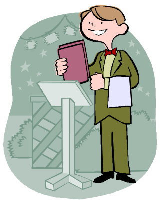
Introduction to Graphics Using SAS/GRAPH® Software

Mike Kalt
Education Division
SAS Institute

2010 SESUG
September 26–28
Savannah, GA

Today's Menu

- Section 1: Course Logistics and Overview
- Section 2: Producing Scatter and Line Plots
- Section 3: Producing Bar Charts
- Section 4: Customizing the Appearance of SAS/GRAPH Output
- Section 5: Generating Graphs for Use in Other Applications
- Section 6: Conclusion



Section 1: Course Logistics and Overview

- **1.1 Some Logistical Issues**
- 1.2 Types of Output Produced by SAS/GRAPH
- 1.3 Overview of "Classic" SAS/GRAPH
- 1.4 SAS/GRAPH and the Output Delivery System (ODS)

What This Presentation Is (and Isn't)

This is an ***introduction*** to creating basic graphics with SAS/GRAPH.

- The **SAS/GRAPH I: Essentials** introductory course from SAS Education runs 3 full days.
- This seminar runs 50 minutes. Do the math ☺
- You will learn:
  - How to create basic plots and charts
  - How to modify (somewhat) the appearance of graphs
  - How to export graphs to other applications

Enterprise Guide vs. SAS/GRAPH

Enterprise Guide generates SAS/GRAPH code "behind the scenes".

- Enterprise Guide:
  - is easier to use
  - lets you design graphs using a point and click interface, and save or modify the generated code.
- SAS/GRAPH:
  - provides more powerful customization features
  - can produce some types of graphs that Enterprise Guide can't.

This seminar will show you how to write SAS/GRAPH programs. You can use what you learn here to write your own programs or to modify code generated by Enterprise Guide.

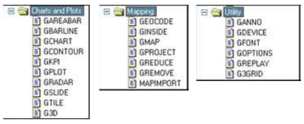
Section 1: Course Logistics and Overview

- 1.1 Some Logistical Issues
- **1.2 Types of Output Produced by SAS/GRAPH**
- 1.3 Overview of "Classic" SAS/GRAPH
- 1.4 SAS/GRAPH and the Output Delivery System (ODS)

Two Types of SAS/GRAPH Output

SAS/GRAPH programs produce graphics using two very distinct systems. The two systems are:

- Original or "Classic" SAS/GRAPH ("G" procedures)
- Statistical Graphics ("SG" procedures).

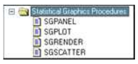


Charts and Plots
- GAREABAR
- GBARLINE
- GCHART
- GCONTOUR
- GKPI
- GPLOT
- GRADAR
- GSLIDE
- GTILE
- G3D

Mapping
- GEOCODE
- GINSIDE
- GMAP
- GPROJECT
- GREDUCE
- GREMOVE
- MAPIMPORT

Utility
- GANNO
- GDEVICE
- GFONT
- GOPTIONS
- GREPLAY
- G3GRID

Original or "Classic" SAS/GRAPH



Statistical Graphics Procedures
- SGPANEL
- SGPLOT
- SGRENDER
- SGSCATTER

+
Selected SAS
Statistical Procedures

Statistical Graphics

This seminar primarily focuses on the original SAS/GRAPH procedures GCHART and GPLOT.

Section 1: Course Logistics and Overview

1.1 Some Logistical Issues

1.2 Types of Output Produced by SAS/GRAPH

1.3 Overview of "Classic" SAS/GRAPH

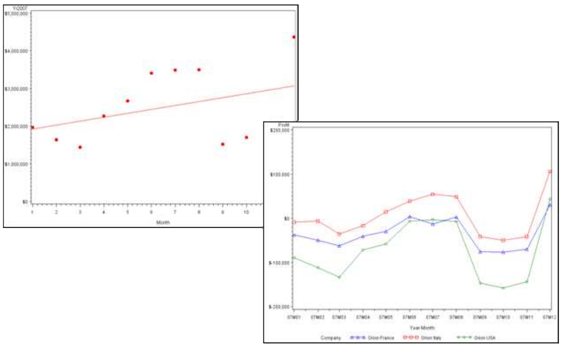
1.4 SAS/GRAPH and the Output Delivery System (ODS)

"Classic" SAS/GRAPH Procedures

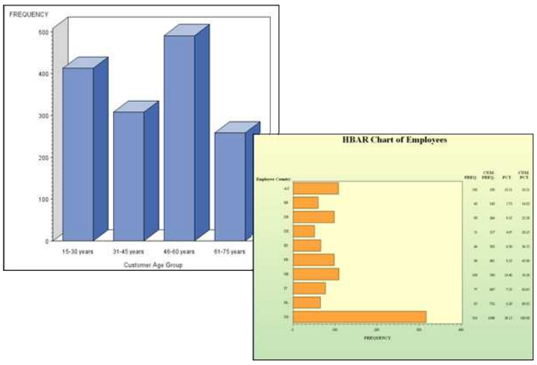
"Classic" SAS/GRAPH procedures can be used to create the following types of graphs:

- two-dimensional scatter plots and line plots, including many types of statistical plots (GPLOT)
- bar, block, pie, and star charts (GCHART)
- three-dimensional scatter and surface plots (G3D)
- contour plots (GCONTOUR)
- maps (GMAP)
- text slides (GSLIDE)
- custom graphs (Annotate Facility)

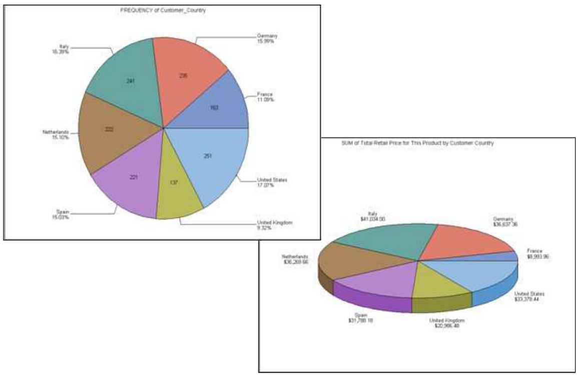
PROC GPLOT: Scatter and Line Plots



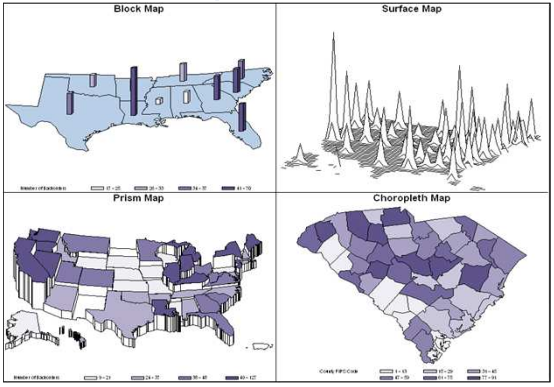
PROC GCHART: Bar Charts



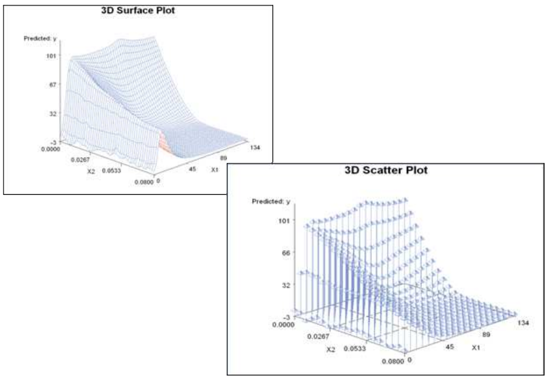
PROC GCHART: Pie Charts



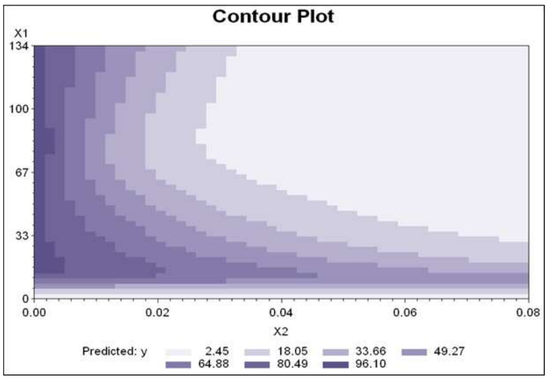
PROC GMAP: Maps



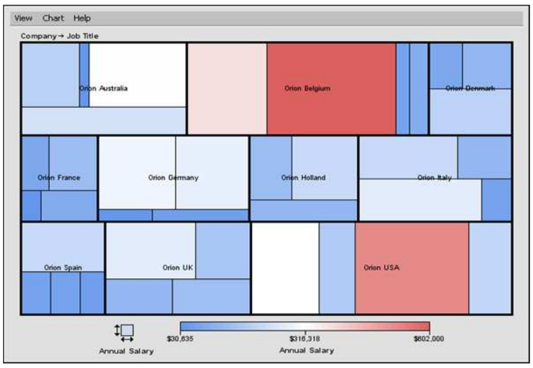
PROC G3D: 3-D Surface Plots/Scatter Plots



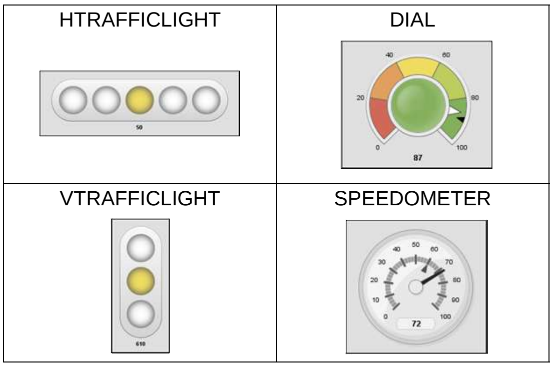
PROC GCONTOUR: Contour Plots



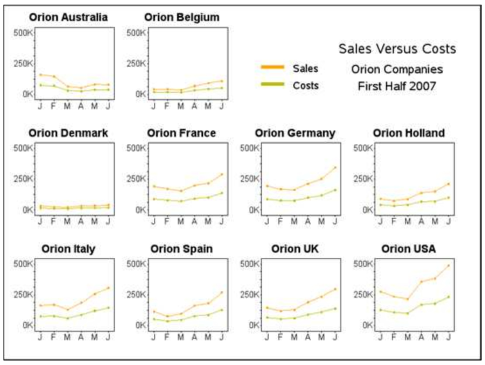
PROC GTILE: Tile Charts



PROC GKPI: KPI Charts for Dashboards



PROC GREPLAY: Multiple Graphs Per Page



ANNOTATE Facility: Add Annotation to Graphs

Let it snow! Let it snow! Let it snow!
Click snowflakes to see historical-snowfall data!

SAS/GRAPH Program Structure and Syntax

SAS/GRAPH procedure code uses the general syntax framework of other SAS procedures.

GOPTIONS *graphics-specific-options*;
<SAS/GRAPH global statements>
PROC *graphics-procedure-name* DATA=*libref.data set*;
    other procedure statements / *<statement options>*;
RUN;

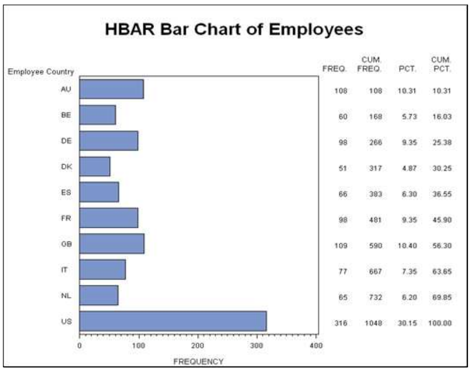
Sample SAS/GRAPH Program

The following program produces a horizontal (HBAR) chart showing a frequency count of the values of the variable `employee_country`:

```
title 'HBAR Bar Chart of Employees';
proc gchart data=employees;
   hbar employee_country;
run;
```

Most SAS/GRAPH procedures have a statement that invokes the procedure (PROC GCHART), and another "action" statement that requests a specific type of graph (HBAR).

