CHAPTER

13

The GCHART Procedure

Overview

The GCHART procedure 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. The charted variables can be either numeric or character.

The procedure calculates these statistics:

- frequency or cumulative frequency counts
- percentages or cumulative percentages
- sums
- means.

Use the GCHART procedure to

- display and compare exact and relative magnitudes
- examine the contribution of parts to the whole
- analyze where data are out of balance.

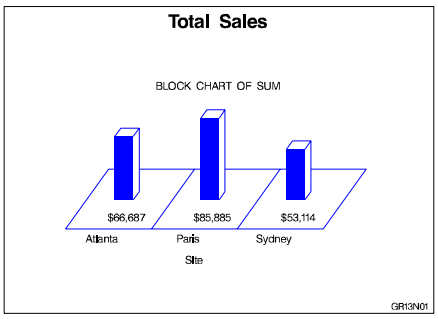
About Block Charts

Block charts display the relative magnitude of data with blocks of varying height, each set in a square that represents a category of data (midpoint). Because block charts do not use axes, they are most useful when the relative magnitude of the blocks is more significant than the exact magnitude of any particular block.

Figure 13.1 on page 520 shows a simple block chart of total sales for three manufacturing sites. Each site is a midpoint and occupies one square. The name of the site (the midpoint value) is printed below the square. Midpoint values are, by default, arranged in ascending order from left to right. The label below the midpoint grid names the chart variable.

Sales for the site (the chart statistic) are represented by the height of the block; sales amount (the formatted statistic value) is printed below the block. The heading above the blocks describes the type of statistic, in this case SUM.

Figure 13.1 Block Chart (GR13N01)



The program for this chart is in Example 1 on page 582. For more information on producing block charts, see “BLOCK Statement” on page 532.

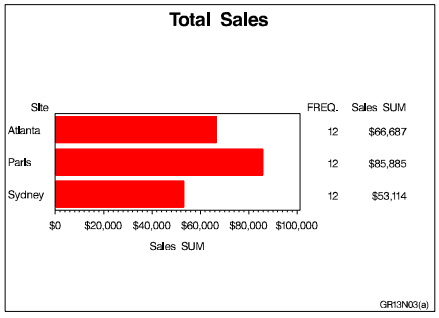
About Bar Charts

Horizontal and vertical bar charts display the magnitude of data with bars, each of which represents a category of data (midpoint). The length (or height) of the bars represents the value of the chart statistic for the corresponding midpoint.

Figure 13.2 on page 521 shows a simple horizontal bar chart of total sales for three manufacturing sites. Each site is a midpoint and is displayed as a bar. The name of the site (the midpoint value) is printed on the midpoint axis beside the bar. Midpoint values are, by default, arranged in ascending order from top to bottom of the chart and labeled with the name of the chart variable.

The chart statistics, in this case total sales for each site, are represented by the length of the bars. The response axis displays the scale of values for the chart statistic. The table of statistics to the right of the bars displays the exact statistic for each bar. Both a column in the table and the response axis are labeled with the name of the summary variable and the type of statistic.

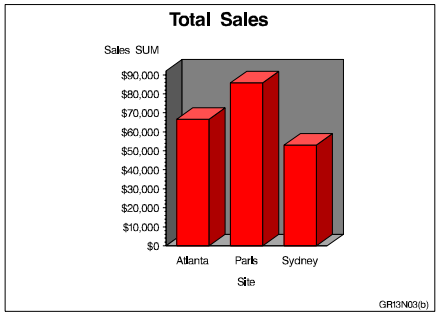
Figure 13.2 Horizontal Bar Chart (GR13N03(a))



The program for this chart is Example 3 on page 586.

Figure 13.3 on page 521 shows the same data presented as a vertical bar chart. The two types of bar charts have essentially the same characteristics, except that horizontal bar charts by default display a table of statistic values to the right of the bars, while vertical bar charts can optionally display the statistic value above or inside of each bar.

Figure 13.3 Vertical Bar Chart (GR13N03(b))



The program for this chart is Example 3 on page 586. For more information on producing horizontal and vertical bar charts, see “HBAR, HBAR3D, VBAR, and VBAR3D Statements” on page 541.

About Pie and Donut Charts

Pie and donut charts represent the relative contribution of parts to the whole by displaying data as wedge-shaped "slices" of a circle (either a "pie" or "donut"). Each slice represents a category of data (midpoint). The size of each slice (length of the arc) represents the contribution of the corresponding midpoint to the total chart statistic. Donut charts look like pie charts except that they have a hole in the middle in which you can place text.

Figure 13.4 on page 522 shows a pie chart of total sales for three manufacturing sites. Each site is a midpoint and is displayed as a slice. By default, the slices are ordered counterclockwise beginning at the 3 o'clock position.

Sales for the site (the chart statistic) are represented by the size of the slice. Both the sales amount (the formatted value of the chart statistic) and the name of the site (the midpoint value) are printed outside of the slice. You can also label pie slices with the percentage of the total statistic value that they represent. The heading above the pie describes the type of statistic (SUM), and names the summary variable (SALES) and the chart variable (SITE).

Figure 13.5 on page 522 show the three-dimensional version of the same pie chart.

Figure 13.4 Pie Chart (GR13N08(a))

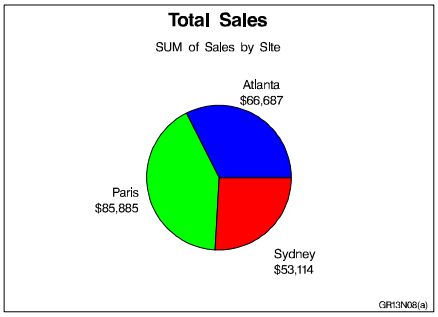
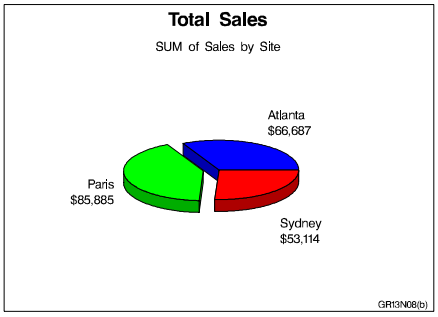


Figure 13.5 3D Pie Chart (GR13N08(b))



The program for these charts is Example 8 on page 610. For more information on producing pie or donut charts, see “PIE, PIE3D, and DONUT Statements” on page 559.

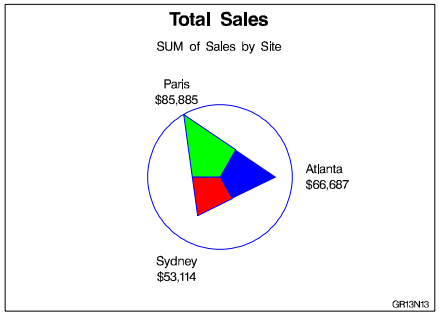
About Star Charts

Star charts display data as lines ("spines") radiating from the center of a circle toward the perimeter. Each spine represents a category of data (midpoint). The length of a spine represents the magnitude of the chart statistic for that midpoint starting at the center of the circle, which by default represents 0. The radius of the circle is the length of the longest spine (greatest statistic value) in the chart. Instead of spines, star charts can also display the chart statistic as slices, which are enclosed areas formed by connecting the ends of the spines.

Figure 13.6 on page 523 shows the total sales for the three manufacturing sites as a star chart. Each site is a midpoint and is displayed as a spine. By default the ends of the spines are connected and they are ordered counterclockwise beginning at the 3 o'clock position.

Sales for the site (the chart statistic) are represented by the length of the spine. Both the sales amount (the formatted statistic value) and the name of the site (the midpoint value) are printed outside of the star chart. You can also label star charts with the percentage of the total statistic value that they represent. The heading above the chart describes the type of statistic (SUM), and names the summary variable (SALES) and the chart variable (SITE).

Figure 13.6 Star Chart (GR13N13)



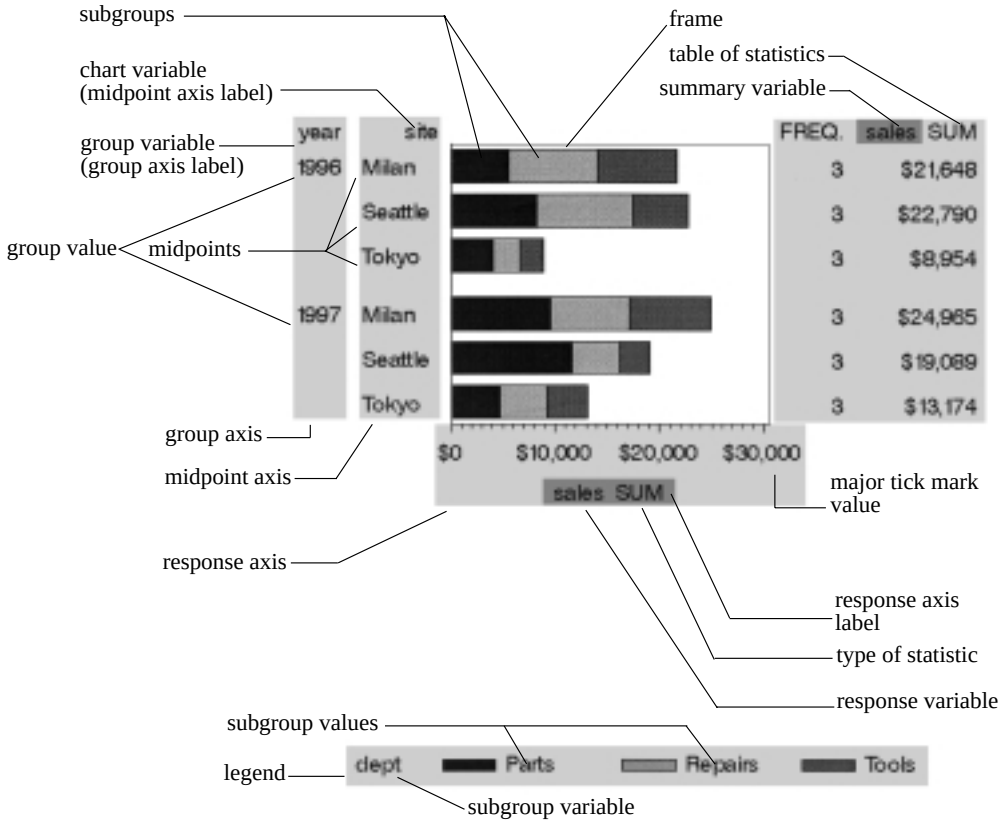
The program for this chart is Example 13 on page 620. For more information on producing star charts, see “STAR Statement” on page 573.

Concepts

The GCHART procedure produces charts based on the values of a *chart variable*. These values are represented by a set of *midpoints*. The chart itself displays information about the chart variable in the form of *chart statistics*.

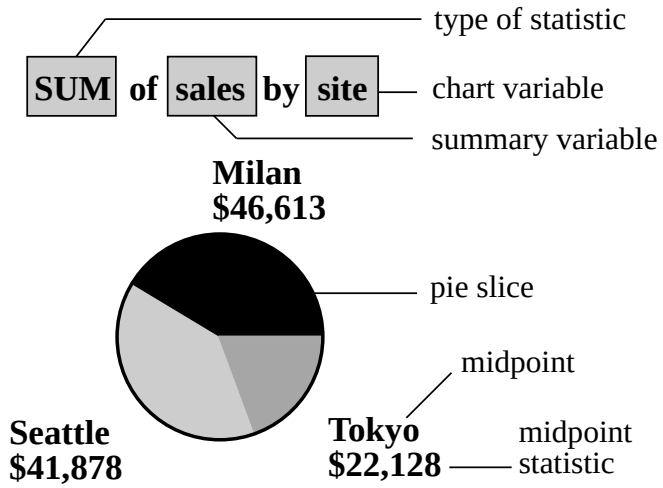
Figure 13.7 on page 524 and Figure 13.8 on page 524 illustrate these terms as well as other terms used with the GCHART procedure.

Figure 13.7 Terms Used with Bar Charts



Bar charts have two axes: a midpoint axis that shows the categories of data, and a response axis that displays the scale of values for the chart statistic. The response axis is divided into evenly spaced intervals identified with major tick marks that are labeled with the corresponding statistic value. Minor tick marks are evenly distributed between the major tick marks. Each axis is labeled with the chart variable name or label. The response axis is also labeled with the statistic type.

Figure 13.8 Terms Used with Pie and Donut Charts



Pie charts show statistics based on values of a variable called the chart variable. Generally, the values of the chart variable are represented by the slices in the chart. Next to each pie slice a number (or character string) appears that identifies the value or range of values assigned to that slice by the GCHART procedure. This number (or character string) is known as the *midpoint* for that slice. The statistic value for each midpoint is displayed beneath the midpoint. The slices in the chart represent all the values of the chart variable included in the chart. The number of degrees included in each slice represents the statistic value for the midpoint.

About Chart Variables

The *chart variable* is the variable in the input data set whose values determine the categories of data represented by the bars, blocks, slices, or spines. The chart variable generates the midpoints to which each observation in the data set contribute.

The chart variable can be either character or numeric. Character chart variables contain character values, which are always discrete. Numeric chart variables fall into two categories: discrete and continuous.

- *Discrete variables* contain a finite number of specific numeric values that are to be represented on the chart. For example, a variable that contains years, such as 1984 or 2001, is a discrete variable.
- *Continuous variables* contain a range of numeric values that are to be represented on the chart. For example, a variable of temperature data that contains real values between 0 and 212 is a continuous variable.

Numeric chart variables are always treated as continuous variables unless the DISCRETE option is used in the action statement.

Missing Values

By default, the GCHART procedure ignores missing midpoint values for the chart variable. If you specify the MISSING option, missing values are treated as a valid midpoint and are included on the chart. Missing values for the group and subgroup variables are always treated as valid groups and subgroups.

When the value of the variable that is specified in the FREQ= option is missing, 0, or negative, the observation is excluded from the calculation of the chart statistic.

When the value of the variable specified in the SUMVAR= option is missing, the observation is excluded from the calculation of the chart statistic.

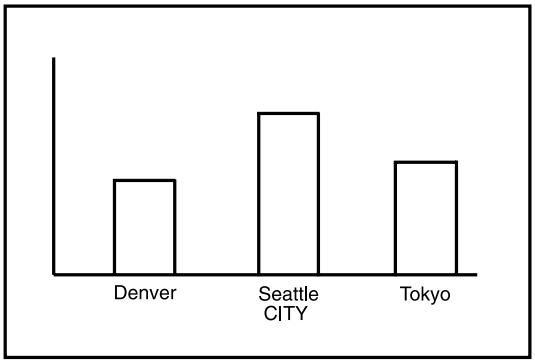
About Midpoints

Midpoints are the values of the chart variable that identify categories of data. By default, midpoints are selected or calculated by the procedure. The way the procedure handles the midpoints depends on whether the values of the chart variable are character, discrete numeric, or continuous numeric.

Character Values

A character chart variable generates a midpoint for each unique value of the variable. For example, if the chart variable CITY contains the names of three different cities, each city is a midpoint, resulting in three midpoints for the chart:

Figure 13.9 Character Midpoints



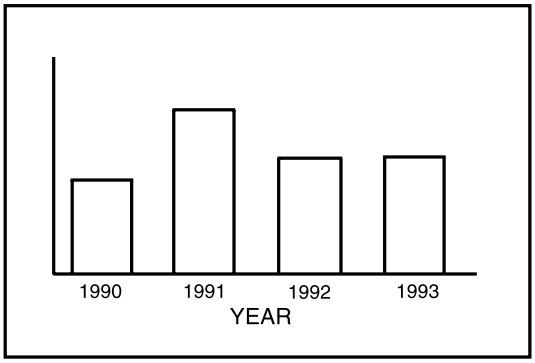
(In pie charts, midpoint values that compose a small percentage of the total for the chart may be placed in the OTHER slice and will not produce a separate midpoint.)

By default, character midpoints are arranged in alphabetic order. If a character variable has an associated format, the values are arranged in order of the formatted values.

Discrete Numeric Values

A numeric chart variable used with the DISCRETE option generates a midpoint for each unique value of the chart variable. For example, the numeric variable YEAR used with DISCRETE produces one midpoint for each year:

Figure 13.10 Discrete Numeric Midpoints



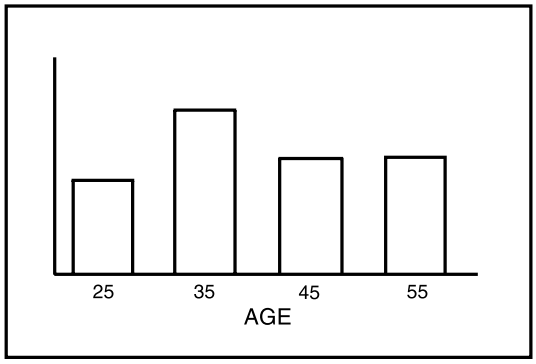
By default, numeric midpoints are arranged in ascending order. If the numeric variable has an associated format, each formatted value generates a separate midpoint. Formatted numeric variables are ordered by the unformatted numeric values.

Continuous Numeric Values

A continuous numeric variable generates midpoints that represent ranges of values. By default, the GCHART procedure determines the ranges, calculates the median value of each range, and displays the appropriate median value at each midpoint on the chart. A value that falls exactly halfway between two midpoints is placed in the higher range.

For example, the numeric variable AGE produces four midpoints, each of which represents a ten-year age range; the median value of the range is displayed at each midpoint:

Figure 13.11 Continuous Numeric Midpoints



By default, midpoints of ranges are arranged in ascending order.

Selecting and Ordering Midpoints

For character or discrete numeric values, you can use the MIDPOINTS= option to rearrange the midpoints or to exclude midpoints from the chart. For example, to change the default alphabetic order of the midpoints in Figure 13.9 on page 526, specify

```
midpoints='Tokyo' 'Denver' 'Seattle'
```

To exclude the midpoint for Denver, specify

```
midpoints='Tokyo' 'Seattle'
```

In this case, values excluded by the option are not included in the calculation of the chart statistic.

You can order or select discrete numeric midpoint values just as you do character values, but you omit the quotation marks when specifying numeric values.

For continuous numeric variables, use the LEVELS= or MIDPOINTS= option to change the number of midpoints, to control the range of values each midpoint represents, or to change the order of the midpoints. To control the range of values each midpoint represents, use the MIDPOINTS= option to specify the median value of each range. For example, to select the ranges 20-29, 30-39, and 40-49, specify

```
midpoints=25 35 45
```

Alternatively, to select the number of midpoints that you want and let the procedure calculate the ranges and medians, use the LEVELS= option.

You can also use formats to control the ranges of continuous numeric variables, but in that case the values are no longer continuous but discrete.

Note: You cannot use MIDPOINTS= to exclude continuous numeric values from the chart because values below or above the ranges specified by the option are automatically included in the first and last midpoints, respectively. To exclude continuous numeric values from a chart, use a WHERE statement in a DATA step or the WHERE= DATA set option. △

See also the description of the LEVELS= and MIDPOINTS= options for the appropriate statement.

About Chart Statistics

The *chart statistic* is the statistical value calculated for the chart variable and represented by each block, bar, or slice. The GCHART procedure calculates six chart statistics; the default statistic is frequency.

The examples given in the descriptions of these statistics assume a data set with two variables, CITY and SALES. The values of CITY are *Denver*, *Seattle*, and *Tokyo*. There are 21 observations: seven for Denver, nine for Seattle, and five for Tokyo.

Frequency

The frequency statistic is the total number of observations in the data set for each midpoint. For example, seven observations of the chart variable, CITY, contain the value *Denver*, so the frequency for the *Denver* midpoint is 7.

Cumulative Frequency

The cumulative frequency statistic adds the frequency for the current midpoint to the frequency of all of the preceding midpoints. For example, the frequency for the *Denver* midpoint is 7, and the frequency for the next midpoint, *Seattle*, is 9, so the cumulative frequency for *Seattle* is 16.

You cannot request cumulative frequency with the DONUT, PIE, PIE3D, or STAR statements.

Percentage

The percentage statistic is calculated by dividing the frequency for each midpoint by the total frequency count for all midpoints in the chart or group and multiplying it by 100. For example, the frequency count for the *Denver* midpoint is 7 and the total frequency count for the chart is 21, so the percentage statistic for *Denver* is 33.3%.

Cumulative Percentage

The cumulative percentage statistic adds the percentage for the current midpoint to the percentage for all of the preceding midpoints in the chart or group. For example, the percentage for the *Denver* midpoint is 33.3, and the percentage for the next midpoint, *Seattle*, is 42.9, so the cumulative percentage for *Seattle* is 76.2.

You cannot request cumulative percentage with the DONUT, PIE, PIE3D, or STAR statements.

Sum

The sum statistic is the total of the values for the SUMVAR= variable for each midpoint. For example, if you specify SUMVAR=SALES and the values of the SALES variable for the seven *Denver* observations are 8734, 982, 1504, 3207, 4502, 624, and 918, the sum statistic for the *Denver* midpoint is 20,471.

You must use the SUMVAR= option to specify the variable for which you want the sum statistic.

Mean

The mean statistic is the average of the values for the SUMVAR= variable for each midpoint. For example, if TYPE=MEAN and SUMVAR=SALES, the mean statistic for the *Denver* midpoint is 2924.42.

You must use the SUMVAR= option to specify the variable for which you want the mean statistic.

Calculating Weighted Statistics

By default, each observation is counted only once in the calculation of the chart statistic. To calculate weighted statistics in which an observation can be counted more than once, use the FREQ= option. This option identifies a variable whose values are used as a multiplier for the observation in the calculation of the statistic. If the value of the FREQ= variable is missing, 0, or negative, the observation is excluded from the calculation.

For example, to use a variable called COUNT to produce weighted statistics, assign FREQ=COUNT. This table shows how the values of COUNT would affect the statistic calculation:

When COUNT is...	the observation is counted this many times in the statistic calculation...
1	1
5	5
.	0
-3	0

If you use the SUMVAR= option, the SUMVAR= variable value for an observation is multiplied by the FREQ= variable value for the observation for use in calculating the chart statistic.

By default, the percentage and cumulative percentage statistics are calculated based on the frequency. If you want to chart a percentage or cumulative percentage based on a sum, you can use the FREQ= option to specify a variable to use for the "sum" calculation and specify the PCT statistic, as shown in this example:

```
freq=count type=pct
```

Because the variable that is used by FREQ= determines the number of times an observation is counted, the value of COUNT is the equivalent of the sum statistic.

See also the descriptions of the TYPE=, SUMVAR=, and FREQ= options for the action statements.

About Patterns

When a chart needs one or more patterns, the procedure uses either

- default patterns and outlines that are automatically generated by SAS/GRAPH or
- patterns, colors, and outlines that are defined by PATTERN statements, graphics options, and procedure options.

The following sections summarize pattern behavior for the GCHART procedure. For more information, see “PATTERN Statement” on page 211.

Default Patterns and Outlines

In general, the default pattern that the GCHART procedure uses is a solid fill that it rotates once through the colors list, skipping the foreground color. The procedure also outlines all areas in the foreground color. (Typically, the foreground color is the first color in the device's colors list.)

Specifically, the GCHART procedure uses default patterns and outlines when you

- do not specify *any* PATTERN statements, and
- do not use the CPATTERN= graphics option, and
- do not use the COLORS= graphics options (that is, you use the device's default colors list and it has more than one color), and
- do not use the COUTLINE= option in the action statement.

If all of these conditions are true, then the GCHART procedure

- selects the first default fill pattern, which is always solid, and rotates it through the colors list, generating one solid pattern for each color. If the first color in the device's colors list is black (or white), the procedure skips that color and begins generating patterns with the next color.
- uses the foreground color to outline every patterned area.

If the procedure needs additional patterns, GCHART selects the next default pattern fill that is appropriate to the type of chart and rotates it through the colors list, skipping the foreground color as before. The procedure continues in this fashion until it has generated enough patterns for the chart.

Changing any of these conditions may change or override the default behavior:

- If you specify a colors list with the COLORS= option in a GOPTIONS statement and the list contains more than one color, the procedure rotates the default solid pattern through that list, using every color, even if the foreground color is black (or white). The default outline color remains the foreground color.
- If you specify either COLORS=(*one-color*) or the CPATTERN= graphics option, the default fill pattern changes from solid to the list of appropriate hatch patterns. The procedure uses the specified color to generate one pattern definition for each hatch pattern in the list. The default outline color remains the foreground color.
- Whenever there are PATTERN definitions in effect, whether or not the GCHART procedure can use them, the default outline color for all patterns changes from foreground to SAME, as described in the following section.

For a description of these graphics options, see Chapter 9, "Graphics Options and Device Parameters Dictionary," on page 301.

User-defined Patterns and Outlines

You can use PATTERN statements to explicitly specify patterns, including color or fill type or both. Different types of charts require different types of pattern fills. For complete information on all pattern types, see "PATTERN Statement" on page 211. See also the section on controlling patterns and colors for each chart type.

When you use PATTERN statements, the procedure uses the specified patterns until all of the PATTERN definitions they generate have been used. Then, if more patterns are required, it returns to the default pattern rotation.

Whenever you specify *any* PATTERN statement, the default pattern outline changes. Instead of the foreground color, the outline color is the same as the fill color; for example, a blue bar has a blue outline. The effect is the same as specifying COUTLINE=SAME. Even when the procedure runs out of user-defined patterns and generates default patterns, the outlines continue to match the interior pattern color.

To change the outline color of any pattern, whether default or user-defined, use the COUTLINE= option in the action statement that generates the chart.

Version 6 Patterns

If you specify the V6COMP graphics option, the procedure generates patterns by rotating the appropriate Version 6 default patterns through all of the colors in the colors list. With V6COMP, all patterns are outlined in the same color as the fill.

Procedure Syntax

Requirements: At least one BLOCK, HBAR, HBAR3D, VBAR, VBAR3D, PIE, PIE3D, DONUT, or STAR statement is required.

Global statements: AXIS, FOOTNOTE, LEGEND, PATTERN, TITLE

Reminder: The procedure can include the BY, FORMAT, LABEL, and WHERE statements as well as the SAS/GRAPH NOTE statement.

Supports:

RUN-group processing

Output Delivery System (ODS)

PROC GCHART<DATA=*input-data-set*>
<ANNOTATE=*Annotate-data-set*>
<GOUT=<*libref.*>*output-catalog*>
<IMAGEMAP=*output-data-set*;>

BLOCK *chart-variable(s)* </ *option(s)*;>

HBAR | HBAR3D | VBAR | VBAR3D*chart-variable(s)* </ *option(s)*;>

PIE | PIE3D | DONUT *chart-variable(s)* </ *option(s)*;>

STAR *chart-variable(s)* </ *option(s)*;>

PROC GCHART Statement

Identifies the data set containing the chart variables. Optionally specifies annotation and an output catalog.

Requirements: An input data set is required.

Syntax

PROC GCHART<DATA=*input-data-set*>
<ANNOTATE=*Annotate-data-set*>
<GOUT=<*libref.*>*output-catalog*>
<IMAGEMAP=*output-data-set*;>

Options

PROC GCHART statement options affect all graphs produced by the procedure.

ANNOTATE=*Annotate-data-set*
ANNO=*Annotate-data-set*
specifies a data set to annotate all graphs that are produced by the GCHART procedure. To annotate individual graphs, use ANNOTATE= in the action statement.

See also: Chapter 10, "The Annotate Data Set," on page 403

DATA=*input-data-set*
specifies the SAS data set that contains the variable(s) to chart. By default, the procedure uses the most recently created SAS data set.

See also: "SAS Data Sets" on page 25 and "About Chart Variables" on page 525

GOUT=<*libref.*>*output-catalog*
specifies the SAS catalog in which to save the graphics output that is produced by the GCHART procedure. If you omit the libref, SAS/GRAPH looks for the catalog in the temporary library called WORK and creates the catalog if it does not exist.

