and how it is filled (solid). (We will see how the patterns can be controlled in a little bit.) So by coding the patternid=midpoint, we are telling SAS to change the pattern every time the midpoint changes.

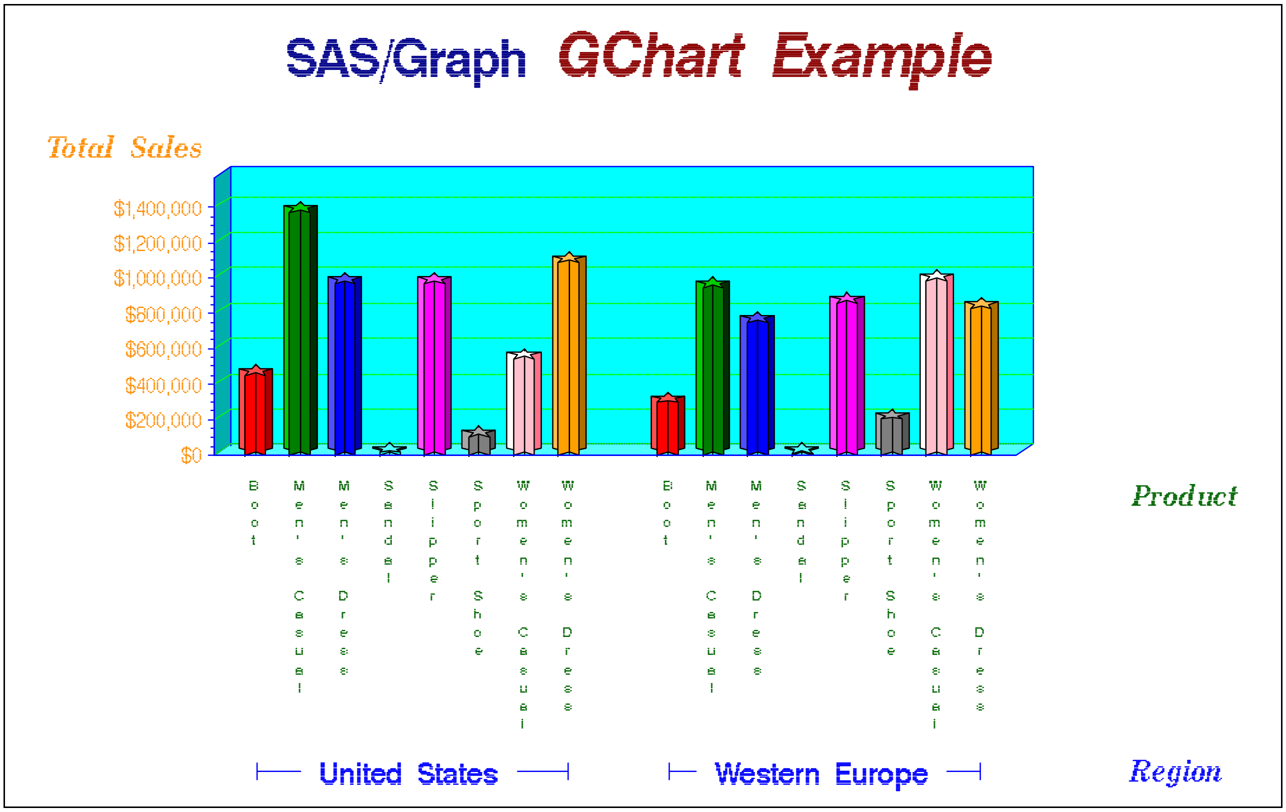
```
vbar3d Product /
       caxis=blue   ctext=darkblue
       autoref  lref=2 cref=lime
       sum   sumvar=sales
       maxis=axis1  raxis=axis2
       shape=star   cframe=cyan
       patternid=midpoint;
       run;
```