PROC CGHART: Example



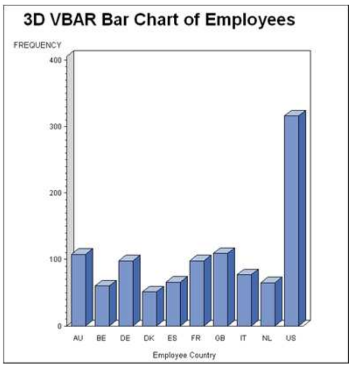
```
         CUM.           CUM.
FREQ.   FREQ.   PCT.    PCT.
  108     108   10.31   10.31
   60     168    5.73   16.03
   98     266    9.35   25.38
   51     317    4.87   30.25
   66     383    6.30   36.55
   98     481    9.35   45.90
  109     590   10.40   56.30
   77     667    7.35   63.65
   65     732    6.20   69.85
  316    1048   30.15  100.00
```

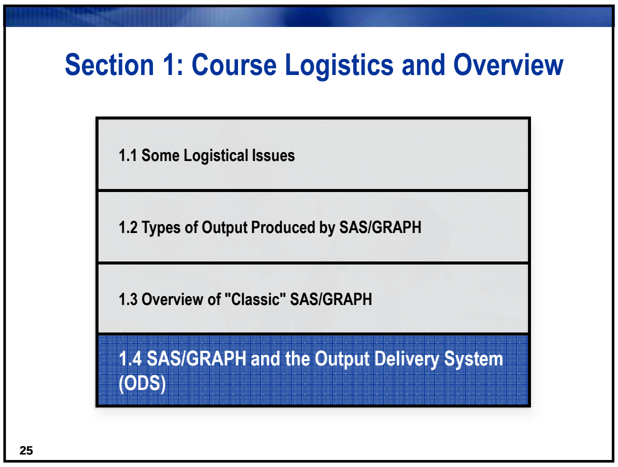
Modified SAS/GRAPH Program

Modify the previous program to produce a three-dimensional vertical bar chart:

```
title '3D VBAR Bar Chart of Employees';
proc gchart data=employees;
   vbar3d employee_country;
run;
```

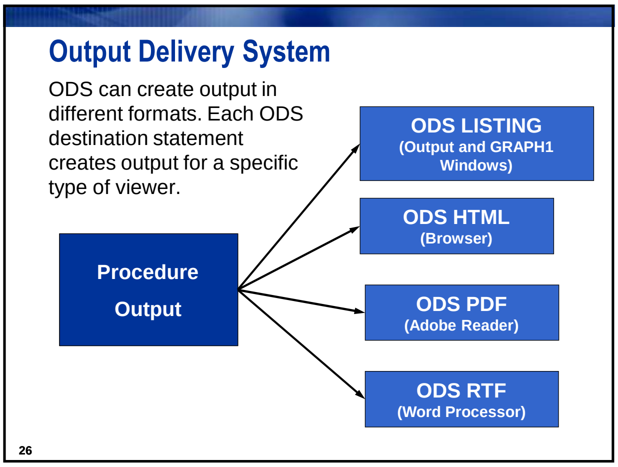
SAS/GRAPH Program Output





Section 1: Course Logistics and Overview

- 1.1 Some Logistical Issues
- 1.2 Types of Output Produced by SAS/GRAPH
- 1.3 Overview of "Classic" SAS/GRAPH
- **1.4 SAS/GRAPH and the Output Delivery System (ODS)**

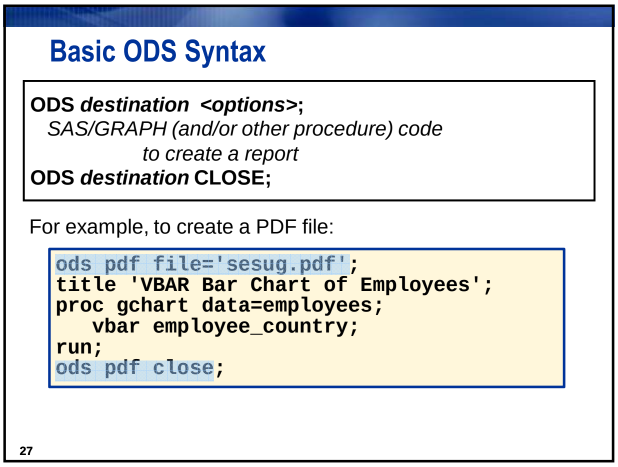


Output Delivery System

ODS can create output in different formats. Each ODS destination statement creates output for a specific type of viewer.

Procedure Output

- **ODS LISTING** (Output and GRAPH1 Windows)
- **ODS HTML** (Browser)
- **ODS PDF** (Adobe Reader)
- **ODS RTF** (Word Processor)

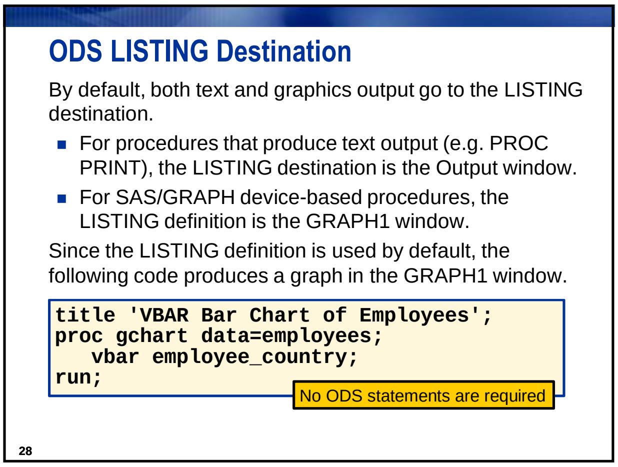


Basic ODS Syntax

ODS *destination* *<options>*;
SAS/GRAPH (and/or other procedure) code
to create a report
ODS *destination* CLOSE;

For example, to create a PDF file:

```
ods pdf file='sesug.pdf';
title 'VBAR Bar Chart of Employees';
proc gchart data=employees;
   vbar employee_country;
run;
ods pdf close;
```



ODS LISTING Destination

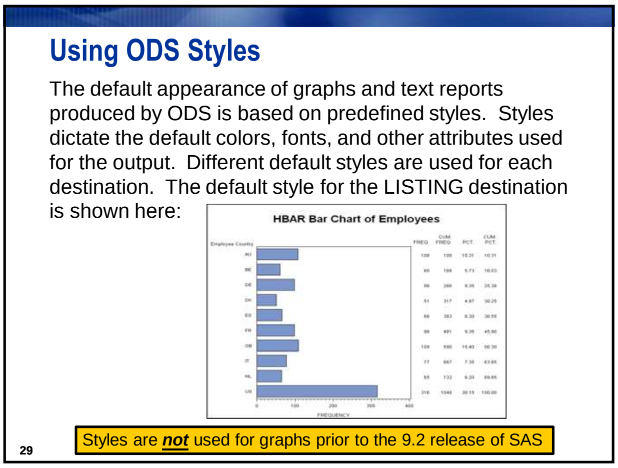
By default, both text and graphics output go to the LISTING destination.

- For procedures that produce text output (e.g. PROC PRINT), the LISTING destination is the Output window.
- For SAS/GRAPH device-based procedures, the LISTING definition is the GRAPH1 window.

Since the LISTING definition is used by default, the following code produces a graph in the GRAPH1 window.

```
title 'VBAR Bar Chart of Employees';
proc gchart data=employees;
   vbar employee_country;
run;
```

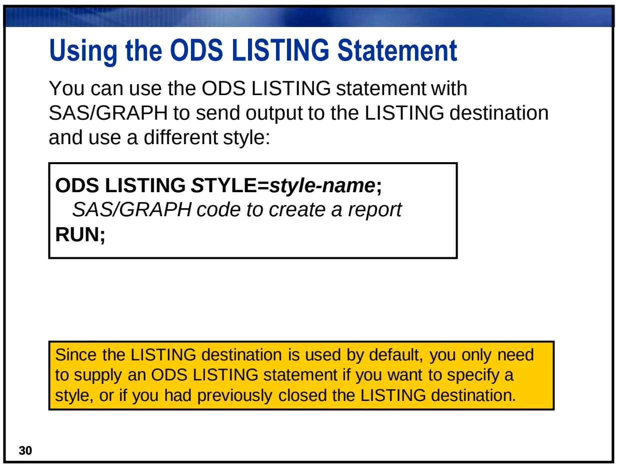
No ODS statements are required



Using ODS Styles

The default appearance of graphs and text reports produced by ODS is based on predefined styles. Styles dictate the default colors, fonts, and other attributes used for the output. Different default styles are used for each destination. The default style for the LISTING destination is shown here:

Styles are ***not*** used for graphs prior to the 9.2 release of SAS

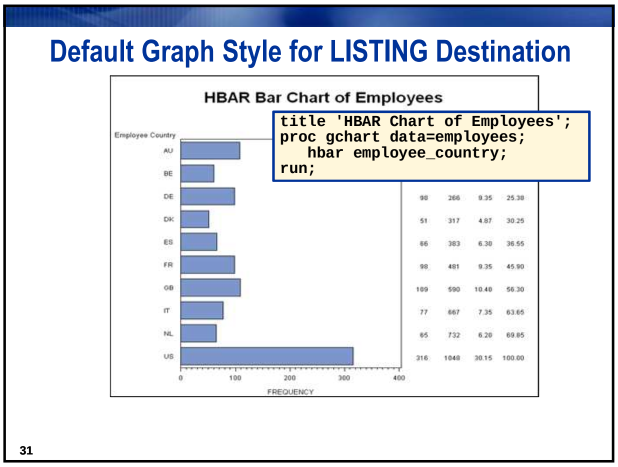


Using the ODS LISTING Statement

You can use the ODS LISTING statement with SAS/GRAPH to send output to the LISTING destination and use a different style:

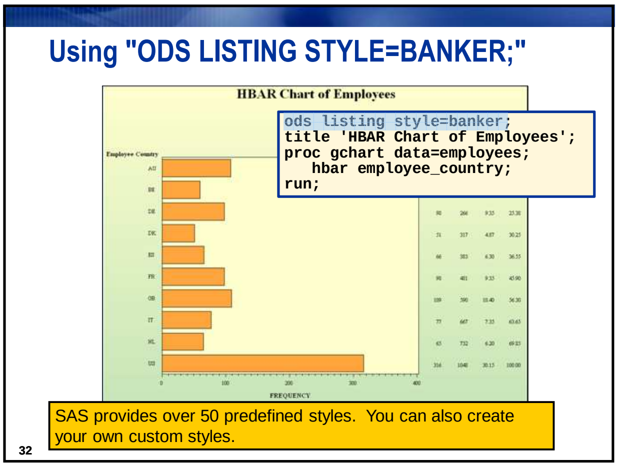
ODS LISTING STYLE=*style-name*;
SAS/GRAPH code to create a report
RUN;

Since the LISTING destination is used by default, you only need to supply an ODS LISTING statement if you want to specify a style, or if you had previously closed the LISTING destination.



