SAS/GRAPH® 9.2
Graph Template Language Reference
Second Edition

SAS® Documentation

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. . 2010. . *SAS/GRAPH® 9.2: Graph Template Language Reference, Second Edition*. Cary, NC: SAS Institute Inc.

SAS/GRAPH® 9.2: Graph Template Language Reference, Second Edition

ISBN 978-1-60764-675-4

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

1st electronic book, June 2010
2nd electronic book, July 2010
1st printing, June 2010

Contents

PART 6 Legend Statements 489

PART 7 Text Statements 507

PART 8 Runtime Programming Features 541

PART 9 Appendices 557

What’s New in SAS/GRAPH 9.2 Graph Template Language

Overview

The third maintenance release of the SAS/GRAPH Graph Template Language provides the DISCRETEOFFSET= option, which enables you to offset the positions of bars, plot markers, and other data representations on a discrete axis.

Support for Offsets on Discrete Axes

The new DISCRETEOFFSET= option enables you to specify an offset for the data that are represented on a discrete axis. The offset is useful for graphing multiple response variables on a common axis in the graph.

For example, if multiple BARCHART statements with different response variables are specified within an overlay-type layout, the bars for matching category values are, by default, centered on the midpoint axis values. Thus, the bars from the multiple bar charts are superimposed on one another. By setting a different offset value on each BARCHART statement, you can display the bars for common response values side by side.

Similarly, within an overlay-type layout, the plot markers for a scatter plot are, by default, centered on the discrete X values, or discrete Y values, or both. If a SCATTERPLOT statement and other plots are used together on a discrete axis, the scatter markers might be superimposed over other graph data. By setting a different offset on each plot statement, you can offset the data on the common axis. This prevents the scatter markers from being superimposed over the other graph data.

The DISCRETEOFFSET= option is available on the following plot statements:

- BARCHART statement
- BARCHARTPARM statement
- BOXPLOT statement
- BOXPLOTPARM statement
- DROPLINE statement
- REFERENCELINE statement
- SCATTERPLOT statement
- SERIESPLOT statement
- STEPPLOT statement

Part 1

Fundamentals

Chapter 1
Overview

Graph Template Language (GTL)

GTL and the Output Delivery System (ODS)

The SAS/GRAPH *Graph Template Language* (GTL) is an extension to the Output Delivery System (ODS) that enables you to create sophisticated analytical graphics that are not available from traditional SAS/GRAPH procedure statements. For example, using the GTL you can generate Model-Fit plots, Distribution Plots, Comparative plots, Prediction Plots, and more.

The graphics produced by the GTL are generated by template definitions that control the graph format and appearance and specify the variable roles to represent in the graph display. The graphs can then be rendered by associating the templates with a data source.

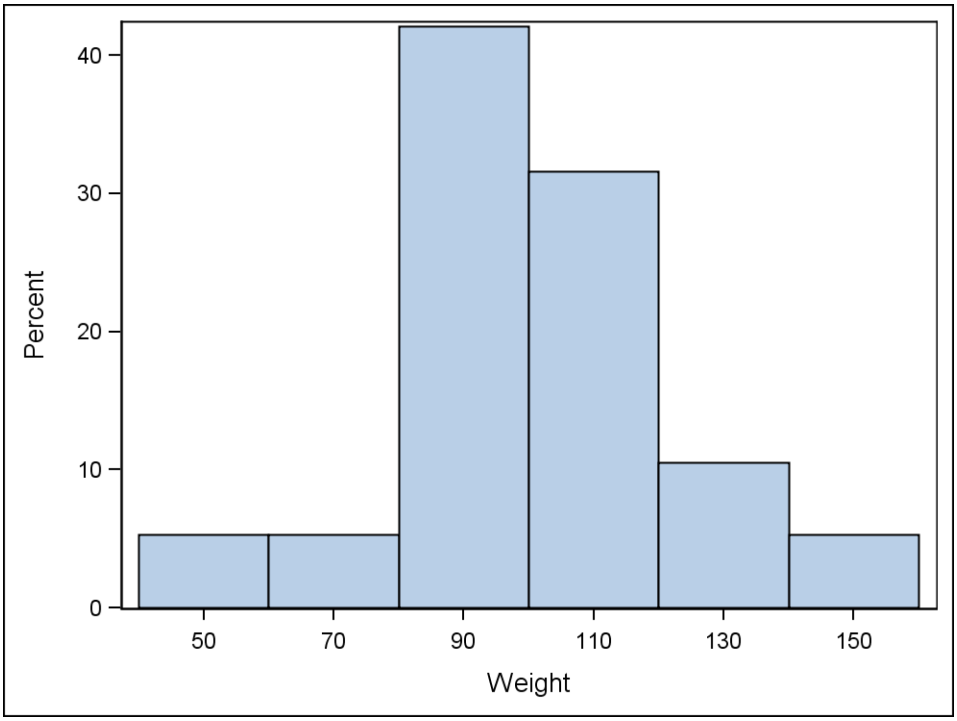
- The GTL templates are defined with PROC TEMPLATE. The GTL includes conditional statements that can be used to determine what graph features are rendered, layout statements that specify the arrangement of graph features, plot statements that request specific plot types (such as histograms and scatter plots), and text and legend statements that specify titles, footnotes, legends, and other text-based graph elements.

- The GTL templates are rendered using the SGRENDER procedure, which specifies a data source that contains appropriate data values and the template to use for rendering the graph.
- You can also modify predefined GTL templates that the SAS System delivers for use on the SAS statistical procedures. For information about modifying existing templates, refer to *SAS/STAT User's Guide*.

This manual provides a complete reference to the Graphics Template Language. For detailed usage information, consult the *SAS/GRAPH: Graph Template Language User's Guide*.

A Quick Example

The data set SASHELP.CLASS is delivered with the SAS System. It includes data columns named HEIGHT and WEIGHT, which store height and weight measures for a small sample of subjects. The Graphics Template Language can be used to generate a histogram that shows the distribution of weight recorded in that data set:



The following SAS program produces the graph:

```
proc template;
  define statgraph histogram;
    begingraph;
      layout overlay;
        histogram weight;
      endlayout;
    endgraph;
  end;
run;
```

```
ods graphics / width=450px;
ods listing;
proc sgrender data=sashelp.class
  template=histogram;
run;
```

- The DEFINE STATGRAPH statement on PROC TEMPLATE opens a definition block for defining a graphics template named HISTOGRAM. The HISTOGRAM template is stored in the template folder (also called the "template store," by default located in SASUSER.TEMPLAT).
- The template definition for HISTOGRAM specifies two GTL statements within a BEGINGRAPH/ENDGRAPH block: LAYOUT OVERLAY and HISTOGRAM.
- The LAYOUT OVERLAY statement is one of the most fundamental layout statements. It can overlay the results of one or more plot statements, each of which shares the same plot area, axes, and legends. The layout in this example specifies only a single element: a HISTOGRAM with bars showing the distribution of observations of the data column named WEIGHT.
- The ENDLAYOUT statement ends the layout block, the ENDGRAPH statement ends the graph definition, and the END statement ends the template definition.
- The ODS GRAPHICS statement uses the WIDTH= option to set a width for the output graph. Because the HEIGHT= option is not specified, GTL manages the graph's aspect ratio and set an appropriate height.
- The ODS LISTING statement opens the LISTING destination for the output. The LISTING destination is open by default. However, explicitly specifying it ensures that it is active for this graph in case another ODS destination has been set in the current SAS session.
- The DATA= option on PROC SGRENDER specifies SASHELP.CLASS as the data source for the graph. TEMPLATE= specifies HISTOGRAM as the template definition to use for rendering the graph.

Template Compilation

A GTL template describes the structure and appearance of a graph to be produced, similar to the way a TABLE template describes the organization and content of a table.

All templates are stored, compiled programs. The following source program produces a simple GTL template named SCATTER:

```
proc template;
  define statgraph scatter;
    begingraph;
      layout overlay;
        scatterplot x=height y=weight;
      endlayout;
    endgraph;
 end;
run;
```

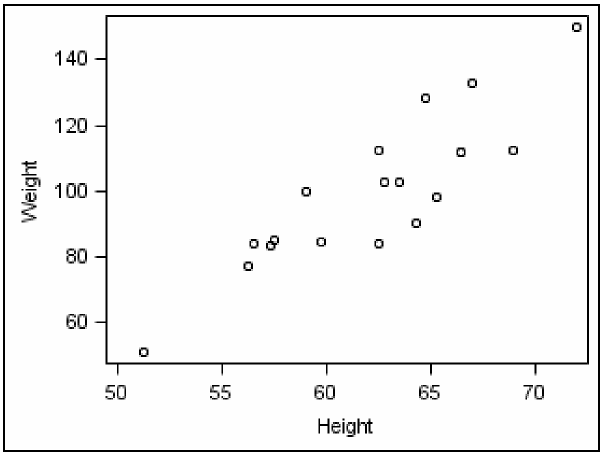
When this code is submitted, the statement keywords and options are parsed, just as with any other procedure. If no syntax error is detected, an output template named SCATTER is created and stored in the default template folder SASUSER.TEMPLAT. No graph is produced. Note the following:

- Any required arguments in the template must be specified. In this example, X= and Y= in the SCATTERPLOT statement must specify variables for the analysis, but no checking for the existence of these variables is done at compile time. (Unlike other SAS procedures, PROC TEMPLATE does not perform a compile and then run sequence, which includes variable validation.)
- No reference to an input data set appears in the template.

Run Time Actions

To produce a graph, a GTL template must be bound to a data source using the SGRENDER procedure. The following example uses SGRENDER to bind the SCATTER template to the SAS data set SASHELP.CLASS, which is delivered with the SAS system:

```
ods listing;
proc sgrender data=sashelp.class
              template=scatter;
run;
```



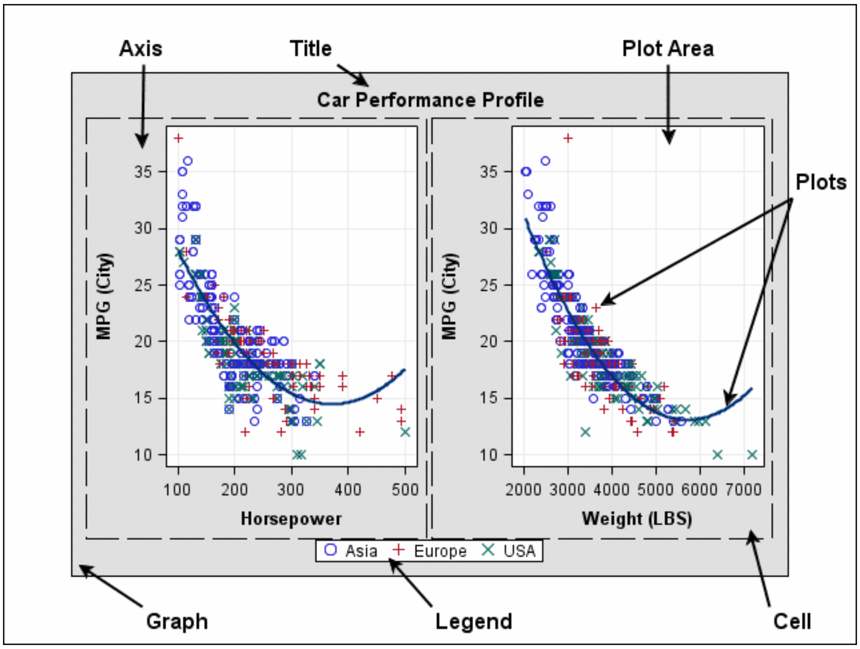
Generally, an ODS data object is constructed by comparing the template references to column names with variables that exist in the current data set. In the current example, SASHELP.CLASS contains variables named HEIGHT and WEIGHT. Because these variable names match the variables that are named on template SCATTER, variables HEIGHT and WEIGHT are added to the data object, while other variables in SASHELP.CLASS are ignored. (It is possible for a template to define new computed columns based on existing columns.)

After all the observations have been read, the data object and template definition are passed to a graph renderer, which produces an image file for the graph. The image file is then automatically integrated into the ODS destination. In this example, a PNG image is created in the LISTING destination. The visual properties of the graph are determined by the ODS style that is in effect.

Note: Template SCATTER is a restrictive definition: it can create a plot only with variables named HEIGHT and WEIGHT. A GTL template can be made more flexible by introducing dynamics or macro variables that supply variables and other information at run time. For more information, see "Flexible Templates" on page 12.

Basic Anatomy of an ODS Graph

The GTL is flexible and able to produce many different types of graphs with varying layout features. The following figure shows the basic anatomy of an ODS graph:



Graph
: the output produced from all of the statements that are nested in a BEGINGRAPH statement block. The graph comprises all of the graphics elements in the template definition.

Title Area
: area for one or more titles. This area is always displayed above all cells in the graph.

Footnote Area
: area for one or more footnotes. This area is always displayed below all cells in the graph.

Cell
: refers collectively to the area containing the plot areas. In this diagram, there are two cells, each of which contains two axes for the plot area. A cell can also contain descriptive text and legends. Graphs are often described as single-cell or multi-cell.

Plot Area
: the display area for plot-statement results. This area is bounded by the axes (when present) and can also contain data labels and other text that annotates the graph.

Axis
: refers collectively to the axis line, the major and minor tick marks, the major tick values, and the axis label.

Plots
: refers collectively to all plot statements that can be overlaid in the plot area. This includes graphical items such as fit lines, scatter plots, reference lines, and many others.

Legend
: refers collectively to one or more legend entries, each made up of a graphical value and a text label. The legend can also have a title and border. Legends can also display a color ramp corresponding to a continuous response range.

Graphical Layouts

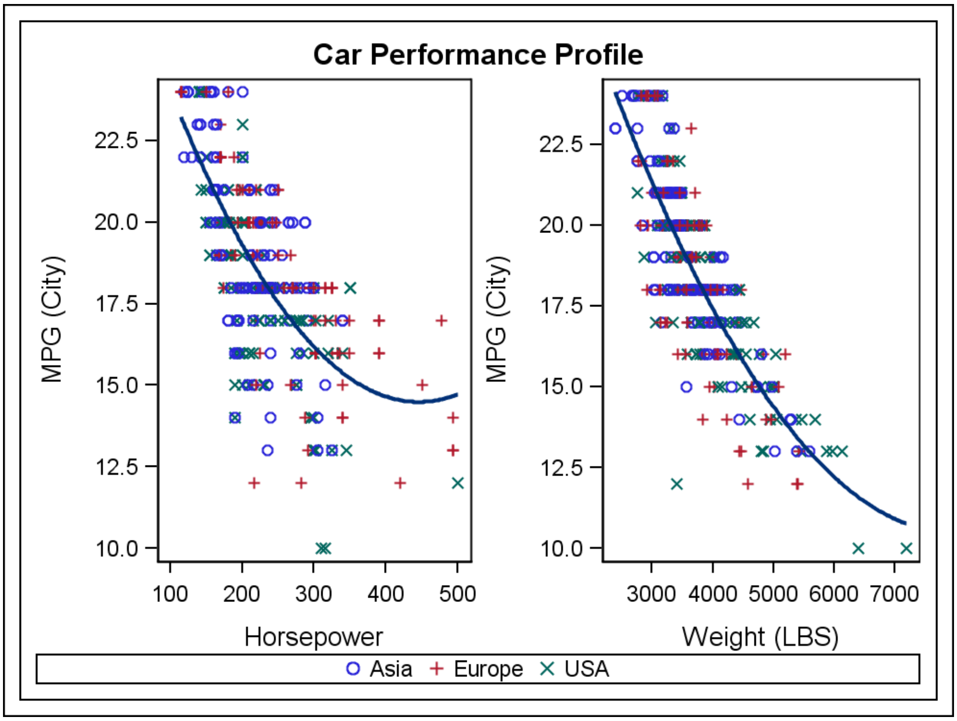
One of most powerful features of the GTL is the syntax built around hierarchical statement blocks called “layouts.” The outermost layout block determines

- The overall organization of the graph—whether it uses a single-cell or a multi-cell display.
- What statements are allowed in the block. Generally, layout blocks can contain plots, lines of text, a legend, or even another layout.
- How the contained statements interact.

Table 1.1 Outermost Layouts in GTL

Layout	Description
OVERLAY	General purpose layout for displaying 2-D plots in a single-cell.
OVERLAY3D	Layout for displaying 3-D plots in a single-cell.
OVERLAYEQUATED	Specialized OVERLAY with equated axes.
GRIDDED	Basic grid of plots. All cells are independent.
LATTICE	Advanced multi-cell layout. Axes can be shared across columns or rows and be external to grid. Many grid labeling and alignment features.
DATALATTICE	Generates a classification panel from the values of 1 or 2 classifiers.
DATAPANEL	Generates a classification panel from the values of n classifiers.

For example, the following graph is a two-cell graph produced using the LAYOUT LATTICE statement as the outermost template in the layout.



The LAYOUT LATTICE statement is typically used to create a multi-cell layout of plots that are aligned across columns and rows. In the following template, which produced the graph, plot statements are specified within nested LAYOUT OVERLAY statements. Thus, the LATTICE automatically aligns the plot areas and tick display areas in the plots. The LATTICE layout is a good layout to choose when you want to compare the results of related plots.

```
proc template;
  define statgraph lattice;
  begingraph;
    entrytitle "Car Performance Profile";
    layout lattice / border=true pad=10 opaque=true
                     rows=1 columns=2 columngutter=3;
      layout overlay;
        scatterplot x=horsepower y=mpg_city /
          group=origin name="cars";
        regressionPlot x=horsepower y=mpg_city / degree=2;
      endlayout;

      layout overlay;
        scatterplot x=weight y=mpg_city / group=origin;
        regressionPlot x=weight y=mpg_city / degree=2;
      endlayout;

      sidebar;
        discretelegend "cars";
      endsidebar;
    endlayout;
  endgraph;
  end;
run;
```

For detailed information about each layout, see the chapter for that layout type.

Plots

The plots in the GTL are classified in different ways, depending on the context of the discussion.

Within layout blocks, plots are often classified according to graphical dimension: whether they are projected in two or three visual dimensions. Thus, plots in the GTL are often referred to as 2-D or 3-D plots, based on their graphical dimensions, not their data dimensions.

Relative to their input data, plots are classified according to the statements that calculate summary statistics from raw input data, and those that use calculated statistics as input parameters on the plot statement. Thus, many GTL plot statements have two versions: BARCHART and BARCHARTPARM, HISTOGRAM and HISTOGRAMPARM, and so forth. The main distinction between such plots is the nature of the input data that they accept:

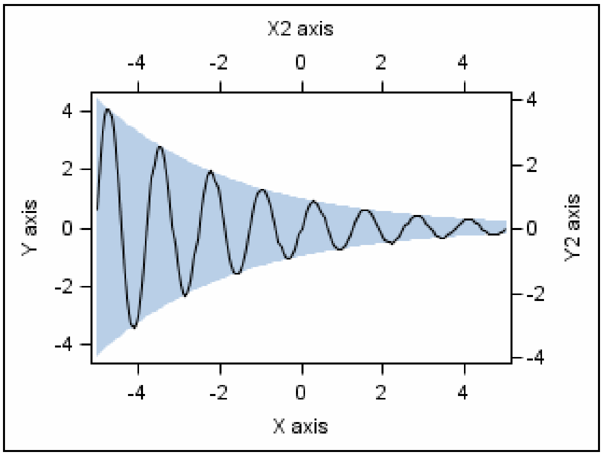
- The "non-parm" version (for example, BARCHART) computes its values from raw, unsummarized data. For example, a BARCHART computes the summary values it needs for the bars in the chart. Such plots are often referred to as "computed plots."
- The "parm" version (for example, BARCHARTPARM) does not summarize or compute values from the input data but instead simply renders the input data it is given. Thus, the input data must be pre-summarized, perhaps by a SAS procedure. The "parm" version of plots, often referred to as "parameterized plots," produce the same result as the non-parm version. However, they don't perform the calculations or data summarizations needed to achieve the result.

Chapter 12, "Key Concepts for Using Plots," on page 115 discusses general concepts that apply across plot types. For detailed information about a particular plot, see the chapter for that plot.

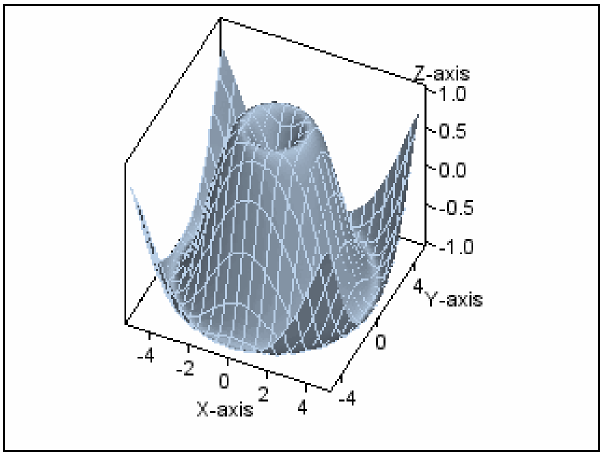
Axes

The GTL uses various criteria to determine the displayed axis features for a graph. Generally, axis features are based on the layout type, the order of plot statements in the layout and the options specified on those statements, the use of "primary" and "secondary" axes on the plots (when secondary axes are supported), the plot type, the column(s) of data that contribute to defining the axis range, and the data formats for the contributing data columns.

Depending on the layout type, 2-D plots can have up to four independent axes that can be displayed: X, Y, X2, and Y2. The X and Y axes are considered the primary axes, and the X2 and Y2 axes are considered the secondary axes. By default, the X2 and Y2 axes are not displayed. When requested, the secondary axes can be displayed as copies of the primary axes, or data can be mapped separately to them. The following figure identifies the X, Y, X2, and Y2 axes.



All 3-D plots display the standard X, Y, and Z axes.



For more information about axis features in GTL, see Chapter 41, “Axis Features in Layouts,” on page 395.

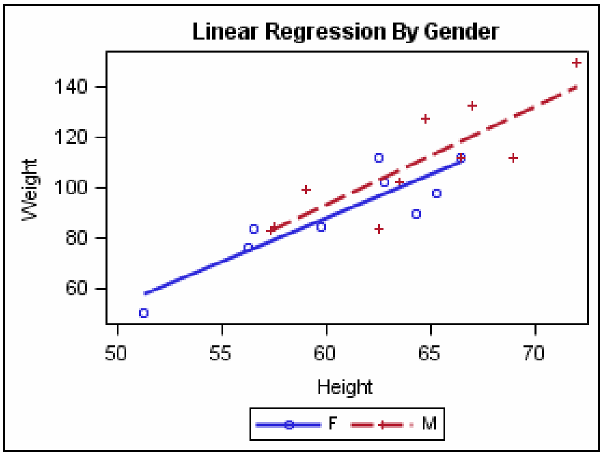
Legends

Many plot statements support a GROUP= option that partitions the data into unique values, performs separate analysis, if necessary, and automatically assigns distinct visual properties to each group value. The visual properties of group values are defined by the style in effect.

Legends are not automatically displayed for plots with group values. Rather, an appropriate legend statement must be added to the template to generate the desired legend. In the following example, a legend is added to display markers and line patterns that show the association between the group values from a scatter plot and corresponding linear regression lines. The example shows the mechanism that GTL uses to associate a legend with its corresponding plot(s): a name is assigned to each plot that must be represented in the legend, and these names are then used as arguments for the legend statement (in this case, DISCRETELEGEND).

```
proc template;
 define statgraph scatterfit;
  begingraph;
```

```
        entrytitle "Linear Regression By Gender";
         layout overlay;
           scatterplot x=height y=weight / group=sex name="scat";
           regressionplot x=height y=weight/ group=sex name="reg";
           discretelegend "scat" "reg" / border=true;
         endlayout;
        endgraph;
      end;
    run;
```



For more information about managing legends in GTL, see *SAS/GRAPH: Graph Template Language User's Guide*.

Flexible Templates

Several features in the GTL can make template definitions less restrictive on input data and more general in nature. These features enable a single compiled template to produce many output variations.

Expressions and Functions

In the GTL, expressions can be used to compute constants and data columns. The expressions must be enclosed in an EVAL construct. Within the expression you can use DATA step functions, arithmetic operators, and other special functions supported by the GTL.

Expressions are also useful in text statements like ENTRY and ENTRYTITLE. Both of these statements support rich text and have special text commands such as {SUP}, {SUB}, and {UNICODE}, which enable subscripting, superscripting, and Unicode characters.

The following template shows how the ± symbol is included in the title line using its hexadecimal Unicode value. Also, new data columns are computed for the upper and lower error bars of the scatter plot, based on the input columns MEANWEIGHT and STDERR.

```
    proc template;
      define statgraph expression;
        begingraph;
        entrytitle "Errorbars show " {unicode "00B1"x} "2 SE";
         layout overlay;
          scatterplot x=age y=meanweight /
        yerrorlower=eval(meanweight - 2*stderr)
        yerrorupper=eval(meanweight + 2*stderr);
     seriesplot x=age y=meanweight;
    endlayout;
   endgraph;
  end;
run;
```

For more information about using expressions, see Chapter 54, “Expressions,” on page 545. For more information about using functions, see Chapter 55, “Functions,” on page 549.

Dynamics and Macro Variables

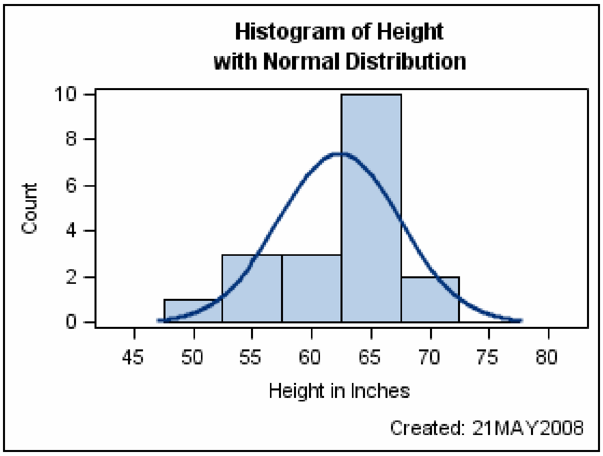
An extremely useful technique for generalizing templates is to define dynamics, macro variables, or both. The dynamics and macro variables resolve when the template is executed. The following PROC TEMPLATE statements can be used in a DEFINE STATGRAPH block:

Template Statement	**Purpose**	**Value supplied by...**
DYNAMIC	defines dynamic(s)	either of the following: • DYNAMIC= suboption of ODS= option of FILE PRINT • DYNAMIC statement of PROC SGRENDER
MVAR	defines macro variable(s)	%LET or CALL SYMPUT( )
NMVAR	defines macro variable(s) that resolves to a number(s)	%LET or CALL SYMPUT( )
NOTES	provides information about the graph definition	user-supplied text

The following example defines a template named DYNAMICS that can create a histogram and density plot for any variable. It defines both macro variables and dynamics for run time substitution. No data-dependent information is hard coded in the template.

Note: You can initialize macro variables with %LET statements and dynamics with SGRENDER’s DYNAMIC statement.

```
proc template;
 define statgraph dynamics;
   mvar SYSDATE9 SCALE;
   nmvar BINS;
   dynamic VAR VARLABEL;
   begingraph;
     entrytitle "Histogram of " VAR;
     entrytitle "with Normal Distribution";
     layout overlay / xaxisopts=(label=VARLABEL);
       histogram VAR /  scale=SCALE  nbins=BINS;
       densityplot VAR / normal();
     endlayout;
     entryfootnote halign=right "Created: " SYSDATE9 /
       textattrs=GraphValueText;
   endgraph;
 end;
run;

%let bins=6;
%let scale=count;
proc sgrender data=sashelp.class
              template=dynamics;
  dynamic var="Height" varlabel="Height in Inches";
run;
```



For more information about using dynamics and macro variables, see Chapter 53, “Dynamics and Macro Variables,” on page 543.

Conditional Logic

Using conditional logic, you can create templates that have multiple visual results or output representations, depending on existing conditions. The evaluation of a logical expression must generate one or more complete statements (not portions of statements). All conditional logic uses one of the following constructs:

if (condition)
  statement(s);
endif;
	if (condition)
	  statement(s);
	else
	  statement(s);
	endif;

On the IF statement, *condition* must be enclosed in parentheses. The *condition* can be any standard SAS expression involving arithmetic, logical operators, comparison operators, Boolean operators, or concatenation operators. The expression can also use SAS DATA step functions. The expression resolves to a single numeric value, which is *true* or *false*.

In the following example, a histogram is conditionally overlaid with a normal distribution curve, a Kernel Density Estimate distribution curve, both, or neither:

```
proc template;
  define statgraph conditional;
   dynamic VAR VARLABEL BINS CURVE;
```

```
      begingraph;
       entrytitle "Histogram of " VAR;
       layout overlay / xaxisopts=(label=VARLABEL);
        histogram VAR /  nbins=BINS;

        if (upcase(CURVE) in ("ALL"  "KERNEL"))
           densityplot VAR / kernel() name="k"
                             legendlabel="Kernel"
                             lineattrs=(pattern=dash);
        endif;

        if (upcase(CURVE) in ("ALL" "NORMAL"))
           densityplot VAR / normal() name="n"
                             legendlabel="Normal";
        endif;

        discretelegend "n" "k";
      endlayout;
     endgraph;
    end;
   run;
```

Note that the legend syntax does not have to be made conditional. At run time, each plot name in the legend is checked. If the plot does not exist, its name is removed from the legend name list. If no names appear in the DISCRETELEGEND statement, the legend “drops out” and the histogram is resized to fill the remaining space.

For more information about using conditional logic, see Chapter 56, “Conditional Logic,” on page 553.

Output

When using the GTL, you focus primarily on defining template definitions that produce specific graphs and generate a particular output layout. Ultimately, you must also tailor the graphical environment to get the exact output that you desire. The ODS GRAPHICS statement is available for tailoring the graphical environment, and ODS styles enable you to manage the output appearance.

ODS GRAPHICS Statement

The ODS GRAPHICS statement is used to modify the environment in which graphics templates are executed. The ODS GRAPHICS statement is used to control

- whether ODS graphics is enabled
- the type and name of the image created
- the size of the image
- whether features such as scaling and anti-aliasing are used.

The following ODS GRAPHICS statement uses the HEIGHT= and WIDTH= options to set an aspect ratio for the output image.

```
ods graphics on / height=175px width=200px;
 proc sgrender data=sashelp.class
               template=scatter;
 run;
ods graphics off;
```

For more information about using the ODS GRAPHICS statement in GTL, see *SAS/GRAPH: Graph Template Language User's Guide*. For a more complete discussion of the ODS GRAPHICS statement, see *SAS Output Delivery System: User's Guide*.

ODS Styles

When any graphics template is executed, there is always an ODS style in effect that governs the appearance of the output. The following ODS statement sends graphics output to the RTF output destination using the LISTING style:

```
ods rtf style=listing;

ods graphics on / height=175px width=200px border=off;
 proc sgrender data=sashelp.class
              template=scatter;
   run;
ods graphics off;

ods rtf close;
```

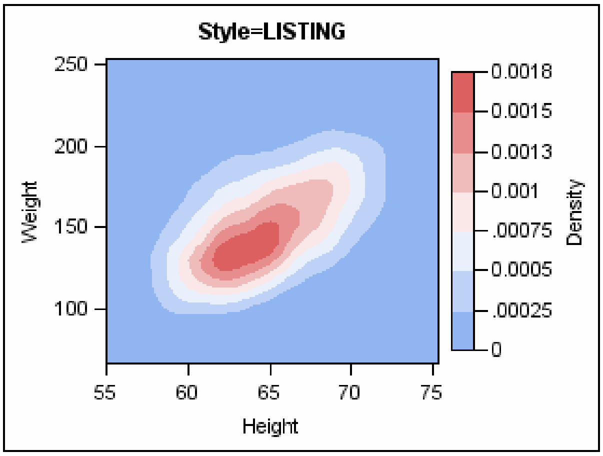
Support for ODS styles is highly integrated into GTL syntax. By default, the graphical appearance features of most plot and text statements are mapped to corresponding style elements and associated attributes. Because of this, your output tables and graphs always have a reasonable overall appearance. Moreover, output for a given ODS destination has a consistent look (for example, table colors and graph colors don't clash).

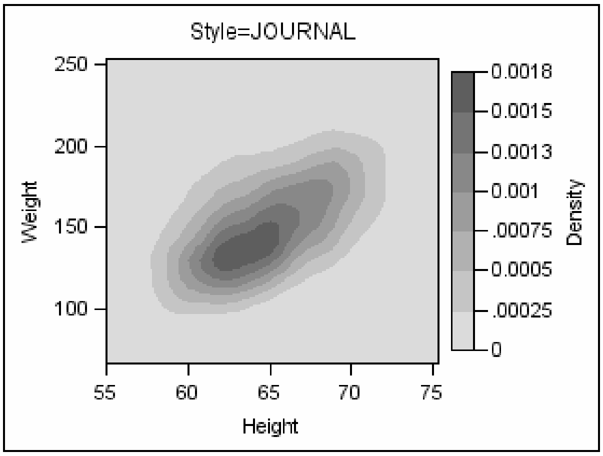
The following figures show how a graph's appearance can be changed by using references to style elements to set the graph's appearance options. This technique permits changes in graph appearance by style modification instead of graphical template modification. The graphs in the figures are generated with the following GTL statement:

```
contourplotparm x=x y=y z=density /
  contourtype=fill nhint=9
  colormodel=ThreeColorRamp ;
```

The following style template shows the definition for the ThreeColorRamp style element:

```
style ThreeColorRamp /
  endcolor     = GraphColors("gramp3cend")
  neutralcolor = GraphColors("gramp3cneutral")
  startcolor   = GraphColors("gramp3cstart");
```





For more information about the use of ODS styles in GTL, see *SAS/GRAPH: Graph Template Language User's Guide*. For a more complete discussion of ODS styles, see *SAS Output Delivery System: User's Guide*. The *SAS/STAT User's Guide* also has a detailed discussion for using styles with ODS Graphics.

About the Examples in this Documentation

The programs in this documentation often provide all of the code that you need to generate the graphs that are shown in the figures. We encourage you to copy and paste the code into your SAS session and generate the graphs for yourself. The graphs that you generate in the LISTING destination are rendered in their default 640 pixel by 480 pixel size. The exception is the examples that show you how to change the graph size.

The graphical output in this documentation does not show graphs in their default size because of the limitations of the production system used. The maximum graph width that can be included in this document is 495 pixels. Hence, all graphs are scaled down to fit.

When graphs that are produced with ODS graphics are reduced in size, several automatic processes take place to optimize the appearance of the output. Among the differences between default size graphs and smaller graphs are that the smaller graphs have scaled down font sizes. Also, their numeric axes might display a reduced number of ticks and tick values. Thus, the graphs that you generate from the example programs will not always look identical to the graphs that are shown in the figures. However, both graphs will accurately represent the data.

When producing your graphical output, you can scale the graph size and also modify font attributes if needed. The *SAS/GRAPH: Graph Template Language User's Guide* provides chapters that explain how to set fonts, DPI, anti-aliasing, and other features that contribute to producing professional-looking graphics of any size in any output format.

Part 2

Graph Block

Chapter 2
BEGINGRAPH Statement

Description and Syntax

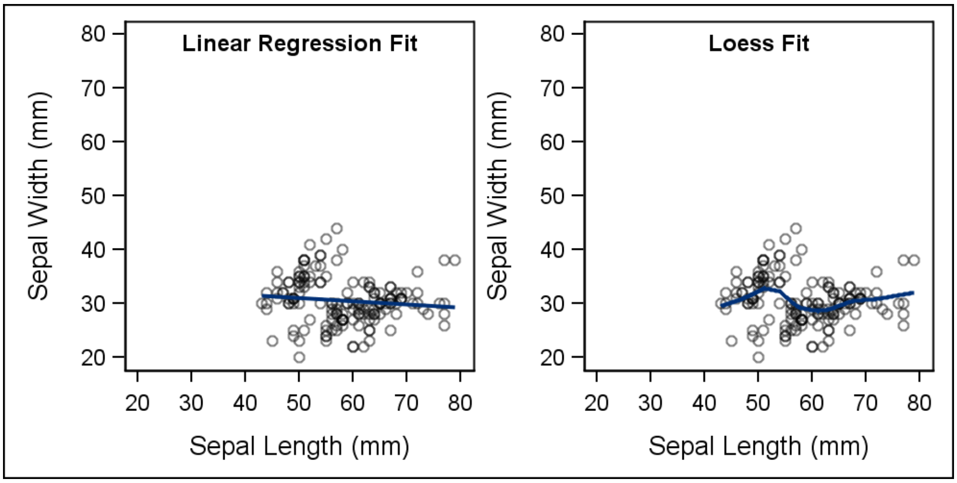
Defines the outermost container for a single *GTL-layout-block* and one or more *GTL-global-statements*.

BEGINGRAPH *</option(s)>*;
  <GTL-global-statements>
  GTL-layout-block
  <GTL-global-statements>
ENDGRAPH;

Example Program and Statement Details

Example Graph

The following graph was generated by the "Example Program" on page 22:



Example Program

The BEGINGRAPH statement block is a required outermost container for any graph template. One if its purposes is to support options that control the graph size. When you know that the default graph a template produces is better viewed with a different size or aspect ratio, you can declare default sizing information. To do so, use the DESIGNWIDTH= option, or the DESIGNHEIGHT= option, or both. This program shows one way to set the width and height of two graph cells to be equal.

```
proc template;
  define statgraph begingraph;
    dynamic XVAR YVAR;
    begingraph / designwidth=640px designheight=320px;
    layout lattice / columns=2;
      layout overlayequated / equatetype=square;
        entry "Linear Regression Fit" /
          valign=top texttattrs=(weight=bold);
        scatterplot x=XVAR y=YVAR / datatransparency=.5;
        regressionplot x=XVAR y=YVAR;
      endlayout;
      layout overlayequated / equatetype=square;
        entry "Loess Fit" /
          valign=top texttattrs=(weight=bold);
        scatterplot x=XVAR y=YVAR / datatransparency=.5;
        loessplot x=XVAR y=YVAR;
      endlayout;
    endlayout;
    endgraph;
  end;
run;

proc sgrender data=sashelp.iris template=begingraph;
  dynamic title="Square Plot"
    xvar="SepalLength" yvar="SepalWidth";
run;
```

Statement Summary

All template definitions in the Graphics Template Language must start with a BEGINGRAPH statement and conclude with an ENDGRAPH statement.

