The CHART Procedure

Overview

The CHART procedure produces vertical and horizontal bar charts, block charts, pie charts, and star charts. These types of charts graphically display values of a variable or a statistic associated with those values. The charted variable can be numeric or character.

PROC CHART is a useful tool to visualize data quickly, but if you need to produce presentation-quality graphics that include color and various fonts, you can use SAS/GRAFHP software. The GCHART procedure in SAS/GRAFHP software produces the same types of charts as PROC CHART does. In addition, PROC GCHART can produce donut charts.
The following sections explain the different types of charts that PROC CHART can produce. All of the charts illustrate the results from a multiple-choice survey of 568 people, with five possible responses that range from “always” to “never.”

**About Bar Charts**

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

Output 7.1 on page 174 shows a vertical bar chart that displays the number of responses for the five categories from the survey data. The following statements produce the output:

```sas
options nodate pageno=1 linesize=80
   pagesize=30;

proc chart data=survey;
   vbar response / sumvar=count
       midpoints='Always' 'Usually' 'Sometimes' 'Rarely' 'Never';
run;
```

Output 7.1 Vertical Bar Chart

Output 7.2 on page 175 shows the same data presented in a horizontal 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. The following statements produce the output:

```sas
proc chart data=survey;
   hbar response / sumvar=count
       midpoints='Always' 'Usually' 'Sometimes' 'Rarely' 'Never';
run;
```
options nodate pageno=1 linesize=80
    pagesize=60;

proc chart data=survey;
    hbar response / sumvar=count
        midpoints='Always' 'Usually'
            'Sometimes' 'Rarely' 'Never';
run;

Output 7.2 Horizontal Bar Chart

<table>
<thead>
<tr>
<th>Response</th>
<th>Count Sum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Always</td>
<td>************ 106.0000</td>
</tr>
<tr>
<td>Usually</td>
<td>************************************************** 202.0000</td>
</tr>
<tr>
<td>Sometimes</td>
<td>************************ 119.0000</td>
</tr>
<tr>
<td>Rarely</td>
<td>******************* 97.0000</td>
</tr>
<tr>
<td>Never</td>
<td>********* 44.0000</td>
</tr>
</tbody>
</table>

About Block Charts

Block charts display the relative magnitude of data by using blocks of varying height, each set in a square that represents a category of data. Output 7.3 on page 176 shows the number of each survey response in the form of a block chart.

options nodate pageno=1 linesize=80
    pagesize=30;

proc chart data=survey;
    block response / sumvar=count
        midpoints='Always' 'Usually'
            'Sometimes' 'Rarely' 'Never';
run;
Output 7.3  Block Chart

### About Pie Charts

Pie charts represent the relative contribution of parts to the whole by displaying data as wedge-shaped slices of a circle. Each slice represents a category of the data. Output 7.4 on page 177 shows the survey results divided by response into five pie slices. The following statements produce the output:

```sas
options nodate pageno=1 linesize=80 pagesize=35;
proc chart data=survey;
pie response / sumvar=count;
run;
```

---

![Pie Chart Diagram](image)
About Star Charts

With PROC CHART, you can produce star charts that show group frequencies, totals, or mean values. A star chart is similar to a vertical bar chart, but the bars on a star chart radiate from a center point, like spokes in a wheel. Star charts are commonly used for cyclical data, such as measures taken every month or day or hour, or for data like these in which the categories have an inherent order ("always" meaning more frequent than "usually" which means more frequent than "sometimes"). Output 7.5 on page 178 shows the survey data displayed in a star chart. The following statements produce the output:

```sas
options nodate pageno=1 linesize=80
       pagesize=60;

proc chart data=survey;
   star response / sumvar=count;
run;
```
Output 7.5  Star Chart

The SAS System 1
Center = 0  Sum of Count by Response  Outside = 202

Never
************* 44
***** *****
*** ***
** **
**

Rarely
97 *
** *
** *
** *

Sometimes
119 *
** *
** *
** *

Always
106 *
** *
***** *

Usually
202

Procedure Syntax

Requirement: You must use at least one of the chart-producing statements.
Tip: Supports the Output Delivery System (see Chapter 2, “Fundamental Concepts for Using Base SAS Procedures”)
Reminder: You can use the ATTRIB, FORMAT, LABEL, and WHERE statements. See Chapter 3, “Statements with the Same Function in Multiple Procedures,” for details. You can also use any global statements as well. See Chapter 2, “Fundamental Concepts for Using Base SAS Procedures,” for a list.

PROC CHART <option(s)>;
   BLOCK variable(s) </ option(s)>;
   BY <DESCENDING> variable-1
PROC CHART Statement

PROC CHART <option(s)>;

Options

DATA=SAS-data-set
identifies the input SAS data set.

Main discussion: "Input Data Sets" on page 18

Restriction: You cannot use PROC CHART with an engine that supports concurrent access if another user is updating the data set at the same time.

FORMCHAR <position(s))>="formatting-character(s)"

defines the characters to use for constructing the horizontal and vertical axes, reference lines, and other structural parts of a chart. It also defines the symbols to use to create the bars, blocks, or sections in the output.

position(s)
identifies the position of one or more characters in the SAS formatting-character string. A space or a comma separates the positions.