Default Graph Style for LISTING Destination

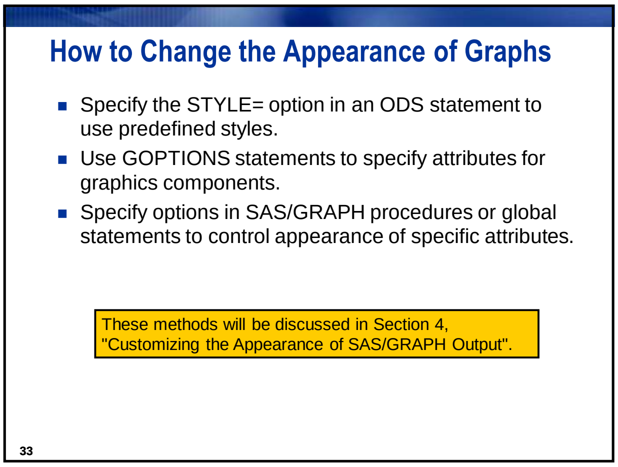
```
title 'HBAR Chart of Employees';
proc gchart data=employees;
   hbar employee_country;
run;
```



Using "ODS LISTING STYLE=BANKER;"

```
ods listing style=banker;
title 'HBAR Chart of Employees';
proc gchart data=employees;
   hbar employee_country;
run;
```

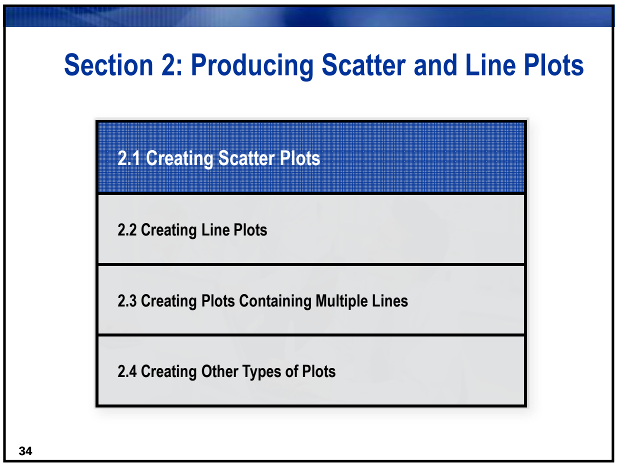
SAS provides over 50 predefined styles. You can also create your own custom styles.



How to Change the Appearance of Graphs

- Specify the STYLE= option in an ODS statement to use predefined styles.
- Use GOPTIONS statements to specify attributes for graphics components.
- Specify options in SAS/GRAPH procedures or global statements to control appearance of specific attributes.

These methods will be discussed in Section 4, "Customizing the Appearance of SAS/GRAPH Output".



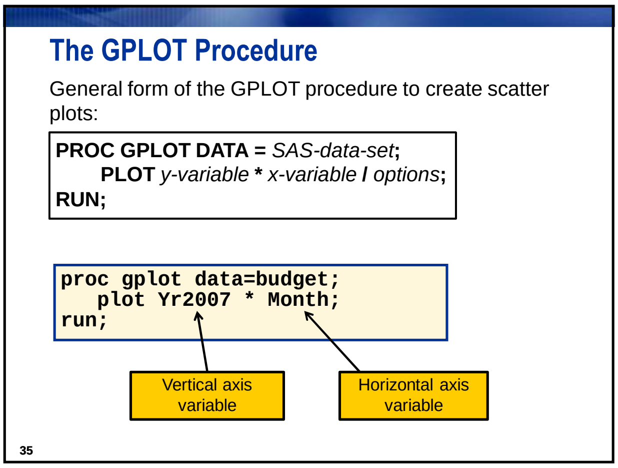
Section 2: Producing Scatter and Line Plots

2.1 Creating Scatter Plots

2.2 Creating Line Plots

2.3 Creating Plots Containing Multiple Lines

2.4 Creating Other Types of Plots



The GPLOT Procedure

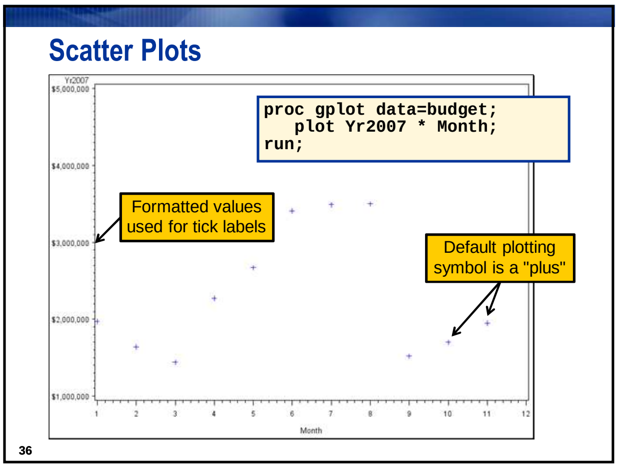
General form of the GPLOT procedure to create scatter plots:

PROC GPLOT DATA = *SAS-data-set***;**
      PLOT *y-variable* ***** *x-variable* **/** *options***;**
RUN;

```
proc gplot data=budget;
   plot Yr2007 * Month;
run;
```

Vertical axis variable

Horizontal axis variable

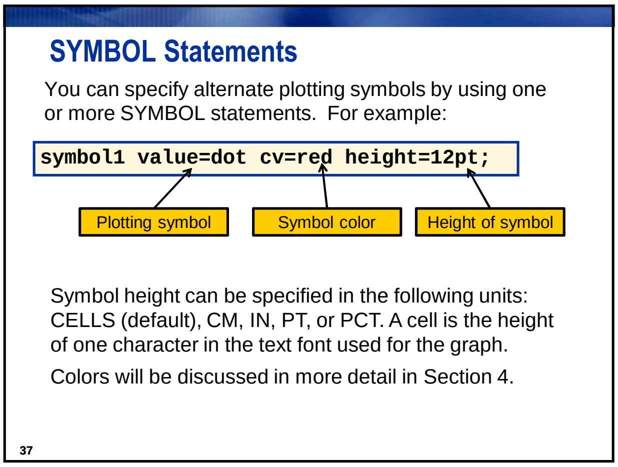


Scatter Plots

```
proc gplot data=budget;
   plot Yr2007 * Month;
run;
```

Formatted values used for tick labels

Default plotting symbol is a "plus"



SYMBOL Statements

You can specify alternate plotting symbols by using one or more SYMBOL statements. For example:

```
symbol1 value=dot cv=red height=12pt;
```

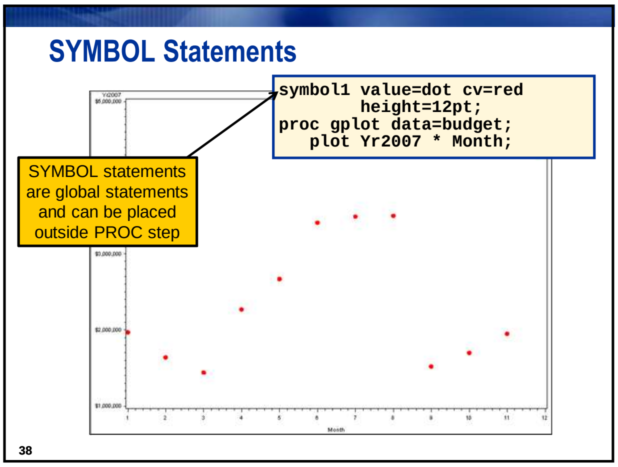
Plotting symbol

Symbol color

Height of symbol

Symbol height can be specified in the following units: CELLS (default), CM, IN, PT, or PCT. A cell is the height of one character in the text font used for the graph.

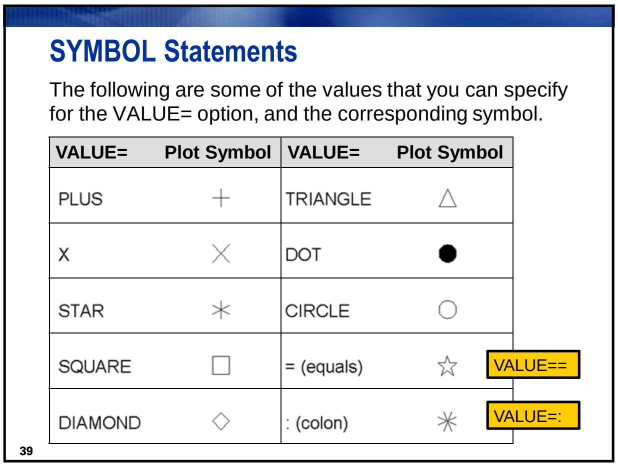
Colors will be discussed in more detail in Section 4.



SYMBOL Statements

```
symbol1 value=dot cv=red
        height=12pt;
proc gplot data=budget;
   plot Yr2007 * Month;
```

SYMBOL statements are global statements and can be placed outside PROC step



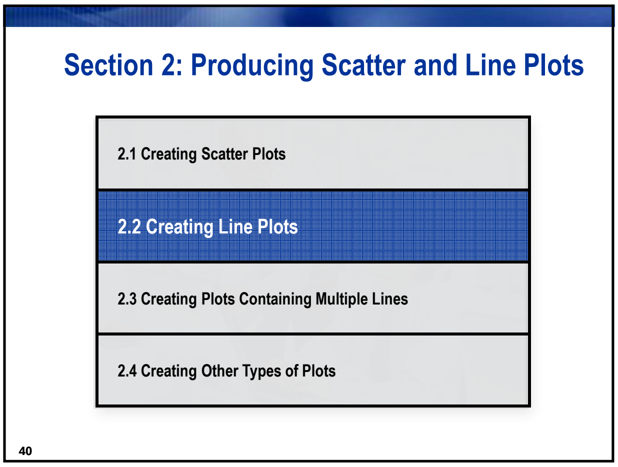
SYMBOL Statements

The following are some of the values that you can specify for the VALUE= option, and the corresponding symbol.

VALUE=	Plot Symbol	VALUE=	Plot Symbol
PLUS	+	TRIANGLE	△
X	×	DOT	●
STAR	✳	CIRCLE	○
SQUARE	□	= (equals)	☆
DIAMOND	◇	: (colon)	✳

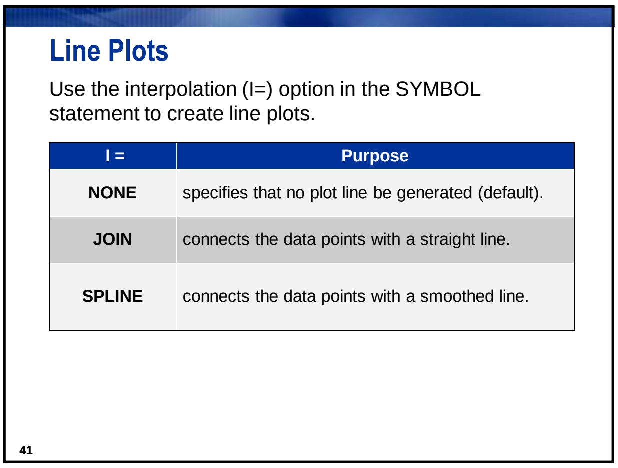
VALUE==

VALUE=:



Section 2: Producing Scatter and Line Plots

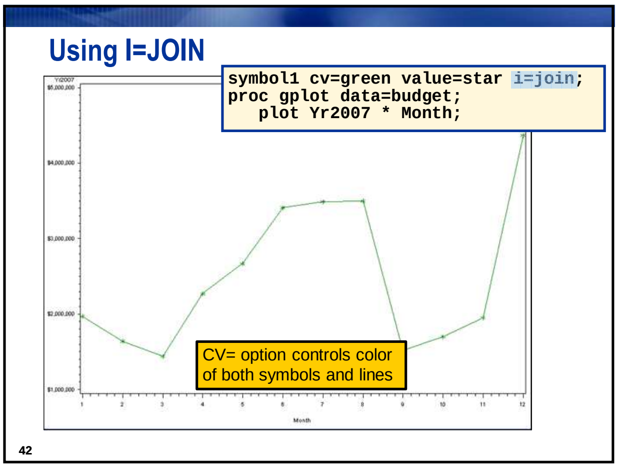
- 2.1 Creating Scatter Plots
- **2.2 Creating Line Plots**
- 2.3 Creating Plots Containing Multiple Lines
- 2.4 Creating Other Types of Plots



Line Plots

Use the interpolation (I=) option in the SYMBOL statement to create line plots.

