Chapter 25

The PLOT Procedure

ABSTRACT 397
INTRODUCTION 397
SPECIFICATIONS 406
  PROC PLOT Statement 406
  BY Statement 408
  PLOT Statement 409
DETAILS 416
  Missing Values 416
  Hidden Observations 416
  RUN Groups 416
  Generating Data with Program Statements 416
  Printed Output 417
EXAMPLES 418
  Example 1: Plotting Observed Data 418
  Example 2: Using a Plot Character and Defining Tick Marks 420
  Example 3: Plotting the Graph of an Equation 422
  Example 4: Plotting Predicted versus Actual Values 424
  Example 5: Contour Plotting 426
  Example 6: Labeling an Axis with Date Values 428
  Example 7: The Effect of the HEXPAND Option 430
  Example 8: The Effect of the NOMISS Option 431

ABSTRACT

The PLOT procedure graphs one variable against another, producing a printer plot. The coordinates of each point on the plot correspond to the two variables’ values in one or more observations of the input data set.

INTRODUCTION

The PLOT procedure plots the values of two variables for each observation in an input SAS data set. All you need to do to produce a plot is to tell the procedure which variables to plot. The examples in this section all use data about the high and low values of the Dow Jones Industrial Average between 1954 and 1987.
You can also use the PLOT procedure to

- plot character as well as numeric variables
- specify the length and width of the plot
- reverse the order of the values on either axis
- draw contour plots with shading intensity determined by a third variable in the data set.

SPECIFICATIONS

The PLOT procedure is controlled by the following statements:

```plaintext
PROC PLOT <option-list>;
   BY variable-list;
   PLOT request-list /< option-list>;
```

A PLOT statement must be present to tell the procedure which variables to plot. Any number of PLOT statements can be specified each time the procedure is invoked, and any number of plots can be requested in one PLOT statement.

PROC PLOT Statement

```plaintext
PROC PLOT <option-list>;
```

The PROC PLOT statement supports a variety of options. Table 25.1 summarizes these options. Detailed descriptions follow the table in alphabetic order.

<table>
<thead>
<tr>
<th>Class</th>
<th>Option</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input option</td>
<td>DATA=</td>
<td>specifies the input data set</td>
</tr>
<tr>
<td>Axis options</td>
<td>UNIFORM</td>
<td>scales axes uniformly</td>
</tr>
<tr>
<td></td>
<td>NOMISS</td>
<td>excludes observations with missing values from calculation of axes</td>
</tr>
<tr>
<td>Appearance</td>
<td>NOLEGEND</td>
<td>suppresses the legend at the top of each plot</td>
</tr>
<tr>
<td>options</td>
<td>VTOH=</td>
<td>specifies the aspect ratio of the output device</td>
</tr>
<tr>
<td></td>
<td>FORMCHAR=</td>
<td>specifies the characters to use to construct the borders of the plot</td>
</tr>
<tr>
<td>Sizing options</td>
<td>HPERCENT=</td>
<td>specifies the percentage of the horizontal page to use for each plot</td>
</tr>
<tr>
<td></td>
<td>VPERCENT=</td>
<td>specifies the percentage of the vertical page to use for each plot</td>
</tr>
</tbody>
</table>

DATA=SAS-data-set

names the input SAS data set to use. If you do not specify the DATA= option, PROC PLOT uses the most recently created SAS data set.
FORMCHAR <(index-list)> = 'formchar-string'
specifies the characters to use to construct the borders of the plot. The
formchar-string is a string up to 11 characters long defining the 2 bar
characters, vertical and horizontal, and the 9 corner characters:
upper left, upper middle, upper right, middle left, middle middle (cross),
middle right, lower left, lower middle, and lower right. Of these, PROC
PLOT uses only the horizontal and vertical bars, the four corners, and
the cross. If the FORMCHAR= option in the PROC PLOT statement is
not given, the procedure uses the formchar-string supplied with the
FORMCHAR= system option. The default for the system option is
FORMCHAR= '---- | + | --'.
You can use any character string or a string of hexadecimal constants
to customize the plot's appearance. Use an index-list to specify which
default form character each supplied character replaces, or replace the
entire default string by specifying the full 11-character replacement string
with no index list. For example, change the four corners to asterisks by
using

formchar\(3 5 9 11)\* ****
Specifying eleven blanks, as follows, produces plots with no borders:

formchar= '
PROC PLOT uses only the following FORMCHAR characters:

<table>
<thead>
<tr>
<th>Character</th>
<th>Position</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>vertical bar</td>
<td>1</td>
<td>|</td>
</tr>
<tr>
<td>horizontal bar</td>
<td>2</td>
<td>-</td>
</tr>
<tr>
<td>upper left corner</td>
<td>3</td>
<td>-</td>
</tr>
<tr>
<td>upper right corner</td>
<td>5</td>
<td>-</td>
</tr>
<tr>
<td>cross (for tick marks)</td>
<td>7</td>
<td>+</td>
</tr>
<tr>
<td>lower left corner</td>
<td>9</td>
<td>-</td>
</tr>
<tr>
<td>lower right corner</td>
<td>11</td>
<td>-</td>
</tr>
</tbody>
</table>

HPERCENT=percent-list
HPCT=percent-list
specifies the percentage of the horizontal page to use for each plot. By
default, the PLOT procedure uses one page for each plot. After reaching
the end of the percent-list, PROC PLOT cycles back to the beginning of
the list. When you use the HPERCENT= option, the procedure tries to
fit as many plots as possible on a page. For example, this option
specification prints three plots per page horizontally:

hpercent=33
Each plot is one-third of a page wide.
The next option specification also prints three plots per page, but the
first one is twice as big as the other two:

hpercent=50 25 25
By specifying a zero in the list, you can force PROC PLOT to go to a
new page even though it could fit the next plot on the same page.
For example, this option specification produces plots that are only one-third of a page wide, but each plot is on a separate page:

\[ \text{hpercent=33 0} \]

You can use the \text{HPERCENT=} option to print very wide plots. For example, this example produces plots three pages wide:

\[ \text{hpercent=100} \]

At the beginning of every \text{BY} group and after each \text{RUN} statement, \text{PROC PLOT} returns to the beginning of the \text{percent-list} and starts printing a new page.

\text{NOLEGEND}

suppresses the legend at the top of each plot. The legend lists the names of the variables being plotted and the plotting symbols used in the plot.

\text{NOMISS}

excludes observations for which either variable is missing from the calculation of the axes. Normally, \text{PROC PLOT} draws an axis based on all the values of the variable being plotted, including points for which the other variable is missing. The \text{HAXIS=} option overrides the effect of the \text{NOMISS} option on the horizontal axis; the \text{VAXIS=} option overrides the effect on the vertical axis.

\text{UNIFORM}

uniformly scales axes across \text{BY} groups. Uniform scaling allows you to directly compare the plots for different values of the \text{BY} variables.

\text{VPERCENT=percent-list}
\text{VPCT=percent-list}

specifies the percentage of the vertical page to use for each plot, just as \text{HPERCENT=} specifies the horizontal percentage. If you use a percentage greater than 100, the \text{PLOT} procedure prints sections of the plot on successive pages.

\text{VTOH=character-height/character-width}

specifies the aspect ratio (vertical to horizontal) of the characters on the output device. If you use the \text{VTOH=} option, \text{PROC PLOT} spaces tick marks so that the distance between horizontal tick marks is nearly equal to the distance between vertical tick marks. The \text{VTOH=} option has no effect if you use the \text{HSPACE=} and the \text{VSPACE=} options in the \text{PLOT} statement.

\text{BY Statement}

\text{BY} variable-list;

A \text{BY} statement can be used with \text{PROC PLOT} to obtain separate plots on observations in groups defined by the \text{BY} variables.