See also: "Storing Graphics Output in SAS Catalogs" on page 49

IMAGEMAP=*output-data-set*
creates a SAS data set that contains information about the graph and about areas in the graph. This information includes the shape and coordinates of the areas and is used to build an HTML file that links the graph areas to other files or images. This linking provides drill-down functionality on the graph. The Imagemap data set also contains the information that is stored in the variables referenced by the HTML= and HTML_LEGEND= options. Therefore, in order to use IMAGEMAP= to create an HTML file, you must also use the HTML= option or the HTML_LEGEND= option or both.

See also: "Customizing Web Pages for Drill-down Graphs" on page 100

BLOCK Statement

Creates block charts in which the height of the blocks represents the value of the chart statistic for each category of data.

Requirements: At least one chart variable is required.
Global statements: LEGEND, PATTERN, TITLE, FOOTNOTE
Supports: Drill-down functionality

Description The BLOCK statement specifies the variable or variables that define the categories of data to chart. This statement automatically

- determines the midpoints
- calculates the chart statistic for each midpoint (the default is FREQ)
- scales the blocks according to the statistic value
- assigns patterns and colors to the block faces and the grid; the default block pattern is solid.

You can use statement options to select or order the midpoints (blocks), to change the type of chart statistic, and to modify the appearance of the chart. You can also specify additional variables by which to group, subgroup, or sum the data.

Block charts allow grouping, which organizes the blocks into rows based on the values of a group variable, and subgrouping, which subdivides the blocks into segments based on the values of a subgroup variable.

In addition, you can use global statements to modify the block patterns and the legend, as well as add titles, footnotes, and notes to the chart. You can also use an Annotate data set to enhance the chart.

Note: If you get a message that the chart is too large to display on your terminal or printer, try one or both of the following: △

- reduce the size of the character cells defined for the output device by specifying larger values for the HPOS= and VPOS= graphics options

- decrease the size of the chart text with the HTEXT= graphics option.

See "About the Graphics Output Area" on page 29 for a details.

Syntax

BLOCK *chart-variable(s)* </ *option(s)*;>

option(s) can be one or more options from any or all of the following categories:

- appearance options
  - ANNOTATE=*Annotate-data-set*
  - BLOCKMAX=*max-value*
  - CAXIS=*grid-color*
  - COUTLINE=*block-outline-color* | SAME
  - CTEXT=*text-color*
  - LEGEND=LEGEND<1...99>
  - NOHEADING
  - NOLEGEND
  - PATTERNID=BY |GROUP |MIDPOINT |SUBGROUP
  - WOUTLINE=*block-outline-width*
- midpoint options
  - DISCRETE
  - GROUP=*group-variable*
  - LEVELS=*number-of-midpoints*
  - MIDPOINTS=*value-list*
  - MIDPOINTS=OLD
  - MISSING
  - SUBGROUP=*subgroup-variable*
- statistic options
  - FREQ=*numeric-variable*
  - G100
  - SUMVAR=*summary-variable*
  - TYPE=*statistic*
- catalog entry description options
  - DESCRIPTION='*entry-description*'
  - NAME='*entry-name*'
- ODS options
  - HTML=*variable*
  - HTML_LEGEND=*variable*

Required Arguments

chart-variable(s)
specifies one or more variables that define the categories of data to chart. Each chart variable draws a separate chart. All variables must be in the input data set. Separate multiple chart variables with blanks. The values of a chart variable used with the BLOCK statement have a maximum length of 13.

See also: "About Chart Variables" on page 525

Options

Options in a BLOCK statement affect all graphs produced by that statement. You can specify as many options as you want and list them in any order. For details on specifying colors, see Chapter 7, "SAS/GRAPH Colors," on page 139. For a complete description of the graphics options, see Chapter 9, "Graphics Options and Device Parameters Dictionary," on page 301.

ANNOTATE=*Annotate-data-set*
ANNO=*Annotate-data-set*
specifies a data set to annotate charts produced by the BLOCK statement.

Note: Annotate coordinate systems 1, 2, 7, and 8 (data system coordinates) are not valid with block charts. △

See also: Chapter 10, "The Annotate Data Set," on page 403

BLOCKMAX=*max-value*
specifies the chart statistic value of the tallest block on the chart. This option lets you produce a series of block charts using the same scale. All blocks are rescaled as if *max-value* were the maximum value on the chart.

CAXIS=*grid-color*
specifies the color for the midpoint grid. By default, the midpoint grid uses the foreground color (usually the first color in the colors list).

Featured in: Example 2 on page 584

COUTLINE=*block-outline-color* | SAME
outlines all blocks or all block segments and legend values in the subgroup legend (if it appears) using the specified color. SAME specifies that the outline color of a block or a block segment or a legend value is the same as the interior pattern color.

The default outline color depends on the PATTERN statement:

- If you do not specify the PATTERN statement, the default outline color is the foreground color (usually the first color in the colors list).
- If you specify the PATTERN statement or the V6COMP graphics option, the default is COUTLINE=SAME.

Note: If you specify empty patterns, (VALUE=EMPTY in a PATTERN statement) you should not change the outline color from the default value, SAME, to a single color. Otherwise all the outlines will be one color and you will not be able to distinguish between the empty areas. △

See also: "Controlling Block Chart Patterns and Colors" on page 539 and "About Patterns" on page 529

Featured in: Example 2 on page 584

CTEXT=*text-color*
specifies a color for all text on the chart. Text includes the values and labels for the midpoint grid, the subgroup legend, and the descriptive statistic values. If you omit CTEXT=, PROC GCHART searches for a color specification in this order:

1. the CTEXT= option in a GOPTIONS statement
2. the first color in the colors list (the default).

CTEXT= is overridden by the COLOR= suboption of the LABEL= or VALUE= option in a LEGEND definition assigned to the subgroup legend. The suboption

determines the color of the legend label or the color of the legend value descriptions, respectively.

DESCRIPTION='*entry-description*'
DES='*entry-description*'
specifies the description of the catalog entry for the chart. The maximum length for *entry-description* is 40 characters. The description does not appear on the chart. By default, the GCHART procedure assigns a description of the form BLOCK CHART OF *variable*, where *variable* is the name of the chart variable.

The *entry-description* can include the #BYLINE, #BYVAL, and #BYVAR substitution options, which work as they do when used on TITLE, FOOTNOTE, and NOTE statements. For more information, refer to the description of the options on page 262, and "Substituting BY Line Values in a Text String" on page 266. The 40-character limit applies before the substitution takes place for these options; thus, if in the SAS program the entry-description text exceeds 40 characters, it is truncated to 40 characters, and then the substitution is performed.

The descriptive text is shown in the "description" portion of each of the following:

- in the Results window
- among the catalog-entry properties that you can view from the Explorer window
- in the Table of Contents that is generated when you use CONTENTS= on an ODS HTML statement (see "Linking to Output through a Table of Contents" on page 86), assuming the GCHART output is generated while the contents page is open
- in the Description field of the PROC GREPLAY window

DISCRETE
treats a numeric chart variable as a discrete variable rather than as a continuous variable. The GCHART procedure creates a separate midpoint and, hence, a separate grid square and block for each unique value of the chart variable. If the chart variable has a format associated with it, each formatted value is treated as a midpoint.

The LEVELS= option is ignored when you use DISCRETE. The MIDPOINTS= option overrides DISCRETE.

FREQ=*numeric-variable*
specifies a variable whose values weight the contribution of each observation in the computation of the chart statistic. Each observation is counted the number of times specified by the value of *numeric-variable* for that observation. If the value of *numeric-variable* is missing, 0, or negative, the observation is not used in the statistic calculation. Noninteger values of *numeric-variable* are truncated to integers.

FREQ= is valid with all chart statistics.

Because you cannot use the PERCENT, CPERCENT, FREQ, or CFREQ statistics with the SUMVAR= option, you must use the FREQ= option to calculate percentages, cumulative percentages, frequencies, or cumulative frequencies based on a sum.

See also: "Calculating Weighted Statistics" on page 529

G100
calculates the percentage and cumulative percentage statistics separately for each group. When you use G100, the individual percentages reflect the contribution of the midpoint to the group and total 100 percent for each group. G100 is ignored unless you also use the GROUP= option.

By default, the individual percentages reflect the contribution of the midpoint to the entire chart and total 100 percent for the entire chart.

GROUP=*group-variable*
organizes the data according to the values of *group-variable*. *Group-variable* can be either character or numeric and is always treated as a discrete variable.

GROUP= produces a group grid that contains a separate row of blocks for each unique value of the group variable. Each row contains a square for each midpoint. The groups are arranged from front to back in ascending order of the group variable values. These values are printed to the left of each row; the group variable name or label is printed above the list of group values.

By default, each group includes all midpoints, even if no observations for the group fall within the midpoint range. Missing values for *group-variable* are treated as a valid group.

Featured in: Example 2 on page 584

HTML=*variable*
identifies the variable in the input data set whose values create links in the HTML file that is created by the ODS HTML statement. These links are associated with an area of the chart and point to the data or graph you wish to display when the user drills down on the area.

HTML_LEGEND=*variable*
identifies the variable in the input data set whose values create links in the HTML file that is created by the ODS HTML statement. These links are associated with a legend value and point to the data or graph that you wish to display when the user drills down on the value.

LEGEND=LEGEND<1...99>
assigns the specified LEGEND definition to the legend generated by the SUBGROUP= option. The LEGEND= option itself does *not* generate a legend.

LEGEND= is ignored if

- SUBGROUP= is not used.
- the specified LEGEND definition is not in effect.
- the NOLEGEND option is used.
- the PATTERNID= option is set to any value other than SUBGROUP; that is, the value of PATTERNID= is BY or GROUP or MIDPOINT.

To create a legend based on the chart midpoints instead of the subgroups, use the chart variable as the subgroup variable:

```
block city / subgroup=city;
```

See also: "LEGEND Statement" on page 187 and SUBGROUP= on page 538

Featured in: Example 2 on page 584

LEVELS=*number-of-midpoints*
specifies the number of midpoints for the numeric chart variable. The range for each midpoint is calculated automatically using the algorithm described in Terrell and Scott (1985). LEVELS= is ignored if

- the chart variable is character type.
- the DISCRETE option is used.
- the MIDPOINTS= option is used.

MIDPOINTS=*value-list*
specifies the midpoint values for the blocks. The way you specify *value-list* depends on the type of variable:

- *For numeric chart variables*, *value-list* is either an explicit list of values, or a starting and an ending value with an interval increment, or a combination of both forms:

  n <...*n*>

  n TO *n* <BY *increment*>

  n <...*n*> TO *n* <BY *increment*> <*n* <...*n*>>

  If a numeric variable has an associated format, the specified values must be the *unformatted* values.

  By default, numeric variable values are treated as continuous (if you omit the DISCRETE option), and

  - the lowest midpoint consolidates all data points from negative infinity to the median of the first two midpoints
  - the highest midpoint consolidates all data points from the median of the last two midpoints up to infinity
  - all other values in *value-list* specify the median of a range of values, and the GCHART procedure calculates the midpoint values.

  If you include the DISCRETE option, each value in *value-list* specifies a unique numeric value.

- *For character chart variables*, *value-list* is a list of unique character values enclosed in quotation marks and separated by blanks:

  'value-1' <...*'value-n'*>

  If a character variable has an associated format, the specified values must be the *formatted* values.

For a complete description of *value-list*, see the ORDER= on page 168 option in the AXIS statement.

If *value-list* for either type of variable specifies so many midpoints that the axis values overwrite each other, the values may be unreadable. In this case the procedure writes a warning to the SAS log. On many devices, you can correct crowded values by increasing the number of cells in your graphics display using the HPOS= and VPOS= graphics options.

See also: "About Midpoints" on page 525

Featured in: Example 2 on page 584

MIDPOINTS=OLD
tells the GCHART procedure to calculate default midpoints using the algorithm used in Release 82.4 and Version 5 of SAS/GRAPH software. The MIDPOINTS=OLD option is ignored unless the chart variable is numeric.

MISSING
accepts a missing value as a valid midpoint for the chart variable. By default, observations with missing values are ignored. Missing values are always valid for the group and subgroup variables.

NAME='*entry-name*'
specifies the name of the catalog entry for the graph. The maximum length for *entry-name* is eight characters. The default name is GCHART. If the name duplicates an existing entry name, SAS/GRAPH software adds a number to the duplicate name to create a unique name– for example, GCHART1.

NOHEADING
suppresses the heading describing the type of statistic. By default the heading is printed at the top of each block chart.

Featured in: Example 2 on page 584

NOLEGEND
suppresses the legend automatically generated by the SUBGROUP= option. NOLEGEND is ignored if the SUBGROUP= option is not used.

PATTERNID=BY | GROUP | MIDPOINT | SUBGROUP
specifies the way fill patterns are assigned. By default, PATTERNID=SUBGROUP. Values for PATTERNID= are as follows:

BY
changes patterns each time the value of the BY variable changes. All blocks use the same pattern if the GCHART procedure does not include a BY statement.

GROUP
changes patterns every time the value of the group variable changes. All blocks in each group (row) use the same pattern, but a different pattern is used for each group.

MIDPOINT
changes patterns every time the midpoint value changes. If you use the GROUP= option, the respective midpoint patterns are repeated for each group.

SUBGROUP
changes patterns every time the value of the subgroup variable changes. The blocks must be subdivided by the SUBGROUP= option for the SUBGROUP value to have an effect. Without SUBGROUP=, all block faces have the same pattern.

Note: If you use the SUBGROUP= option and specify a PATTERNID= value other than SUBGROUP, the block segments use the same pattern and are indistinguishable. △

See also: "Controlling Block Chart Patterns and Colors" on page 539

Featured in: Example 7 on page 596

SUBGROUP=*subgroup-variable*
divides the blocks into segments according to the values of *subgroup-variable*. *Subgroup-variable* can be either character or numeric and is always treated as a discrete variable. SUBGROUP= creates a separate segment within each block for every unique value of the subgroup variable for that midpoint.

If PATTERNID=SUBGROUP (the default setting), each segment is filled with a different pattern, and a legend providing a key to the patterns is automatically generated. If the value of PATTERNID= is anything other than SUBGROUP, the segments are all the same color, the legend is suppressed, and the subgrouping effect is lost.

By default the legend appears at the bottom of the chart. To modify the legend, assign a LEGEND definition with the LEGEND= option. To suppress the legend, specify NOLEGEND.

See also: "LEGEND Statement" on page 187

Featured in: Example 2 on page 584

SUMVAR=*summary-variable*
specifies a numeric variable for sum or mean calculations. The GCHART procedure calculates the sum or, if requested, the mean of *numeric-variable* for each midpoint. The resulting statistics are represented by the height of the blocks in each square.

The values of a summary variable used with the BLOCK statement have a maximum length of 8.

When you use SUMVAR=, the TYPE= option value must be either SUM or MEAN. With SUMVAR=, the default is TYPE=SUM.

Featured in: Example 1 on page 582

TYPE=*statistic*
specifies the chart statistic.

- If the SUMVAR= option is not used, *statistic* can be one of the following:

  FREQ
  frequency (the default)

  CFREQ
  cumulative frequency

  PERCENT PCT
  percentage

  CPERCENT CPCT
  cumulative percentage

- If SUMVAR= is used, *statistic* can be either:

  SUM
  sum (the default)

  MEAN
  mean

Because you cannot specify the statistics PERCENT, CPERCENT, FREQ, or CFREQ in conjunction with SUMVAR=, you must use FREQ= to calculate percentages, cumulative percentages, frequencies, or cumulative frequencies based on a sum. See also "Calculating Weighted Statistics" on page 529.

If you specify TYPE=MEAN and use the SUBGROUP= option, the height of the block represents the mean for the entire midpoint. The subgroup segments are proportional to the subgroup's contribution to the sum for the block.

See also: "About Chart Statistics" on page 528

Featured in: Example 2 on page 584

WOUTLINE=*block-outline-width*
specifies the width of the block outline in pixels.

Controlling Block Chart Patterns and Colors

Default patterns and outlines In a block chart, only the front faces of the blocks display patterns. By default, the procedure

- fills the block faces with bar/block patterns, beginning with the default fill, SOLID, and rotating it through the colors list. When the solid patterns are exhausted, the procedure selects the next default bar/block pattern and rotates it through the colors list. It continues in this fashion until all of the required patterns have been assigned.

  If you use the device's default colors and the first color in the list is either black or white, the procedure does not create a pattern in that color. If you specify a colors list with the COLORS= graphics option, the procedure uses all of the colors in the list to generate the patterns.
- outlines blocks and block segments using the first color in the colors list.
- colors the midpoint grid with the first color in the colors list.

See “About Patterns” on page 529 for more information on how the GCHART procedure assigns default patterns and outlines.

User-defined patterns To override the default patterns and select fills and colors for the blocks or block segments, use the PATTERN statement. Only bar/block patterns are valid; all other pattern fills are ignored. For a complete description of all bar/block patterns, see VALUE= on page 214 in “PATTERN Statement” on page 211.

Whenever you use PATTERN statements, the default pattern outline color changes to SAME. That is, the outline color is the same as the fill color. To specify the outline color, use the COUTLINE= on page 534 option.

When patterns change The PATTERNID= option controls when the pattern changes. By default, PATTERNID=SUBGROUP. Therefore, when you use the SUBGROUP= option to subdivide the blocks, the pattern automatically changes each time the subgroup value changes, and each subdivision of the block displays a different pattern. As a result, the number of values for the SUBGROUP= variable determines the number of block patterns on the chart. If you do not subdivide the blocks, all blocks use the same pattern.

Instead of changing the pattern for each subgroup, you can change the pattern for each midpoint, each group, or each BY group, by changing the value of PATTERNID=. See the PATTERNID= on page 538 option for details.

Axis color By default, axis elements use the first color in the colors list. To change the grid color, use the CAXIS= option. To change the axis text color, use the CTEXT= option.

Controlling Block Chart Text

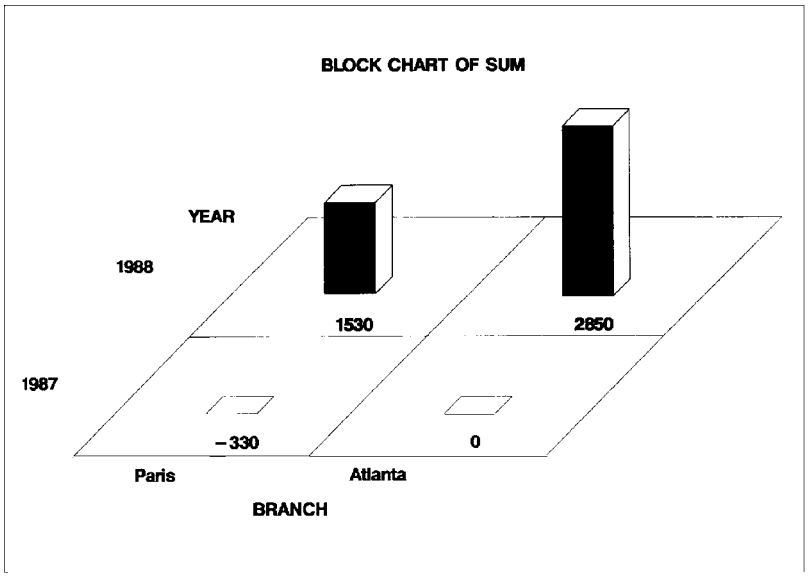
To control the font and size of text on the chart, use the FTEXT= and HTEXT= graphics options. See Chapter 9, “Graphics Options and Device Parameters Dictionary,” on page 301 for a description of these options.

Because block charts do not use AXIS statements, you must use a LABEL statement instead to suppress the label for the midpoint variable. See Example 2 on page 584.

Displaying Negative or Zero Values

The relative block heights in the chart represent the scaled value of the chart statistic value for the midpoint. If the statistic has a value of 0 or, in the case of sum and mean, a negative value, the base of the block is drawn in the square for the corresponding midpoint. Figure 13.12 on page 541 shows an example of a chart with 0 and negative statistic values.

Figure 13.12 Block Chart with 0 and Negative Statistic Values



HBAR, HBAR3D, VBAR, and VBAR3D Statements

Create horizontal or vertical bar charts in which the length or height of the bars represents the value of the chart statistic for each category of data.

Requirements: At least one chart variable is required.

Global statements: AXIS, LEGEND, PATTERN, TITLE, FOOTNOTE

Supports: Drill-down functionality

Description The HBAR, HBAR3D, VBAR, and VBAR3D statements specify the variable or variables that define the categories of data to chart. These statements automatically

- determine the midpoints
- calculate the chart statistic for each midpoint (the default is FREQ)
- scale the response axis and the bars according to the statistic value
- determine bar width and spacing
- assign patterns to the bars; the default bar/block pattern is SOLID.
- draw a frame around the axis area using the first color in the colors list.

You can use statement options to select or order the midpoints (bars), to control the tick marks on the response axis, to change the type of chart statistic, to display specific statistics, and to modify the appearance of the chart. You can also specify additional variables by which to group, subgroup, or sum the data.

All bar charts allow grouping, which uses an additional category to organize the bars into groups, and subgrouping, which divides the bars into segments.

In addition, you can use global statements to modify the axes (including requesting a logarithmic axis), the bar patterns, and the legend, as well as add titles, footnotes, and notes to the chart. You can also use an Annotate data set to enhance the chart.

Syntax

HBAR | HBAR3D | VBAR | VBAR3D *chart-variable(s)* </ *option(s)*; >

option(s) can be one or more options from any or all of the following categories:

- appearance options
  - ANNOTATE=*Annotate-data-set*
  - CAXIS=*axis-color*
  - CERROR=*error-bar-color*
  - CFRAME=*background-color*
  - COUTLINE=*bar-outline-color* | SAME
  - CTEXT=*text-color*
  - FRAME | NOFRAME
  - GSPACE=*group-spacing*
  - LEGEND=LEGEND<1...99>
  - NOLEGEND
  - PATTERNID=BY | GROUP | MIDPOINT | SUBGROUP
  - SHAPE=*3D-bar-shape* (HBAR3D and VBAR3D only)
  - SPACE=*bar-spacing*
  - WIDTH=*bar-width*
  - WOUTLINE=*bar-outline-width*
- statistic options
  - CFREQ
  - CFREQLABEL='*column-label*' (HBAR and HBAR3D only)
  - CLM=*confidence-level*
  - CPERCENT
  - CPERCENTLABEL='*column-label*' (HBAR and HBAR3D only)
  - ERRORBAR=BARS | BOTH | TOP
  - FREQ
  - FREQLABEL='*column-label*' (HBAR and HBAR3D only)
  - FREQ=*numeric-variable*
  - G100
  - INSIDE=*statistic* (VBAR and VBAR3D only)
  - MEAN
  - MEANLABEL='*column-label*' (HBAR and HBAR3D only)
  - NOSTATS (HBAR and HBAR3D only)
  - OUTSIDE=*statistic* (VBAR and VBAR3D only)
  - PERCENT
  - PERCENTLABEL='*column-label*' (HBAR and HBAR3D only)
  - SUM
  - SUMLABEL='*column-label*' (HBAR and HBAR3D only)
  - SUMVAR=*summary-variable*
  - TYPE=*statistic*
- midpoint options
  - DISCRETE
  - GROUP=*group-variable*

LEVELS=*number-of-midpoints*
MIDPOINTS=*value-list*
MIDPOINTS=OLD
MISSING
SUBGROUP=*subgroup-variable*

- axes options
  ASCENDING
  AUTOREF
  AXIS=AXIS<1...99>
  CLIPREF
  DESCENDING
  FRONTREF
  GAXIS=AXIS<1...99>
  MAXIS=AXIS<1...99>
  MINOR=*number-of-minor-ticks*
  NOAXIS
  NOBASEREF
  NOZERO
  RAXIS=*value-list* | AXIS<1...99>
  REF=*value-list*
- catalog entry description options
  DESCRIPTION='*entry-description*'
  NAME='*entry-name*'
- ODS options
  HTML=*variable*
  HTML_LEGEND=*variable*

Required Arguments

chart-variable(s)
specifies one or more variables that define the categories of data to chart. Each chart variable draws a separate chart. All variables must be in the input data set. Multiple chart variables must be separated with blanks.

See also: "About Chart Variables" on page 525

Options

Options in an HBAR, HBAR3D, VBAR, or VBAR3D statement affect all graphs that are produced by that statement. You can specify as many options as you want and list them in any order. For details on specifying colors, see Chapter 7, "SAS/GRAPH Colors," on page 139. For a complete description of the graphics options, see Chapter 9, "Graphics Options and Device Parameters Dictionary," on page 301.

ANNOTATE=*Annotate-data-set*
ANNO=*Annotate-data-set*
specifies a data set to annotate charts produced by the bar chart statement.

See also: Chapter 10, "The Annotate Data Set," on page 403

ASCENDING
arranges the bars in ascending order of the value of the chart statistic. By default, bars are arranged in ascending order of midpoint value, without regard to the lengths of the bars. ASCENDING reorders the bars from shortest to longest. In horizontal bar charts the ordering is top to bottom; in vertical bar charts the ordering is left to right.

If you also use the GROUP= option, the reordering is performed separately for each group, so the order of the midpoints may be different for each group.

ASCENDING overrides any midpoint order specified with the MIDPOINTS= option or specified in the ORDER= option in an AXIS statement assigned to the midpoint axis.

AUTOREF
draws a reference line at each major tick mark on the response axis. To draw reference lines at specific points on the response axis, use the REF= option.

By default, reference lines in 2D bar charts are drawn in front of the bars. To draw reference lines behind the bars, use the CLIPREF option.

By default, reference lines in 3D bar charts are drawn on the back plane of the axis. To draw reference lines in front of the bars, use the FRONTREF option.

Featured in: Example 5 on page 591

AXIS=AXIS<1...99>
See RAXIS= on page 555.

CAXIS=*axis-color*
specifies a color for the response and midpoint axis lines and for the default axis area frame. If you omit the CAXIS= option, PROC GCHART searches for a color specification in this order:

1. the COLOR= option in AXIS definitions
2. the first color in the colors list (the default).

CFRAME=*background-color*
CFR=*background-color*
specifies the color with which to fill the axis area (in 2D bar charts) or the 3D back plane (in 3D bar charts).

The axis area color does not affect the frame color, which is always the same as the midpoint axis line color and controlled by the CAXIS= option. By default, the axis area in 2D bar charts is not filled.

CFRAME= is overridden by the NOFRAME option.

Note: If the background color, the bar color, and the outline color are the same, you may not be able to distinguish the bars. △

Featured in: Example 4 on page 588

CERROR=*error-bar-color*
specifies the color of error bars in bar charts. The default is the color of the response axis, which is controlled by the CAXIS= option.

CFREQ
displays the cumulative frequency statistic in the table of statistics. Default statistics are suppressed when you request specific statistics but all requested statistics are displayed.

CFREQLABEL='*column-label*' (HBAR and HBAR3D only)
specifies the text of the column label for the CFREQ statistic in the table of statistics. *Column-label* can be up to 32 characters long, but a single line of the label can be no more than 24 characters. By default, a label with more than one word will break as

close to the center of the line as possible. A double space in the string forces a line break.

CLIPREF
clips the reference lines at the bars. This makes the reference lines appear to be behind the bars. Because CLIPREF is the default for 3D bar charts, it affects only 2D charts.

Featured in: Example 5 on page 591

CLM=*confidence-level*
draws on a horizontal or vertical bar chart confidence intervals (error bars) with the specified percentage confidence level. Values for *confidence-level* must be greater than or equal to 50 and strictly less than 100. The default is 95. By default, CLM= draws the intervals using See ERRORBAR= for details on how error bars are computed and drawn.