Within a BEGINGRAPH block, at least one *GTL-layout-block* is required. It can be a LATTICE, GRIDDED, OVERLAY, OVERLAYEQUATED, OVERLAY3D, DATALATTICE, or DATAPANEL layout block. This layout block should contain at least one plot statement.

The *GTL-global-statements* include one or more ENTRYTITLE, ENTRYFOOTNOTE, or IF / ELSEIF/ ELSE statements. Any of these global statements can precede or follow the *GTL-layout-block*.

By default, graphs are rendered at 640px by 480px (4:3 aspect ratio). To change the graph size for the output of a single BEGINGRAPH block, use the DESIGNWIDTH= and DESIGNHEIGHT= options on that BEGINGRAPH statement. For example, the template in the “Example Program” on page 22 uses DESIGNHEIGHT= to change the graph height to 320px. To prevent the graph width from automatically scaling to preserve the 4:3 aspect ratio, it uses DESIGNWIDTH= to maintain the 640px width. The setting ensures that each graph cell is square (320px by 320px).

Note: To change the graph sizes for all templates in the current SAS session, use the WIDTH= and HEIGHT= options in the ODS GRAPHICS statement. Size settings in the ODS GRAPHICS statement override size settings in the BEGINGRAPH statement and remain in effect unless they are changed on another ODS GRAPHICS statement.

The following template defines a square graph (equal height and width, 1:1 aspect ratio) by setting the design width equal to the internal default height (480px). The setting is made with DESIGNWIDTH=DEFAULTDESIGNHEIGHT:

```
proc template;
  define statgraph squareplot;
    dynamic title xvar yvar;
    begingraph / designwidth=defaultDesignHeight;
      entrytitle title;
      layout overlayequated / equatetype=square;
        scatterplot x=xvar y=yvar;
        regressionplot x=xvar y=yvar;
      endlayout;
    endgraph;
  end;
run;
```

If this template were executed with the following GRENDER procedure statement, a 480px by 480px graph would be created:

```
proc sgrender data=mydata template="squareplot" ;
dynamic title="Square Plot" xvar="time1" yvar="time2";
run;
```

If the ODS GRAPHICS statement’s WIDTH= or HEIGHT= options change the design width or design height, the *squareplot* template’s 1:1 aspect ratio would still be honored. Thus, both of the following GRENDER procedure statements would create a 550px by 550px graph:

```
ods graphics / width=550px;
proc sgrender data=mydata template="squareplot" ;
```

```
dynamic title="Square Plot" xvar="time1" yvar="time2";
run;

ods graphics / height=550px;
proc sgrender data=mydata template="squareplot" ;
dynamic title="Square Plot" xvar="time1" yvar="time2";
run;
```

Options

Statement Option	Description
BACKGROUNDCOLOR	Specifies the color of the graph background.
BORDER	Specifies whether a border is drawn around the graph.
BORDERATTRS	Specifies the properties of the border line around the graph.
DESIGNHEIGHT	Specifies the design height of the graph.
DESIGNWIDTH	Specifies the design width of the graph.
PAD	Specifies the amount of extra space that is added inside the graph border.

BACKGROUNDCOLOR=*style-reference* | *color*
specifies the color of the graph background.

Default: The GraphBackground:Color style reference.

style-reference
A reference of the form *style-element:style-attribute*. Only the style attribute named COLOR is used.

BORDER=*boolean*
specifies whether a border is drawn around the graph.

Default: TRUE

Interaction: If this option is set to FALSE, the BORDERATTRS= option is ignored.

BORDERATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)
specifies the attributes of the border line around the graph. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Line Options" on page 564 for available *line-options*.

Default: The GraphBorderLines style element.

Interaction: BORDER=TRUE must be set for this option to have any effect.

DESIGNHEIGHT=DEFAULTDESIGNHEIGHT | *dimension*
specifies the design height of the graph.

Default: DEFAULTDESIGNHEIGHT. This value is obtained from the SAS Registry key **Products** ⇨ **Graph** ⇨ **ODS** ⇨ **StatGraph** ⇨ **DefaultDesignHeight** when the graph is rendered. The initial value of this registry key is 640px.

Interaction: The design height can be overridden at run time with the HEIGHT= option of the ODS GRAPHICS statement (external to the template).

The value of the DefaultDesignHeight registry key can be changed. Doing so would affect the design height of all templates that do not include an explicit dimension for the design height.

DESIGNWIDTH=DEFAULTDESIGNWIDTH | *dimension*

specifies the design width of the graph.

Default: DEFAULTDESIGNWIDTH. This value is obtained from the SAS Registry key **Products** ⇨ **Graph** ⇨ **ODS** ⇨ **StatGraph** ⇨ **DefaultDesignWidth** when the graph is rendered. The initial value of this registry key is 480px.

Interaction: The design width can be overridden at run time with the WIDTH= option of the ODS GRAPHICS statement (external to the template).

The value of the DefaultDesignWidth registry key can be changed. Doing so would affect the design width of all templates that do not include an explicit dimension for the design width.

PAD=*dimension* | *(pad-options)*

specifies the amount of extra space that is added inside the graph border.

Default: The default padding for all sides is 10. Values without units are in pixels (px). A unit must be provided if other than pixels.

dimension

Specifies a dimension to use for the extra space at the left, right, top, and bottom of the layout border.

(*pad-options*)

Enables separate settings for the left, right, top, and bottom padding dimensions. Use the *pad-options* to create non-uniform padding. These options must be enclosed in parentheses. Each option is specified as a *name* = *value* pair. Sides not assigned padding are padded with the default amount.

Values without units are in pixels (px). A unit must be provided if other than pixels.

LEFT=*dimension*	specifies the amount of extra space added to the left side.
RIGHT=*dimension*	specifies the amount of extra space added to the right side.
TOP=*dimension*	specifies the amount of extra space added to the top.
BOTTOM=*dimension*	specifies the amount of extra space added to the bottom.

Part 3

Layout Statements

Chapter 3

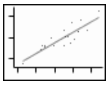
Summary of Layout Statements

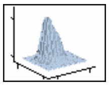
Overview

Layout blocks always begin with the LAYOUT keyword followed by a keyword indicating the purpose of the layout. All layout blocks end with an ENDLAYOUT statement.

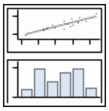
The following sections summarize the available layouts. To learn more about a layout, see the chapter devoted to that layout.

Overlay-type Layouts

Layout (Description)	Graphics Allowed and Cells Produced	Comments	Example
OVERLAY (Single Cell)	2-D (1 cell)	General purpose layout for superimposing 2-D plots	
OVERLAYEQUATED (Single Cell)	2-D (1 cell)	Specialized OVERLAY with equated axes	
PROTOTYPE (Single Cell)	2-D (1 cell)	Specialized LAYOUT used only as child layout of DATAPANEL or DATALATTICE	

Layout (Description)	Graphics Allowed and Cells Produced	Comments	Example
OVERLAY3D (Single Cell)	3-D (1 cell)	General purpose 3-D layout for superimposing 3-D plots.	

Multi-cell Layouts

Layout (Description)	Graphics Allowed and Cells Produced	Comments	Example
LATTICE (Advanced Multi-cell)	2-D (1 or more cells)	All cells must be predefined. Axes can be shared across columns or rows and be external to grid. Many grid labeling and alignment features.	
GRIDDED (Simple Multi-cell)	2-D (1 or more cells)	All cells must be predefined. Axes independent for each cell. Very simple multi-cell container.	

Data-driven Layouts

Layout (Description)	Graphics Allowed and Cells Produced	Comments	Example
DATAPANEL (Classification Panel)	2-D (1 or more cells)	Displays a panel of similar graphs based on data subsets by classification variable(s). Number of cells is based on crossings of *n* classification variable(s).	
DATALATTICE (Classification Panel)	2-D (1 or more cells)	Displays a panel of similar graphs based on data subsets by classification variable(s). Number of cells is based on crossings of 1 or 2 classification variables.	

Chapter 4

LAYOUT DATALATTICE Statement

Description and Syntax

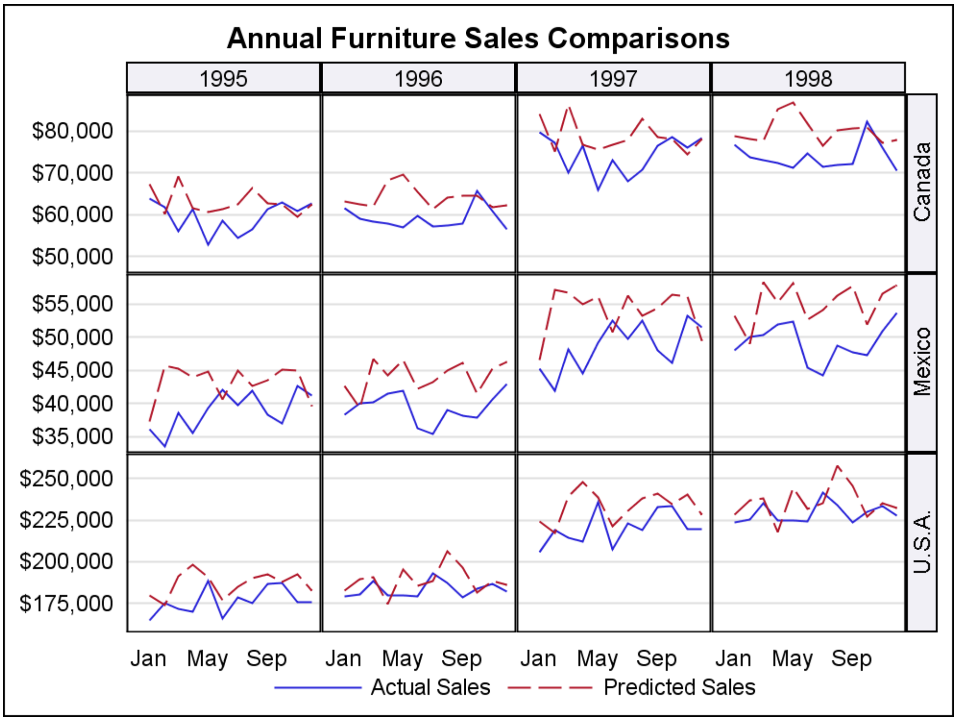
Creates a grid of graphs based on one or two classification variables and a graphical prototype. By default, a separate instance of the prototype (a data cell) is created for each possible combination of the classification variables.

LAYOUT DATALATTICE argument(s) <*/option(s)*>;
graph-prototype-block;
<*sidebar-statement-block(s)*;>
ENDLAYOUT;

Example Program and Statement Details

Example Graph

The following graph was generated by the “Example Program” on page 34:



Example Program

This example shows the result of using row and column classification variables. In this case, a four-column, three-row data lattice is created:

- The classification values are placed as row or column labels by default.
- The ROWDATARANGE = UNION option assures that an axis range is computed separately for each row using the data ranges of all Y= variables in that row. This facilitates the visual comparison of the data cells.
- A SIDEBAR block is used to place the legend at the bottom of the lattice.

```
proc template;
 define statgraph layoutdatalattice;
  begingraph;
    entrytitle "Annual Furniture Sales Comparisons";
    layout datalattice rowvar=country columnvar=year /
      rowdatarange=union
      headerlabeldisplay=value
      headerbackgroundcolor=GraphAltBlock:color
      rowaxisopts=(display=(tickvalues) griddisplay=on
        linearopts=(tickvalueformat=dollar12.))
      columnaxisopts=(display=(tickvalues)
        timeopts=(tickvalueformat=monname3.));
      layout prototype / cycleattrs=true;
        seriesplot x=month y=TotalActual / name="Actual";
        seriesplot x=month y=TotalPredict / name="Predict";
      endlayout;
      sidebar / align=bottom;
        discretelegend "Actual" "Predict" / border=false;
      endsidebar;
    endlayout;
    endgraph;
  end;
run;

proc summary data=sashelp.prdsal2 nway;
    class country year month;
    var actual predict;
    output out=prdsal2 sum=TotalActual TotalPredict;
run;

proc sgrender data=prdsal2 template=layoutdatalattice;
run;
```

Statement Summary

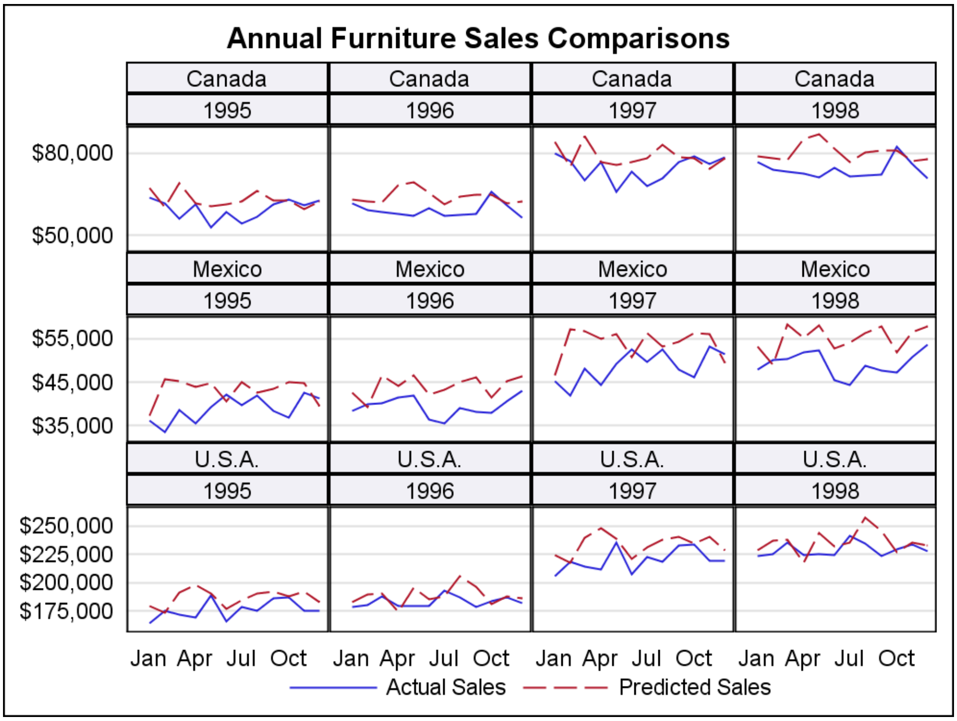
The LAYOUT DATALATTICE statement makes it easy to create a grid of graphs, based on the values of one or two classifications variables. To create a grid that is based on more than two classification variables, or to have more control over the grid layout, use LAYOUT DATAPANEL instead.

By default, the number of cells in the layout is determined by the number of value pairings that are possible for the classification values plus any empty cells needed to complete the last row/column of the grid. The contents of each data cell are based on a graph prototype that you specify in the *graph-prototype-block*. You can enhance the display using one or more *sidebar-statement-blocks*. For classification variables that have many values, you can use the COLUMNS= and or ROWS= options and the PANELNUMBER= option to generate multiple panel displays.

Classification variables for the layout are specified on the ROWVAR= argument (to specify a row variable), or the COLUMNVAR= argument (to specify a column variable), or both arguments to specify both a column and a row variable. The graph prototype for each data cell's contents is specified within a "Prototype Block" on page 37 block, and sidebars are specified within SIDEBAR blocks. The LAYOUT PROTOTYPE and SIDEBAR blocks are nested within the LAYOUT DATALATTICE block.

By default, the first data cell to be filled is in the layout's top left corner. Use the START= option to change the starting data cell to the bottom left corner.

Rather than display the header labels outside the grid, you can set HEADERLABELLOCATION= INSIDE to display them inside the grid, as shown in the following figure:



Note: The DATALATTICE layout is designed to be the outermost layout in the template.

Required Argument(s)

One of the ROWVAR= or COLUMNVAR= arguments is required. Both can be specified. Each specifies a single classification variable. Any one of the following uses is valid:

LAYOUT DATALATTICE ROWVAR=*class-var*

LAYOUT DATALATTICE COLUMNVAR=*class-var*

LAYOUT DATALATTICE ROWVAR=*class-var* COLUMNVAR=*class-var*

If you do not explicitly manage columns and rows using the COLUMNS= and ROWS= options, the default layout behavior is as follows:

- If both ROWVAR= and COLUMNVAR= are specified, a data cell is created for each of the value pairings that are possible for the classification values of the specified variables. If the ROWVAR variable has R distinct values and the COLUMNVAR variable has C distinct values, the dimension of grid produced is R x C.
- If only the ROWVAR variable is used, an R x 1 grid is produced.
- If only the COLUMNVAR variable is used, a 1 x C grid is produced.

Variable values are always retained in data order. Formats can be assigned to class variables to create classification levels (for example, an AGEGROUPFMT. format for numeric AGE).

The output size does not grow automatically as the number of cells increases. To set a panel size for the current template, use the DESIGNHEIGHT= and DESIGNWIDTH= options in the BEGINGRAPH statement. To set a panel size for all templates in the current SAS session, use the HEIGHT= and WIDTH= options in the ODS GRAPHICS statement. Size settings in the ODS GRAPHICS statement override size settings in the BEGINGRAPH statement. The default output width is 640px, and the default output height is 480px.

As the number of cells in the grid increases, the size of each cell decreases. At some point the cells might become so small that a meaningful graph cannot be rendered. The CELLHEIGHTMIN= and CELLWIDTHMIN= options set a threshold for the smallest cell. If the actual cell height or width becomes smaller, no panel is drawn. The default minimum cell size is CELLHEIGHTMIN=100px and CELLWIDTHMIN=100px.

Using the default panel size and cell size, the DATALATTICE layout accommodates a grid of about 24 cells (6 columns by 4 rows). If you know that the number of cells is larger, you should increase the overall panel size, or decrease the minimum cell size, or both. You can also use ROWS=, COLUMNS=, and PANELNUMBER= options to partition your data so a number of smaller grids are produced that cumulatively show all of the value crossings.

Prototype Block

You must specify a single *graph-prototype-block* within the LAYOUT DATALATTICE block, using the following syntax:

LAYOUT PROTOTYPE </*option(s)*>;

  GTL-statement(s);

ENDLAYOUT;

The *graph-prototype-block* determines the graphical content of each data cell and is repeated within each data cell, based on the subsets of the classification variables.

For more information about the LAYOUT PROTOTYPE block and the list of available options, see "LAYOUT PROTOTYPE Statement" on page 109.

Sidebar Blocks

A LAYOUT DATALATTICE enables you to display one or more *sidebars* outside of the axes. A sidebar spans across columns or rows and is useful for displaying information that applies to all of the columns or all of the rows. For example, sidebars are useful for displaying a legend.

A SIDEBAR statement has the following syntax:

SIDEBAR </ ALIGN=BOTTOM | TOP | LEFT | RIGHT>;

  GTL-statement(s);

ENDSIDEBAR;

- ALIGN=BOTTOM is the default alignment.
- You can specify up to four SIDEBAR blocks in a LAYOUT DATALATTICE, one for each of the bottom, top, left, and right sidebar positions.
- The LAYOUT DATALATTICE automatically aligns a sidebar with the layout columns or rows.
- Only one statement (such as ENTRY or DISCRETELEGEND) or one layout block (such as LAYOUT GRIDDED) is allowed in a SIDEBAR block. To create multi-line text in a sidebar, nest ENTRY statements within a LAYOUT GRIDDED block.

The following example shows a SIDEBAR block that displays a legend at the top of the layout grid.

```
sidebar / align=top;
   discretelegend 'p' 'a' / across=2;
endsidebar;
```

Options

Statement Option	Description
BACKGROUNDCOLOR	Specifies the color of the layout background.
BORDER	Specifies whether a border is drawn around the layout.
BORDERATTRS	Specifies the properties of the border line.
CELLHEIGHTMIN	Specifies the minimum height of a cell in the grid.
CELLWIDTHMIN	Specifies the minimum width of a cell in the grid.
COLUMNAXISOPTS	Specifies axis options for all columns.
COLUMNDATARANGE	Specifies how the X-axes of instances of the *graph-prototype* are scaled.
COLUMNGUTTER	Specifies the amount of empty space that is between the columns.
COLUMNHEADERS	Specifies where to position the outside column header.
COLUMNS	Specifies the number of columns in the layout.
HEADERBACKGROUNDCOLOR	Specifies the background color of the cell headers.
HEADERLABELATTRS	Specifies the text properties for the cell headers.
HEADERLABELDISPLAY	Indicates whether the automatic cell header text includes variable name and value or just the value.
HEADERLABELLOCATION	Indicates whether the cell header is placed within each cell (INSIDE) or as row and column headers external to the lattice (OUTSIDE).
HEADEROPAQUE	Specifies whether the background for the cell headers is opaque.
INSET	Specifies what information is displayed in an inset.
INSETOPTS	Specifies the location and appearance options for the inset information.
OPAQUE	Specifies whether the layout background is opaque.

Statement Option	Description
PAD	Specifies the amount of extra space that is added inside the layout border.
PANELNUMBER	Specifies the number of the panel to produce.
ROWAXISOPTS	Specifies axis options for all rows.
ROWDATARANGE	Specifies how the Y-axes of instances of the *graph-prototype* are scaled.
ROWGUTTER	Specifies amount of empty space between the rows.
ROWHEADERS	Specifies where to position the outside row header.
ROWS	Specifies the number of rows in the layout.
SKIPEMPTYCELLS	Specifies whether the external axes skip the empty cells in a partially filled grid.
START	Indicates whether to start populating the grid from the top left or bottom left corner.

BACKGROUNDCOLOR=*style-reference* | *color*
specifies the color of the layout background.

Default: The GraphBackground:Color style reference.

style-element
A reference in the form *style-element:style-attribute*. Only the style-attribute named COLOR is used.

Interaction: OPAQUE=TRUE must be in effect for the color to be seen. By default, OPAQUE=FALSE.

BORDER=*boolean*
specifies whether a border is drawn around the layout.

Default: FALSE

Interaction: If this option is set to FALSE, the BORDERATTRS option is ignored.

BORDERATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)
specifies the attributes of the border line around the layout. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Line Options" on page 564 for available *line-options*.

Default: The GraphBorderLines style element.

Interaction: BORDER=TRUE must be set for this option to have any effect.

CELLHEIGHTMIN=*dimension*
specifies the minimum height of a cell in the grid.

Default: 100px

Use this option in conjunction with the CELLWIDTHMIN= option to set the minimum cell size.

The overall size of the panel is constrained by the HEIGHT= and WIDTH= options in the ODS GRAPHICS statement. As the number of cells in the grid increases, the size of each cell decreases. At some point the cell becomes so small that a meaningful graph cannot be rendered. This option sets the minimum height threshold for all cells. If the actual cell height becomes smaller, no panel is drawn.

CELLWIDTHMIN=*dimension*

specifies the minimum width of a cell in the grid.

Default: 100px

Use this option in conjunction with the CELLHEIGHTMIN= option to set the minimum cell size.

The overall size of the panel is constrained by the HEIGHT= and WIDTH= options in the ODS GRAPHICS statement. As the number of cells in the grid increases, the size of each cell decreases. At some point the cell becomes so small that a meaningful graph cannot be rendered. This option sets the minimum width threshold for all cells. If the actual cell width becomes smaller, no panel is drawn.

COLUMNAXISOPTS=(*axis-options*)

specifies X-axis options for all columns. For a list of options, see "Axis Options for LAYOUT DATALATTICE/DATAPANEL".

Requirement: Axis options must be enclosed in parentheses and separated by spaces.

COLUMNDATARANGE=UNIONALL | UNION

specifies how the X-axes of instances of the *graph-prototype* are scaled.

Default: UNIONALL

UNIONALL

scales the X-axis data ranges across all layout columns and panels (when PANELNUMBER= is in effect).

UNION

scales the X-axis data ranges separately for each column on a per-panel basis. The scaling does not span across multiple panels.

Use the COLUMNAXISOPTS= option to control shared axis features.

Use the PANELNUMBER= option for a discussion of how to create multiple panels.

COLUMNGUTTER=*dimension*

specifies the amount of empty space that is between the columns.

Default: 0

If there are *n* columns, then there are *n*-1 gutters.

COLUMNHEADERS=TOP | BOTTOM | BOTH

specifies where to position the outside column header.

Default: TOP

TOP

specifies that column header text appears at the top of the layout.

BOTTOM

specifies that column header text appears at the bottom of the layout.

BOTH
specifies that column header text alternates between the top and bottom of the layout column by column.

Interaction: HEADERLABELLOCATION=OUTSIDE must be set for this option to have any effect.

COLUMNS=*integer*
specifies the number of columns in the layout.

Default:

- If this option is not specified, the number of columns is dynamically adjusted to equal the number of classifier values for the COLUMVAR= variable.
- If this option is specified, that many columns are created. If the number of COLUMNVAR classifier values is greater than the specified number of columns, no graph is created for some classifier values. If the number of classifier values is smaller than the specified number of columns, extra empty columns are created.

Interaction: The overall grid size is constrained by the HEIGHT= and WIDTH= options in the ODS GRAPHICS statement. As the grid size grows, the cell size shrinks. To control the minimum size of a cell use the CELLHEIGHTMIN= and CELLWIDTHMIN= options.

The START= option affects the how the columns are populated.

The PANELNUMBER= option enables you to create multiple smaller grids that completely partition the classifier values.

HEADERBACKGROUNDCOLOR=*style-reference* | *color*
specifies the background color of the cell headers.

Default: The GraphHeaderBackground:Color style reference.

style-reference
A reference in the form *style-element:style-attribute*. Only the style-attribute named COLOR is used.

Interaction: HEADEROPAQUE=TRUE must be in effect for the color to be seen.

HEADERLABELATTRS=*style-element* | *style-element* (*text-options*) | (*text-options*)
specifies the color and font attributes of the data labels. See “General Syntax for Attribute Options” on page 563 for the syntax on using a *style-element* and “Text Options” on page 566 for available *text-options*.

Default: The GraphValueText style element.

HEADERLABELDISPLAY=NAMEVALUE | VALUE
indicates whether the automatic cell header text includes variable name and value or just the value.

Default: NAMEVALUE

Example: If the classification variables were Country and Product, HEADERLABEL=NAMEVALUE would produce cell headers such as

```
Country=CANADA
Product=TABLE
```

HEADERLABEL=VALUE would produce

```
CANADA
TABLE
```

HEADERLABELLOCATION=OUTSIDE | INSIDE

indicates whether the cell header is placed within each cell (INSIDE) or as row and column headers external to the lattice (OUTSIDE).

Default: OUTSIDE

HEADEROPAQUE=*boolean*

specifies whether the background for cell headers is opaque (TRUE) or transparent (FALSE).

Default: TRUE

Interaction: When this option is set to FALSE, the background color for cell headers is not used.

INSET=(*variable-list*)

specifies what information is displayed in an inset. The *variable-list* defines one or more variables whose names and values appear as a small table in the data cells. The variables can be either numeric or character. Variable names are separated by spaces.

Default: no default

Restriction: No predefined information is available for the inset. You must create the desired inset information as part of your input data. This is most typically done as follows (see the chapter on classification panels and the chapter on insets in the **SAS/GRAPH Graph Template Language User's Guide** for complete examples):

- Create a separate data set for the inset columns making sure that the column names are different from the other columns used in graph. The number observations of inset data should match the number of cells in the classification panel. The ordering of the inset observations should be the same as population order of the cells of the classification panel taking into account the ROWVAR= and COLUMNVAR= arguments and the START= option. Typically, the number of observations for the inset data is smaller than the other input data for the graph.

- Merge the inset data set with the data set for the graph using a DATA or PROC SQL step. Do not match-merge the observations of the two data sets (no BY processing). The resulting data set typically has the inset columns padded with missing values.

- Use the merged data set to produce the graph, specifying the inset column names in this option's *variable-list*.

The variable values are associated with the data cells by data order. That is, the first observation from all the variables in *variable-list* are used in the first data cell, the second observation from all variables in *variable-list* are used in the second data cell, and so on. If a value is missing for an observation, the corresponding *name-value* pair is skipped in the affected data cell.

The location and appearance of the inset is controlled by the INSETOPTS= option.

INSETOPTS=(*appearance-options*)

specifies location and appearance options for the inset information.

The *appearance-options* can be any one or more of the settings that follow. The options must be enclosed in parentheses, and each option is specified as a *name* = *value* pair.

AUTOALIGN=NONE | AUTO | (*location-list*)

specifies whether the inset is automatically aligned within the layout.

Default: NONE

NONE
Do not automatically align this inset. This inset's position is set by the HALIGN= and VALIGN= *appearance-options*.

AUTO
Attempt to center this inset in the area that is farthest from any surrounding markers. Data cells might have different inset placements.

(*location-list*)
Restrict this inset's possible locations to those locations in the specified *location-list*, and use the *location-list* position that least collides with the data cell's other graphics features. The *location-list* is blank-separated and can contain any of these locations: TOPLEFT TOP TOPRIGHT LEFT CENTER RIGHT BOTTOMLEFT BOTTOM BOTTOMRIGHT. Example: AUTOALIGN = (TOPRIGHT TOPLEFT)

Interaction: When AUTOALIGN=AUTO or (*location-list*), the HALIGN= and VALIGN= are ignored.

BACKGROUNDCOLOR= *style-reference* | *color*
specifies the color of the inset background

Default: The background is transparent. No color is assigned.

style-reference
A reference of the form *style-element : style-attribute*. Only the style-attribute named COLOR is used.

BORDER= *boolean*
specifies whether a border is displayed around the inset.

Default: FALSE

HALIGN=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the inset.

Default: LEFT

Interaction: This option is ignored unless AUTOALIGN=NONE.

OPAQUE= *boolean*
specifies whether the inset background is opaque (TRUE) or transparent (FALSE).

Default: FALSE

Interaction: When OPAQUE=FALSE, the background color is not used.

TEXTATTRS=*style-element* | *style-element* (*text-options*) | (*text-options*)
specifies the text properties of the entire inset. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Text Options" on page 566 for available *text-options*.

Default: The GraphDataText style element.

VALIGN=TOP | CENTER |BOTTOM
specifies the vertical alignment of the inset.

Default: TOP

Interaction: This option is ignored unless AUTOALIGN=NONE.

OPAQUE= *boolean*
specifies whether the layout background is opaque (TRUE) or transparent (FALSE).

Default: FALSE

Interaction: When this option is set to FALSE, the background color is not used.

PAD=*dimension | (pad-options)*
specifies the amount of extra space that is added inside the layout border.

Default: The default padding for all sides is 0. Values without units are in pixels (px). A unit must be provided if other than pixels.

dimension
Specifies a dimension to use for the extra space at the left, right, top, and bottom of the layout border.

(*pad-options*)
Enables separate settings for the left, right, top, and bottom padding dimensions. Use the *pad-options* to create non-uniform padding. These options must be enclosed in parentheses. Each option is specified as a *name = value* pair. Sides not assigned padding are padded with the default amount.

Values without units are in pixels (px). A unit must be provided if other than pixels.

LEFT=*dimension*	specifies the amount of extra space added to the left side.
RIGHT=*dimension*	specifies the amount of extra space added to the right side.
TOP=*dimension*	specifies the amount of extra space added to the top.
BOTTOM=*dimension*	specifies the amount of extra space added to the bottom.

PANELNUMBER=*positive-integer*
specifies the number of the panel to produce.

Default: 1

This option enables you to partition a large grid into a number of smaller grids under these conditions:

- You set a grid size explicitly (ROWS= and COLUMNS= when ROWVAR and COLUMNVAR variables are specified; ROWS= when only a ROWVAR variable is specified; COLUMNS= when only a COLUMNVAR variable is specified)
- The grid size is smaller in one or both of the dimensions of the default dynamically generated grid.
- You execute the template N times and increment the panel number each time. N is determined by CEIL(all rows * all columns / grid rows * grid columns).

Example: Suppose ROWVAR=R (R has 10 unique values) and COLUMNVAR=C (C has 11 unique values). The dynamic grid has 10 rows and 11 columns and you would have to make the HEIGHT=and WIDTH= quite large to enable 110 plots to be displayed. By setting some smaller grid size, say ROWS=3 and COLUMNS=4, and by making the value of PANELNUMBER= a dynamic or macro variable, you can create 10 panels (9 with 12 data cells and 1 with 2 data cells) that collectively display all 110 possible crossings. You simply invoke PROC SGRENDER or a DATA step 10 times, incrementing the dynamic value for PANELNUMBER each time.

ROWAXISOPTS=(*axis-options*)

specifies Y-axis options for all rows. For a list of options, see "Axis Options for LAYOUT DATALATTICE/DATAPANEL" on page 467.

Requirement: Axis options must be enclosed in parentheses and separated by spaces.

ROWDATARANGE=UNIONALL | UNION

specifies how the Y-axes of instances of the *graph-prototype* are scaled.

Default: UNIONALL

UNIONALL

scales the Y-axis data ranges across all layout rows and panels (when PANELNUMBER= is in effect).

UNION

scales the Y-axis data ranges separately for each row in the layout on a per-page basis. The scaling does not span across multiple pages.

Use the ROWAXISOPTS= option to control shared axis features.

See the PANELNUMBER= option for a discussion of how to create multiple panels.

ROWGUTTER=*dimension*

specifies the amount of empty space between the rows.

Default: 0

If there are *n* rows, then there are *n*-1 gutters.

ROWHEADERS=RIGHT | LEFT | BOTH

specifies where to position the outside row header.

Default: RIGHT

RIGHT

specifies that row header appears at the right of the layout.

LEFT

specifies that row header appears at the left of the layout.

BOTH

specifies that row header alternates between the right and left of the layout row by row.

HEADERLABELLOCATION=OUTSIDE must be set for this option to have any effect.

ROWS=*integer*

specifies the number of rows in the layout.

Default:

- If this option is not specified, the number of rows is dynamically adjusted to equal the number of classifier values for the ROWVAR= variable.
- If this option is specified, the specified number of rows is created. If the number of ROWVAR classifier values is greater than the specified number of rows, no graph is created for some classifier values. If the number of classifier values is smaller than the specified number of rows, extra empty rows are created.

Interaction: The overall grid size is constrained by the HEIGHT= and WIDTH= options in the ODS GRAPHICS statement. As the grid size grows, the cell size shrinks. To control the minimum size of a cell use the CELLHEIGHTMIN= and CELLWIDTHMIN= options.

The START= option affects the way the rows are populated.

The PANELNUMBER= option enables you to create multiple smaller grids that completely partition the classifier values.

SKIPEMPTYCELLS=*boolean*

specifies whether the external axes skip the empty cells in a partially filled grid.

Default: FALSE

FALSE

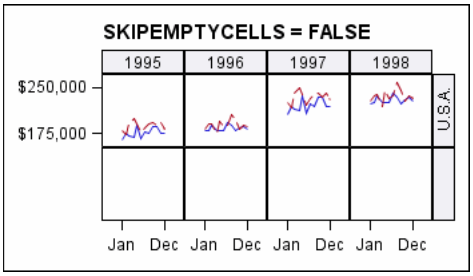
External axes are displayed at their normal locations, even if there are empty cells at one or more of the locations.

TRUE

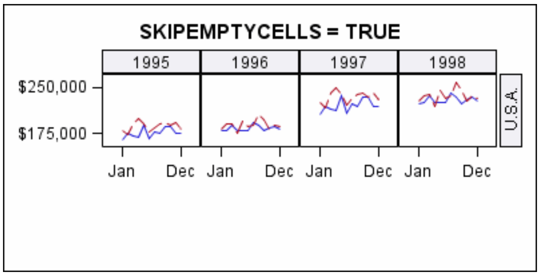
External axes skip empty cells and "snap" to the nearest data cell, both vertically and horizontally. Though the empty cells are not displayed, the data cells in the grid are not enlarged to fill the area.

Discussion: Whenever the number of unique COLUMNVAR= classifier values (data cells) is not evenly divisible by the COLUMNS= value, or the number of unique ROWVAR= classifier values (data cells) is not evenly divisible by the ROWS= value, then one or more panels is partially filled with data cells and padded with empty cells to complete the grid.

In this example, there are 4 column-data cells and 3 row-data cells arranged in a 4-column, 2-row grid. This is default appearance of the last panel:



When SKIPEMPTYCELLS=TRUE, the empty padding cells of all panels are removed and external axis ticks and tick values snap to the data cells:

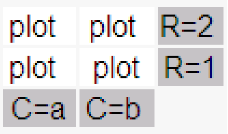


START=TOPLEFT | BOTTOMLEFT

indicates whether to start populating the grid from the top left or bottom left corner.

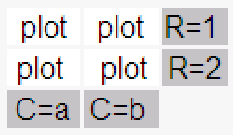
Default: TOPLEFT

If ROWVAR=R has values in the sort order 1 and 2 and COLUMNVAR=C has values in the sort order *a* and *b*, then START=BOTTOMLEFT is populated as follows:



plot	plot	R=2
plot	plot	R=1
C=a	C=b	

START=TOPLEFT is populated as follows:



plot	plot	R=1
plot	plot	R=2
C=a	C=b	

Chapter 5
LAYOUT DATAPANEL Statement

Description and Syntax

Creates a grid of graphs based on one or more classification variables and a graphical prototype. By default, a separate instance of the prototype (a data cell) is created for each actual combination of the classification variables.

LAYOUT DATAPANEL CLASSVARS = (*class-var1*…*class-varN*) </*option(s)*> ;

graph-prototype-block;

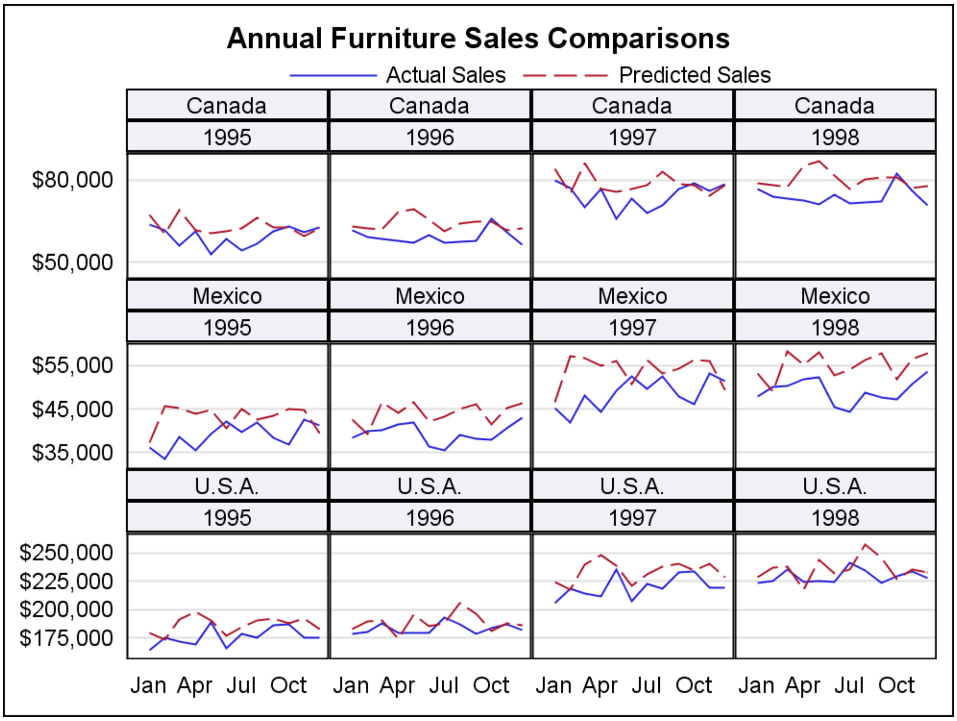
<*sidebar-statement-block(s)*;>

ENDLAYOUT;

Example Program and Statement Details

Example Graph

The following graph was generated by the “Example Program” on page 50:



Example Program

This example shows a four-column, three-row data panel using two classification variables. With this layout, each data cell is subsetted and labeled with the values of the classification variables.