When a \text{BY} statement appears, the procedure expects the input data set to be sorted in order of the \text{BY} variables or to have an appropriate index. If your input data set is not sorted in ascending order, you can do one of the following:

\begin{itemize}
  \item Use the \text{SORT} procedure with a similar \text{BY} statement to sort the data.
  \item If appropriate, use the \text{BY} statement option \text{NOTSORTED} or \text{DESCENDING}.
  \item Create an index on the \text{BY} variables you want to use. For more information on creating indexes and using the \text{BY} statement with indexed data sets, see Chapter 17, “The DATASETS Procedure.”
\end{itemize}

\text{PROC PLOT} produces a new page each time the value of the \text{BY} variable changes.
PLOT Statement

PLOT request-list</option-list>;

The PLOT statement requests the plots to be produced by PROC PLOT. You must use at least one PLOT statement. You can include many PLOT statements, and you can specify many plot requests in one PLOT statement.

Each element of the request-list in the PLOT statement specifies the variables (vertical and horizontal) to plot and the plotting symbol to use to mark the points on the plot. The request can take the following forms:

vertical*horizontal
names the variable to plot on the vertical axis and the variable to plot on the horizontal axis.

For example, the following statements request a plot of Y by X:

```
proc plot;
   plot y*x;
run;
```

Y appears on the vertical axis, X on the horizontal axis.

This form of the plot request uses the default method of choosing a plotting symbol to mark plot points. When a point on the plot represents the values of one observation in the data set, PROC PLOT puts the character A at that point. When a point represents the values of two observations, the character B appears. When a point represents values of three observations, the character C appears, and so on through the alphabet. The character Z is used for the occurrence of 26 or more observations at the same printing position.

vertical*horizontal=’character’
names the variables to plot on the vertical and horizontal axes and specifies a plotting symbol to mark each point on the plot. A single character is used to represent values from one or more observations.

For example, the following statements request a plot of Y by X, with each point on the plot represented by a plus sign (+):

```
proc plot;
   plot y*x=’+’;
run;
```

vertical*horizontal=variable
names the variables to plot on the vertical and horizontal axes and specifies a variable whose values are to mark each point on the plot. The variable can be either numeric or character. The first (left-most) nonblank character in the formatted value of the variable is used as the plotting symbol. When more than one observation maps to the same plotting position, the value from the first observation marks the point.

For example, in the following statements SEX is a character variable with values of FEMALE and MALE: the values F and M mark each observation on the plot:

```
proc plot;
   plot height*weight=sex;
run;
```
Note: The plotting symbol is the first nonblank character of each value even if more than one value starts with the same letter.
To request two or more plots, write one request after another:

```
proc plot;
  plot a*b r=s;
run;
```

If you want to plot all the combinations of one set of variables with another, you can use a grouping specification. Enclose each set of variables in parentheses, joining them with an asterisk (*). For example, the following PLOT statements are equivalent:

```
plot (y x)*(a b);
```

```
plot y*a y*b x*a x*b;
```

You can also abbreviate a variable list to request a number of plots:
```
plot y*(a--z);
```

If both the vertical and horizontal specifications request more than one variable and a variable appears in both lists, it will not be plotted against itself. For example, the following statement plots all combinations of the variables A, B, and C with B, C, and D, but it does not plot B*B and C*C:

```
plot (a b c)*(b c d);
```

To plot all unique combinations of a list of variables, simply omit the second list. Thus, the following two statements are equivalent:
```
plot (a b c);
```

```
plot a*b a*c b*c;
```

Table 25.2 summarizes the options you can use in the PLOT statement. A slash (/) separates these options from the plot requests. No slash is needed if no options are specified. Detailed descriptions of the options follow the table in alphabetic order.