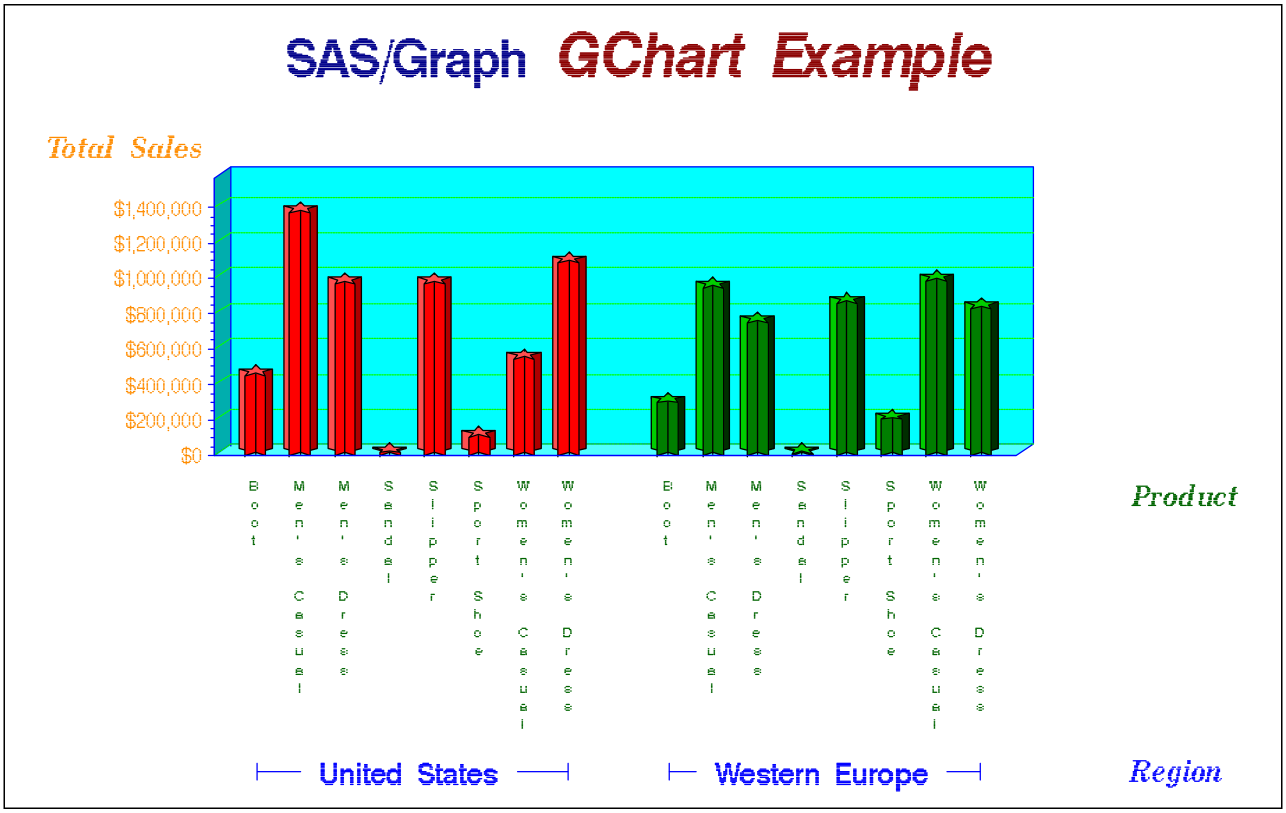
One of the powerful features of GCHART is its ability to do **grouped bar charts** by simply coding the **group=**variable and **gaxis=** options. Used the where and run-group processing so we could see just two of the regions.

```
axis3  label=(c=blue h=1.5 f=centbi)
       value=(f=swissb c=blue  h=1.5);

where Region in("United States","Western Europe");
vbar3d Product /
       shape=star   caxis=blue   ctext=darkblue
       maxis=axis1  raxis=axis2
       autoref  lref=2 cref=lime
       cframe=cyan
       sum   sumvar=sales
       patternid=midpoint
       group=Region  gaxis=axis3  ;
       run;
```