Featured in: Example 6 on page 595

COUTLINE=*bar-outline-color* | SAME
outlines all bars or bar segments and legend values in the subgroup legend (if it appears) using the specified color. SAME specifies that the outline color of a bar or a bar segment or a legend value is the same as the interior pattern color.

The default outline color depends on the PATTERN statement:

- If you do not specify a PATTERN statement, the default outline color is the foreground color (the first color in the colors list).
- If you specify the PATTERN statement or the V6COMP graphics option, the default is COUTLINE=SAME.

Note: For 2D bar charts, if you specify empty patterns, (VALUE=EMPTY in a PATTERN statement) you should not change the outline color from the default value, SAME, to a single color. Otherwise all the outlines will be one color and you will not be able to distinguish between the empty areas. △

COUTLINE= is not valid when SHAPE=CYLINDER.

See also: "Controlling Bar Chart Patterns and Colors" on page 558 and "About Patterns" on page 529

Featured in: Example 3 on page 586, Example 5 on page 591 and Example 6 on page 595

CPERCENT
CPCT
displays the cumulative percentage statistic in the table of statistics. Default statistics are suppressed when you request specific statistics but all requested statistics are displayed.

CPERCENTLABEL='*column-label*' (HBAR and HBAR3D only)
specifies the text of the column label for the CPERCENT statistic in the table of statistics. *Column-label* can be up to 32 characters long, but a single line of the label can be no more than 24 characters. By default, a label with more than one word will break as close to the center of the line as possible. A double space in the string forces a line break.

CTEXT=*text-color*
specifies the color of all text on the chart that is not otherwise assigned a color. Text includes axis values and axis labels in the response, midpoint, and group axes; the

subgroup legend; and the displayed statistics. If you omit CTEXT=, PROC GCHART searches for a color specification in this order:

1. the CTEXT= option in a GOPTIONS statement
2. the first color in the colors list (the default).

CTEXT= overrides the color specification for the axis label and the tick mark values in the COLOR= option in an AXIS definition assigned to an axis.

CTEXT= is overridden by

- the COLOR= suboption of the LABEL= or VALUE= option in a LEGEND definition assigned to the subgroup legend. In this case the suboption determines the color of the legend label or the color of the legend value descriptions, respectively.
- the COLOR= suboption of a LABEL= or VALUE= option in an AXIS definition assigned to an axis. In this case the suboption determines the color of the axis label or the color of the tick mark values, respectively.

DESCENDING

arranges the bars in descending order of the value of the chart statistic. By default, bars are arranged in ascending order of midpoint value, without regard to the lengths of the bars. DESCENDING reorders the bars from longest to shortest. In horizontal bar charts the ordering is top to bottom; in vertical bar charts the ordering is left to right. If you also use the GROUP= option, the reordering is performed separately for each group, so the order of the midpoints may be different for each group.

DESCENDING overrides any midpoint order that is specified with the MIDPOINTS= option or that is specified in the ORDER= option in an AXIS statement assigned to the midpoint axis.

DESCRIPTION='*entry-description*'
DES='*entry-description*'

specifies the description of the catalog entry for the chart. The maximum length for *entry-description* is 40 characters. The description does not appear on the chart. By default, the GCHART procedure assigns a description of the form HBAR CHART OF *variable*, where *variable* is the name of the chart variable.

The *entry-description* can include the #BYLINE, #BYVAL, and #BYVAR substitution options, which work as they do when used on TITLE, FOOTNOTE, and NOTE statements. For more information, refer to the description of the options on page 262, and "Substituting BY Line Values in a Text String" on page 266. The 40-character limit applies before the substitution takes place for these options; thus, if in the SAS program the entry-description text exceeds 40 characters, it is truncated to 40 characters, and then the substitution is performed.

The descriptive text is shown in the "description" portion of each of the following:

- in the Results window
- among the catalog-entry properties that you can view from the Explorer window
- in the Table of Contents that is generated when you use CONTENTS= on an ODS HTML statement (see "Linking to Output through a Table of Contents" on page 86), assuming the GCHART output is generated while the contents page is open
- in the Description field of the PROC GREPLAY window

Featured in: Example 7 on page 596

DISCRETE

treats a numeric chart variable as a discrete variable rather than as a continuous variable. The GCHART procedure creates a separate midpoint and, hence, a separate

bar for each unique value of the chart variable. If the chart variable has a format associated with it, each formatted value is treated as a midpoint.

The LEVELS= option is ignored when you use DISCRETE. The MIDPOINTS= option overrides DISCRETE. The ORDER= option in an AXIS statement that is assigned to the midpoint axis can rearrange or exclude discrete midpoint values.

Featured in: Example 7 on page 596

ERRORBAR=BARS | BOTH | TOP

draws confidence intervals on a horizontal or vertical bar chart for either of the following:

- the mean of the SUMVAR= variable for each midpoint if you specify TYPE=MEAN
- the percentage of observations assigned to each midpoint if you specify TYPE=PCT with no SUMVAR= option.

The ERRORBAR= option cannot be used with values of the TYPE= option other than MEAN or PCT.

Valid values for ERRORBAR= are:

BARS
draws error bars as bars half the width of the main bars.

BOTH
draws error bars as two ticks joined by a line (default).

TOP
draws the error bar as a tick for the upper confidence limit that is joined to the top of the bar by a line.

By default, ERRORBAR= uses a confidence level of 95 percent. You can specify different confidence levels with the CLM= option.

When you use ERRORBAR= with TYPE=PCT, the confidence interval is based on a normal approximation. Let TOTAL be the total number of observations, and PCT be the percentage assigned to a given midpoint. The standard error of the percentage is approximated as

```
APSTDERR=100 * SQRT((PCT/100) * (1--(PCT/100)) / TOTAL);
```

Let LEVEL be the confidence level specified using the CLM= option, with a default value of 95. The upper confidence limit for the percentage is computed as

```
UCLP = PCT + APSTDERR * PROBIT( 1-(1-LEVEL/100)/2 );
```

The lower confidence limit for the percentage is computed as

```
LCLP = PCT - APSTDERR * PROBIT( 1-(1-LEVEL/100)/2 );
```

When you use ERRORBAR= with TYPE=MEAN, the sum variable must have at least two non-missing values for each midpoint. If the GROUP= option is used, each midpoint within a group must also have two non-missing values. Let N be the number of observations assigned to a midpoint, MEAN be the mean of those observations, and STD be the standard deviation of the observations. The standard error of the mean is computed as

```
STDERR = STD / SQRT(N);
```

Let LEVEL be the confidence level specified using the CLM= option, with a default value of 95. The upper confidence limit for the mean is computed as

```
UCLM = MEAN + STDERR * TINV( 1-(1-LEVEL/100)/2, N-1);
```

The lower confidence limit for the mean is computed as

```
LCLM = MEAN - STDERR * TINV( 1-(1-LEVEL/100)/2, N-1);
```

If you want the error bars to represent a given number C of standard errors instead of a confidence interval, and if the number of observations assigned to each midpoint is the same, then you can find the appropriate value for the CLM= option by running a DATA step. For example, if you want error bars that represent one standard error (C=1) with a sample size of N , you can run the following DATA step to compute the appropriate value for the CLM= option and assign that value to a macro variable &LEVEL:

```
data null;
c = 1;
n = 10;
level = 100 * (1 - 2 * (1 - probt( c, n-1)));
put all;
call symput('level',put(level,best12.));
run;
```

Then when you run the GCHART procedure, you can specify CLM=&LEVEL.

Note that this trick does not work precisely if different midpoints have different numbers of observations. However, choosing an average value for N may yield sufficiently accurate results for graphical purposes if the sample sizes are large or do not vary much.

Featured in: Example 6 on page 595

FRAME | NOFRAME
FR | NOFR

specifies whether the 2D axis area frame or the 3D backplane is drawn. The default is FRAME, which draws a frame around the axis area (in 2D bar charts) or generates a colored 3D backplane (in 3D bar charts).

The color of the frame or of the backplane outline is the color of the midpoint axis, which is determined by the CAXIS= option. To fill the 2D axis area with a background color, or to specify a color for the 3D backplane, use the CFRAME= option.

With 2D bar charts, NOFRAME removes the axis frame. It overrides CFRAME= and also removes the colored fill it specifies.

With 3D bar charts, NOFRAME removes the colored backplane, leaving the vertical and horizontal axis planes and axes. To remove these planes, use the NOPLANE option in the AXIS statement. To remove one or more axis elements, use either the AXIS statement or the NOAXIS option.

If the V6COMP graphics option is in effect, the default is NOFRAME.

Featured in: Example 7 on page 596 and Example 6 on page 595

FREQ

displays the frequency statistic in the table of statistics. Default statistics are suppressed when you request specific statistics but all requested statistics are displayed.

Featured in: Example 5 on page 591

FREQLABEL='*column-label*' (HBAR and HBAR3D only)

specifies the text of the column label for the FREQ statistic in the table of statistics. *column-label* can be up to 32 characters long, but a single line of the label can be no more than 24 characters. By default, a label with more than one word will break as close to the center of the line as possible. A double space in the string forces a line break.

Featured in: Example 5 on page 591 and Example 6 on page 595

FREQ=*numeric-variable*
specifies a variable whose values weight the contribution of each observation in the computation of the chart statistic. Each observation is counted the number of times that is specified by the value of *numeric-variable* for that observation. If the value of *numeric-variable* is missing, 0, or negative, the observation is not used in the statistic calculation. Noninteger values of *numeric-variable* are truncated to integers. FREQ= is valid with all chart statistics.

Because you cannot use TYPE=PERCENT, TYPE=CPERCENT, TYPE=FREQ, or TYPE=CFREQ with the SUMVAR= option, you must use FREQ= to calculate percentages, cumulative percentages, frequencies, or cumulative frequencies based on a sum.

See also: "Calculating Weighted Statistics" on page 529

FRONTREF
specifies that reference lines drawn by the AUTOREF or REF= options should be drawn in front of the bars. By default, reference lines in 3D bar charts are drawn on the back plane of the axis. Because FRONTREF is the default for 2D bar charts, it affects only 3D charts.

G100
calculates the percentage and cumulative percentage statistics separately for each group. When you use G100, the individual percentages reflect the contribution of the midpoint to the group and total 100 percent for each group. G100 is ignored unless you also use the GROUP= option.

By default, the individual percentages reflect the contribution of the midpoint to the entire chart and total 100 percent for the entire chart.

GAXIS=AXIS<1...99>
assigns the specified AXIS definition to the group axis. (A group axis is created when you use the GROUP= option.) You can use the AXIS definition to modify the order of the groups, the text of the labels, and appearance of the axis. GAXIS= is ignored if the specified AXIS definition does not exist.

The AXIS statement options MAJOR= and MINOR= are ignored in AXIS definitions assigned to the group axis because the axis does not use tick marks. A warning message is written to the SAS log if these options appear in the AXIS definition.

To remove groups from the chart, use the ORDER= option in the AXIS statement.

To suppress the brackets drawn around the values on the group axis in vertical bar charts, use the NOBRACKETS option in the AXIS statement.

See also: "AXIS Statement" on page 162

Featured in: Example 7 on page 596

GROUP=*group-variable*
organizes the data according to values of *group-variable*. *Group-variable* can be either character or numeric and is always treated as a discrete variable.

GROUP= produces a separate group of bars for each unique value of the group variable. Missing values for *group-variable* are treated as a valid group. The groups are arranged in ascending order of the group variable values.

By default, each group includes all midpoints, even if no observations for the group fall within the midpoint range, meaning that no bar is drawn at the midpoint. Use the NOZERO option to suppress midpoints with no observations.

GROUP= also produces a *group axis* that lists the values that distinguish the groups. The group axis has no axis line but displays the group variable name or label. To modify the group axis, assign an AXIS definition with the GAXIS= option.

In horizontal bar charts, the group axis is to the left of the midpoint axis and the groups are arranged from top to bottom, starting with the lowest value at the top.

In vertical bar charts, the group axis is below the midpoint axis and the groups are arranged from left to right starting with the lowest value at the left. If the group label in a vertical bar chart is narrower than all the bars in the group, brackets are added to the label to emphasize which bars belong in each group. Group brackets are not displayed if the space between the group values is less than one and one-half character cells. Use the NOBRACKETS option in the AXIS statement to suppress the group brackets.

Featured in: Example 7 on page 596

GSPACE=*group-spacing*
specifies the amount of extra space between groups of bars. *Group-space* can be any non-negative number. Units are character cells. Use GSPACE=0 to leave no extra space between adjacent groups of bars. In this case, the same space appears between groups of bars as between the bars in the same group.

GSPACE= is ignored unless you also use the GROUP= option. By default, the GCHART procedure calculates group spacing based on size of the axis area and the number of bars in the chart.

If the requested spacing results in a chart that is too large to fit in the space available for the midpoint axis, an error message is written to the SAS log and no chart is produced.

Featured in: Example 7 on page 596

HTML=*variable*
identifies the variable in the input data set whose values create links in the HTML file created by the ODS HTML statement. These links are associated with an area of the chart and point to the data or graph you wish to display when the user drills down on the area.

HTML_LEGEND=*variable*
identifies the variable in the input data set whose values create links in the HTML file created by the ODS HTML statement. These links are associated with a legend value and point to the data or graph you wish to display when the user drills down on the value.

INSIDE=*statistic* (VBAR and VBAR3D only)
displays the values of the specified statistic inside the bars of a vertical bar chart. *Statistic* can be one of the following:

- FREQ
- CFREQ
- CPERCENT | CPCT
- MEAN
- PERCENT | PCT
- SUM

If the bars are subgrouped, only the following statistics are valid:

- FREQ
- PERCENT | PCT
- SUBPCT
- SUM

With subgroups, PERCENT displays the percent contribution of each subgroup to the midpoint value of the bar, based on frequency. The PERCENT values for each subgroup total the percent contribution of the bar to the whole. For example, if the percent contribution of the whole bar is 60%, the PERCENT statistic for all the subgroups in that bar will total 60%. To calculate PERCENT based on the SUMVAR= variable, use the FREQ= and TYPE= options. For details, see "Calculating Weighted Statistics" on page 529.

SUBPCT displays the percent contribution of each subgroup to the total bar. The SUBPCT values for each subgroup total the percent contribution to the whole bar. Because of rounding, the total of the percents may not equal 100.

Only one type of statistic can be printed inside the bars, but you can specify a second statistic with OUTSIDE=.

See also: "About Chart Statistics" on page 528 for a complete description of statistic types

Featured in: Example 4 on page 588 Example 7 on page 596

and LEGEND=LEGEND<1...99>
assigns the specified LEGEND definition to the legend generated by the SUBGROUP= option. The LEGEND= option itself does *not* generate a legend.

LEGEND= is ignored if

- SUBGROUP= is not used.
- the specified LEGEND definition is not in effect.
- the NOLEGEND option is used.
- the PATTERNID= option is set to any value other than SUBGROUP; that is, the value of PATTERNID= is BY or GROUP or MIDPOINT.

To create a legend based on the chart midpoints instead of the subgroups, use the chart variable as the subgroup variable:

```
hbar city / subgroup=city;
```

See also: "LEGEND Statement" on page 187 and SUBGROUP= on page 556 option

Featured in: Example 4 on page 588

LEVELS=*number-of-midpoints*
specifies the number of midpoints for a numeric chart variable. The range for each midpoint is calculated automatically, using the algorithm in Terrell and Scott (1985). LEVELS= is ignored if

- the chart variable is character type
- the DISCRETE option is used
- the MIDPOINTS= option is used.

MAXIS=AXIS<1...99>
assigns the specified AXIS definition to the midpoint axis. The MAXIS= option is ignored if the specified AXIS definition does not exist.

See also: "About Midpoints" on page 525 and "AXIS Statement" on page 162

Featured in: Example 4 on page 588

MEAN
displays the mean statistic in the table of statistics. Default statistics are suppressed when you request specific statistics, but all requested statistics are displayed. By default, the column heading includes the name of the variable for which the mean is calculated. MEAN is ignored unless you also use the SUMVAR= option.

MEANLABEL='*column-label*' (HBAR and HBAR3D only)
specifies the text of the column label for the MEAN statistic in the table of statistics. *column-label* can be up to 32 characters long, but a single line of the label can be no more than 24 characters. By default, a label with more than one word will break as close to the center of the line as possible. A double space in the string forces a line break.

Featured in: Example 6 on page 595

MIDPOINTS=*value-list*
specifies the midpoint values for the bars. The way you specify *value-list* depends on the type of the chart variable.

- *For numeric chart variables*, *value-list* is either an explicit list of values, or a starting and an ending value with an interval increment, or a combination of both forms:

  n <...*n*>

  n TO *n* <BY *increment*>

  n<...*n*> TO *n* <BY *increment*> <*n* <...*n*>>

  If a numeric variable has an associated format, the specified values must be the *unformatted* values.

  By default, numeric variable values are treated as continuous (if you omit the DISCRETE option), and

  - the lowest midpoint consolidates all data points from negative infinity to the median of the first two midpoints
  - the highest midpoint consolidates all data points from the median of the last two midpoints up to infinity
  - all other values in *value-list* specify the median of a range of values, and the GCHART procedure calculates the midpoint values.

  If you include the DISCRETE option, each value in *value-list* specifies a unique numeric value.

- *For character chart variables*, *value-list* is a list of unique character values enclosed in quotation marks and separated by blanks:

  'value-1' <...*'value-n'*>

  If a character variable has an associated format, the specified values must be the *formatted* values.

For a complete description of *value-list*, see the ORDER= on page 168 option in the AXIS statement.

If the *value-list* for either type of variable specifies so many midpoints that the axis values overwrite each other, the values may be unreadable. In this case the procedure writes a warning to the SAS log. On many devices, this problem can be corrected by either adjusting the size of the text with the HTEXT= graphics option or

by increasing the number of cells in your graphics display using the HPOS= and VPOS= graphics options.

The ORDER= option in the AXIS statement overrides the order specified in the MIDPOINTS= option. The bar chart statement options ASCENDING and DESCENDING also override both MIDPOINTS= and ORDER= in the AXIS statement.

See also: "About Midpoints" on page 525

Featured in: Example 5 on page 591

MIDPOINTS=OLD
tells the GCHART procedure to calculate default midpoints using the algorithm used in Release 82.4 and Version 5 of SAS/GRAPH. MIDPOINTS=OLD is ignored unless the chart variable is numeric type.

MINOR=*number-of-minor-ticks*
specifies the number of minor tick marks between each major tick mark on the response axis.

MINOR= in a bar chart statement overrides the number of minor tick marks specified in the MINOR= option in an AXIS definition assigned to the response axis with the RAXIS= option.

MISSING
accepts a missing value as a valid midpoint for the chart variable. By default, observations with missing values are ignored. Missing values are always valid for group and subgroup variables.

NAME='*entry-name*'
specifies the name of the catalog entry for the graph. The maximum length for *entry-name* is eight characters. The default name is GCHART. If the name duplicates an existing entry name, SAS/GRAPH software adds a number to the duplicate name to create a unique name– for example, GCHART1.

Featured in: Example 7 on page 596

NOAXIS
suppresses all axes, including axis lines, axis labels, axis values, and all major and minor tick marks. NOAXIS overrides the GAXIS=, MAXIS=, and RAXIS= options.

To remove selected axis elements such as lines, values or labels, use specific AXIS statement options.

NOAXIS does not suppress either the default frame or an axis area fill requested by the CFRAME= option. To remove the axis frame or the 3D backplane, use the NOFRAME option in the procedure. To remove the horizontal or vertical axis planes, use the NOPLANE option in the AXIS statement.

NOBASEREF
suppresses the zero reference line when the SUM or MEAN chart statistic has negative values.

NOLEGEND
suppresses the legend that is automatically generated by the SUBGROUP= option. NOLEGEND is ignored if the SUBGROUP= option is not used.

NOSTATS (HBAR and HBAR3D only)
suppresses the table of statistics. NOSTATS suppresses both the default statistics and specific statistics requested by the FREQ, CFREQ, PERCENT, CPERCENT, SUM, and MEAN options.

NOZERO
suppresses any midpoints for which there are no corresponding values of the chart variable and, hence, no bar. NOZERO usually is used with the GROUP= option to

suppress midpoints when not all values of the chart variable are present for every group or if the chart statistic for the bar is 0.

Note: If a bar is omitted and if you have also specified bar labels with the VALUE= option in an AXIS statement, the labels may be shifted and not displayed with the correct bar. △

Featured in: Example 7 on page 596

OUTSIDE=*statistic* (VBAR and VBAR3D only)
displays the values of the specified statistic above the bars of a vertical bar chart. *Statistic* can be one of the following:

- FREQ
- CFREQ
- PERCENT | PCT
- CPERCENT | CPCT
- SUM
- MEAN

Only one type of statistic can be printed outside the bars, but you can specify a second statistic with INSIDE=.

See also: "About Chart Statistics" on page 528 for a complete description of statistic types

Featured in: Example 4 on page 588 Example 7 on page 596

PATTERNID=BY | GROUP | MIDPOINT | SUBGROUP
specifies the way fill patterns are assigned. By default, PATTERNID=SUBGROUP. Values for PATTERNID= are as follows:

BY
changes patterns each time the value of the BY variable changes. All bars use the same pattern if the GCHART procedure does not include a BY statement.

GROUP
changes patterns every time the value of the group variable changes. All bars in each group use the same pattern, but a different pattern is used for each group.

MIDPOINT
changes patterns every time the midpoint value changes. If you use the GROUP= option, the respective midpoint patterns are repeated for each group.

SUBGROUP
changes patterns every time the value of the subgroup variable changes. The bars must be subdivided by the SUBGROUP= option for the SUBGROUP value to have an effect. Without SUBGROUP=, all bars have the same pattern.

Note: If you use the SUBGROUP= option and specify a PATTERNID= value other than SUBGROUP, the bar segments use the same pattern and are indistinguishable. △

See also: "Controlling Bar Chart Patterns and Colors" on page 558

Featured in: Example 4 on page 588 and Example 7 on page 596

PERCENT
PCT
displays the percentage statistic in the table of statistics. Default statistics are suppressed when you request specific statistics but all requested statistics are displayed.

PERCENTLABEL='*column-label*' (HBAR and HBAR3D only)
specifies the text of the column label for the PERCENT statistic in the table of statistics. *column-label* can be up to 32 characters long, but a single line of the label can be no more than 24 characters. By default, a label with more than one word will break as close to the center of the line as possible. A double space in the string forces a line break.

RAXIS=*value-list* | AXIS<1...99>
AXIS=*value-list* | AXIS<1...99>
specifies values for the major tick mark divisions on the response axis or assigns the specified AXIS definition to the axis. See the MIDPOINTS= on page 552 option for a description of *value-list*. By default, the GCHART procedure scales the response axis automatically and provides an appropriate number of tick marks.

You can specify negative values, but negative values are reasonable only when TYPE=SUM or TYPE=MEAN and one or more of the sums or means are less than 0. Frequency and percentage values are never less than 0.

For lists of values, a separate major tick mark is created for each individual value. A warning message is written to the SAS log if the values are not evenly spaced.

If the values represented by the bars are larger than the highest tick mark value, the bars are truncated at the highest tick mark.

If you use a BY statement with the PROC GCHART statement, the same response axes are produced for each BY group when RAXIS=*value-list* is used or if there is an ORDER= list in the AXIS statement assigned to the response axis.

See also: "AXIS Statement" on page 162

Featured in: Example 4 on page 588 and Example 7 on page 596

REF=*value-list*
draws reference lines at the specified points on the response axis. See the MIDPOINTS= on page 552 option for a description of *value-list*.

Values can be listed in any order, but should be within the range of values represented by the response axis. A warning is written to the SAS log if any of the points are off of the axis, and no reference line is drawn for such points. You can use the AUTOREF option to draw reference lines automatically at all of the major tick marks.

By default, reference lines in 3D bar charts are drawn on the back plane of the axis. To draw the reference lines in front of the bars, use the FRONTREF option.

SHAPE=*3D-bar-shape* (HBAR3D and VBAR3D only)
specifies the shape of the bars in charts that are produced with the HBAR3D and VBAR3D statements. *3D-bar-shape* can be one of the following:

- BLOCK | B (the default)
- CYLINDER | C
- HEXAGON | H
- PRISM | P
- STAR | S

The COUTLINE= option is not valid when SHAPE=CYLINDER.

Featured in: Example 7 on page 596

SPACE=*bar-spacing*
specifies the amount of space between individual bars or between the bars within each group if you also use the GROUP= option. *Bar-space* can be any non-negative number, including decimal values. Units are character cells. By default, the GCHART procedure calculates spacing based on the size of the axis area and the number of bars on the chart. Use SPACE=0 to leave no space between adjacent bars.

SPACE= is ignored if the specified spacing requests a chart that is too large to fit in the space available for the midpoint axis, and a warning message is issued.

Featured in: Example 4 on page 588 and Example 7 on page 596

SUBGROUP=*subgroup-variable*
divides the bars into segments according to the values of *subgroup-variable*. *Subgroup-variable* can be either character or numeric and is always treated as a discrete variable. SUBGROUP= creates a separate segment within each bar for every unique value of the subgroup variable for that midpoint.

If PATTERNID=SUBGROUP (the default setting), each segment is filled with a different pattern and a legend that provides a key to the patterns is automatically generated. If the value of PATTERNID= is anything other than SUBGROUP, the segments are all the same color, the legend is suppressed, and the subgrouping effect is lost.

By default the legend appears at the bottom of the chart. To modify the legend, assign a LEGEND definition with the LEGEND= option. To suppress the legend, specify NOLEGEND.

See also: "LEGEND Statement" on page 187

Featured in: Example 4 on page 588, Example 7 on page 596 and Example 5 on page 591

SUM
displays the sum statistic in the table of statistics. Default statistics are suppressed when you request specific statistics but all requested statistics are displayed. By default, the column heading includes the name of the variable for which the sum is calculated. SUM is ignored unless you also use the SUMVAR= option.

SUMLABEL='*column-label*' (HBAR and HBAR3D only)
specifies the text of the column label for the SUM statistic in the table of statistics. *Column-label* can be up to 32 characters long, but a single line of the label can be no more than 24 characters. By default, a label with more than one word will break as close to the center of the line as possible. A double space in the string forces a line break.

SUMVAR=*summary-variable*
specifies a numeric variable for sum or mean calculations. The GCHART procedure calculates the sum or, if requested, the mean of *summary-variable* for each midpoint. The resulting statistics are represented by the length of the bars along the response axis, and they are displayed at major tick marks.

When you use SUMVAR=, the TYPE= option must be either SUM or MEAN. With SUMVAR=, the default is TYPE=SUM.

Featured in: Example 3 on page 586 and Example 6 on page 595

TYPE=*statistic*
specifies the chart statistic.

- If the SUMVAR= option is not used, *statistic* can be one of the following:

  FREQ
  frequency (the default)

  CFREQ
  cumulative frequency

  PERCENT PCT
  percentage

  CPERCENT CPCT
  cumulative percentage

- If SUMVAR= is used, *statistic* can be:

SUM
sum (the default)

MEAN
mean

Because you cannot use TYPE=FREQ, TYPE=CFREQ, TYPE=PERCENT, or TYPE=CPERCENT with SUMVAR=, you must use FREQ= to calculate percentages, cumulative percentages, frequencies, or cumulative frequencies based on a sum. See also "Calculating Weighted Statistics" on page 529.

If you specify TYPE=MEAN and use the SUBGROUP= option, the height or length of the bar represents the mean for the entire midpoint. The subgroup segments are proportional to the subgroup's contribution to the sum for the bar. See also SUBGROUP= on page 556.

