CHAPTER
38
The TIMEPLOT Procedure

Overview

The TIMEPLOT procedure plots one or more variables over time intervals. A listing of variable values accompanies the plot. Although the plot and the listing are similar to those produced by the PLOT and PRINT procedures, PROC TIMEPLOT output has these distinctive features:

- The vertical axis always represents the sequence of observations in the data set; thus, if the observations are in order of date or time, the vertical axis represents the passage of time.
- The horizontal axis represents the values of the variable that you are examining. Like PROC PLOT, PROC TIMEPLOT can overlay multiple plots on one set of axes so that each line of the plot can contain values for more than one variable.
- A plot produced by PROC TIMEPLOT may occupy more than one page.
- Each observation appears sequentially on a separate line of the plot; PROC TIMEPLOT does not hide observations as PROC PLOT sometimes does.
- The listing of the plotted values may include variables that do not appear in the plot.

Output 38.1 on page 1248 illustrates a simple report that you can produce with PROC TIMEPLOT. This report shows sales of refrigerators for two sales representatives during the first six weeks of the year. The statements that produce the output follow. A DATA step on page 1258 creates the data set SALES.

```
options linesize=64 pagesize=60 nodate
        pageno=1;


proc timeplot data=sales;
   plot icebox;
   id month week;
   title 'Weekly Sales of Refrigerators';
   title2 'for the';
   title3 'First Six Weeks of the Year';
run;
```

Output 38.1 Simple Report Created with PROC TIMEPLOT

```
                Weekly Sales of Refrigerators                1
                           for the
                    First Six Weeks of the Year

Month     Week      Icebox      min                          max
                                2520.04                   3550.43
                              *-------------------------------*
    1        1     3450.94     |                              I  |
    1        1     2520.04     |I                                |
    1        2     3240.67     |                        I        |
    1        2     2675.42     |     I                           |
    1        3     3160.45     |                    I            |
    1        3     2805.35     |          I                      |
    1        4     3400.24     |                             I   |
    1        4     2870.61     |            I                    |
    2        1     3550.43     |                               I |
    2        1     2730.09     |       I                         |
    2        2     3385.74     |                             I   |
    2        2     2670.93     |     I                           |
                              *-------------------------------*
```

Output 38.2 on page 1248 is a more complicated report of the same data set that is used to create Output 38.1 on page 1248. The statements that create this report

- create one plot for the sale of refrigerators and one for the sale of stoves
- plot sales for both sales representatives on the same line
- identify points on the plots by the first letter of the sales representative's last name
- control the size of the horizontal axis
- control formats and labels.

For an explanation of the program that produces this report, see Example 5 on page 1266.

Output 38.2 More Complex Report Created with PROC TIMEPLOT

```
                Weekly Appliance Sales for the First Quarter                1

              Seller :Kreitz  Seller :LeGrange
   Month  Week           Stove              Stove      min                   max
                                                       $184.24          $2,910.37
                                                      *-------------------------*
January       1   $1,312.61          $728.13           |     L    K              |
January       2     $222.35          $184.24           |!                        |
January       3   $2,263.33          $267.35           | L                 K     |
January       4   $1,787.45          $274.51           | L             K       |
February      1   $2,910.37          $397.98           |  L                     K|
February      2     $819.69        $2,242.24           |      K            L     |
                                                      *-------------------------*
```

```
                Weekly Appliance Sales for the First Quarter                2

                     Kreitz          LeGrange
   Month  Week          Icebox            Icebox      min                   max
                                                       $2,520.04        $3,550.43
                                                      *-------------------------*
January       1   $3,450.94        $2,520.04           |L                      K |
January       2   $3,240.67        $2,675.42           |    L            K       |
January       3   $3,160.45        $2,805.35           |       L        K        |
January       4   $3,400.24        $2,870.61           |         L           K   |
February      1   $3,550.43        $2,730.09           |     L                  K|
February      2   $3,385.74        $2,670.93           |    L                K   |
                                                      *-------------------------*
```

Procedure Syntax

Requirements: At least one PLOT statement

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 TIMEPLOT <DATA=*SAS-data-set*>
<MAXDEC=*number*> <UNIFORM>;