I =	Purpose
NONE	specifies that no plot line be generated (default).
JOIN	connects the data points with a straight line.
SPLINE	connects the data points with a smoothed line.

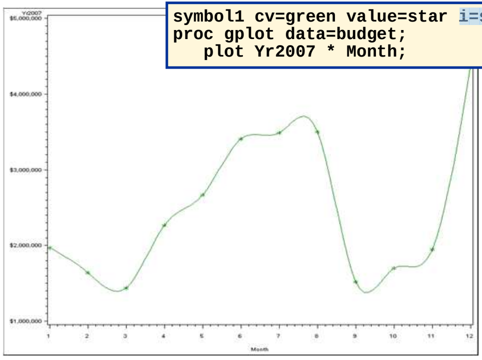


Using I=JOIN

```
symbol1 cv=green value=star i=join;
proc gplot data=budget;
   plot Yr2007 * Month;
```

CV= option controls color of both symbols and lines

Using I=SPLINE



```
symbol1 cv=green value=star i=spline;
proc gplot data=budget;
   plot Yr2007 * Month;
```

SYMBOL Statements

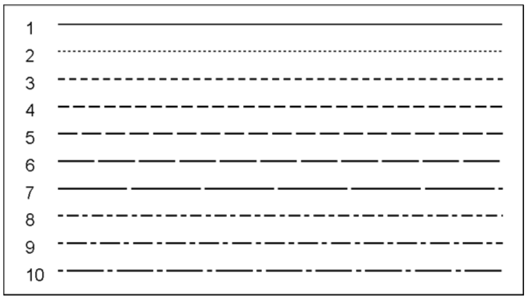
Options in the SYMBOL statement can control attributes of the plot line.

SYMBOL*n options***;**

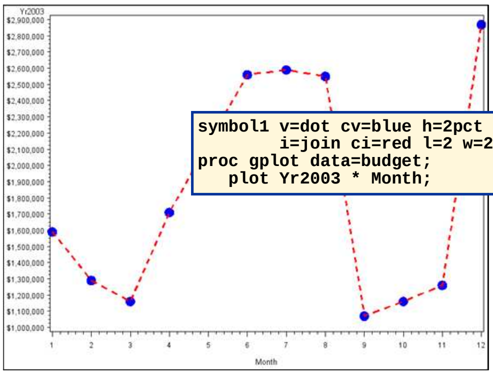
Option	Purpose
WIDTH = *n* **W =** *n*	specifies the plot line thickness.
LINE = *line-type* **L =** *line-type*	specifies a plot line type (solid, dashed, dotted) value ranging from 1 to 46.
CI = *line-color*	specifies the color of the plot line only.
COLOR = *color* **C =** *color*	specifies the color for both plot symbol and plot line.

SYMBOL Statements

The following is a sample of values that you can specify for the LINE= option and the corresponding line type.



The SYMBOL Statement



```
symbol1 v=dot cv=blue h=2pct
        i=join ci=red l=2 w=2;
proc gplot data=budget;
   plot Yr2003 * Month;
```

Section 2: Producing Scatter and Line Plots

2.1 Creating Scatter Plots

2.2 Creating Line Plots

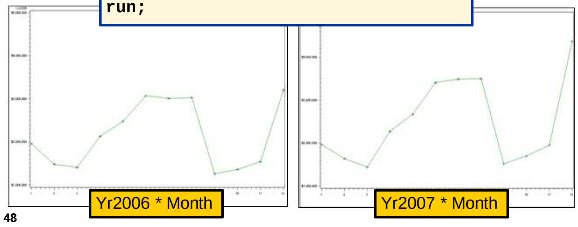
2.3 Creating Plots Containing Multiple Lines

2.4 Creating Other Types of Plots

Multiple Plot Requests

If you submit multiple PLOT requests in a single PLOT statement, each request produces a separate plot. The same SYMBOL statement is used for each plot.

```
symbol1 cv=green v=star i=join;
proc gplot data=budget;
   plot Yr2006 * Month
        Yr2007 * Month;
run;
```



Plots Containing Multiple Lines

The following methods can be used to produce multiple plot lines within one set of coordinated axes:

- use the OVERLAY option in the PLOT statement to overlay two or more plot requests
- use a PLOT2 statement in addition to the PLOT statement to produce multiple plot lines with two vertical axes
- Use a third, classification variable to produce a separate plot line for each value of that variable.

The OVERLAY Option

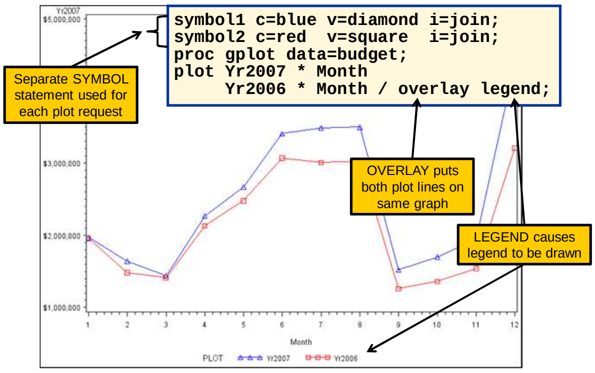
The OVERLAY option places all the plots that are generated by the PLOT statement on one set of axes.

```
plot Yr2007 * Month
     Yr2006 * Month / overlay legend;
```

Note the following when using the OVERLAY option:

- The horizontal axis variable should be the same for all plot requests.
- A legend is produced only if you include the LEGEND option.
- A separate SYMBOL definition is used for each plot.

The OVERLAY Option



```
symbol1 c=blue v=diamond i=join;
symbol2 c=red  v=square  i=join;
proc gplot data=budget;
plot Yr2007 * Month
     Yr2006 * Month / overlay legend;
```

Separate SYMBOL statement used for each plot request

OVERLAY puts both plot lines on same graph

LEGEND causes legend to be drawn

The PLOT2 Statement

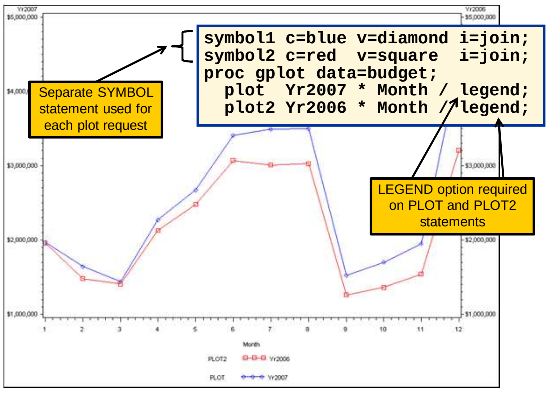
The PLOT2 statement produces a plot with a vertical axis on the right side of the graph. The syntax for the PLOT2 statement is identical to that for the PLOT statement.

Example:

```
plot  Yr2007 * Month / legend;
plot2 Yr2006 * Month / legend;
```

To include a complete legend, the LEGEND option must be specified on **both** the PLOT and PLOT2 statements.

The PLOT2 Statement



```
symbol1 c=blue v=diamond i=join;
symbol2 c=red  v=square  i=join;
proc gplot data=budget;
  plot  Yr2007 * Month / legend;
  plot2 Yr2006 * Month / legend;
```

Separate SYMBOL statement used for each plot request

LEGEND option required on PLOT and PLOT2 statements

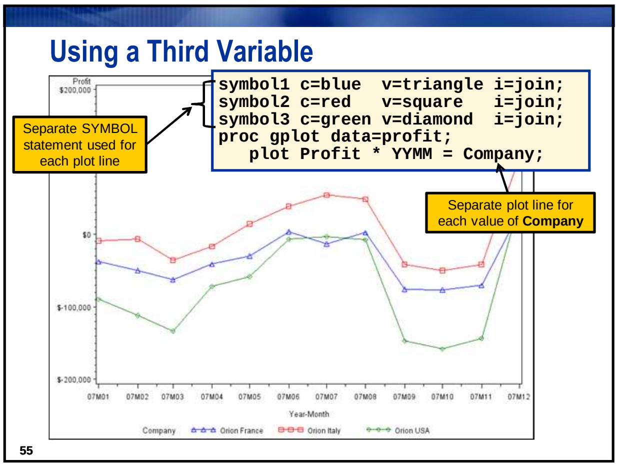
Using a Third Variable

A third (classificiation) variable can be used to produce multiple plot lines, one for each value of the variable.

Example:

```
plot Profit * YYMM = Company;
```

- A separate plot line is produced for each value of **Company**.
- A PLOT statement of this form produces a legend automatically.
- A separate SYMBOL definition is used for each value of **Company**.

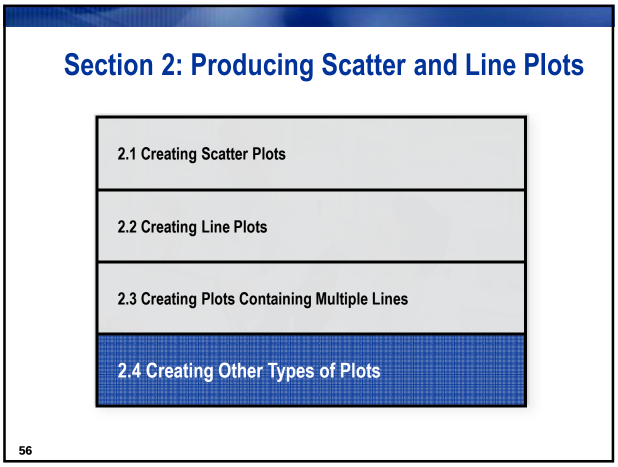


Using a Third Variable

```
symbol1 c=blue  v=triangle i=join;
symbol2 c=red   v=square   i=join;
symbol3 c=green v=diamond  i=join;
proc gplot data=profit;
   plot Profit * YYMM = Company;
```

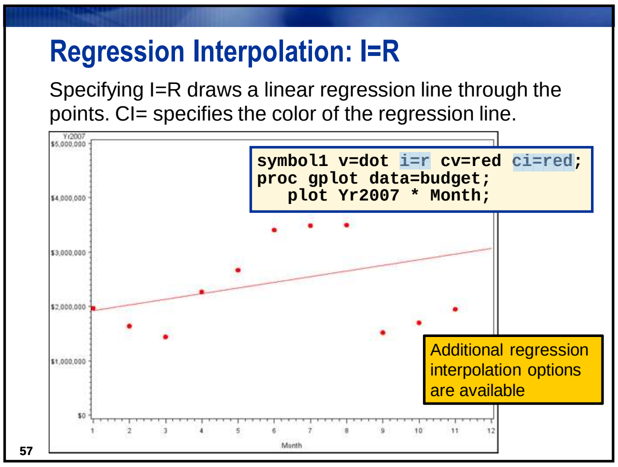
Separate SYMBOL statement used for each plot line