<table>
<thead>
<tr>
<th>Table 25.2 Summary of PLOT Statement Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
</tr>
<tr>
<td>-------</td>
</tr>
<tr>
<td>Axis options</td>
</tr>
<tr>
<td>Axis options</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

(continued)
<table>
<thead>
<tr>
<th>Class</th>
<th>Option</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>HREVERSE</td>
<td>reverses the order of the values on the horizontal axis</td>
</tr>
<tr>
<td></td>
<td>VREVERSE</td>
<td>reverses the order of the values on the vertical axis</td>
</tr>
<tr>
<td></td>
<td>HEXPAND</td>
<td>expands the horizontal axis to minimize the margins at the sides of the plot</td>
</tr>
<tr>
<td></td>
<td>VEXPAND</td>
<td>expands the vertical axis to minimize the margins above and below the plot</td>
</tr>
<tr>
<td></td>
<td>HSPACE=</td>
<td>specifies the number of print positions between tick marks on the horizontal axis</td>
</tr>
<tr>
<td></td>
<td>VSPACE=</td>
<td>specifies the number of print positions between tick marks on the vertical axis</td>
</tr>
<tr>
<td>Reference line options</td>
<td>HREF=</td>
<td>draws lines on the plot perpendicular to the specified values on the horizontal axis</td>
</tr>
<tr>
<td></td>
<td>VREF=</td>
<td>draws lines on the plot perpendicular to the specified values on the vertical axis</td>
</tr>
<tr>
<td></td>
<td>HREFCHAR=</td>
<td>specifies the character to use to draw reference HREF= reference lines</td>
</tr>
<tr>
<td></td>
<td>VREFCHAR=</td>
<td>specifies the character to use to draw reference VREF= reference lines</td>
</tr>
<tr>
<td>Appearance option</td>
<td>BOX</td>
<td>draws a border around the entire plot</td>
</tr>
<tr>
<td>Sizing options</td>
<td>HPOS=</td>
<td>specifies the number of print positions on the horizontal axis</td>
</tr>
<tr>
<td></td>
<td>VPOS=</td>
<td>specifies the number of print positions on the vertical axis</td>
</tr>
<tr>
<td>Overlaying option</td>
<td>OVERLAY</td>
<td>overlays all plots specified in the PLOT statement on one set of axes</td>
</tr>
<tr>
<td>Contour options</td>
<td>CONTOUR</td>
<td>draws a contour plot using plotting symbols with varying degrees of shading</td>
</tr>
<tr>
<td></td>
<td>S&lt;level&gt;=</td>
<td>specifies the plotting symbol to use for specified contour level</td>
</tr>
<tr>
<td></td>
<td>SLIST=</td>
<td>specifies plotting symbols for multiple contour levels</td>
</tr>
</tbody>
</table>
BOX
draws a border around the entire plot, rather than just on the left side
and bottom.

CONTOUR<=number-of-levels>
draws a contour plot using plotting symbols with varying degrees of
shading where number-of-levels is the number of levels for dividing the
range of the response variable. The plot request must be of the form
vertical*horizontal=variable where variable is a numeric variable in the
data set. The intensity of shading is determined by the values of this
variable. The value of number-of-levels can range from 1 to 10. If you
specify simply CONTOUR, the default value is 10. For example, these
statements request a plot whose points vary in darkness depending on
the value of Z:

    proc plot;
    plot a*b=z / contour=10;

PROC PLOT uses 10 darkness levels, as specified by the CONTOUR
option.
Overprinting, if it is allowed, is used to produce the shading.
Otherwise, single characters varying in darkness are used. The
CONTOUR option is most effective when the plot is dense.