BY <DESCENDING> *variable-1*
<…<DESCENDING> *variable-n*>
<NOTSORTED>;

CLASS *variable(s)*;

ID *variable(s)*;

PLOT *plot-request(s)/option(s)*;

To do this	Use this statement
Produce a separate plot for each BY group	BY
Group data according to the values of the class variables	CLASS
Print in the listing the values of the variables that you identify	ID
Specify the plots to produce	PLOT

PROC TIMEPLOT Statement

PROC TIMEPLOT <DATA=*SAS-data-set*>
<MAXDEC=*number*> <UNIFORM>;

Options

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

MAXDEC=*number*
specifies the maximum number of decimal places to print in the listing.

Interaction: A decimal specification in a format overrides a MAXDEC= specification.

Default: 2

Range: 0-12

Featured in: Example 4 on page 1264

UNIFORM
uniformly scales the horizontal axis across all BY groups. By default, PROC TIMEPLOT separately determines the scale of the axis for each BY group.

Interaction: UNIFORM also affects the calculation of means for reference lines (see REF= on page 1256).

BY Statement

Produces a separate plot for each BY group.

Main discussion: "BY" on page 68

BY <DESCENDING> *variable-1*
<...<DESCENDING> *variable-n*>
<NOTSORTED>;

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. These variables are called *BY variables*.

Options

DESCENDING
specifies that the data set is 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 data 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.

CLASS Statement

Groups data according to the values of the class variables.

Tip: PROC TIMEPLOT uses the formatted values of the CLASS variables to form classes. Thus, if a format groups the values, the procedure uses those groups.

Featured in: Example 5 on page 1266

CLASS *variable(s)*;

Required Arguments

variable(s)
specifies one or more variables that the procedure uses to group the data. Variables in a CLASS statement are called *class variables*. Class variables can be numeric or character. Class variables can have continuous values, but they typically have a few discrete values that define the classifications of the variable. You do not have to sort the data by class variables.

The values of the class variables appear in the listing. PROC TIMEPLOT prints and plots one line each time the combination of values of the class variables changes. Therefore, the output typically is more meaningful if you sort or group the data according to values of the class variables.

Using Multiple CLASS Statements

You can use any number of CLASS statements. If you use more than one CLASS statement, PROC TIMEPLOT simply concatenates all variables from all of the CLASS statements. The following form of the CLASS statement includes three variables:

CLASS *variable-1 variable-2 variable-3*;

It has the same effect as this form:

CLASS *variable-1*;

CLASS *variable-2*;

CLASS *variable-3*;

Using a Symbol Variable

Normally, you use the CLASS statement with a symbol variable (see the discussion of plot requests on page 1254). In this case, the listing of the plot variable contains a column for each value of the symbol variable, and each row of the plot contains a point for each value of the symbol variable. The plotting symbol is the first character of the formatted value of the symbol variable. If more than one observation within a class has the same value of a symbol variable, PROC TIMEPLOT plots and prints only the first occurrence of that value and writes a warning message to the SAS log.

ID Statement

Prints in the listing the values of the variables that you identify.

Featured in: Example 1 on page 1258

ID *variable(s)*;

Required Arguments

variable(s)
identifies one or more *ID variables* to print in the listing.

PLOT Statement

Specifies the plots to produce.

Tip: Each PLOT statement produces a separate plot.

PLOT *plot-request(s)/option(s)*;

Table 38.1 on page 1253 summarizes the options available in the PLOT statement.

Table 38.1 Summary of Options for the PLOT Statement

To do this	Use this option
Customize the axis	
Specify the range of values to plot on the horizontal axis, as well as the interval represented by each print position on the horizontal axis	AXIS=
Order the values on the horizontal axis with the largest value in the leftmost position	REVERSE
Control the appearance of the plot	
Connect the leftmost plotting symbol to the rightmost plotting symbol with a line of hyphens (-)	HILOC
Connect the leftmost and rightmost symbols on each line of the plot with a line of hyphens (-) regardless of whether the symbols are reference symbols or plotting symbols	JOINREF
Suppress the name of the symbol variable in column headings when you use a CLASS statement	NOSYMNAME
Suppress the listing of the values of the variables that appear in the PLOT statement	NPP
Specify the number of print positions to use for the horizontal axis	POS=
Create and customize a reference line	
Draw lines on the plot that are perpendicular to the specified values on the horizontal axis	REF=
Specify the character for drawing reference lines	REFCHAR=
Display multiple plots on the same set of axes	
Plot all requests in one PLOT statement on one set of axes	OVERLAY
Specify the character to print if multiple plotting symbols coincide	OVPCHAR=