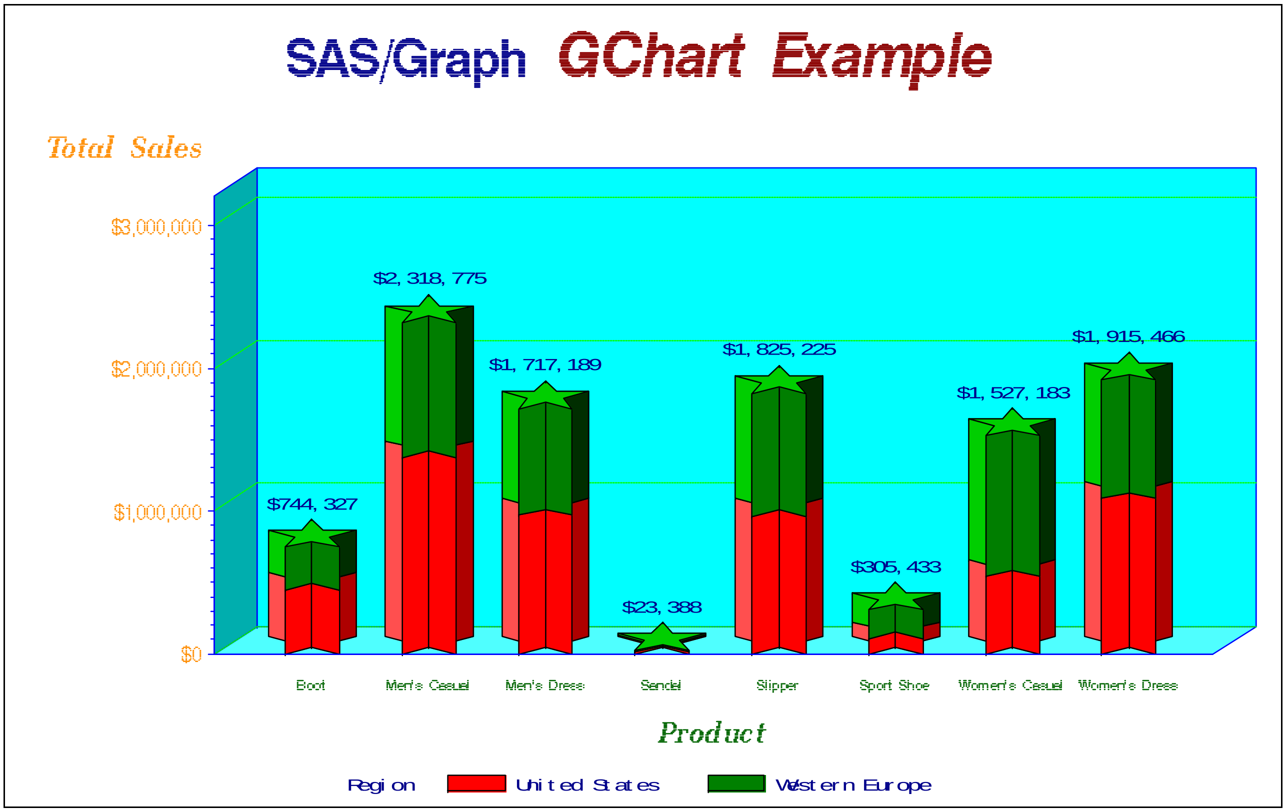


Changed the patternid option to make the pattern in the bars change when the group changes **(patternid=group)** and get a different look altogether.



But what I feel is one of the ***really*** powerful features is the ***subgroup bar chart***. Simply code the **subgroup**= option to define which variables value to subgroup by. The patternid option defaults to **patternid=subgroup** because any of the other options do not make much sense when used with a subgroup.

```
where Region in("United States","Western Europe");
vbar3d Product /
       shape=star    caxis=blue    ctext=darkblue
       maxis=axis1   raxis=axis2
       autoref  lref=2 cref=lime
       cframe=cyan
       sum    sumvar=sales
       subgroup=Region     ;
       run;
```