See also: "About Chart Statistics" on page 528 for a complete description of statistic types

Featured in: Example 6 on page 595

WIDTH=*bar-width*
specifies the width of the bars. By default, the GCHART procedure selects a bar width that accommodates the midpoint values displayed on the midpoint axis using a hardware font and a height of one cell. Units for *bar-width* are character cells. The value for *bar-width* must be greater than 0, but it does not have to be an integer, for example,

```
vbar site / width=1.5;
```

If the requested bar width results in a chart that is too large to fit in the space available for the midpoint axis, the procedure issues a warning in the log and ignores the WIDTH= specification. If the specified width is too narrow, the procedure may display the midpoint values vertically.

Featured in: Example 4 on page 588

WOUTLINE=*bar-outline-width*
specifies the width of the outline in pixels. WOUTLINE= affects both the bar and the subgroup outlines.

The Chart Statistic and the Response Axis

In bar charts, the scale of values of the chart statistic is displayed on the response axis. By default, the response axis is divided into evenly spaced intervals identified with major tick marks that are labeled with the corresponding statistic value. Minor tick marks are evenly distributed between the major tick marks unless a log axis has been requested. For sum and mean statistics, the major tick marks are labeled with values of the SUMVAR= variable (formatted if the variable has an associated format). The response axis is also labeled with the statistic type.

Displaying Statistics in Vertical Bar Charts You can display the chart statistic for each bar either above or inside the bars of a vertical bar chart by using the OUTSIDE= or INSIDE= option in the VBAR or VBAR3D statement. The bars must be wide enough to accommodate the text. You can adjust the width of the bars with the WIDTH= option. To change the size of the text, use the HTEXT= graphics option.

Specifying Logarithmic Axes Logarithmic axes can be specified with the AXIS statement. See "AXIS Statement" on page 162 for a complete discussion.

About the Table of Statistics in Horizontal Bar Charts

In addition to displaying one statistic graphically on the response axis, the HBAR and HBAR3D statements print a table of statistic values to the right of the bars. When the value of TYPE= is FREQ, CFREQ, PERCENT, or CPERCENT, the frequency, cumulative frequency, percentage, and cumulative percentage statistics are printed next to the bars by default. When TYPE=SUM, the frequency and sum statistic values are printed by default. When TYPE=MEAN, the frequency and mean statistic values are printed by default. For sum and mean, the name of the SUMVAR= variable is added to the heading for the column of values.

Modifying the Table of Statistics You can use the FREQ, CFREQ, PERCENT, CPERCENT, SUM, and MEAN options to select only certain statistics. Without the SUMVAR= option, only the frequency, cumulative frequency, percentage, and cumulative percentage statistics can be printed. With SUMVAR=, all statistics, including the sum and mean, can be printed. You can suppress all statistics with the NOSTATS option.

To change the column labels for any statistic in the table, use one or more of the statistic column label options: FREQLABEL=, CFREQLABEL=, PERCENTLABEL=, CPERCENTLABEL=, SUMLABEL=, and MEANLABEL=.

To control the font and size of the text in the table of statistics, use the HTEXT= and FTEXT= graphics options.

Ordering and Selecting Midpoints

To rearrange character or discrete numeric midpoint values or to select ranges for numeric values, use the MIDPOINTS= option. Remember that although changing the number of midpoints for numeric variables may change the range of values for individual midpoints, it does not change the range of values for the chart as a whole. For details, see “About Midpoints” on page 525.

Like MIDPOINTS=, the ORDER= option in the AXIS statement can rearrange the order of the midpoints or suppress the display of discrete numeric or character values. However, ORDER= cannot calculate the midpoints for a continuous numeric variable, or exclude values from the calculations. For details, see the description of the ORDER= on page 168 option.

Controlling Bar Chart Patterns and Colors

Default patterns and outlines Each bar in a bar chart is filled with a pattern. By default, the procedure

- fills the bars with bar/block patterns, beginning with the default fill, SOLID, and rotating it through the colors list. When the solid patterns are exhausted, the procedure selects the next default bar/block pattern and rotates it through the colors list. It continues in this fashion until all of the required patterns have been assigned.

  Note: 3D bar charts always uses solid patterns. △

  If you use the device's default colors and the first color in the list is either black or white, the procedure does not create a pattern in that color. If you specify a colors list with the COLORS= graphics option, the procedure uses all the colors in the list to generate the patterns.
- outlines bars and bar segments using the first color in the colors list.

See “About Patterns” on page 529 for more information on how the GCHART procedure assigns default patterns and outlines.

User-defined patterns To override the default patterns and select fills and colors for the bars or bar segments, use the PATTERN statement. Only bar/block patterns are valid; all other pattern fills are ignored. For a complete description of all bar/block patterns, see VALUE= on page 214 in "PATTERN Statement" on page 211.

Whenever you use PATTERN statements, the default pattern outline color changes to SAME. That is, the outline color is the same as the fill color. To specify the outline color, use the COUTLINE= option (see COUTLINE= on page 545).

When patterns change The PATTERNID= option controls when the pattern changes. By default, PATTERNID=SUBGROUP. Therefore, when you use the SUBGROUP= option to subdivide the bars, the pattern automatically changes each time the subgroup value changes, and each subdivision of the bar displays a different pattern. As a result, the number of values for the SUBGROUP= variable determines the number of bar patterns on the chart. If you do not subdivide the bars, all bars use the same pattern.

Instead of changing the pattern for each subgroup, you can change the pattern for each midpoint, each group, or each BY group by changing the value of PATTERNID=. See the PATTERNID= on page 554 option for details.

Axis color By default, axis elements use the first color in the colors list or the colors that are specified by AXIS statement color options. However, action statement options can also control the color of the axis lines, text, and frame.

To change the color of...	Use this option...
the axis text	CTEXT=
the axis lines	CAXIS=
the area within the frame	CFRAME=

PIE, PIE3D, and DONUT Statements

Create pie or donut charts in which the size of a pie slice represents the value of the chart statistic for that category of data in relation to the total chart statistic for all categories.

Requirements: At least one chart variable is required.

Global statements: LEGEND, PATTERN, TITLE, FOOTNOTE

Supports: Drill-down functionality

Description The PIE, PIE3D, and DONUT statements specify the variable or variables that define the categories of data to chart. These statements automatically

- determine the midpoints.
- calculate the chart statistic for each midpoint (the default is FREQ).
- scale each slice to represent its chart statistic. No slice is drawn if the chart statistic for the midpoint is 0.
- order the slices by midpoint value in ascending order starting at the three o'clock position and proceeding counterclockwise around the pie.
- print the slice name (midpoint value) and slice value (chart statistic) beside each slice.
- assign patterns and colors to the slices. The default pie/star pattern is PSOLID.

You can use statement options to select or order the midpoints (slices), to change the type of chart statistic, and to modify the appearance of the chart, including the content and position of the slice labels, and patterns used by the slices. You can also specify additional variables by which to group, subgroup, or sum the data. Statement options can also produce special effects, such as exploded or invisible slices.

Donut and pie charts allow grouping and subgrouping. Grouping creates two or more separate pie or donut charts that display in rows or columns on one graph. Subgrouping creates a separate ring of slices within the circle for each value of the subgroup variable. The concentric rings of the subgrouped pie or donut chart make it easy to compare slice values between subgroups.

In addition, you can use global statements to modify patterns and legends, as well as add titles, footnotes, and notes to the chart. You can also use an Annotate data set to enhance the chart.

Syntax

PIE | PIE3D | DONUT *chart-variable(s)* </ *option(s)*;>

option(s) can be one or more options from any or all of the following categories:

- appearance options
  - ANNOTATE=*Annotate-data-set*
  - CFILL=*fill-color*
  - COUTLINE=*slice-outline-color* | SAME
  - EXPLODE=*value-list*
  - FILL=SOLID | X
  - INVISIBLE=*value-list*
  - NOHEADING
  - WOUTLINE=*slice-outline-width*
- statistic options
  - FREQ=*numeric-variable*
  - SUMVAR=*summary-variable*
  - TYPE=*statistic*
- midpoint options
  - DISCRETE
  - LEVELS=*number-of-midpoints*
  - MIDPOINTS=*value-list*
  - MIDPOINTS=OLD
  - MISSING
  - OTHER=*percent-of-total*
- grouping and subgrouping options
  - ACROSS=*number-of-columns*
  - DOWN=*number-of-rows*
  - GROUP=*group-variable*
  - NOGROUPHEADING
  - SUBGROUP=*subgroup-variable*
- slice-ordering options
  - ANGLE=*degrees*
  - ASCENDING

CLOCKWISE
DESCENDING
JSTYLE

- slice-labeling options
  - CTEXT=*text-color*
  - LEGEND | LEGEND=LEGEND<1...99>
  - MATCHCOLOR
  - NOLEGEND
  - OTHERLABEL='*text-string*'
  - PERCENT=ARROW | INSIDE | NONE | OUTSIDE
  - SLICE=ARROW | INSIDE | NONE | OUTSIDE
  - VALUE=ARROW | INSIDE | NONE | OUTSIDE
- donut-labeling options (DONUT only):
  - DONUTPCT=*percent*
  - LABEL=(*text argument(s)*)
- catalog entry description options
  - DESCRIPTION='*entry-description*'
  - NAME='*entry-name*'
- ODS options
  - HTML=*variable*
  - HTML_LEGEND=*variable*

Required Arguments

chart-variable(s)
specifies one or more variables that define the categories of data to chart. Each chart variable draws a separate chart. All variables must be in the input data set. Separate multiple chart variables with blanks.

See also: "About Chart Variables" on page 525

Options

Options in a PIE, PIE3D, or DONUT statement affect all graphs that are produced by that statement. You can specify as many options as you want and list them in any order. For details on specifying colors, see Chapter 7, "SAS/GRAPH Colors," on page 139. For a complete description of the graphics options, see Chapter 9, "Graphics Options and Device Parameters Dictionary," on page 301.

ACROSS=*number-of-columns*
draws *number-of-columns* pies across the procedure output area. ACROSS is ignored unless you also use the GROUP= option. By default, ACROSS=1.

If *number-of-columns* calls for more pies than fit horizontally in the graphics output area, no pies are drawn and an error message is written to the SAS log.

If the DOWN= option also is used, the pies are drawn in left-to-right and top-to-bottom order.

Featured in: Example 11 on page 616

ANGLE=*degrees*
starts the first slice at the specified angle. A value of 0 for *degrees* corresponds to the 3 o'clock position. *Degrees* can be either positive or negative. Positive values move

the starting position in the counterclockwise direction; negative values move the starting position clockwise. By default, ANGLE=0. Successive slices are drawn counterclockwise from the starting slice.

ANNOTATE=*Annotate-data-set*
ANNO=*Annotate-data-set*
specifies a data set to annotate charts produced by the PIE, PIE3D, or DONUT statement.

Note: Annotate coordinate systems 1, 2, 7, and 8 (data system coordinates) are not valid with pie or donut charts. △

See also: Chapter 10, "The Annotate Data Set," on page 403

ASCENDING
arranges the slices in ascending order of the value of the chart statistic. By default, slices are arranged in ascending order of midpoint value, without regard to size. ASCENDING reorders the slices from smallest to largest. The OTHER slice is still last regardless of its size.

If you also use the GROUP= option, the reordering is performed separately for each group, so the order of the midpoint values may be different for each pie or donut.

ASCENDING overrides any midpoint order that is specified with the MIDPOINTS= option.

CFILL=*fill-color*
specifies one color for all patterns in the chart, regardless of whether the fill is solid or hatch. For the PIE3D statement, the fill is always solid. For the PIE and DONUT statements, if no pattern is specified on the PATTERN statement or with the FILL= option, the procedure starts with the default solid fill and then, beginning with P2N0, uses each default pie hatch pattern with the specified color. For the outline color, the procedure uses the foreground color, which is the first color in the colors lists. Use COUTLINE= to specify a different outline color. CFILL= overrides any other pattern color specification and controls the color of all slices.

See also: "Controlling Bar Chart Patterns and Colors" on page 558 and "About Patterns" on page 529

Featured in: Example 10 on page 614

CLOCKWISE
draws the slices clockwise starting at the 12 o'clock position. Although this position implies ANGLE=90, you can use ANGLE= to specify a different starting angle.

Featured in: Example 11 on page 616

COUTLINE=*slice-outline-color* | SAME
outlines all slices, rings (subgroups), and legend values (if a legend appears) in the specified color. SAME specifies that the outline color of a slice or a slice segment or a legend value is the same as the interior pattern color.

The default outline color depends on the PATTERN statement:

- If no PATTERN statement is specified, the default outline color is the foreground color (the first color in the colors list).
- If a PATTERN statement or the V6COMP graphics options is specified, the default is COUTLINE=SAME.

Note: If you specify empty patterns (VALUE=PEMPTY in a PATTERN statement), you should not change the outline color from the default value, SAME, to a single color. Otherwise, all of the outlines will be one color and you will not be able to distinguish between the empty areas. △

See also: "Controlling Slice Patterns and Colors" on page 571 and "About Patterns" on page 529

Featured in: Example 8 on page 610, Example 9 on page 612 and Example 11 on page 616

CTEXT=*text-color*

specifies the color for all text on the chart that is not otherwise assigned a color. Text includes all slice labels, the chart heading, and group headings if grouping is used. CTEXT= also affects the color of the slice label arrows. See "Selecting and Positioning Slice Labels" on page 570.

If you omit CTEXT=, PROC GCHART searches for a color specification in this order:

1. the CTEXT= option in a GOPTIONS statement
2. the first color in the colors list (the default).

The MATCHCOLOR option overrides the CTEXT= option for slice labels.

Featured in: Example 9 on page 612 and Example 11 on page 616

DESCENDING

arranges the slices in descending order of the value of the chart statistic. By default, slices are arranged in ascending order of midpoint value, without regard to size. DESCENDING reorders the slices from largest to smallest. The OTHER slice is still last, regardless of its size.

If you also use the GROUP= option, the reordering is performed separately for each group, so the order of midpoint values may be different for each pie or donut.

DESCENDING overrides any midpoint order that is specified with the MIDPOINTS= option.

Featured in: Example 11 on page 616

DESCRIPTION='*entry-description*'
DES='*entry-description*'

specifies the description of the catalog entry for the chart. The maximum length for *entry-description* is 40 characters. The description does not appear on the chart. By default, the GCHART procedure assigns a description of the form PIE (or PIE3D or DONUT) CHART OF *variable*, where *variable* is the name of the chart variable.

The *entry-description* can include the #BYLINE, #BYVAL, and #BYVAR substitution options, which work as they do when used on TITLE, FOOTNOTE, and NOTE statements. For more information, refer to the description of the options on page 262, and "Substituting BY Line Values in a Text String" on page 266. The 40-character limit applies before the substitution takes place for these options; thus, if in the SAS program the entry-description text exceeds 40 characters, it is truncated to 40 characters, and then the substitution is performed.

The descriptive text is shown in the "description" portion of each of the following:

- in the Results window
- among the catalog-entry properties that you can view from the Explorer window
- in the Table of Contents that is generated when you use CONTENTS= on an ODS HTML statement (see "Linking to Output through a Table of Contents" on page 86), assuming the GCHART output is generated while the contents page is open
- in the Description field of the PROC GREPLAY window

DISCRETE

treats a numeric chart variable as a discrete variable rather than as a continuous variable. The GCHART procedure creates a separate midpoint and, hence, a separate

slice for each unique value of the chart variable. If the chart variable has a format associated with it, each formatted value is treated as a midpoint.

The LEVELS= option is ignored when you use DISCRETE. The MIDPOINTS= option overrides DISCRETE.

DONUTPCT=*percent* (DONUT only)
specifies the size of the donut hole in percent of the radius of the whole chart. Values of *percent* range from 0 to 99. By default, DONUTPCT=25.

Featured in: Example 9 on page 612

DOWN=*number-of-rows*
draws *number-of-rows* pies vertically in the procedure output area. The DOWN= option is ignored unless you also use the GROUP= option. By default, DOWN=1.

If *number-of-rows* calls for more pies than fit vertically in the graphics area of the output device, no pies are drawn and an error message is written to the SAS log.

If you also use the ACROSS= option, the pies are drawn in left-to-right and top-to-bottom order.

EXPLODE=*value-list*
pulls the specified slices slightly out from the rest of the pie for added emphasis. *Value-list* is the list of midpoint values for the slices to be exploded. See the MIDPOINTS= on page 567 option for a description of *value-list*.

The values in the value list must match the existing midpoints exactly, including the case of character midpoints. Any values in the list that do not correspond to existing midpoints are ignored.

When you use EXPLODE=, the radius is reduced to allow room for exploded slices.

EXPLODE= does not work with subgroups.

Featured in: Example 8 on page 610

FILL=SOLID | X
specifies the fill pattern for *all* slices in the chart:

SOLID S
rotates a solid fill through the colors list as many times as necessary. This is the default.

X
rotates a single hatch pattern through the colors list as many times as necessary. PIE3D does not support FILL=X.

If you use default device colors (the COLORS= option is omitted), the fill skips the first color in the colors list.

FILL= overrides any pattern that is specified in PATTERN statements.

By default, the outline color is the first color in the colors list. If PATTERN statements are used to specify colors, the slice outline color matches the slice fill color.

By default, the fill patterns take the colors from the current colors list in rotation. If any PATTERN statements have been defined, the colors in the PATTERN definitions are used, in order, before the default color rotation.

See also: "Controlling Bar Chart Patterns and Colors" on page 558 and "PATTERN Statement" on page 211

FREQ=*numeric-variable*
specifies a variable whose values weight the contribution of each observation in the computation of the chart statistic. Each observation is counted the number of times specified by the value of *numeric-variable* for that observation. If the value of

numeric-variable is missing, 0, or negative, the observation is not used in the statistic calculation. Noninteger values of *numeric-variable* are truncated to integers.

FREQ= is valid with all chart statistics.

Because you cannot use TYPE=PERCENT or TYPE=FREQ with the SUMVAR= option, you must use FREQ= to calculate percentages and frequencies based on a sum.

See also: "Calculating Weighted Statistics" on page 529

GROUP=*group-variable*

organizes the data according to values of *group-variable* and produces a separate pie (or donut) chart for each unique value of *group-variable*. *Group-variable* can be either character or numeric and is always treated as a discrete variable. Missing values for *group-variable* are treated as a valid group. By default, each group includes only those midpoints with nonzero chart statistic values.

By default, the charts are produced in ascending order of group variable value and each is drawn on a separate page or display. Therefore, the effect of GROUP= is essentially the same as using a BY statement except that GROUP= causes the midpoints with the same value to use the same color and fill pattern. To place more than one pie on a page or display, use the ACROSS= or DOWN= options, or both.

See also: "BY Statement" on page 177

Featured in: Example 12 on page 618

HTML=*variable*

identifies the variable in the input data set whose values create links in the HTML file that is created by the ODS HTML statement. These links are associated with an area of the chart and point to the data or graph that you wish to display when the user drills down on the area.

HTML_LEGEND=*variable*

identifies the variable in the input data set whose values create links in the HTML file created by the ODS HTML statement. These links are associated with a legend value and point to the data or graph that you wish to display when the user drills down on the value.

INVISIBLE=*value-list*

makes the specified slices invisible, as if they had been removed from the pie. Labels are not printed for invisible slices. *Value-list* is the list of midpoint values for the invisible slices. See the MIDPOINTS= on page 567 option for a description of *value-list*.

The values in the value list must match the existing midpoints exactly, including the case of character midpoints. Any values in the list that do not correspond to existing midpoints are ignored.

JSTYLE

arranges the midpoints in descending order of the statistic value and draws the slices clockwise starting at the 12 o'clock position. The JSTYLE option has the same effect as specifying both the DESCENDING and CLOCKWISE options.

LABEL=(*text argument(s)*) (DONUT only)

defines the text that is displayed in the donut hole. *Text-argument(s)* defines the text or the appearance of the label, or both. *Text-argument(s)* can be one or more of the following:

'text-string'

provides the text of the label. Enclose each string in quotation marks. Separate multiple strings with blanks.

text-description-suboption
modifies a characteristic such as the font, color, or size of the text string(s) that follows it. *Text-description-suboption* can be

ANGLE=*degrees*

COLOR=*color*

FONT=*font*

HEIGHT=*text-height <units>*

JUSTIFY=LEFT | CENTER | RIGHT

ROTATE=*degrees*

See "Text Description Suboptions" on page 570 for a complete description.

Specify as many text strings and text description suboptions as you want, but enclose them all in one set of parentheses.

Featured in: Example 9 on page 612

LEGEND | LEGEND=LEGEND<1...99>
generates a legend for the slice names (midpoint values) instead of printing them beside the slices. The legend displays each slice name and its associated pattern. This option also suppresses the display of the chart statistic values. To display the chart statistics, use the VALUE= option.

If you use the SUBGROUP= option, the legend is automatically generated. However, because patterning is always by midpoint, the legend still describes the midpoint values, not the subgroups.

Note: If you request a legend and the slices use hatch patterns, the patterns in the slices are oriented to be visually equivalent to the legend. △

Specifying LEGEND=LEGEND*n* assigns the specified LEGEND statement to the legend.

See also: "LEGEND Statement" on page 187 and SUBGROUP= on page 568 option

Featured in: Example 9 on page 612 Example 11 on page 616

LEVELS=*number-of-midpoints*
specifies the number of midpoints for a numeric chart variable. The range for each midpoint is calculated automatically. LEVELS= is ignored if

- the chart variable is character type
- the DISCRETE option is used
- the MIDPOINTS= option is used.

MATCHCOLOR
uses the slice pattern color for all slice labels. MATCHCOLOR overrides the color that is specified in the CTEXT= option.

MIDPOINTS=*value-list*
specifies the midpoint values for the slices. The way you specify *value-list* depends on the type of variable:

- *For numeric chart variables*, *value-list* is either an explicit list of values, or a starting and an ending value with an interval increment, or a combination of both forms:

  n <...*n*>

  n TO *n* <BY *increment*>

  <*n*...> *n* TO *n* <BY *increment*> <*n* <...*n*>>

  If a numeric variable has an associated format, the specified values must be the *unformatted* values.

  By default, numeric variable values are treated as continuous (if you omit the DISCRETE option), and

  - the lowest midpoint consolidates all data points from negative infinity to the median of the first two midpoints
  - the highest midpoint consolidates all data points from the median of the last two midpoints up to infinity
  - all other values in *value-list* specify the median of a range of values, and the GCHART procedure calculates the midpoint values.

  If you include the DISCRETE option, each value in *value-list* specifies a unique numeric value.

- *For character chart variables*, *value-list* is a list of unique character values enclosed in quotation marks and separated by blanks:

  'value-1' <...*'value-n'*>

  If a character variable has an associated format, the specified values must be the *formatted* values.

For a complete description of *value-list*, see the ORDER= on page 168 option in the AXIS statement.

Midpoints that represent small percentages are collected into a generic midpoint named OTHER. See the OTHER= on page 568 option and the OTHERLABEL= on page 568 option for more information.

See also: "About Midpoints" on page 525

Featured in: Example 10 on page 614

MIDPOINTS=OLD
tells the GCHART procedure to calculate default midpoints using the algorithm used in Release 82.4 and Version 5 of SAS/GRAPH. The MIDPOINTS=OLD option is ignored unless the numeric chart variable is numeric type.

MISSING
accepts a missing value as a valid midpoint for the chart variable. By default, observations with a missing value are ignored. Missing values are always valid for the group and subgroup variable.

NAME='*entry-name*'
specifies the name of the catalog entry for the graph. The maximum length for *entry-name* is eight characters. The default name is GCHART. If the name duplicates an existing entry name, SAS/GRAPH software adds a number to the duplicate name to create a unique name– for example, GCHART1.

NOGROUPHEADING
suppresses the headings that are normally printed above each pie when you use the GROUP= option.

NOHEADING
suppresses the heading that is normally printed at the top of each page or display of output.

Featured in: Example 9 on page 612

NOLEGEND
suppresses the legend that is automatically generated by the SUBGROUP= option. NOLEGEND is ignored if the SUBGROUP= option is not used.

OTHER=*percent-of-total*
collects all midpoints with chart statistic values less than or equal to *percent-of-total* into a generic midpoint named OTHER. The value of *percent-of-total* can be 0 to 100; the default value is 4. Therefore, any slice that represents 4 percent or less of the total is put in the OTHER category.

Note: If you specify a small value for *percent-of-total*, the GCHART procedure may not be able to label all of the small slices. △

The OTHER slice is the last slice in the pie, regardless of the order of the slices. (In other words, it is the slice immediately before the starting slice.)

If only one midpoint falls into the OTHER category, its slice is displayed in its normal position in the pie and retains its original label. For example, suppose a pie has these slices and percent values: Coal 35%, Gas 15%, Hydro 5%, and Oil 45%. If you specify OTHER=5, Hydro remains the third slice instead of becoming the last slice.

Featured in: Example 11 on page 616 and Example 12 on page 618

OTHERLABEL='*text-string*'
specifies a text string up to 16 characters for the label for the OTHER slice. The default label is OTHER.

Featured in: Example 11 on page 616

PERCENT=ARROW | INSIDE | NONE | OUTSIDE
prints the percentage represented by each slice using the specified labeling method. For a description of the option values, see "Selecting and Positioning Slice Labels" on page 570. By default, PERCENT=NONE (percentage is not displayed).

Whether the slice percent displays with or without decimal places, depends on the range of values across the chart.The only way to control the appearance of these values is to calculate the percentage with a DATA step or statistical procedure and use the resulting data set as input to the GCHART procedure. Assign the variable that contains the calculated percentages to the SUMVAR= option.

Featured in: Example 10 on page 614 and Example 12 on page 618

SLICE=ARROW | INSIDE | NONE | OUTSIDE
controls the position and style of the slice name (midpoint value) for each slice. For a description of the option values, see "Selecting and Positioning Slice Labels" on page 570. By default, SLICE=OUTSIDE (the name is outside of the slice).

Featured in: Example 10 on page 614 and Example 12 on page 618

SUBGROUP=*subgroup-variable*
divides the chart into concentric rings according to the values of *subgroup-variable*. *Subgroup-variable* can be either character or numeric and is always treated as a discrete variable.

The width of the rings, which is the same for each subgroup, is determined by the radius of the pie and the size of the donut hole, if any.

By default, the subgroup rings are ordered from the outside in, alphabetically (if character) or numerically (if numeric). If the JSTYLE option is also used, the order of the slices within the subgroups is determined by the outermost subgroup. Any

inner subgroup that contains a value that is not in the outer subgroup, places the new slice for that value either last or just before the "other" slice, if one is present. That slice order is continued for any remaining subgroups.

Each ring is labeled with its subgroup value; labels are placed to the right of the chart. If the GROUP= option is also used, only the first (upper left) chart on each page is labeled.

By default the subgroups are outlined in the foreground color. To specify an outline color, use the COUTLINE= option.

SUBGROUP= automatically generates a legend for the midpoint values (not the subgroup values) and suppresses display of the chart statistic. By default the legend appears at the bottom of the chart. To modify the legend, assign a LEGEND definition. To suppress the legend, specify NOLEGEND. To display the chart statistic, use the VALUE= option.

If EXPLODE is also used, it is ignored.

See also: "Controlling Bar Chart Patterns and Colors" on page 558 and "LEGEND Statement" on page 187

Featured in: Example 9 on page 612 and Example 10 on page 614

SUMVAR=*summary-variable*

specifies a numeric variable for sum or mean calculations. The GCHART procedure calculates the sum or, if requested, the mean of *numeric-variable* for each midpoint. The resulting statistics are represented by the size of the slice and displayed beside of each slice.

When you use SUMVAR=, the TYPE= option must be either SUM or MEAN. With SUMVAR=, the default is TYPE=SUM.