Required Arguments

plot-request(s)
specifies the variable or variables to plot and, optionally, the plotting symbol to use. By default, each plot request produces a separate plot.

A plot request can have the following forms. You can mix different forms of requests in one PLOT statement (see Example 4 on page 1264).

variable(s)
identifies one or more numeric variables to plot. PROC TIMEPLOT uses the first character of the variable name as the plotting symbol.

Featured in: Example 1 on page 1258

(*variable(s)*)='*plotting-symbol*'
identifies one or more numeric variables to plot and specifies the plotting symbol to use for all variables in the list. You can omit the parentheses if you use only one variable.

Featured in: Example 2 on page 1260

(*variable(s)*)=*symbol-variable*
identifies one or more numeric variables to plot and specifies a *symbol variable*. PROC TIMEPLOT uses the first nonblank character of the formatted value of the symbol variable as the plotting symbol for all variables in the list. The plotting symbol changes from one observation to the next if the value of the symbol variable changes. You can omit the parentheses if you use only one variable.

Featured in: Example 3 on page 1262

Options

AXIS=*axis-specification*
specifies the range of values to plot on the horizontal axis, as well as the interval represented by each print position on the axis. PROC TIMEPLOT labels the first and last ends of the axis, if space permits.

- For numeric values, *axis-specification* can be one of the following or a combination of both:

  n<. . .*n*>

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

  The values must be in either ascending or descending order. Use a negative value for *increment* to specify descending order. The specified values are spaced evenly along the horizontal axis even if the values are not uniformly distributed. Numeric values can be specified in the following ways:

Specification	Comments
`axis=1 2 10`	Values are 1, 2, and 10.
`axis=10 to 100 by 5`	Values appear in increments of 5, starting at 10 and ending at 100.
`axis=12 10 to 100 by 5`	A combination of the two previous forms of specification.

- For axis variables that contain datetime values, *axis-specification* is either an explicit list of values or a starting and an ending value with an increment specified:

  '*date-time-value*'*i* <. . . '*date-time-value*'*i*>

  '*date-time-value*'*i* **TO** '*date-time-value*'*i*
  <**BY** *increment*>

'*date-time-value*'*i*
any SAS date, time, or datetime value described for the SAS functions INTCK and INTNX. The suffix *i* is one of the following:

D	date
T	time
DT	datetime

increment
one of the valid arguments for the INTCK or INTNX functions. For dates, *increment* can be one of the following:

DAY

WEEK

MONTH

QTR

YEAR
For datetimes, *increment* can be one of the following:

DTDAY

DTWEEK

DTMONTH

DTQTR

DTYEAR
For times, *increment* can be one of the following:

HOUR

MINUTE

SECOND
For example,

```
axis='01JAN95'd to '01JAN96'd by month
axis='01JAN95'd to '01JAN96'd by qtr
```

For descriptions of individual intervals, see the chapter on dates, times, and intervals in *SAS Language Reference: Concepts*.

Note: You must use a FORMAT statement to print the tick-mark values in an understandable form. △

Interaction: The value of POS= (see POS= on page 1256) overrides an interval set with AXIS=.

Tip: If the range that you specify does not include all your data, PROC TIMEPLOT uses angle brackets (< or >) on the left or right border of the plot to indicate a value outside the range.

Featured in: Example 2 on page 1260

HILOC
connects the leftmost plotting symbol to the rightmost plotting symbol with a line of hyphens (-).

Interactions: If you specify JOINREF, PROC TIMEPLOT ignores HILOC.