Default: Omitting (position(s)), is the same as specifying all 20 possible SAS formatting characters, in order.

Range: PROC CHART uses 6 of the 20 formatting characters that SAS provides. Table 7.1 on page 180 shows the formatting characters that PROC CHART uses. Figure 7.1 on page 180 illustrates the use of formatting characters commonly used in PROC CHART.

formatting-character(s)
lists the characters to use for the specified positions. PROC CHART assigns characters in formatting-character(s) to position(s), in the order that they are listed. For instance, the following option assigns the asterisk (*) to the second formatting character, the pound sign (#) to the seventh character, and does not alter the remaining characters:

formchar(2,7)="*#"

Interaction: The SAS system option FORMCHAR= specifies the default formatting characters. The system option defines the entire string of formatting characters. The FORMCHAR= option in a procedure can redefine selected characters.

Tip: You can use any character in formatting-characters, including hexadecimal characters. If you use hexadecimal characters, you must put an x after the closing
quote. For instance the following option assigns the hexadecimal character 2D to the second formatting character, the hexadecimal character 7C to the seventh character, and does not alter the remaining characters:

```
formchar(2,7)='2D7C'x
```

**See also:** For information on which hexadecimal codes to use for which characters, consult the documentation for your hardware.

<table>
<thead>
<tr>
<th>Position</th>
<th>Default</th>
<th>Used to draw</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>Vertical axes in bar charts, the sides of the blocks in block charts, and reference lines in horizontal bar charts. In side-by-side bar charts, the first and second formatting characters appear around each value of the group variable (below the chart) to indicate the width of each group.</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>Horizontal axes in bar charts, the horizontal lines that separate the blocks in a block chart, and reference lines in vertical bar charts. In side-by-side bar charts, the first and second formatting characters appear around each value of the group variable (below the chart) to indicate the width of each group.</td>
</tr>
<tr>
<td>7</td>
<td>+</td>
<td>Tick marks in bar charts and the centers in pie and star charts.</td>
</tr>
<tr>
<td>9</td>
<td>-</td>
<td>Intersection of axes in bar charts.</td>
</tr>
<tr>
<td>16</td>
<td>/</td>
<td>Ends of blocks and the diagonal lines that separate blocks in a block chart.</td>
</tr>
<tr>
<td>20</td>
<td>*</td>
<td>Circles in pie and star charts.</td>
</tr>
</tbody>
</table>

**LPI=value**

specifies the proportions of PIE and STAR charts. The value is determined by

Table 7.1 Formatting Characters Used by PROC CHART

<table>
<thead>
<tr>
<th>Position</th>
<th>Default</th>
<th>Used to draw</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>Vertical axes in bar charts, the sides of the blocks in block charts, and reference lines in horizontal bar charts. In side-by-side bar charts, the first and second formatting characters appear around each value of the group variable (below the chart) to indicate the width of each group.</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>Horizontal axes in bar charts, the horizontal lines that separate the blocks in a block chart, and reference lines in vertical bar charts. In side-by-side bar charts, the first and second formatting characters appear around each value of the group variable (below the chart) to indicate the width of each group.</td>
</tr>
<tr>
<td>7</td>
<td>+</td>
<td>Tick marks in bar charts and the centers in pie and star charts.</td>
</tr>
<tr>
<td>9</td>
<td>-</td>
<td>Intersection of axes in bar charts.</td>
</tr>
<tr>
<td>16</td>
<td>/</td>
<td>Ends of blocks and the diagonal lines that separate blocks in a block chart.</td>
</tr>
<tr>
<td>20</td>
<td>*</td>
<td>Circles in pie and star charts.</td>
</tr>
</tbody>
</table>

Figure 7.1 Formatting Characters Commonly Used in PROC CHART Output

Mean Yearly Pie Sales Grouped by Flavor within Bakery Location

<table>
<thead>
<tr>
<th>Flavor</th>
<th>Bakery</th>
</tr>
</thead>
<tbody>
<tr>
<td>a b c r</td>
<td>Clyde</td>
</tr>
<tr>
<td>p l h h</td>
<td>--------</td>
</tr>
<tr>
<td>p u e u</td>
<td>------ Oak------</td>
</tr>
<tr>
<td>l e r b</td>
<td>------ Samford------</td>
</tr>
<tr>
<td>e b r a</td>
<td>Bakery</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Flavor</th>
<th>Bakery</th>
</tr>
</thead>
<tbody>
<tr>
<td>a b c r</td>
<td>e y r</td>
</tr>
<tr>
<td>p l h h</td>
<td>e y r</td>
</tr>
<tr>
<td>p u e u</td>
<td>e y r</td>
</tr>
<tr>
<td>l e r b</td>
<td>r b r</td>
</tr>
<tr>
<td>e b r a</td>
<td>r r</td>
</tr>
</tbody>
</table>

LPI=value

specifies the proportions of PIE and STAR charts. The value is determined by
For example, if you have a printer with 8 lines per inch and 12 columns per inch, specify LPI=6.6667.