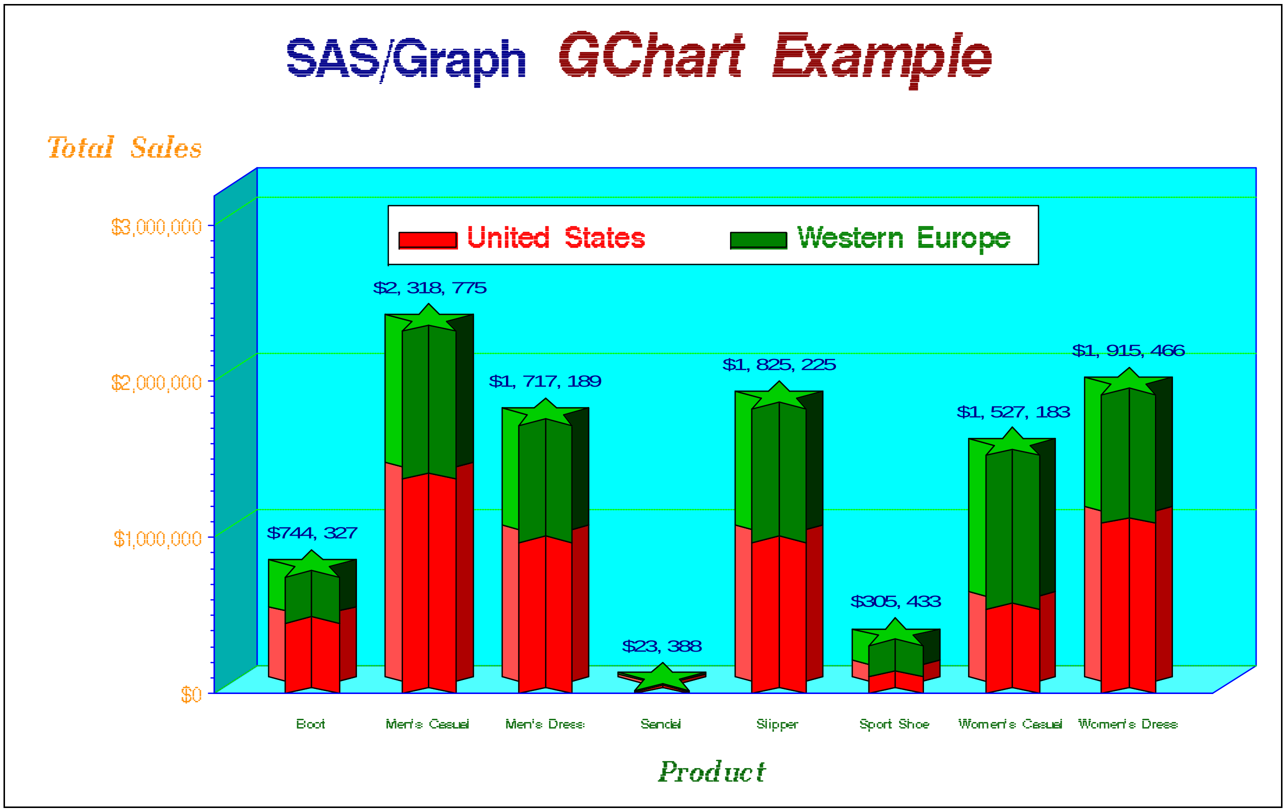


By default SAS will generate a legend of the subgroups (subdivided bars). The legend can also be controlled in a very similar way as the axis are – a **LEGEND** statement (which will look a lot like an axis statement). Then simply code the graph option (**legend=**) to point it to the appropriate legend definition.

```
legend1  frame
       position=(top center inside)     label=none
       value=(f=swissb  h=1.5
                    t=1  c=red
                    t=2  c=green);
```

Take notice how the above code makes the region name in the legend the same as the fill color.

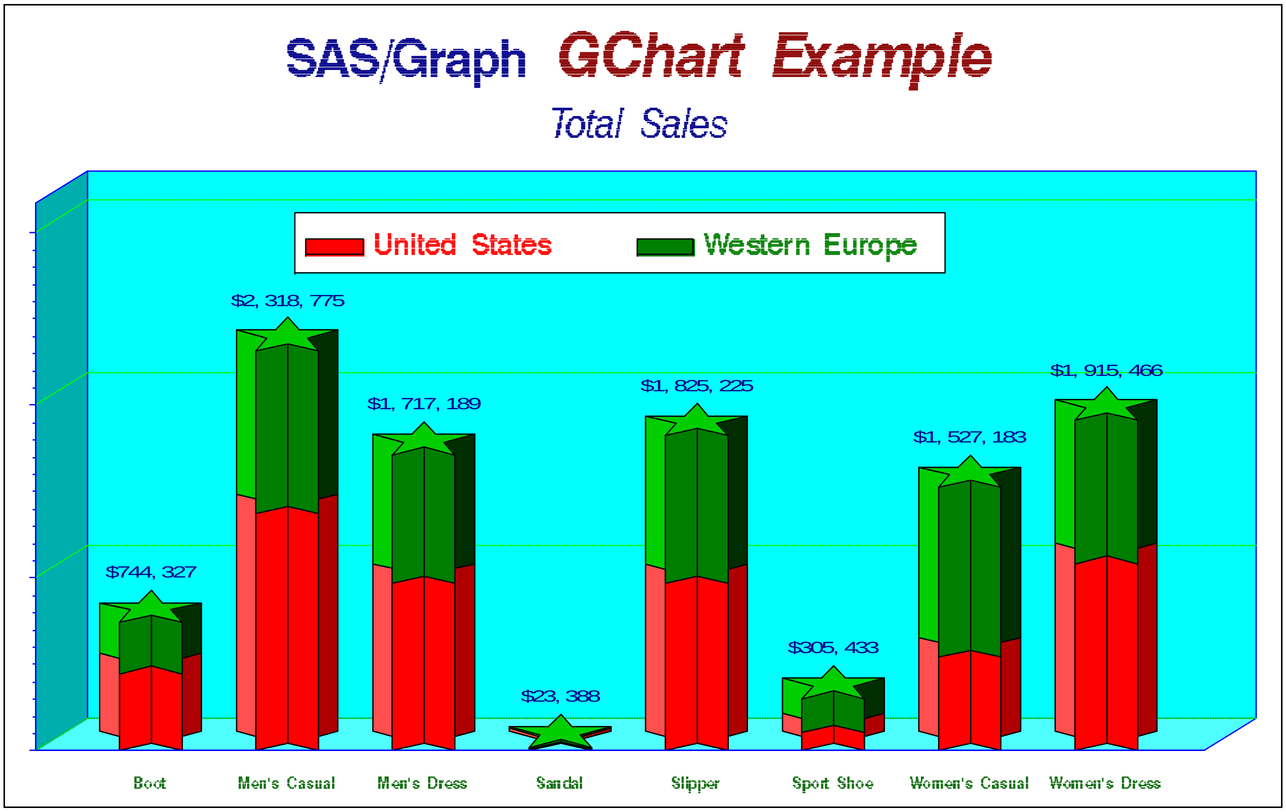
```
where Region in("United States","Western Europe");
vbar3d Product /
       shape=star    caxis=blue    ctext=darkblue
       maxis=axis1   raxis=axis2
       autoref  lref=2 cref=lime
       cframe=cyan   sum    sumvar=sales
       subgroup=Region     legend=legend1  ;
```