Featured in: Example 8 on page 610

TYPE=*statistic*

specifies the chart statistic.

- If the SUMVAR= option is not used, *statistic* can be one of the following:

  FREQ
  frequency (the default)

  PERCENT PCT
  percentage

- If SUMVAR= is used, *statistic* can be one of the following:

  SUM
  sum (the default)

  MEAN
  mean

Because you cannot use TYPE=FREQ or TYPE=PERCENT with SUMVAR=, you must use FREQ= to calculate percentages or frequencies based on a sum. See also "Calculating Weighted Statistics" on page 529.

See also: "About Chart Statistics" on page 528

VALUE=ARROW | INSIDE | NONE | OUTSIDE

controls the position and style of the slice value (chart statistic) for each slice. For a description of the option values see "Selecting and Positioning Slice Labels" on page 570. By default, VALUE=OUTSIDE (the value is outside the slice).

Featured in: Example 10 on page 614 and Example 11 on page 616

WOUTLINE=*slice-outline-width*

specifies the width of the outline in pixels. WOUTLINE= affects both the slice and the subgroup outlines.

Text Description Suboptions

The LABEL= option in the DONUT statement uses text description suboptions to change the color, height, justification, font, and angle of the following text string(s).

ANGLE=*degrees*
A=*degrees*
specifies the angle at which the baseline of the text string(s) is rotated with respect to the horizontal. A positive value for *degrees* moves the baseline counterclockwise; a negative value moves it clockwise. By default, ANGLE=0 (horizontal).

COLOR=*color*
C=*color*
specifies the color for the text string(s). The COLOR= suboption stays in effect until another COLOR= specification is encountered. If you omit COLOR=, LABEL= uses the first color in the colors list. It ignores the CTEXT= graphics option. See Chapter 7, "SAS/GRAPH Colors," on page 139 for details on specifying *color*.

FONT=*font*
F=*font*
specifies the font for the text string(s). If you omit FONT=, LABEL= uses the font that is specified by the FTEXT= graphics option. If no font is specified, it uses the default hardware font, NONE. See Chapter 6, "SAS/GRAPH Fonts," on page 125 for details on specifying *font*.

HEIGHT=*text-height <units>*
H=*text-height <units>*
specifies the height of the text string(s). *Text-height* is the number of units. If you omit HEIGHT=, LABEL= uses the height that is specified by the HTEXT= graphics option. If no text height is specified and if the default text height is too large for the donut hole, the size of the label is reduced to fit. *Units* can be CELLS | CM | IN | PCT | PT. If you omit *units*, HEIGHT= uses the unit that is specified by the GUNIT= graphics option, or the default unit, CELLS.

JUSTIFY=LEFT | CENTER | RIGHT
J=L | C | R
specifies the alignment of the text string(s). By default, JUSTIFY=CENTER.

ROTATE=*degrees*
specifies the angle at which each character is rotated with respect to the baseline of the text string. The angle is measured from the current text baseline angle specified by the ANGLE= suboption. A positive value for *degrees* rotates the character counterclockwise; a negative value rotates it clockwise. By default, ROTATE=0 (parallel to the baseline).

Selecting and Positioning Slice Labels

By default, each slice is labeled with its midpoint value (slice name) and its chart statistic value (slice value), which are printed outside of the slice. You can control where and how these labels are displayed with the SLICE= and VALUE= options, respectively. In addition, each slice can display the percentage its midpoint contributes to the total chart statistic(slice percent). Use the PERCENT= option to request slice percent.

The SLICE=, VALUE=, and PERCENT= options use the same values:

ARROW
places the text outside the slice and connects the text to the slice with a line. This labeling method reduces the radius of the pie. The arrow uses the color that is specified by CTEXT= in the PIE, PIE3D, or DONUT statement. If CTEXT= is omitted, the arrow uses the first color in the colors list.

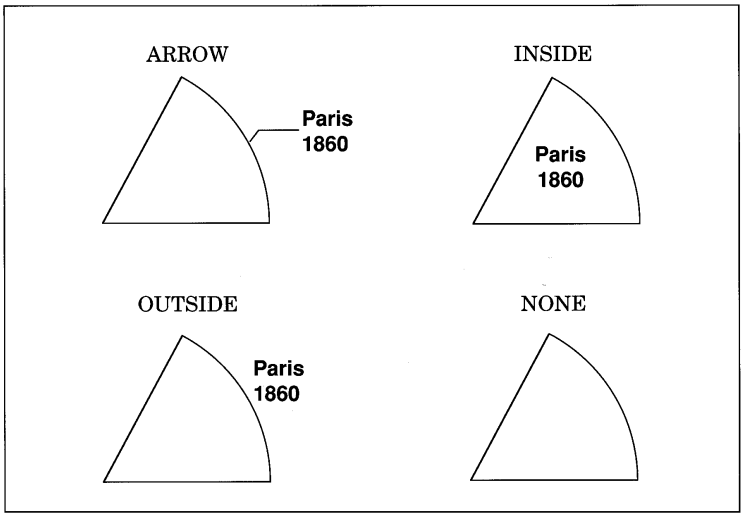
INSIDE
places the text inside the slice. The label overlays the slice fill patterns. This labeling method increases the radius of the pie.

NONE
suppresses the text.

OUTSIDE
places the text outside of the slice.

Figure 13.13 on page 571 illustrates these values.

Figure 13.13 Slice Labeling Methods



The SLICE= and VALUE= options are dependent on each other. If you specify only VALUE= or only SLICE=, the other option automatically uses the same labeling method. PERCENT= is independent of these two.

Be careful about the combinations that you specify. For example, if you specify PERCENT=ARROW and VALUE=OUTSIDE, the line that connects the percentage information to each slice may overlay the statistic value.

If your pie has many slices, the labels may overlap, particularly if there are several small slices together. You can correct the overlapping labels by using

- FTEXT= graphics option to decrease the size of the labels.
- the Graphics Editor to adjust the labels by moving or resizing the text.
- ANGLE= to change the orientation of the pie.
- MIDPOINTS= to rearrange slices so that small slices are not together.
- OTHER= to group more midpoints into the OTHER category.
- the HPOS= and VPOS= graphics options to increase the number of cells in your display. (See “About the Graphics Output Area” on page 29 for details.)

Controlling Slice Patterns and Colors

Pie and donut charts are always patterned by midpoint. Even when you specify subgrouping, the patterning method does not change from midpoint to subgroup.

Default patterns and outlines Each slice in a pie or donut chart is filled with a pattern. By default, the procedure

- fills the slices with pie/star patterns, beginning with the default fill, PSOLID, and rotating through the colors in the colors list. When the solid patterns are exhausted, the procedure selects the next default pie/star pattern and rotates it through the colors list. It continues in this fashion until all of the required patterns have been assigned.

  Note: PIE3D always uses solid patterns. △

  If you use the device's default colors and the first color in the list is either black or white, the procedure does not create a pattern in that color. If you specify a colors list with the COLORS= graphics option, the procedure uses all the colors in the list to generate the patterns.
- outlines slices and subgroup segments using the first color in the colors list. To change the outline color, use the COUTLINE= option.

See "About Patterns" on page 529 for more information on how the GCHART procedure assigns default patterns and outlines.

Controlling patterns You can control slice patterns and their outlines in several ways.

- To select a different fill for the slices, such as empty or hatched, you can
  - request a single hatched fill pattern for all slices by specifying the FILL=X option on the PIE or DONUT statement. The pattern specified by FILL=X rotates through the colors list as many times as needed to generate all of the patterns that are required by the chart. If you specify a single color with either CFILL= or the graphics option, CPATTERN=, all slices use the same color as well as the same pattern.
  - specify a pattern with the VALUE= option in the PATTERN statement. Only pie/star patterns are valid; all other pattern specifications are ignored. For a complete description of all pie/star patterns, see VALUE= on page 217 in "PATTERN Statement" on page 211.

    If no color options are specified, the procedure rotates each specified fill once through the colors list. Otherwise the PATTERN statement generates one pattern definition for the specified pattern and color. When all of the specified patterns are exhausted, the procedure starts rotating through the default pie/star patterns, beginning with PSOLID.
- To select colors for the slices, you can
  - specify a single pattern color with the CFILL= option, or with the CPATTERN= graphics option, or with a colors list of one color. For the PIE and DONUT statements, CFILL= starts with the default solid color and uses the foreground color for outlines, whereas CPATTERN= and a colors list of one color skip the solid pattern and, beginning with P2N0, use each pie/star hatch pattern with the specified color, and use the fill color for the outline color.
  - specify only COLOR= in one or more PATTERN statements. In this case, the procedure creates a solid pattern for each specified color. When it runs out of PATTERN statements, it returns to the default patterns, beginning with PSOLID, and rotates them each through the colors list. Whenever you specify a PATTERN statement, the default outline color is SAME.
- To define specific patterns and colors for the slices, use PATTERN statements and specify both the VALUE= and COLOR= options. If you provide fewer PATTERN definitions than the chart requires, the GCHART procedure uses the default pattern rotation for the slices that are drawn after all of the defined patterns are exhausted.

Whenever you use PATTERN statements, the default outline color changes to SAME. That is, the outline color is the same as the fill color. To change the outline color, use the COUTLINE= on page 562 option.

See "About Patterns" on page 529 for more information on how the GCHART procedure uses patterns and outlines. See "PATTERN Statement" on page 211 for a description of default pie/star patterns.

Modifying the Statistic Heading and the Group Heading

By default, the procedure prints a heading at the top of each pie (or donut) chart that indicates the type of statistic charted and the name of the chart variable– for example, *SUM of SALES by SITE*. You can suppress this heading with the NOHEADING option.

When you use the GROUP= option, a heading is printed above each pie indicating the name of the group variable and its value for the particular pie– for example, *SITE=Paris*. You can suppress these headings with the NOGROUPHEADING option. You can also suppress the variable name *SITE=* so that only the value *Paris* remains. To do this, use a LABEL statement and assign a null value to the variable name, for example,

```
label site='00'x;
```

Because the AXIS statement cannot be used by the PIE, PIE3D, and DONUT statements, you should use the FTEXT= and HTEXT= graphics options to control the font and height of text on the chart. Increasing the value of the HTEXT= graphics option decreases the size of the pie if any slice labels are positioned outside.

STAR Statement

Creates star charts in which the length of the spines represents the value of the chart statistic for each category of data or midpoint.

Requirements: At least one chart variable is required.

Global statements: FOOTNOTE, PATTERN, TITLE,

Supports: Drill-down functionality (slices only)

Description The STAR statement specifies the variable or variables that define the categories of data to chart. This statement automatically

- determines the midpoints.
- calculates the chart statistic for each midpoint (the default is FREQ).
- scales each spine or slice to represent the chart statistic. Slices or spines are drawn for all midpoints where the value of the chart statistic is greater than the value that is specified in the STARMIN= option.
- arranges the spines or slice counterclockwise around the star in ascending order of midpoint value, starting at the three o'clock position.
- prints the midpoint value and chart statistic beside each spine or slice.
- assigns patterns to the slices.

If all the data to be charted with the STAR statement are positive, the center of the star represents 0 and the outside circle represents the maximum value. If negative values are calculated for the chart statistic, the center represents the minimum value

in the data. You can specify other values for the center and outside of the circle with the STARMIN= and STARMAX= options.

You can also use statement options to select or order the midpoints, to change the type of chart statistic, and to modify the appearance of the chart, including the content and position of the spine or slice labels, and patterns that fill the slice. You can specify additional variables by which to group or sum the data.

Star charts allow grouping, which creates two or more separate charts that display in rows or columns on one graph.

In addition, you can use global statements to modify patterns as well as add titles, footnotes, and notes to the chart. You can also use an Annotate data set to enhance the chart.

Syntax

STAR *chart-variable(s)* </ *option(s)*;>

option(s) can be one or more options from any or all of the following categories:

- appearance options
  - ANGLE=*degrees*
  - ANNOTATE=*Annotate-data-set*
  - CFILL=*fill-color*
  - COUTLINE=*star-outline-color* | SAME
  - FILL=SOLID | X
  - NOCONNECT
  - STARMAX=*max-value*
  - STARMIN=*min-value*
  - WOUTLINE=*slice-outline-width*
- statistic options
  - FREQ=*numeric-variable*
  - SUMVAR=*summary-variable*
  - TYPE=*statistic*
- midpoint options
  - DISCRETE
  - LEVELS=*number-of-midpoints*
  - MIDPOINTS=*value-list*
  - MIDPOINTS=OLD
  - MISSING
- grouping and subgrouping options
  - ACROSS=*number-of-columns*
  - DOWN=*number-of-rows*
  - GROUP=*group-variable*
- labeling options
  - CTEXT=*text-color*
  - MATCHCOLOR
  - NOGROUPHEADING
  - NOHEADING
  - PERCENT=ARROW | INSIDE | NONE | OUTSIDE
  - SLICE=ARROW | INSIDE | NONE | OUTSIDE

VALUE=ARROW | INSIDE | NONE | OUTSIDE

- catalog entry description options

  DESCRIPTION='*entry-description*'

  NAME='*entry-name*'

- ODS options

  HTML=*variable*

Required Arguments

chart-variable(s)
specifies one or more variables that define the categories of data to chart. Each chart variable draws a separate chart. All variables must be in the input data set. Separate multiple chart variables with blanks.

See also: "About Chart Variables" on page 525

Options

Options in a STAR statement affect all of the graphs that are produced by that statement. You can specify as many options as you want and list them in any order. For details on specifying colors, see Chapter 7, "SAS/GRAPH Colors," on page 139.

ACROSS=*number-of-columns*
draws *number-of-columns* stars across the procedure output area. ACROSS= is ignored unless you also use the GROUP= option. By default, ACROSS=1. If *number-of-columns* calls for more stars than fit horizontally in the graphics area of the output device, no stars are drawn and an error message is written to the SAS log.

If you also use the DOWN= option, the star charts are drawn in left-to-right and top-to-bottom order.

ANGLE=*degrees*
starts the first slice at the specified angle. A value of 0 for *degrees* corresponds to the 3 o'clock position. *Degrees* can be either positive or negative. Positive values move the starting position counterclockwise; negative values move the starting position clockwise.

If the star chart uses spines instead of slices, *degrees* specifies the angle of the position halfway between the first spine and the last spine.

By default, ANGLE=0, which places the first spine or the center of the first slice of the star at the 0 degree position. Successive star spines or slices are drawn counterclockwise from the starting position.

ANNOTATE=*Annotate-data-set*
ANNO=*Annotate-data-set*
specifies a data set to annotate charts that are produced by the STAR statement.

Note: Annotate coordinate systems 1, 2, 7, and 8 (data system coordinates) are not valid with star charts. △

See also: Chapter 10, "The Annotate Data Set," on page 403

CFILL=*fill-color*
specifies one color for all slices in the chart, regardless of whether the fill is solid or hatch. If no pattern is specified on the PATTERN statement or with the FILL= option, the procedure starts with the default solid fill and then, beginning with P2N0, uses each default star hatch pattern with the specified color. For the outline color, the procedure uses the foreground color, which is the first color in the colors

lists. Use COUTLINE= to specify a different outline color. CFILL= overrides any other pattern color specification and controls the color of all slices.

COUTLINE=*star-outline-color* | SAME
specifies the color for the circle that surrounds the star chart and for the slice outlines or spines.

SAME specifies that the outline color of a slice is the same as the interior pattern color. Specifying COUTLINE=SAME affects only slice outlines and has no effect on the color of the circle.

The default circle color is the first color in the colors list (the foreground color). The default slice outline color depends on the PATTERN statement:

- If you do not specify the PATTERN statement, the default outline color is the foreground color (the first color in the colors list).
- If you do not specify the PATTERN statement or the V6COMP graphics options, the default is COUTLINE=SAME.

Note: If you specify empty patterns, (VALUE=PEMPTY in a PATTERN statement) you should not change the outline color from the default value, SAME, to a single color. Otherwise all the outlines will be one color and you will not be able to distinguish between the empty areas. △

See also: "Selecting Patterns for the Star Charts" on page 580 and "About Patterns" on page 529

Featured in: Example 14 on page 621

CTEXT=*text-color*
specifies a color for all text on the chart. Text includes all slice labels, the chart heading, and group headings if grouping is used.

If you omit CTEXT=, PROC GCHART searches for a color specification in this order:

1. the CTEXT= option in a GOPTIONS statement
2. the first color in the colors list (the default).

The MATCHCOLOR option overrides the CTEXT= option for star slice labels.

DESCRIPTION='*entry-description*'
DES='*entry-description*'
specifies the description of the catalog entry for the chart. The maximum length for *entry-description* is 40 characters. The description does not appear on the chart. By default, the GCHART procedure assigns a description of the form STAR CHART OF *variable*, where *variable* is the name of the chart variable.

The *entry-description* can include the #BYLINE, #BYVAL, and #BYVAR substitution options, which work as they do when used on TITLE, FOOTNOTE, and NOTE statements. For more information, refer to the description of the options on page 262, and "Substituting BY Line Values in a Text String" on page 266. The 40-character limit applies before the substitution takes place for these options; thus, if in the SAS program the entry-description text exceeds 40 characters, it is truncated to 40 characters, and then the substitution is performed.

The descriptive text is shown in the "description" portion of each of the following:

- in the Results window
- among the catalog-entry properties that you can view from the Explorer window
- in the Table of Contents that is generated when you use CONTENTS= on an ODS HTML statement (see "Linking to Output through a Table of Contents" on page 86), assuming the GCHART output is generated while the contents page is open
- in the Description field of the PROC GREPLAY window

DISCRETE
treats a numeric chart variable as a discrete variable rather than as a continuous variable. The GCHART procedure creates a separate midpoint and, hence, a separate star slice for each unique value of the chart variable. If the variable has a format associated with it, each format value is treated as a separate value.

The LEVELS= option is ignored when you use the DISCRETE option. The MIDPOINTS= option overrides the DISCRETE option.

Featured in: Example 14 on page 621

DOWN=*number-of-rows*
draws *number-of-rows* stars vertically in the procedure output area. The DOWN= option is ignored unless you also use the GROUP= option. If *number-of-rows* calls for more stars than fit vertically in the graphics area of the output device, no stars are drawn and an error message is written to the SAS log.

If you also use the ACROSS= option, the stars are drawn in left-to-right and top-to-bottom order.

FILL=SOLID | X
specifies the fill pattern for *all* slices in the star chart:

SOLID
S
rotates a solid fill through the colors list as many times as necessary. This is the default.

X
rotates a single hatch pattern through the colors list as many times as necessary.

If you use default device colors (the COLORS= option is omitted), the fill skips the first color in the colors list.

FILL= overrides any patterns that are specified in PATTERN statements.

By default, the fill patterns take the colors from the current colors list in rotation. If any PATTERN statements have been defined, the colors in the PATTERN definitions are used, in order, before the default color rotation.

Featured in: Example 14 on page 621

FREQ=*numeric-variable*
specifies a variable whose values weight the contribution of each observation in the computation of the chart statistic. Each observation is counted the number of times that are specified by the value of *numeric-variable* for that observation. If the value of *numeric-variable* is missing, 0, or negative, the observation is not used in the statistic calculation. Noninteger values of *numeric-variable* are truncated to integers.

FREQ= is valid with all chart statistics.

Because you cannot use TYPE=PERCENT or TYPE=FREQ with the SUMVAR= option, you must use FREQ= to calculate percentages and frequencies based on a sum.

See also: "Calculating Weighted Statistics" on page 529

GROUP=*variable*
organizes the data according to values of *group-variable* and produces a separate star chart for each unique value of *group-variable*. *Group-variable* can be either character or numeric and is always treated as a discrete variable. Missing values for *group-variable* are treated as a valid group.

By default, the charts are produced in ascending order of group variable value and each is drawn on a separate page or display. Therefore, the effect of GROUP= is essentially the same as using a BY statement except that GROUP= causes the midpoints with the same value to use the same color and fill pattern. To place more than one star chart on a page or display, use the ACROSS= or DOWN= options, or both.

HTML=*variable*
identifies the variable in the input data set whose values create links in the HTML file that is created by the ODS HTML statement. These links are associated with an area of the chart and point to the data or graph that you wish to display when the user drills down on the area. Only star charts with slices support drill-down.

LEVELS=*number-of-midpoints*
specifies number of midpoints for a numeric chart variable. The range for each midpoint is calculated automatically using the algorithm described by Terrell and Scott (1985). LEVELS= is ignored if

- the chart variable is character type
- the DISCRETE option is used
- the MIDPOINTS= option is used.

MATCHCOLOR
uses the slice pattern color for all slice labels. MATCHCOLOR overrides the color that is specified in the CTEXT= option. If the chart uses spines instead of slices, the spine color is used for the slice label and value text.

MIDPOINTS=*value-list*
specifies the midpoint values for the slices. The way you specify *value-list* depends on the type of variable:

- *For numeric chart variables, value-list* is either an explicit list of values, or a starting and an ending value with an interval increment, or a combination of both forms:

  n <...*n*>

  n TO *n* <BY *increment*>

  n <...*n*> TO *n* <BY *increment*> <*n* <...*n*>>

  If a numeric variable has an associated format, the specified values must be the *unformatted* values.

  By default, numeric variable values are treated as continuous (if you omit the DISCRETE option), and

  - the lowest midpoint consolidates all data points from negative infinity to the median of the first two midpoints
  - the highest midpoint consolidates all data points from the median of the last two midpoints up to infinity
  - all other values in *value-list* specify the median of a range of values, and the GCHART procedure calculates the midpoint values.

  If you include the DISCRETE option, each value in *value-list* specifies a unique numeric value.

- *For character chart variables, value-list* is a list of unique character values enclosed in quotation marks and separated by blanks:

  'value-1' <...'value-n'>

  If a character variable has an associated format, the specified values must be the *formatted* values.

For a complete description of *value-list*, see the ORDER= on page 168 option in the AXIS statement.

See also: "About Midpoints" on page 525

MIDPOINTS=OLD
tells the GCHART procedure to calculate default midpoints using the algorithm that is used in Release 82.4 and Version 5 of SAS/GRAPH. The MIDPOINTS=OLD option is ignored unless the chart variable is numeric type.

MISSING
accepts a missing value as a valid midpoint for the chart variable. By default, observations with a missing value are ignored. Missing values are always valid for the group variable.

NAME='*entry-name*'
specifies the name of the catalog entry for the graph. The maximum length for *entry-name* is eight characters. The default name is GCHART. If the name duplicates an existing entry name, SAS/GRAPH software adds a number to the duplicate name to create a unique name– for example, GCHART1.

NOCONNECT
draws only star spines without connecting lines. By default, the spines are connected to form slices.

Featured in: Example 14 on page 621

NOGROUPHEADING
suppresses the headings normally printed above each star when you use the GROUP= option.

NOHEADING
suppresses the heading normally printed at the top of each page or display of star chart output.

Featured in: Example 14 on page 621

PERCENT=ARROW | INSIDE | NONE | OUTSIDE
prints the percentage represented by each slice using the specified labeling method. For a description of the option values see "Selecting and Positioning Spine and Slice Labels" on page 580. By default, PERCENT=NONE (percentage is not displayed).

SLICE=ARROW | INSIDE | NONE | OUTSIDE
controls the position and style of the slice name (midpoint value) for each slice. For a description of the option values, see "Selecting and Positioning Spine and Slice Labels" on page 580. By default, SLICE=OUTSIDE (the name is outside the slice).

STARMAX=*max-value*
scales the chart so that the outside (or edge) of the circle represents the value that is specified by *max-value*. By default, the value for STARMAX= is the maximum chart statistic value.

STARMIN=*min-value*
scales the chart so that the center of the circle represents the value that is specified by *min-value*. By default, STARMIN=0. If the chart statistic has negative values, by default the value for STARMIN= is the minimum chart statistic value.

SUMVAR=*summary-variable*
specifies a numeric variable for sum or mean calculations. The GCHART procedure calculates the sum or, if requested, the mean of the value of *numeric-variable* for each midpoint. The resulting statistics are represented by the size of the slice and displayed beside each slice.

When you use SUMVAR=, the TYPE= option must be either SUM or MEAN. With SUMVAR=, the default is TYPE=SUM.

Featured in: Example 13 on page 620

TYPE=*statistic*
specifies the chart statistic.

- If the SUMVAR= option is not used, *statistic* can be one of the following:

  FREQ
  frequency (the default)

PERCENT PCT
percentage

If SUMVAR= is used, *statistic* can be one of the following:

SUM
sum (the default)

MEAN
mean

Because you cannot use TYPE=FREQ or TYPE=PERCENT with SUMVAR=, you must use FREQ= to calculate percentages or frequencies based on a sum. See also "Calculating Weighted Statistics" on page 529.

See also: "About Chart Statistics" on page 528

VALUE=ARROW | INSIDE | NONE | OUTSIDE
controls the position and style of the slice value (chart statistic) for each slice. For a description of the option values, see "Selecting and Positioning Spine and Slice Labels" on page 580. By default, VALUE=OUTSIDE (the value is outside of the slice).

WOUTLINE=*slice-outline-width*
specifies the width of the outline in pixels. WOUTLINE= affects both the slice and the subgroup outlines.

Selecting and Positioning Spine and Slice Labels

By default, each spine or slice is labeled with its midpoint value and its chart statistic value, which are printed outside of the circle. You can control where and how these labels are displayed with the SLICE= and VALUE= options, respectively. In addition, each spine can display the percentage that its midpoint contributes to the total chart statistic (spine percent). Use the PERCENT= option to request spine percent.

The SLICE=, VALUE=, and PERCENT= options use the same values:

ARROW
places the text outside of the star circle and connects the text to the circle with a line. The line points to the spine or the center of the slice. The arrow uses the color that is specified by CTEXT= in the STAR statement. If you omit CTEXT=, the arrow uses the first color in the colors list.

INSIDE
places the text inside the star circle.

NONE
suppresses the text.

OUTSIDE
places the text outside the star circle.

Figure 13.13 on page 571 illustrates these values.

The SLICE= and VALUE= options are dependent on each other. If you specify only VALUE= or only SLICE=, the other option automatically uses the same labeling method. PERCENT= is independent of these two.

Be careful about the combinations that you specify. For example, if you specify PERCENT=ARROW and VALUE=OUTSIDE, the line that connects the percentage information to each spine may overlay the statistic value.

Selecting Patterns for the Star Charts

Star charts are always patterned by midpoint.

Default patterns and outlines Each slice in a star chart is filled with a pattern. By default, the procedure

- fills the slices with pie/star patterns, beginning with the default fill, PSOLID, and rotating through the colors in the colors list. When the solid patterns are exhausted, the procedure selects the next default pie/start pattern and rotates it through the colors list. It continues in this fashion until all the required patterns have been assigned.

  If you use the device's default colors and the first color in the list is either black or white, the procedure does not create a pattern in that color. If you specify a colors list with the COLORS= graphics option, the procedure uses all of the colors in the list to generate the patterns.
- outlines slices using the first color in the colors list. To change the outline color, use the COUTLINE= option.

See "About Patterns" on page 529 for more information on how the GCHART procedure assigns default patterns and outlines.

Controlling patterns You can control slice patterns and their outlines in several ways.

- To select a different fill for the slices, such as empty or hatched, you can
  - request a single hatched fill pattern for all slices by specifying the FILL=X option on the STAR statement. The pattern that is specified by FILL=X rotates through the colors list as many times as needed to generate all the patterns required by the chart. If you specify a single color with either CFILL= or the graphics option, CPATTERN=, all slices use the same color as well as the same pattern.
  - specify a pattern with the VALUE= option in the PATTERN statement. Only pie/star patterns are valid; all other pattern specifications are ignored. For a complete description of all pie/star patterns, see VALUE= on page 217 in "PATTERN Statement" on page 211.