**Default:** 6

---

**BLOCK Statement**

Produce a block chart.

**Featured in:** Example 6 on page 203

**BLOCK variable(s) /*option(s)>**

**Required Arguments**

**variable(s)**

specifies the variables for which PROC CHART produces a block chart, one chart for each variable.

**Options**

The options available on the BLOCK, HBAR, PIE, STAR, and VBAR statements are documented in “Customizing All Types of Charts” on page 185.

**Statement Results**

Because each block chart must fit on one output page, you may have to adjust the SAS system options LINESIZE= and PAGESIZE= if you have a large number of charted values for the BLOCK variable and for the variable specified in the GROUP= option.

<table>
<thead>
<tr>
<th>GROUP= Value</th>
<th>LS= 132</th>
<th>LS= 120</th>
<th>LS= 105</th>
<th>LS= 90</th>
<th>LS= 76</th>
<th>LS= 64</th>
</tr>
</thead>
<tbody>
<tr>
<td>0, 1</td>
<td>9</td>
<td>8</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>8</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>5, 6</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>

If the value of any GROUP= level is longer than three characters, the maximum number of charted values for the BLOCK variable that can fit may be reduced by one.
BLOCK level values truncate to 12 characters. If you exceed these limits, PROC CHART produces a horizontal bar chart instead.

**BY Statement**

*Produces a separate chart for each BY group.*

**Required Arguments**

**variable**

specifies the variable that the procedure uses to form BY groups. You can specify more than one variable. If you do not use the NOTSORTED option in the BY statement, the observations in the data set must either be sorted by all the variables that you specify, or they must be indexed appropriately. Variables in a BY statement are called BY variables.

**Options**

**DESCENDING**

specifies that the observations are sorted in descending order by the variable that immediately follows the word DESCENDING in the BY statement.

**NOTSORTED**

specifies that observations are not necessarily sorted in alphabetic or numeric order. The observations are grouped in another way, for example, chronological order.

The requirement for ordering or indexing observations according to the values of BY variables is suspended for BY-group processing when you use the NOTSORTED option. In fact, the procedure does not use an index if you specify NOTSORTED. The procedure defines a BY group as a set of contiguous observations that have the same values for all BY variables. If observations with the same values for the BY variables are not contiguous, the procedure treats each contiguous set as a separate BY group.

**HBAR Statement**

*Produces a horizontal bar chart.*

**Tip:** HBAR charts can print either the name or the label of the chart variable.

**Featured in:** Example 5 on page 202
The CHART Procedure

PIE Statement

PIE variable(s) <</option(s)>;

Required Argument

variable(s)

specifies the variables for which PROC CHART produces a pie chart, one chart for each variable.

Options

The options available on the BLOCK, HBAR, PIE, STAR, and VBAR statements are documented in “Customizing All Types of Charts” on page 185.

Statement Results

PROC CHART determines the number of slices for the pie in the same way that it determines the number of bars for vertical bar charts. Any slices of the pie accounting for less than three print positions are grouped together into an “OTHER” category.
The pie's size is determined only by the SAS system options LINESIZE= and PAGESIZE=. By default, the pie looks elliptical if your printer does not print 6 lines per inch and 10 columns per inch. To make a circular pie chart on a printer that does not print 6 lines and 10 columns per inch, use the LPI= option on the PROC CHART statement. See the description of LPI= on page 180 for the formula that gives you the proper LPI= value for your printer.

If you try to create a PIE chart for a variable with more than 50 levels, PROC CHART produces a horizontal bar chart instead.

---

**STAR Statement**

Produces a star chart.

```
STAR variable(s) <option(s)>;
```

**Required Argument**

`variable(s)`

specifies the variables for which PROC CHART produces a star chart, one chart for each variable.

**Options**

The options available on the BLOCK, HBAR, PIE, STAR, and VBAR statements are documented in “Customizing All Types of Charts” on page 185.

**Statement Results**

The number of points in the star is determined in the same way as the number of bars for vertical bar charts.

If all the data have positive values, the center of the star represents zero and the outside circle represents the maximum value. If the data contain negative values, the center represents the minimum. See the description of the AXIS= option on page 186 for more information about how to specify maximum and minimum values. For information about how to specify the proportion of the chart, see the description of the LPI= option on page 180.

If you try to create a star chart for a variable with more than 24 levels, PROC CHART produces a horizontal bar chart instead.

---

**VBAR Statement**

Produces a vertical bar chart.

**Featured in:** Example 1 on page 192, Example 2 on page 194, Example 3 on page 196, Example 4 on page 200
**VBAR variable(s) <option(s)>;**

**Required Argument**

`variable(s)`

specifies the variables for which PROC CHART produces a vertical bar chart, one chart for each variable.

**Options**

The options available on the BLOCK, HBAR, PIE, STAR, and VBAR statements are documented in “Customizing All Types of Charts” on page 185.

**Statement Results**

PROC CHART prints one page per chart. Along the vertical axis, PROC CHART describes the chart frequency, the cumulative frequency, the chart percentage, the cumulative percentage, the sum, or the mean. At the bottom of each bar, PROC CHART prints a value according to the value of the `TYPE=option`, if specified. For character variables or discrete numeric variables, this value is the actual value represented by the bar. For continuous numeric variables, the value gives the midpoint of the interval represented by the bar.