I had to throw this example in that is slight variation of the one above just to show you how you can label a chart and not need the axis.

```
title1 c=darkblue h=2.5
       f=swissb  "SAS/Graph "
       c=darkred h=3.0
       f=swissbi "GChart Example";
title2 c=darkblue h=2.0
       f=swissi  "Total Sales";
axis1
       label=none
       value=(f=swissb c=darkgreen h=0.8);
axis2
       label=none
       value=none   ;
legend1  frame
       position=(top center inside)
       label=none
       value=(f=swissb  h=1.5
                     t=1  c=red
                     t=2  c=green);
```

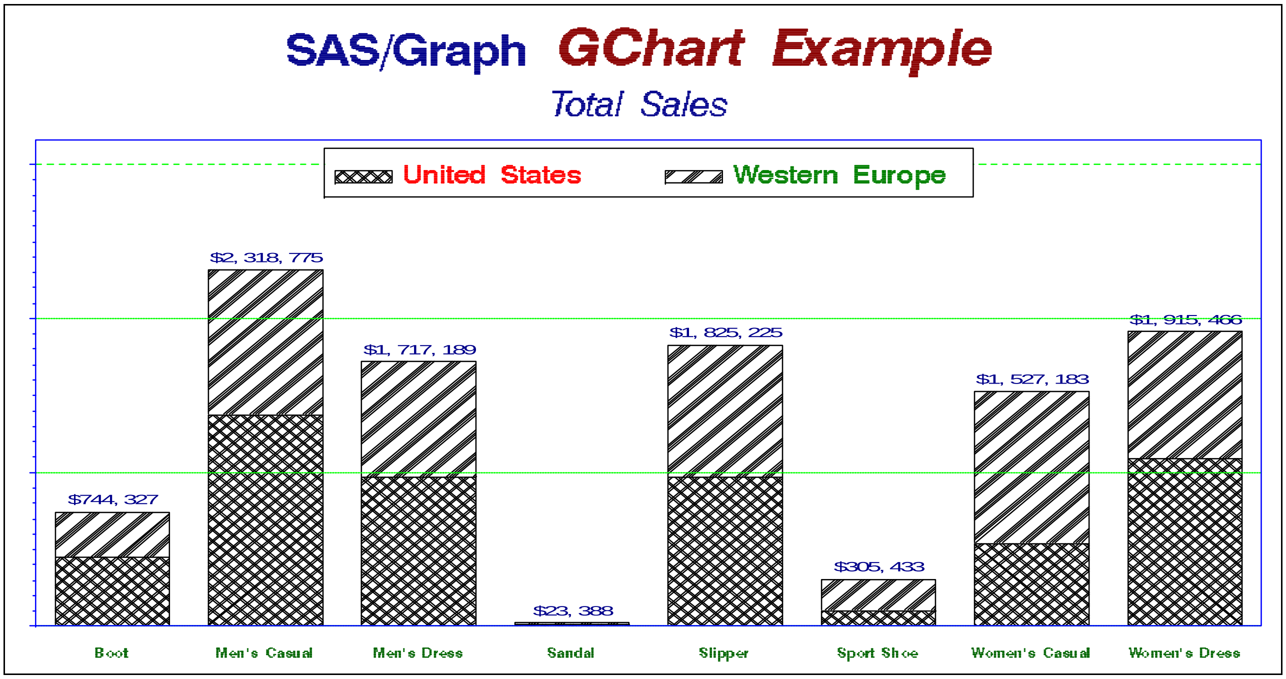
```
proc gchart data=sashelp.shoes;
       where Region in("United
States","Western Europe");
       vbar3d Product /
              shape=star
              caxis=blue
              ctext=darkblue
              maxis=axis1
              raxis=axis2
              autoref  lref=2 cref=lime
              cframe=cyan
              sum   sumvar=sales
              subgroup=Region
              patternid=subgroup
              legend=legend1;
       run;
```



The graphs that have been generated thus far can be copied and pasted into emails, word documents, etc. with not problems. BUT, if they are to put into a doc and then reproduced in black and white, the solid filled bars in different colors will just come out as one color—gray.

One thing you can do to avoid this problem is change the patterns used to something other than solid colors –with the PATTERN statement. (Looks a lot like a symbol statement.)

```
pattern1 c=black  v=x3 ; *** Crosshatch pattern, density of 3;
pattern2 c=black  v=r5 ; *** Right slanted lines, density of 5;
```



GOPTIONS STATEMENT

Thus far we have just let the graph we are requesting to be displayed back to us in a Window (obviously the default—duh). But that is OK for now because it is a really good way to learn and eventually use as a development tool to get the graph just the way you want.

In reality what is happening is, by default, SAS is using the device driver of WIN, which directs the graphics output back to a window. Just like you need print device drivers on your Microsoft Windows platforms in order to print to attached or network printers, SAS needs a graphics device driver to know how and where to produce the graph. Device drivers basically take whatever you give it and convert it into something the output device can reproduce, either onto the screen or some kind of output (printer) device.

Just like you can tell your Microsoft Office product which of the defined printers on your platform you wish to print to, you can use the **GOPTIONS** statement in SAS. There are WAY too many options to cover here, but we will mention a few important one.