- The ROWDATARANGE = UNION option assures that an axis range is computed separately for each row using the data ranges of all Y= variables in that row. This facilitates the visual comparison of the data cells.
- A SIDEBAR block is used to place the legend at the bottom of the lattice.

```
proc template;
 define statgraph layoutdatapanel;
  begingraph;
    entrytitle "Annual Furniture Sales Comparisons";
    layout datapanel classvars=(country year) /
      columns=4 rows=3 rowdatarange=union
      headerlabeldisplay=value
      headerbackgroundcolor=GraphAltBlock:color
      rowaxisopts=(display=(tickvalues) griddisplay=on
        linearopts=(tickvalueformat=dollar12.))
      columnaxisopts=(display=(tickvalues)
        timeopts=(tickvalueformat=monname3.));
      layout prototype / cycleattrs=true;
        seriesplot x=month y=TotalActual / name="Actual";
        seriesplot x=month y=TotalPredict / name="Predict";
      endlayout;
      sidebar / align=top;
        discretelegend "Actual" "Predict" / border=false;
      endsidebar;
    endlayout;
  endgraph;
 end;
run;

proc summary data=sashelp.prdsal2 nway;
  class country year month;
  var actual predict;
  output out=prdsal2 sum=TotalActual TotalPredict;
run;

proc sgrender data=prdsal2 template=layoutdatapanel;
run;
```

Statement Summary

The LAYOUT DATAPANEL statement creates a grid of graphs, based on the values of one or more classifications variables. The main differences between this layout and the DATALATTICE layout is that this layout supports more than two classification variables, and it provides more control over the grid layout.

By default, the number of cells in the layout is determined by a crosstabulation table of all the classification variables plus any empty cells needed to complete the last row/column of the grid. The contents of each data cell are based on a graph prototype that you specify in the *graph-prototype-block*. You can enhance the display using one or more *sidebar-statement-blocks*. For classification variables that have many values, you can use the COLUMNS= option or the ROWS= option, or both with the PANELNUMBER= option to generate multiple panel displays.

The order of the value pairings for the classification variables is determined by the order that the variables are specified on the CLASSVARS= argument. The last named variable's values vary most rapidly (like nested DO loops). Variable values are always returned in data order.

By default, the first data cell to be filled is in the layout's top left corner, and data cells are filled from left-to-right, top-to-bottom. Use the START= option to change the starting data cell to the bottom left corner, and use the ORDER= option to determine whether data cells fill by column or by row. See the START= option for illustrations on how START= and ORDER= interact to manage the fill sequence for data cells.

Note: The DATAPANEL layout is designed to be the outermost layout in the template.

Required Argument

CLASSVARS=(*class-var1 ...class-varN*)

specifies a list of classification variables. By default, a data cell is created for each crossing of these variables in the input data. The total number of grid cells created is the result of a crosstabulation table of all the classification variables plus any empty cells needed to complete the last row/column of the grid. You can request that data cells be generated for all possible crossings, even when the class variables have no values at those crossings. For more information, see the SPARSE= option.

Variable values are always retained in data order. Formats can be assigned to class variables to create classification levels (for example, an AGEGROUPFMT. format for numeric AGE).

The output size does not grow automatically as the number of cells increases. To set a panel size for the current template, use the DESIGNHEIGHT= and DESIGNWIDTH= options in the BEGINGRAPH statement. To set a panel size for all templates in the current SAS session, use the HEIGHT= and WIDTH= options in the ODS GRAPHICS statement. Size settings in the ODS GRAPHICS statement override size settings in the BEGINGRAPH statement. The default output width is 640px, and the default output height is 480px.

As the number of cells in the grid increases, the size of each cell decreases. At some point the cells might become so small that a meaningful graph cannot be rendered. The CELLHEIGHTMIN= and CELLWIDTHMIN= options set a threshold for the smallest cell. If the actual cell height or width becomes smaller, no panel is drawn. The default minimum cell size is CELLHEIGHTMIN=100px and CELLWIDTHMIN=100px.

Using the default panel size and cell size, the DATAPANEL layout accommodates a grid of about 24 cells (6 columns by 4 rows). If you know that the number of cells is larger, you should increase the overall panel size, or decrease the minimum cell size, or both. You can also use ROWS=, COLUMNS=, and PANELNUMBER= options to partition your data so a number of smaller grids are produced that cumulatively show all of the value crossings.

Prototype Block

You must specify a single *graph-prototype-block* within the LAYOUT DATAPANEL block, using the following syntax:

LAYOUT PROTOTYPE </*option(s)*>;
    GTL-statements;
ENDLAYOUT;

The *graph-prototype-block* determines the graphical content of each data cell and is repeated within each data cell, based on the subsets of the classification variables.

For more information about the LAYOUT PROTOTYPE block and the list of available options, see "LAYOUT PROTOTYPE Statement" on page 109.

Sidebar Blocks

A LAYOUT DATAPANEL enables you to display *sidebars* outside of the axis areas. A sidebar spans across columns or rows and is useful for displaying information that applies to all of the columns or all of the rows. For example, sidebars are useful for displaying a legend.

A SIDEBAR statement has the following syntax:

SIDEBAR </ ALIGN= BOTTOM | TOP | LEFT | RIGHT >;
    GTL-statement(s);
ENDSIDEBAR;

- ALIGN=BOTTOM is the default alignment.
- You can specify up to four SIDEBAR blocks in a LAYOUT DATAPANEL, one for each of the bottom, top, left, and right sidebar positions.
- The LAYOUT DATAPANEL automatically aligns a sidebar with the layout columns or rows.
- Only one statement (such as ENTRY or DISCRETELEGEND) or one layout block (such as LAYOUT GRIDDED) is allowed in a SIDEBAR block. To create multi-line text in a sidebar, nest ENTRY statements within a LAYOUT GRIDDED block.

The following example shows a SIDEBAR block that displays a legend at the top of the layout grid.

```
sidebar / align=top;
  discretelegend 'p' 'a' / across=2;
endsidebar;
```

Options

Statement Option	Description
BACKGROUNDCOLOR	Specifies the color of the layout background.
BORDER	Specifies whether a border is drawn around the layout.
BORDERATTRS	Specifies the properties of the border line.
CELLHEIGHTMIN	Specifies the minimum height of a cell in the grid.
CELLWIDTHMIN	Specifies the minimum width of a cell in the grid.
COLUMNAXISOPTS	Specifies axis options for all columns.
COLUMNDATARANGE	Specifies how the X-axes of instances of the *graph-prototype* are scaled.
COLUMNGUTTER	Specifies the amount of empty space that is between the columns.
COLUMNS	Specifies the number of columns in the layout.
HEADERBACKGROUNDCOLOR	Specifies the background color of the cell headers.
HEADERLABELATTRS	Specifies the text properties for the cell headers.
HEADERLABELDISPLAY	Indicates whether the automatic cell header text includes variable name and value or just the value.
HEADEROPAQUE	Specifies whether the background for the cell headers is opaque.
INSET	Specifies what information is displayed in an inset.
INSETOPTS	Specifies the location and appearance options for the inset information.
OPAQUE	Specifies whether the layout background is opaque.

Statement Option	Description
ORDER	Specifies whether cells are populated by column priority or by row priority.
PAD	Specifies the amount of extra space that is added inside the layout border.
PANELNUMBER	Specifies the number of the panel to produce.
ROWAXISOPTS	Specifies axis options for all rows.
ROWDATARANGE	Specifies how the Y-axes of instances of the *graph-prototype* are scaled.
ROWGUTTER	Specifies amount of empty space between the rows.
ROWS	Specifies the number of rows in the layout.
SKIPEMPTYCELLS	Specifies whether the external axes skip the empty cells in a partially filled grid.
SPARSE	Specifies whether crossings of the class variables include only the crossing in the data or all possible crossings.
START	Indicates whether to start populating the data cells from the top left or bottom left corner.

BACKGROUNDCOLOR=*style-reference* | *color*
specifies the color of the layout background.

Default: The GraphBackground:Color style reference.

style-reference
A reference in the form *style-element:style-attribute*. Only the style-attribute named COLOR is used.

Interaction: OPAQUE= TRUE must be in effect for the color to be seen. By default, OPAQUE=FALSE.

BORDER=*boolean*
specifies whether a border is drawn around the layout.

Default: FALSE

BORDERATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)
specifies the attributes of the border line around the layout. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Line Options" on page 564 for available *line-options*.

Default: The GraphBorderLines style element.

Interaction: BORDER=TRUE must be set for this option to have any effect.

CELLHEIGHTMIN=*dimension*
specifies the minimum height of a cell in the grid.

Default: 100px

Use this option in conjunction with the CELLWIDTHMIN= option to set the minimum cell size.

The overall size of the panel is constrained by the HEIGHT= and WIDTH= options in the ODS GRAPHICS statement. As the number of cells in the grid increases, the size of each cell decreases. At some point the cell becomes so small that a meaningful graph cannot be rendered. This option sets the minimum height threshold for all cells. If the actual cell height becomes smaller, no panel is drawn.

CELLWIDTHMIN=*dimension*

specifies the minimum width of a cell in the grid.

Default: 100px

Use this option in conjunction with the CELLHEIGHTMIN= option to set the minimum cell size.

The overall size of the panel is constrained by the HEIGHT= and WIDTH= options in the ODS GRAPHICS statement. As the number of cells in the grid increases, the size of each cell decreases. At some point the cell becomes so small that a meaningful graph cannot be rendered. This option sets the minimum width threshold for all cells. If the actual cell width becomes smaller, no panel is drawn.

COLUMNAXISOPTS=(*axis-options*)

specifies X-axis options for all columns. For a list of options, see "Axis Options for LAYOUT DATALATTICE/DATAPANEL" on page 467.

Requirement: Axis options must be enclosed in parentheses and separated by spaces.

COLUMNDATARANGE=UNIONALL | UNION

specifies how the X-axes of instances of the *graph-prototype* are scaled.

Default: UNIONALL

UNIONALL

scales the X-axis data ranges across all layout columns and panels (when PANELNUMBER= is in effect).

UNION

scales the X-axis data ranges separately for each column on a per-panel basis. The scaling does not span across multiple panels.

Use the COLUMNAXISOPTS= option to control shared axis features.

See the PANELNUMBER= option for a discussion of how to create multiple panels.

COLUMNGUTTER=*dimension*

specifies the amount of empty space that is between the columns.

Default: 0

If there are *n* columns, then there are *n*-1 gutters.

COLUMNS=*integer*

specifies the number of columns in the layout.

Default:

- If this option is not specified and ROWS= is specified, the number of data cells (and columns) increases dynamically to allow all classifier values to be presented.
- If both this option and ROWS= are specified, a grid of that size is created, regardless of the number of classifier values. If the number of classifier values is greater than the grid size, no graphs is created for some classifier values. If the number of

classifier values is small and the grid size large, there might be empty cells created.

Interaction: The overall grid size is constrained the HEIGHT= and WIDTH= options in the ODS GRAPHICS statement. As the grid size grows, the cell size shrinks. To control the minimum size of a cell use the CELLHEIGHTMIN= and CELLWIDTHMIN= options.

The START= and ORDER= options affect the how the rows are populated.

The PANELNUMBER= option enables you to create multiple smaller grids that completely partition the classifier values.

HEADERBACKGROUNDCOLOR=*style-reference* | *color*
specifies the background color of the cell headers.

Default: The GraphHeaderBackground:Color style reference.

style-reference
A reference in the form *style-element:style-attribute*. Only the style-attribute named COLOR is used.

Interaction: HEADEROPAQUE=TRUE must be in effect for the color to be seen.

HEADERLABELATTRS=*style-element* | *style-element* (*text-options*) | (*text-options*)
specifies the color and font attributes of the data labels. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Text Options" on page 566 for available *text-options*.

Default: The GraphValueText style element.

HEADERLABELDISPLAY=NAMEVALUE | VALUE
indicates whether the automatic cell header text includes variable name and value or just the value.

Default: NAMEVALUE

Example: If the classification variables were Country and Product, HEADERLABEL=NAMEVALUE would produce cell headers such as

```
Country=CANADA
Product=TABLE
```

and HEADERLABEL=VALUE would produce

```
CANADA
TABLE
```

HEADEROPAQUE=*boolean*
specifies whether the background for cell headers is opaque (TRUE) or transparent (FALSE).

Default: TRUE

Interaction: When this option is set to FALSE, the background color for cell headers is not used.

INSET=(*variable-list*)
specifies what information is displayed in an inset. The *variable-list* defines one or more variables whose names and values appear as a small table in the data cells. The variables can be either numeric or character. Variable names are separated by spaces.

Default: no default

Restriction: No predefined information is available for the inset. You must create the desired inset information as part of your input data. This is most typically done as follows (see the chapter on classification panels and the chapter on insets in the **SAS/GRAPH Graph Template Language User's Guide** for complete examples):

- Create a separate data set for the inset columns making sure that the column names are different from the other columns used in graph. The number observations of inset data should match the number of cells in the classification panel. The ordering of the inset observations should be the same as population order of the cells of the classification panel, taking into account the CLASSVARS= argument and the ORDER= and START= options. Typically, the number of observations for the inset data is smaller than the other input data for the graph.
- Merge the inset data set with the data set for the graph using a DATA or PROC SQL step. Do not match-merge the observations of the two data sets (no BY processing). The resulting data set typically has the inset columns padded with missing values.
- Use the merged data set to produce the graph, specifying the inset column names in this option's *variable-list*.

The variable values are associated with the data cells by data order. That is, the first observation from all the variables in *variable-list* are used in the first data cell, the second observation from all variables in *variable-list* are used in the second data cell, and so on. If a value is missing for an observation, the corresponding *name-value* pair is skipped in the affected data cell.

The location and appearance of the inset is controlled by the INSETOPTS= option.

INSETOPTS=(*appearance-options*)
specifies location and appearance options for the inset information.

The *appearance-options* can be any one or more of the settings that follow. The options must be enclosed in parentheses, and each option is specified as a *name* = *value* pair.

AUTOALIGN=NONE | AUTO | (*location-list*)
specifies whether the inset is automatically aligned within the layout.

Default: NONE

NONE	Do not automatically align this inset. This inset's position is set by the HALIGN= and VALIGN= *appearance-options*.
AUTO	Attempt to center this inset in the area that is farthest from any surrounding markers. Data cells might have different inset placements.
(*location-list*)	Restrict this inset's possible locations to those locations in the specified *location-list*, and use the *location-list* position that least collides with the data cell's other graphics features. The *location-list* is blank-separated and can contain any of these locations: TOPLEFT TOP TOPRIGHT LEFT CENTER RIGHT BOTTOMLEFT BOTTOM BOTTOMRIGHT. Example: AUTOALIGN = (TOPRIGHT TOPLEFT)

Interaction: When AUTOALIGN=AUTO or (*location-list*), the HALIGN= and VALIGN= *appearance-options* are ignored.

BACKGROUNDCOLOR= *style-reference* | *color*
specifies the color of the inset background

Default: The background is transparent. No color is assigned.

style-reference	A reference of the form *style-element : style-attribute*. Only the style-attribute named COLOR is used.

BORDER= *boolean*
specifies whether a border is displayed around the inset.

Default: FALSE

HALIGN=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the inset.

Default: LEFT

Interaction: This option is ignored unless AUTOALIGN=NONE.

OPAQUE= *boolean*
specifies whether the inset background is opaque (TRUE) or transparent (FALSE).

Default: FALSE

Interaction: When OPAQUE=FALSE, the background color is not used.

TEXTATTRS=*style-element* | *style-element* (*text-options*) | (*text-options*)
specifies the text properties of the entire inset. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Text Options" on page 566 for available *text-options*.

Default: The GraphDataText style element.

VALIGN=TOP | CENTER |BOTTOM
specifies the vertical alignment of the inset.

Default: TOP

Interaction: This option is ignored unless AUTOALIGN=NONE.

OPAQUE= *boolean*
specifies whether the layout background is opaque (TRUE) or transparent (FALSE).

Default: FALSE

Interaction: When this option is set to FALSE, the background color is not used.

ORDER=ROWMAJOR | COLUMNMAJOR
specifies whether data cells are populated by column priority or by row priority.

Default: ROWMAJOR

COLUMNMAJOR
fills the data cells by columns, from the starting position.

ROWMAJOR
fills the data cells by rows, from the starting position.

PACKED
fills all pages in ROWMAJOR order, using START=TOPLEFT. When the last page is reached and it is a partial page (fewer than ROWS*COLUMNS), the number of rows and columns of cells is adjusted, based on an internal algorithm.

Interaction: The starting point for rendering data cells is controlled by the START= option. See the START= option for examples.

PAD=*dimension* | *(pad-options)*

specifies the amount of extra space that is added inside the layout border.

Default: The default padding for all sides is 0. Values without units are in pixels (px). A unit must be provided if other than pixels.

dimension

Specifies a dimension to use for the extra space at the left, right, top, and bottom of the layout border.

(*pad-options*)

Enables separate settings for the left, right, top, and bottom padding dimensions. Use the *pad-options* to create non-uniform padding. These options must be enclosed in parentheses. Each option is specified as a *name* = *value* pair. Sides not assigned padding are padded with the default amount.

Values without units are in pixels (px). A unit must be provided if other than pixels.

LEFT=*dimension*	specifies the amount of extra space added to the left side.
RIGHT=*dimension*	specifies the amount of extra space added to the right side.
TOP=*dimension*	specifies the amount of extra space added to the top.
BOTTOM=*dimension*	specifies the amount of extra space added to the bottom.

PANELNUMBER=*positive-integer*

specifies the number of the panel to produce.

Default: 1

This option enables you to partition a large grid into a number of smaller sized grids under these conditions:

- You set a grid size explicitly (ROWS= and COLUMNS= options).
- The grid size (`gridrows x gridcolumns`) is smaller than the total number of classifier levels.
- You execute the template N times and increment the panel number each time. N is determined by CEIL(`total-classification-levels / gridrows x gridcolumns`).

Example: Suppose there are two classifiers (CLASS1 has 10 unique values and CLASS2 has 11 unique values). By setting some smaller grid size, say ROWS=3 and COLUMNS=4, and making the value of PANELNUMBER= a dynamic or macro variable, you can create 10 panels (9 panels with 12 data cells and 1 panel with 2 data cells) that collectively display all 110 possible crossings. You simply invoke PROC SGRENDER or a DATA step 10 times, incrementing the dynamic value for PANELNUMBER each time.

ROWAXISOPTS=(*axis-options*)

specifies Y-axis options for all rows. For a list of options, see "Axis Options for LAYOUT DATALATTICE/DATAPANEL" on page 467.

Requirement: Axis options must be enclosed in parentheses and separated by spaces.

ROWDATARANGE=UNIONALL | UNION

specifies how the Y-axes of instances of the *graph-prototype* are scaled.

Default: UNIONALL

UNIONALL

scales the Y-axis data ranges across all layout rows and panels (when PANELNUMBER= is in effect).

UNION

scales the Y-axis data ranges separately for each row in the layout on a per-page basis. The scaling does not span across multiple pages.

Use the ROWAXISOPTS= option to control shared axis features.

See the PANELNUMBER= option for a discussion of how to create multiple panels.

ROWGUTTER=*dimension*

specifies the amount of empty space that is before the first row, between the rows, and after the last row.

Default: 0

If there are *n* rows, then there are *n*-1 gutters.

ROWS=*integer*

specifies the amount of empty space between the rows.

Default:

- If this option is not specified and COLUMNS= is specified, the number of data cells (and rows) increases dynamically to allow all classifier values to be presented.
- If both this option and COLUMNS= are specified, a grid of that size is created, regardless of the number of classifier values. If the number of classifier values is greater than the grid size, no graphs is created for some classifier values. If the number of classifier values is small and the grid size large, there might be empty cells created.

Interaction: The overall grid size is constrained the HEIGHT= and WIDTH= options in the ODS GRAPHICS statement. As the grid size grows, the cell size shrinks. To control the minimum size of a cell use the CELLHEIGHTMIN= and CELLWIDTHMIN= options.

The START= and ORDER= options affect the way the rows are populated.

The PANELNUMBER= option enables you to create multiple smaller grids that completely partition the classifier values.

SKIPEMPTYCELLS=*boolean*

specifies whether the external axes skip the empty cells in a partially filled grid.

Default: FALSE

FALSE

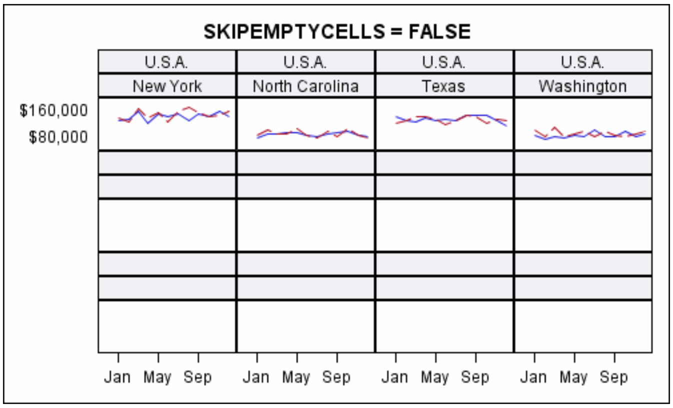
External axes are displayed at their normal locations, even if there are empty cells at one or more of the locations.

TRUE

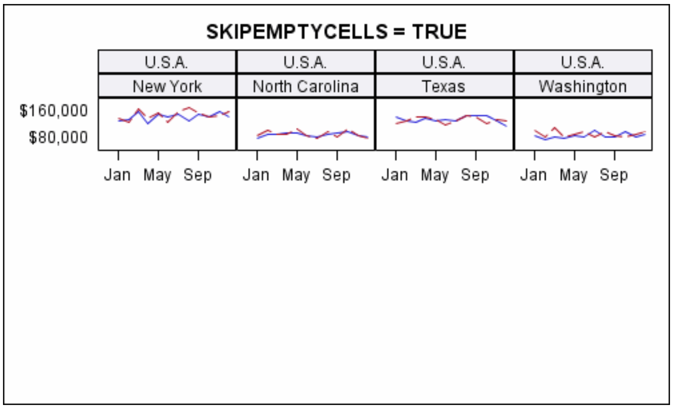
External axes skip empty cells and "snap" to the nearest data cell, both vertically and horizontally. Though the empty cells are not displayed, the data cells in the grid are not enlarged to fill the area.

Discussion: Whenever the total number of classifier crossings (data cells) is not evenly divisible by the panel size (columns * rows), the last panel is partially filled with data cells and padded with empty cells to complete the grid.

In this example, there are 16 data cells arranged in a 4-column, 3-row grid. This is default appearance of the last panel:



When SKIPEMPTYCELLS=TRUE, the empty padding cells of the last panel are removed and external axis ticks and tick values snap to the data cells:



Note that SKIPEMPTYCELLS=TRUE removes only the empty padding cells on the last panel. It does not remove any data cells that have no crossing values and therefore no graph (these data cells are displayed when SPARSE=TRUE).

SPARSE=*boolean*

Specifies whether crossings of the class variables include only the crossings in the data or all possible crossings.

Default: FALSE

FALSE

Data cells are created only for crossings of the class variables that are in the data.

TRUE

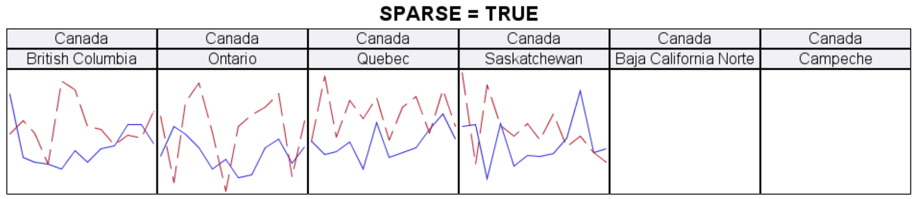
The number of data cells is the product of the unique values for each classification variable.

By default, if a crossing of the class variables has a missing value as part of the data, a data cell is created for it.

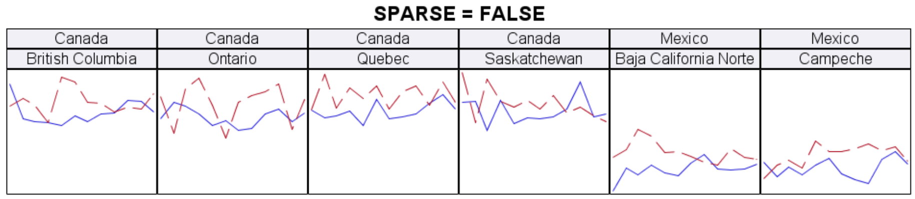
Discussion: In this example, the classification variables are COUNTRY and STATE. There are 3 distinct values of COUNTRY (Canada, Mexico, and U.S.A.) Within Canada and Mexico there are 4 states, and within U.S.A. there are 8 states. All state names are unique to each country. Therefore, there are 16 unique STATE values and

48 unique crossings of COUNTRY and STATE, but there are data for only 16 of the crossings.

Assume that a data panel layout is created with COLUMNS=6 and SPARSE=TRUE, meaning to display all possible crossings. This is what the first row would look like, with blank data cells being added whenever there are no data values for a crossing:



When SPARSE=FALSE the crossings of the classifiers with no data are automatically removed. This compacts the display:



START=TOPLEFT | BOTTOMLEFT
indicates whether to start populating the grid cells from the top left or bottom left corner.

Default: TOPLEFT

Interaction: If ORDER=ROWMAJOR (the default) and START=TOPLEFT (the default), a 2 row 2 column grid is populated as

cell 1	cell 2
cell 3	cell 4

If ORDER=ROWMAJOR (the default) and START=BOTTOMLEFT, a 2 row 2 column grid is populated as

cell 3	cell 4
cell 1	cell 2

If ORDER=COLUMNMAJOR and START=BOTTOMLEFT, a 2 row 2 column grid is populated as

cell 2	cell 4
cell 1	cell 3

If ORDER=COLUMNMAJOR and START=TOPLEFT, a 2 row 2 column grid is populated as

cell 1	cell 3
cell 2	cell 4

Chapter 6

LAYOUT GRIDDED Statement

Description and Syntax

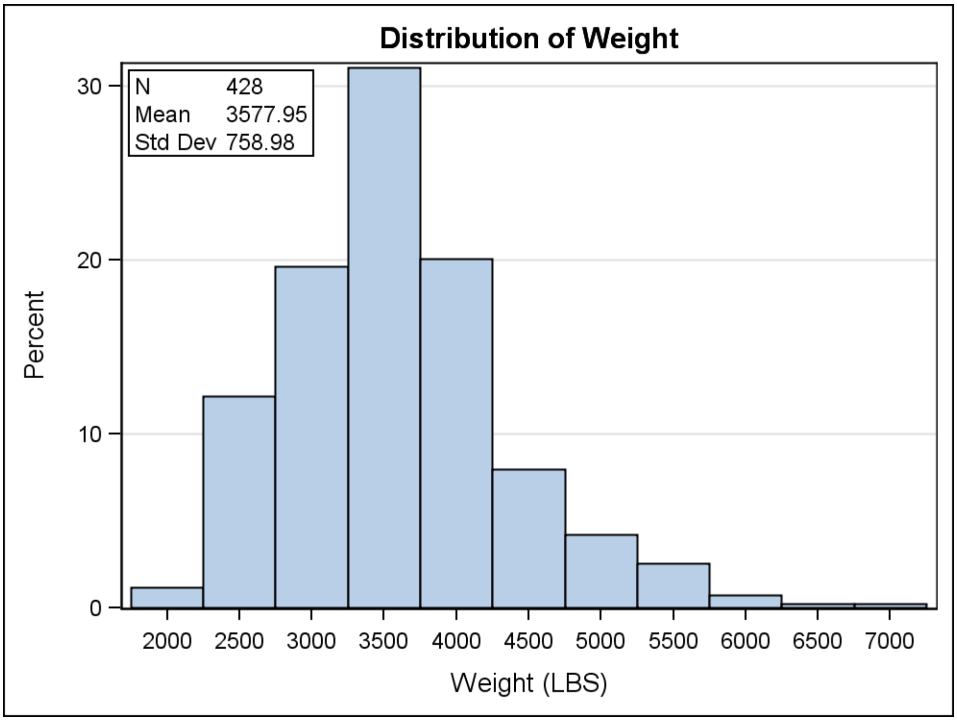
Assembles the results of nested **GTL-statements** into a grid.

LAYOUT GRIDDED *</option(s)>*
GTL-statements;
ENDLAYOUT;

Example Program and Statement Details

Example Graph

The following graph was generated by the “Example Program” on page 66:



Example Program

The GRIDDED layout offers the best way to nest a table of information inside another layout. In the GRIDDED layout, you can control the content, text justification, and fonts of columns. Because this example nests the GRIDDED layout within an OVERLAY layout, you can control where it appears within the plot area. The AUTOALIGN= option enables you to specify a prioritized list of possible positions where the layout should be drawn. The position actually used is the first one that avoids collision with the histogram. Also, the GRIDDED layout is set to be opaque so that the grid lines do not show through.

This example also illustrates a reusable template in the sense that it works for any numeric variable specified by the dynamic variable VAR. Also, SGE functions for computing the N, MEAN, STDDEV of the variable are used in the table to compute the statistics as the template is executed.

```
proc template;
 define statgraph inset;
  dynamic VAR;
  begingraph;
   entrytitle "Distribution of " VAR;
   layout overlay / yaxisopts=(griddisplay=on);
    histogram VAR / scale=percent;
    layout gridded / columns=2
        autoalign=(topleft topright) border=true
        opaque=true backgroundcolor=GraphWalls:color;
     entry halign=left "N";
     entry halign=left eval(strip(put(n(VAR),12.0)));
     entry halign=left "Mean";
     entry halign=left eval(strip(put(mean(VAR),12.2)));
     entry halign=left "Std Dev";
     entry halign=left eval(strip(put(stddev(VAR),12.2)));
    endlayout;
```

```
      endlayout;
    endgraph;
  end;
run;

proc sgrender data=sashelp.cars template=inset;
   dynamic VAR="Weight";
run;
```

Statement Summary

A GRIDDED layout is commonly used to create small tables of text that are nested within other layouts. The layout might also be used to span and center a single entry (a legend, for example) across a set of grids. Or it might be used to display a grid of graphs when there is no need to scale axis data ranges or align graphs across grid cells.

The GRIDDED layout automatically decides how much area to allocate to cell contents:

- text items have a fixed size based on the amount of text and the font properties
- graphs take up the remaining space.

The layout's grid size is determined by the COLUMNS= and ROWS= options. The resulting columns and rows can be separated by areas called "gutters", which are controlled by the COLUMNGUTTER= and ROWGUTTER= options.

By default, the results of the *GTL-statements* are placed into the grid sequentially from left to right, wrapping to a new row each time the current row is filled. You can use the ORDER= option to fill cells from top to bottom down a column. In that case, the layout cells wrap to a new column each time the current column is filled.

Options

Statement Option	Description
AUTOALIGN	Specifies whether this layout is automatically aligned within its parent when nested within an overlay-type layout.
BACKGROUNDCOLOR	Specifies the color of the layout background.
BORDER	Specifies whether a border is drawn around the layout.
BORDERATTRS	Specifies the properties of the border line.
COLUMNGUTTER	Specifies the amount of empty space between the columns.
COLUMNS	Specifies the number of columns in the layout.
HALIGN	Specifies this layout's horizontal alignment within its parent when nested within an overlay-type layout.

Statement Option	Description
LOCATION	Specifies whether the layout appears inside or outside the plot area when nested within an overlay-type layout.
OPAQUE	Specifies whether the layout background is opaque.
ORDER	Specifies whether cells are populated using column priority or row priority.
PAD	Specifies the amount of extra space that is added inside the layout border.
ROWGUTTER	Specifies the amount of empty space between the rows.
ROWS	Specifies the number of rows in the layout.
VALIGN	Specifies this layout's vertical alignment within its parent when nested within an overlay-type layout.

AUTOALIGN=NONE | AUTO | *(location-list)*

specifies whether this layout is automatically aligned within its parent when nested within an overlay-type layout. For more information about how child positions are determined in an overlay-type layout, see the LAYOUT OVERLAY's "Example Program and Statement Details" on page 89.

Default: NONE

NONE

Do not automatically align this layout within its parent layout. This layout's position within its parent layout is therefore set by the HALIGN= and VALIGN= options.

AUTO

Available only if the parent layout contains a scatter plot; ignored otherwise. Within the parent layout, attempt to center this layout in the area that is farthest from any surrounding data point markers.

(*location-list*)

Within the parent layout, restrict this layout's possible locations to those locations in the specified *location-list*, and use the *location-list* position that least collides with the parent layout's other graphics features. The *location-list* is blank-separated and can contain any of these locations: TOPLEFT, TOP, TOPRIGHT, LEFT, CENTER, RIGHT, BOTTOMLEFT, BOTTOM, and BOTTOMRIGHT.

Restriction: This option is available only when this layout statement is nested within an overlay-type layout. This option is ignored if this statement is the outermost layout or if the parent layout is not an overlay-type layout.

Interaction: If this option is enabled, it overrides the HALIGN= and VALIGN= options.

Interaction: This option is ignored if LOCATION=OUTSIDE.

BACKGROUNDCOLOR=*style-reference* | *color*

specifies the color of the layout background.

Default: The GraphBackground:Color style reference.

style-reference
A reference in the form *style-element:style-attribute*. Only the style-attribute named COLOR is used.

Interaction: OPAQUE=TRUE must be in effect for the color to be seen. By default, OPAQUE=FALSE.

BORDER=*boolean*
specifies whether a border is drawn around the layout.

Default: FALSE

Interaction: If this option is set to FALSE, the BORDERATTRS option is ignored.

BORDERATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)
specifies the attributes of the border line around the layout. See “General Syntax for Attribute Options” on page 563 for the syntax on using a *style-element* and “Line Options” on page 564 for available *line-options*.

Default: The GraphBorderLines style element.

Interaction: BORDER=TRUE must be set for this option to have any effect.

COLUMNGUTTER=*dimension*
specifies the amount of empty space between the columns.

Default: 0

If there are *n* columns, then there are *n*-1 gutters.

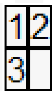
COLUMNS=*integer*
specifies the number of columns in the layout.

Default: If ORDER=ROWMAJOR, the default is 1. If ORDER=COLUMNMAJOR, as many columns are created as needed to satisfy the ROWS= request.

Restriction: Assuming ORDER=ROWMAJOR, if COLUMNS=*n* and there are *m* cells defined, and *n* > *m*, then only *m* columns are created (there are *n* - *m* cells with zero size).

This option is used to create a grid with a fixed number of columns, without concern for how many rows. For example, the following settings ensure that columns 1 and 2 in the first row are filled with content, as shown in the figure:

```
layout gridded / columns=2 order=rowmajor border=true ;
  entry '1' /border=true;
  entry '2' /border=true;
  entry '3' /border=true;
endlayout;
```



HALIGN=CENTER | LEFT | RIGHT
specifies this layout’s horizontal alignment within its parent when nested within an overlay-type layout. For more information about how child positions are determined in an overlay-type layout, see the LAYOUT OVERLAY’s “Example Program and Statement Details” on page 89.

Default: CENTER

Restriction: This option is available only when this statement is nested within an overlay-type layout. It is ignored if this statement is the outermost layout or if the parent layout is not an overlay-type layout.

Interaction: If the AUTOALIGN= option is enabled, this option is ignored.

LOCATION=INSIDE | OUTSIDE
specifies whether the legend appears inside or outside the plot area when nested within an overlay-type layout.

Default: INSIDE

Restriction: This option has effect only when the GRIDDED layout block appears within an overlay-type layout. For more information about how child positions are determined in an overlay-type layout, see the LAYOUT OVERLAY's "Example Program and Statement Details" on page 89.

Interaction: The actual position is determined by this option's setting plus the settings for the AUTOALIGN= or HALIGN= and VALIGN= options.

OPAQUE= *boolean*
specifies whether the layout background is opaque (TRUE) or transparent (FALSE).

Default: FALSE

Interaction: When this option is set to FALSE, the background color is not used.

ORDER=ROWMAJOR | COLUMNMAJOR
specifies whether cells are populated using column priority or by row priority.

Default: ROWMAJOR

ROWMAJOR
fills all the columns in a row, from left to right, before going to the next row.

COLUMNMAJOR
fills all the rows in a column, from top to bottom, before going to the next column.

Requirement: When this option is set to COLUMNMAJOR, the ROWS= option must be specified to indicate how many rows to fill before wrapping to the next column. The default number of rows is 1.

Requirement: When this option is set to ROWMAJOR, the COLUMNS= option must be specified to indicate how many columns to fill before wrapping to the next row. The default number of columns is 1.

Interaction: The ROWS= option is ignored when ORDER=ROWMAJOR. The COLUMNS= option is ignored when ORDER=COLUMNMAJOR.

PAD=*dimension* | *(pad-options)*
specifies the amount of extra space that is added inside the layout border.

Default: The default padding for all sides is 0. Values without units are in pixels (px). A unit must be provided if other than pixels.

dimension
Specifies a dimension to use for the extra space at the left, right, top, and bottom of the layout border.

(*pad-options*)
Enables separate settings for the left, right, top, and bottom padding dimensions. Use the *pad-options* to create non-uniform padding. These options must be enclosed in parentheses. Each option is specified as a *name* = *value* pair. Sides not assigned padding are padded with the default amount.