PROC CHART can automatically scale the vertical axis, determine the bar width, and choose spacing between the bars. However, by using options, you can choose bar intervals and the number of bars, include missing values in the chart, produce side-by-side charts, and subdivide the bars. If the number of characters per line (`LINESIZE=`) is not sufficient to display all vertical bars, PROC CHART produces a horizontal bar chart instead.

---

**Customizing All Types of Charts**

Many options in PROC CHART are valid in more than one statement. This section describes the options that you can use on the chart-producing statements.

<table>
<thead>
<tr>
<th>To do this</th>
<th>Use this option</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify that numeric variables are discrete</td>
<td><code>DISCRETE</code></td>
</tr>
<tr>
<td>Specify a frequency variable</td>
<td><code>FREQ=</code></td>
</tr>
<tr>
<td>Specify that missing values are valid levels</td>
<td><code>MISSING</code></td>
</tr>
<tr>
<td>Specify the variable for which values or means are displayed</td>
<td><code>SUMVAR=</code></td>
</tr>
<tr>
<td>Specify the statistic represented in the chart</td>
<td><code>TYPE=</code></td>
</tr>
<tr>
<td>Specify groupings</td>
<td></td>
</tr>
<tr>
<td>Group the bars in side-by-side charts</td>
<td><code>GROUP=</code></td>
</tr>
<tr>
<td>Specify that group percentages sum to 100</td>
<td><code>G100</code></td>
</tr>
<tr>
<td>Group the bars in side-by-side charts</td>
<td><code>GROUP=</code></td>
</tr>
<tr>
<td>Specify the number of bars for continuous variables</td>
<td><code>LEVELS=</code></td>
</tr>
</tbody>
</table>
To do this | Use this option
--- | ---
Define ranges for continuous variables | MIDPOINTS=
Divide the bars into categories | SUBGROUP=

**Compute statistics**

- Compute the cumulative frequency for each bar | CFREQ
- Compute the cumulative percentage for each bar | CPERCENT
- Compute the frequency for each bar | FREQ
- Compute the mean of the observations for each bar | MEAN
- Compute the percentage of total observations for each bar | PERCENT
- Compute the total number of observations for each bar | SUM

**Control output format**

- Print the bars in ascending order of size | ASCENDING
- Specify the values for the response axis | AXIS=
- Print the bars in descending order of size | DESCENDING
- Specify extra space between groups of bars | GSPACE=
- Suppress the default header line | NOHEADER
- Allow no space between vertical bars | NOSPACE
- Suppress the statistics | NOSTATS
- Suppress the subgroup legend or symbol table | NOSYMBOL
- Suppress the bars with zero frequency | NOZEROS
- Draw reference lines | REF=
- Specify the spaces between bars | SPACE=
- Specify the symbols within bars or blocks | SYMBOL=
- Specify the width of bars | WIDTH=

**Options**

**ASCENDING**

prints the bars and any associated statistics in ascending order of size within groups.

**Alias:** ASC

**Restriction:** Available only on the HBAR and VBAR statements

**AXIS=value-expression**

specifies the values for the response axis, where value-expression is a list of individual values, each separated by a space, or a range with a uniform interval for the values. For example, the following range specifies tick marks on a bar chart from 0 to 100 at intervals of 10:

```
  hbar x / axis=0 to 100 by 10;
```

**Restriction:** Not available on the PIE statement

**Restriction:** Values must be uniformly spaced, even if you specify them individually.

**Restriction:** For frequency charts, values must be integers.
Interaction: For BLOCK charts, AXIS= sets the scale of the tallest block. To set the scale, PROC CHART uses the maximum value from the AXIS=list. If no value is greater than 0, PROC CHART ignores the AXIS= option.

Interaction: For HBAR and VBAR charts, AXIS= determines tick marks on the response axis. If the AXIS= specification contains only one value, the value determines the minimum tick mark if the value is less than 0, or determines the maximum tick mark if the value is greater than 0.

Interaction: For STAR charts, a single AXIS= value sets the minimum (the center of the chart) if the value is less than zero, or sets the maximum (the outside circle) if the value is greater than zero. If the AXIS= specification contains more than one value, PROC CHART uses the minimum and maximum values from the list.

Interaction: If you use AXIS= and the BY statement, PROC CHART produces uniform axes over BY groups.

CAUTION: Values in value-expression override the range of the data. For example, if the data range is 1 to 10 and you specify a range of 3 to 5, only the data in the range 3 to 5 appear on the chart. Values out of range produce a warning message in the SAS log.

CFREQ
prints the cumulative frequency.

Restriction: Available only on the HBAR statement

CPERCENT
prints the cumulative percentages.

Restriction: Available only on the HBAR statement

DESCENDING
prints the bars and any associated statistics in descending order of size within groups.