JOINREF
connects the leftmost and rightmost symbols on each line of the plot with a line of hyphens (-), regardless of whether the symbols are reference symbols or plotting

symbols. However, if a line contains only reference symbols, PROC TIMEPLOT does not connect the symbols.

Featured in: Example 3 on page 1262

NOSYMNAME
suppresses the name of the symbol variable in column headings when you use a CLASS statement. If you use NOSYMNAME, only the value of the symbol variable appears in the column heading.

Featured in: Example 5 on page 1266

NPP
suppresses the listing of the values of the variables that appear in the PLOT statement.

Featured in: Example 3 on page 1262

OVERLAY
plots all requests in one PLOT statement on one set of axes. Otherwise, PROC TIMEPLOT produces a separate plot for each plot request.

Featured in: Example 4 on page 1264

OVPCHAR='*character*'
specifies the character to print if multiple plotting symbols coincide. If a plotting symbol and a character in a reference line coincide, PROC TIMEPLOT prints the plotting symbol.

Default: at sign (@)

Featured in: Example 5 on page 1266

POS=*print-positions-for-plot*
specifies the number of print positions to use for the horizontal axis.

Default: If you omit both POS= and AXIS=, PROC TIMEPLOT initially assumes that POS=20. However, if space permits, this value increases so that the plot fills the available space.

Interaction: If you specify POS=0 and AXIS=, the plot fills the available space. POS= overrides an interval set with AXIS= (see the discussion of AXIS= on page 1254).

See also: "Page Layout" on page 1257

Featured in: Example 1 on page 1258

REF=*reference-value(s)*
draws lines on the plot that are perpendicular to the specified values on the horizontal axis. The values for *reference-value(s)* may be constants, or you may use the form

MEAN(*variable(s)*)

If you use this form of REF=, PROC TIMEPLOT evaluates the mean for each variable that you list and draws a reference line for each mean.

Interaction: If you use the UNIFORM option in the PROC TIMEPLOT statement, the procedure calculates the mean values for the variables over all observations for all BY groups. If you do not use UNIFORM, the procedure calculates the mean for each variable for each BY group.

Interaction: If a plotting symbol and a reference character coincide, PROC TIMEPLOT prints the plotting symbol.

Featured in: Example 3 on page 1262 and Example 4 on page 1264

REFCHAR='*character*'
specifies the character for drawing reference lines.

Default: vertical bar (|)

Interaction: If you are using the JOINREF or HILOC option, do not specify a value for REFCHAR= that is the same as a plotting symbol because PROC TIMEPLOT will interpret the plotting symbols as reference characters and will not connect the symbols as you expect.

Featured in: Example 3 on page 1262

REVERSE
orders the values on the horizontal axis with the largest value in the leftmost position.

Featured in: Example 4 on page 1264

Results

Data Considerations

The input data set usually contains a date variable to use as either a class or an ID variable. Although PROC TIMEPLOT does not require an input data set sorted by date, the output is usually more meaningful if the observations are in chronological order. In addition, if you use a CLASS statement, the output is more meaningful if the input data set groups observations according to combinations of class variable values. (For more information see "CLASS Statement" on page 1251.)

Procedure Output

Page Layout

For each plot request, PROC TIMEPLOT prints a listing and a plot. PROC TIMEPLOT determines the arrangement of the page as follows:

- If you use POS=, the procedure
  - determines the size of the plot from the POS= value
  - determines the space for the listing from the width of the columns of printed values, equally spaced and with a maximum of five positions between columns
  - centers the output on the page.
- If you omit POS=, the procedure
  - determines the width of the plot from the value of the AXIS= option
  - expands the listing to fill the rest of the page.

If there is not enough room to print the listing and the plot for a particular plot request, PROC TIMEPLOT produces no output and writes the following error message to the SAS log:

```
ERROR:  Too many variables/symbol values
        to print.
```

The error does not affect other plot requests.

Contents of the Listing

The listing in the output contains different information depending on whether or not you use a CLASS statement. If you do not use a CLASS statement (see Example 1 on

page 1258), PROC TIMEPLOT prints (and plots) each observation on a separate line. If you do use a CLASS statement, the form of the output varies depending on whether or not you specify a symbol variable (see "Using a Symbol Variable" on page 1252).