Several of these options have a default value just like a regular SAS system option regardless of what device you specify, which of course you can override.

Border|Noborder--put a border around the entire graph.
Cback= -- Color of background of entire graph.
Cby= -- Color of BY line.
Ctext= -- Color for all text in the body of the graph.
Ctitle= -- Color for all titles.
Device= -- Device driver.
Fby= -- Font for BY lines.
Ftext= -- Font for all text in the body of the graph.
Ftitle= -- Font for all titles.

Gsfname= -- Fileref of output file.
Gsfmode=append|replace -- disposition of output file.
Hby= -- Height of BY line
Htext= -- Height of all text in the body of the graph.
Htitle= -- Height of all titles.
Reset=(all|global|axis|symbol|pattern|legend|title)
-- Reset back to system default.
Targetdevice= -- Device driver that will eventually be used to produce final graph.

Other options values will be set based upon which device you specify, which you can also override.

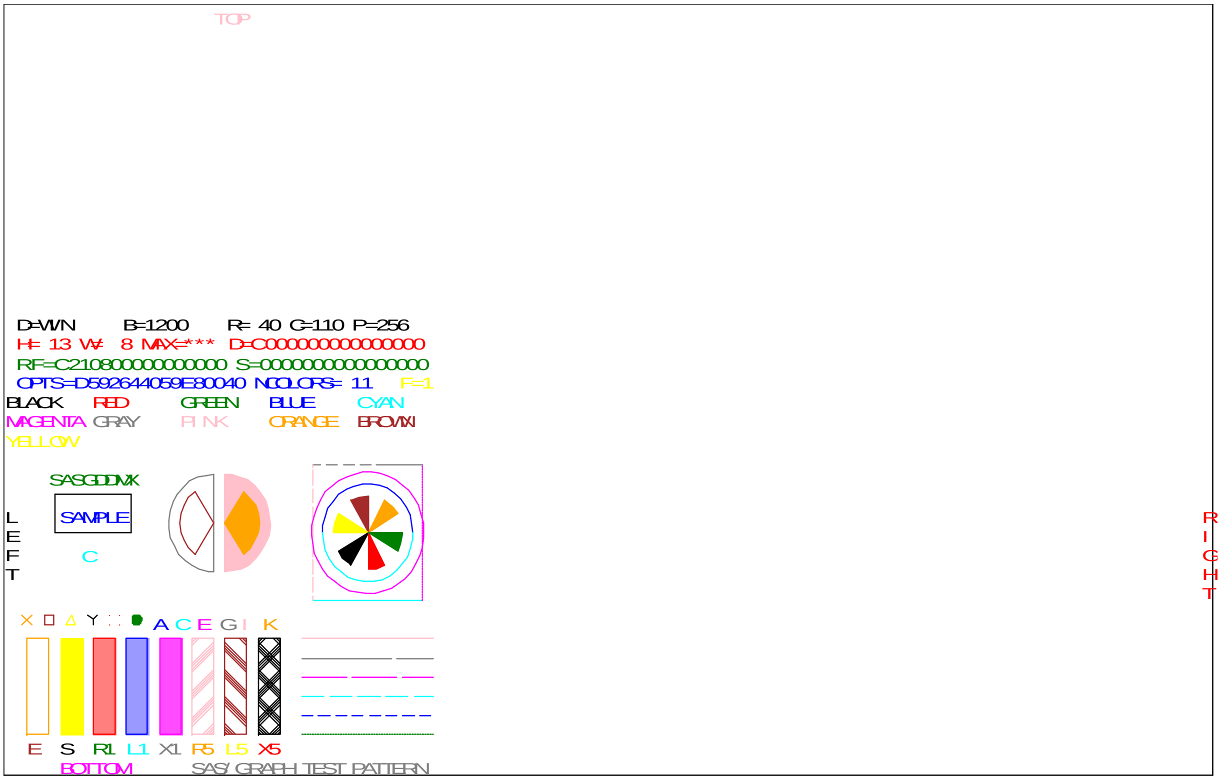
Colors=() -- Colors list.
Hsize= -- Physical size of graph in horizontal inches.
Hpos= -- Number of cells in the horizontal.
Xpixels= -- Physical size of graph in horizontal pixels.

Vsize= -- Physical size of graph in vertical inches.
Vpos= -- Number of cells in the vertical.
Ypixels= -- Physical size of graph in vertical pixels.