Alias: DESC

Restriction: Available only on the HBAR and VBAR statements

DISCRETE
specifies that a numeric chart variable is discrete rather than continuous. Without DISCRETE, PROC CHART assumes that all numeric variables are continuous and automatically chooses intervals for them unless you use MIDPOINTS= or LEVELS=.

FREQ
prints the frequency of each bar to the side of the chart.

Restriction: Available only on the HBAR statement

FREQ=variable
specifies a data set variable that represents a frequency count for each observation. Normally, each observation contributes a value of one to the frequency counts. With FREQ=, each observation contributes its value of the FREQ= value.

Restriction: If the FREQ= values are not integers, PROC CHART truncates them.

Interaction: If you use SUMVAR=, PROC CHART multiplies the sums by the FREQ= value.

GROUP=variable
produces side-by-side charts, with each chart representing the observations that have a common value for the GROUP= variable. The GROUP= variable can be character
or numeric and is assumed to be discrete. For example, the following statement produces a frequency bar chart for men and women in each department:

```plaintext
vbar gender / group=dept;
```

Missing values for a GROUP= variable are treated as valid levels.

**Restriction:** Available only on the BLOCK, HBAR, and VBAR statements

**Featured in:** Example 4 on page 200, Example 5 on page 202, Example 6 on page 203

**GSPACE=n**

specifies the amount of extra space between groups of bars. Use GSPACE=0 to leave no extra space between adjacent groups of bars.

**Restriction:** Available only on the HBAR and VBAR statements

**Interaction:** PROC CHART ignores GSPACE= if you omit GROUP=.

**G100**

specifies that the sum of percentages for each group equals 100. By default, PROC CHART uses 100 percent as the total sum. For example, if you produce a bar chart that separates males and females into three age categories, the six bars, by default, add to 100 percent; however, with G100, the three bars for females add to 100 percent, and the three bars for males add to 100 percent.

**Restriction:** Available only on the BLOCK, HBAR, and VBAR statements

**Interaction:** PROC CHART ignores G100 if you omit GROUP=.

**LEVELS=number-of-midpoints**

specifies the number of bars that represent each chart variable when the variables are continuous.

**MEAN**

prints the mean of the observations represented by each bar.

**Restriction:** Available only on the HBAR statement and only when you use SUMVAR= and TYPE=

**Restriction:** Not available when TYPE=CFREQ, CPERCENT, FREQ, or PERCENT

**MIDPOINTS=midpoint-specification | OLD**

defines the range of values that each bar, block, or section represents by specifying the range midpoints.

The value for MIDPOINTS= is one of the following:

**midpoint-specification**

specifies midpoints, either individually, or across a range at a uniform interval. For example, the following statement produces a chart with five bars; the first bar represents the range of values of X with a midpoint of 10, the second bar represents the range with a midpoint of 20, and so on:

```plaintext
vbar x / midpoints=10 20 30 40 50;
```

Here is an example of a midpoint specification for a character variable:

```plaintext
vbar x / midpoints='JAN' 'FEB' 'MAR';
```

Here is an example of specifying midpoints across a range at a uniform interval:

```plaintext
vbar x / midpoints=10 to 100 by 5;
```
OLD specifies an algorithm that PROC CHART used in previous versions of SAS to choose midpoints for continuous variables. The old algorithm was based on the work of Nelder (1976). The current algorithm that PROC CHART uses if you omit OLD is based on the work of Terrell and Scott (1985).

**Default:** Without MIDPOINTS=, PROC CHART displays the values in the SAS System’s normal sorted order.

**Restriction:** When the VBAR variables are numeric, the midpoints must be given in ascending order.

MISSING specifies that missing values are valid levels for the chart variable.

NOHEADER suppresses the default header line printed at the top of a chart.

**Alias:** NOHEADING

**Restriction:** Available only on the BLOCK, PIE, and STAR statements

**Featured in:** Example 6 on page 203

NOSTATS suppresses the statistics on a horizontal bar chart.

**Alias:** NOSTAT

**Restriction:** Available only on the HBAR statement

NOSYMBOL suppresses printing of the subgroup symbol or legend table.

**Alias:** NOLEGEND

**Restriction:** Available only on the BLOCK, HBAR, and VBAR statements

**Interaction:** PROC CHART ignores NOSYMBOL if you omit SUBGROUP=.

NOZEROS suppresses any bar with zero frequency.

**Restriction:** Available only on the HBAR and VBAR statements

PERCENT prints the percentages of observations having a given value for the chart variable.

**Restriction:** Available only on the HBAR statement

**REF=**value(s) draws reference lines on the response axis at the specified positions.

**Restriction:** Available only on the HBAR and VBAR statements

**Tip:** The REF= values should correspond to values of the TYPE= statistic.

**Featured in:** Example 4 on page 200

**SPACE=**n specifies the amount of space between individual bars.

**Restriction:** Available only on the HBAR and VBAR statements

**Tip:** Use SPACE=0 to leave no space between adjacent bars.

**Tip:** Use the GSPACE= option to specify the amount of space between the bars within each group.
SUBGROUP=variable

subdivides each bar or block into characters that show the contribution of the values of variable to that bar or block. PROC CHART uses the first character of each value to fill in the portion of the bar or block that corresponds to that value, unless more than one value begins with the same first character. In that case, PROC CHART uses the letters A, B, C, and so on to fill in the bars or blocks. If the variable is formatted, PROC CHART uses the first character of the formatted value.