HAXIS=tick-value-list
specifies the tick-mark values to space equally along the horizontal axis.
When the variable is numeric, you must give HAXIS= values in either
ascending or descending order.
The following statements ask for a plot of Y by X, with tick marks at
10, 15, 20, and so on up to 100 on the horizontal axis:

    proc plot;
    plot y*x / haxis=10 to 100 by 5;
    run;

Numeric values need not be uniformly distributed; a specification of
the following form is valid and produces a logarithmic plot:

    haxis=10 100 1000 10000

If PROC PLOT cannot determine the function implied by the axis
specification, it uses simple linear interpolation between the points.
To determine whether PLOT correctly interpolates a function you wish
to use, you can generate data with the DATA step that determines the
function and see whether it appears linear when plotted. For example,
the following statements produce a linear plot:

    data test;
    do y=1 to 3 by .1;
      x=10**y;
      output;
    end;
    run;

    proc plot data=test;
    plot y*x / haxis=10 100 1000 10000;
    run;

You can list the values of character variables in any order.
In addition, the following HAXIS= specifications are valid:

\texttt{haxis='01JAN85'd to '01JAN86'd by month}

or

\texttt{haxis='01JAN85'd to '01JAN86'd by qtr}

In these examples, the FROM and TO values can be any of the valid SAS date, time, or datetime values described for the SAS functions INTCK and INTNX. (See Chapter 11, "SAS Functions," in SAS Language: Reference.) The BY value can be any of the valid values listed for the \textit{interval} argument in the SAS functions INTCK and INTNX. You must use a \texttt{FORMAT} statement to print the tick-mark values in an understandable form.

\textbf{HEXPAND}

expands the horizontal axis to minimize the margins at the sides of the plot and to maximize the distance between tick marks, if possible.

Normally, PROC PLOT looks at the minimum difference between each pair of the five lowest ordered values of each variable (the \textit{delta}) and ensures that there is no more than one of these intervals per print position on the final scaled axis, if possible. If there is not enough room to do this, and if PROC PLOT guesses that the data were artificially generated, it puts a fixed number of deltas in each print position. Otherwise it ignores the value.

\textbf{HPOS=axis-length}

specifies the number of print positions on the horizontal axis. The maximum value of \textit{axis-length} that allows a plot to fit on one page is three positions less than the value of the \texttt{LINESIZE=} system option because you must allow room for the procedure to print information next to the vertical axis. The exact maximum depends on the number of characters in the vertical variable's values. If \textit{axis-length} is too large to fit on a line, PROC PLOT ignores the option.

\textbf{HREF=value-list}

draws lines on the plot perpendicular to the specified values on the horizontal axis. PROC PLOT includes the values you specify with the HREF= option on the horizontal axis unless you specify otherwise with the HAXIS= option.

For example, the following statements request a plot of Y by X with a line perpendicular to the value 5 on the horizontal axis:

\begin{verbatim}
proc plot,
   plot y*x / href=5;
\end{verbatim}

The following statements draw a plot with reference lines intersecting the horizontal axis at 10, 20, 30, and so on up to 100:

\begin{verbatim}
proc plot;
   plot y*x / href=10 to 100 by 10;
\end{verbatim}

\textbf{HREFCHAR='character'}

specifies the character to use to draw the HREF= reference lines. If you do not specify a character with the HREFCHAR= option, PROC PLOT uses the vertical bar character (\texttt{|}) by default. (See the FORMCHAR= option earlier in PROC PLOT Statement.)

\textbf{HREVERSE}

reverses the order of the values on the horizontal axis.

\textbf{HSPACE=number-of-print-positions}

specifies the number of print positions between tick marks on the horizontal axis.
HZERO
assigns a value of zero to the first tick mark on the horizontal axis.
PROC PLOT ignores the HZERO option if the horizontal variable has
negative values or if the HAXIS= option specifies a range that does not
begin with zero.

OVERLAY
overlays all plots specified in the PLOT statement on one set of axes.
The variables (or labels of variables if specified in the PROC step) from
the first plot label the axes. Unless you use the HAXIS= or the VAXIS=
option, PROC PLOT automatically scales the axes in the way that best
fits all the variables.
When the SAS system option OVP is in effect and overprinting is
allowed, the plots are superimposed; otherwise, when NOOVP is in
effect, PROC PLOT uses the plotting symbol from the first plot to
represent points appearing in more than one plot. In such a case, the
output includes a message telling you how many observations are
hidden.