Separate plot line for each value of **Company**



Section 2: Producing Scatter and Line Plots

- 2.1 Creating Scatter Plots
- 2.2 Creating Line Plots
- 2.3 Creating Plots Containing Multiple Lines
- **2.4 Creating Other Types of Plots**

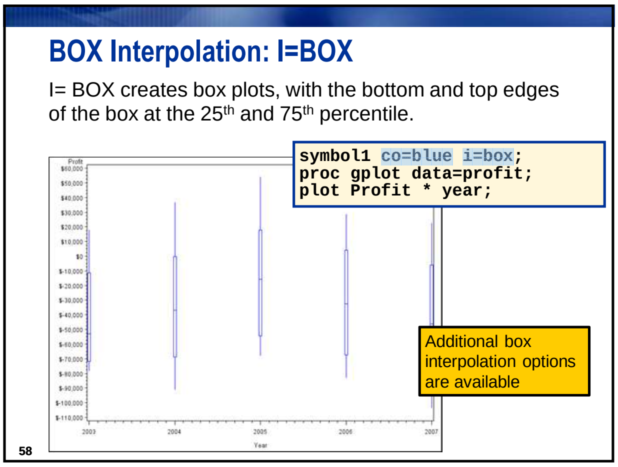


Regression Interpolation: I=R

Specifying I=R draws a linear regression line through the points. CI= specifies the color of the regression line.

```
symbol1 v=dot i=r cv=red ci=red;
proc gplot data=budget;
   plot Yr2007 * Month;
```

Additional regression interpolation options are available

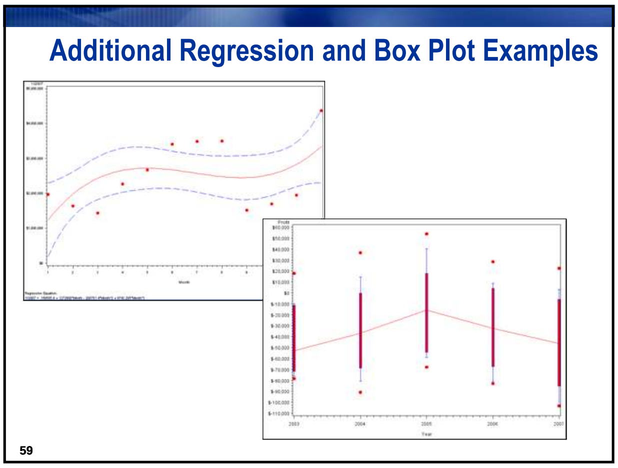


BOX Interpolation: I=BOX

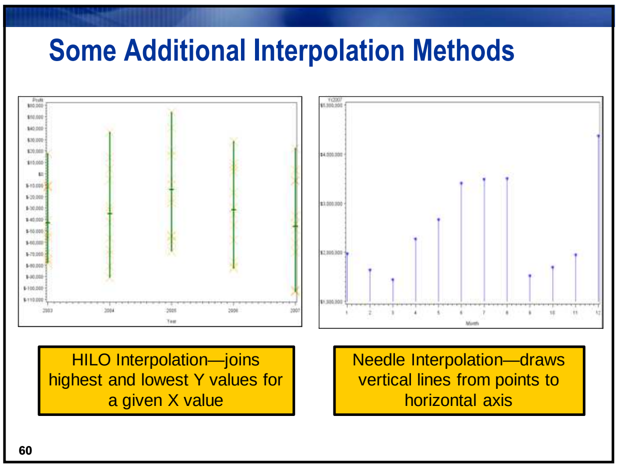
I= BOX creates box plots, with the bottom and top edges of the box at the 25th and 75th percentile.

```
symbol1 co=blue i=box;
proc gplot data=profit;
plot Profit * year;
```

Additional box interpolation options are available



Additional Regression and Box Plot Examples



Some Additional Interpolation Methods

HILO Interpolation—joins highest and lowest Y values for a given X value

Needle Interpolation—draws vertical lines from points to horizontal axis

Section 3: Producing Bar Charts

3.1 Creating Vertical and Horizontal Bar Charts

3.2 Creating Grouped and Subgrouped Bar Charts

The GCHART Procedure

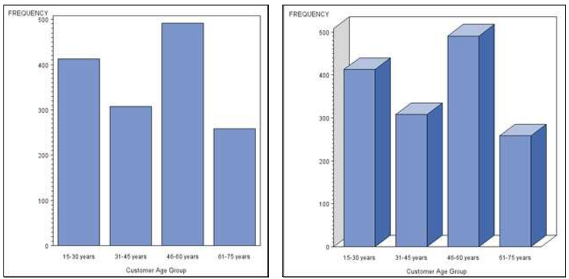
General form of the GCHART procedure to create vertical and horizontal bar charts:

PROC GCHART DATA = *SAS-data-set*;
    VBAR *chart-variable(s)* **/** *options*;
    VBAR3D *chart-variable(s)* **/** *options*;
    HBAR *chart-variable(s)* **/** *options*;
    HBAR3D *chart-variable(s)* **/** *options*;
RUN;

The ***chart variable*** *(also known as the* ***midpoint*** *variable)* is the variable that is used to determine the number of bars in the chart. By default, the length of the bar represents a frequency count for the values of the chart variable.

Vertical Bar Charts

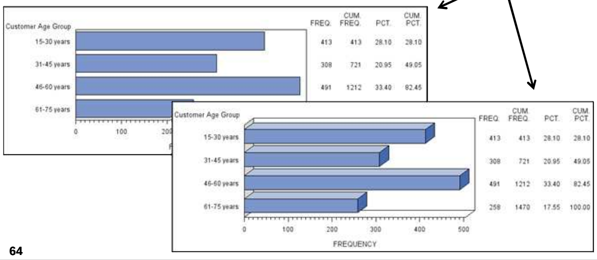
```
proc gchart data=orders;
   vbar Customer_Age_Group;
   vbar3d Customer_Age_Group;
```



Horizontal Bar Charts

```
proc gchart data=orders;
   hbar Customer_Age_Group;
   hbar3d Customer_Age_Group;
```

Statistics are displayed by default in HBAR and HBAR3D charts


```
Customer Age Group                FREQ.  CUM.   PCT.   CUM.
                                         FREQ.         PCT.
15-30 years                        413    413  28.10  28.10
31-45 years                        308    721  20.95  49.05
46-60 years                        491   1212  33.40  82.45
61-75 years
```

```
Customer Age Group                FREQ.  CUM.   PCT.   CUM.
                                         FREQ.         PCT.
15-30 years                        413    413  28.10  28.10
31-45 years                        308    721  20.95  49.05
46-60 years                        491   1212  33.40  82.45
61-75 years                        258   1470  17.55 100.00
```

Chart Variables

When the chart variable is a character variable, PROC GCHART draws a bar for each value of the variable.

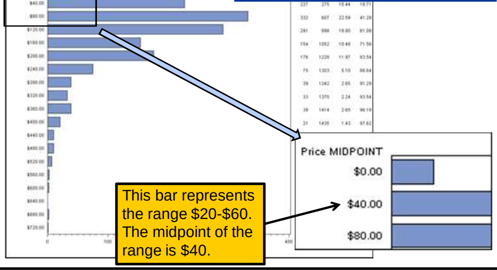
When the chart variable is a numeric variable, PROC GCHART, by default, does the following:

- treats the variable's values as a continuous range and divides the range into intervals
- represents each interval with a bar
- labels the bar with the interval midpoint value

If the Chart Variable is Numeric

In this example, the value of **Price** is divided into ranges and each bar represents a range. The bar label displays the midpoint of the range:

```
proc gchart data=orders;
   hbar price;
```

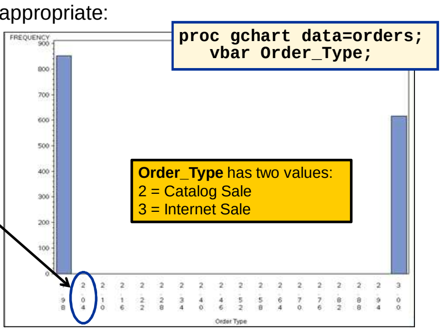


This bar represents the range $20-$60. The midpoint of the range is $40.

If the Chart Variable Has Few Values

If the chart value is numeric and has a small number of values, the axis is still divided into ranges, and the output may not be appropriate:

```
proc gchart data=orders;
   vbar Order_Type;
```



2.04 represents the midpoint of the interval 2.01 to 2.07

Order_Type has two values:
2 = Catalog Sale
3 = Internet Sale

Controlling the Midpoints of Bars

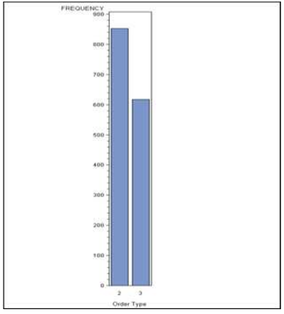
The following options control the midpoints of bars in a chart:

Option	Purpose
DISCRETE	draws a bar for each value of the numeric chart variable.
LEVELS = ***n***	specifies the number of midpoints (bars) for a numeric chart variable.
RANGE	displays the range of values the bar represents instead of the midpoint value.

Using the DISCRETE Option

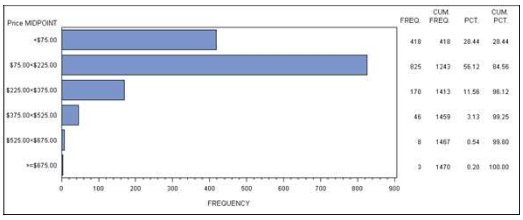
Example:

```
proc gchart data=orders;
   vbar Order_Type / discrete;
```



Using the LEVELS= and RANGE= Options

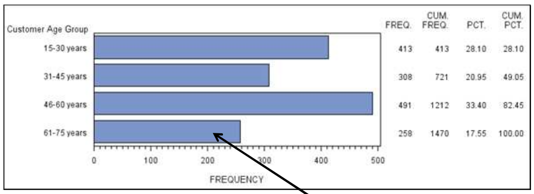
```
proc gchart data=orders;
hbar Price / levels=6 range;
```



Price MIDPOINT	FREQ.	CUM. FREQ.	PCT.	CUM. PCT.
<$75.00	418	418	28.44	28.44
$75.00<$225.00	825	1243	56.12	84.56
$225.00<$375.00	170	1413	11.56	96.12
$375.00<$525.00	46	1459	3.13	99.25
$525.00<$675.00	8	1467	0.54	99.80
>=$675.00	3	1470	0.20	100.00

What Does the Length of a Bar Represent?

By default, the length of each bar represents a frequency count for that value of the chart variable.



Customer Age Group	FREQ.	CUM. FREQ.	PCT.	CUM. PCT.
15-30 years	413	413	28.10	28.10
31-45 years	308	721	20.95	49.05
46-60 years	491	1212	33.40	82.45
61-75 years	258	1470	17.55	100.00

Length of this bar represents the number of observations (258) with a value of "61-75 years" for **Customer_Age_Group**

Other Possible Summary Statistics

The length of the bars can represent other statistics, such as:

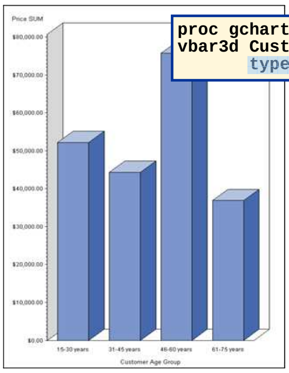
- the percentage of observations have each value of the chart variable
- the sum or mean of another variable for each value of the chart variable (such as total of SALES for each age group)

Specifying a Chart Statistic

The following options specify a statistic to be represented by the length of the bars in a chart:

Option	Purpose
TYPE = *statistic*	specifies what statistic the length of the bar represents. Possible statistics include FREQ (default), PERCENT, SUM, or MEAN. If you request SUM or MEAN, the SUMVAR= option is required.
SUMVAR = *numeric-variable*	specifies the variable to be used for sum or mean calculations.