Missing Values

Four types of variables can appear in the listing from PROC TIMEPLOT: plot variables, ID variables, class variables, and symbol variables (as part of some column headers). Plot variables and symbol variables can also appear in the plot.

Observations with missing values of a class variable form a class of observations.

In the listing, missing values appear as a period (.), a blank, or a special missing value (the letters A through Z and the underscore (_) character).

In the plot, PROC TIMEPLOT handles different variables in different ways:

- An observation or class of observations with a missing value of the plot variable does not appear in the plot.
- If you use a symbol variable (see the discussion of plot requests on page 1254), PROC TIMEPLOT uses a period (.) as the symbol variable on the plot for all observations with a missing value of the symbol variable.

Examples

Example 1: Plotting a Single Variable

Procedure features:

ID statement

PLOT statement arguments:

simple plot request

POS=

This example

- uses a single PLOT statement to plot sales of refrigerators
- specifies the number of print positions to use for the horizontal axis of the plot
- provides context for the points in the plot by printing in the listing the values of two variables that are not in the plot.

Program

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

The data set SALES contains weekly information on the sales of refrigerators and stoves by two sales representatives.

```
data sales;
   input Month Week Seller $ Icebox Stove;
```

```
   datalines;
1 1 Kreitz   3450.94 1312.61
1 1 LeGrange 2520.04  728.13
1 2 Kreitz   3240.67  222.35
1 2 LeGrange 2675.42  184.24
1 3 Kreitz   3160.45 2263.33
1 3 LeGrange 2805.35  267.35
1 4 Kreitz   3400.24 1787.45
1 4 LeGrange 2870.61  274.51
2 1 Kreitz   3550.43 2910.37
2 1 LeGrange 2730.09  397.98
2 2 Kreitz   3385.74  819.69
2 2 LeGrange 2670.93 2242.24
;
```

The plot variable, Icebox, appears in both the listing and the output. POS= provides 50 print positions for the horizontal axis.

```
proc timeplot data=sales;
   plot icebox / pos=50;
```

The values of the ID variables, Month and Week, appear in the listing.

```
   id month week;
```

The TITLE statements specify titles for the report.

```
   title 'Weekly Sales of Iceboxes';
   title2 'for the';
   title3 'First Six Weeks of the Year';
run;
```

Output

The column headers in the listing are the variables' names. The plot uses the default plotting symbol, which is the first character of the plot variable's name.

```
                          Weekly Sales of Iceboxes                          1
                                   for the
                            First Six Weeks of the Year

Month     Week      Icebox      min                                      max
                                2520.04                                3550.43
                              *-------------------------------------------------*
    1        1     3450.94     |                                            I    |
    1        1     2520.04     |I                                                |
    1        2     3240.67     |                                      I          |
    1        2     2675.42     |         I                                       |
    1        3     3160.45     |                                I                |
    1        3     2805.35     |               I                                 |
    1        4     3400.24     |                                         I       |
    1        4     2870.61     |                  I                              |
    2        1     3550.43     |                                              I  |
    2        1     2730.09     |           I                                     |
    2        2     3385.74     |                                         I       |
    2        2     2670.93     |       I                                         |
                              *-------------------------------------------------*
```

Example 2: Customizing an Axis and a Plotting Symbol

Procedure features:

ID statement

PLOT statement arguments:

using a plotting symbol

AXIS=

Other features:

LABEL statement

PROC FORMAT

SAS system options:

FMTSEARCH=

Data set: SALES on page 1258

This example

- specifies the character to use as the plotting symbol
- specifies the minimum and maximum values for the horizontal axis as well as the interval represented by each print position
- provides context for the points in the plot by printing in the listing the values of two variables that are not in the plot
- uses a variable's label as a column header in the listing
- creates and uses a permanent format.