The characters used in the chart and the values that they represent are given in a legend at the bottom of the chart. The subgroup symbols are ordered A through Z and 0 through 9 with the characters in ascending order.

PROC CHART calculates the height of a bar or block for each subgroup individually and then rounds the percentage of the total bar up or down. So the total height of the bar may be higher or lower than the same bar without the SUBGROUP= option.

**Restriction:** Available only on the BLOCK, HBAR, and VBAR statements

**Interaction:** If you use both TYPE=MEAN and SUBGROUP=, PROC CHART first calculates the mean for each variable listed in the SUMVAR= option, then subdivides the bar into the percentages contributed by each subgroup.

**Featured in:** Example 3 on page 196

SUM

prints the total number of observations that each bar represents.

**Restriction:** Available only on the HBAR statement and only when you use both SUMVAR= and TYPE=

**Restriction:** Not available when TYPE=CFREQ, CPERCENT, FREQ, or PERCENT

SUMVAR=variable

specifies the variable for which either values or means (depending on the value of TYPE=) PROC CHART displays in the chart.

**Interaction:** If you use SUMVAR= and you use TYPE= with a value other than MEAN or SUM, TYPE=SUM overrides the specified TYPE= value.

**Tip:** Both HBAR and VBAR charts can print labels for SUMVAR= variables if you use a LABEL statement.

**Featured in:** Example 3 on page 196, Example 4 on page 200, Example 5 on page 202, Example 6 on page 203

SYMBOL=character(s)

specifies the character or characters that PROC CHART uses in the bars or blocks of the chart when you do not use the SUBGROUP= option.

**Default:** asterisk (*)

**Restriction:** Available only on the BLOCK, HBAR, and VBAR statements

**Interaction:** If the SAS system option OVP is in effect and if your printing device supports overprinting, you can specify up to three characters to produce overprinted charts.

**Featured in:** Example 6 on page 203

TYPE=statistic

specifies what the bars or sections in the chart represent. The statistic is one of the following:

CFREQ

specifies that each bar, block, or section represent the cumulative frequency.
CPERCENT
   specifies that each bar, block, or section represent the cumulative percentage.
   Alias: CPCT

FREQ
   specifies that each bar, block, or section represent the frequency with which a
   value or range occurs for the chart variable in the data.

MEAN
   specifies that each bar, block, or section represent the mean of the SUMVAR=
   variable across all observations belonging to that bar, block, or section.
   Interaction: With TYPE=MEAN, you can only compute MEAN and FREQ statistics.
   Featured in: Example 4 on page 200

PERCENT
   specifies that each bar, block, or section represent the percentage of observations
   that have a given value or that fall into a given range of the chart variable.
   Alias: PCT
   Featured in: Example 2 on page 194

SUM
   specifies that each bar, block, or section represent the sum of the SUMVAR=
   variable for the observations corresponding to each bar, block, or section.
   Default: FREQ (unless you use SUMVAR=, which causes a default of SUM)
   Interaction: With TYPE=SUM, you can only compute SUM and FREQ statistics.

WIDTH=n
   specifies the width of the bars on bar charts.

Restriction: Available only on the HBAR and VBAR statements

---

Concepts

---

Variable Characteristics

- Character variables and formats cannot exceed a length of 16.
- For continuous numeric variables, PROC CHART automatically selects display
  intervals, although you can explicitly define interval midpoints.
- For character variables and discrete numeric variables, which contain several
  distinct values rather than a continuous range, the data values themselves define
  the intervals.

---

Results

---

Missing Values
Missing values are not considered as valid levels for the chart variable when you use the MISSING option.

Missing values for a GROUP= or SUBGROUP= variable are treated as valid levels.

PROC CHART ignores missing values for the FREQ= option and the SUMVAR= option.

If the value of the FREQ= variable is missing, zero, or negative, the observation is excluded from the calculation of the chart statistic.

If the value of the SUMVAR= variable is missing, the observation is excluded from the calculation of the chart statistic.

---

Examples

With PROC CHART, you can produce several types of charts within a single PROC step, but in this chapter, each example shows only one chart.

Example 1: Producing a Simple Frequency Count

**Procedure features:**
- **VBAR statement**

This example produces a vertical bar chart that shows a frequency count for the values of the chart variable.

**Program**

```plaintext
options nodate pageno=1 linesize=80 pagesize=60;

data shirts;
   input Size $ @@;
datalines;
medium   large
large    large
large    medium
medium   small
small    medium
medium   large
small    medium
large    large
large    small
medium   medium
medium   medium
```

The data set SHIRTS contains the sizes of a particular shirt sold during a week at a clothing store, one observation for each shirt sold.
The VBAR statement produces a vertical bar chart for the frequency counts of the Size values.

```sas
proc chart data=shirts;
  vbar size;
  title 'Number of Each Shirt Size Sold';
run;
```
Output

The frequency chart shows the store's sales of the shirt for the week: 9 large shirts, 11 medium shirts, and 6 small shirts.