S<contour-level> = 'character-list'
specifies the plotting symbol to use for a single contour level. You can
use this option repeatedly. When PROC PLOT produces contour plots,
it automatically chooses the symbols to use for each level of intensity.
You can use the S option to override these symbols and specify your
own. The contour-level is a whole number between 1 and the highest
contour level (determined by the CONTOUR option). You can include
up to three characters in character-list. If overprinting is not allowed,
PROC PLOT uses only the first character.

For example, to specify three levels of shading for the Z variable, use
the following statements:

proc plot;
    plot y*x=z / contour=3
        s1='A' s2='+' s3='XOA';
run;

You can also specify the plotting symbols as hexadecimal constants:

proc plot;
    plot y*x=z / contour=3
        s1='7A' s2='7F' s3='A6';
run;

This feature was designed especially for printers where the hex constants
can represent grey-scale fill characters.
See the SLIST= option for an alternate way to specify plotting
symbols for a contour plot.

SLIST = 'character-list-1' <...'character-list-n'>
specifies plotting symbols for multiple contour levels. Each character-list
specifies the plotting symbol for one contour level: the first character-list
for the first level, the second character-list for the second level, and so
on.

For example, the following statements are equivalent:

plot y*x=z / contour=5 s1='.' s2=':' s3='!' s4='-' s5='+O';

plot y*x=z / contour=5 slist='.' ':' '!': '=' 'O';
If you do not specify a plotting symbol for each contour level, PROC PLOT uses the default symbols for the remaining levels. By default, PROC PLOT uses the equivalent of

\[ \text{slist} = \text{'. ' ', '-' ' ' '+' ' 0 ' 'X' ' 'W' ' '9' ' 'I'} \]

This sequence is normally satisfactory for contour plots because the symbols appear as lightest (.) to darkest (#); however, you can change any of these symbols using the S= option. For example, some printers print the asterisk (*) as a very light symbol. If this is the case with your printer, you can substitute another, darker, symbol for the asterisk (the ninth symbol in the list) by specifying, for example, the following statement in the PROC PLOT statement:

\[ \text{s9 = '6'} \]

**VAXIS=** *tick-value-list*

specifies the tick-mark values to space equally along the vertical axis. The VAXIS= option follows the same rules as the HAXIS= option, described earlier.

**VEXPAND**

expands the vertical axis to minimize the margins above and below the plot and to maximize the space between vertical tick marks, if possible. The behavior of the VEXPAND option is the same as that of the HEXPAND option, described earlier.

**VPOS=** *axis-length*

specifies the number of print positions on the vertical axis. The maximum value for *axis-length* that allows a plot to fit on one page is 8 lines less than the value of the PAGESIZE= system option because you must allow room for the procedure to print information under the horizontal axis. The exact maximum depends on the titles used, whether or not plots are overlayed, and whether or not CONTOUR is specified. If the value of *axis-length* specifies a plot that cannot fit on one page, the plot spans multiple pages.

**VREF=** *value-list*

draws lines on the plot perpendicular to the specified values on the vertical axis. The VREF= option behaves like the HREF= option, described earlier.

**VREFCHAR=** *character*

specifies the character to use to draw the VREF= reference lines. If you do not specify a character with the VREFCHAR= option, PROC PLOT uses the horizontal bar character (-) by default. (See the FORMCHAR= option earlier in PROC PLOT Statement.)

**VREVERSE**

reverses the order of the values on the vertical axis.

**VSPACE=** *number-of-lines*

specifies the number of print lines between tick marks on the vertical axis.

**VZERO**

assigns a value of zero to the first tick mark on the vertical axis. PROC PLOT ignores the VZERO option if the vertical variable has negative values or if the VAXIS= option specifies a range that does not begin with zero.