Values without units are in pixels (px). A unit must be provided if other than pixels.

LEFT=*dimension*
specifies the amount of extra space added to the left side.

RIGHT=*dimension*
specifies the amount of extra space added to the right side.

TOP=*dimension*
specifies the amount of extra space added to the top.

BOTTOM=*dimension*
specifies the amount of extra space added to the bottom.

ROWGUTTER= *dimension*
specifies the amount of empty space between the rows.

Default: 0

If there are *n* rows, then there are *n*-1 gutters.

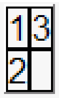
ROWS=*integer*
specifies the number of rows in the layout.

Default: If ORDER=COLUMNMAJOR, the default is 1. If ORDER=ROWMAJOR, this option is ignored and as many rows are created as needed to satisfy the COLUMNS= request.

Restriction: Assuming ORDER=COLUMNMAJOR, if ROWS=*n* and there are *m* cells defined, and *n* > *m*, then only *m* rows are created (there are *n* - *m* cells with zero size).

This option is used to create a grid with a fixed number of rows, without concern for how many columns. For example, the following settings ensure that rows 1 and 2 in the first column are filled with content, as shown in the figure:

```
layout gridded / rows=2 order=columnmajor border=true ;
  entry '1' /border=true;
  entry '2' /border=true;
  entry '3' /border=true;
endlayout;
```



1	3
2	

VALIGN=CENTER | TOP | BOTTOM
specifies this layout's vertical alignment within its parent when nested within an overlay-type layout. For more information about how child positions are determined in an overlay-type layout, see the LAYOUT OVERLAY's "Example Program and Statement Details" on page 89.

Default: CENTER

Restriction: This option is available only when this statement is nested within an overlay-type layout. It is ignored if this statement is the outermost layout or if the parent layout is not an overlay-type layout.

Interaction: If the AUTOALIGN= option is enabled, this option is ignored.

Chapter 7

LAYOUT LATTICE Statement

Description and Syntax

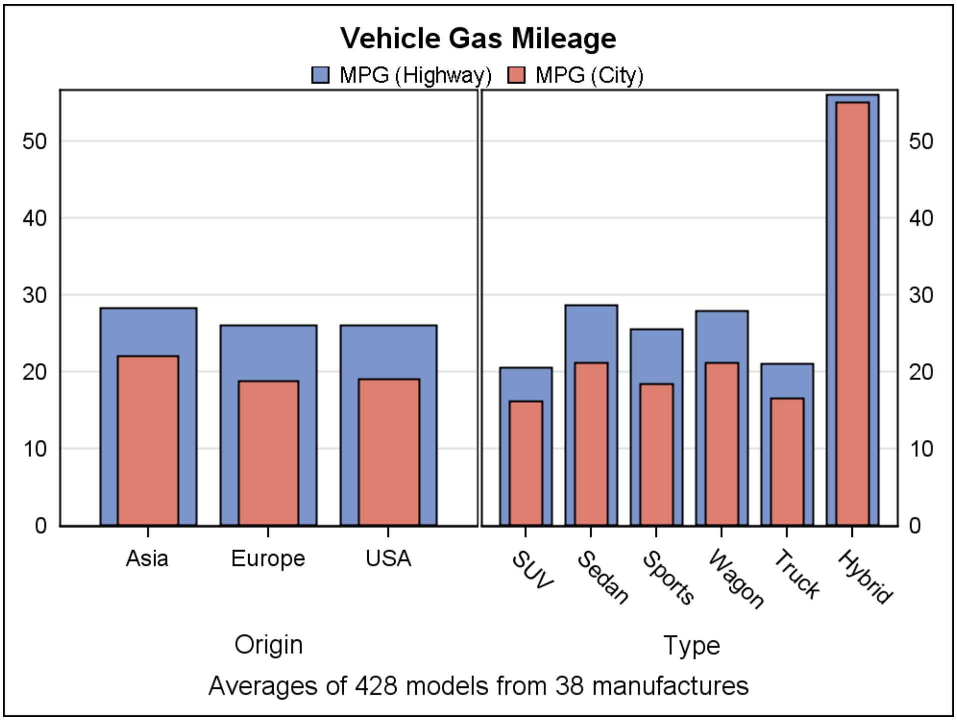
Creates a grid of graphs that automatically aligns plot areas and tick display areas across grid cells to facilitate data comparisons among graphs.

LAYOUT LATTICE </*option(s)*>;
    GTL-statement(s) | *cell-statement-block(s)*;
    <*axis-statement-block(s)*;>
    <*header-statement-block(s)*;>
    <*sidebar-statement-block(s)*;>
ENDLAYOUT;

Example Program and Statement Details

Example Graph

The following graph was generated by the “Example Program” on page 74:



Example Program

This example shows a two-cell lattice layout (two columns, one row). The ROWAXES block makes the primary and secondary Y axes external to both cells. The secondary Y axis is displayed on the right.

- The ROWDATARANGE = UNION option assures that the data ranges of all Y= variables in the row cells are considered to construct a common axis range. This facilitates the visual comparison of the cells.
- A SIDEBAR block is used to place the legend at the top of the lattice.

```
proc template;
 define statgraph layoutlattice;
  begingraph;
  entrytitle "Vehicle Gas Mileage";
  entryfootnote "Averages of 428 models from 38 manufactures";
    layout lattice / columns=2 rowdatarange=union;
     layout overlay / cycleattrs=true;
       barchart x=origin y=mpg_highway /
        stat=mean barwidth=.8 name="H" ;
       barchart x=origin y=mpg_city /
        stat=mean barwidth=.5 name="C" ;
     endlayout;
     layout overlay / cycleattrs=true;
       barchart x=type y=mpg_highway /
        stat=mean barwidth=.8;
       barchart x=type y=mpg_city /
        stat=mean barwidth=.5;
     endlayout;
     sidebar / align=top;
       discretelegend "H" "C" / border=false;
     endsidebar;
```

```
          rowaxes;
            rowaxis / display=(tickvalues)
             displaysecondary=(tickvalues) griddisplay=on;
          endrowaxes;
         endlayout;
       endgraph;
      end;
     run;
     proc sgrender data=sashelp.cars template=layoutlattice;
     run;
```

Statement Summary

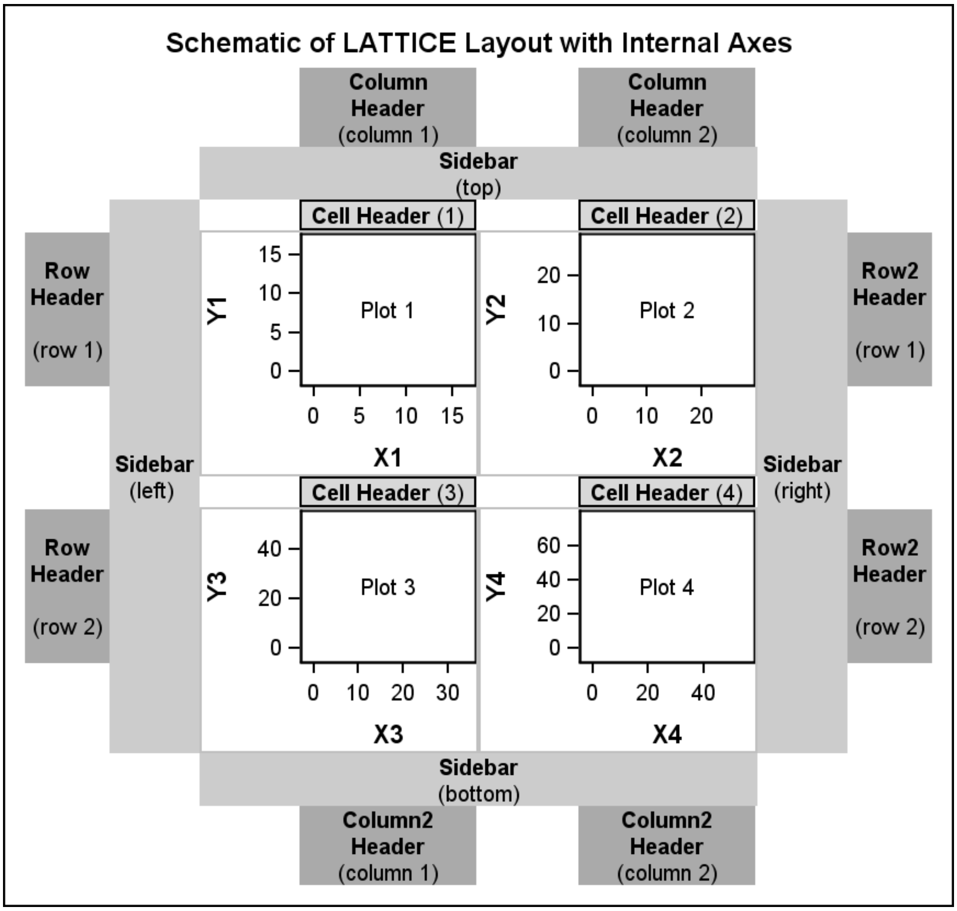
The LAYOUT LATTICE statement creates a grid of graphs that are aligned across columns and rows. For plot statements that are specified in the layout block or nested in a LAYOUT OVERLAY statement, the LATTICE layout automatically aligns the plot areas and tick display areas in the plots.

Note: To achieve the alignment, the LATTICE layout automatically aligns plot areas and tick display areas across columns and rows. Also, it overrides axis-offset settings on the OVERLAY layouts that you specify in those columns and rows. (For details about offsets and the tick display area, see "Adjusting Axis Offsets" on page 403.) If you do not want this alignment, you might use LAYOUT GRIDDED instead. For example, if you have a heterogeneous panel of graphs, such as a mix of scatter plots, box plots, bar charts, or other types of graphs, you might consider using LAYOUT GRIDDED rather than LAYOUT LATTICE.

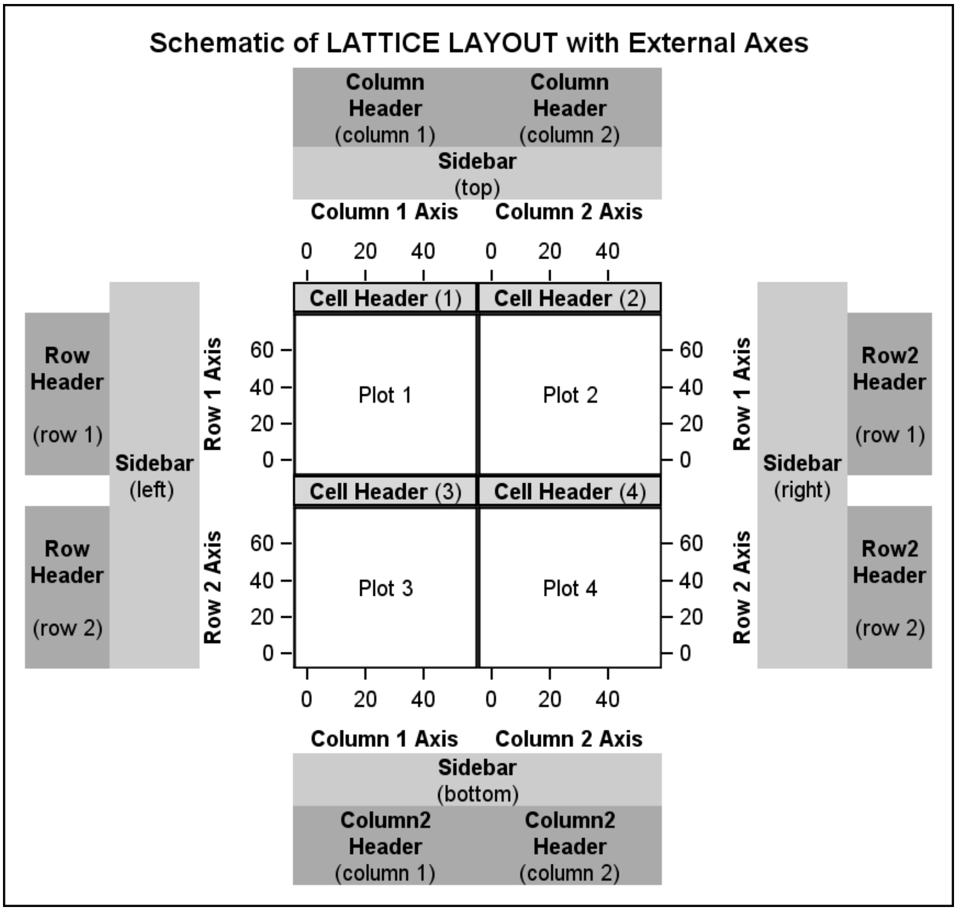
The layout can unify the scale of the data ranges that are displayed in the plots, based on the values set for the COLUMNDATARANGE and ROWDATARANGE= options. The data ranges can be scaled separately for each column, for each row, or for both. Or they can be scaled across all columns, all rows, or all of both.

When the data-range scales are unified, you can simplify the layout by displaying only the external axes that apply to all of the graphs across the corresponding columns or rows. See "Axis Statements" on page 79 for more details.

The following figure shows the parts of the Lattice layout with the default axis display (internal axes are displayed).



This next figure shows the parts of the Lattice layout when the graph display is simplified so that only external axes are displayed.



The columns and rows can be separated by areas called "gutters," which are controlled by the COLUMNGUTTER= and ROWGUTTER= options. In addition, the COLUMNWEIGHTS= and ROWWEIGHTS= options can be used to allocate a proportion of available space to each row and column.

The LATTICE layout automatically decides how much area to allocate to cell contents:

- text items have a fixed size based on the amount of text and the font properties
- graphs take up the remaining space.

The layout's grid size is determined by the COLUMNS= and ROWS= options.

By default, the results of the *GTL-statements* are placed into the grid sequentially from left to right, wrapping to a new row each time the current row is filled. You can use the ORDER= option to fill cells from top to bottom down a column. In that case, the layout cells wrap to a new column each time the current column is filled.

Cell Contents

The content of each cell in a LAYOUT LATTICE is specified by *GTL-statement(s)* that can be specified independently or enclosed in a CELL block. When you enclose the statements in a CELL block, you can specify one or more header lines at the top of the cell. If you do not need a cell header in a cell, there is no need to use the CELL block.

The following general syntax is used for the contents of each cell in a LAYOUT LATTICE:

GTL-statement(s) | *cell-statement-block(s)*

A *cell-statement-block*, when used, has the following syntax:

CELL;

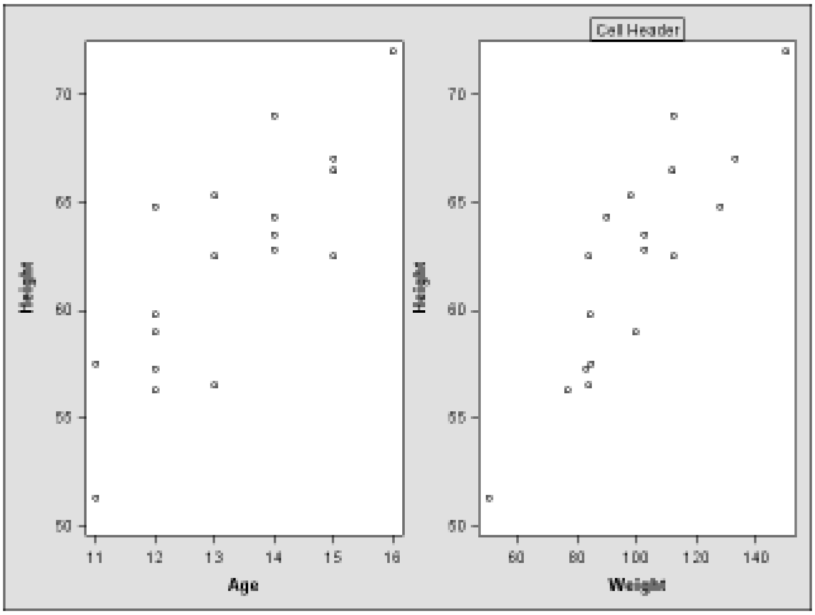
<CELLHEADER; *GTL-statement(s);*ENDCELLHEADER;>

GTL-statement(s);

ENDCELL;

- The contents of each cell is generated by *GTL-statement(s)* that can be specified independently or enclosed in a CELL block.
- Whether specified independently or enclosed in a CELL block, the *GTL-statement(s)* can include text statements, plot statements, or layout statements.
- Within a CELL block, a CELLHEADER block can be used to generate one or more header lines within the cell. Each header line is specified on a separate *GTL-statement* within the CELLHEADER block. The header block is typically used to specify one or more text statements, but other statements are allowed within the block. For example, you could specify a LAYOUT GRIDDED statement to produce a grid of text for the header
- Only one CELLHEADER block can be used per CELL block. If more than one is specified, only the last one is used.
- If a CELLHEADER block is not specified in a CELL block, the enclosed *GTL-statement(s)* produce the same results they would produce if they were specified independently.

The following example shows a LAYOUT LATTICE block that uses one *GTL-statement* and one *cell-statement-block* to generate the two-column layout shown in the following figure:



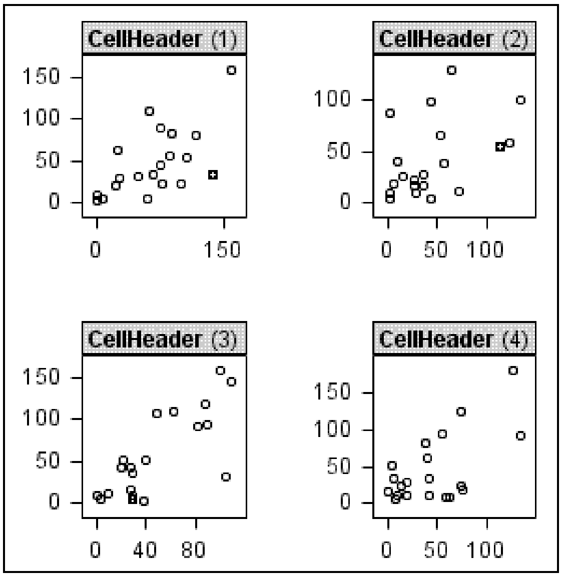
```
begingraph;
  layout lattice /
    columngutter=5 columns=2;

 /* independent plot statement - defines first cell */
    scatterplot x=age y=height;

 /* cell block - defines second cell */
    cell;
```

```
      cellheader;
        entry "Cell Header" / border =true;
      endcellheader;

      scatterplot x=weight y=height;

    endcell;
  endlayout;
endgraph;
```

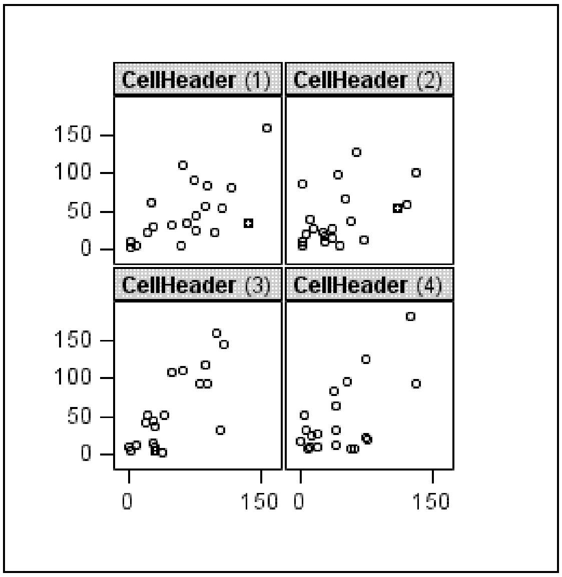
Axis Statements

The axis statements can be used to simplify and clarify the layout by displaying only the external axes in the resulting graph.

The following figure shows the default layout with internal axes displayed:



This next figure shows a simplified layout with only the external axes displayed:



Axis statements are useful only if the data ranges across the affected columns or rows are comparable and can be unified to a common scale. For example, external axes are not supported if an affected lattice cell contains a LAYOUT OVERLAYEQUATED statement.

To unify column data ranges, use the COLUMNDATARANGE option. To unify row data ranges, use the ROWDATARANGE= option. If the axis ranges are not unified for the affected columns or rows, the axis statements in the layout are ignored.

For columns, external X axes (top, bottom, or both) are specified within a COLUMNAXES block, nesting one or more COLUMAXIS statements within the block. If a lattice cell contains a LAYOUT OVERLAY with the XAXISOPTS= option specified, the XAXISOPTS= option is ignored. In such cases, the desired axis features should be specified in the COLUMNAXIS statement. An X2AXISOPTS= option is also ignored if specified on the OVERLAY.

For rows, external Y axes (left, right, or both) are specified within a ROWAXES block, nesting one or more ROWAXIS statements within the block. If a lattice cell contains a LAYOUT OVERLAY with the YAXISOPTS= option specified, the YAXISOPTS= option is ignored. In such cases, the desired axis features should be specified in the ROWAXIS statement. A Y2AXISOPTS= option is also ignored if specified on the OVERLAY.

The axis statements have the following general syntax:

COLUMNAXES;
    COLUMNAXIS / *axis-option(s)*;
    <…COLUMNAXIS-*n*;>
ENDCOLUMNAXES;

ROWAXES;
    ROWAXIS / *axis-option(s)*;
    <…ROWAXIS-*n*;>
ENDROWAXES;

- Only one COLUMNAXES block can be used in the LATTICE layout block. If more than one is specified, only the last one is used.
- Within a COLUMNAXES block, the number of COLUMNAXIS statements should match the number of columns.
- Only one ROWAXES block can be used in the LATTICE layout block. If more than one is specified, only the last one is used.
- Within a ROWAXES block, the number of ROWAXIS statements should match the number of rows.
- If the number COLUMNLAXIS or ROWAXIS statements is greater than the number needed, the extra statements are ignored. If the number of statements is fewer than the number needed, the additional COLUMNAXIS or ROWAXIS statements are automatically generated with DISPLAY=NONE options in effect.

For the list of *axis-options*, see "Axis Options for LAYOUT LATTICE" on page 437.

The following example shows a LAYOUT LATTICE block that uses a ROWAXES block to set external axes and display gridlines for the row display.

```
begingraph;
  layout lattice /
     rowdatarange=union
     columns=2;

     /* axis definitions */
```

```
         rowaxes;
            rowaxis /griddisplay=on;
         endrowaxes;

         /* cell contents */
         scatterplot x=x y=t;
         scatterplot x=x y=y;

      endlayout;
    endgraph;
```

Here, the LAYOUT LATTICE statement specifies the ROWDATARANGE option to unify the data ranges across rows in the layout. Because LAYOUT LATTICE specifies COLUMNS=2 and there are two plot statements in the template, the resulting graph has two columns and only one row. Thus, only one ROWAXIS statement is needed in the ROWAXES block to specify axis attributes for that row of graphs.

Header Statements

Header statements are used to display one or more headers for the columns and rows in a Lattice layout. Each statement is specified as a block in the form *statement - ENDstatement*. The header block is typically used to specify one or more text statements, but other statements are allowed within the block. For example, you could specify a LAYOUT GRIDDED statement to produce a grid of text for the header.

The general syntax for a COLUMNHEADERS statement is

COLUMNHEADERS;
   GTL-statement(s);
ENDCOLUMNHEADERS;

The following header statements are available:

COLUMNHEADERS	specifies a header for the primary (bottom) column-header position.
COLUMN2HEADERS	specifies a header for the secondary (top) column-header position.
ROWHEADERS	specifies a header for the primary (left) row-header position. ENTRY statements can be used to specify rotated text.
ROW2HEADERS	specifies a header for the secondary (right) row-header position. ENTRY statements can be used to specify rotated text.

- The LAYOUT LATTICE aligns headers with the columns, or the rows, or both.
- Each of the header blocks COLUMNHEADERS, COLUMN2HEADERS, ROWHEADERS, and ROW2HEADERS can be used once in a LAYOUT LATTICE block. If more than one block is specified for one of the statements, only the last specified block for that statement is used.

The following example shows a LAYOUT LATTICE block that uses a COLUMNHEADERS block to display column headers above the left and right columns in the layout.

```
begingraph;
   layout lattice / columns=2;

      /* Lattice header definitions */
      columnheaders;
         entry "Left Column";
         entry "Right Column";
      endcolumnheaders;

      /* cell contents */
      scatter x=x y=t;
      scatter x=x y=y;

   endlayout;
 endgraph;
```

Sidebar Statements

A LAYOUT LATTICE supports the display of a *sidebar* between a row or column header and an external axis. (See the figures in .)“Example Program and Statement Details” on page 73 A sidebar spans across columns or rows and is useful for displaying information that applies to all of the columns or all of the rows. For example, sidebars are useful for displaying legends.

A SIDEBAR statement has the following syntax:

SIDEBAR </ ALIGN= BOTTOM | TOP | LEFT | RIGHT>;
    GTL-statement(s);
ENDSIDEBAR;

- ALIGN=BOTTOM is the default alignment.
- You can specify up to four SIDEBAR blocks in a LAYOUT LATTICE, one for each of the top, bottom, left, and right sidebar positions.
- If two or more SIDEBAR blocks have the same alignment, the sidebar information forms two or more columns (ALIGN=LEFT or ALIGN=RIGHT) within the sidebar area. Or it forms two or more rows (ALIGN=TOP or ALIGN=BOTTOM) within the sidebar area.
- Only one statement (such as ENTRY or DISCRETELEGEND) or one layout block (such as LAYOUT GRIDDED) is allowed in a SIDEBAR block. To create multi-line text in a sidebar, nest ENTRY statements within a LAYOUT GRIDDED block.
- The LAYOUT LATTICE automatically aligns a sidebar with the layout columns or rows.
- Using ENTRY statements with the ROTATE= option, it is possible to place rotated text in the right or left sidebars.

The following example shows a LAYOUT LATTICE block that uses a SIDEBAR block to display a top sidebar in the layout.

```
begingraph;
 layout lattice / columns=2;

  sidebar / align=top;
    layout gridded / border=true ;
      entry "Top Sidebar" ;
      entry "(spans both columns)";
    endlayout;
  endsidebar;

  scatterplot x=x y=t;
  scatterolot x=x y=y;

 endlayout;
begingraph;
```

Options

Statement Option	Description
AUTOALIGN	specifies whether this layout is automatically aligned within its parent when nested within an overlay-type layout.
BACKGROUNDCOLOR	Specifies the color of the layout background.
BORDER	Specifies whether a border is drawn around the layout.
BORDERATTRS	Specifies the properties of the border line.
COLUMNDATARANGE	Specifies how the X-axis data ranges of graphs within the layout columns are scaled.
COLUMNGUTTER	Specifies the amount of empty space between the columns.
COLUMNS	Specifies the number of columns in the layout.
COLUMNWEIGHTS	Specifies the fractional proportion of each cell relative to the overall grid width, not including headers, sidebars, and column axes.
HALIGN	Specifies this layout’s horizontal alignment within its parent when nested within an overlay-type layout.
OPAQUE	Specifies whether the layout background is opaque.
ORDER	Specifies whether cells are populated with column priority or row priority.
PAD	Specifies the amount of extra space that is added inside the layout border.
ROWDATARANGE	Specifies how the Y-axis data ranges of graphs within the layout are scaled.
ROWGUTTER	Specifies amount of empty space between the rows.

Statement Option	Description
ROWS	Specifies the number of rows in the layout.
ROWWEIGHTS	Specifies the fractional proportion of each cell relative to the overall grid height, not including headers, sidebars, and row axes.
SKIPEMPTYCELLS	Specifies whether the external axes skip the unused cells in a partially filled lattice.
VALIGN	Specifies this layout's vertical alignment within its parent when nested within an overlay-type layout.

AUTOALIGN= NONE | AUTO | *(location-list)*

specifies whether this layout is automatically aligned within its parent when nested within an overlay-type layout. For more information about how child positions are determined in an overlay-type layout, see the LAYOUT OVERLAY's "Example Program and Statement Details" on page 89.

Default: NONE

NONE

Do not automatically align this layout within its parent layout. This layout's position within its parent layout is therefore set by the HALIGN= and VALIGN= options.

AUTO

Available only if the parent layout contains a scatter plot; ignored otherwise. Within the parent layout, attempt to center this layout in the area that is farthest from any surrounding data point markers.

(*location-list*)

Within the parent layout, restrict this layout's possible locations to those locations in the specified *location-list*, and use the *location-list* position that least collides with the parent layout's other graphics features. The *location-list* is blank-separated and can contain any of these locations: TOPLEFT, TOP, TOPRIGHT, LEFT, CENTER, RIGHT, BOTTOMLEFT, BOTTOM, and BOTTOMRIGHT.

```
dynamic VAR STAT1 STAT2 STAT3;
layout overlay;
  histogram VAR;
  layout lattice / AUTOALIGN=(TOPRIGHT TOPLEFT)
columns=1;
     entry STAT1;
     entry STAT2;
     entry STAT3;
  endlayout;
endlayout;
```

Here the LATTICE layout is the child of the OVERLAY layout. The child layout appears in either the top right or top left position, based on which position has more "unoccupied" area.

Restriction: This option is available only when this layout statement is nested within an overlay-type layout. This option is ignored if this statement is the outermost layout or if the parent layout is not an overlay-type layout.

Interaction: If this option is enabled, it overrides the HALIGN= and VALIGN= options.

BACKGROUNDCOLOR=*style-reference* | *color*
specifies the color of the layout background.

Default: The GraphBackground:Color style reference.

style-element
A reference in the form *style-element:style-attribute*. Only the style-attribute named COLOR is used.

Interaction: OPAQUE=TRUE must be in effect for the color to be seen. By default, OPAQUE=FALSE.

BORDER=*boolean*
specifies whether a border is drawn around the layout.

Default: FALSE

Interaction: If this option is set to FALSE, the BORDERATTRS option is ignored.

BORDERATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)
specifies the attributes of the border line around the layout. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Line Options" on page 564 for available *line-options*.

Default: The GraphBorderLines style element.

Interaction: BORDER=TRUE must be set for this option to have any effect.

COLUMNDATARANGE= DATA | UNION | UNIONALL
specifies how the X-axis data ranges of graphs within the layout columns are scaled.

Default: DATA

DATA
scales the X-axis data ranges separately for each cell in the layout.

UNION
scales the X-axis data ranges separately for each column in the layout. This setting is supported only if all plots across the column can share the same data range and axis type. For more information, see "Plot Axis Types Must Agree on Common Axes" on page 400.

UNIONALL
scales the X-axis data ranges across all columns in the layout. This setting is supported only if all plots across the column can share the same data range and axis type. For more information, see "Plot Axis Types Must Agree on Common Axes" on page 400.

Restriction: If any plot statement in any cell contains an XAXIS=X2 option, this option is ignored if it is set to UNION or UNIONALL. The union setting causes all data to be mapped to the X axis.

Interaction: Axes are always internal to the cell, by default. To externalize column axes, 1) set this option to UNION or UNIONALL 2) specify a COLUMNAXES block with as many COLUMNAXIS statements as there are columns.

If column axes are externalized, and if a lattice cell contains a LAYOUT OVERLAY with the XAXISOPTS= option specified, the XAXISOPTS option is ignored. In such cases, the COLUMNAXIS statement should be used to specify desired axis features.

COLUMNGUTTER=*dimension*
specifies the amount of empty space between the columns.

Default: 0

If there are **n** columns, then there are **n**-1 gutters.

COLUMNS=*integer*
specifies the number of columns in the layout.

Default: If ORDER=ROWMAJOR, the default is 1. If ORDER=COLUMNMAJOR, as many columns are created as are needed to satisfy the ROWS= request.

Interaction: If both ROWS=*n* and COLUMNS=*m* is specified, an *n* by *m* grid of cells is created. If the number of statements that define cell contents is greater than *n* x *m*, the grid size does not expand and some statements are not displayed. If the number of statements that define cell contents is less than *n* x *m*, the grid will contain empty cells.

If this option is not defined and ORDER=COLUMNMAJOR, the number of columns is dynamically determined by the number of defined cells.

COLUMNWEIGHTS= (*numeric-list*)
specifies the fractional proportion of each cell relative to the overall grid width, not including headers, sidebars, and column axes.

Requirement: *numeric-list* must be enclosed in parentheses. If there are *n* columns, then the list specified should contain *n* values and the sum of the weights should be 1.0.

HALIGN= CENTER | LEFT | RIGHT
specifies this layout's horizontal alignment within its parent when nested within an overlay-type layout. For more information about how child positions are determined in an overlay-type layout, see the LAYOUT OVERLAY's "Example Program and Statement Details" on page 89.

Default: CENTER

Restriction: This option is available only when this statement is nested within an overlay-type layout. It is ignored if this statement is the outermost layout or if the parent layout is not an overlay-type layout.

Interaction: If the AUTOALIGN= option is enabled, this option is ignored.

Discussion: In the following example, the LATTICE layout is the child of the OVERLAY layout and is positioned in the OVERLAY's top right corner.

```
dynamic VAR STAT1 STAT2 STAT3;
layout overlay;
  histogram VAR;
  layout lattice / VALIGN=TOP HALIGN=RIGHT
columns=1;
     entry STAT1;
     entry STAT2;
     entry STAT3;
  endlayout;
endlayout;
```

OPAQUE= *boolean*
specifies whether the layout background is opaque (TRUE) or transparent (FALSE).

Default: FALSE

Interaction: When this option is set to FALSE, the background color is not used.

ORDER= ROWMAJOR | COLUMNMAJOR
specifies whether cells are populated with column priority or by row priority.

Default: ROWMAJOR

ROWMAJOR
fills all the columns in a row, from left to right, before going to the next row.

COLUMNMAJOR
fills all the rows in a column, from top to bottom, before going to the next column.

Requirement: If this option is set to COLUMNMAJOR, the ROWS= option must be specified to indicate how many rows to fill before wrapping to the next column. The default number of rows is 1.

Requirement: If this option is set to ROWMAJOR, the COLUMNS= option must be specified to indicate how many columns to fill before wrapping to the next column. The default number of columns is 1.

PAD=*dimension* | *(pad-options)*
specifies the amount of extra space that is added inside the layout border.

Default: The default padding for all sides is 0. Values without units are in pixels (px). A unit must be provided if other than pixels.

dimension
Specifies a dimension to use for the extra space at the left, right, top, and bottom of the layout border.

(*pad-options*)
Enables separate settings for the left, right, top, and bottom padding dimensions. Use the *pad-options* to create non-uniform padding. These options must be enclosed in parentheses. Each option is specified as a *name* = *value* pair. Sides not assigned padding are padded with the default amount.

Values without units are in pixels (px). A unit must be provided if other than pixels.

LEFT=*dimension*	specifies the amount of extra space added to the left side.
RIGHT=*dimension*	specifies the amount of extra space added to the right side.
TOP=*dimension*	specifies the amount of extra space added to the top.
BOTTOM=*dimension*	specifies the amount of extra space added to the bottom.

ROWDATARANGE= DATA | UNION | UNIONALL
specifies how the Y-axis data ranges of graphs within the layout rows are scaled.

Default: DATA

DATA
scales the Y-axis data ranges separately for each cell in the layout.

UNION
scales the Y-axis data ranges separately for each row in the layout. This setting is supported only if all plots down the row can share the same data range and axis type. For more information, see "Plot Axis Types Must Agree on Common Axes" on page 400.

UNIONALL
scales the Y-axis data ranges across all rows in the layout. This setting is supported only if all plots down the row can share the same data range and axis type. For more information, see "Plot Axis Types Must Agree on Common Axes" on page 400.

Restriction: If any plot statement in any cell contains a YAXIS=Y2 option, this option is ignored whenever it is set to UNION or UNIONALL. The union setting causes all data to be mapped to the Y axis.

Interaction: Axes are always internal to the cell, by default. To externalize row axes, 1) set this option to UNION or UNIONALL 2) specify a ROWAXES block with as many ROWAXIS statements as there are rows.

If row axes are externalized, and if a lattice cell contains a LAYOUT OVERLAY with the YAXISOPTS= option specified, the YAXISOPTS option is ignored. In such cases, the ROWAXIS statement should be used to specify desired axis features.

ROWGUTTER=*dimension*
specifies amount of empty space between the rows.

Default: 0

If there are *n* rows, then there are *n*-1 gutters.

ROWS=*integer*
specifies the number of rows in the layout.

Default: If ORDER=COLUMNMAJOR, the default is 1. If ORDER=ROWMAJOR, as many ROWS are created as needed to satisfy the COLUMNS= request.

Interaction: If both ROWS=*n* and COLUMNS=*m* is specified, an *n* by *m* grid of cells is created. If the number of statements that define cell contents is greater than *n* x *m*, the grid size does not expand and some statements are not displayed. If the number of statements that define cell contents is less than *n* x *m*, the grid will contain empty cells.

ROWWEIGHTS=(*numeric-list*)
specifies the fractional proportion of each cell relative to the overall grid height, not including headers, sidebars, and row axes.

Requirement: *numeric-list* must be enclosed in parentheses. If there are *n* columns, then the list specified should contain *n* values and the sum of the weights should be 1.0.

SKIPEMPTYCELLS=*boolean*
specifies whether the external axes skip the unused cells in a partially filled lattice.

Default: FALSE

FALSE
External axes are displayed at their normal locations.

TRUE
Empty cells are skipped by external axes and the axes “snap” to the nearest populated cell, both vertically and horizontally.

VALIGN=CENTER | TOP | BOTTOM
specifies this layout’s vertical alignment within its parent when nested within an overlay-type layout. For more information about how child positions are determined in an overlay-type layout, see the LAYOUT OVERLAY’s “Example Program and Statement Details” on page 89.

Default: CENTER

Restriction: This option is available only when this statement is nested within an overlay-type layout. It is ignored if this statement is the outermost layout or if the parent layout is not an overlay-type layout.

Interaction: If the AUTOALIGN= option is enabled, this option is ignored.

Chapter 8

LAYOUT OVERLAY Statement

Description and Syntax

Builds a composite from one or more *GTL-statements*. The composite could be an entire graph. Or, if this layout is nested in a GRIDDED or LATTICE layout, the composite typically provides contents for one cell in the parent layout.