    If no color options are specified, the procedure rotates each specified fill once through the colors list. Otherwise the PATTERN statement generates one pattern definition for the specified pattern and color. When all of the specified patterns are exhausted, the procedure starts rotating through the default pie/star patterns, beginning with PSOLID.
- To select colors for the slices, you can
  - specify a single pattern color with the CFILL= option, or with the CPATTERN= graphics option, or with a colors list of one color. If you use CFILL=, the procedure starts with the default solid color and uses the foreground color for outlines. If you use CPATTERN= or a colors list of one color, the procedure skips the default solid fill and, beginning with P2N0, uses each default pie/star hatch pattern with the specified color, and uses the fill color for the outline color.
  - specify only COLOR= in one or more PATTERN statements. In this case, the procedure creates a solid pattern for each specified color. When it runs out of PATTERN statements, it returns to the default patterns, beginning with PSOLID, and rotates them each through the colors list. Whenever you specify a PATTERN statement, the default outline color is SAME.
- To define specific patterns and colors for the slices, use PATTERN statements and specify both the VALUE= and COLOR= options. If you provide fewer PATTERN definitions than the chart requires, the GCHART procedure uses the default pattern rotation for the slices that are drawn after all defined patterns are exhausted.

Whenever you use PATTERN statements, the default outline color changes to SAME. That is, the outline color is the same as the fill color. To change the outline color, use the COUTLINE= on page 562 option.

See “About Patterns” on page 529 for more information on how the GCHART procedure uses patterns and outlines. See “PATTERN Statement” on page 211 for a description of default pie/star patterns.

Modifying the Statistic Heading and the Group Heading

By default,the procedure prints a heading at the top of each chart indicating the type of statistic charted and the name of the chart variable– for example, **SUM of SALES by SITE**You can suppress this heading with the NOHEADING option.

When you use the GROUP= option, a heading is printed above each star indicating the name of the group variable and its value for the particular star– for example, **SITE=Paris**You can suppress these headings with the NOGROUPHEADING option. You can also suppress the variable name *SITE*= so that only the value *Paris* remains. To do this, use a LABEL statement and assign a null value to the variable name, as shown in this example:

```
label site='00'x;
```

Because the AXIS statement cannot be used by the STAR statement, you should use the FTEXT= and HTEXT= graphics options to control the font and height of text on the chart. Increasing the value of HTEXT= decreases the size of the star if any slice labels are positioned outside. For a description of these graphics options, see Chapter 9, “Graphics Options and Device Parameters Dictionary,” on page 301.

Examples

Example 1: Specifying the Sum Statistic in a Block Chart

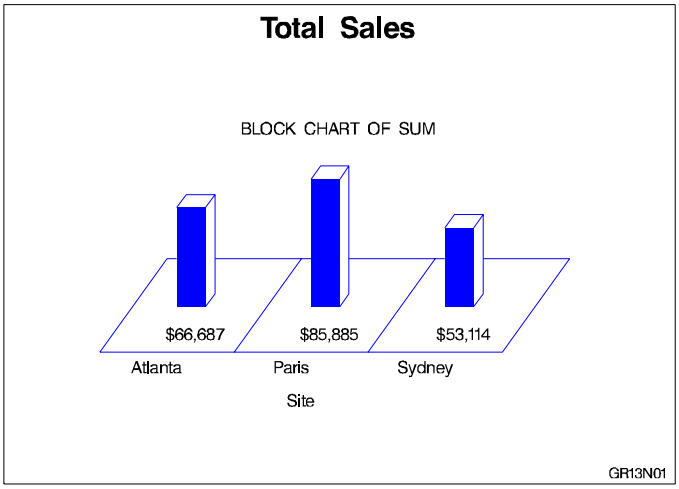
Procedure features:

BLOCK statement option:

SUMVAR=

Other features: FORMAT statement

Sample library member: GR13N01



This example produces a block chart of total sales for three sites by charting the values of the character variable SITE and calculating the sum of the variable SALES for each site. It prints formatted values of the sales statistics below the blocks.

The chart uses default patterns and colors. The block faces use the default pattern fill, which is solid. Because a colors list is specified in the GOPTIONS statement, the default fill color is the first color in the list, blue. The midpoint grid and the block outlines also use the first color in the list.

All the blocks use the same pattern because by default patterns change for subgroups and in this chart subgroups are not specified.

Assign the libref and set the graphics environment. CTEXT= specifies the color for all text on the output. COLORS= specifies the colors list, which is used by the default patterns and outlines.

```
libname reflib 'SAS-data-library';
goptions reset=global gunit=pct border cback=white
         ctext=black colors=(blue green red)
         ftext=swiss ftitle=swissb
         htitle=6 htext=3.5;
```

Create data set TOTALS. TOTALS contains quarterly sales data for three manufacturing sites for one year. Sales figures are broken down by department.

```
data reflib.totals;
   length dept $ 7 site $ 8;
   input dept site quarter sales;
   datalines;
Parts   Sydney  1 7043.97
Parts   Atlanta 1 8225.26
Parts   Paris   1 5543.97
...more data lines...
Tools   Sydney  4 1775.74
Tools   Atlanta 4 3424.19
Tools   Paris   4 6914.25
;
```

Define title and footnote.

```
title 'Total Sales';
footnote j=r 'GR13N01 ';
```

Produce the block chart. The BLOCK statement produces a block chart. SUMVAR= calculates the sum of SALES for each value of the chart variable SITE. With SUMVAR= the default statistic is SUM. The summary variable SALES is assigned a dollar format.

```
proc gchart data=reflib.totals;
   format sales dollar8.;
   block site / sumvar=sales;
run;
quit;
```

Example 2: Grouping and Subgrouping a Block Chart

Procedure features:

BLOCK statement options:

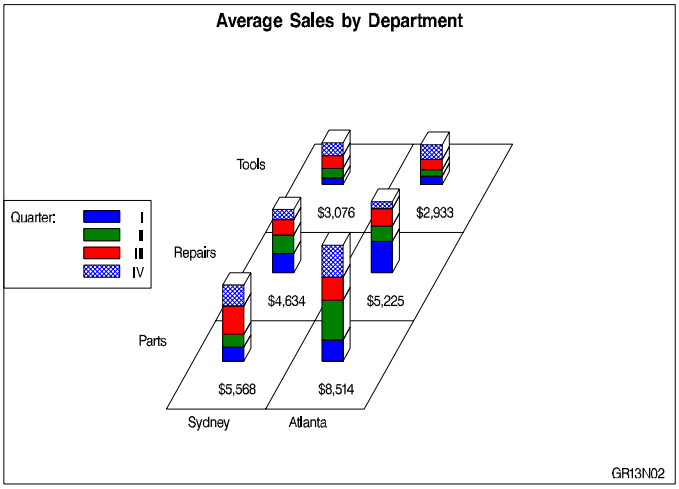
CAXIS=
COUTLINE=
GROUP=
LEGEND=
MIDPOINTS=
NOHEADING
SUBGROUP=
TYPE=

Other features:

LABEL statement
LEGEND statement
Default pattern rotation

Data set: TOTALS

Sample library member: GR13N02



This example shows average quarterly sales for each department at two of the three manufacturing sites in the TOTALS data set; it excludes the Paris site from the chart.

The program groups the chart data (sites) by department, and subgroups department sales data by quarter. Each site is a midpoint. Because the sites are grouped by department, each midpoint has a separate square for each department and the height of the block represents total sales for that department.

The blocks are subgrouped to show how quarterly sales contribute to total sales; each segment represents sales for a quarter. A legend explaining the subgroup patterns appears below the midpoint grid.

The subgroups use four default patterns, The first three patterns are created by rotating the first default fill, solid, through the three colors in the colors list defined in the GOPTIONS statement. The fourth default pattern is created by using the second default pattern fill, X1, with the first color in the colors list, blue.

Because the first color in the colors list is also the default color for several other elements, the program includes several options that override the default: CTEXT= colors all text, CAXIS= colors the midpoint grid, COUTLINE= colors the pattern outline. For more information on patterns and colors, see “Controlling Block Chart Patterns and Colors” on page 539.

Assign the libref and set the graphics environment. COLORS= specifies a colors list that is used by the default patterns. CTEXT= specifies black for all text.

```
libname reflib 'SAS-data-library';
goptions reset=global gunit=pct border cback=white
         colors=(blue green red) ctext=black
         ftitle=swissb ftext=swiss htitle=4 htext=3;
```

Define title and footnote.

```
title 'Average Sales by Department';
footnote j=r 'GR13N02  ';
```

Define legend characteristics. LABEL= assigns new text to the legend label. CBORDER= draws a black frame around the legend.

```
legend1 cborder=black
        label=('Quarter:')
        position=(middle left outside)
        mode=protect
        across=1;
```

Produce the block chart. The LABEL statement suppresses the midpoint and group labels by assigning a null hexadecimal string to each variable name.

```
proc gchart data=reflib.totals;
   format quarter roman.;
   format sales dollar8.;
   label site='00'x dept='00'x;
```

TYPE= specifies the chart statistic as the mean value of the summary variable SALES for each site. MIDPOINTS= selects the two sites and the order in which they appear. GROUP= creates a separate row of blocks for each different value of DEPT. SUBGROUP= divides each block into separate segments for the four quarters. LEGEND= assigns the LEGEND1 statement to the graph. NOHEADING suppresses the default heading that would otherwise appear above the chart. CAXIS= colors the grid black. COUTLINE= specifies the outline color for the blocks.

```
   block site / sumvar=sales
                type=mean
                midpoints='Sydney' 'Atlanta'
                group=dept
                subgroup=quarter
                legend=legend1
                noheading
                coutline=black
                caxis=black;
run;
quit;
```

Example 3: Specifying the Sum Statistic in Bar Charts

Procedure features:

HBAR statement options:

SUMVAR=

VBAR3D statement options:

SUMVAR=
COUTLINE=

Other features:

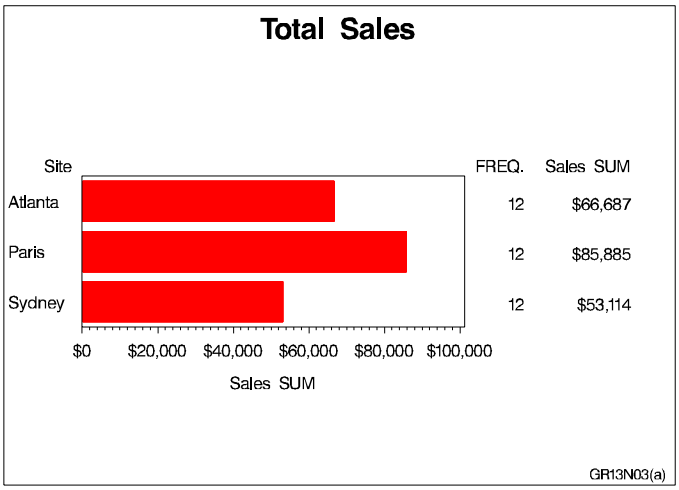
FORMAT statement

PATTERN statement

RUN-group processing

Data set: TOTALS

Sample library member: GR13N03



This example produces two bar charts that show total sales for three sites by charting the values of the character variable SITE and calculating the sum of the variable SALES for each site.

In the horizontal bar chart shown above, the summary statistics are printed by default to the right of the bars and display the formatted values of SALES.

The bars use the default pattern fill, which is solid. Because a colors list is specified in the GOPTIONS statement, the first default pattern color is the first color in the list. To avoid having black bars, the program uses a PATTERN statement to specify the pattern color. Using a PATTERN statement causes the default bar outline color to match the fill color. All the bars display the same pattern because by default patterns change for subgroups and in this chart subgroups are not specified.

The output also shows the frame that is drawn by default around the axis area.

The second bar chart is a 3D vertical bar chart, shown in the following output. Vertical bar charts do not generate a table of statistics and by default do not print any chart statistics. This chart uses the same pattern as the horizontal bar chart, but the VBAR3D statement specifies a black outline for the bars.

Assign the libref and set the graphics environment. COLORS= specifies the colors list, which is used by the default patterns and outlines.

```
libname reflib 'SAS-data-library';
goptions reset=global gunit=pct border
         colors=(black red blue green)
         cback=white ftitle=swissb ftext=swiss
         htitle=6 htext=3.5;
```

Define title and footnote for the first chart.

```
title1 'Total Sales';
footnote1 h=3 j=r 'GR13N03(a) ';
```

Specify a color for the pattern. The PATTERN statement explicitly defines RED as the color for the first solid pattern.

```
pattern1 color=red;
```

Produce the horizontal bar chart. The HBAR statement produces a two-dimensional bar chart. SUMVAR= calculates the sum of SALES for each value of the chart variable SITE. The default statistic for SUMVAR= is SUM. The summary variable SALES is assigned a dollar format.

```
proc gchart data=reflib.totals;
   format sales dollar8.;
   hbar site / sumvar=sales;
run;
```

Produce the vertical bar chart. Because the procedure supports RUN-group processing, you do not have to repeat the PROC GCHART statement to generate the second chart. The VBAR3D statement produces a three-dimensional vertical bar chart. The FOOTNOTE1 statement replaces the previous footnote. COUTLINE= assigns a black outline to the bars.

```
   footnote1 h=3 j=r 'GR13N03(b) ';
   vbar3d site / sumvar=sales
                 coutline=black;
run;
quit;
```

Example 4: Subgrouping a 3D Vertical Bar Chart

Procedure features:

VBAR statement options:

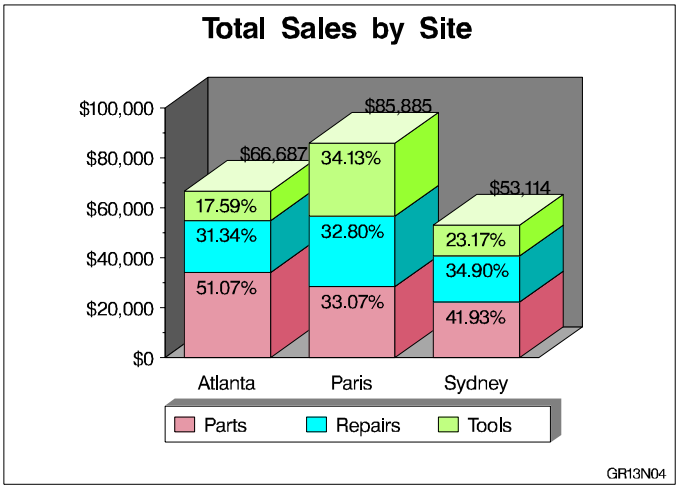
CFRAME=
INSIDE=SUBPCT
LEGEND=
MAXIS=
OUTSIDE=SUM
RAXIS=
SPACE=
SUBGROUP=
WIDTH=

Other features:

AXIS statement
FORMAT statement
GOPTIONS statement
OFFSHADOW=
LEGEND statement
PATTERN statement

Data set: TOTALS

Sample library member: GR13N04



This example subgroups by department the 3D vertical bar chart of total sales for each site that is shown in Example 3 on page 586. In addition to subdividing the bars to show the amount of sales for each department for each site, the chart displays statistics both inside and outside of the bars. OUTSIDE=SUM prints the total sales for the site above each bar. INSIDE=SUBPCT prints the percent each department contributed to the total sales for its site inside of each subgroup segment.

The legend has a block-effect shadow whose color matches the backplane. The graphics option OFFSHADOW= defines the size and position of the block shadow. Both the LEGEND statement and the AXIS statement use the ORIGIN= option to line up the legend and the chart by explicitly positioning their lower left corners.

Assign the libref and set the graphics environment. OFFSHADOW= defines the depth of the block around the legend box. The positive values position the shadow above and to the right of the legend.

```
libname reflib 'SAS-data-library';
goptions reset=global gunit=pct border cback=white
         colors=(black red green blue) ftitle=swissb
         ftext=swiss htitle=6 htext=4
         offshadow=(1.5,1.5);
```

Define title and footnote.

```
title1 'Total Sales by Site';
footnote1 h=3 j=r 'GR13N04 ';
```

Modify the midpoint axis. LABEL= suppresses the axis label. ORIGIN= positions the left end of the horizontal axis at a point that is 25% of the width of the graphics output area.

```
axis1 label=none
      origin=(24,);
```

Modify the response axis. ORDER= specifies the major tick values for the response axis. OFFSET= moves the top tick mark to the end of the axis line.

```
axis2 label=none
      order=(0 to 100000 by 20000)
      minor=(number=1)
      offset=(,0);
```

Modify the legend. LABEL= suppresses the legend label. SHAPE= defines the size of the legend values. CBORDER= draws a black frame around the legend. CBLOCK= specifies a gray block that matches the 3D planes. ORIGIN= specifies the same position as in the AXIS1 statement.

```
legend1 label=none
        shape=bar(3,3)
        cborder=black
        cblock=gray
        origin=(24,);
```

Define pattern characteristics. PATTERN statements define the colors that are assigned to subgroups. Light colors allow the black labels to show up. Default pattern fill is solid.

```
pattern1 color=lipk;
pattern2 color=cyan;
pattern3 color=lime;
```

Produce the vertical bar chart. SUBGROUP= creates a separate bar segment for each department. INSIDE= prints the subgroup percent statistic inside each bar segment. OUTSIDE= prints the sum statistic above each bar. WIDTH= makes the bars wide enough to display the statistics. SPACE= controls the space between the bars. MAXIS= assigns the AXIS1 statement to the midpoint axis. RAXIS= assigns the AXIS2 statement to the response axis. LEGEND= assigns the LEGEND1 statement to the subgroup legend. CFRAME= specifies the color for the 3D planes.

```
proc gchart data=reflib.totals;
   format quarter roman.;
   format sales dollar8.;
   vbar3d site / sumvar=sales
                 subgroup=dept
                 inside=subpct
                 outside=sum
                 width=9
                 space=4
                 maxis=axis1
                 raxis=axis2
                 cframe=gray
                 coutline=black
                 legend=legend1;
run;
quit;
```

Example 5: Controlling Midpoints and Statistics in a Horizontal Bar Chart

Procedure features:

HBAR statement options:

AUTOREF
COUTLINE=
CLIPREF
SUBGROUP=

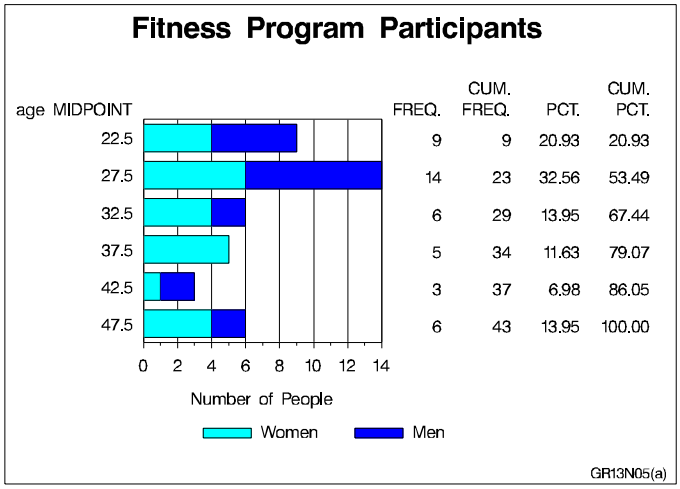
HBAR3D statement options:

FREQ
FREQLABEL=
MIDPOINTS=

Other features:

AXIS statement
LEGEND statement
PATTERN statement
RUN-group processing

Sample library member: GR13N05



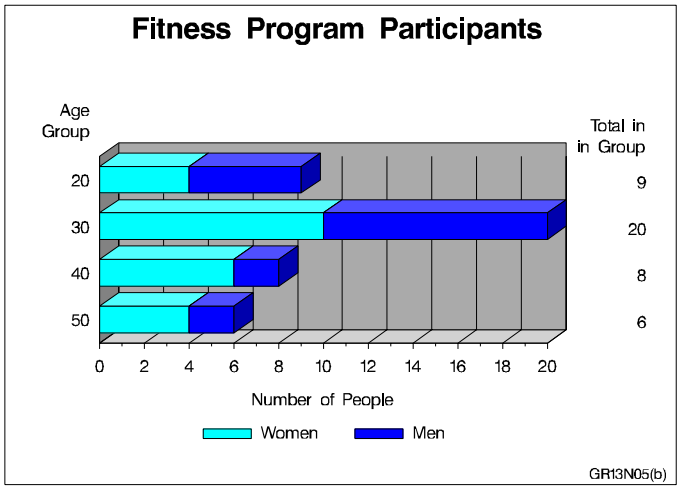
This example uses the FITNESS data set to produce a horizontal bar chart that shows the number of people in each age group in a fitness program.

It charts the numeric variable AGE, with the frequency statistic. Because the values of AGE are continuous, the procedure automatically divides the ages into ranges and displays the midpoint of each age range. The frequency statistic calculates the number of observations in each range. The chart statistic defaults to FREQ because the SUMVAR= and TYPE= options are omitted. The table of statistics displays all the statistic values.

The program also subgroups each age group bar to show the number of men and women in the group. Because the default value for the PATTERNID= option is SUBGROUP, the procedure automatically assigns a different pattern to each subgroup and the PATTERNID= option is unnecessary.

PATTERN statements specify the colors for the subgroups. Whenever the GCHART procedure uses PATTERN statements, the default outline color of the bars changes from black to the color of the bar. Because this program uses PATTERN statements, it also uses COUTLINE= to specify a black outline for the bars.

The second part of this example modifies the midpoint axis and the table of statistics, and uses RUN-group processing to produce the following chart. This part of the program specifies the midpoint value for each bar and requests only the FREQ statistic for the table.



Assign the libref and set the graphics environment. Black is the first color in the colors list and, by default, is used for all text and for the axis lines and frame. Therefore, it is not necessary to use CTEXT= (GOPTIONS statement) and CAXIS= (HBAR statement) to specify a color.

```
libname reflib 'SAS-data-library';
goptions reset=global gunit=pct border cback=white
         colors=(black blue green red) ftext=swiss
         ftitle=swissb htitle=6 htext=3.5;
```

Create the data set FITNESS. FITNESS contains age and sex of participants, as well as the number of times they exercise each week and their resting heart rate and aerobic power.

```
data reflib.fitness;
   input age sex $ heart exer aero;
   datalines;
28  M  86  2   36.6
41  M  76  3   26.7
30  M  78  2   33.8
...more data lines...
29  M  54  3   44.8
48  F  66  2   28.9
36  F  66  2   33.2
;
```

Define the title and footnote.

```
title1 'Fitness Program Participants';
footnote h=3 j=r 'GR13N05(a) ';
```

Modify the response axis. OFFSET= moves the first and last tick marks to the ends of the axis line.

```
axis1 label=('Number of People')
      minor=(number=1)
      offset=(0,0);
```

Modify the legend. VALUE= specifies the text that describes the values.

```
legend1 label=none
        value=('Women' 'Men');
```

Define pattern colors for the subgroups. The procedure automatically assigns a pattern to each subgroup, using the default fill, SOLID, with the specified color.

```
pattern1 color=cyan;
pattern2 color=blue;
```

Produce the first horizontal bar chart. Because neither MIDPOINTS= nor DISCRETE is used, the procedure automatically selects the midpoints. SUBGROUP= divides the bars according to the values of SEX and automatically generates a legend. AUTOREF adds reference lines to the chart at each major tick mark. CLIPREF positions the reference lines behind the bars. COUTLINE= specifies the outline color for the bars.

```
proc gchart data=reflib.fitness;
   hbar age / subgroup=sex
              legend=legend1
              autoref
              clipref
              coutline=black
              raxis=axis1;
run;
```

Define the footnote for the second chart.

```
footnote h=3 j=r 'GR13N05(b) ';
```

Modify the response axis for the second chart. ORDER= places major tick marks on the response axis at intervals of 2.

```
axis1 order=(0 to 20 by 2)
      label=('Number of People')
      minor=(number=1)
      offset=(0,0);
```

Modify the midpoint axis label for the second chart.

```
axis2 label=('Age ' j=r 'Group');
```

Produce the second horizontal bar chart with modified midpoints. MIDPOINTS= specifies the middle value of the range of values represented by each bar. FREQ requests that only the frequency statistic appears in the table. FREQLABEL= specifies the text for the column header in the table of statistics.

```
   hbar3d age / midpoints=(20 30 40 50)
                freq
                freqlabel='Total in Group'
                subgroup=sex
                autoref
                maxis=axis2
                raxis=axis1
                legend=legend1
                coutline=black;
run;
quit;
```

Example 6: Generating Error Bars in a Horizontal Bar Chart

Procedure features:

HBAR statement options:

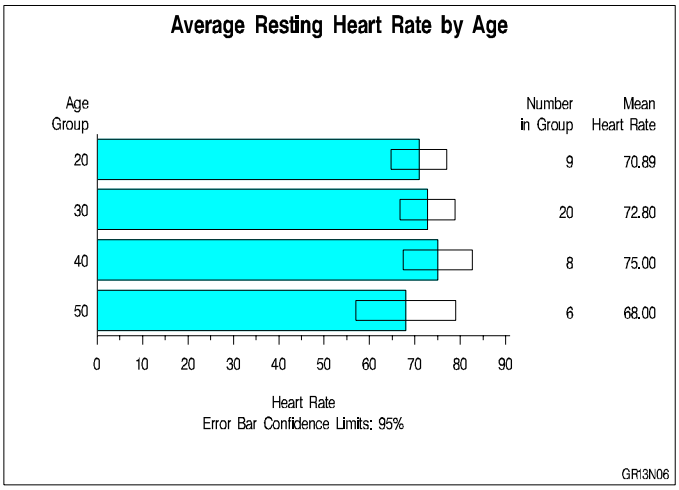
CLM=
COUTLINE=
ERRORBAR=
FREQLABEL=
MEANLABEL=
NOFRAME
SUMVAR=
TYPE=

Other features:

AXIS statement
PATTERN statement

Data set: FITNESS

Sample library member: GR13N06



This example uses the FITNESS data set to chart the mean heart rate for each age group with error bars showing the confidence limits for the average. The response axis label describes the confidence limit for the error bars. To make the error bars easier to read, the program suppresses the frame that the procedure draws around the axis area. Descriptive column head labels in the table of statistics replace the statistic names that appear by default.

Assign the libref and set the graphics environment.

```
libname reflib 'SAS-data-library';
goptions reset=global gunit=pct border cback=white
         colors=(black blue green red) ftext=swiss
         ftitle=swissb htitle=5 htext=3.5;
```

Define the title and footnote.

```
title1 'Average Resting Heart Rate by Age';
footnote h=3 j=r 'GR13N06 ';
```

Modify the axis labels. AXIS1 is assigned to the response axis and AXIS2 is assigned to the midpoint axis.

```
axis1 label=('Heart Rate' j=c
             'Error Bar Confidence Limits: 95%')
      minor=(number=1);
axis2 label=('Age' j=r 'Group');
```

Define a color for the bars. The PATTERN statement uses the default fill, SOLID, with the specified color to create a pattern for the bars. Using a PATTERN statement causes the default bar outline color to be the same as the fill color. COUTLINE= in the HBAR statement assigns a black outline.

```
pattern1 color=cyan;
```

Produce the horizontal bar chart. SUMVAR= calculates the mean of the variable HEART for all the observations in each midpoint group. TYPE= specifies the mean statistic for the summary variable, HEART. FREQLABEL= and MEANLABEL= specify new column labels for the frequency and mean statistics. ERRORBAR= draws the error bars as empty bars and CLM= specifies the confidence level. COUTLINE= outlines the bars in black. NOFRAME suppresses the axis area frame.

```
proc gchart data=reflib.fitness;
   hbar age / type=mean
              sumvar=heart
              freqlabel='Number in Group'
              meanlabel='Mean  Heart Rate'
              errorbar=bars
              clm=95
              midpoints=(20 30 40 50)
              raxis=axis1
              maxis=axis2
              noframe
              coutline=black;
run;
quit;
```

Example 7: Creating Bar Charts with Drill-down for the Web

Procedure Features:

VBAR3D statement

ODS Features:

ODS HTML statement:

ANCHOR=
BODY=
CONTENTS=
FRAME=
NEWFILE
NOGTITLE
PATH=

Other Features:

AXIS statement
BY statement
FORMAT statement
GOPTIONS statement
LEGEND statement
PATTERN statement
RUN-group processing
TITLE statement
WHERE statement

Sample library member: GR13N07

This example shows how to create 3D bar charts with drill-down functionality for the Web. In addition to showing how to use the ODS HTML statement and the HTML options to create the drill-down, the example also illustrates other VBAR3D statement options.