<table>
<thead>
<tr>
<th>Frequency</th>
<th>Number of Each Shirt Size Sold</th>
</tr>
</thead>
<tbody>
<tr>
<td>Large</td>
<td>1</td>
</tr>
<tr>
<td>Medium</td>
<td>1</td>
</tr>
<tr>
<td>Small</td>
<td>1</td>
</tr>
</tbody>
</table>

Example 2: Producing a Percentage Bar Chart

Procedure features:

**VBAR statement option:**
This example produces a vertical bar chart. The chart statistic is the percentage for each category of the total number of shirts sold.

**Program**

```plaintext
options nodate pageno=1 linesize=80 pagesize=60;

proc chart data=shirts;
   vbar size / type=percent;
   title 'Percentage of Total Sales for Each Shirt Size';
run;
```

The **VBAR** statement produces a vertical bar chart. **TYPE=** specifies percentage as the chart statistic for the variable **Size**.
The chart shows the percentage of total sales for each shirt size. Of all the shirts sold, about 42.3 percent were medium, 34.6 were large, and 23.1 were small.

<table>
<thead>
<tr>
<th>Percentage</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>40</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>

Example 3: Subdividing the Bars into Categories

Procedure features:
VBAR statement options:
SUBGROUP=
SUMVAR=

This example

- produces a vertical bar chart for categories of one variable with bar lengths that represent the values of another variable.
- subdivides each bar into categories based on the values of a third variable.

**Program**

```sas
options nodate pageno=1 linesize=80 pagesize=60;

The PIESALES data set contains the number of each flavor of pie sold for two years at three bakeries owned by the same company - one bakery on Samford Avenue, one on Oak Street, and one on Clyde Drive.

data piesales;
  input Bakery $ Flavor $ Year Pies_Sold;
datalines;
  Samford apple 1995 234
  Samford apple 1996 288
  Samford blueberry 1995 103
  Samford blueberry 1996 143
  Samford cherry 1995 173
  Samford cherry 1996 195
  Samford rhubarb 1995 26
  Samford rhubarb 1996 28
  Oak apple 1995 319
  Oak apple 1996 371
  Oak blueberry 1995 174
  Oak blueberry 1996 206
  Oak cherry 1995 246
  Oak cherry 1996 311
  Oak rhubarb 1995 51
  Oak rhubarb 1996 56
  Clyde apple 1995 313
  Clyde apple 1996 415
  Clyde blueberry 1995 177
  Clyde blueberry 1996 201
  Clyde cherry 1995 250
  Clyde cherry 1996 328
  Clyde rhubarb 1995 60
  Clyde rhubarb 1996 59
;
```
The VBAR statement produces a vertical bar chart with one bar for each pie flavor. SUBGROUP= divides each bar into sales for each bakery.

```plaintext
proc chart data=piesales;
  vbar flavor / subgroup=bakery
sumvar=pies_sold;
  title 'Pie Sales by Flavor Subdivided by Bakery Location';
run;
```
The bar that represents the sales of apple pies, for example, shows 1,940 total pies across both years and all three bakeries. The symbol for the Samford Avenue bakery represents the 522 pies at the top, the symbol for the Oak Street bakery represents the 690 pies in the middle, and the symbol for the Clyde Drive bakery represents the 728 pies at the bottom of the bar for apple pies. By default, the labels along the horizontal axis are truncated to eight characters.

<table>
<thead>
<tr>
<th>Pies_Sold Sum</th>
<th>Flavor Subdivided by Bakery Location</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>apple</td>
</tr>
<tr>
<td>1800 +</td>
<td>SSSSS</td>
</tr>
<tr>
<td>1600 +</td>
<td>SSSSS</td>
</tr>
<tr>
<td>1400 +</td>
<td>SSSSS</td>
</tr>
<tr>
<td>1200 +</td>
<td>SSSSS</td>
</tr>
<tr>
<td>1000 +</td>
<td>SSSSS</td>
</tr>
<tr>
<td>800 +</td>
<td>SSSSS</td>
</tr>
<tr>
<td>600 +</td>
<td>SSSSS</td>
</tr>
<tr>
<td>400 +</td>
<td>SSSSS</td>
</tr>
<tr>
<td>200 +</td>
<td>SSSSS</td>
</tr>
</tbody>
</table>

Symbol Bakery: C Clyde  O Oak  S Samford
Example 4: Producing Side-by-Side Bar Charts

Procedure features:
- **VBAR statement options:**
  - GROUP=
  - REF=
  - SUMVAR=
  - TYPE=

Data set: PIESALES on page 197

This example charts the mean values of a variable for the categories of another variable, creates side-by-side bar charts for the categories of a third variable, and draws reference lines across the charts.

Program

```sas
options nodate pageno=1 linesize=80 pagesize=60;

proc chart data=piesales;
    vbar flavor / group=bakery
        ref=100 200 300
        sumvar=pies_sold
        type=mean;
    title 'Mean Yearly Pie Sales Grouped by Flavor';
    title2 'within Bakery Location';
run;
```

The VBAR statement produces side-by-side vertical bar charts to compare the sales across values of Bakery, specified by GROUP=. Each Bakery group contains a bar for each Flavor value. REF= draws reference lines to mark sales at 100, 200, and 300.