LAYOUT OVERLAY <*/option(s)*>;

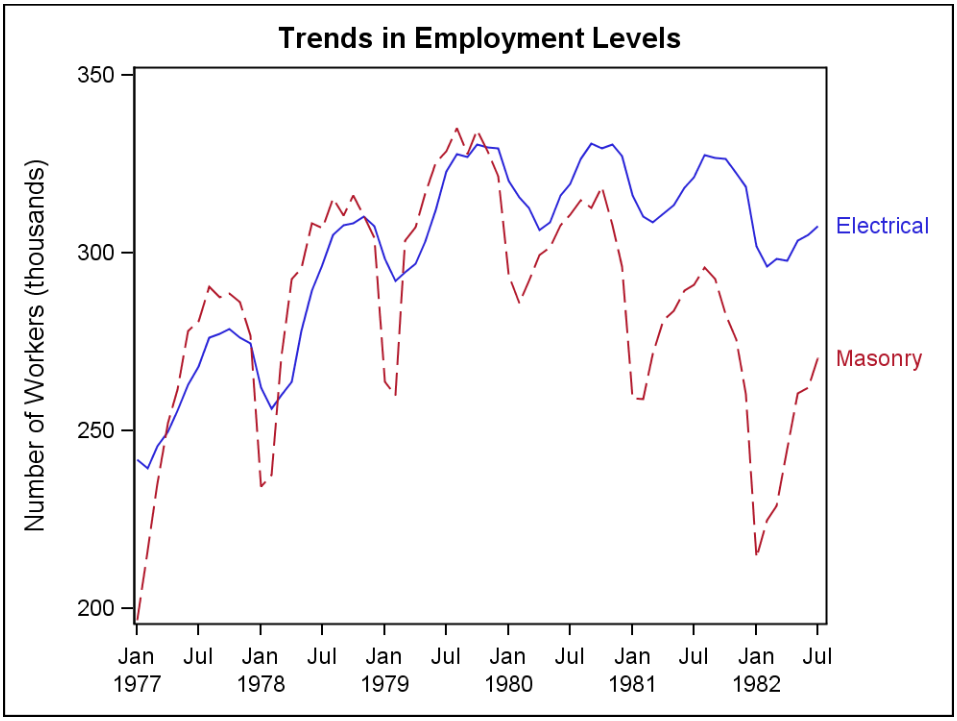
GTL-statements;

ENDLAYOUT;

Example Program and Statement Details

Example Graph

The following graph was generated by the "Example Program" on page 90:



Example Program

```
proc template;
  define statgraph layoutoverlay;
    begingraph;
      entrytitle "Trends in Employment Levels";
      layout overlay / cycleattrs=true
         xaxisopts=(display=(ticks tickvalues))
         yaxisopts=(label="Number of Workers (thousands)");
        seriesplot x=date y=electric /
         curvelabel="Electrical"
         curvelabellocation=outside;
        seriesplot x=date y=masonry / curvelabel="Masonry"
         curvelabellocation=outside;
      endlayout;
    endgraph;
  end;
run;
proc sgrender data=sashelp.workers template=layoutoverlay;
run;
```

Statement Summary

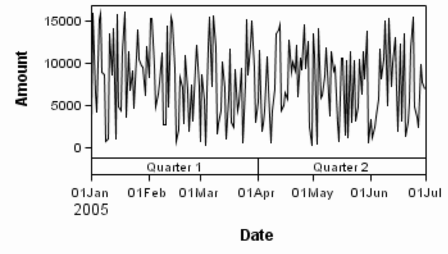
The LAYOUT OVERLAY statement builds a composite using one or more *GTL-statements*. You can specify one or more 2-D plots within the layout, provided all plots can share the same type of axes. You can also specify one or more insets, such as nested layout statements (for example, LAYOUT GRIDDED), ENTRY statements, and legend statements (CONTINUOUSLEGEND or DISCRETELEGEND).

The following general logic applies to rendering the composite:

- All plot statements are rendered first. Plot statement results are always rendered in the plot area. The plots are stacked on top of one another in the order in which they are specified, with the last one on top. It is possible for one plot's graphical data to obscure graphical data beneath it. You can control this by selectively ordering the plot statements, or by using transparency on the individual plots, or by doing both.
- The insets are rendered next, in the order in which they are specified. As with the plot statements, it is possible for the insets to obscure the results of other statements in the layout.
- To control the horizontal and vertical positioning of an inset, use the inset statement's AUTOALIGN= option, or its HALIGN= and VALIGN= options. Each nested inset determines its own relative position in the parent OVERLAY.
- Some insets, like legends, can be positioned inside or outside of the plot area using the inset statement's LOCATION= option. The inset's AUTOALIGN= or HALIGN= and VALIGN= settings are then relative to that location.

Generally, the first specified plot determines the layout's default axis characteristics. To enable another plot to define the axis characteristics, set PRIMARY=TRUE for that plot. For more information about the default axis characteristics, see "When Plots Share Data and a Common Axis" on page 397.

See Also: LAYOUT OVERLAYEQUATED and LAYOUT OVERLAY3D.



Options

Statement Option	Description
BACKGROUNDCOLOR	Specifies the color of the layout background.
BORDER	Specifies whether a border is drawn around the layout.
BORDERATTRS	Specifies the properties of the border line.
CYCLEATTRS	Specifies whether the default visual attributes of markers, lines, and fills in nested plot statements automatically change from plot to plot.
OPAQUE	Specifies whether the layout background is opaque.
PAD	Specifies the amount of extra space that is added inside the layout border.

Statement Option	Description
WALLCOLOR	Specifies the fill color of the plot wall area.
WALLDISPLAY	Specifies whether the plot's wall and wall outline are displayed.
XAXISOPTS	Specifies one or more X axis options.
X2AXISOPTS	Specifies one or more X2 axis options.
YAXISOPTS	Specifies one or more Y axis options.
Y2AXISOPTS	Specifies one or more Y2 axis options.

BACKGROUNDCOLOR=*style-reference* | *color*
specifies the color of the layout background.

Default: The GraphBackground:Color style reference.

style-reference
A reference in the form *style-element:style-attribute*. Only the style-attribute named COLOR is used.

Interaction: OPAQUE=TRUE must be in effect for the color to be seen. By default, OPAQUE=FALSE.

BORDER=*boolean*
specifies whether a border is drawn around the layout.

Default: FALSE

BORDERATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)
specifies the attributes of the border line around the layout. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Line Options" on page 564 for available *line-options*.

Default: The GraphBorderLines style element.

Interaction: BORDER=TRUE must be set for this option to have any effect.

CYCLEATTRS=*boolean*
specifies whether the default visual attributes of markers, lines, and fills in nested plot statements automatically change from plot to plot. See "Cycling Through Group Attributes in Overlaid Plots" on page 121 for more information.

Default: FALSE

FALSE
does not cycle the default visual attributes of multiple plots. For example, if you overlay three series plots, each series line has the same default visual properties.

TRUE
the system looks at the plots in the layout and tries to use the GraphData1 - GraphDataN style elements to assign different visual properties to applicable plots (scatter plots and series plots and others). For example, if you overlay three series plots, the three series lines receive different default visual properties (line pattern and color). These default properties are determined by the next GraphDataN style element not already used. Some plots in the layout do not participate in the cycling (for example, reference lines and drop lines).

OPAQUE= *boolean*
specifies whether the layout background is opaque (TRUE) or transparent (FALSE).

Default: FALSE

Interaction: When this option is set to FALSE, the background color is not used.

PAD=*dimension* | *(pad-options)*
specifies the amount of extra space that is added inside the layout border.

Default: The default padding for all sides is 0. Values without units are in pixels (px). A unit must be provided if other than pixels.

dimension
Specifies a dimension to use for the extra space at the left, right, top, and bottom of the layout border.

(*pad-options*)
Enables separate settings for the left, right, top, and bottom padding dimensions. Use the *pad-options* to create non-uniform padding. These options must be enclosed in parentheses. Each option is specified as a *name* = *value* pair. Sides not assigned padding are padded with the default amount.

Values without units are in pixels (px). A unit must be provided if other than pixels.

LEFT=*dimension* — specifies the amount of extra space added to the left side.

RIGHT=*dimension* — specifies the amount of extra space added to the right side.

TOP=*dimension* — specifies the amount of extra space added to the top.

BOTTOM=*dimension* — specifies the amount of extra space added to the bottom.

WALLCOLOR=*style-reference* | *color*
specifies the fill color of the plot wall area.

Default: The GraphWalls:Color style reference.

style-reference
A reference in the form *style-element:style-attribute*. Only the style- attribute named COLOR is used.

Interaction: This option is ignored if WALLDISPLAY=NONE or WALLDISPLAY=(OUTLINE).

WALLDISPLAY=STANDARD | ALL | NONE | *(display-options)*
specifies whether the plot's wall and wall outline are displayed.

Default: STANDARD

STANDARD
Displays a filled wall. The setting of the FRAMEBORDER= ON | OFF attribute of the GraphWalls style element determines whether the wall outline is displayed.

ALL
Displays a filled, outlined wall.

NONE
Displays no wall, no wall outline.

(*display-options*)
These options must be enclosed in parentheses and include one of the following:

OUTLINE	displays the wall outline.
FILL	displays a filled wall area.

Use the WALLCOLOR= option to control the fill color of the wall.

The appearance attributes of the wall outline are set by the GraphAxisLine style element.

XAXISOPTS= (*axis-options*)
specifies one or more X axis options. For a list of options, see "Axis Options for LAYOUT OVERLAY" on page 405.

Requirement: Axis options must be enclosed in parentheses and separated by spaces.

X2AXISOPTS= (*axis-options*)
specifies one or more X2 axis options. For a list of options, see "Axis Options for LAYOUT OVERLAY" on page 405.

Requirement: Axis options must be enclosed in parentheses and separated by spaces.

YAXISOPTS= (*axis-options*)
specifies one or more Y axis options. For a list of options, see "Axis Options for LAYOUT OVERLAY" on page 405.

Requirement: Axis options must be enclosed in parentheses and separated by spaces.

Y2AXISOPTS= (*axis-options*)
specifies one or more Y2 axis options. For a list of options, see "Axis Options for LAYOUT OVERLAY" on page 405.

Requirement: Axis options must be enclosed in parentheses and separated by spaces.

Chapter 9

LAYOUT OVERLAYEQUATED Statement

Description and Syntax

Builds a composite from one or more **GTL-statements**. The composite could be an entire graph. Or, if this layout is nested in another layout, such as a GRIDDED layout, the composite typically provides contents for one cell in the parent layout. In an OVERLAYEQUATED layout, the display unit of the X axis always equals the display unit of the Y axis.

LAYOUT OVERLAYEQUATED *</option(s)>*;

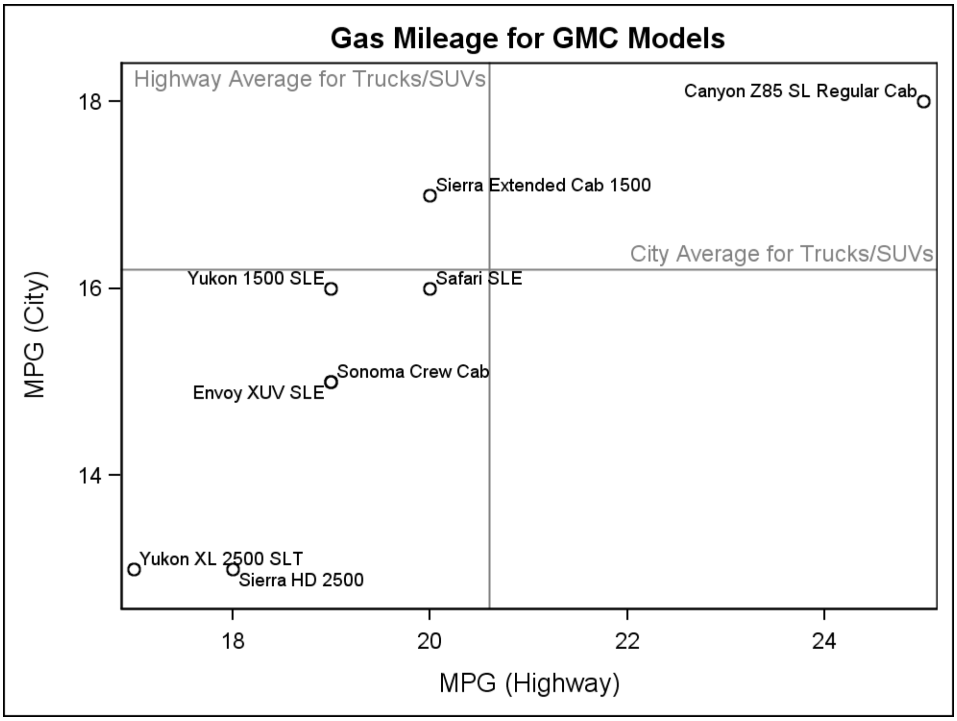
GTL-statements;

ENDLAYOUT;

Example Program and Statement Details

Example Graph

The following graph was generated by the “Example Program” on page 96:



Example Program

```
proc template;
  define statgraph layoutoverlayequated;
    begingraph;
      entrytitle "Gas Mileage for GMC Models";
      layout overlayequated / equatetype=fit;
        referenceline y=16.2 /
         curvelabel="City Average for Trucks/SUVs"
         curvelabellocation=inside
         curvelabelattrs=GraphReference;
        referenceline x=20.6 /
         curvelabel="Highway Average for Trucks/SUVs"
         curvelabellocation=inside
         curvelabelattrs=GraphReference;
        scatterplot x=mpg_highway y=mpg_city /
         datalabel=model;
      endlayout;
    endgraph;
  end;
run;
proc sgrender data=sashelp.cars
     template=layoutoverlayequated;
  where make="GMC";
run;
```

Statement Summary

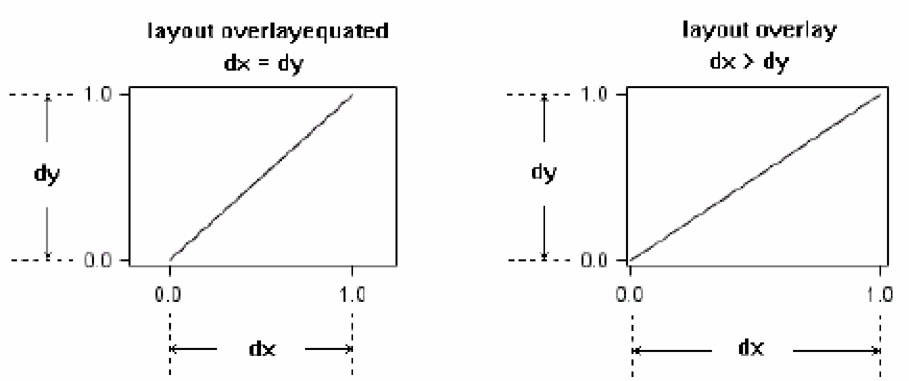
The LAYOUT OVERLAYEQUATED statement is similar to the LAYOUT OVERLAY statement: it builds a composite using one or more *GTL-statements*. Similar to a LAYOUT OVERLAY, you can specify one or more 2-D plots within the layout, provided all plots can share the same type of axes. (Additional restrictions are discussed in a moment.) You can also specify one or more insets.

As an overlay-type layout, OVERLAYEQUATED has the same behavioral characteristics as an OVERLAY layout. It uses the same general logic for rendering the composite (see "LAYOUT OVERLAY Statement" on page 89 for details), and its default axis characteristics are generally determined by the first specified plot, unless you use PRIMARY=TRUE on an alternative plot statement (see "When Plots Share Data and a Common Axis" on page 397).

OVERLAYEQUATED differs from OVERLAY in several ways. With OVERLAYEQUATED,

- The axis type for both X and Y axes is always linear. Thus, plot types that have discrete or binned axes cannot be used within this layout (for example, BOXPLOT, BOXPLOTPARM, BARCHARTPARM, HISTOGRAM, and HISTOGRAMPARM).
- For equal data intervals on both axes, the display distance is the same. For example, an interval of 2 on the X axis maps to the same display distance as an interval of 2 on the Y axis.
- The aspect ratio of the plot display equals the aspect ratio of the plot data. In other words, a 45 degree slope in data is represented by a 45 degree slope in the display. The EQUATETYPE= option determines how the axes are drawn.

The following figure illustrates how a series plot might map differently when specified in an OVERLAYEQUATED layout versus an OVERLAY layout:



A LAYOUT OVERLAYEQUATED statement enables you to specify one or more of the following XY plots: SCATTERPLOT, SERIESPLOT, NEEDLEPLOT, STEPPLOT, VECTORPLOT, BANDPLOT, LOESSPLOT, REGRESSIONPLOT, PBSPLINEPLOT, and CONTOURPLOTPARM. As long as one of these plots is present, you can also add FRINGEPLOT, LINEPARM, MODELBAND, REFERENCELINE, DROPLINE, and insets as ENTRY, DISCRETELEGEND, and CONTINUOUSLEGEND.

From a data standpoint, this layout has only two independent axes, X and Y. If any plots within the layout block use an XAXIS=X2 or YAXIS=Y2 option, the option is ignored and the data are mapped to the X or Y axis. To display X2 and Y2 axes, use the DISPLAYSECONDARY= suboption of the XAXISOPTS= and YAXISOPTS= options.

If an OVERLAYEQUATED statement is nested in a LATTICE layout, some of the LATTICE's alignment and external axis features are not supported on the OVERLAYEQUATED layout.

Options

Statement Option	Description
BACKGROUNDCOLOR	Specifies the color of the layout background.
BORDER	Displays the border around the layout.
BORDERATTRS	Specifies the properties of the border line.
COMMONAXISOPTS	Specifies one or more axis options to be applied to all displayed axes.
CYCLEATTRS	Specifies whether the default visual attributes of markers, lines, and fills in nested plot statements automatically change from plot to plot.
EQUATETYPE	Specifies how to draw the axis area.
OPAQUE	Specifies whether the layout background is opaque.
PAD	Specifies the amount of extra space that is added inside the layout border.
WALLCOLOR	Specifies the fill color of the plot wall area.
WALLDISPLAY	Specifies whether the plot's wall and wall outline are displayed.
XAXISOPTS	Specifies one or more X axis options.
YAXISOPTS	Specifies one or more Y axis options.

BACKGROUNDCOLOR=*style-reference* | *color*
specifies the color of the layout background.

Default: The GraphBackground:Color style reference.

style-element
A reference in the form *style-element:style-attribute*. Only the style-attribute named COLOR is used.

Interaction: OPAQUE=TRUE must be in effect for the color to be seen. By default, OPAQUE=FALSE.

BORDER=*boolean*
specifies whether a border is drawn around the layout.

Default: FALSE

BORDERATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)
specifies the attributes of the border line around the layout. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Line Options" on page 564 for available *line-options*.

Default: The GraphBorderLines style element.

Interaction: BORDER=TRUE must be set for this option to have any effect.

COMMONAXISOPTS= (*common-equated-axis-options*)

specifies one or more axis options to be applied to all displayed axes. For a list of options, see “Options That Apply in Common to Both Equated Axes” on page 456

Requirement: Axis options must be enclosed in parentheses and separated by spaces.

CYCLEATTRS=*boolean*

specifies whether the default visual attributes of markers, lines, and fills in nested plot statements automatically change from plot to plot. See “Cycling Through Group Attributes in Overlaid Plots” on page 121 for more information.

Default: FALSE

FALSE

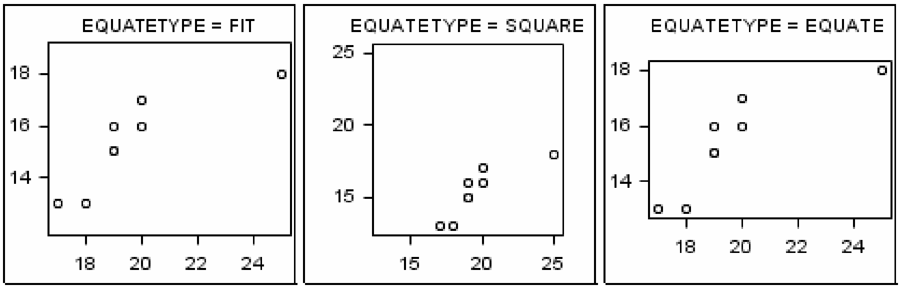
does not cycle the default visual attributes of multiple plots. For example, if you overlay three series plots, each series line has the same default visual properties.

TRUE

the system looks at the plots in the layout. It tries to use the GraphData1 - GraphDataN style elements to assign different visual properties to applicable plots (scatter plots and series plots and others). For example, if you overlay three series plots, the three series lines receive different default visual properties (line pattern and color). These default properties are determined by the next GraphDataN style element not already used. Some plots in the layout do not participate in the cycling (for example, reference lines and drop lines).

EQUATETYPE= FIT | SQUARE | EQUATE

specifies how to draw the axis area.



Default: FIT

FIT

X and Y axes have equal increments between tick values. The data ranges of both axes are compared to establish a common increment size. The axes might be of different lengths and have a different number of tick marks. Each axis represents its own data range. One axis might be extended to use available space in the plot area. If a TICKVALUELIST= or TICKVALUESEQUENCE= axis option is used on COMMONAXISOPTS=, it is ignored.

SQUARE

Both the X and Y axes have the same length and the same major tick values. The axis length and tick values are chosen so that the minimum and maximum of both X and Y appear in the range of values appearing on both axes.

EQUATE

Same as FIT except that neither axis is extended to use available space in the plot area.

OPAQUE= *boolean*
specifies whether the layout background is opaque (TRUE) or transparent (FALSE).

Default: FALSE

Interaction: When this option is set to FALSE, the background color is not used.

PAD=*dimension* | *(pad-options)*
specifies the amount of extra space that is added inside the layout border.

Default: The default padding for all sides is 0. Values without units are in pixels (px). A unit must be provided if other than pixels.

dimension
Specifies a dimension to use for the extra space at the left, right, top, and bottom of the layout border.

(*pad-options*)
Enables separate settings for the left, right, top, and bottom padding dimensions. Use the *pad-options* to create non-uniform padding. These options must be enclosed in parentheses. Each option is specified as a *name* = *value* pair. Sides not assigned padding are padded with the default amount.

Values without units are in pixels (px). A unit must be provided if other than pixels.

LEFT=*dimension*	specifies the amount of extra space added to the left side.
RIGHT=*dimension*	specifies the amount of extra space added to the right side.
TOP=*dimension*	specifies the amount of extra space added to the top.
BOTTOM=*dimension*	specifies the amount of extra space added to the bottom.

WALLCOLOR=*style-reference* | *color*
specifies the fill color of the plot wall area.

Default: The GraphWalls:Color style reference.

style-element
A reference in the form *style-element:style-attribute*. Only the style-attribute named COLOR is used.

Interaction: This option is ignored if WALLDISPLAY=NONE or WALLDISPLAY=(OUTLINE).

WALLDISPLAY= STANDARD | ALL | NONE | *(display-options)*
specifies whether the plot's wall and wall outline are displayed.

Default: STANDARD

STANDARD
Displays a filled wall. The setting of the FRAMEBORDER= ON | OFF attribute of the GraphWalls style element determines whether the wall outline is displayed.

ALL
Displays a filled, outlined wall.

NONE
Displays no wall, no wall outline.

(*display-options*)
A list of options, enclosed in parentheses, that must include one of the following:

OUTLINE	displays the wall outline.
FILL	displays the plot wall.

Use the WALLCOLOR= option to control the fill color of the wall.

The appearance attributes of the wall outline are set by the GraphAxisLine style element.

XAXISOPTS= (*equated-axis-options*)
specifies one or more X axis options. For a list of options, see "Options That Apply Separately to an X or Y Equated Axis" on page 459

Requirement: Axis options must be enclosed in parentheses and separated by spaces.

YAXISOPTS= (*equated-axis-options*)
specifies one or more Y axis options. For a list of options, see "Options That Apply Separately to an X or Y Equated Axis" on page 459

Requirement: Axis options must be enclosed in parentheses and separated by spaces.

Chapter 10

LAYOUT OVERLAY3D Statement

Description and Syntax

Builds a 3-D composite from one or more *GTL-statements*. The composite could be an entire graph. Or, if this layout is nested in a GRIDDED or LATTICE layout, the composite typically provides contents for one cell in the parent layout.

LAYOUT OVERLAY3D *</option(s)>*;

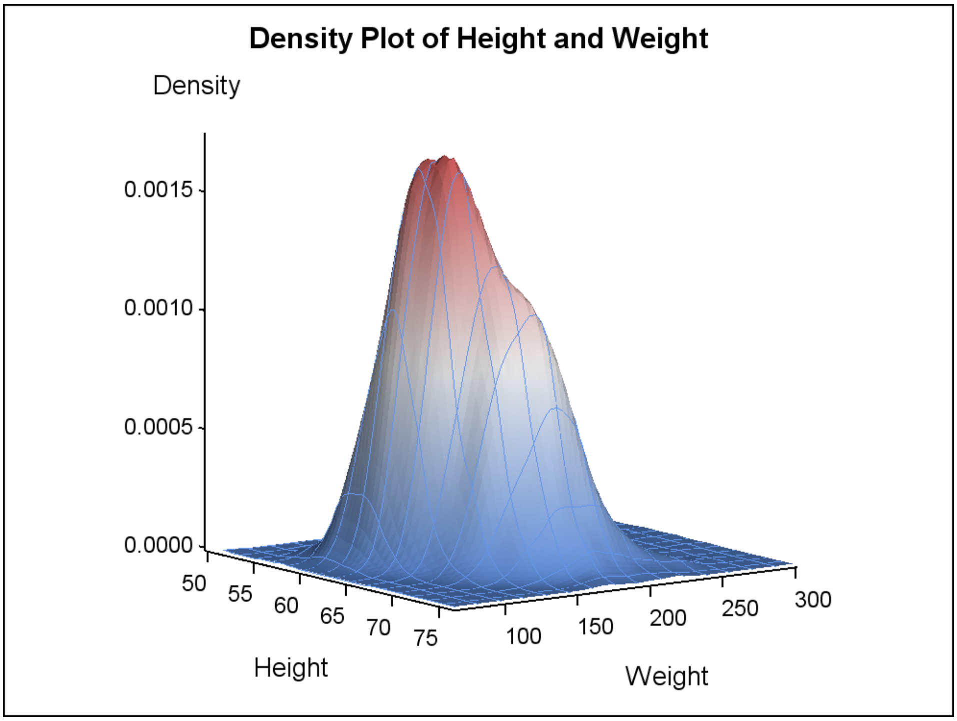
GTL-statements;

ENDLAYOUT;

Example Program and Statement Details

Example Graph

The following graph was generated by the "Example Program" on page 104:



Example Program

```
proc template;
  define statgraph layoutoverlay3d;
    begingraph;
      entrytitle "Density Plot of Height and Weight";
      layout overlay3d / tilt=10  rotate=54
                         walldisplay=none cube=false;
        surfaceplotparm x=height y=weight z=density /
          surfacecolorgradient=density;
      endlayout;
    endgraph;
  end;
run;
proc sgrender data=sashelp.gridded template=layoutoverlay3d;
run;
```

Statement Summary

The LAYOUT OVERLAY3D statement builds a 3-D composite using one or more *GTL-statements*. You can specify one or more 3-D plots within the layout, provided all plots can share the same type of axes. You can also specify “annotations” (for example, with one or more ENTRY statements or LAYOUT GRIDDED statements). However, annotations in the OVERLAY3D layout are more likely to collide with other graphics features than are annotations in other overlay-type layouts.

As an overlay-type layout, OVERLAY3D has the same behavioral characteristics as an OVERLAY layout. It uses the same general logic for rendering the composite (see “LAYOUT OVERLAY Statement” on page 89 for details), and its default axis characteristics are generally determined by the first specified plot, unless you use

PRIMARY=TRUE on another plot statement (see "When Plots Share Data and a Common Axis" on page 397).

Within an OVERLAY3D layout, a graph's bounding cube can be tilted, rotated, and zoomed to provide a different viewpoint. By default, the outline of the bounding cube is displayed and the viewing rotation angle is 57 degrees, the tilt angle is 20 degrees, and the zoom factor is 1. See the CUBE=, ROTATE=, TILT=, and ZOOM= options for information about how to change the viewpoint.

Options

Statement Option	Description
BACKGROUNDCOLOR	Specifies the color of the layout background.
BORDER	Specifies whether a border is drawn around the layout.
BORDERATTRS	Specifies the properties of the border line.
CUBE	Specifies whether the layout displays the lines that indicate the complete bounding cube of the axis planes.
CYCLEATTRS	Specifies whether the default visual attributes of lines and fills in nested plot statements automatically change from plot to plot.
OPAQUE	Specifies whether the layout background is opaque.
PAD	Specifies the amount of extra space that is added inside the layout border.
ROTATE	Specifies the angle of rotation in degrees.
TILT	Specifies the angle of tilt in degrees.
WALLCOLOR	Specifies the fill color of the plot's walls.
WALLDISPLAY	Specifies whether the plot's walls and wall outlines are displayed.
XAXISOPTS	Specifies one or more X axis options.
YAXISOPTS	Specifies one or more Y axis options.
ZAXISOPTS	Specifies one or more Z axis options.
ZOOM	Specifies a zoom factor.

BACKGROUNDCOLOR=*style-reference* | *color*
specifies the color of the layout background.

Default: The GraphBackground:Color style reference.

style-element

A reference in the form *style-element:style-attribute*. Only the style-attribute named COLOR is used.

Interaction: OPAQUE=TRUE must be in effect for the color to be seen. By default, OPAQUE=FALSE.

BORDER=*boolean*

specifies whether a border is drawn around the layout.

Default: FALSE

BORDERATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)

specifies the attributes of the border line around the layout. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Line Options" on page 564 for available *line-options*.

Default: The GraphBorderLines style element.

Interaction: BORDER=TRUE must be set for this option to have any effect.

CUBE=*boolean*

specifies whether the layout displays the lines that indicate the complete bounding cube of the axis planes.

Default: TRUE

The color, thickness, and pattern of the cube lines are determined by the GraphAxisLines style element.

The cube lines are displayed independently of the wall borders and axis lines. Because some cube lines coincide with wall borders and axis lines, it might appear that turning off wall borders or axis lines has no effect when CUBE=TRUE.

CYCLEATTRS=*boolean*

specifies whether the visual attributes of lines and fills in nested plot statements automatically change from plot to plot. See "Cycling Through Group Attributes in Overlaid Plots" on page 121 for more information.

Default: FALSE

FALSE

does not cycle the default visual attributes of multiple plots. For example, if you overlay two surface plots, each surface has the same visual properties.

TRUE

the system looks at the plots in the layout and tries to use the GraphData1 - GraphDataN style elements to assign different visual properties to the plots. For example, if you overlay two surface plots, the two surfaces receive different visual properties. These default properties are determined by the next GraphDataN style element not already used.

OPAQUE= *boolean*

specifies whether the layout background is opaque (TRUE) or transparent (FALSE).

Default: FALSE

Interaction: When this option is set to FALSE, the background color is not used.

PAD=*dimension* | *(pad-options)*

specifies the amount of extra space that is added inside the layout border.

Default: The default padding for all sides is 0. Values without units are in pixels (px). A unit must be provided if other than pixels.

dimension

Specifies a dimension to use for the extra space at the left, right, top, and bottom of the layout border.

(*pad-options*)

Enables separate settings for the left, right, top, and bottom padding dimensions. Use the *pad-options* to create non-uniform padding. These options must be enclosed in parentheses. Each option is specified as a *name* = *value* pair. Sides not assigned padding are padded with the default amount.

Values without units are in pixels (px). A unit must be provided if other than pixels.

LEFT=*dimension*	specifies the amount of extra space added to the left side.
RIGHT=*dimension*	specifies the amount of extra space added to the right side.
TOP=*dimension*	specifies the amount of extra space added to the top.
BOTTOM=*dimension*	specifies the amount of extra space added to the bottom.

ROTATE=*number*

Specifies the angle of rotation. Rotation is measured in a clockwise direction about a virtual axis parallel to the Z axis (vertical) and passing through the center of the bounding cube. A counterclockwise rotation can be specified with a negative value.

Default: 54

TILT=*number*

Specifies the angle of tilt in degrees. Tilt is measured in a clockwise direction about a virtual axis parallel to the X axis (vertical) and passing through the center of the bounding cube. A counterclockwise rotation can be specified with a negative value.

Default: 20

WALLCOLOR=*style-reference* | *color*

specifies the fill color of the plot's walls.

Default: The GraphWalls:Color style reference.

style-reference

A reference in the form *style-element:style-attribute*. Only the style- attribute named COLOR is used.

Interaction: This option is ignored if WALLDISPLAY=NONE or WALLDISPLAY=(OUTLINE).

WALLDISPLAY= STANDARD | ALL | NONE | *(display-options)*

specifies whether the plot's walls and wall outlines are displayed.

Default: STANDARD

STANDARD

Displays an outlined wall. The setting of the FRAMEBORDER= ON | OFF attribute of the GraphWalls style element determines whether the wall outline is displayed.

ALL

Displays a filled, outlined wall.

NONE

Displays no wall fill, no wall outline.

(*display-options*)
A list of options, enclosed in parentheses, that must include one of the following:

OUTLINE	displays the wall outline.
FILL	displays the plot wall.

Use the WALLCOLOR= option to control the color of the wall.

Also see the CUBE= option.

XAXISOPTS= (*axis-options*)
specifies one or more X axis options. For a list of options, see "Axis Options for LAYOUT OVERLAY3D" on page 425.

Requirement: Axis options must be enclosed in parentheses and separated by spaces.

YAXISOPTS= (*axis-options*)
specifies one or more Y axis options. For a list of options, see "Axis Options for LAYOUT OVERLAY3D" on page 425.

Requirement: Axis options must be enclosed in parentheses and separated by spaces.

ZAXISOPTS= (*axis-options*)
specifies one or more Z axis options. For a list of options, see "Axis Options for LAYOUT OVERLAY3D" on page 425.

Requirement: Axis options must be enclosed in parentheses and separated by spaces.

ZOOM= *positive-number*
specifies a zoom factor. Factors greater than 1 move closer to the bounding cube, less than 1 move farther away

Default: 1

Chapter 11

LAYOUT PROTOTYPE Statement

Description and Syntax

Builds a composite from one or more *plot-statements*. The composite is used as a prototype or "rubber stamp" that repeats in each cell of a parent DATALATTICE or DATAPANEL layout.

LAYOUT PROTOTYPE </option(s)>

plot-statements;

ENDLAYOUT;

Statement Details

LAYOUT PROTOTYPE is a Nested Statement

The LAYOUT PROTOTYPE statement is not an independent layout; it must be nested in a LAYOUT DATAPANEL or LAYOUT DATALATTICE block. Only plot statements can be used within the LAYOUT PROTOTYPE block – no nested layouts. Additional restrictions are listed below.

The *plot-statements* determine the graphical content of the cells in the parent layout, based on the subsetting of the specified classification variables. For an example, see “LAYOUT DATALATTICE Statement” on page 33 or “LAYOUT DATAPANEL Statement” on page 49.

A PROTOTYPE layout is essentially a restricted OVERLAY layout with the same general rules for overlaying plots. The main difference is that there are no axis options available on the LAYOUT PROTOTYPE statement. Axis properties are set with the ROWAXISOPTS= and COLUMNAXISOPTS= options of the parent DATAPANEL or DATALATTICE statement.

The following restrictions also apply to a LAYOUT PROTOTYPE block:

- Only non-computed 2-D plots can be included in the LAYOUT PROTOTYPE block. Thus, plots like BOXPLOT, HISTOGRAM, DENSITYPLOT, ELLIPSE, LOESSPLOT, MODELBAND, and REGRESSIONPLOT cannot be used. In addition, the prototype cannot contain a SCATTERPLOTMATRIX or 3-D plot.
- A plot statement cannot be used if it contains a column defined with an EVAL expression.
- You can add one or more 2-D plots to the graph area that the LAYOUT PROTOTYPE statement creates, provided all of the graphs can share the same axis type.
- ENTRY, DISCRETELEGEND, and CONTINUOUSLEGEND statements cannot be used in the prototype layout.
- If you include a plot statement with a CURVELABEL= option (such as SERIESPLOT), only CURVELABELLOCATION=INSIDE is supported.
- If you include a plot statement that supports a CLIP= option (such as LINEPARM or ELLIPSEPARM), the CLIP value is always set to TRUE.
- If any plot statement in the *plot-statements* contains a YAXIS=Y2 or XAXIS=X2 option, this option is ignored and all data are mapped to the X and Y axes.

Options

Statement Option	Description
CYCLEATTRS	Specifies whether the default visual attributes of markers, lines, and fills in nested plot statements automatically change from plot to plot.
WALLCOLOR	Specifies the fill color of the plot wall area.
WALLDISPLAY	Specifies whether the plot's wall and wall outline are displayed.

CYCLEATTRS=*boolean*

specifies whether the default visual attributes of markers, lines, and fills in nested plot statements automatically change from plot to plot. See "Cycling Through Group Attributes in Overlaid Plots" on page 121 for more information.

Default: FALSE

FALSE

does not cycle the default visual attributes of multiple plots. For example, if you overlay three series plots, each series line has the same default visual properties.

TRUE

the system looks at the plots in the layout. It tries to use the GraphData1 - GraphDataN style elements to assign different visual properties to applicable plots (scatter plots and series plots and others). For example, if you overlay three series plots, the three series lines receive different default visual properties (line pattern and color). These default properties are determined by the next GraphDataN style element not already used. Some plots in the layout do not participate in the cycling (for example, reference lines and drop lines).

WALLCOLOR=*style-reference* | *color*

specifies the fill color of the plot wall area.

Default: The GraphWalls:Color style reference.

style-reference
A reference in the form *style-element:style-attribute*. Only the style- attribute named COLOR is used.

Interaction: This option is ignored if WALLDISPLAY=NONE or WALLDISPLAY=(OUTLINE).

WALLDISPLAY=STANDARD | ALL | NONE | *(display-options)*
specifies whether the plot wall and wall outline are displayed.

Default: STANDARD

STANDARD
Displays a filled wall. The setting of the FRAMEBORDER= ON | OFF attribute of the GraphWalls style element determines whether the wall outline is displayed.

ALL
Displays a filled, outlined wall.

NONE
Displays no wall, no wall outline.

(*display-options*)
These options must be enclosed in parentheses and include one of the following:

OUTLINE	displays the wall outline.
FILL	displays a filled wall area.

Use the WALLCOLOR= option to control the color of the wall.

The appearance attributes of the wall outline are set by the GraphAxisLine style element.

Part 4

Plot Statements

Chapter 12
Key Concepts for Using Plots

Minimum Requirements to Generate a Plot

ODS graphics are generated by template definitions that determine a graph's layout and appearance and specify the variable roles to be represented in the graph display. A graph can be rendered from a compiled template by associating the template with a data source at run time.