Specifying a Summary Statistic

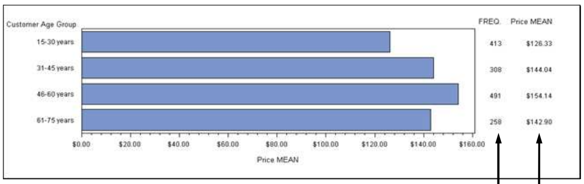


```
proc gchart data=orders;
vbar3d Customer_Age_Group /
         type=sum sumvar=Price;
```

For each value of **Customer_Age_Group**, the height of bars represents the sum of the values of **Price**

Specifying a Summary Statistic

```
proc gchart data=orders;
   hbar Customer_Age_Group /
        type=mean sumvar=Price;
```



Length of bars represents the mean value of **Price** for each value of **Customer_Age_Group**

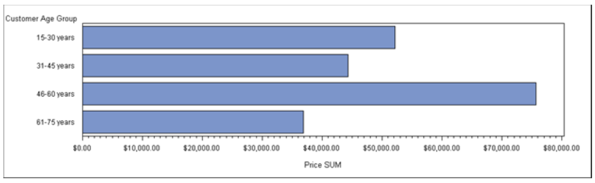
Statistics displayed depend on the value of SUMVAR= and TYPE=

Suppressing Statistics on HBAR Charts

```
proc gchart data=orders;
   hbar Customer_Age_Group /
        sumvar=Price
        type=sum
        nostats;
```

statistic represented by bars

no statistics displayed on chart



Section 3: Producing Bar Charts

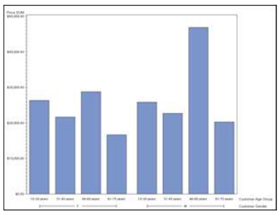
3.1 Creating Vertical and Horizontal Bar Charts

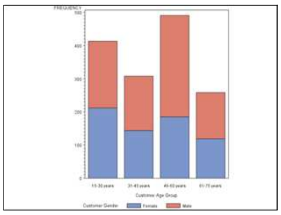
3.2 Creating Grouped and Subgrouped Bar Charts

Grouped and Subgrouped Bar Charts

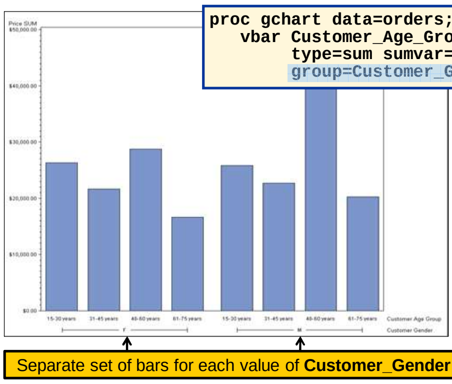
The bars of a vertical and horizontal bar chart can be organized as follows:

- **Groups**, with a group of bars for each value of a group variable
- **Subgroups**, with each bar subdivided according to values of a subgroup variable





Grouped Bar Chart

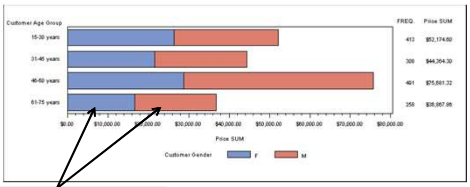


```
proc gchart data=orders;
   vbar Customer_Age_Group /
        type=sum sumvar=Price
        group=Customer_Gender;
```

Separate set of bars for each value of **Customer_Gender**

Subgrouped Bar Chart

```
proc gchart data=orders;
   hbar Customer_Age_Group /
        sumvar=Price
        type=sum
        subgroup=Customer_Gender;
```



Bars are divided into segments based on value of **Customer Gender**

Each subgroup value is assigned a different pattern

Section 4: Customizing the Appearance of SAS GRAPH Output

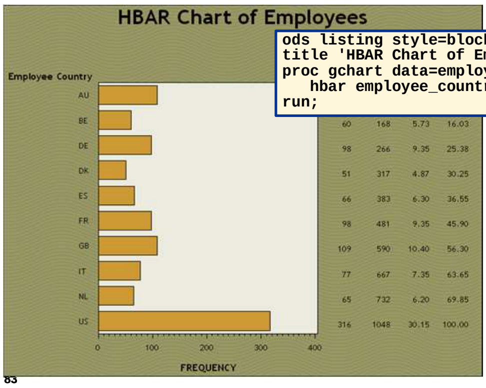
Overriding Style and Graph Attributes

You can override default graph attributes using one or more of these methods:

- specify alternate ODS styles or create your own style
- use the GOPTIONS statement
- use options specified within the procedure, or on global statements such as TITLE, FOOTNOTE, AXIS, LEGEND, PATTERN, and SYMBOL statements

---

Using an Alternate Style



```
ods listing style=blockprint;
title 'HBAR Chart of Employees';
proc gchart data=employees;
   hbar employee_country;
run;
```

Section 4: Customizing the Appearance of SAS GRAPH Output

The GOPTIONS Statement

The GOPTIONS statement specifies options that control various aspects of the SAS/GRAPH environment. The syntax is similar to the OPTIONS statement in Base SAS. For example:

```
goptions hsize=6 vsize=4 ; /* set height and width */
                           /* of graph             */
```

To reset all GOPTIONS to their defaults, specify:

```
goptions reset=all;
```

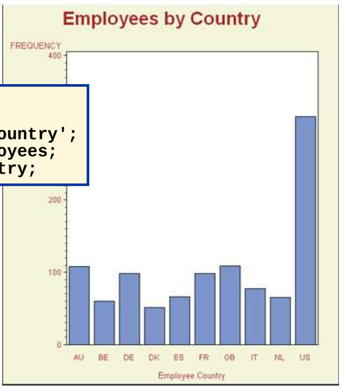
Using GOPTIONS to Control Color

Common GOPTIONS that impact color are shown below.

To. . .	Use This GOPTION
Specify the background color of the graph.	CBACK=
Select the default color for all text	CTEXT=

CBACK= and CTEXT= Options

```
goptions cback=beige
         ctext=brown;
title 'Employees by Country';
proc gchart data=employees;
   vbar employee_country;
```



Text Font and Height GOPTIONS

Common GOPTIONS that control text appearance are:

To. . .	Use This GOPTION
Set the default font for all text	FTEXT=
Specify the default height of all text	HTEXT=

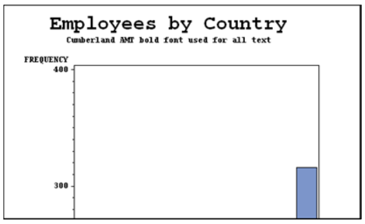
If the font name contains a blank, the name must be enclosed in blanks. For example

```
goptions ftext='Arial Black';
```

FTEXT= Option

The FTEXT= option affects the text in the graph output, including the title.

```
goptions ftext='Cumberland AMT/bold';
title 'Employees by Country';
title2 'Cumberland AMT bold font used for all text';
```



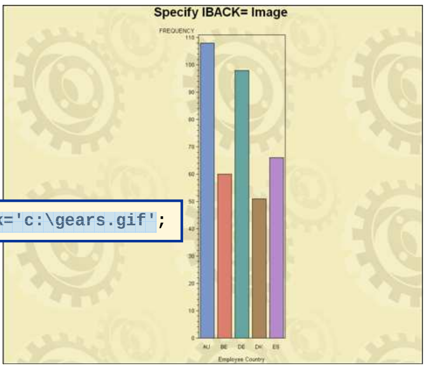
Appearance GOPTIONS

GOPTIONS that control the size and appearance of the graphics area include:

To. . .	Use This GOPTION
Set the horizontal and vertical dimensions of the graph in inches	HSIZE= VSIZE=
Draw a border around the graphics area	BORDER
Specify an image file to display in a graph's background area	IBACK=

Using the IBACK= Option

The default IBACK= behavior is for the image to be tiled in the graphics area.



```
goptions iback='c:\gears.gif';
```

Section 4: Customizing the Appearance of SAS GRAPH Output

SAS/GRAPH Colors

You can specify color names using any combination of these color-naming schemes, such as:

- RGB (red, green, blue) color values
- Gray scale color names
- SAS color names (defined in the SAS Registry)
- HLS (hue, lightness, saturation) values
- CMYK (cyan, magenta, yellow, black)
- HSV (hue, saturation, brightness).

RGB Color Codes

Use the RGB color-naming scheme to specify a color in terms of its red, green, and blue components. Color names are of the form CX***rrggbb***, where

- CX indicates to use the RGB color specification
- ***rr*** is the red component
- ***gg*** is the green component
- ***bb*** is the blue component.

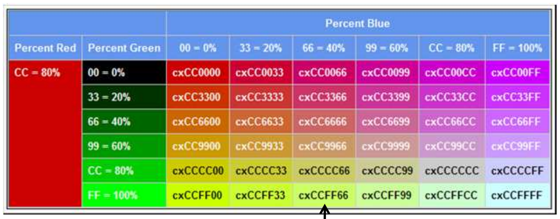
The components are given as hexadecimal numbers in the range 00 through FF. Most Web page (HTML) colors use RGB color values.

Example:

```
goptions cback=cx97b4a7;
```

RGB Color Examples

Some selected RGB colors are shown below:



		Percent Blue					
Percent Red	Percent Green	00 = 0%	33 = 20%	66 = 40%	99 = 60%	CC = 80%	FF = 100%
CC = 80%	00 = 0%	cxCC0000	cxCC0033	cxCC0066	cxCC0099	cxCC00CC	cxCC00FF
	33 = 20%	cxCC3300	cxCC3333	cxCC3366	cxCC3399	cxCC33CC	cxCC33FF
	66 = 40%	cxCC6600	cxCC6633	cxCC6666	cxCC6699	cxCC66CC	cxCC66FF
	99 = 60%	cxCC9900	cxCC9933	cxCC9966	cxCC9999	cxCC99CC	cxCC99FF
	CC = 80%	cxCCCC00	cxCCCC33	cxCCCC66	cxCCCC99	cxCCCCCC	cxCCCCFF
	FF = 100%	cxCCFF00	cxCCFF33	cxCCFF66	cxCCFF99	cxCCFFCC	cxCCFFFF

cxCCFF66 is:
- 80% red
- 100% green
- 40% blue

Gray-Scale Color Codes

Use gray-scale codes to specify colors in terms of gray components. Gray-scale color names are of the form GRAY*xx*, where *xx* is the lightness of the gray.

The lightness component is given as a hexadecimal number in the range 00 through FF.

Color	Gray Scale Name
white (100% lightness)	GRAYFF
black (0% lightness)	GRAY00
60% lightness	GRAY99

Example:

```
goptions cback=grayaa;
```

SAS Color Names

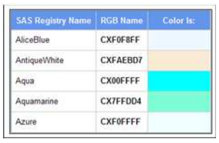
SAS provides a set of color names that you can use to specify colors.