There are ***dozens*** (probably hundreds) of device drivers defined in the SAS device catalog. You can display, print, and even modify the device definitions using the **GDEVICE** procedure. See the Reference Manual for more info. But I use the **GTESTIT** procedure to get some of the basic information about the driver. The info is generated in the SASLOG and produces 3 pictures. I usually only want to see the first one (pic=1).

```
Proc gtestit pic=1;run;
```

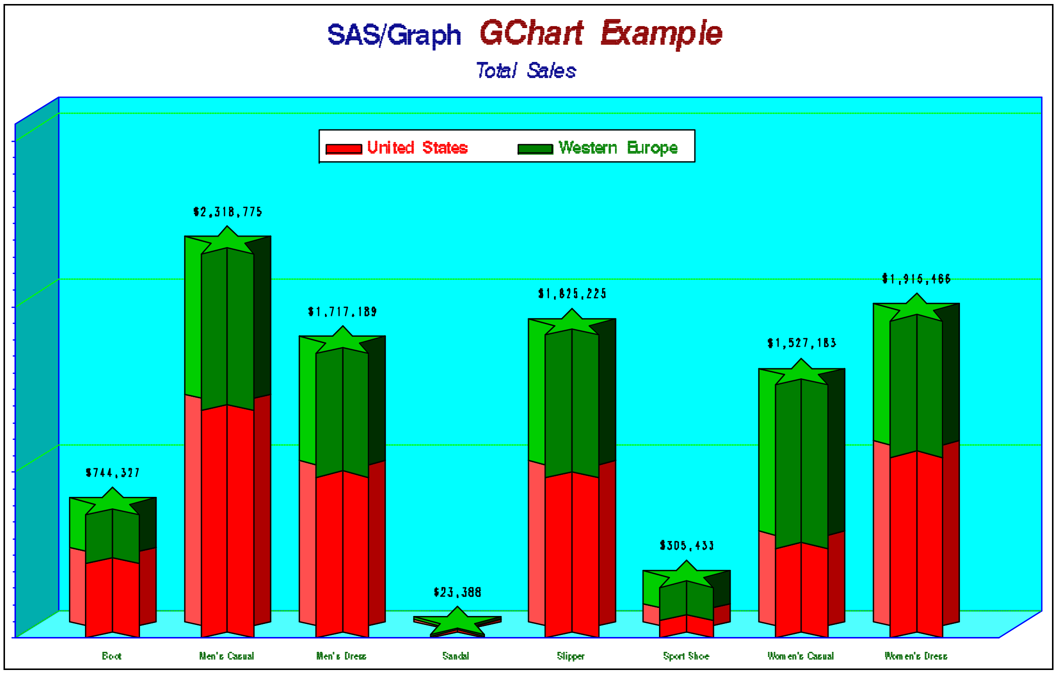
```
SASLOG:
D=WIN      B=1200    R= 40 C=110 P=256
H= 13 W=  8 MAX=*** D=C000000000000000
RF=C210800000000000 S=0000000000000000
OPTS=D592644059E80040 NCOLORS= 11
Background color = WHITE
Color  1 = BLACK
Color  2 = RED
Color  3 = GREEN
Color  4 = BLUE
Color  5 = CYAN
Color  6 = MAGENTA
Color  7 = GRAY
Color  8 = PINK
Color  9 = ORANGE
Color 10 = BROWN
Color 11 = YELLOW
Ratio = 0.59091
Hsize = 9.16667
Vsize = 5.41667
F=1
```


TOP
D=WIN B=1200 R= 40 C=110 P=256
H= 13 W= 8 MAX=*** D=C000000000000000
RF=C210800000000000 S=0000000000000000
OPTS=D592644059E80040 NCOLORS= 11 F=1
BLACK RED GREEN BLUE CYAN
MAGENTA GRAY PINK ORANGE BROWN
YELLOW
SASGDDMX
SAMPLE
C
LEFT
RIGHT
A C E G I K
E S R1 L1 X1 R5 L5 X5
BOTTOM
SAS/GRAPH TEST PATTERN

The most useful info off this is the R= and C= on the first real line of info and the **COLORS** list option a few lines below. The R= (rows) gives us the **VPOS** option value, and the C= (columns) gives us the **HPOS** option value. Basically, the **HPOS/VPOS** values define how many rows & columns the graph has which will define the size of the characters (text) on the graph. If you increase the **HPOS/VPOS** values, the size of text will decrease; if the values decrease, the size of the text will increase.