The following SAS program shows the basic structure needed to meet the minimum requirements for generating a plot using GTL:

```
proc template;
  define statgraph minimumreq;
    begingraph;
      layout overlay;
        scatterplot x=weight y=height;
      endlayout;
    endgraph;
  end;
run;

proc sgrender data=sashelp.class template=minimumreq;
run;
```

- The DEFINE STATGRAPH statement on PROC TEMPLATE is required to open a definition block for defining and naming a graphics template. The END statement closes the template definition.
- A BEGINGRAPH statement block is required to define the outermost container for the graph. The ENDGRAPH statement closes the block.
- At least one layout statement block is required for specifying the elements that compose the graph. To generate a plot, the layout block must contain at least one plot statement. The ENDLAYOUT statement closes the layout block.
- The PROC TEMPLATE statement must be run to compile the template and save it in the template store (SASUSER.TEMPLAT by default).
- The PROC SGRENDER statement is required to produce a graph from a compiled template. The DATA= option specifies a run-time data source to use, and the TEMPLATE= option specifies the template to use. The input data source must satisfy any restrictions that are imposed by the template. For example, it must contain any variables that have been specified on the template's GTL statements.

ODS Graphics Environment

The ODS GRAPHICS statement manages the settings of the ODS Graphics environment and is a statement you will probably use frequently in your SAS sessions. For example, the ODS GRAPHICS statement provides options that control the physical aspects of your graphs, such as the image size and the name of the image file that is created for the graph.

The default image size of 640 pixels by 480 pixels (4:3 aspect ratio) for ODS Graphics is set in the SAS Registry. You can change the image size using the WIDTH= option, or the HEIGHT= option, or both in the ODS GRAPHICS statement. To name the output image file, use the IMAGENAME= option.

The following ODS GRAPHICS statement sets a 320 pixel width for the graph and names the output image file *modelfit*:

```
ods graphics / width=320px
               imagename="modelfit" reset;

proc sgrender data=sashelp.class template=modelfit;
run;

ods graphics off;
```

- The WIDTH= option sets the image width to 320 pixels. Because no HEIGHT= option is used, SAS uses the design aspect ratio of the graph to compute the appropriate height. (The width of 320px is half the default width, so SAS sets the height to 240px, which is half the default height.)
- The IMAGENAME= option sets the name of the output image file to *modelfit*. The RESET option ensures that each time the graph is produced, the previous version of the image file is replaced. Otherwise, image names are incremented (modelfit1, modelfit2, and so on) every time the graph is produced.

In general, it is good practice to specify only one sizing option without the other—just the WIDTH= option or just the HEIGHT= option. That way SAS can maintain the design aspect ratio of the graph, which might be important for many graphs. For example, a graph that has multiple columns or a statistics table on the side needs a wide aspect ratio. Specifying both width and height in such cases might produce unpredictable results.

Note: Size settings in the ODS GRAPHICS statement affect all of the graphs that are rendered in the SAS session, unless they are changed by another ODS GRAPHICS statement. The size for a graph produced by an individual template can be set with the DESIGNWIDTH= and DESIGNHEIGHT= options in the BEGINGRAPH statement. Size settings in the ODS GRAPHICS statement override size settings in the BEGINGRAPH statement and remain in effect unless they are changed on another ODS GRAPHICS statement or ODS GRAPHICS are turned off.

For more information about using the ODS GRAPHICS statement in GTL, see *SAS/GRAPH: Graph Template Language User's Guide*. For a more complete discussion of the ODS GRAPHICS statement, see *SAS Output Delivery System: Procedures Guide* .

Display Attributes

Overview

The display attributes for the lines, colors, marker symbols, and text used in a graph are derived from the ODS style that is in effect when the graph is produced. These display attributes might also be influenced by grouped data. To override default display attributes, all GTL plot statements provide options that manage the graph’s visual appearance. For example, a BOXPLOT statement provides an OUTLIERATTRS= option that manages the visual appearance of outliers.

Two ways are generally available for modifying a graph’s display attributes:

- Change the ODS style that is in effect for the graph. “ODS Styles” on page 16 provides an overview of the use of styles in a graph, and the usage guide discusses the use of styles in more detail.
- Override default style settings using GTL statement options. Some examples are given in the sections that follow.

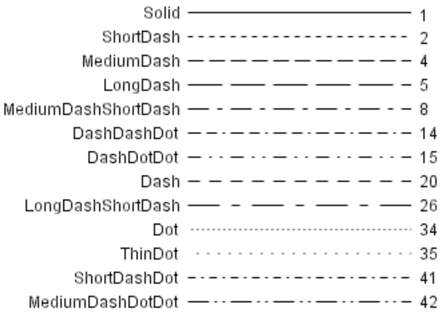
Display Attributes for Non-Grouped Data

“Display Attributes” on page 563 documents the attribute settings that can be specified for the lines, data markers, text, or area fills in a plot. The defaults for these attributes are defined on style elements, but you can use attribute options on the plot statement to change the defaults.

For example, the LINEPARM statement provides a LINEATTRS= option that specifies the color, line pattern, or line thickness of the plot line. For non-grouped data, if you do not set a line pattern in your template, then the default line pattern for the plot is obtained from the GraphDataDefault:LineStyle style reference.

To change the default line pattern, a PATTERN= suboption on LINEATTRS= is available. Display 12.1 on page 118 shows the most common line patterns available for the PATTERN= suboption.

Display 12.1 *"Common Line Patterns"*



Solid	1
ShortDash	2
MediumDash	4
LongDash	5
MediumDashShortDash	8
DashDashDot	14
DashDotDot	15
Dash	20
LongDashShortDash	26
Dot	34
ThinDot	35
ShortDashDot	41
MediumDashDotDot	42

- the left column shows the names for the line patterns
- the center column illustrates the type of line the name requests
- the right column shows the SAS/GRAPH line-style numbers for the line patterns

"Available Line Patterns" on page 566 provides the complete list of line patterns that can be used with the GTL.

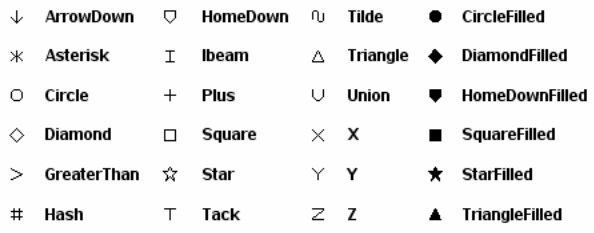
In the following template definition, the LINEPARM statement's LINEATTRS= option overrides the default line pattern by specifying PATTERN=DASH:

```
proc template;
  define statgraph patternchange;
    begingraph;
      layout overlay;
        scatterplot y=height x=weight;
        lineparm yintercept=intercept  slope=slope /
          lineattrs=(pattern=dash);
      endlayout;
    endgraph;
  end;
```

Other display options can be managed the same way. For example, the SCATTERPLOT statement provides a MARKERATTRS= option that specifies the color, size, symbol, and weight of the plot data markers. For non-grouped data, if you do not set a marker symbol in your template, then the default marker symbol is obtained from the GraphDataDefault:MarkerSymbol style reference.

To change the default marker symbol, a SYMBOL= suboption on MARKERATTRS= is available. Display 12.2 on page 118 shows the marker symbols available for the SYMBOL= suboption.

Display 12.2 *"Marker Symbols"*



↓	ArrowDown	▽	HomeDown	∿	Tilde	●	CircleFilled
⁎	Asterisk	I	Ibeam	△	Triangle	◆	DiamondFilled
○	Circle	+	Plus	∪	Union	▼	HomeDownFilled
◇	Diamond	□	Square	×	X	■	SquareFilled
>	GreaterThan	☆	Star	Y	Y	★	StarFilled
#	Hash	⊤	Tack	Z	Z	▲	TriangleFilled

In the following template definition, the SCATTERPLOT statement's MARKERATTRS= option overrides the default marker symbol by specifying SYMBOL=CIRCLEFILLED, which uses a filled circle to represent the data points.

```
proc template;
  define statgraph symbolchange;
    begingraph;
      layout overlay;
        scatterplot y=height x=weight /
          markerattrs=(symbol=circlefilled);
      endlayout;
    endgraph;
  end;
```

Display Attributes for Grouped Data

"Display Attributes" on page 563 documents the attribute settings that you can specify for the lines, data markers, text, or area fills in a plot. For grouped data (when you use the GROUP= option on a plot statement), each distinct group value can be represented in the graph by a different combination of line pattern, color, and marker symbol (depending on the graph type). The defaults for these features are set by the LineStyle, Color, ContrastColor, and MarkerSymbol attributes of the GraphData1 - GraphDataN style elements.

Display 12.1 on page 118 shows the common line patterns available, and Display 12.2 on page 118 shows the marker symbols available.

For grouped plots, the style in effect and the plot settings determine which line patterns, area fills, and plot symbols are used. If different line patterns, colors, and/or marker symbols are used to represent group values, then the style determines the sequences or each that are used for the group values (as discussed in "Cycling Through Group Attributes in Overlaid Plots" on page 121, other plot settings might also influence the sequence). The sequence is repeated as many times as needed to provide a line pattern, color, and/or marker symbol for each group value.

You can use attribute options on the plot statement to change the default display attributes used for group data. For example, in the following template definition, the LINEPARM statement's LINEATTRS= option specifies PATTERN=DASH. This explicit setting overrides the default line rotation for the plot lines and uses dashed lines for all of the plots, leaving color to distinguish among group values.

```
proc template;
  define statgraph dashedline;
    begingraph;
      layout overlay;
        scatterplot y=height x=weight / group=gender;
        lineparm yintercept=intercept  slope=slope / group=gender
          lineattrs=(pattern=dash);
      endlayout;
    endgraph;
  end;
```

Rather than setting the same line pattern on all group values, you can change the default sequence of line patterns that is used for grouped values. To do so, set the LineStyle attribute in each of the style elements GraphData1 through GraphDataN.

In the following example, a style is defined to change the default line pattern for the first two lines in the pattern sequence. The style is derived from the DEFAULT style (the default style for the ODS HTML destination). Values are set for the LineStyle attributes in the

GraphData1 and GraphData2 style elements. The first default line in the sequence has long dashes (style value 6) and the second line has short dashes (style value 4). The LineStyle settings for the remaining GraphData elements are not set and so are derived from the parent style (DEFAULT). This new line sequence is used as the default line sequence for any plot that uses the MyDefault style.

```
proc template;
  define style Styles.MyDefault;
    parent=Styles.Default;

    style GraphData1 /
      LineStyle=6;
    style GraphData2 /
      LineStyle=4;
  end;

  define statgraph testPattern;
    begingraph;
      layout overlay;
        scatterplot y=height x=weight / group=gender;
        lineparm yintercept=intercept  slope=slope / group=gender
          lineattrs=(pattern=MyDefault);
      endlayout;
    endgraph;
  end;
```

Similarly for grouped data, you can set the MarkerSymbol attribute in each of the style elements GraphData1 through GraphDataN. In the following example, a style is defined to change the default sequence that is used for the first three marker symbols in grouped plots. Values are set for the MarkerSymbol attributes in the GraphData1 through GraphData3 style elements. This new sequence is used as the default marker symbol sequence for any plot that uses the MyDefault style.

```
proc template;
  define style Styles.MyDefault;
    parent=Styles.Default;

    style GraphData1 /
      MarkerSymbol=DIAMOND;
    style GraphData2 /
      MarkerSymbol=CROSS;
    style GraphData3 /
      MarkerSymbol=CIRCLE;
  end;

  define statgraph testSymbols;
    begingraph;
      layout Overlay;
        scatterPlot y=height x=weight / group=age
          markerattrs=(symbol=MyDefault);
      endlayout;
    endgraph;
  end;
```

Cycling Through Group Attributes in Overlaid Plots

Overlay-type layouts provide the CYCLEATTRS= options that specifies whether the default visual attributes of lines, marker symbols, and area fills in nested plot statements automatically change from plot to plot. When CYCLEATTRS=TRUE, all applicable plot statements (SCATTERPLOT, SERIESPLOT, and others) are sequentially assigned the next unused GraphDataN style element. (The sequence is overridden for plot statements that have an explicit setting, either through a style element assignment or option settings.) No plot retains its default (implicit) style element.

In the following example, assuming ungrouped data, the series plots are assigned line properties based on the GraphData1, GraphData2, and GraphData3 style elements. The reference line uses GraphReference, not GraphData4.

```
layout overlay / cycleattrs=true;
  seriesplot x=date y=var1;
  seriesplot x=date y=var2;
  seriesplot x=date y=var3;
  referenceline x=cutoff / lineattrs=GraphReference;
endlayout;
```

If one of the plots in this example uses grouped data, the grouped plots also participate in the default cycles. For example, if the second plot has three groups, it generates three plots, which are assigned line properties based on the GraphData2, GraphData3, and GraphData4 style elements.

If the plot statement that uses grouped data also uses the INDEX= option to manage the group values (see "Remapping Groups for Grouped Data" on page 121), the INDEX= option overrides the default behavior. In that case, the grouped plots do not participate in the default cycling.

When one or more of the plots within the layout override the default cycling behavior, the arrangement of the plots within the layout might affect the default mapping of the GraphDataN elements to those statements that participate in the default cycling.

Remapping Groups for Grouped Data

Indexing can be used to collapse the number of groups that are represented in a graph. For example, if 10 groups are in the data, indexes 1 and 2 can be assigned to the first two groups, and index 3 can be assigned to all other groups. The third through tenth data groups are treated as a single group in the graph.

Indexing can control the order in which colors, area fills, marker symbols, and line styles are mapped to group values in a graph. This ordering method is needed only for coordinating the data display of multiple graphs when the default mapping would cause group values to be mismatched between graphs.

For example, consider two studies of three drugs, A, B, and C. If Study 1 uses all three drugs, then the first combination of color and marker symbol is mapped to Drug A. The second combination of color and marker symbol is mapped to Drug B, and the third is mapped to Drug C. If Study 2 omits Drug A, then the first combination of color and marker symbol is mapped to Drug B, and the second is mapped to Drug C. If the two graphs are viewed together, then this default mapping causes the group values to be mismatched. The visual attributes that represent Drug A in the first graph represent Drug B in the second graph. Those that represent Drug B in the first graph represent Drug C in the second group.

The GROUP= option mappings can be made consistent between the two graphs by creating an index column for each study. For these example studies, the GROUP and INDEX columns are the following:

Table 12.1 *Study 1*

Drug1	**Index1**
A	1
B	2
B	2
C	3

Table 12.2 *Study 2*

Drug2	**Index2**
B	2
C	3
C	3

If the graph for Study 1 specifies INDEX=INDEX1 and the graph for Study 2 specifies INDEX=INDEX2, then the second combination of color and marker symbol is mapped to Drug B in both graphs. The third combination of color and marker symbol is mapped to Drug C in both graphs.

Interactions Between Options

When you use GTL statement options to manage the graph display, interactions between options might cause some option settings to be ignored. For example, an ENTRYTITLE statement provides BORDER= and BORDERATTRS= options for managing a border line around the graph title. Border attributes that are set on the BORDERATTRS= option have no effect on the graph title unless the title border line is displayed by setting BORDER=TRUE.

Similarly, if a BOXPLOT statement's DISPLAY= option suppresses the display of outliers in a box plot, then using the OUTLIERATTRS= option to set outlier attributes has no effect. The OUTLIERATTRS= settings only take effect if DISPLAY= enables the display of outliers.

The option interactions are not limited to options that simply manage visual elements. For example, on a BOXPLOT, if the EXTREME= option extends the box whiskers beyond the fences, then outliers are suppressed in the plot and options that affect the outliers are ignored, if set.

The documentation for each GTL statement identifies the option interactions that might occur on that statement.

Location and Position of Curve Labels

Overview

On plots that generate a curve line (a series plot or a density plot, for example), you can specify a label for the curve line. You can also determine the label's location in the graph. For example, the SERIESPLOT statement provides the following options for managing a curve label:

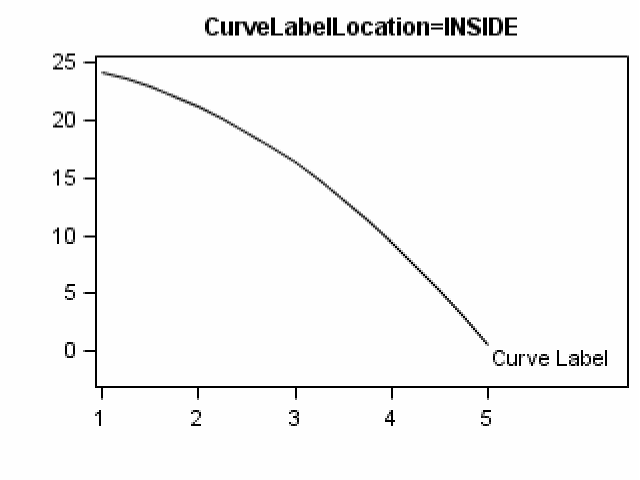
CURVELABEL
Specifies a label for the curve line.

CURVELABELLOCATION
Specifies the location of the curve line label relative to the plot area.

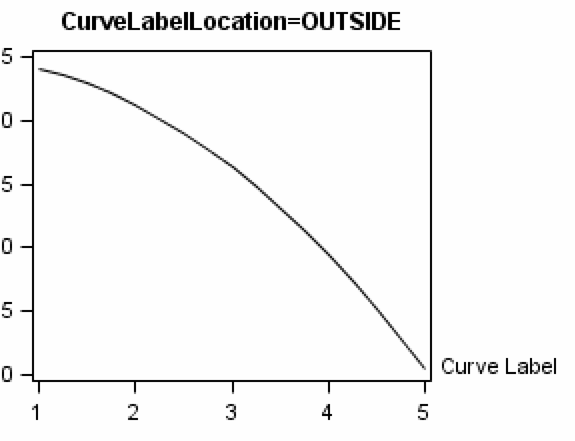
CURVELABELPOSITION
Specifies the position of the label relative to the curve line.

Curve Label Location Relative to the Plot Area

By default, the label for a curve line is displayed inside of the plot area. The following figure shows the default location of the label for a series plot labeled "Curve Label":



Depending on the shape of the curve line, its distribution of values, and the other plot elements that must be displayed within the plot area, GTL might have to add an offset (see "Adjusting Axis Offsets" on page 403) to one of the plot's axis lines to provide enough room for the curve label. To prevent the offset of the axis line, you can move the curve label outside of the plot area by specifying CURVELABELLOCATION=OUTSIDE on the plot statement:



Regardless of whether the curve label is displayed inside or outside of the plot area, you can use the CURVELABELPOSITION= option to adjust the label's position relative to the curve line.

Curve Label Position Relative to the Curve Line

Given a curve label's location inside or outside of the plot area, a plot statement's CURVELABELPOSITION= option can adjust the label's position relative to the curve line. For example, the following positions are available for a series plot (for some plots, START and END are not available):

AUTO
Only used when CURVELABELLOCATION=OUTSIDE. The curve label is positioned automatically near the end series line along unused axes whenever possible (typically Y2 or X2) to avoid collision with tick values.

MAX
Forces the curve label to appear near maximum series values (typically, to the right).

MIN
Forces the curve label to appear near minimum series values (typically, to the left).

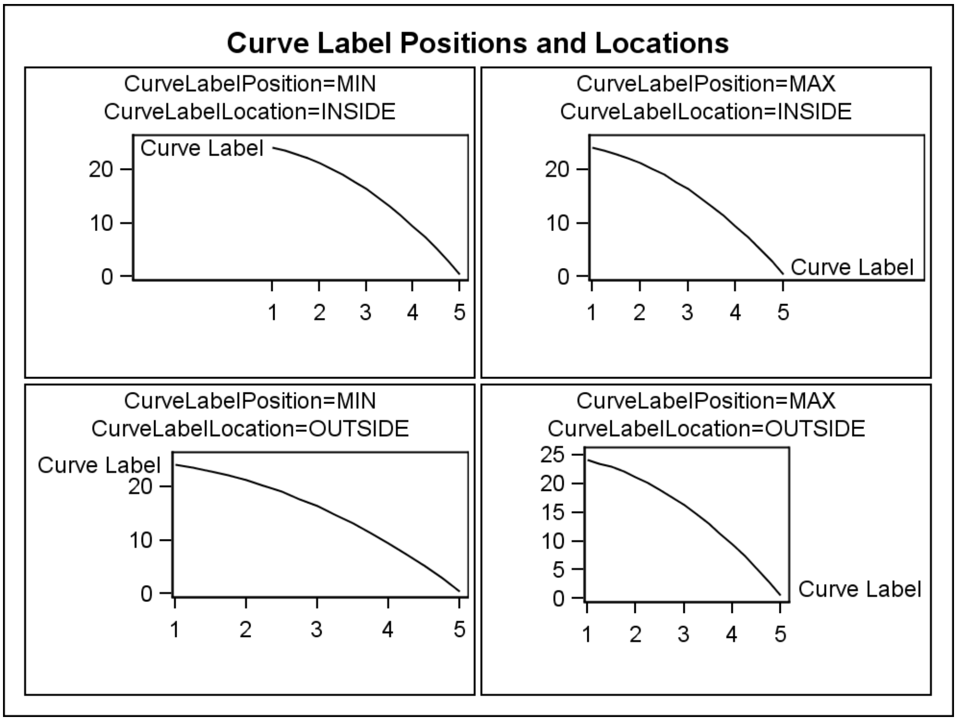
START
Only used when CURVELABELLOCATION=INSIDE. Forces the curve label to appear near the beginning of the curve. Particularly useful when the curve line has a spiral shape.

END
Only used when CURVELABELLOCATION=INSIDE. Forces the curve label to appear near the end of the curve Particularly useful when the curve line has a spiral shape.

When CURVELABELLOCATION=INSIDE, you can choose whether to position the curve label near the START or END of the curve, or near the minimum data values (MIN) or maximum data values (MAX). START and END use a different algorithm than MIN and MAX. They are particularly useful for spiral-shaped curves whose end points do not correlate with the minimum and maximum data values. In those cases, START or END provide "better" label locations than MIN and MAX.

When CURVELABELLOCATION=OUTSIDE and CURVELABELPOSITION=AUTO, a "good" position is automatically chosen to avoid collision with the axis information.

The following figure shows the different combinations of label locations and positions:



- The minimum or maximum axis tick marks can be adjusted (see "Adjusting Axis Offsets" on page 403) so that the label can be placed inside the plot area. Increasing label length decreases the area available for displaying plots.
- When CURVLABELLOCATION=OUTSIDE, you can set the CURVELABELPOSITION to MIN or MAX, but the label might collide with the axis ticks and tick values, unless you are aware of where the axes are positioned.

Chapter 13
BANDPLOT Statement

Description and Syntax

Creates a band plot that typically shows confidence or prediction limits.

BANDPLOT X = *column* | *expression*
LIMITLOWER = *number* | *numeric-column* | *expression*
LIMITUPPER = *number* | *numeric-column* | *expression* </*option(s)*>;

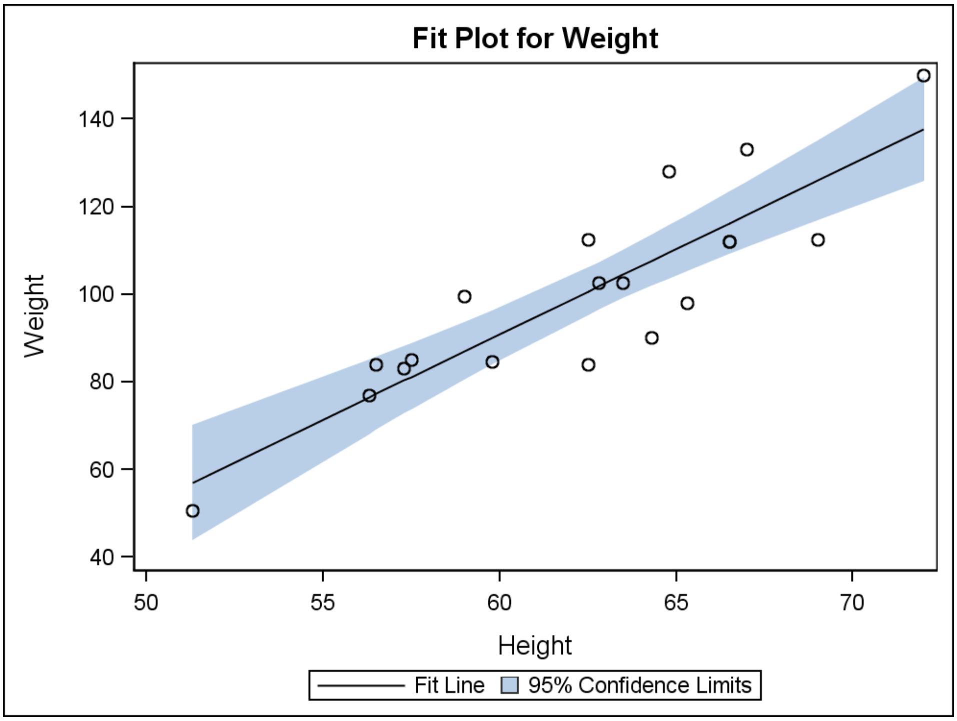
BANDPLOT Y = *numeric-column* | *expression*
LIMITLOWER = *number* | *numeric-column* | *expression*
LIMITUPPER = *number* | *numeric-column* | *expression* </option(s)>;

Example Program and Statement Details

The following sections provide a basic overview of the BANDPLOT statement.

Example Graph

The following graph was generated by the "Example Program" on page 128:



Example Program

```
proc template;
  define statgraph bandplot;
    begingraph;
      entrytitle "Fit Plot for Weight";
      layout overlay;
        bandplot x=height limitupper=uppermean
          limitlower=lowermean /
          name="band" modelname="fit"
          legendlabel="95% Confidence Limits";
        scatterplot x=height y=weight / primary=true;
        seriesplot x=height y=predict / name="fit"
          legendlabel="Fit Line";
        discretelegend  "fit" "band";
      endlayout;
    endgraph;
  end;
run;

proc sgrender data=sashelp.classfit template=bandplot;
run;
```

Statement Summary

A band plot can specify an X column with Y upper and lower limits, or a Y column with X upper and lower limits. If you specify the X argument, you must specify LIMITLOWER and LIMITUPPER arguments for the Y values to apply the limits to the Y axis. If you specify the Y argument, you must specify LIMITLOWER and LIMITUPPER arguments for the X values to apply the limits to the X axis.

When you use a BANDPLOT statement to display prediction or confidence limits, the band plot can be used with another plot that specifies a fitted model. For example, it can be used with a series or step plot. In these cases, use the BANDPLOT option MODELNAME= or TYPE= to identify the interpolation for the band.

You can use the BANDPLOT statement in displays that are independent of other plots. For example, a band plot can be used to define yellow and green areas in an OVERLAY LAYOUT statement that also contains a scatter plot. This use implies concern for any of the scatter plot values that fall in the yellow area and comfort for any values that fall in the green area. For this use, the upper and lower limits would be specified by a constant.

Arguments

Overview

Either an X= or Y= argument must be specified. Only one of them can be used. In addition, the LIMITLOWER= and LIMITUPPER= arguments must be used to specify the lower and upper lines for the band.

Argument Descriptions When X= Is Used

X=*column* | *expression*
specifies X values. Numeric or character values can be used.

LIMITLOWER=*number* | *numeric-column* | *expression*
specifies a constant or column representing the Y value(s) of the lower band line. If a constant is specified, a straight line is drawn.

LIMITUPPER=*number* | *numeric-column* | *expression*
specifies a constant or column representing the Y value(s) of the upper band line. If a constant is specified, a straight line is drawn.

Argument Descriptions When Y= Is Used

Y=*column* | *expression*
specifies Y values. Numeric or character values can be used.

LIMITLOWER=*number* | *numeric-column* | *expression*
specifies a constant or column representing the X value(s) of the lower band line. If a constant is specified, a straight line is drawn.

LIMITUPPER=*number* | *numeric-column* | *expression*
specifies a constant or column representing the X value(s) of the upper band line. If a constant is specified, a straight line is drawn.

Options

Statement Option	Description
CONNECTORDER	Specifies how to connect the data points to form the band lines.
CURVELABELATTRS	Specifies the color and font attributes of the upper and lower band labels.
CURVELABELLOWER	Specifies a label for the lower band limit.

Statement Option	Description
CURVELABELUPPER	Specifies a label for the upper band limit.
CURVELABELLOCATION	Specifies the location of the band labels relative to the plot area.
CURVELABELPOSITION	Specifies the position of the band labels relative to the curve line.
DATATRANSPARENCY	Specifies the degree of the transparency of the band.
DISPLAY	Specifies whether to display an outlined band area, a filled band area, or an outlined and filled band area.
EXTEND	Specifies whether a constant band is to be drawn to the area bounded by the axes.
FILLATTRS	Specifies the appearance of the band area.
GROUP	Creates a separate band plot for each unique group value of the specified column.
INDEX	Specifies indices for mapping line attributes (color and pattern) to one of the GraphData1 - GranphDataN style elements.
JUSTIFY	Specifies the location of the data point relative to the step when TYPE=STEP
LEGENDLABEL	Specifies the label for use in a legend.
MODELNAME	Specifies the name of the plot from which to derive the interpolation of the band.
NAME	Assigns a name to a plot statement for reference in other template statements.
OUTLINEATTRS	Specifies the properties of the band lines.
ROLENAME	Specifies roles for information defined by data columns.
TIP	Specifies the information to display when the cursor is positioned over a graph element.
TIPFORMAT	Specifies display formats for information defined by roles.
TIPLABEL	Specifies display labels for information defined by roles.
TYPE	Specifies how the data points of the lower and upper limits are joined.

Statement Option	Description
XAXIS	Specifies whether data are mapped to the primary X (bottom) axis or the secondary X2 (top) axis.
YAXIS	Specifies whether data are mapped to the primary Y (left) axis or the secondary Y2 (right) axis.

CONNECTORDER=VALUES | AXIS
specifies how to connect the data points to form the band lines.

Default: VALUES

VALUES
Connects data points in the order read from the X variable (or Y variable).

AXIS
Connects data points as they occur left-to-right along the X axis (or bottom-to-top along the Y axis).

Use CONNECTORDER= to assure the expected connect order for certain types of series lines (for example, time series) when the input data might not be sorted by the X variable (or Y variable).

CURVELABELATTRS=*style-element* | *style-element* (*text-options*) | (*text-options*)
specifies the color and font attributes of the band labels. See “General Syntax for Attribute Options” on page 563 for the syntax on using a *style-element* and “Text Options” on page 566 for available *text-options*.

Default:

- For non-grouped data, the GraphValueText style element.
- For grouped data, text color is derived from the GraphData1:ContrastColor - GraphDataN:ContrastColor style references. The font is derived from the GraphValueText style element.

CURVELABELLOWER= *"string"* | *column*
specifies a label for the lower band limit.

Default: no default

Interaction: For non-grouped data, use "string".

Interaction: For grouped data, use a column to define the lower band labels for each group value. All of the labels for a specific group value must be the same. Otherwise, the results are unpredictable.

The font and color attributes for the label are specified by the CURVELABELATTRS= option.

CURVELABELUPPER= *"string"* | *column*
specifies a label for the upper band limit.

Default: no default

Interaction: For non-grouped data, use "*string*".

Interaction: For grouped data, use a column to define the upper band labels for each group value. All of the labels for a specific group value must be the same. Otherwise, the results are unpredictable.

The font and color attributes for the label are specified by the CURVELABELATTRS= option.

CURVELABELLOCATION= INSIDE | OUTSIDE
specifies the location of the band labels relative to the plot area.

Default: INSIDE

INSIDE
inside the plot area

OUTSIDE
outside the plot area

Restriction: OUTSIDE cannot be used when the BANDPLOT is used in multicell layouts such as LATTICE, DATAPANEL, or DATALATTICE where axes might be external to the grid.

Interaction: This option is used in conjunction with the CURVELABELPOSITION= option to determine where the band labels appear. For more information, see "Location and Position of Curve Labels" on page 123.

CURVELABELPOSITION=AUTO | MAX | MIN |START | END
Specifies the position of the band labels relative to the curve line.

Default: AUTO when CUVELABELLOCATION=OUTSIDE. END when CURVELABELLOCATION=INSIDE.

AUTO
Only used when CURVELABELLOCATION=OUTSIDE. Band labels are positioned automatically near the band boundary along unused axes whenever possible (typically Y2 and X2).

MAX
Forces the band labels to appear near maximum band values (maximum-values for horizontal curves, and maximum Y-values for vertical curves).

MIN
Forces the band label to appear near minimum band values (minimum X-values for horizontal curves, and minimum Y-values for vertical curves)

START
Only used when CURVELABELLOCATION=INSIDE. Forces band labels to appear near the beginning of the curve. Particularly useful when the curve line has a spiral shape.

END
Only used when CURVELABELLOCATION=INSIDE. Forces band labels to appear near the end of the curve. Particularly useful when the curve line has a spiral shape.

Restriction: The AUTO setting is not supported if CURVELABELLOCATION=INSIDE is specified. The START and END settings are not supported if CURVELABELLOCATION=OUTSIDE is specified.

Interaction: This option is used in conjunction with the CURVELABELLOCATION= option to determine where the band labels appear. For more information, see "Location and Position of Curve Labels" on page 123.

DATATRANSPARENCY=*number*
specifies the degree of the transparency of the band. This option affects both the fill and outline.

Default: 0

Range: 0 (opaque) to 1 (entirely transparent)

DISPLAY=STANDARD | ALL | (*display-options*)

specifies whether to display an outlined band area, a filled band area, or an outlined and filled band area.

Default: The value of the DisplayOpts attribute of the GraphBand style element, which is DisplayOpts="FILL" by default.

STANDARD

displays filled band with no outline

ALL

displays an outlined, filled band

(*display-options*)

a list of space-delimited options, enclosed in parentheses, that must include one of the following:

OUTLINE—displays an outlined band

FILL—displays a filled band

Use the OUTLINEATTRS= and FILLATTRS= options to control the appearance of the band.

EXTEND=*boolean*

specifies whether the constant or "step" band is to be drawn to the area bounded by the axes.

Default: FALSE

Requirement: When this option is used for a constant band, constants must be specified for the upper and lower band limits. This requirement does not apply to "step" bands.

Restriction: This option is ignored if the X or Y value is character.

If this option is not specified, then there can be a small gap between the line and the axis. The gap is controlled by the axis offset. If the offset is set to 0, then there is no gap.

FILLATTRS=*style-element* | *style-element* (*fill-options*) | (*fill-options*)

specifies the appearance of the filled band area. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Fill Options" on page 564 for available *fill-options*.

Default:

- For non-grouped data, the GraphDataDefault:Color style reference.
- For grouped data, the Color attribute of GraphData1 - GraphDataN style elements.

Interaction: For this option to have any effect, the fill must be enabled by the ODS style or the DISPLAY= option.

GROUP=*column* | *expression*

creates a separate band plot for each unique group value of the specified column.

Default:

- If the band outline is enabled by the ODS style or the DISPLAY= option, each distinct group value is represented in the graph by a different combination of outline color (defined by the GraphData1:ContrastColor - GraphDataN:ContrastColor style references) and outline pattern (defined by the attribute of the GraphData1:LineStyle - GraphDataN:LineStyle style references).

- If the band fill is enabled by the ODS style or the DISPLAY= option, each distinct group value is represented in the graph by a different fill color (defined by the GraphData1:Color - GraphDataN:Color style references).

Restriction: This option can be used only when a numeric column is specified for the upper limit or the lower limit of the band plot. The other limit could be a constant, if desired.

Interaction: To label grouped band plots, you must specify CURVELABELLOWER==*column* and CURVELABELUPPER==*column*.

Interaction: The group values are mapped in the order of the data, unless the INDEX= option is used to alter the default sequence of colors and line patterns.

The representations that are used to identify the groups can be overridden individually. For example, each distinct group value is represented by a different line pattern for the band lines, but the PATTERN= suboption of the OUTLINEATTRS= option could be used to assign the same line pattern to all band outlines.

INDEX=*positive-integer-column* | *expression*
specifies indices for mapping line attributes (color and pattern) to one of the GraphData1 - GranphDataN style elements.

Default: no default

Restriction: For this option to take effect, the GROUP= option must also be specified.

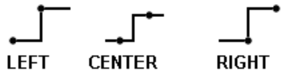
Interaction: All of the indexes for a specific group value must be the same. Otherwise, the results are unpredictable.

Interaction: The index values are 1-based indices. For the style definitions in GraphData1 - GraphDataN, if the index value is greater than N, then a modulo operation remaps that index value to a number less than N to determine which style to use.

Interaction: If this option is not used, then the group values are mapped in the order of the data.

Discussion: Indexing can be used to collapse the number of groups that are represented in a graph. For more information, see "Remapping Groups for Grouped Data" on page 121.

JUSTIFY=LEFT | CENTER | RIGHT
specifies the location of the data point relative to the step.



Default: LEFT

Requirement: TYPE= must also be specified for this option to have any effect.

Interaction: If the MODELNAME= option is specified, this option is ignored.

LEGENDLABEL= *"string"*
specifies the label of the legends.

Default: The *string* specified on the NAME= option.

Interaction: If the GROUP= option is specified, this option is ignored.

MODELNAME= *"plot-name"*
specifies the name of the plot from which to derive the interpolation for the band.

Default: no default

Requirement: *plot-name* must be the name that has been assigned on the associated plot's NAME= option.

Interaction: This option overrides the JUSTIFY= and TYPE= options.

If this option is not specified, then the interpolation is set by the TYPE= option.

NAME=*"string"*
assigns a name to a plot statement for reference in other template statements.

Default: no default

Restriction: The *string* is case sensitive, cannot contain spaces, and must define a unique name within the template.

Interaction: The *string* is used as the default legend label if the LEGENDLABEL= option is not used.

The specified name is used primarily in legend statements to coordinate the use of colors and line patterns between the graph and the legend.

OUTLINEATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)
specifies the appearance of the band outlines. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Line Options" on page 564 for available *line-options*.

Default:

- For non-grouped data, the GraphDataDefault style element.
- For grouped data, the ContrastColor and LineStyle attributes of the GraphData1 - GraphDataN style elements.

Interaction: For this option to have any effect, the outline must be enabled by the ODS style or the DISPLAY= option.

ROLENAME=(*role-name-list*)
specifies user-defined roles for information contained in data columns.

Default: no default

(*role-name-list*)
a blank-separated list of *role-name* = *column* pairs.

For example, ROLENAME= (TIP1=OBS) assigns the column OBS to the user-defined role TIP1.

Requirement: The role names that you choose must be unique and different from the pre-defined roles X, Y, LIMITUPPER, LIMITLOWER, GROUP, CURVELABELUPPER, and CURVELABELLOWER.