Program

```
libname proclib 'SAS-data-library';
```

The SAS system option FMTSEARCH= adds the SAS data library PROCLIB to the search path that is used to locate formats.

```
options nodate pageno=1 linesize=80 pagesize=60
        fmtsearch=(proclib);
```

PROC FORMAT creates a permanent format for Month. The LIBRARY= option specifies a permanent storage location so that the formats are available in subsequent SAS sessions. This format is used for examples throughout this chapter.

```
proc format library=proclib;
   value monthfmt 1='January'
                  2='February';
run;
```

The plot variable, Icebox, appears in both the listing and the output. The plotting symbol is 'R'. AXIS= sets the minimum value of the axis to 2500 and the maximum value to 3600. BY 25 specifies that each print position on the axis represents 25 units (in this case, dollars).

```
proc timeplot data=sales;
   plot icebox='R' / axis=2500 to 3600 by 25;
```

The values of the ID variables, Month and Week, appear in the listing.

```
   id month week;
```

The LABEL statement associates a label with the variable Icebox for the duration of the PROC TIMEPLOT step. PROC TIMEPLOT uses the label as the column header in the listing.

```
   label icebox='Refrigerator';
```

The FORMAT statement assigns a format to use for Month in the report. The TITLE statements specify titles.

```
   format month monthfmt.;
   title 'Weekly Sales of Refrigerators';
   title2 'for the';
   title3 'First Six Weeks of the Year';
run;
```

Output

The column headers in the listing are the variables' names (for Month and Week, which have no labels) and the variable's label (for Icebox, which has a label). The plotting symbol is **R** (for Refrigerator).

```
                        Weekly Sales of Refrigerators                          1
                                    for the
                                First Six Weeks of the Year

   Month    Week    Refrigerator    min                                      max
                                    2500                                    3600
                                   *----------------------------------------*
January        1         3450.94   |                                     R  |
January        1         2520.04   | R                                      |
January        2         3240.67   |                              R         |
January        2         2675.42   |       R                                |
January        3         3160.45   |                          R             |
January        3         2805.35   |            R                           |
January        4         3400.24   |                                 R      |
January        4         2870.61   |               R                        |
February       1         3550.43   |                                    R   |
February       1         2730.09   |         R                              |
February       2         3385.74   |                                 R      |
February       2         2670.93   |       R                                |
                                   *----------------------------------------*
```

Example 3: Using a Variable for a Plotting Symbol

Procedure features:

ID statement

PLOT statement arguments:

using a variable as the plotting symbol
JOINREF
NPP
REF=
REFCHAR=

Data set: SALES on page 1258

Formats: MONTHFMT. on page 1261

This example

- specifies a variable to use as the plotting symbol to distinguish between points for each of two sales representatives
- suppresses the printing of the values of the plot variable in the listing
- draws a reference line to a specified value on the axis and specifies the character to use to draw the line
- connects the leftmost and rightmost symbols on each line of the plot.

Program

```
libname proclib 'SAS-data-library';
```

The SAS system option FMTSEARCH= adds the SAS data library PROCLIB to the search path that is used to locate formats.

```
options nodate pageno=1 linesize=80 pagesize=60
        fmtsearch=(proclib);
```

The PLOT statement specifies both the plotting variable, Stove, and a symbol variable, Seller. The plotting symbol is the first letter of the formatted value of the Seller (in this case, **L** or **K**).

```
proc timeplot data=sales;
   plot stove=seller /
```

NPP suppresses the appearance of the plotting variable, Stove, in the listing.

```
                        npp
```

REF= and REFCHAR= draw a line of colons at the sales target of $1500.

```
                        ref=1500 refchar=':'
```

JOINREF connects the leftmost and rightmost symbols on each line of the plot.

```
                        joinref
```

AXIS= sets the minimum value of the horizontal axis to 100 and the maximum value to 3000. BY 50 specifies that each print position on the axis represents 50 units (in this case, dollars).

```
                        axis=100 to 3000 by 50;
```

The ID statement writes the values of the ID variables, Month and Week, in the listing.

```
   id month week;
```

The FORMAT statement assigns a format to use for Month in the report. The TITLE statements specify titles.

```
   format month monthfmt.;
   title 'Weekly Sales of Stoves';
   title2 'Compared to Target Sales of $1500';
   title3 'K for Kreitz; L for LaGrange';
run;
```