For creating output with drill-down for the Web, the example shows how to

- explicitly name the HTML files and open and close them throughout the program
- specify names and destination for the GIF files created by the ODS HTML statement and the GIF device driver
- assign anchor names to the graphics output
- use the HTML= and HTML_LEGEND= procedure options to assign link targets
- use BY-group processing to store multiple graphs in one file or in individual files
- use incremented anchor names and incremented file names.

For more information, see “ODS HTML Statement” on page 200 in Chapter 8, “SAS/GRAPH Statements,” on page 159.

For creating 3D bar charts, the example shows how to

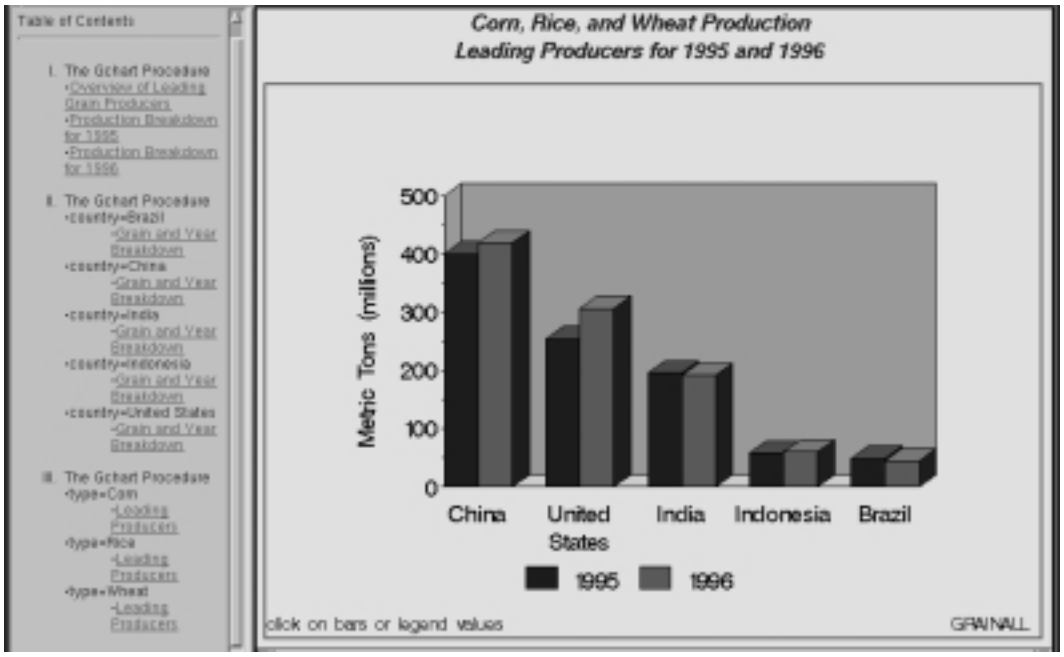
- group the midpoints, including patterning bars by group, modifying the group axis, adjusting the space between groups of bars
- identify midpoint values with a legend instead of labeling each bar
- subgroup bars
- remove an axis and its axis plane
- add reference lines.

The introduction to each part lists the VBAR3D options that it features.

The program generates twelve linked bar charts that display data about the world's leading grain producers. The data contain the amount of grain produced by five countries in 1995 and 1996. Each of these countries is one of the three leading producers of wheat, rice, or corn, worldwide.

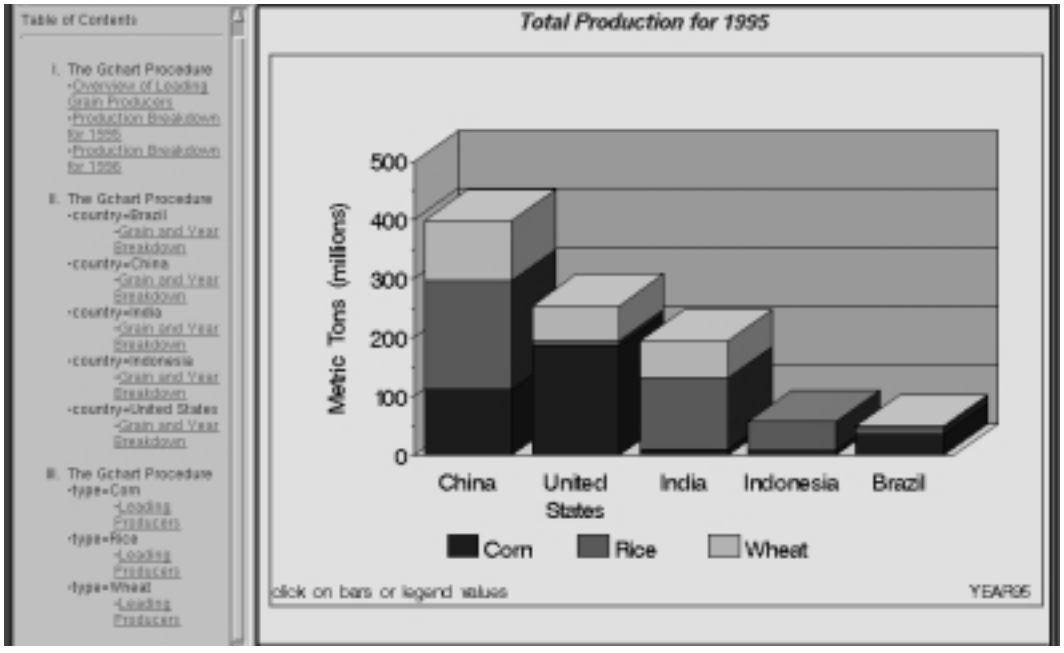
The first chart, shown in Figure 13.14 on page 598 as it appears in a browser, is an overview of the data that shows the total grain production for the five countries for both years.

Figure 13.14 Browser View of Overview Graph



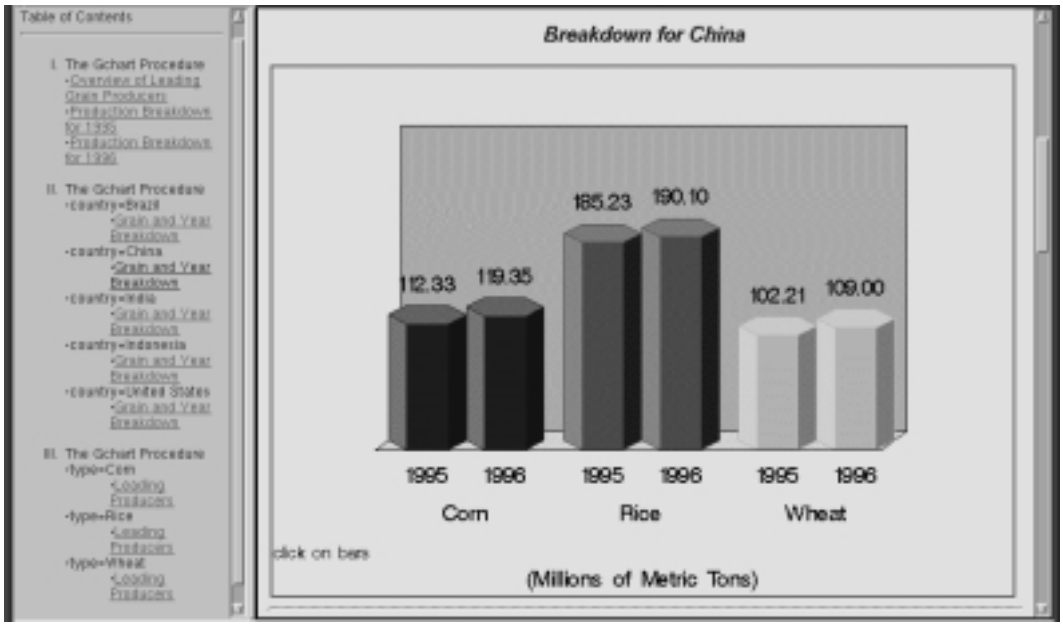
The next two charts break down grain production by year. These charts are linked to the legend values in Figure 13.14 on page 598. For example, when you select the legend value for 1995, the graph in Figure 13.15 on page 598 appears.

Figure 13.15 Browser View of Year Breakdown for 1995



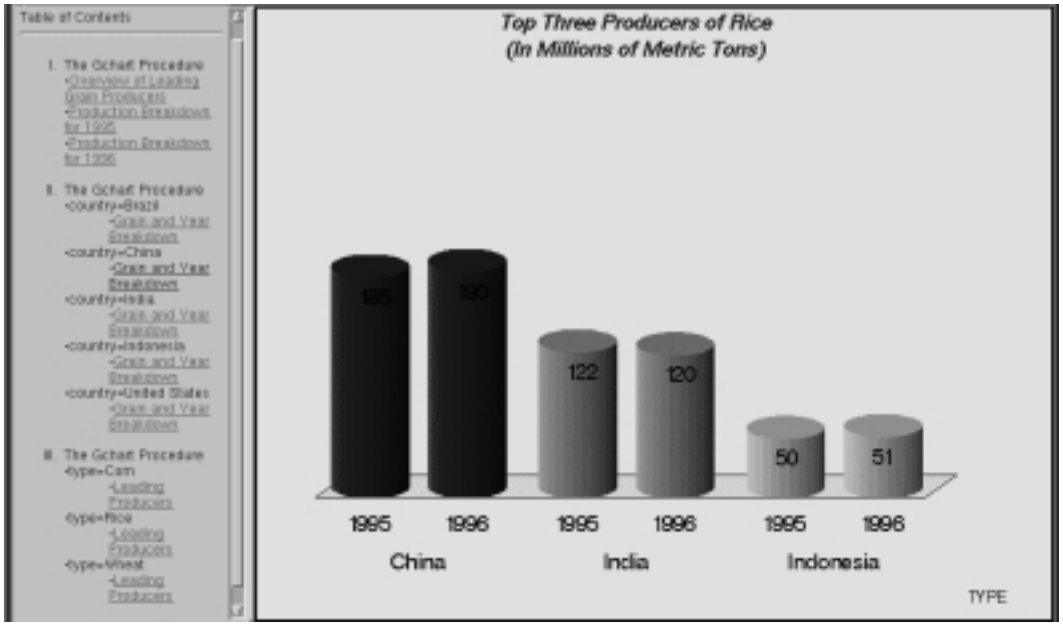
Another group of charts breaks down the data by country. These charts are linked to the bars. For example, when you drill down on the bar for China in either Figure 13.14 on page 598 or Figure 13.15 on page 598, the graph in Figure 13.16 on page 599 appears.

Figure 13.16 Browser View of Breakdown for China



Finally the data is charted by grain type. These graphs are linked to the bars in Figure 13.16 on page 599. If you select the legend value or bar for **`Rice`**, Figure 13.17 on page 599 appears.

Figure 13.17 Browser View of Breakdown for Rice



This program is divided into four parts:

- "Example 7, Part A" on page 600 generates the graph shown in Figure 13.14 on page 598.

- "Example 7, Part B" on page 605 generates the pair of graphs represented by Figure 13.15 on page 598.
- "Example 7, Part C" on page 606 generates the five graphs represented by Figure 13.16 on page 599.
- "Example 7, Part D" on page 608 generates the three graphs represented by Figure 13.17 on page 599.

Example 7, Part A

Features: VBAR3D options:

DES=
DISCRETE
GROUP=
GSPACE=
HTML=
HTML_LEGEND=
NAME=
SUBGROUP=

ODS HTML options:

BODY=
CONTENTS=
FRAME=
GPATH=
NOGTITLE

The first part of the program, which includes setting the graphics environment and creating the data set, does the following:

- Adds three HTML variables to the data set. The variables contain the link targets for all of the graphs that support drill-down functionality. The HREF values for the HTML variables in the data set contain this information about the link targets:
  - the name of the body file that is the target. BODY= in the ODS HTML statement names the body file.
  - the anchor name of the output if the target file contains more than one graph. By default, all output is assigned a unique anchor name unless you specify a name with ANCHOR= in the ODS HTML statement.
- Opens the HTML destination for the frame and contents files and the first body file.
- Creates one grouped 3D vertical bar chart (shown in Figure 13.14 on page 598) with drill-down on the bars and legend values. The bars, which represent total production for each year for each country, are grouped and labeled by COUNTRY. Instead of displaying the year below each bar, the program suppresses the midpoint values with an AXIS statement and creates a legend that associates bar color and year. To create the legend, the chart variable YEAR is assigned to the SUBGROUP= option. Because the chart variable and the subgroup variable are the same, each bar contains only one "subgroup." As a result, the subgroup legend has an entry for each value of YEAR, thereby creating a legend for the midpoints. The values of COUNTRY label each group of bars.
- Assigns the HTML variables that contain link information for the bars and for the legend values to the HTML= and HTML_LEGEND= options, respectively.

Assign the Web-server path. FILENAME assigns the fileref ODSOUT, which specifies a destination for the HTML and GIF files produced by the example program. To assign that location as the HTML destination for program output, ODSOUT is specified later in the program on the ODS HTML statement's PATH= option. ODSOUT must point to a Web-server location if procedure output is to be viewed on the Web.

```
filename odsout 'path-to-Web-server-space';
```

Close the ODS listing destination for procedure output, and set the graphics environment. To conserve system resources, the graphics output is not displayed in the GRAPH window, although it is written to the graphics catalog and to the GIF files.

```
ods listing close;
goptions reset=global gunit=pct
         htitle=6 htext=4 ftitle=zapfb ftext=swiss;
```

Create the data set GRAINLDR. GRAINLDR contains data about grain production in five countries for 1995 and 1996. The quantities in AMOUNT are in thousands of metric tons. MEGTONS converts these quantities to millions of metric tons.

```
data grainldr;
   length country $ 3 type $ 5;
   input year country $ type $ amount;
   megtons=amount/1000;
   datalines;
1995 BRZ  Wheat    1516
1995 BRZ  Rice     11236
1995 BRZ  Corn     36276
1995 CHN  Wheat    102207
1995 CHN  Rice     185226
1995 CHN  Corn     112331
1995 IND  Wheat    63007
1995 IND  Rice     122372
1995 IND  Corn     9800
1995 INS  Wheat    .
1995 INS  Rice     49860
1995 INS  Corn     8223
1995 USA  Wheat    59494
1995 USA  Rice     7888
1995 USA  Corn     187300
1996 BRZ  Wheat    3302
1996 BRZ  Rice     10035
1996 BRZ  Corn     31975
1996 CHN  Wheat    109000
1996 CHN  Rice     190100
1996 CHN  Corn     119350
1996 IND  Wheat    62620
1996 IND  Rice     120012
1996 IND  Corn     8660
1996 INS  Wheat    .
```

```
1996 INS  Rice     51165
1996 INS  Corn     8925
1996 USA  Wheat    62099
1996 USA  Rice     7771
1996 USA  Corn     236064
;
```

Add three HTML variables to GRAINLDR to create the NEWGRAIN data set. Each HTML variable is assigned the targets for a certain variable value. These targets are specified by the HREF attribute within an AREA element in the HTML file. Each HREF value specifies the HTML body file and, optionally, the name of the anchor within the body file that identifies the target graph. The HTML variable YEARDRILL contains the targets for the values of the variable YEAR.

```
data newgrain;
     set grainldr;
     length yeardrill typedrill countrydrill $ 40;
     if year=1995 then
        yeardrill='HREF="year95_body.html"';
     else if year=1996 then
        yeardrill='HREF="year96_body.html"';
```

The HTML variable COUNTRYDRILL contains the targets for the values of the variable COUNTRY. Because the graphs of COUNTRY are in one file, the targets must include the anchor name.

```
     if country='BRZ' then
        countrydrill='HREF="country_body.html#country"';
     else if country='CHN' then
        countrydrill='HREF="country_body.html#country1"';
     else if country='IND' then
        countrydrill='HREF="country_body.html#country2"';
     else if country='INS' then
        countrydrill='HREF="country_body.html#country3"';
     else if country='USA' then
        countrydrill='HREF="country_body.html#country4"';
```

The HTML variable TYPEDRILL contains the names of the files that are the targets for the values of the variable TYPE.

```
     if type='Corn' then
        typedrill='HREF="type1_body.html"';
     else if type='Rice' then
        typedrill='HREF="type2_body.html"';
     else if type='Wheat' then
        typedrill='HREF="type3_body.html"';
run;
```

Create a format for the values of COUNTRY.

```
proc format;
     value $country 'BRZ' = 'Brazil'
                    'CHN' = 'China'
                    'IND' = 'India'
                    'INS' = 'Indonesia'
                    'USA' = 'United States';
run;
```

Define pattern colors for all graphs. To avoid solid black patterns (BLACK is the first color in the colors list), explicitly assign the pattern colors.

```
pattern1 color=blue;
pattern2 color=green;
pattern3 color=cyan;
```

Define legend characteristics for all legends. OFFSET= moves the legend down.

```
legend1 label=none
        shape=bar(4,4)
        position=(bottom center)
        offset=(-3);
```

Assign the GOPTIONS for ODS HTML destination. DEVICE= generates the SAS/GRAPH output as GIF files. TRANSPARENCY makes the background of the graphs the same as the Web-page background. NOBORDER turns off the border around the graphics output area.

```
goptions transparency device=gif noborder;
```

Open the ODS HTML destination for the ODS graphics output. BODY= names the file for storing the HTML output. CONTENTS= names the HTML file that contains the table of contents to the HTML procedure output. The contents file links to each of the body files written to the HTML destination. FRAME= names the HTML file that integrates the contents and body files. PATH= specifies the ODSOUT fileref as the HTML destination for all the HTML and GIF files. NOGTITLE suppress the graph titles from the SAS/GRAPH output and displays them through the HTML page.

```
ods html body='grain_body.html'
         frame='grain_frame.html'
         contents='grain_contents.html'
         path=odsout
         nogtitle;
```

Suppress the label and values for the midpoint axis. The midpoint values 1995 and 1996 do not appear below each bar.

```
axis1 label=none value=none;
```

Modify the response axis. ANGLE=90 prints the axis label vertically.

```
axis2 label=(angle=90 'Metric Tons (millions)')
      minor=(n=1)
      order=(0 to 500 by 100)
      offset=(0,0);
```

Suppress the label and order the values for the group axis. Because the values of COUNTRY are formatted, ORDER= must specify their formatted value.

```
axis3 label=none
      order=('China' 'United States' 'India'
             'Indonesia' 'Brazil')
      split=' ';
```

Define titles and footnote. The footnote uses the catalog entry name to identify the graph.

```
title1 'Corn, Rice, and Wheat Production';
title2 h=2 'Leading Producers for 1995 and 1996';
footnote1 j=l h=3 'click on bars or legend values' j=r h=3 'GRAINALL ';
```

Generate the vertical bar chart that summarizes all grain production for all countries for both years. DISCRETE creates a separate bar for each unique value of YEAR. GROUP= groups the bars by country. To create a legend for midpoint values, SUBGROUP= is assigned the chart variable. GSPACE= controls the space between the groups of bars.

```
proc gchart data=newgrain;
     format country $country.;
     vbar3d year / discrete
                   sumvar=megtons
                   group=country
                   subgroup=year
                   legend=legend1
                   space=0
                   width=5
                   gspace=5
                   maxis=axis1
                   raxis=axis2
                   gaxis=axis3
                   cframe=grayaa
                   coutline=black
```

HTML= specifies COUNTRYDRILL as the variable that contains the targets for the bars. HTML_LEGEND= specifies YEARDRILL as the variable that contains the targets for the legend values. Specifying HTML variables causes SAS/GRAPH to add an image map to the HTML body file. NAME= specifies the name of the catalog entry. Because the PATH= destination is a file storage location and not a specific file name, the catalog entry name GRAINALL is automatically assigned to the GIF file. DES= specifies the description that is stored in the graphics catalog and used in the Table of Contents.

```
                    html=countrydrill
                    html_legend=yeardrill
                    name='grainall'
                    des='Overview of leading grain producers';
run;
```

Example 7, Part B

Features: VBAR3D options:

- AUTOREF
- HTML=
- HTML_LEGEND=
- SUBGROUP=
- SPACE=
- NAME=

ODS HTML options:

- BODY=

In the second part, the PROC GCHART step continues, using RUN-group processing and WHERE statements to produce two graphs of grain production for each year, one of which is shown in Figure 13.15 on page 598. Each bar represents a country and is subgrouped by grain type. As before, both the bars and the legend values are links to other graphs. The bars link to targets stored in COUNTRYDRILL and the legend values link to targets in TYPEDRILL. These two graphs not only *contain* links, they *are* the link targets for the legend values in Figure 13.14 on page 598. Before each graph is generated, the ODS HTML statement opens a new body file in which to store the output. Because each of these graphs is stored in a separate file, the HREF attributes that are stored in the variable YEARDRILL point only to the file. The name of the file is specified by the BODY= option in the ODS HTML statement. For example, this is the HREF attribute that points to the graph of 1995 and is stored in the variable YEARDRILL:

```
HREF=year95_body.html
```

YEARDRILL is assigned to the HTML_LEGEND= option in Part A.

Open a new body file for the graph of 1995 production. Assigning a new body file closes GRAIN_BODY.HTML. The contents and frame files, which remain open, will provide links to all body files.

```
ods html body='year95_body.html' path=odsout;
```

Define the title and footnote for the chart.

```
title1 'Total Production for 1995';
footnote1 j=l h=3 'click on bars or legend values' j=r h=3 'YEAR95 ';
```

Subset the data for 1995 and generate the vertical bar chart for 1995. AUTOREF draws a reference line on the backplane for every major tick mark value. SUBGROUP= creates a separate bar segment for each department. SPACE= controls the space between the bars. HTML= names the variable that contains the targets for the bars. HTML_LEGEND= names the variable that contains the targets for the legend values. The GIF files use the catalog entry name specified by NAME=.

```
where year=1995;
vbar3d country / sumvar=megtons
                 subgroup=type
                 autoref
                 html=countrydrill
                 html_legend=typedrill
                 legend=legend1
                 cframe=grayaa
                 space=3
                 coutline=black
                 maxis=axis3
                 raxis=axis2
                 name='year95'
                 des='Production Breakdown for 1995';
run;
```

Open a new body file for the graph of 1996 production. Assigning a new body file closes YEAR95_BODY.HTML.

```
ods html body='year96_body.html' path=odsout;
```

Define title and footnote for the second graph.

```
title1 'Total Production for 1996';
footnote1 j=l h=3 'click on bars or legend values' j=r h=3 'YEAR96 ';
```

Subset the data for 1996 and generate the vertical bar chart for 1996.

```
where year=1996;
vbar3d country / sumvar=megtons
                 subgroup=type
                 autoref
                 html=countrydrill
                 html_legend=typedrill
                 legend=legend1
                 cframe=grayaa
                 space=3
                 coutline=black
                 maxis=axis3
                 raxis=axis2
                 name='year96'
                 des='Production Breakdown for 1996';
run;
quit;
```

Example 7, Part C

Features: VBAR3D options:

DES=
GAXIS=
GROUP=
HTML=
NAME=
OUTSIDE=

PATTERNID=
RAXIS=
SHAPE=

ODS HTML options:
BODY=
ANCHOR=

The third part produces the five graphs that show the breakdowns by country. These graphs are generated with BY-group processing and are all stored in one body file. When the file is displayed in the browser, all the graphs appear in one frame that can be scrolled. Because the graphs are stored in one file, the links to them must explicitly point to the location of each graph in the file, not just to the file. This location is defined by an anchor. ODS HTML assigns anchor names by default, but you can specify anchor names with the ANCHOR= option. When the procedure uses BY-group processing to generate multiple pieces of output, ODS automatically increments the anchor name to produce a unique name for each graph. This example assigns the base name {mono country} to ANCHOR=. The graphs created by this part are referenced by the COUNTRYDRILL variable. With BY-group processing the catalog entry name also increments automatically. NAME= specifies *country* as the base name for the graphics output. Because you cannot specify a different description for each graph, DES= specifies a generic description for the HTML Table of Contents.

Sort the data set for the graphs of production by country. The data must be sorted in order of the BY variable before running PROC GCHART with BY-group processing.

```
proc sort data=newgrain out=country;
by country;
run;
```

Open a new body file and specify the base anchor name for the graphs of individual countries. Assigning a new body file closes YEAR96_BODY.HTML. Because all the graphs generated by the BY-group processing are stored in one file, each one is automatically assigned an anchor name. ANCHOR= specifies a base name for these anchors.

```
ods html body='country_body.html'
         anchor='country'
         gfootnote
         path=odsout;
```

Redefine AXIS2 to change the range of values and suppress all axis elements. Setting all the label and tick mark options to NONE and assigning a line style of 0 removes the response axis. NOPLANE removes the 3D axis plane. Specifying ORDER= makes all the graphs use the same range of values.

```
axis2 order=(0 to 250 by 50)
      label=none
      value=none
      style=0
      major=none
      minor=none
      noplane;
```

Suppress the axis label for the midpoint axis.

```
axis4 label=none;
```

Suppress the default BY-line and define a title that includes the BY-value. #BYVAL inserts the value of the BY variable COUNTRY into the title of each report.

```
options nobyline;
title1 'Breakdown for #byval(country)';
footnote1 j=l h=3 'click on bars';
footnote2 j=c '(Millions of Metric Tons)';
```

Generate the vertical bar chart of production for each country.GROUP= groups the bars by country. PATTERNID= assigns patterns by group value. SHAPE= assigns the bar shape. OUTSIDE= displays the SUM statistic above the bars. RAXIS= assigns the AXIS statement that removes all exis elements. GAXIS= assigns the AXIS statement that removes the label. HTML= specifies TYPEDRILL as the variable that contains the targets for the bars. NAME= specifies the name of the catalog entry. The graphics catalog entry name increments so the GIF files are named sequentially from COUNTRY to COUNTRY4. The DES= option specifies a general description that appears in the table of contents for all five graphs.

```
proc gchart data=country;
     format country $country.;
     by country;
     vbar3d year / discrete
                   sumvar=megtons
                   patternid=group
                   group=type
                   shape=hexagon
                   outside=sum
                   html=typedrill
                   width=12
                   gspace=3
                   space=0
                   cframe=grayaa
                   raxis=axis2
                   gaxis=axis4
                   maxis=axis4
                   name='country'
                   des='Grain and Year Breakdown';
 run;
```

Example 7, Part D

Features: VBAR3D options:

INSIDE=

NOZERO

ODS HTML options:

BODY=

NEWFILE=TABLE

Like Part C, this part uses BY-group processing to generate three graphs that show the three leading producers for each type of grain. The program subsets the data and suppresses midpoints with no observations. Instead of storing all of the output in one body file, it stores each graph in a separate file. To do this, the program uses the ODS HTML option NEWFILE=TABLE. When NEWFILE=TABLE is used with BY-group processing, each new piece of output automatically generates a new body file and simply increments the name of the file that is specified by BODY=. Because each graph is stored in a separate file, the links to these graphs reference only the file name and do not require an anchor name. The graphs created by this part are referenced by the TYPEDRILL variable.