This option provides a way to add to the data columns that appear in tooltips specified by the TIP= option.

TIP=(*role-list*)
specifies the information to display when the cursor is positioned over the band plot. If this option is used, it replaces all the information displayed by default. Roles for columns that do not contribute to the band plot can be specified along with roles that do.

Default: The columns assigned to these roles are automatically included in the tooltip information: X, Y, LIMITUPPER, LIMITLOWER, and GROUP.

(*role-list*)
an ordered, blank-separated list of unique BANDPLOT and user-defined roles. BANDPLOT roles include X, Y, LIMITUPPER, LIMITLOWER, GROUP, INDEX, CURVELABELUPPER, and CURVELABELLOWER.

Note: CURVELABELUPPER and CURVELABELLOWER are considered roles only when they are assigned a column of values. They are not considered roles and do not display tooltips when assigned a string.

User-defined roles are defined with the ROLENAME= option.

The following example displays tooltips for the columns assigned to the roles X, LIMITUPPER, and LIMITLOWER as well as the column OBS, which is not assigned to any pre-defined BandPlot role. The OBS column must first be assigned a role.

```
ROLENAME=(TIP1=OBS)
TIP=(TIP1 X LIMITUPPER LIMITLOWER)
```

Requirement: To generate tooltips, you must include an ODS GRAPHICS ON statement that has the IMAGEMAP option specified, and write the graphs to the ODS HTML destination.

Interaction: The labels and formats for the TIP variables can be controlled with the TIPLABEL= and TIPFORMAT= options.

TIPFORMAT=(*role-format-list*)
specifies display formats for tip columns.

Default: The column format of the variable assigned to the role or BEST6. if no format is assigned to a numeric column.

(*role-format-list*)
a list of **role-name = format** pairs separated by blanks.

```
ROLENAME=(TIP1=OBS)
TIP=(TIP1 X LIMITUPPER LIMITLOWER)
TIPFORMAT=( LIMITUPPER= 5.3 LIMITLOWER= 5.3 )
```

Requirement: This option provides a way to control the formats of columns that appear in tooltips. Only the roles that appear in the TIP= option are used. Columns must be assigned to the roles for this option to have any effect. See the ROLENAME= option.

TIPLABEL=(*role-label-list*)
specifies display labels for tip columns.

Default: The column label or column name of the variable assigned to the role.

(*role-label-list*)
a list of *role-name = "string"* pairs separated by blanks.

```
ROLENAME=(TIP1=OBS)
TIP=(TIP1 X LIMITUPPER LIMITLOWER)
TIPLABEL=( LIMITUPPER="90% CLU"
LIMITLOWER="90% CLL" )
```

Requirement: This option provides a way to control the labels of columns that appear in tooltips. Only the roles that appear in the TIP= option are used. Columns must be assigned to the roles for this option to have any effect. See the ROLENAME= option.

TYPE= SERIES | STEP

specifies how the data points for lower and upper band boundaries are interpolated.

Default: SERIES

SERIES

The data points are joined directly using line segments (as in a SeriesPlot).

STEP

The data points are connected using a step function (as in a StepPlot).

Interaction: TYPE=STEP must be specified to enable the JUSTIFY= option.

Interaction: If the MODELNAME= option is specified, this option is ignored.

XAXIS=X | X2

specifies whether data are mapped to the primary X (left) axis or to the secondary X2 (right) axis.

Default: X

Interaction: This option is ignored if the X= argument is not specified.

Interaction: The overall plot specification and the layout type determine the axis display. For more information, see "How Axis Features are Determined" on page 395.

YAXIS=Y | Y2

specifies whether data are mapped to the primary Y (bottom) axis or to the secondary Y2 (top) axis.

Default: Y

Interaction: This option is ignored if the Y= argument is not specified.

Interaction: The overall plot specification and the layout type determine the axis display. For more information, see "How Axis Features are Determined" on page 395.

Chapter 14
BARCHART Statement

Description and Syntax

Creates a bar chart computed from input data.

The following syntax creates a bar chart with bars that, by default, represent frequency counts or percents of X:

BARCHART X = *column* | *expression* </option(s)>;

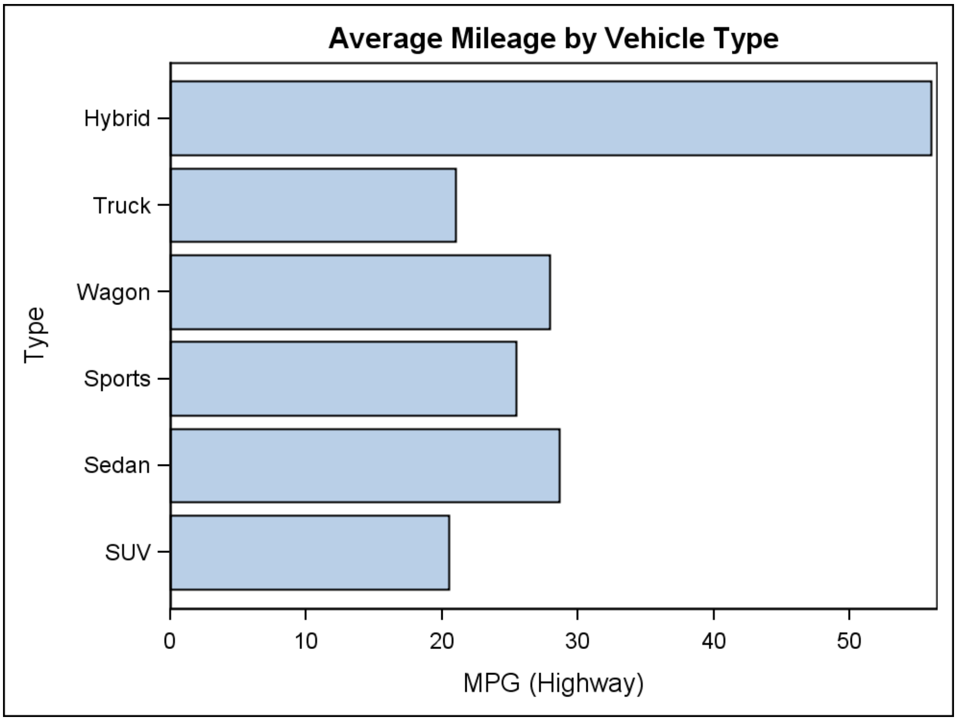
The following syntax creates a bar chart with bars representing summarized values of Y categorized by X:

BARCHART X = *column* | *expression*
Y = *numeric-column* | *expression* *</option(s)>*;

Example Program and Statement Details

Example Graph

The following graph was generated by the "Example Program" on page 140:



Example Program

```
proc template;
  define statgraph barchart;
    begingraph;
      entrytitle "Average Mileage by Vehicle Type";
      layout overlay;
        barchart x=type  y=mpg_highway /
          stat=mean orient=horizontal;
      endlayout;
    endgraph;
  end;
run;

proc sgrender data=sashelp.cars template=barchart;
run;
```

Statement Summary

The input data for the BARCHART statement is raw, unsummarized input data, and the statement calculates appropriate summarization statistics (sum, mean, and so on).

By default, if the data values are character, the bars in the chart appear in the order in which the X values are present in the input data. If the data values are character, the values are presented in ascending order. For non-grouped data, duplicated values of X are summarized into a unique value. All values are treated as discrete. For grouped data, the values of X are summarized as needed (see the GROUP= option).

The BARCHART statement only performs discrete binning for a numeric X. Use HISTOGRAM to create a chart with interval binning for X.

Arguments

X=*column* | *expression*
specifies the column or expression for the category values. This argument is required.

Y=*numeric-column* | *expression*
specifies the numeric column or expression for the response values. This argument is optional.

Options

Statement Option	Description
BARLABEL	Specifies whether the bar statistic value is displayed at the end of the bar.
BARLABELATTRS	Specifies the text properties of the bar label text.
BARLABELFORMAT	Specifies the text format used to display the bar label.
BARWIDTH	Specifies the bar width as a ratio of the maximum width.
CONNECTATTRS	Specifies the properties of the line connecting bars.
DATATRANSPARENCY	Specifies the degree of the transparency of the bar fill, bar outline, error bars, and connect line, if displayed.
DISCRETEOFFSET	Specifies an amount to offset all bars from the category midpoints.
DISPLAY	Specifies which bar features are displayed.
FILLATTRS	Specifies the appearance of the filled bar area.
GROUP	Creates a separate segment within each bar for each unique group value of the specified column.
GROUPORDER	Specifies the sorting order of grouped bars.
INDEX	Specifies indices for mapping line attributes (color and pattern) to one of the GraphData1 - GranphDataN style elements.
LEGENDLABEL	Specifies the label of the legends.
NAME	Assigns a name to a plot statement for reference in other template statements.

Statement Option	Description
ORIENT	Specifies the orientation of the Y axis and the bars.
OUTLINEATTRS	Specifies the properties of the bar outlines.
PRIMARY	Specifies that the data columns for this chart and the chart type be used for determining default axis features.
STAT	Specifies the statistic to be computed for the Y-axis.
TIP	Specifies the information to display when the cursor is positioned over a bar.
TIPFORMAT	Specifies display formats for information defined by roles.
TIPLABEL	Specifies display labels for information defined by roles.
URL	Specifies an HTML page to display when a bar is selected.
XAXIS	Specifies whether data are mapped to the primary X (bottom) axis or the secondary X2 (top) axis.
YAXIS	Specifies whether data are mapped to the primary Y (left) axis or the secondary Y2 (right) axis.

BARLABEL=*boolean*

specifies whether the bar statistic value is displayed at the end of the bar.

Default: FALSE

TIP: The font and color attributes for the label are specified by the BARLABELATTRS= option. The text format is specified by the BARLABELFORMAT= option.

BARLABELATTRS=*style-element* | *style-element* (*text-options*) | (*text-options*)

specifies the text properties of the bar label text. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Text Options" on page 566 for available *text-options*.

Default: The GraphDataText style element.

Interaction: For this option to take effect, BARLABEL=TRUE must be specified.

BARLABELFORMAT=*format*

specifies the text format used to display the bar label.

Default: The column format assigned to the Y= variable or BEST6.2 if no format is assigned.

Interaction: For this option to take effect, BARLABEL=TRUE must be specified.

BARWIDTH=*number*

specifies the width of a bar as a ratio of the maximum possible width.

Default: .85. By default, the bar width automatically adjusts based on the number of bars to be displayed and the wall width.

Range: 0 (narrowest) to 1 (widest)

This option is needed only to change the default behavior. For example, to remove any inter-bar gap, set BARWIDTH=1.

CONNECTATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)

specifies the appearance of the bar connect lines. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Line Options" on page 564 for available *line-options*.

Default: The GraphConnectLine style element.

DATATRANSPARENCY=*number*

specifies the degree of the transparency of the bar fill, bar outline, error bars, and connect line, if displayed.

Default: 0

Range: 0 (opaque) to 1 (entirely transparent)

DISCRETEOFFSET=*number*

specifies an amount to offset all bars from the category midpoints.

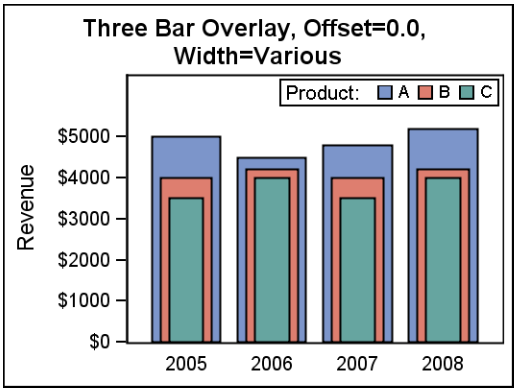
Note: This feature is for the third maintenance release of SAS 9.2 and later.

Default: 0 (no offset, all bars are centered on the category midpoints)

Range: -0.5 to +0.5, where .5 represents half the distance between category ticks. Normally, a positive offset is to the right when ORIENT=VERTICAL, and up when ORIENT=HORIZONTAL. (If the layout's axis options set REVERSE=TRUE, then the offset direction is also reversed.)

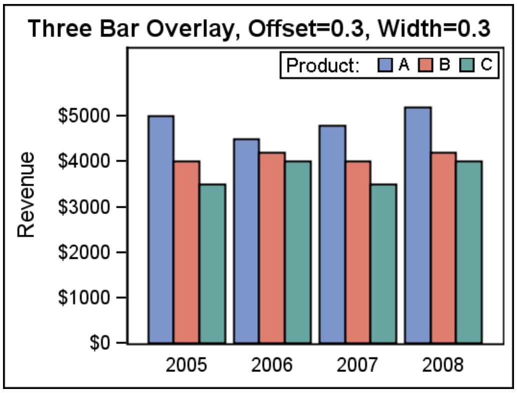
Details: This feature is useful for graphing multiple response variables side by side on a common axis. By default within an overlay-type layout, if multiple BARCHART statements are used with different response variables, the bars for matching category values are centered on the midpoints and the bars are superimposed. To make it easier to distinguish among superimposed bars, you can assign a different BARWIDTH= setting to each BARCHART statement in the overlay:

```
layout overlay / cycleattrs=true
    xaxisopts=(display=(tickvalues))
    yaxisopts=(label="Revenue" offsetmax=0.2);

  barchart x=year y=A_revenue / stat=sum name="A"
    legendlabel="A" barwidth=0.8 ;
  barchart x=year y=B_revenue / stat=sum name="B"
    legendlabel="B" barwidth=0.6 ;
  barchart x=year y=C_revenue / stat=sum name="C"
    legendlabel="C" barwidth=0.4 ;

  discretelegend "A" "B" "C" / title="Product:"
    location=inside halign=right valign=top;
endlayout;
```



To place the different response values side by side, you can assign a different offset to each BARCHART statement. If desired, the BARWIDTH= option can be used in conjunction with DISCRETEOFFSET= to create narrower bars that require less width within the plot area:

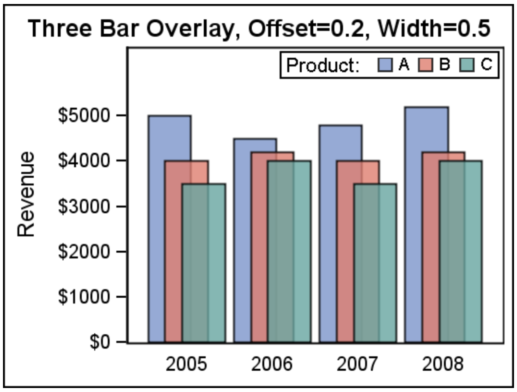
```
layout overlay / cycleattrs=true
    xaxisopts=(display=(tickvalues))
    yaxisopts=(label="Revenue" offsetmax=0.2);

  barchart x=year y=A_revenue / stat=sum name="A"
    legendlabel="A"
    discreteoffset=-0.3 barwidth=0.3 ;
  barchart x=year y=B_revenue / stat=sum name="B"
    legendlabel="B"
    discreteoffset=0    barwidth=0.3 ;
  barchart x=year y=C_revenue / stat=sum name="C"
    legendlabel="C"
    discreteoffset=+0.3 barwidth=0.3 ;

  discretelegend "A" "B" "C" / title="Product:"
    location=inside halign=right valign=top;
endlayout;
```



Different combinations of DISCRETEOFFSET and BARWIDTH can be used to get the effect that you want. Gaps can be created between bars by providing a narrower bar width. Or, bars can be overlapped if the bar widths are increased in proportion to the discrete offset.

```
layout overlay / cycleattrs=true
    xaxisopts=(display=(tickvalues))
    yaxisopts=(label="Revenue" offsetmax=0.2);
```

```
  barchart x=year y=A_revenue / stat=sum name="A"
    legendlabel="A" datatransparency=0.2
    discreteoffset=-0.2 barwidth=0.5 ;
  barchart x=year y=B_revenue / stat=sum name="B"
    legendlabel="B" datatransparency=0.2
    discreteoffset=0   barwidth=0.5 ;
  barchart x=year y=C_revenue / stat=sum name="C"
    legendlabel="C" datatransparency=0.2
    discreteoffset=+0.2 barwidth=0.5 ;

  discretelegend "A" "B" "C" / title="Product:"
    location=inside halign=right valign=top;
endlayout;
```



DISPLAY=STANDARD | ALL | (*display-options*)
specifies which bar features to display.

Default: STANDARD

STANDARD
displays outlined, filled bars

ALL
displays outlined, filled bars, and connect lines, and error bars

(*display-options*)
a list of options, enclosed in parentheses, that must include one of the following:

OUTLINE—displays outlined bars

FILL—displays filled bars

CONNECT—Displays line segments connecting adjacent midpoints at the end of each bar

Restriction: Connect lines are not drawn for grouped data.

Use the OUTLINEATTRS= and FILLATTRS= options to control the appearance of the bars. Use CONNECTATTRS= to control the appearance of the connect lines.

FILLATTRS=*style-element* | *style-element* (*fill-options*) | (*fill-options*)
specifies the appearance of the filled bar area. See “General Syntax for Attribute Options” on page 563 for the syntax on using a *style-element* and “Fill Options” on page 564 for available *fill-options*.

Default:

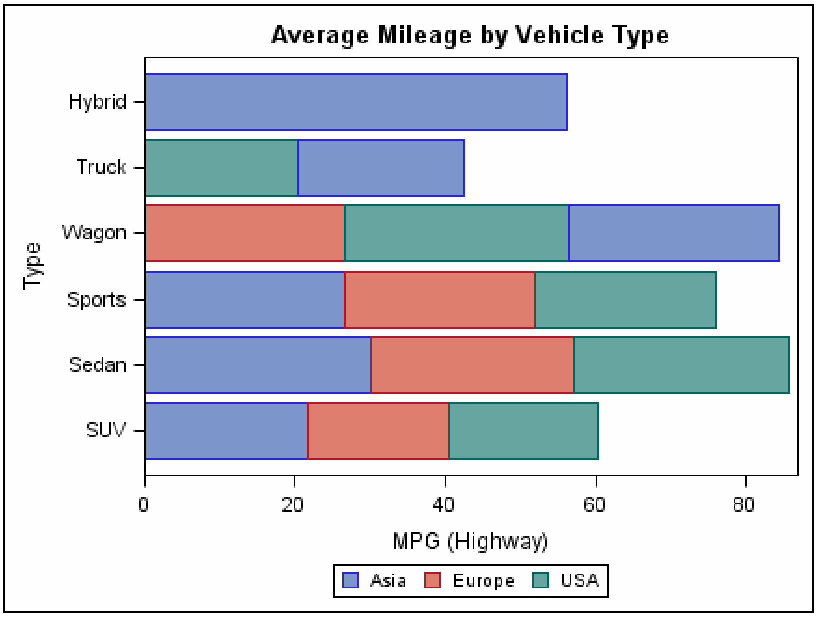
- For non-grouped data, the GraphDataDefault:Color style reference.

- For grouped data, the Color attribute of GraphData1 - GraphDataN style elements.

GROUP=*column* | *expression*

creates a separate bar segment for each unique group value of the specified column. For example, the *sashelp.cars* data used in the "Example Program" on page 140 contains a column named ORIGIN, which identifies the region that produces each car. If this option were used on the example program's BARCHART statement, as shown in the following code, the bars in the chart would display a separate bar segment to represent the regions that manufactured each type of car:

```
layout overlay;
  barchart x=type y=mpg_highway /
    stat=mean orient=horizontal
    group=origin name="grouped";
  discretelegend "grouped";
endlayout;
```



Default:

- If bar fills are enabled by the ODS style or the DISPLAY= option, each distinct group value is represented in the graph by a different fill color defined by the GraphData1:Color - GraphDataN:Color style references.
- If bar outlines are enabled by the ODS style or the DISPLAY= option, each distinct group value is represented in the graph by a different outlined color defined by the GraphData1:ContrastColor - GraphDataN:ContrastColor style references.

Interaction: Connect lines are not drawn for grouped data.

Interaction: By default, the group values are mapped in the order of the data. The GROUPORDER= option can be used to control the sorting order of the grouped bar segments. The INDEX= option can be used to alter the default sequence of colors and line patterns.

The representations that are used to identify the groups can be overridden individually. For example, each distinct group value is represented by a different line pattern for the bar outlines, but the PATTERN= setting on the OUTLINEATTRS= option could be used to assign the same line pattern to all bar outlines and connect lines.

GROUPORDER=DATA | ASCENDING | DESCENDING

specifies the ordering of bars within a group and within the legend.

Default: DATA

DATA

shows each group of bars in data order of the group column.

ASCENDING

shows each group of bars in ascending order.

DESCENDING

shows each group of bars in descending order.

Interaction: This option is ignored unless GROUP= is specified.

INDEX=*positive-integer-column* | *expression*

specifies indices for mapping line attributes (color and pattern) to one of the GraphData1 - GranphDataN style elements.

Default: no default

Restriction: For this option to take effect, the GROUP= option must also be specified.

Interaction: All of the indexes for a specific group value must be the same. Otherwise, the results are unpredictable.

Interaction: The index values are 1-based indices. For the style definitions in GraphData1 - GraphDataN, if the index value is greater than N, then a modulo operation remaps that index value to a number less than N to determine which style to use.

Interaction: If this option is not used, then the group values are mapped in the order of the data.

Discussion: Indexing can be used to collapse the number of groups that are represented in a graph. For more information, see "Remapping Groups for Grouped Data" on page 121.

LEGENDLABEL= "*string*"

specifies a label for use in a legend.

Default: The Y-variable label. If a label is not defined, the Y-variable name.

Interaction: If the GROUP= option is specified, this option is ignored.

NAME= "*string*"

assigns a name to a plot statement for reference in other template statements.

Default: no default

Restriction: The *string* is case sensitive, cannot contain spaces, and must define a unique name within the template.

The specified name is used primarily in legend statements to coordinate the use of colors and line patterns between the graph and the legend.

ORIENT= VERTICAL | HORIZONTAL

specifies the orientation of the Y axis and the bars.

Default: VERTICAL

OUTLINEATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)

specifies the appearance of the bar outlines. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Line Options" on page 564 for available *line-options*.

Default:

- For non-grouped data, the ContrastColor and LineThickness attributes of the GraphOutlines style element.
- For grouped data, the ContrastColor and LineThickness attribute of the GraphData1 - GraphDataN style elements.

Interaction: For this option to have any effect, outlines must be enabled by the ODS style or the DISPLAY= option.

PRIMARY=*boolean*

specifies that the data columns for this plot and the plot type be used for determining default axis features.

Default: FALSE

Restriction: This option is ignored if the plot is placed under a GRIDDED or LATTICE layout block.

Details: This option is needed only when two or more plots within an overlay-type layout contribute to a common axis. For more information, see "When Plots Share Data and a Common Axis" on page 397

STAT= FREQ | PCT | SUM | MEAN

specifies the statistic to be computed for the Y-axis.

Default: FREQ for bar charts with no Y=column. SUM for bar charts with a Y= column.

For bar charts with no Y= column:

FREQ

Frequency count

PCT

Percent

For bar charts with a Y= column:

SUM

Sum

MEAN

Mean

When this option is used with the GROUP=group option, the specified statistic is computed for each segment that is created for the unique group values.

TIP=(*role-list*)

specifies the information to display when the cursor is positioned over a bar. If this option is used, it replaces all the information displayed by default.

Default: The columns assigned to these roles are automatically included in the tooltip information: X, Y, and GROUP.

(*role-list*)

an ordered, blank-separated list of unique BARCHART roles. BARCHART roles include X, Y, INDEX, and GROUP.

The following example displays tooltips for the columns assigned to the roles X and Y only:

```
TIP= (X Y)
```

Requirement: To generate tooltips, you must include an ODS GRAPHICS ON statement that has the IMAGEMAP option specified, and write the graphs to the ODS HTML destination.

Interaction: The labels and formats for the TIP variables can be controlled with the TIPLABEL= and TIPFORMAT= options.

TIPFORMAT=(*role-format-list*)
specifies display formats for tip columns.

Default: The column format of the variable assigned to the role or BEST6. if no format is assigned to a numeric column.

(*role-format-list*)
a list of *role-name* = *format* pairs separated by blanks.

```
TIP=(Y)
TIPFORMAT=( Y=DOLLAR12.)
```

Requirement: This option provides a way to control the formats of columns that appear in tooltips. Only the roles that appear in the TIP= option are used. Columns must be assigned to the roles for this option to have any effect.

TIPLABEL=(*role-label-list*)
specifies display labels for tip columns.

Default: The column label or column name of the variable assigned to the role.

(*role-label-list*)
a list of *role-name* = *"string"* pairs separated by blanks.

```
TIP=(Y)
TIPLABEL=(Y="Average Sales")
```

Requirement: This option provides a way to control the labels of columns that appear in tooltips. Only the roles that appear in the TIP= option are used. Columns must be assigned to the roles for this option to have any effect.

URL=*character-column*
specifies an HTML page to display when the bar is selected.

Default: no default

character-column
each value of the column should be a valid HTML page reference (HREF). Example: *http://www.sas.com/technologies/analytics/index.html*.

Requirement: To generate selectable bars, you must include an ODS GRAPHICS ON statement that has the IMAGEMAP option specified, and write the graphs to the ODS HTML destination.

For non-grouped data, the values of the column are expected to be same for each unique X value. If they are not, the results might be unpredictable. The URL value can be blank for some X values, meaning that no action is taken when the bars for those X values are selected. The URL value can be the same for different X values, meaning that the same action is taken when the bars for those X values are selected.

For grouped data, the values of the column are expected to be the same for each unique X and GROUP combination.

XAXIS=X | X2
specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

Default: X

Interaction: The overall plot specification and the layout type determine the axis display. For more information, see "How Axis Features are Determined" on page 395.

YAXIS=Y | Y2
specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

Default: Y

Interaction: This option is ignored if the Y= argument is not specified.

Interaction: The overall plot specification and the layout type determine the axis display. For more information, see "How Axis Features are Determined" on page 395.

Chapter 15

BARCHARTPARM Statement

Description and Syntax

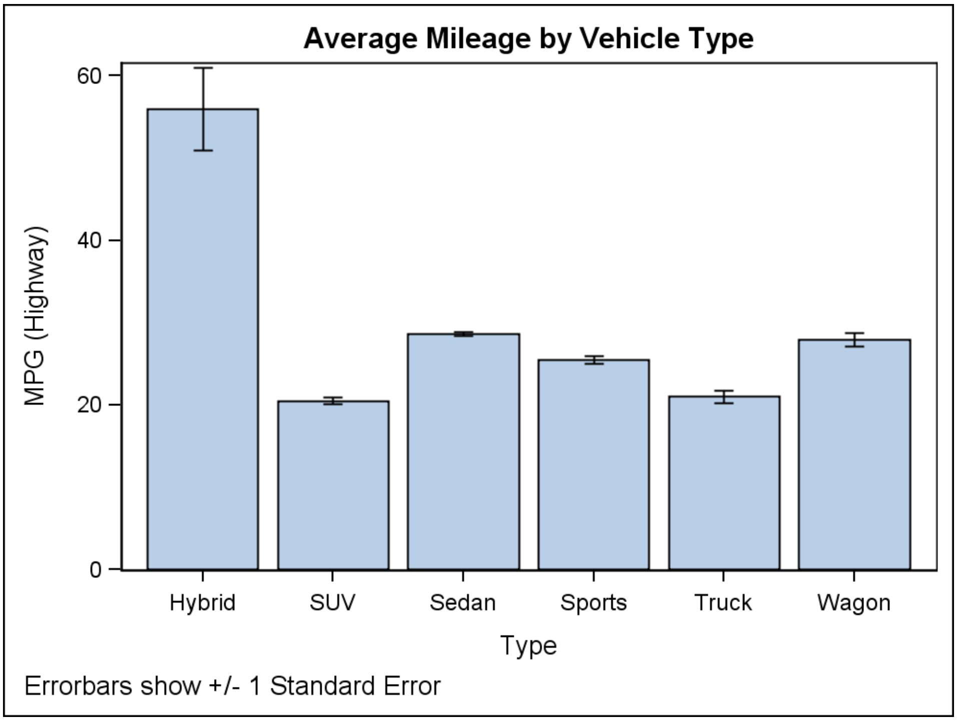
Creates a bar chart specified by pre-summarized data.

BARCHARTPARM X = *column* | *expression*
Y = *numeric-column* | *expression* </*option(s)*>;

Example Program and Statement Details

Example Graph

The following graph was generated by the “Example Program” on page 152



Example Program

```
proc template;
  define statgraph barchartparm;
    begingraph;
      entrytitle "Average Mileage by Vehicle Type";
      entryfootnote halign=left
        "Errorbars show +/- 1 Standard Error";
      layout overlay;
        barchartparm x=type y=mean /
          errorlower=eval(mean-stderr)
          errorupper=eval(mean+stderr) ;
      endlayout;
    endgraph;
  end;
run;

/* create summarized data for barchartparm */
proc summary data=sashelp.cars nway;
  class type;
  var mpg_highway;
  output out=mileage mean=mean stderr=stderr ;
run;

proc sgrender data=mileage template=barchartparm;
run;
```

Statement Summary

The input data for the BARCHARTPARM statement is expected to be pre-summarized, with appropriate summarization statistics (sum, mean, and so on) computed for the Y column. When the X values are not unique, the display is not meaningful.

By default, if the data values are character, the bars in the chart appear in the order in which the X values are present in the input data. If the data values are character, the values are presented in ascending order.

Required Arguments

X=*column* | *expression*
specifies the column for the unique category values. All values are treated as discrete.

Y=*numeric-column* | *expression*
specifies the column for the response values.

Options

Statement Option	Description
BARWIDTH	Specifies the bar width as a ratio of the maximum width.
CONNECTATTRS	Specifies the properties of the line connecting bars.
DATATRANSPARENCY	Specifies the degree of the transparency of the bar fill, bar outline, error bars, and connect line, if displayed.
DISCRETEOFFSET	Specifies an amount to offset all bars from the category midpoints.
DISPLAY	Specifies which bar features are displayed.
ERRORBARATTRS	Specifies the attributes of the error bars.
ERRORLOWER	Specifies the values of the lower endpoints on the error bars.
ERRORUPPER	Specifies the values of the upper endpoints on the error bars.
FILLATTRS	Specifies the appearance of the filled bar area.
GROUP	Creates a separate segment within each bar for each unique group value of the specified column.
GROUPORDER	Specifies the sorting order of grouped bars.

Statement Option	Description
INDEX	Specifies indices for mapping line attributes (color and pattern) to one of the GraphData1 - GranphDataN style elements.
LEGENDLABEL	Specifies the label of the legends.
NAME	Assigns a name to a plot statement for reference in other template statements.
ORIENT	Specifies the orientation of the Y axis and the bars.
OUTLINEATTRS	Specifies the properties of the bar outlines.
PRIMARY	Specifies that the data columns for this chart and the chart type be used for determining default axis features.
ROLENAME	Specifies roles for information defined by data columns.
TIP	Specifies the information to display when the cursor is positioned over a bar.
TIPFORMAT	Specifies display formats for information defined by roles.
TIPLABEL	Specifies display labels for information defined by roles.
URL	Specifies an HTML page to display when a bar is selected.
XAXIS	Specifies whether data are mapped to the primary X (bottom) axis or the secondary X2 (top) axis.
YAXIS	Specifies whether data are mapped to the primary Y (left) axis or the secondary Y2 (right) axis.

BARWIDTH=*number*

specifies the width of a bar as a ratio of the maximum possible width.

Default: .85. By default, the bar width automatically adjusts based on the number of bars to be displayed and the wall width.

Range: 0 (narrowest) to 1 (widest)

This option is needed only to change the default behavior. For example, to remove any inter-bar gap, set BARWIDTH=1.

CONNECTATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)

specifies the appearance of the bar connect lines. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Line Options" on page 564 for available *line-options*.

Default: The GraphConnectLine style element.

DATATRANSPARENCY=*number*

specifies the degree of the transparency of the bar fill, bar outline, error bars, and connect line, if displayed.

Default: 0

Range: 0 (opaque) to 1 (entirely transparent)

DISCRETEOFFSET=*number*

specifies an amount to offset all bars from the category midpoints.

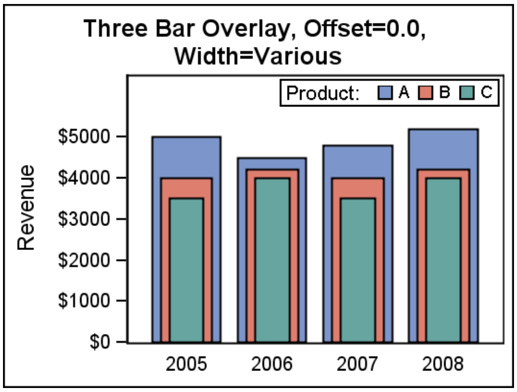
Note: This feature is for the third maintenance release of SAS 9.2 and later.

Default: 0 (no offset, all bars are centered on the category midpoints)

Range: -0.5 to +0.5, where .5 represents half the distance between category ticks. Normally, a positive offset is to the right when ORIENT=VERTICAL, and up when ORIENT=HORIZONTAL. (If the layout's axis options set REVERSE=TRUE, then the offset direction is also reversed.)

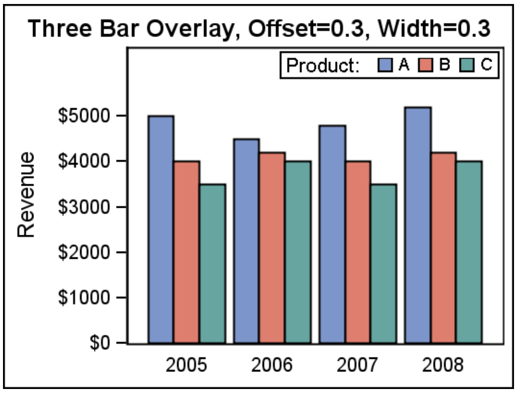
Details: This feature is useful for graphing multiple response variables side by side on a common axis. By default within an overlay-type layout, if multiple BARCHART statements are used with different response variables, the bars for matching category values are centered on the midpoints and the bars are superimposed. To make it easier to distinguish among superimposed bars, you can assign a different BARWIDTH= setting to each BARCHARTPARM statement in the overlay:

```
layout overlay / cycleattrs=true
    xaxisopts=(display=(tickvalues))
    yaxisopts=(label="Revenue" offsetmax=0.2);

  barchartparm x=year y=A_revenue / name="A"
    legendlabel="A" barwidth=0.8 ;
  barchartparm x=year y=B_revenue / name="B"
    legendlabel="B" barwidth=0.6 ;
  barchartparm x=year y=C_revenue / name="C"
    legendlabel="C" barwidth=0.4 ;

  discretelegend "A" "B" "C" / title="Product:"
    location=inside halign=right valign=top;
endlayout;
```



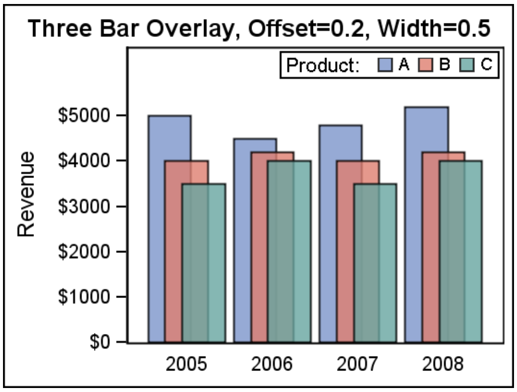
To place the different response values side by side, you can assign a different offset to each BARCHARTPARM statement. If desired, the BARWIDTH= option can be used in conjunction with DISCRETEOFFSET= to create narrower bars that require less width within the plot area:

```
layout overlay / cycleattrs=true
    xaxisopts=(display=(tickvalues))
    yaxisopts=(label="Revenue" offsetmax=0.2);

  barchartparm x=year y=A_revenue / name="A"
    legendlabel="A"
    discreteoffset=-0.3 barwidth=0.3 ;
  barchartparm x=year y=B_revenue / name="B"
    legendlabel="B"
    discreteoffset=0    barwidth=0.3 ;
  barchartparm x=year y=C_revenue / name="C"
    legendlabel="C"
    discreteoffset=+0.3 barwidth=0.3 ;

  discretelegend "A" "B" "C" / title="Product:"
    location=inside halign=right valign=top;
endlayout;
```



Different combinations of DISCRETEOFFSET and BARWIDTH can be used to get the effect that you want. Gaps can be created between bars by providing a narrower bar width. Or, bars can be overlapped if the bar widths are increased in proportion to the discrete offset.

```
layout overlay / cycleattrs=true
    xaxisopts=(display=(tickvalues))
    yaxisopts=(label="Revenue" offsetmax=0.2);

  barchartparm x=year y=A_revenue / name="A"
    legendlabel="A" datatransparency=0.2
    discreteoffset=-0.2 barwidth=0.5 ;
  barchartparm x=year y=B_revenue / name="B"
    legendlabel="B" datatransparency=0.2
    discreteoffset=0   barwidth=0.5 ;
  barchartparm x=year y=C_revenue / name="C"
    legendlabel="C" datatransparency=0.2
    discreteoffset=+0.2 barwidth=0.5 ;

  discretelegend "A" "B" "C" / title="Product:"
    location=inside halign=right valign=top;
endlayout;
```



DISPLAY=STANDARD | ALL | (*display-options*)
specifies which bar features to display.

Default: STANDARD

STANDARD
displays outlined, filled bars

ALL
displays outlined, filled bars and also connect lines

(*display-options*)
a list of options, enclosed in parentheses, that must include one of the following:

OUTLINE—displays outlined bars

FILL—displays filled bars

CONNECT—Displays line segments connecting adjacent midpoints at the end of each bar

Interaction: Connect lines are not drawn for grouped data.

Error bars are automatically displayed whenever the ERRORUPPER= or ERRORLOWER= options are specified.

Use the OUTLINEATTRS= and FILLATTRS= options to control the appearance of the bars. Use CONNECTATTRS= to control the appearance of the connect lines.

ERRORBARATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)
specifies the attributes of the error bars associated with the bars. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Line Options" on page 564 for available *line-options*.

Default:

- For non-grouped data, the GraphError style element.
- For grouped data, the LineStyle and LineThickness attributes of the GraphError style element and the ContrastColor attribute of the GraphData1 - GraphDataN style elements.

ERRORLOWER=*numeric-column* | *expression*
specifies the values of the lower endpoints on the Y error bars.

Default: The lower segment of the error bars is not drawn.

Interaction: If the GROUP= option is specified, this option is ignored.