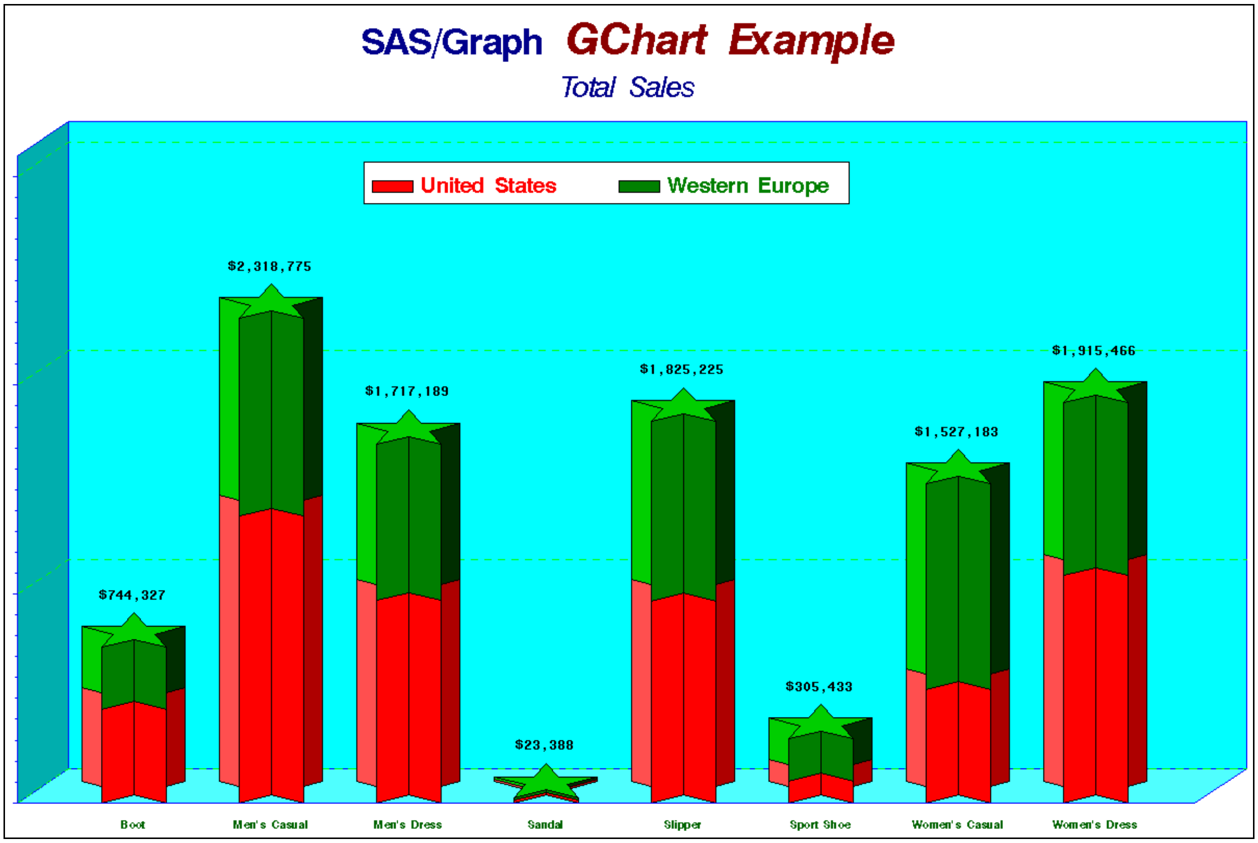
You can control the actual physical size of the picture by using the **HSIZE/VSIZE** or **XPIXELS/YPIXELS** values. **HSIZE/VSIZE** is normally specified in inches. Normally I just have to use trail-and-error to determine the size I need sometimes. Notice what happens with the GCHART from earlier with a larger **HPOS/VPOS**

```
goptions device=win        hpos=150      vpos=60 ;
```



A device like GIF requires an output file specified (with the GSFNAME option).

```
filename graphs 'C:\Documents and Settings\debruns\My Documents\SUGI30';
goptions device=gif
             hpos=150      vpos=60
             xpixels=1200   ypixels=800
             gsfmode=replace
             gsfname=graphs;
proc gchart data=sashelp.shoes;
       where Region in("United States","Western Europe");
       vbar3d Product /
             shape=star    caxis=blue   ctext=black
             maxis=axis1   raxis=axis2
             autoref  lref=2 cref=lime
             cframe=cyan   sum   sumvar=sales
             subgroup=Region     patternid=subgroup
             legend=legend1
             name='gchart4'      ;
       run;
```



I could have specified a full path and filename in the FILENAME statement. But, by just specifying a directory I can use the same fileref and put as many files as I want into the directory by just adding a NAME= option to any graph request statement (ie VBAR, HBAR3D, PLOT, etc) to specify the name of the file. Also note SAS adds the appropriate extension to the filename.

Note that the picture does NOT get displayed back to you in a window. Several drivers are file output only. BUT, one way you can get a feel for how the picture will look using a different driver is the TARGETDEVICE= option. What SAS does is merge (as best it can) the two drivers attributes to simulate what it might look like.

```
goptions device=win        targetdevice=gif
             hpos=150      vpos=60  ;
```

IN SUMMARY

The one thing SAS has done over the years is virtually guarantee code written in prior versions will continue to work in later ones … upward compatibility. Even though some of the new features that have been introduced in new versions over the years make some old techniques obsolete; it is still good to know them.

Every use of every procedure is unique in some ways. All I have attempted to do is give you a good starting point or foundation to better understand how to get SAS/GRAPH to give you what you want.

So good luck!!!

ACKNOWLEDGEMENTS

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

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

AUTHOR

If you have any questions or comments, please write or call:

Dan Bruns
Tennessee Valley Authority
1101 Market Street (MP 2B)
Chattanooga, TN 37402
423/751-6430   Fax 423/751-3163
Email: debruns@tva.gov