Sort the data set for the graphs of leading producers of each grain type.

```
proc sort data=grainldr out=type;
by type;
run;
```

Open a new body file. Assigning a new body file closes COUNTRY_BODY.HTML. NEWFILE=TABLE opens a new body file for each piece of output generated by the procedure. Each new file increments the name specified by BODY= using the number within the body file name as the starting number.

```
ods html body='type1_body.html'
         newfile=table
         path=odsout;
```

Modify the group axis. Because SPLIT= assigns a blank as the split character, the value **United States** automatically prints on two lines.

```
axis5 label=none
      split=' ';
```

Define title and footnote. #BYVAL inserts the value of the BY variable TYPE into the title of each report.

```
title1 'Top Three Producers of #byval(type)';
title2 '(In Millions of Metric Tons)';
footnote j=r h=3 'TYPE ';
```

Generate the vertical bar chart of leading producers for each grain type. BY-group processing generates a separate graph for each value TYPE. Each new graph generates a new body file. NOZERO suppresses the midpoints that do not have any observations. INSIDE= displays the SUM statistic inside the bars.

```
proc gchart data=type (where=(megtons gt 31));
     format country $country.;
     by type;
     vbar3d year / discrete
```

```
                    sumvar=megtons
                    group=country
                    nozero
                    shape=cylinder
                    noframe
                    patternid=group
                    inside=sum
                    width=12
                    maxis=axis4
                    raxis=axis2
                    gaxis=axis5
                    cframe=grayaa
                    name='type'
                    des='Leading Producers';
run;
quit;
```

Close the ODS HTML destination, and open the ODS Listing destination. You must close the HTML destination before you can view the output with a browser.

```
ods html close;
ods listing;
```

Example 8: Specifying the Sum Statistic for a Pie Chart

Procedure features:

PIE statement options:

COUTLINE=
SUMVAR=

PIE3D statement options:

COUTLINE=
EXPLODE=
SUMVAR=

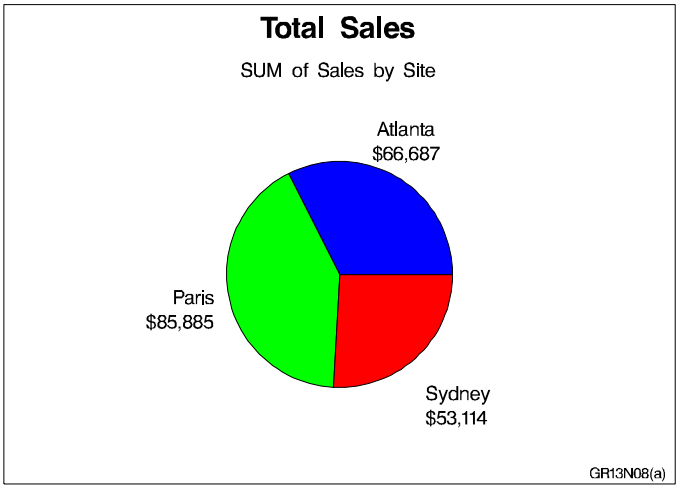
Other features:

FORMAT statement

RUN-group processing

Data set: TOTALS

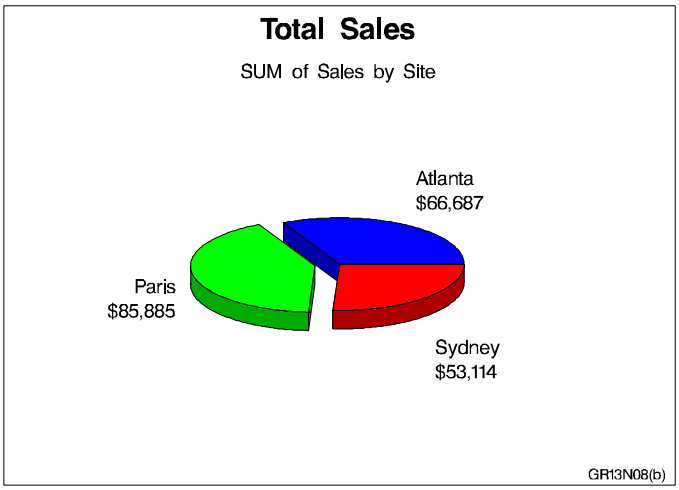
Sample library member: GR13N08



This example produces two pie charts that show total sales for three sites by charting the values of the character variable SITE and calculating the sum of the variable SALES for each site. It represents the statistics as slices of the pie. By default, the midpoint value and the summary statistic are printed beside each slice.

The pie slices use the default pattern fill, which is solid. Because a colors list is specified in the GOPTIONS statement, the default solid patterns rotate through the colors in the list, beginning with the first color, blue. Each slice displays a different color because, by default, pie charts are patterned by midpoint. Because the default outline color is also the first color in the list, the example uses the COUTLINE= option to assign black to the outlines.

The second pie chart is a 3D pie chart with an exploded slice, as shown in the following output.



Assign the libref and set the graphics environment. CTEXT= specifies the color for all text on the output. COLORS= specifies the colors list, which is used by the default patterns and outlines.

```
libname reflib 'SAS-data-library';
goptions reset=global gunit=pct border cback=white
```

```
          colors=(blue green red) ctext=black
          ftitle=swissb ftext=swiss htitle=6 htext=4;
```

Define title and footnote.

```
title 'Total Sales';
footnote j=r 'GR13N08(a) ';
```

Produce the first pie chart. The PIE statement produces a two-dimensional pie chart. SUMVAR= calculates the sum of SALES for each value of the chart variable SITE. The default statistic for SUMVAR= is SUM. The summary variable SALES is assigned a dollar format. COUTLINE= specifies the outline color for the slices.

```
proc gchart data=reflib.totals;
   format sales dollar8.;
   pie site / sumvar=sales
              coutline=black;
run;
```

Define footnote for second pie chart.

```
   footnote j=r 'GR13N08(b) ';
```

Produce the second pie chart. The PIE3D statement produces a three-dimensional pie chart. EXPLODE= separates the slice for Paris from the rest of the pie.

```
   pie3d site / sumvar=sales
                coutline=black
                explode='Paris';
run;
quit;
```

Example 9: Subgrouping a Donut or Pie Chart

Procedure features:

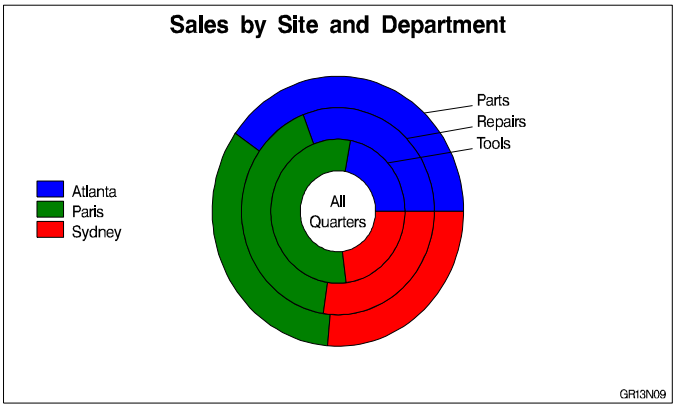
DONUT statement options:

- COUTLINE=
- CTEXT=
- DONUTPCT=
- LABEL=
- LEGEND=
- NOHEADING
- SUBGROUP=

Other features: LEGEND statement

Data set: TOTALS

Sample library member: GR13N09



This example produces a donut chart that is similar to the pie chart in Example 8 on page 610 in that each slice represents total sales for a site and each slice is a different color. However, in this donut chart the sites are subgrouped by department, so that each department is represented as a concentric ring with slices.

Subgrouping suppresses the chart statistic and the midpoint labels. Instead it automatically labels the rings with the subgroup values and generates a legend that shows how the patterns are associated with the midpoint values. Subgrouping a pie chart produces the same results but without the hole in the center.

To allow the donut chart to be as large as possible, the program suppresses the default heading and moves the legend into the space at the left of the chart.

Assign the libref and set the graphics environment.

```
libname reflib 'SAS-data-library';
goptions reset=global gunit=pct border cback=white
         colors=(blue green red) ctext=black
         ftext=swissb ftext=swiss htitle=6 htext=4;
```

Define title and footnote.

```
title 'Sales by Site and Department';
footnote h=3 j=r 'GR13N09 ';
```

Modify the subgroup legend. LABEL= suppresses the legend label. SHAPE= defines the shape of the legend values. POSITION=, OFFSET=, and ACROSS= arrange the legend entries in a column to the left of the pie chart. MODE= allows the legend to occupy the procedure output area.

```
legend1 label=none
        shape=bar(4,4)
        position=(middle left)
        offset=(5,)
        across=1
        mode=share;
```

Produce the donut chart. SUBGROUP= divides the donut into rings. Each ring represents a value of the subgroup variable, DEPT. The DONUTPCT= option controls the size of the donut hole, which contains the text specified by LABEL=. The NOHEADING option suppresses the default heading that contains the name of the chart variable and the type of statistic. LEGEND= assigns the LEGEND1 statement to the chart COUTLINE= specifies the outline color for the slices and subgroup rings. CTEXT= specifies the color used by the donut label and by the subgroup arrows.

```
proc gchart data=reflib.totals;
   format sales dollar8.;
   donut site / sumvar=sales
                subgroup=dept
                donutpct=30
                label=('All' justify=center 'Quarters')
                noheading
                legend=legend1
                coutline=black
                ctext=black;
run;
quit;
```

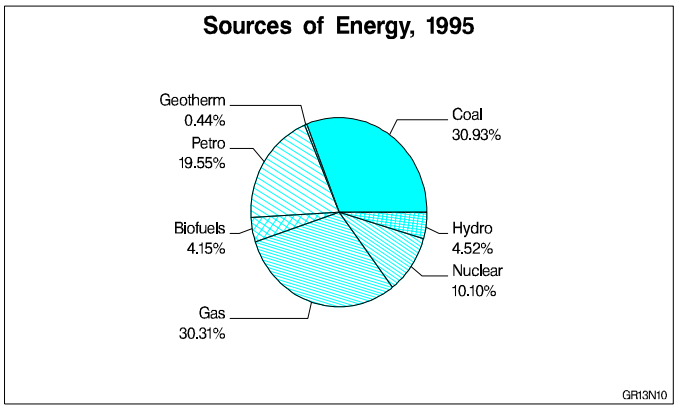
Example 10: Ordering and Labeling Slices in a Pie Chart

Procedure features:

PIE statement options:

CFILL=
MIDPOINTS=
PERCENT=ARROW
SLICE=ARROW
SUBGROUP=
VALUE=NONE

Sample library member: GR13N10



This example produces a pie chart of sources of energy production for 1995. The labeled slices represent the percent of total production for each source. Instead of the sum statistic, each slice displays the percent each midpoint contributes to the whole pie. Arrows connect the midpoint labels to the slices, which are arranged by the MIDPOINTS= option so that the small slices are not next to each other and their labels do not overlap.

Assign the libref and set the graphics environment.

```
libname reflib 'SAS-data-library';
goptions reset=global gunit=pct border cback=white
         colors=(black blue green red cyan lime gray)
         ftext=swiss ftitle=swissb htitle=6 htext=4;
```

Create the data set ENPROD.

ENPROD contains the amount of energy produced (PRODUCED) for seven sources (ENGYTYPE) for two years (YEAR). The amounts of energy produced are in quadrillion btu.

```
data reflib.enprod;
   length engytype $ 8;
   input year engytype produced;
   datalines;
1985 Coal      19.33
1985 Gas       19.22
1985 Petro     18.99
...more data lines...
1995 Hydro      3.21
1995 Geotherm    .31
1995 Biofuels   2.95
;
```

Define title and footnote.

```
title 'Sources of Energy, 1995';
footnote h=3 j=r 'GR13N10 ';
```

Produce the pie chart. The WHERE data set option subsets the data for 1995. OTHER=0 specifies that all midpoints, no matter how small, display a slice. MIDPOINTS= assigns the order of the slices. Each slice displays the percent contribution to total production and the slice name. VALUE=NONE suppresses the chart statistic. Both SLICE= and PERCENT= are assigned the ARROW labeling method to point to the slice, but only one arrow line is displayed. CFILL= specifies a color for the fill used by all slices.

```
proc gchart data=reflib.enprod (where=(year=1995));
     pie engytype / sumvar=produced
                    other=0
                    midpoints='Coal' 'Geotherm' 'Petro'
                              'Biofuels' 'Gas' 'Nuclear'
                              'Hydro'
                    value=none
                    percent=arrow
```

```
                        slice=arrow
                        cfill=cyan
                        noheading;
run;
quit;
```

Example 11: Assigning Patterns and Identifying Midpoints with a Legend

Procedure features:

PIE statement options:

COUTLINE=
CTEXT=
DESCENDING
LEGEND=
OTHER=
OTHERLABEL=
VALUE=INSIDE

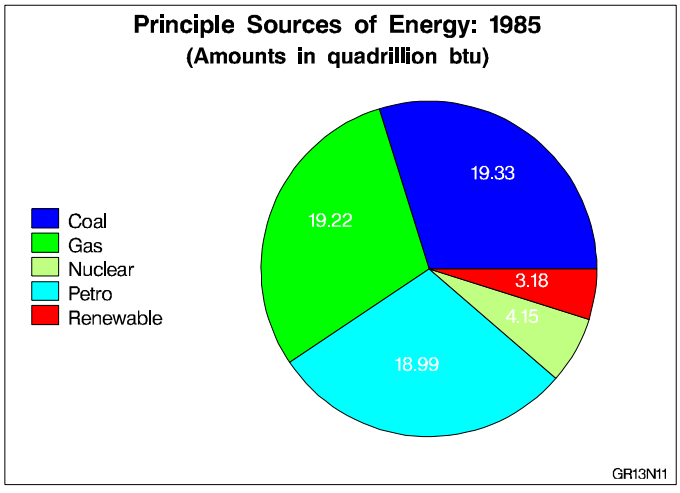
Other features:

LEGEND statement

PATTERN statement

Data set: ENPROD

Sample library member: GR13N11



This example shows the actual amount of energy that is produced by each source for 1985. It displays the chart statistic inside each slice and uses a legend instead of slice labels to identify the slices. Pattern colors are assigned explicitly to each energy source.

All of the variables with midpoint values less than or equal to 5 percent of the total (in this case, *biofuels*, *geotherm*, and *hydro*) are grouped into one slice labeled "Other." The slices are ordered from largest to smallest based on the statistic value. Although the "Other" slice is always last, it is in this case also the smallest.

Assign the libref and set the graphics environment.

```
libname reflib 'SAS-data-library';
goptions reset=global gunit=pct border cback=white
         colors=(black blue green red) ftitle=swissb
         ftext=swiss htitle=5 htext=4;
```

Define a title and footnote.

```
title1 'Principle Sources of Energy: 1985';
title2 font=swissb h=4.5 '(Amounts in quadrillion btu)';
footnote h=3 j=r 'GR13N11 ';
```

Define pattern colors. Each value of the chart variable ENGYTYPE is assigned a pattern whether or not it is displayed as a separate slice. Patterns are assigned to midpoints in the order the values appear in the data set. Because ENGYTYPE is character, the patterns are assigned alphabetically. The eighth pattern is for the "other" slice, which is always last.

```
pattern1 color=black;       /* biofuels     */
pattern2 color=blue;        /* coal         */
pattern3 color=green;       /* gas          */
pattern4 color=gray;        /* geothermal   */
pattern5 color=lipk;        /* hydroelectric */
pattern6 color=lime;        /* nuclear      */
pattern7 color=cyan;        /* petro        */
pattern8 color=red;         /* other        */
```

Modify the legend. LABEL= removes the legend label. VALUE= defines the color for the value labels; by default legend value color is determined by the CTEXT= option in the procedure statement. In this case, CTEXT=WHITE, so the legend statement uses the VALUE= option to override that color specification. ORDER= orders the legend values to match the slice order in the pie chart.

```
legend1 label=none
        position=(left middle)
        offset=(4,)
        across=1
        order=('Coal' 'Gas' 'Petro'
               'Nuclear' 'Other')
        value=(color=black)
        shape=bar(4,4);
```

Create the pie chart. OTHER= collects all the midpoints with statistic values less than or equal to 5 percent of the total into one slice. OTHERLABEL= specifies the label for the "other" slice. DESCENDING arranges the slices in descending order of the statistic value. LEGEND= displays the customized legend created in the LEGEND statement and suppresses the slice labels. VALUE= places the chart statistics inside the slices. CTEXT= specifies white for the statistic text. Because CTEXT= also affects the color of the legend text, the LEGEND statement specifies black text so that the values can be seen. Because the PATTERN statement is used, the default slice outline matches the fill color; COUTLINE= changes the outline color to black.

```
proc gchart data=reflib.enprod(where=(year=1985));
   pie engytype / sumvar=produced
                  other=5
                  otherlabel='Renewable'
                  descending
                  legend=legend1
                  value=inside
                  ctext=white
                  coutline=black
                  noheading;
run;
quit;
```

Example 12: Grouping and Arranging Pie Charts

Procedure features:

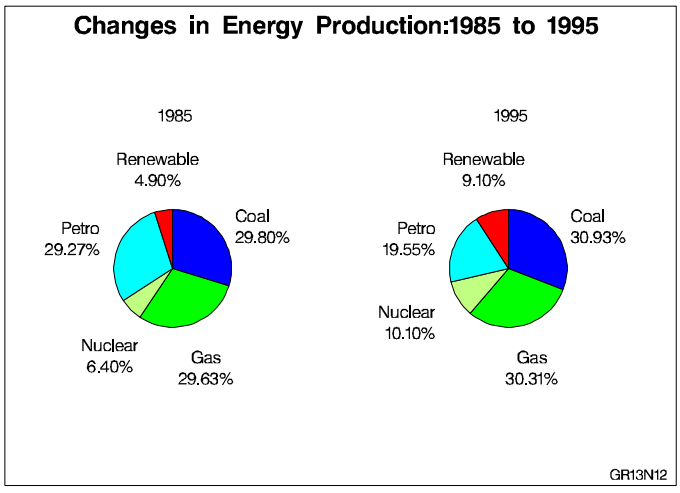
PIE statement options:

ACROSS=
CLOCKWISE
GROUP=
OTHER=
PERCENT=OUTSIDE
SLICE=OUTSIDE

Other features: PATTERN statement

Data set: ENPROD

Sample library member: GR13N12



This example produces two pie charts that shows energy sources by year. Both charts are displayed on one page and are arranged two across. The program uses the CLOCKWISE option to arrange the slices, which begin at the 12 o'clock position and proceed clockwise in alphabetic order of the midpoint.

The chart statistic is suppressed and the midpoint label and the percent of the chart statistic are displayed outside of the slice.

A different color is defined for each energy type. The patterns are assigned in order of midpoint value. Some colors do not appear because the slices they represent are grouped into the OTHER slice, which is assigned the eighth color, red.

Set the graphics environment.

```
goptions reset=global gunit=pct border cback=white
         colors=(black blue green red) ftext=swiss
         ftext=swissb htitle=5 htext=3.5;
```

Define title and footnote.

```
title 'Changes in Energy Production:1985 to 1995';
footnote j=r 'GR13N12 ';
```

Define patterns for the pie slices. PATTERN statements define a different solid color for each midpoint value.

```
pattern1 color=black;       /* biofuels     */
pattern2 color=blue;        /* coal         */
pattern3 color=green;       /* gas          */
pattern4 color=gray;        /* geothermal   */
pattern5 color=lipk;        /* hydroelectric */
pattern6 color=lime;        /* nuclear      */
pattern7 color=cyan;        /* petro        */
pattern8 color=red;         /* other        */
```

Produce the pie charts. The WHERE= data set option selects the data for only two years. The LABEL statement suppresses the variable name, so only the YEAR value is displayed.

```
proc gchart data=reflib.enprod gout=reflib.excat;
     label year='00'x;
```

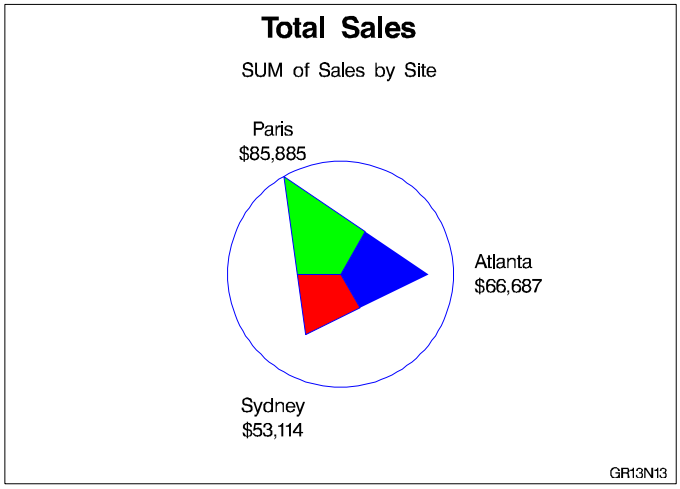
GROUP= creates a separate pie for each year. In combination with GROUP=, ACROSS= draws two charts across one page. OTHER= collects all the midpoints with statistic values less than or equal to 5 percent of the total into one slice. CLOCKWISE begins drawing the slices at the 12 o'clock position in alphabetic order of the midpoint. PERCENT=OUTSIDE and SLICE=OUTSIDE display the labels outside the slices.

```
     pie engytype / sumvar=produced
                    group=year
                    across=2
                    other=5
                    otherlabel='Renewable'
                    clockwise
                    value=none
                    slice=outside
                    percent=outside
```

```
                        coutline=black
                        noheading;
run;
quit;
```

Example 13: Specifying the Sum Statistic in a Star Chart

Procedure features:
STAR statement options:
SUMVAR=
Other features: FORMAT statement
Data set: TOTALS
Sample library member: GR13N013



This example produces a star chart of total sales for three sites by charting the values of the character variable SITE and calculating the sum of the variable SALES for each site. It represents the statistics as slices of the star. The center of the circle represents 0 and the edge of the circle represents the largest value, in this case Paris sales. By default, the spines are joined and filled with a solid pattern to form slices, and the midpoint value and the formatted values of the sales statistics are printed beside each slice.

By default, the circle and the slice outlines use the first color in the colors list, in this case, BLUE.

Assign the libref and set the graphics environment.

```
libname reflib 'SAS-data-library';
goptions reset=global gunit=pct border cback=white
         colors=(blue green red) ctext=black
         ftitle=swissb ftext=swiss htitle=6 htext=4;
```

Define title and footnote.

```
title 'Total Sales';
footnote h=3 j=r 'GR13N13 ';
```

Produce the star chart. SUMVAR= calculates the sum of SALES for each value of the chart variable SITE. Because the TYPE= option is omitted, the default statistic is sum. The FORMAT statement assigns a format to the summary variable SALES.

```
proc gchart data=reflib.totals;
   format sales dollar8.;
   star site  / sumvar=sales;
run;
quit;
```

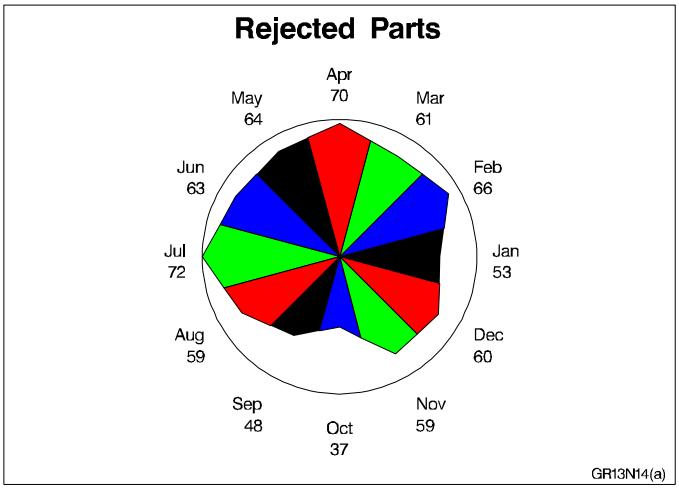
Example 14: Charting a Discrete Numeric Variable in a Star Chart

Procedure features:

STAR statement options:

COUTLINE=
DISCRETE
FILL=
NOCONNECT
NOHEADING
SUMVAR=

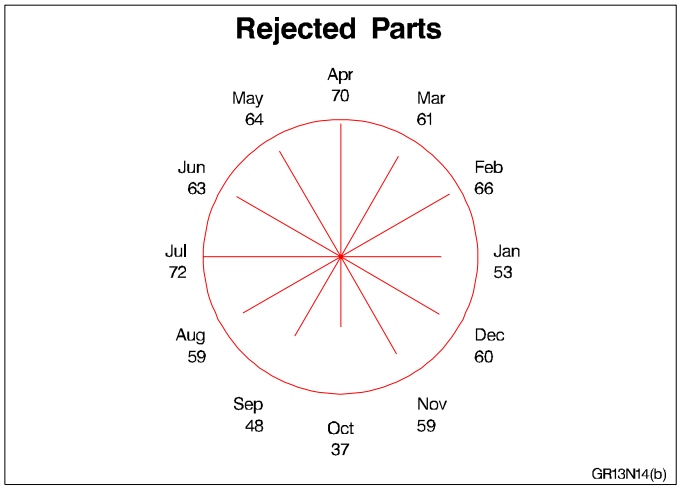
Sample library member: GR13N14



This example produces two star charts that show the total number of parts that were rejected each month for a year. The STAR statement uses the DISCRETE option so that each unique value of the numeric variable DATE is a separate midpoint and has a separate spine. Each slice displays the formatted midpoint value and the chart

statistic. Specifying FILL=S rotates the solid pattern through all the colors in the colors list as many times as necessary to provide patterns for all the slices.

The second chart uses the NOCONNECT option so that the chart uses spines instead of slices.



Set the graphics environment. COLORS= specifies the colors list, which is used by the default patterns and outlines.

```
goptions reset=global gunit=pct border cback=white
         colors=(black blue green red) ftext=swiss
         ftitle=swissb htext=3.5 htitle=6;
```

Create the data set REJECTS. REJECTS contains data on the number of defective parts produced at each of three sites for 12 months. BADPARTS is the number of parts that were rejected at each site for each month.

```
data rejects;
   informat date date9.;
   input site $ date badparts;
   datalines;
Sydney  01JAN1997 22
Sydney  01FEB1997 26
...more data lines...
Paris   01NOV1997 12
Paris   01DEC1997 19
;
```

Define title and footnote.

```
title 'Rejected Parts';
footnote j=r 'GR13N14(a)  ';
```

Produce the first star chart. DISCRETE must be specified because the numeric chart variable, DATE is assigned the WORDDATE3. Using FILL=S fills all the slices with solid patterns.

```
proc gchart data=rejects;
   format date worddate3.;
   star date / discrete
               sumvar=badparts
               noheading
               fill=s;
run;
```

Define footnote for the second chart.

```
footnote j=r 'GR13N14(b) ';
```

Produce the second star chart with slices and a solid fill. NOHEADING suppresses the default heading for the star chart. NOCONNECT suppresses the lines that by default join the ends of the spines. COUTLINE= colors the spines and the circle.

```
   star date / discrete
               sumvar=badparts
               noheading
               noconnect
               coutline=red;
run;
quit;
```

References

Nelder, J. A. (1976), "A Simple Algorithm for Scaling Graphs," *Applied Statistics, Volume 25, Number 1*, London: The Royal Statistical Society.

Terrell, G. R. and Scott, D. W. (1985), "Oversmoothed Nonparametric Density Estimates," *Journal of the American Statistical Association*, 80.

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS/GRAPH® Software: Reference, Version 8*, Cary, NC: SAS Institute Inc., 1999.

SAS/GRAPH® Software: Reference, Version 8

ISBN 1-58025-525-6

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st printing, October 1999