Example:

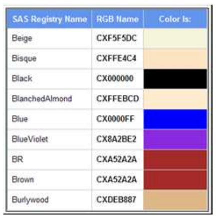
```
goptions cback=FloralWhite
         ctext=blue;
```

SAS Colors

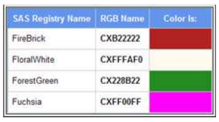
Some selected SAS colors are shown below:



SAS Registry Name	RGB Name	Color Is:
AliceBlue	CXF0F8FF	⬜
AntiqueWhite	CXFAEBD7	⬜
Aqua	CX00FFFF	🟦
Aquamarine	CX7FFDD4	🟩
Azure	CXF0FFFF	⬜



SAS Registry Name	RGB Name	Color Is:
Beige	CXF5F5DC	⬜
Bisque	CXFFE4C4	⬜
Black	CX000000	⬛
BlanchedAlmond	CXFFEBCD	⬜
Blue	CX0000FF	🟦
BlueViolet	CX8A2BE2	🟪
BR	CXA52A2A	🟥
Brown	CXA52A2A	🟥
Burlywood	CXDEB887	🟫



SAS Registry Name	RGB Name	Color Is:
FireBrick	CXB22222	🟥
FloralWhite	CXFFFAF0	⬜
ForestGreen	CX228B22	🟩
Fuchsia	CXFF00FF	🟪

See the SAS Help System for more information on SAS color names.

TITLE and FOOTNOTE Statements

General form of TITLE and FOOTNOTE statements:

TITLE*n options* '*text*'**;**
FOOTNOTE*n options* '*text*'**;**

Values for *n* range from 1 to 10. If *n* is omitted, it is assumed to be 1.

For SAS/GRAPH output, TITLE and FOOTNOTE statements can contain options that control attributes such as size, font, and color.

TITLE and FOOTNOTE Statement Options

To…	Use This Option

Control font face	FONT= or F=
	`title f='Albany AMT' 'Text';`

Control color	COLOR= or C=
	`title c=blue 'Text';`

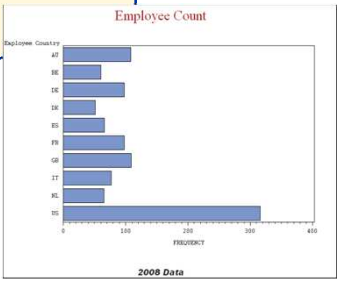
Control text height	HEIGHT or H= `title h=14pt 'Text';`

You can control multiple attributes. For example:

```
title f='Arial' c=red h=3cm 'Text';
```

Controlling the Appearance of Text

```
goptions ftext='Courier';
title f='Times New Roman' h=24pt c=brown
     'Employee Count';
footnote f='Arial Black' h=14pt c=gray33
     '2008 Data';
proc gchart data=employees;
   hbar employee_country/
        nostats;
```

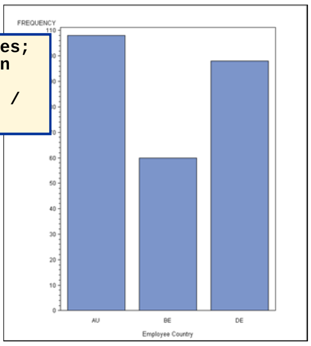


GOPTIONS FTEXT= specifies the font for all text, but is overridden in TITLE and FOOTNOTE statements.

Controlling the Width of Bars

Use the WIDTH= (or W=) option on the chart statement to control the width of bars:

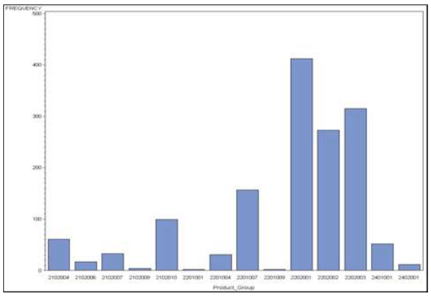
```
proc gchart data=employees;
where employee_country in
    ('AU' 'BE' 'DE');
    vbar employee_country /
      width=15;
```



Width is specified in units of character cells

Bar Patterns

Each bar is filled with a pattern. For simple and grouped bar charts, the bars are filled with the default fill (SOLID) and the first data color specified in the style template

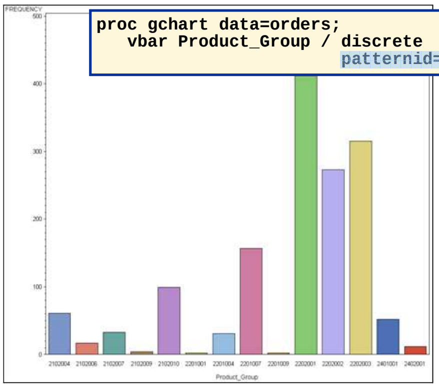


Distributing Pattern Definitions

The PATTERNID= option distributes patterns in a bar chart:

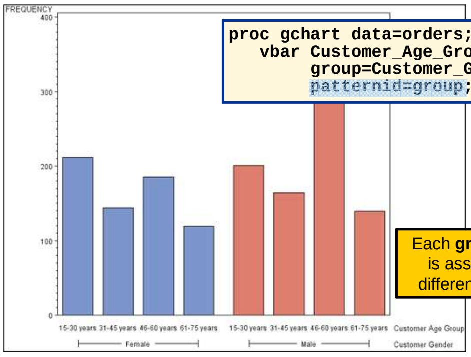
Option	Purpose
PATTERNID = MIDPOINT	assigns a diffferent PATTERN definition to each value of the chart (midpoint) variable.
PATTERNID = GROUP	assigns a different PATTERN definition to each value of the group variable.

PATTERNID=MIDPOINT



```
proc gchart data=orders;
   vbar Product_Group / discrete
                        patternid=midpoint;
```

PATTERNID=GROUP



```
proc gchart data=orders;
   vbar Customer_Age_Group /
        group=Customer_Gender
        patternid=group;
```

Each **group** value is assigned a different pattern

The PATTERN Statement

You can assign patterns explicitly using a PATTERN statement. General form of the PATTERN statement:

PATTERN*n* *options*;

The following options can be specified:

Option	Purpose
COLOR= | C=	specifies the color of the fill pattern.
VALUE= | V=	specifies the fill pattern. The default fill is solid.

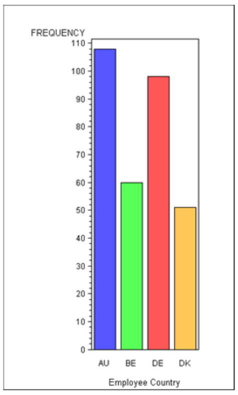
Example:

```
pattern1 c=red v=solid;
pattern2 c=blue v=empty;
```

Using PATTERN Statements

The PATTERN statements below override the default patterns dictated by the style.

```
pattern1 c=lightblue v=s;
pattern2 c=lightgreen v=s;
pattern3 c=lightred v=s;
pattern4 c=lightorange v=s;
vbar employee_country /
     patternid=midpoint;
```



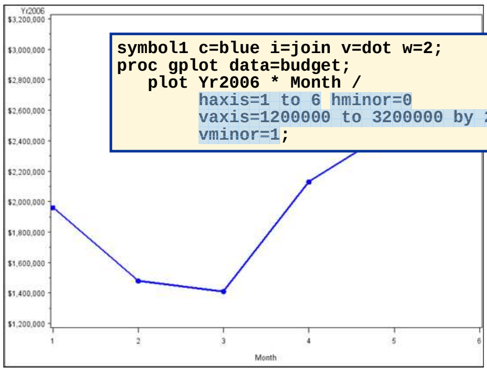
Section 4: Customizing the Appearance of SAS GRAPH Output

Controlling Tick Marks in a Plot

You can use the following PLOT/PLOT2 statement options in the GPLOT procedure to control major and minor tick marks:

Option	Purpose
HAXIS = *value-list* **VAXIS =** *value-list*	specifies values for major tick marks on the horizontal axis and vertical axes
HMINOR = *number* **VMINOR =** *number*	specifies the number of minor tick marks to draw between each major tick mark on the horizontal and vertical axes.

Controlling Tick Marks in a Plot



```
symbol1 c=blue i=join v=dot w=2;
proc gplot data=budget;
   plot Yr2006 * Month /
        haxis=1 to 6 hminor=0
        vaxis=1200000 to 3200000 by 200000
        vminor=1;
```

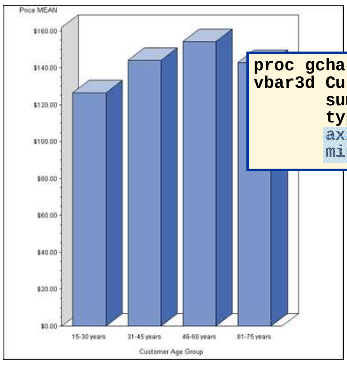
Controlling Tick Marks in a Chart

You can use the following action statement options in the GCHART procedure to control major and minor tick marks:

Option	Purpose
AXIS = *value-list*	specifies values that correspond to major tick marks on the response axis.
MINOR = *number*	specifies the number of minor tick marks drawn between major tick marks on the response axis.

✎ The ***response*** axis displays the scale of values for the chart statistic.

Controlling Tick Marks in a Chart



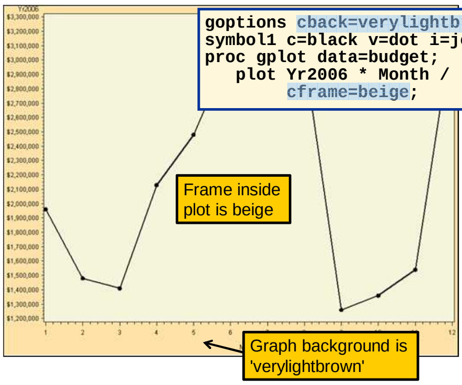
```
proc gchart data=orders;
vbar3d Customer_Age_Group /
       sumvar=Price
       type=mean
       axis=0 to 160 by 20
       minor=3;
```

Controlling the Axis Frame

You can use the following statement options in the GPLOT and GCHART procedures to control the appearance of the axis frame:

Option	Purpose
CFRAME = *color*	specifies a background color for the frame drawn around the axis area.
NOFRAME	specifies that no frame be drawn around the axis area.

Controlling the Axis Frame



```
goptions cback=verylightbrown;
symbol1 c=black v=dot i=join w=2;
proc gplot data=budget;
   plot Yr2006 * Month /
        cframe=beige;
```

Frame inside plot is beige

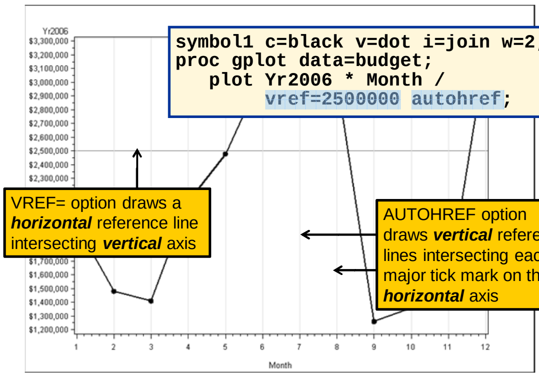
Graph background is 'verylightbrown'

Adding Reference Lines to a Plot

You can use the following PLOT/PLOT2 statement options in the GPLOT procedure to add reference lines:

Option	Purpose
HREF = *value-list*	generates vertical lines that intersect the **horizontal** axis at the specified values.
VREF = *value-list*	generates horizontal lines that intersect the **vertical** axis at the specified values.
AUTOHREF	generates vertical lines that intersect the **horizontal** axis at major tick marks.
AUTOVREF	generates horizontal lines that intersect the **vertical** axis at major tick marks.