Output

The plot uses the first letter of the value of Seller as the plotting symbol.

```
                          Weekly Sales of Stoves                              1
                      Compared to Target Sales of $1500
                         K for Kreitz; L for LaGrange

   Month    Week     min                                                   max
                     100                                                  3000
                    *-------------------------------------------------------*
January         1    |                          K---:                        |
January         1    |              L--------------:                         |
January         2    |  K-------------------------:                         |
January         2    |  L-------------------------:                         |
January         3    |                            :--------------K           |
January         3    |   L------------------------:                         |
January         4    |                            :-----K                    |
January         4    |   L------------------------:                         |
February        1    |                            :----------------------------K  |
February        1    |      L---------------------:                         |
February        2    |               K-------------:                         |
February        2    |                            :--------------L           |
                    *-------------------------------------------------------*
```

Example 4: Superimposing Two Plots

Procedure features:

PROC TIMEPLOT statement options:

MAXDEC=

PLOT statement arguments:

using two types of plot requests
OVERLAY
REF=MEAN(*variable(s)*)
REVERSE

Data set: SALES on page 1258

This example

- superimposes two plots on one set of axes
- specifies a variable to use as the plotting symbol for one plot and a character to use as the plotting symbol for the other plot
- draws a reference line to the mean value of each of the two variables plotted
- reverses the labeling of the axis so that the largest value is at the far left of the plot.

Program

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

MAXDEC= specifies the number of decimal places to display in the listing.

```
proc timeplot data=sales maxdec=0;
```

The PLOT statement requests two plots. One plot uses the first letter of the formatted value of Seller to plot the values of Stove. The other uses the letter **R** (to match the label Refrigerators) to plot the value of Icebox.

```
    plot stove=seller icebox='R' /
```

OVERLAY places the two plots on the same set of axes.

```
                                    overlay
```

REF= draws two reference lines: one perpendicular to the mean of Stove, the other perpendicular to the mean of Icebox.

```
                                    ref=mean(stove icebox)
```

REVERSE orders the values on the horizontal axis from largest to smallest.

```
                                    reverse;
```

The LABEL statement associates a label with the variable Icebox for the duration of the PROC TIMEPLOT step. PROC TIMEPLOT uses the label as the column header in the listing. The TITLE statements specify titles.

```
    label icebox='Refrigerators';
    title 'Weekly Sales of Stoves and Refrigerators';
    title2 'for the';
    title3 'First Six Weeks of the Year';
run;
```

Output

The column header for the variable Icebox in the listing is the variable's label (Refrigerators). One plot uses the first letter of the value of Seller as the plotting symbol. The other plot uses the letter **R**.

```
                    Weekly Sales of Stoves and Refrigerators                  1
                                   for the
                            First Six Weeks of the Year

 Stove     Refrigerators      max                                        min
                               3550.43                                184.24
                              *-----------------------------------------------*
  1313              3451     |R       |                          K  |         |
   728              2520     |        |       R                     |     L   |
   222              3241     |    R   |                             |       K |
   184              2675     |        |    R                        |        L|
  2263              3160     |     R  |          K                  |         |
   267              2805     |        |  R                          |       L |
  1787              3400     | R      |                  K          |         |
   275              2871     |        | R                           |       L |
  2910              3550     |R       | K                           |         |
   398              2730     |        |   R                         |      L  |
   820              3386     | R      |                             |   K     |
  2242              2671     |        |    R     L                  |         |
                              *-----------------------------------------------*
```

Example 5: Showing Multiple Observations on One Line of a Plot

Procedure features:

CLASS statement

PLOT statement arguments:

creating multiple plots
NOSYMNAME
OVPCHAR=

Data set: SALES on page 1258

Formats: MONTHFMT. on page 1261

This example

- groups observations for the same month and week so that sales for the two sales representatives for the same week appear on the same line of the plot
- specifies a variable to use as the plotting symbol
- suppresses the name of the plotting variable from one plot
- specifies a size for the plots so that they both occupy the same amount of space.