SUMVAR= specifies Pies_Sold as the variable that is represented by the lengths of the bars.

TYPE= averages the sales for 1995 and 1996 for each combination of bakery and flavor.
The side-by-side bar charts compare the sales of apple pies, for example, across bakeries. The mean for the Clyde Drive bakery is 364, the mean for the Oak Street bakery is 345, and the mean for the Samford Avenue bakery is 261.
Example 5: Producing a Horizontal Bar Chart for a Subset of the Data

Procedure features:
- HBAR statement options:
  - GROUP=
  - SUMVAR=

Other features:
- WHERE = data set option

Data set: PIESALES on page 197

This example
- produces horizontal bar charts only for observations with a common value
- charts the values of a variable for the categories of another variable
- creates side-by-side bar charts for the categories of a third variable.

Program

```plaintext
options nodate pageno=1 linesize=80 pagesize=60;
WHERE= limits the chart to only the 1995 sales totals.
proc chart data=piesales(where=(year=1995));

The HBAR statement produces side-by-side horizontal bar charts to compare sales across values of Flavor, specified by GROUP=. Each Flavor group contains a bar for each Bakery value.

hbar bakery / group=flavor

SUMVAR= specifies Pies_Sold as the variable whose values are represented by the lengths of the bars.

sumvar=pies_sold;
title '1995 Pie Sales for Each Bakery According to Flavor';
run;
```

Output
### Example 6: Producing Block Charts for BY Groups

**Procedure features:**
- BLOCK statement options:
  - GROUP=
  - NOHEADER=
  - SUMVAR=
  - SYMBOL=
- BY statement

**Other features:**
- PROC SORT
- SAS System options:
  - NOBYLINE
  - OVP
- TITLE statement:
  - #BYVAL specification

**Data set:** PIESALES on page 197

---

This example
- sorts the data set
- produces a block chart for each BY group
- organizes the blocks into a three-dimensional chart
- prints BY group-specific titles.

### 1995 Pie Sales for Each Bakery According to Flavor

<table>
<thead>
<tr>
<th>Flavor</th>
<th>Bakery</th>
<th>Pies_Sold Sum</th>
</tr>
</thead>
<tbody>
<tr>
<td>apple</td>
<td>Clyde</td>
<td>313.0000</td>
</tr>
<tr>
<td></td>
<td>Oak</td>
<td>319.0000</td>
</tr>
<tr>
<td></td>
<td>Samford</td>
<td>234.0000</td>
</tr>
<tr>
<td>blueberries</td>
<td>Clyde</td>
<td>177.0000</td>
</tr>
<tr>
<td></td>
<td>Oak</td>
<td>174.0000</td>
</tr>
<tr>
<td></td>
<td>Samford</td>
<td>103.0000</td>
</tr>
<tr>
<td>cherry</td>
<td>Clyde</td>
<td>250.0000</td>
</tr>
<tr>
<td></td>
<td>Oak</td>
<td>246.0000</td>
</tr>
<tr>
<td></td>
<td>Samford</td>
<td>173.0000</td>
</tr>
<tr>
<td>rhubarb</td>
<td>Clyde</td>
<td>60.0000</td>
</tr>
<tr>
<td></td>
<td>Oak</td>
<td>51.0000</td>
</tr>
<tr>
<td></td>
<td>Samford</td>
<td>26.0000</td>
</tr>
</tbody>
</table>

---

### Pies Sold Sum

<table>
<thead>
<tr>
<th>Pies_Sold Sum</th>
</tr>
</thead>
<tbody>
<tr>
<td>30 60 90 120 150 180 210 240 270 300</td>
</tr>
</tbody>
</table>
Program

options nodate pageno=1 linesize=80 pagesize=60;

PROC SORT sorts PIESALES by year. This is required to produce a separate chart for each year.

proc sort data=piesales out=sorted_piesales;
  by year;
run;

NOBYLINE suppresses the usual BY lines in the output. OVP allows overprinted characters in
the charts.

options nobyline ovp;

The BY statement produces one chart for 1995 sales and one for 1996 sales.

proc chart data=sorted_piesales;
  by year;

The BLOCK statement produces a block chart for each year. Each chart contains a grid (Bakery
values along the bottom, Flavor values along the side) of cells that contain the blocks.

    block bakery / group=flavor

SUMVAR= specifies Pies_Sold as the variable whose values are represented by the lengths of the
blocks.

    sumvar=pies_sold

NOHEADER suppresses the default header line

    noheader

SYMBOL= specifies the symbols in the blocks.
symbol='OX';
title 'Pie Sales for Each Bakery and Flavor';

The #BYVAL specification inserts the year into the second line of the title.

title2 '#byval(year)';
run;

The SAS system option BYLINE resets the printing of the default BY line.

options byline;

Output

Pie Sales for Each Bakery and Flavor
1995

Cherry

Blueberry

Apple

Bakery
References