Adding Reference Lines to a Plot



```
symbol1 c=black v=dot i=join w=2;
proc gplot data=budget;
   plot Yr2006 * Month /
        vref=2500000 autohref;
```

VREF= option draws a ***horizontal*** reference line intersecting ***vertical*** axis

AUTOHREF option draws ***vertical*** reference lines intersecting each major tick mark on the ***horizontal*** axis

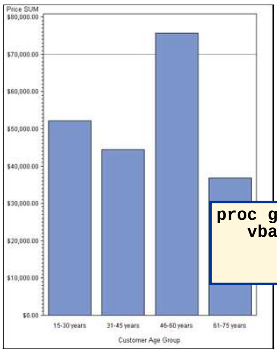
---

Adding Reference Lines to a Chart

You can use the following chart statement options in the GCHART procedure to add reference lines:

Option	Purpose
REF = *value-list*	generates lines that intersect the response axis at the specified values.
AUTOREF	generates lines that intersect the response axis at major tick marks.
CLIPREF	clips the reference lines at the bars to produce the effect of the lines appearing behind the bars.

Adding Reference Lines to a Chart

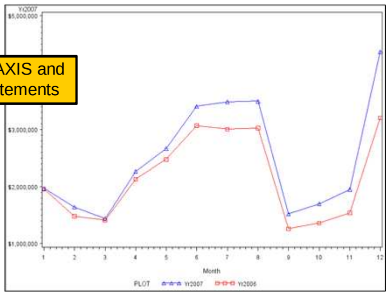


CLIPREF option places reference line behind bars

```
proc gchart data=orders;
   vbar Customer_Age_Group /
        sumvar=Price
        ref=70000 clipref;
```

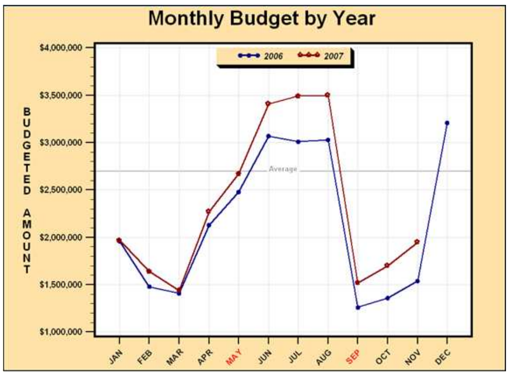
AXIS and LEGEND Statements (Overview)

You can achieve much greater control over the appearance of your graphs by using AXIS and LEGEND statements.



Plot without AXIS and LEGEND statements

Plot Using AXIS and LEGEND Statements



Plot Using AXIS and LEGEND Statements

```
goptions reset=all cback=verylightbrown;
symbol1 c=darkblue v=dot i=join w=3 h=2pct;
symbol2 c=darkred v='5F'x font='Monotype Sorts' i=join w=3 h=2pct;

axis1 w=4 label=(h=16pt f='Arial/bo' a=-90 r=90 'BUDGETED AMOUNT')
      value=(h=12pt f='Arial/bo')
      order=1000000 to 4000000 by 500000
      major=(h=2 pct w=2) minor=(n=4 h=1 pct) length=80 pct
      reflabel=(c=gray99 f='Arial/bo' h=10pt j=center 'Average');

axis2 value=(h=12pt a=45 f='Arial/bo' t=1 'JAN' t=2 'FEB' t=3 'MAR'
            t=4 'APR' t=5 c=red 'MAY' t=6 'JUN' t=7 'JUL' t=8 'AUG'
            t=9 c=red 'SEP' t=10 'OCT' t=11 'NOV' t=12 'DEC')
      label=none minor=none major=(h=2 pct w=2) offset=(5,5) pct;

legend1 position=(top center inside) frame
        cframe=verylightbrown cblock=black
        label=(f='Arial/bo/it' h=12pt ' ')
        value=(f='Arial/bo/it' h=12pt t=1 '2006' t=2 '2007');

proc gplot data=budget;
     plot Yr2006 * Month Yr2007*month/ overlay legend vaxis=axis1
     haxis=axis2 vref=2700000 grid legend=legend1;
title h=28pt  'Monthly Budget by Year';
run;
```

Section 5: Generating Graphs for Use in Other Applications

5.1 Creating Image and Document Files

Image and Document Files

In many cases, you will want to create output files in one of the following standard formats:

- Image Files (contain a graph only)
  - GIF
  - JPEG
  - Windows Metafile (WMF or EMF)
- Document files (can contain images and text)
  - HTML
  - PDF
  - RTF (Rich Text Format)

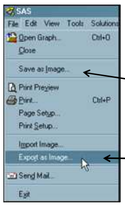
Methods to Create Image and Document Files

You can use the following methods to create image and document files from SAS/GRAPH.

Method	Type of Output Created
Manually save an image file from the GRAPH1 window.	BMP, JPG, TIF, PNG, GIF, WMF, EMF, + more
Use ODS to create a document file	HTML, RTF, or PDF documents that include images and/or tables and text.

Exporting Image Files from the GRAPH1 Window

After a graph is displayed in the GRAPH1 window, you can create an image file by selecting **<u>File</u>** ⇨ **<u>Export as Image…</u>** and then selecting the appropriate file type from the **`Save as type`** box.

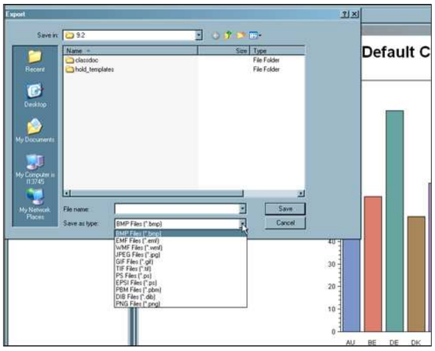


Do ***not*** select 'Save as Image'

Do select 'Export as Image'

Exporting Image Files from the GRAPH1 Window

The image file types are listed in the **`Save as type`** box.



Creating HTML Files with ODS

The ODS HTML statement opens, closes, or manages the HTML destination.

To create an HTML file using ODS, submit the following:

```
ODS HTML PATH='directory-name'
         BODY='HTML-file-name';
  SAS/GRAPH statements to produce graph
  .
  .
  run;
ODS HTML CLOSE;
```

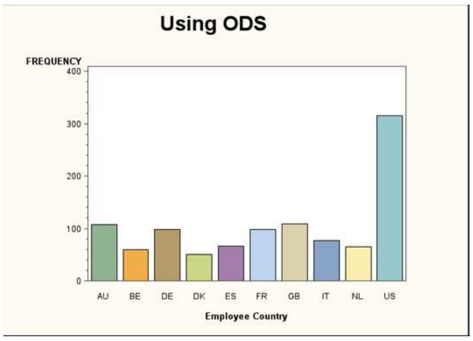
The ODS CLOSE statement must come after the RUN statement.

Creating an HTML File with ODS

```
ods html path='c:\' body='sesug.html' style=analysis;
title 'Using ODS';
proc gchart data=employees;
  vbar employee_country / patternid=midpoint;
run;
ods html close;
```

- The PATH= option specifies the directory into which both the HTML and image files will be placed.
- The BODY= option specifies the name of HTML file to create.

HTML Output Using ANALYSIS Style



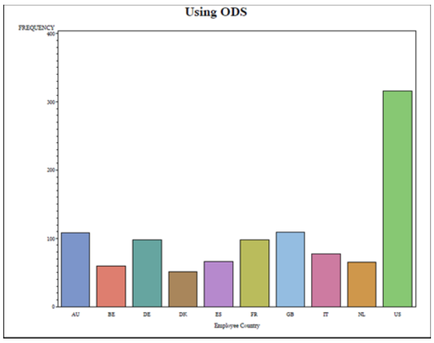
sesug.html

Creating a PDF File

The FILE= option specifies the name and location of the PDF output file.

```
ods pdf file='c:\sesug.pdf';
title 'Using ODS';
proc gchart data=employees;
  vbar employee_country / patternid=midpoint;
run;
ods pdf close;
```

PDF Output



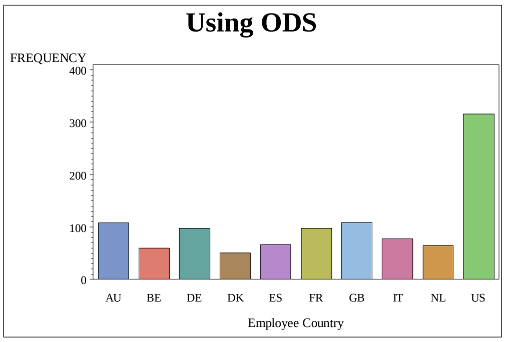
sesug.pdf

Creating an RTF File

The FILE= option specifies the name and location of the RTF output file.

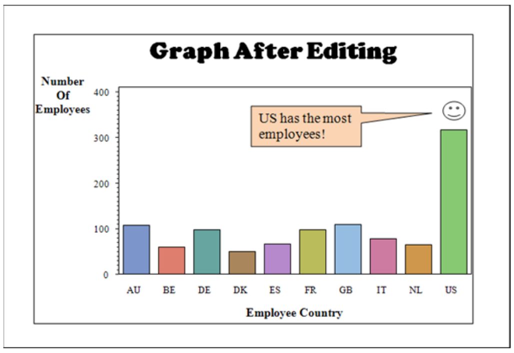
```
ods rtf file='c:\sesug.rtf';
title 'Using ODS';
proc gchart data=employees;
   vbar employee_country / patternid=midpoint;
run;
ods rtf close;
```

RTF Output



sesug.rtf

Graph After Editing in Microsoft Word



Exporting Graphs to Microsoft Office

When exporting graphs to Microsoft Office applications the following methods are recommended:

- To insert a graph into an Excel spreadsheet, PowerPoint presentation, or existing Word document, create an EMF (Enhanced Windows Metafile) image file using the **File→Export as Image** pulldown in the GRAPH1 window. Use **Insert→Picture** in Excel, Powerpoint, or Word.
- To create a new Word document containing a graph, create an RTF file using ODS, and open the file in Word.

Recommended Technical Documents

Detailed information on exporting SAS/GRAPH output to other environments can be found in the following SAS Technical Documents:

- TS-674, “An Introduction to Exporting SAS/GRAPH Output to Microsoft Office”, at **http://support.sas.com/techsup/technote/ts674/ts674.html**
- TS-659, “Exporting SAS/GRAPH Output to PDF Files”, at **http://support.sas.com/techsup/technote/ts659/ts659.pdf**

Section 6: Conclusion

6.1 Additional Information

SAS/GRAPH Training

The following courses are available from the SAS Education Division:

- ***SAS/GRAPH 1: Essentials*** -- covers device-based graphics
- ***Statistical Graphics with ODS*** -- covers SG procedures as well as statistical graphics that can be produced by procedures in Base SAS and SAS/STAT.

See **http://support.sas.com/training** for more information on these courses.

Web Sites

The following web sites contain additional information and samples for SAS/GRAPH:

- **http://support.sas.com/sassamples/graphgallery/** -- a gallery of samples (including code) for both device-based and SG procedures.
- **http://www.robslink.com/SAS/Home.htm** -- hundreds of creative examples of creating complex graphs with SAS/GRAPH
- **https://support.sas.com/rnd/datavisualization/** -- lots of tips, programs, and downloads from the Data Visualization Division at SAS Institute.

Thank you, and enjoy the conference! (what's left)

2010
SESUG
September 26–28
Savannah, GA