Program

The SAS system option FMTSEARCH= adds the SAS data library PROCLIB to the search path that is used to locate formats.

```
libname proclib 'SAS-data-library';


options nodate pageno=1 linesize=80 pagesize=60
        fmtsearch=(proclib);
```

The CLASS statement groups all observations with the same values of Month and Week into one line in the output. Using the CLASS statement with a symbol variable produces in the listing one column of the plot variable for each value of the symbol variable.

```
proc timeplot data=sales;
   class month week;
```

Each PLOT statement produces a separate plot. The plotting symbol is the first character of the formatted value of the symbol variable: **K** for Kreitz; **L** for LaGrange. POS= specifies that each plot uses 25 print positions for the horizontal axis. OVPCHAR= designates the exclamation point as the plotting symbol when the plotting symbols coincide. NOSYMNAME suppresses the name of the symbol variable Seller from the second listing.

```
   plot stove=seller  / pos=25 ovpchar='!';
   plot icebox=seller / pos=25 ovpchar='!' nosymname;
```

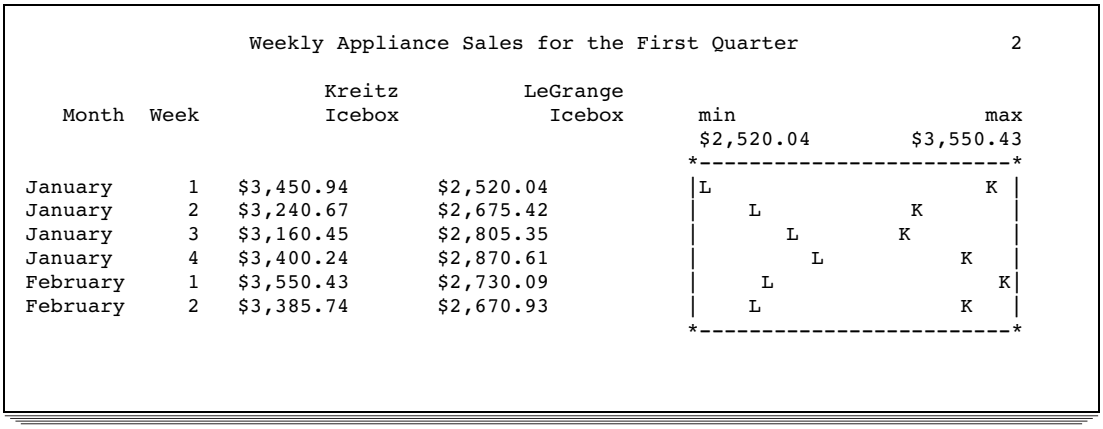
The FORMAT statement assigns formats to use for Stove, Icebox, and Month in the report. The TITLE statement specifies a title.

```
   format stove icebox dollar10.2 month monthfmt.;
   title 'Weekly Appliance Sales for the First Quarter';
run;
```

Output

```
                Weekly Appliance Sales for the First Quarter                1

              Seller :Kreitz  Seller :LeGrange
   Month  Week           Stove             Stove      min                   max
                                                     $184.24          $2,910.37
                                                    *-----------------------*
January       1   $1,312.61         $728.13          |     L    K            |
January       2     $222.35         $184.24          |!                      |
January       3   $2,263.33         $267.35          | L                K    |
January       4   $1,787.45         $274.51          | L             K       |
February      1   $2,910.37         $397.98          |  L                   K|
February      2     $819.69       $2,242.24          |      K            L   |
                                                    *-----------------------*
```



```
                Weekly Appliance Sales for the First Quarter                2

                        Kreitz          LeGrange
   Month   Week          Icebox            Icebox      min                    max
                                                       $2,520.04        $3,550.43
                                                      *-------------------------*
January        1    $3,450.94       $2,520.04         |L                       K |
January        2    $3,240.67       $2,675.42         |    L             K      |
January        3    $3,160.45       $2,805.35         |       L        K       |
January        4    $3,400.24       $2,870.61         |         L           K   |
February       1    $3,550.43       $2,730.09         |     L                  K|
February       2    $3,385.74       $2,670.93         |    L                K   |
                                                      *-------------------------*
```

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS® Procedures Guide, Version 8,* Cary, NC: SAS Institute Inc., 1999. 1729 pp.

SAS® Procedures Guide, Version 8

ISBN 1–58025–482–9

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st printing, October 1999