The error bar values must be absolute data values, not data values relative to the value of the bar.

The appearance of the error bars is controlled by the ERRORBARATTRS= option.

ERRORUPPER=*numeric-column* | *expression*

specifies the values of the upper endpoints on the Y error bars.

Default: The upper segment of the error bars is not drawn.

Interaction: If the GROUP= option is specified, this option is ignored.

The error bar values must be absolute data values, not data values relative to the value of the bar.

The appearance of the error bars is controlled by the ERRORBARATTRS= option.

FILLATTRS=*style-element* | *style-element* (*fill-options*) | (*fill-options*)

specifies the appearance of the filled bar area. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Fill Options" on page 564 for available *fill-options*.

Default:

- For non-grouped data, the GraphDataDefault:Color style reference.
- For grouped data, the GraphData1:Color - GraphDataN:Color style references.

GROUP=*column* | *expression*

creates a separate bar segment for each unique group value of the specified column. For example, for a vertical bar chart, the segments would be stacked to form the bar, and the height of each segment would represent that group value's proportional contribution to the response value.

Default:

- If bar fills are enabled by the ODS style or the DISPLAY= option, each distinct group value is represented in the graph by a different fill color defined by the GraphData1:Color - GraphDataN:Color style references.
- If bar outlines are enabled by the ODS style or the DISPLAY= option, each distinct group value is represented in the graph by a different outlined color defined by the GraphData1:ContrastColor - GraphDataN:ContrastColor style references.

Interaction: Connect lines are not drawn for grouped data.

Interaction: By default, the group values are mapped in the order of the data. The GROUPORDER= option can be used to control the sorting order of the grouped bar segments. The INDEX= option can be used to alter the default sequence of colors and line patterns.

The representations that are used to identify the groups can be overridden individually. For example, each distinct group value is represented by a different line pattern for the bar outlines, but the PATTERN= setting on the OUTLINEATTRS= option could be used to assign the same line pattern to all bar outlines and connect lines.

GROUPORDER=DATA | ASCENDING | DESCENDING

specifies the ordering of bars within a group and within the legend.

Default: DATA

DATA

shows each group of bars in data order of the group column.

ASCENDING

shows each group of bars in ascending order.

DESCENDING

shows each group of bars in descending order.

Interaction: This option is ignored unless GROUP= is specified.

INDEX=*positive-integer-column* | *expression*

specifies indices for mapping line attributes (color and pattern) to one of the GraphData1 - GranphDataN style elements.

Default: no default

Restriction: For this option to take effect, the GROUP= option must also be specified.

Interaction: All of the indexes for a specific group value must be the same. Otherwise, the results are unpredictable.

Interaction: The index values are 1-based indices. For the style definitions in GraphData1 - GraphDataN, if the index value is greater than N, then a modulo operation remaps that index value to a number less than N to determine which style to use.

Interaction: If this option is not used, then the group values are mapped in the order of the data.

Discussion: Indexing can be used to collapse the number of groups that are represented in a graph. For more information, see "Remapping Groups for Grouped Data" on page 121.

LEGENDLABEL= *"string"*

specifies a label for use in a legend.

Default: The Y-variable label. If a label is not defined, the Y-variable name.

Interaction: If the GROUP= option is specified, this option is ignored.

NAME= *"string"*

assigns a name to a plot statement for reference in other template statements.

Default: no default

Restriction: The *string* is case sensitive, cannot contain spaces, and must define a unique name within the template.

The specified name is used primarily in legend statements to coordinate the use of colors and line patterns between the graph and the legend.

ORIENT= VERTICAL | HORIZONTAL

specifies the orientation of the Y axis and the bars.

Default: VERTICAL

OUTLINEATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)

specifies the appearance of the bar outlines. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Line Options" on page 564 for available *line-options*.

Default:

- For non-grouped data, the ContrastColor and LineThickness attributes of the GraphOutlines style element.
- For grouped data, the ContrastColor and LineThickness attribute of the GraphData1 - GraphDataN style elements.

Interaction: For this option to have any effect, outlines must be enabled by the ODS style or the DISPLAY= option.

PRIMARY=*boolean*

specifies that the data columns for this chart be used for determining default axis features.

Default: FALSE

Restriction: This option is ignored if the plot is placed under a GRIDDED or LATTICE layout block.

Details: This option is needed only when two or more plots within an overlay-type layout contribute to a common axis. For more information, see "When Plots Share Data and a Common Axis" on page 397

ROLENAME=(*role-name-list*)

specifies user-defined roles for information contained in data columns.

Default: no default

(*role-name-list*)

a blank-separated list of *role-name = column* pairs.

For example, ROLENAME= (TIP1=PCT) assigns the column PCT to the user-defined role TIP1.

Requirement: The role names that you choose must be unique and different from the pre-defined roles X, Y, ERRORLOWER, ERRORUPPER, GROUP, and INDEX.

This option provides a way to add to the data columns that appear in tooltips specified by the TIP= option.

TIP=(*role-list*)

specifies the information to display when the cursor is positioned over a bar. If this option is used, it replaces all the information displayed by default. Roles for columns that do not contribute to the bar chart can be specified along with roles that do.

Default: The columns assigned to these roles are automatically included in the tooltip information: X, Y, ERRORLOWER, ERRORUPPER, and GROUP.

(*role-list*)

an ordered, blank-separated list of unique BARCHARTPARM and user-defined roles. BARCHARTPARM roles include X, Y, ERRORUPPER, ERRORLOWER, INDEX, and GROUP.

User-defined roles are defined with the ROLENAME= option.

The following example displays tooltips for the columns assigned to the roles X and Y as well as the column PCT, which is not assigned to any pre-defined bar chart role. The PCT column must first be assigned a role.

```
ROLENAME=(TIP1=PCT)
TIP= (TIP1 X Y)
```

Requirement: To generate tooltips, you must include an ODS GRAPHICS ON statement that has the IMAGEMAP option specified, and write the graphs to the ODS HTML destination.

Interaction: The labels and formats for the TIP variables can be controlled with the TIPLABEL= and TIPFORMAT= options.

TIPFORMAT=(*role-format-list*)

specifies display formats for tip columns.

Default: The column format of the variable assigned to the role or BEST6. if no format is assigned to a numeric column.

(*role-format-list*)

a list of *role-name = format* pairs separated by blanks.

```
ROLENAME=(TIP1=PCT)
```

```
TIP=(TIP1 X Y)
TIPFORMAT=( TIP1=PERCENT7.2)
```

Requirement: This option provides a way to control the formats of columns that appear in tooltips. Only the roles that appear in the TIP= option are used. Columns must be assigned to the roles for this option to have any effect. See the ROLENAME= option.

TIPLABEL=(*role-label-list*)
specifies display labels for tip columns.

Default: The column label or column name of the variable assigned to the role.

(*role-label-list*)
a list of *role-name = "string"* pairs separated by blanks.

```
ROLENAME=(TIP1=PCT)
TIP=(TIP1 X Y)
TIPLABEL=( TIP1="Percent")
```

Requirement: This option provides a way to control the labels of columns that appear in tooltips. Only the roles that appear in the TIP= option are used. Columns must be assigned to the roles for this option to have any effect. See the ROLENAME= option.

URL=*character-column*
specifies an HTML page to display when the bar is selected.

Default: no default

character-column
each value of the column should be a valid HTML page reference (HREF). Example: *http://www.sas.com/technologies/analytics/index.html*.

Requirement: To generate selectable bars, you must include an ODS GRAPHICS ON statement that has the IMAGEMAP option specified, and write the graphs to the ODS HTML destination.

For non-grouped data, the values of the column are expected to be same for each unique X value. If they are not, only the first URL value for a given X value is used. The URL value can be blank for some X values, meaning that no action is taken when the bars for those X values are selected. The URL value can be the same for different X values, meaning that the same action is taken when the bars for those X values are selected.

For grouped data, the values of the column are expected to be the same for each unique X and GROUP combination.

XAXIS=X | X2
specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

Default: X

Interaction: The overall plot specification and the layout type determine the axis display. For more information, see "How Axis Features are Determined" on page 395.

YAXIS=Y | Y2
specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

Default: Y

Interaction: The overall plot specification and the layout type determine the axis display. For more information, see "How Axis Features are Determined" on page 395.

Chapter 16

BIHISTOGRAM3DPARM Statement

Description and Syntax

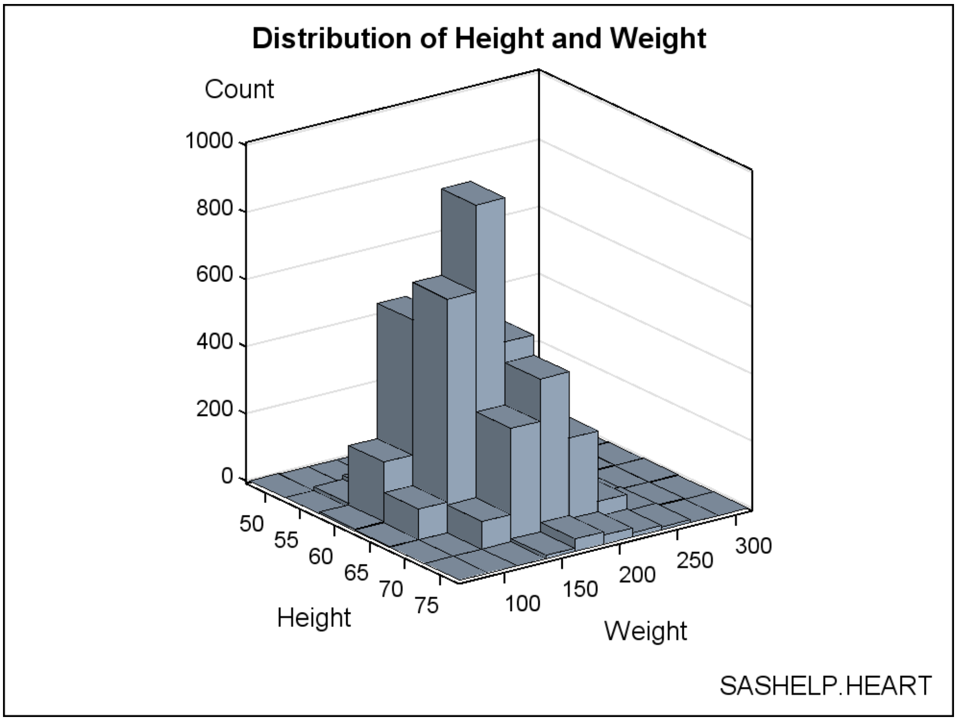
Creates a 3-D bivariate histogram of three variables X, Y, and Z, where values of X and Y have been gridded. The Z variable represents a response value for the frequency, percentage counts, or densities of each bin combination.

BIHISTOGRAM3DPARM X = *numeric-column* | *expression*
Y = *numeric-column* | *expression*
Z = *non-negative-numeric-column* | *expression* </*option(s)*>;

Example Program and Statement Details

Example Graph

The following graph was generated by the "Example Program" on page 164:



Example Program

```
proc template;
  define statgraph bihistogram;
    begingraph;
      entrytitle "Distribution of Height and Weight";
      entryfootnote halign=right "SASHELP.HEART";
      layout overlay3d / cube=false zaxisopts=(griddisplay=on);
        bihistogram3dparm x=height y=weight z=count / display=all;
      endlayout;
    endgraph;
  end;
run;

data heart;
  set sashelp.heart(keep=height weight);
    if height ne . and weight ne .;
  height=round(height,5);
  weight=round(weight,25);
run;

proc summary data=heart nway completetypes;
  class height weight;
  var height;
  output out=stats(keep=height weight count) N=Count;
run;

proc sgrender data=stats template=bihistogram;
run;
```

Statement Summary

The BIHISTOGRAM3DPARM statement does not perform a binning computation on the input columns. Thus, you must pre-bin the data. In addition, input data for the statement must be binned by both X and Y. That is, the values for the X column and the Y column must form a complete, rectangular grid of bins. Input data with non-binned columns should be preprocessed with PROC KDE (SAS/STAT), which enables you to set the number of bins for X and Y. Or the data can be preprocessed with a technique similar to the example, where a pre-defined bin width is used.

The input data for Z= column must be nonnegative. In the graph display, the direction of the Z axis is upwards rather than outwards.

The BIHISTOGRAM3DPARM statement must be specified within a LAYOUT OVERLAY3D statement and cannot be nested under an OVERLAY, OVERLAYEQUATED, or PROTOTYPE layout.

BIHISTOGRAM3DPARM does not support the tooltips that are enabled by the IMAGEMAP= option in the ODS GRAPHICS statement.

Required Arguments

X=*numeric-column* | *expression*
specifies the bin location of the numeric X values.

Y=*numeric-column* | *expression*
specifies the bin location of the numeric Y values.

Z=*non-negative-numeric-column* | *expression*
specifies the response values, such as the frequency counts, percentages, or densities.

Options

Statement Option	**Description**
BINAXIS	Specifies whether to use bins as the basis for axis tick marks for the X and Y axes.
DATATRANSPARENCY	Specifies the degree of the transparency of the bins.
DISPLAY	Specifies whether to display outlined bins, filled bins, or outlined and filled bins.
ENDLABELS	Specifies whether labels are drawn at the endpoints of the X bins and Y bins.
FILLATTRS	Specifies the appearance of the filled bins.
LEGENDLABEL	Specifies a label for use in the legend.
NAME	Assigns a name to a plot statement for reference in other template statements.

Statement Option	Description
OUTLINEATTRS	Specifies the appearance of the bin outlines.
PRIMARY	Specifies that the data columns for this plot and the plot type be used for determining default axis features.
XVALUES	Specifies whether the X values represent midpoints, lower endpoints, or upper endpoints of the bins.
YVALUES	Specifies whether the Y values represent midpoints, lower endpoints, or upper endpoints of the bins.

BINAXIS=*boolean*
specifies whether to use bins as the basis for the X and Y axis tick marks.

Default: TRUE

FALSE
Standard axes are used, ignoring bin boundaries and midpoints set by the ENDLABELS= option.

TRUE
The ENDLABELS= option determines how the axis ticks and value labels are displayed. Some axis options for the X- and Y-axes might not apply.

Interaction: For this option to take effect, this plot must be the primary plot in the parent OVERLAY3D layout. For more information, see the PRIMARY= option.

DATATRANSPARENCY=*number*
specifies the degree of the transparency of the bins.

Default: 0

Range: 0 (opaque) to 1 (entirely transparent)

DISPLAY=STANDARD | ALL | (*display-options*)
specifies whether to display outlined bins, filled bins, or outlined and filled bins.

Default: STANDARD

STANDARD
displays filled bins with no outlines.

ALL
displays outlined, filled bins.

(*display-options*)
a list of options, enclosed in parentheses, that must include at least one of the following:

OUTLINE—displays outlined bins

FILL—displays filled bins

Use the OUTLINEATTRS= and FILLATTRS= options to control the appearance of the bins.

ENDLABELS=*boolean*
specifies whether labels are drawn at the endpoints of the X bins and Y bins.

Default: FALSE. Labels are drawn at the X bin and Y bin midpoints.

Interaction: For this option to take effect, this plot must be the primary plot in the parent OVERLAY3D layout. For more information, see the PRIMARY= option.

Interaction: This option is ignored if BINAXIS=FALSE. By default, BINAXIS=TRUE.

FILLATTRS=*style-element* | *style-element* (*fill-options*) | (*fill-options*)
specifies the appearance of the filled bins. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Fill Options" on page 564 for available *fill-options*.

Default: The GraphDataDefault style element.

LEGENDLABEL= *"string"*
specifies a label for the legend.

Default: The *string* specified on the NAME= option.

NAME=*"string"*
assigns a name to a plot statement for reference in other template statements.

Default: no default

Restriction: The *string* is case sensitive, cannot contain spaces, and must define a unique name within the template.

Interaction: The *string* is used as the default legend label if the LEGENDLABEL= option is not used.

The specified name is used primarily in legend statements to coordinate the use of colors and line patterns between the graph and the legend.

OUTLINEATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)
specifies the appearance of the bin outlines. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Line Options" on page 564 for available *line-options*.

Default: The GraphOutlines style element.

PRIMARY=*boolean*
specifies that the data columns for this plot and the plot type be used for determining default axis features.

Default: FALSE

Restriction: This option is ignored if the plot is placed under a GRIDDED or LATTICE layout block.

Details: This option is needed only when two or more plots within an overlay-type layout contribute to a common axis. For more information, see "When Plots Share Data and a Common Axis" on page 397.

XVALUES= MIDPOINTS | LEFTPOINTS | RIGHTPOINTS
specifies whether the input X values represent midpoints, lower endpoints, or upper endpoints of the bins.

Default: MIDPOINTS

YVALUES= MIDPOINTS | LEFTPOINTS | RIGHTPOINTS
specifies whether the input Y values represent midpoints, lower endpoints, or upper endpoints of the bins.

Default: MIDPOINTS

Chapter 17
BLOCKPLOT Statement

Description and Syntax

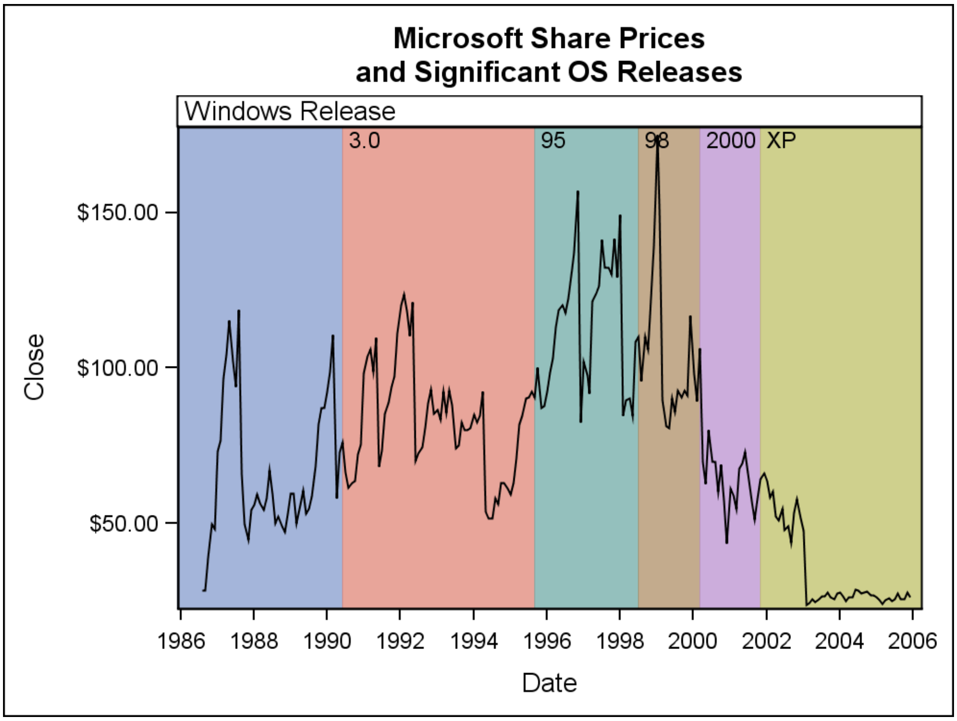
Creates one or more strips of rectangular blocks containing text values. The width of each block corresponds to specified numeric intervals.

BLOCKPLOTX = *column* | *expression*
BLOCK = *column* | *expression* <*/option(s)*>;

Example Program and Statement Details

A block plot contains information about X locations and text values to be associated with corresponding intervals along the X axis. There is no Y axis information in this plot.

BlockPlot Overlaid with SeriesPlot



When overlaid with plots that have a Y axis—a series plot for example—a block plot expands vertically to fill the Y axis range. In the BLOCKPLOT statement, the BLOCK= argument can be used to reference text values from a column. The resulting graph displays those values within the plot wall.

The following example shows how a block plot can be specified with a series plot within an Overlay layout. To prepare data for the graph, “event” information is added to existing data for stock prices. Notice that the first DATA step creates a RELEASE column. That column is later specified on the BLOCK= argument to display text values on the wall of the block plot.

```
data MSevents;
  input Date date9. Release $5.;
  label Release="Windows Release";
datalines;
01jun1990 3.0
01sep1995 95
01jul1998 98
01mar2000 2000
01nov2001 XP
run;

proc sort data=sashelp.stocks(keep=date stock close)
  out=MSstock;
  where stock="Microsoft";
  by date;
run;

data events(drop=temp);
  retain temp;
```

```
     merge MSstock MSevents(in=r);
     by date;
     if first.date and r then temp=release;
     release=temp;
   run;
```

In the combined input data set, notice that the RELEASE value is repeated for each observation within the duration of a defined “event”:

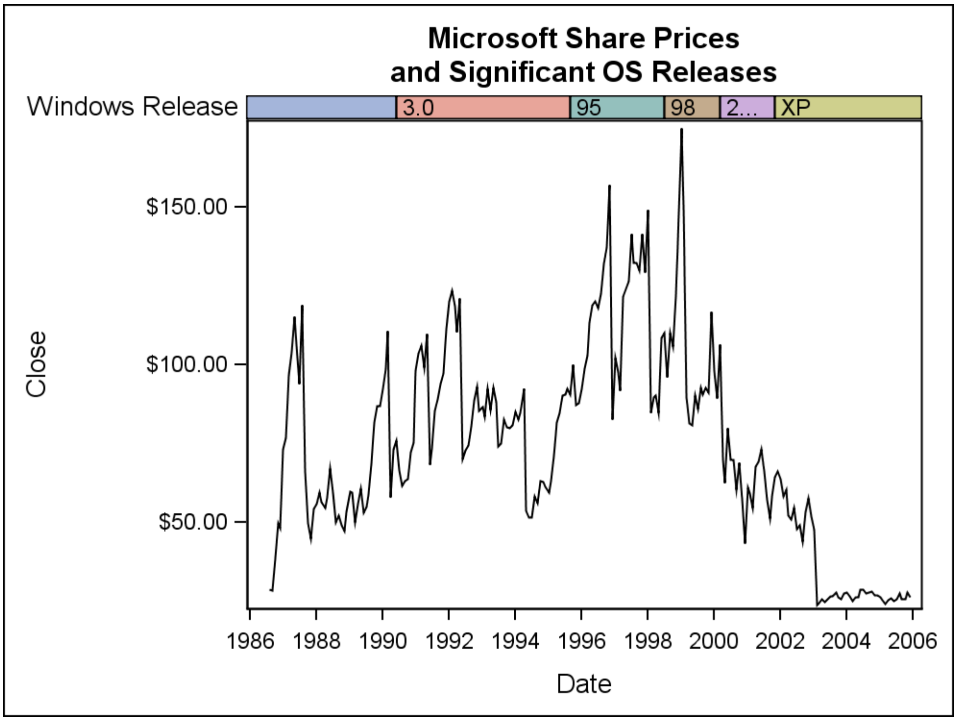
```
Obs    Date      Close   Release
...
45    02APR90   $58.00
46    01MAY90   $73.00
47    01JUN90   $76.00     3.0
48    02JUL90   $66.50     3.0
49    01AUG90   $61.50     3.0
50    04SEP90   $63.00     3.0
...
```

In the following GTL template code, BLOCK=RELEASE is specified in the BLOCKPLOT statement so that the RELEASE values are displayed on the wall of the resulting block plot.

```
proc template;
  define statgraph blockplot1;
    begingraph;
      entrytitle "Microsoft Share Prices";
      entrytitle "and Significant OS Releases";
       layout overlay;
         blockplot x=date block=release /
           datatransparency=.3 valuevalign=top
           labelposition=top display=(fill values label);
        seriesplot x=date y=close;
      endlayout;
    endgraph;
  end;
run;

proc sgrender data=events template=blockplot1;
run;
```

Stand-alone BlockPlot in Lattice Layout

When used as a stand-alone plot in a lattice layout, the block plot’s height can be controlled. Here the lattice ROWWEIGHTS=(.04 .96) option apportions 4% of the vertical space to the block plot. Using this technique, it is possible to include two or more “event” strips in a plot.



```
proc template;
  define statgraph blockplot2;
    begingraph;
      entrytitle "Microsoft Share Prices";
      entrytitle "and Significant OS Releases";
      layout lattice / rowweights=(.04 .96);
        blockplot x=date block=release / datatransparency=.3
          valuefitpolicy=shrink  labelposition=left
          display=(fill label outline values);
        seriesplot x=date y=close;
      endlayout;
    endgraph;
  end;
run;

proc sgrender data=events template=blockplot2;
run;
```

Arguments

X=*column* | *expression*
specifies numeric X axis positions. When the X-axis is numeric and the specified *column* is numeric, values are expected to be in sorted, ascending order. If the X-axis is discrete and the specified *column* is numeric, values are treated as numeric-discrete.

BLOCK=*column* | *expression*
specifies a value for each X position. Numeric values are converted to text strings according to an assigned format or BEST6.

Options

Statement Option	Description
ALTFILLATTRS	Specifies the appearance of alternate block fills.
BLOCKINDEX	Specifies indices for remapping the assignment of color to block values.
CLASS	Creates a separate block plot for each unique value of the specified column or expression.
DATATRANSPARENCY	Specifies the degree of the transparency of the block fill and outline.
DISPLAY	Specifies whether to display an outlined, a filled, or an outlined and filled block area.
FILLATTRS	Specifies the appearance of the block fills.
FILLTYPE	Specifies how the blocks are filled.
LABEL	Specifies an external label for a single block plot.
LABELATTRS	Specifies the appearance of the external BLOCK label(s).
LABELPOSITION	Specifies the alignment of BLOCK label.
NAME	Assigns a name to a plot statement for reference in other template statements.
OUTLINEATTRS	Specifies the properties of the block outlines.
REPEATEDVALUES	Specifies whether contiguous block values that are identical create separate blocks.
VALUEATTRS	Specifies the appearance of the text values.
VALUEFITPOLICY	Specifies how text values are adjusted to fit within the containing block.
VALUEHALIGN	Specifies the horizontal alignment of text with blocks.
VALUEVALIGN	Specifies the vertical alignment of text with blocks.
XAXIS	Specifies whether data are mapped to the primary X (bottom) axis or the secondary X2 (top) axis.

ALTFILLATTRS=*style-element* | *style-element* (*fill-options*) | (*fill-options*)

specifies the appearance of alternate fills. This option in conjunction with the FILLATTRS= option controls fill appearance when FILLTYPE=ALTERNATE. See “General Syntax for Attribute Options” on page 563 for the syntax on using a *style-element* and “Fill Options” on page 564 for available *fill-options*.

Default: The GraphWalls style element.

Requirement: FILLTYPE=ALTERNATE must set for this option to have any effect.

Interaction: For this option to have any effect, the fill must be enabled by the ODS style or the DISPLAY= option.

Interaction: The FILLATTRS= option controls the fill color.

To make all block fill areas the same color, set the FILLATTRS= and ALTFILLATTRS= options to the same value.

BLOCKINDEX=*positive-integer-column* | *expression*

specifies indices for remapping the assignment of the color of the block fills.

Default: no default

Requirement: FILLTYPE=MULTICOLOR must set for this option to have any effect.

Interaction: All of the indexes for a specific block value must be the same. Otherwise, the results are unpredictable.

Interaction: The index values are 1-based indices. For the style definitions in GraphData1 - GraphDataN, if the index value is greater than N, then a modulo operation remaps that index value to a number less than N to determine which style element to use.

Interaction: If this option is not used, then the color values are mapped in the order of GraphData1 - GraphDataN.

CLASS=*column* | *expression*

creates a separate block plot for each unique value of the specified column or expression. Each block plot is labeled externally by the class value.

Default: no default

Interaction: The DISPLAY= option must include LABEL for any external labels to appear.

Interaction: This option overrides the LABEL= option.

The font and color attributes for the external labels are specified by the LABELATTRS= option.

DATATRANSPARENCY=*number*

specifies the degree of the transparency of the block fill and outline. It does not affect block values or label.

Default: 0

Range: 0 (opaque) to 1 (entirely transparent)

DISPLAY=STANDARD | ALL | (*display-options*)

specifies whether to display an outlined, a filled, or an outlined and filled block area. Values and a label can also be added or suppressed.

Default: STANDARD

STANDARD

1) displays an outlined, filled block without text values or a label when used *inside* an overlay-type layout (OUTLINE FILL).

2) displays an outlined, filled block with text values and a label when used as a *stand-alone plot* inside a LATTICE or GRIDDED layout (OUTLINE FILL VALUES LABEL).

ALL
displays all possible features.

(*display-options*)
a list of options, enclosed in parentheses, that must include at least one of the following:

OUTLINE—displays an outlined, non-filled block

FILL—displays a filled, non-outlined block

VALUES—displays internal block values

LABEL—displays external block label(s)

Use the OUTLINEATTRS=, FILLATTRS=, ALTFILLATTRS=, and BLOCKINDEX= options to control the appearance of the blocks.

Use the VALUEATTRS= and LABELATTRS= options to control the text appearance.

FILLATTRS=*style-element* | *style-element* (*fill-options*) | (*fill-options*)
specifies the appearance of the block fills. This option in conjunction with the ALTFILLATTRS= option controls fill appearance when FILLTYPE=ALTERNATE. See “General Syntax for Attribute Options” on page 563 for the syntax on using a *style-element* and “Fill Options” on page 564 for available *fill-options*.

Default: The GraphData1 style element.

Requirement: FILLTYPE=ALTERNATE must set for this option to have any effect.

Interaction: For this option to have any effect, the fill must be enabled by the ODS style or the DISPLAY= option.

Interaction: The ALTFILLATTRS= option controls the alternating fill color.

To make all block fill areas the same color, set the FILLATTRS= and ALTFILLATTRS= options to the same value.

FILLTYPE=MULTICOLOR | ALTERNATE
specifies how the blocks are filled.

Default: MULTICOLOR

MULTICOLOR
Blocks are filled with the COLOR attribute of the style elements GraphData1 - GraphDataN or the style elements indicated by the BLOCKINDEX= column.

ALTERNATE
Blocks are filled alternating between the colors specified by the FILLATTRS= and ALTFILLATTRS= options.

Interaction: For this option to have any effect, the fill must be enabled by the ODS style or the DISPLAY= option.

Interaction: When this option is set to ALTERNATE, the block plot does not support a DISCRETELEGEND entry.

LABEL= *"string"*
specifies an external label for a single block plot.

Default:

- The variable label of the BLOCK= column.

- The variable name of the BLOCK= column, if there is no variable label.

Interaction: If the CLASS= option is specified, this option is ignored.

Interaction: The DISPLAY= option must include LABEL for any external label(s) to appear.

The font and color attributes for the external label are specified by the LABELATTRS= option.

LABELATTRS=*style-element* | *style-element* (*text-options*) | (*text-options*)
specifies the color and font attributes of the external block label(s). See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Text Options" on page 566 for available *text-options*.

Default: The GraphLabelText style element.

LABELPOSITION=LEFT | RIGHT | TOP | BOTTOM
specifies the alignment of BLOCK label.

Default: LEFT

LEFT
Places the label to the left of the strip of block values.

RIGHT
Places the label to the right of the strip of block values.

TOP
Places the label above the strip of block values.

BOTTOM
Places the label below the strip of block values.

NAME=*"string"*
assigns a name to a plot statement for reference in other template statements.

Default: no default

Restriction: The *string* is case sensitive, cannot contain spaces, and must define a unique name within the template.

OUTLINEATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)
specifies the appearance of the block outlines. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Line Options" on page 564 for available *line-options*.

Default: The GraphOutlines style element.

Interaction: For this option to have any effect, the outlines must be enabled by the ODS style or the DISPLAY= option.

Interaction: If labels are displayed in the TOP or BOTTOM position, they are also outlined.

REPEATEDVALUES=*boolean*
specifies whether contiguous block values that are identical create separate blocks.

Default: FALSE

FALSE
If two or more identical block values appear consecutively, only one block is created for them.

TRUE
If two or more identical block values appear consecutively, a separate block is created for each of them.

VALUEATTRS=*style-element* | *style-element* (*text-options*) | (*text-options*)

specifies the appearance of the text values. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Text Options" on page 566 for available *text-options*.

Default: The GraphValueText style element.

VALUEFITPOLICY=TRUNCATE | SHRINK

specifies how text values are adjusted to fit within the containing block.

Default: TRUNCATE

TRUNCATE

Any value that does not fit is truncated. For a numeric column, an asterisk (*) is substituted for the entire value whenever truncation occurs. For a character column, the truncated portion of the text is replaced by an ellipsis (...).

SHRINK

All values are reduced in font size until they all fit.

VALUEHALIGN=LEFT | CENTER | RIGHT | START

specifies the horizontal alignment of the text within the blocks.

Default: LEFT

LEFT

Left-aligned within the block

CENTER

Center-aligned within the block

RIGHT

Right-aligned within the block

START

Center-aligned at the starting value of the block

Interaction: For this option to have any effect, the DISPLAY= option must include VALUE.

Interaction: When REPEATEDVALUES=TRUE and X values are numeric, only CENTER and START can be used for a discrete axis, and only LEFT and START can be used for a linear or log axis. For example, if REPEATEDVALUES=TRUE and the axis is discrete, a setting of RIGHT for this option is ignored and LEFT is used instead.

VALUEVALIGN=TOP | CENTER | BOTTOM

specifies the vertical alignment of the text within the blocks.

Default: CENTER

Interaction: For this option to have any effect, the DISPLAY= option must include VALUE.

If the DISPLAY= option also includes LABEL, the label has the same vertical alignment.

XAXIS=X | X2

specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

Default: X

Interaction: The overall plot specification and the layout type determine the axis display. For more information, see "How Axis Features are Determined" on page 395.

Chapter 18

BOXPLOT Statement

Description and Syntax

Creates box-and-whisker plot(s) computed from input data.

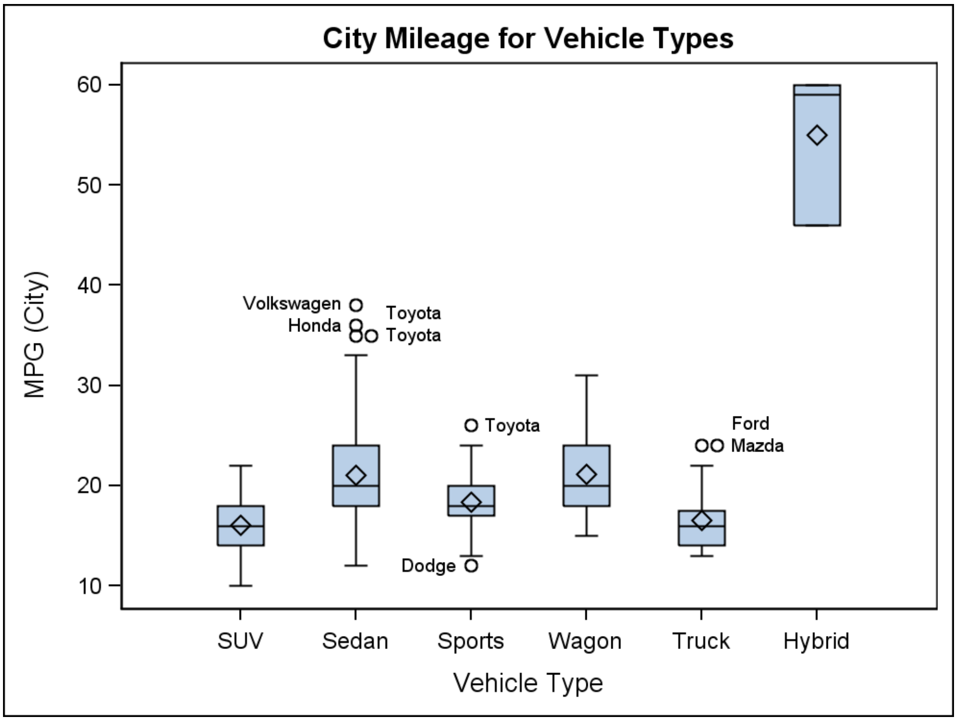
BOXPLOT Y = *numeric-column* | *expression* *</option(s)>*;

BOXPLOT X = *column* | *expression*
Y = *numeric-column* | *expression* </option(s)>;

Example Program and Statement Details

Example Graph

The following graph was generated by the "Example Program" on page 180:



Example Program

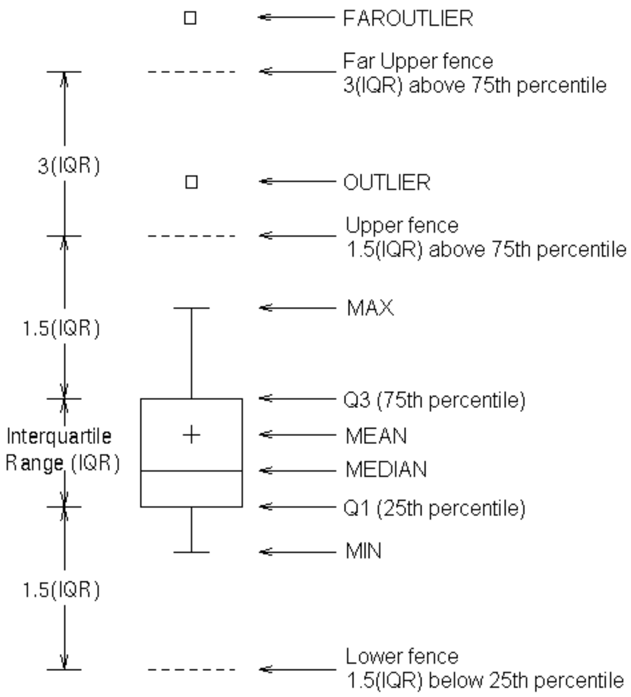
```
proc template;
  define statgraph boxplot;
    begingraph;
      entrytitle "City Mileage for Vehicle Types";
      layout overlay;
        boxplot y=mpg_city x=type /
          datalabel=make spread=true;
      endlayout;
    endgraph;
  end;
run;

proc sgrender data=sashelp.cars template=boxplot;
  label type="Vehicle Type";
run;
```

Statement Summary

The BOXPLOT statement displays a single box if given just a Y argument. It displays multiple boxes if given both Y and X arguments and X has more than one unique value. For numeric or character columns, the X-axis is always of TYPE=DISCRETE.

Two basic box plot representations can be drawn with the BOXPLOT statement: a *schematic (Tukey) box plot* and a *skeletal box plot*. See the EXTREME= option for details.

The following figure illustrates the box plot elements:



As shown in the figure, the bottom and top edges of the box are located at the 25th and 75th percentiles of the sample. Within the box you can display the median (50th percentile) as a line and the mean as a marker (see DISPLAY= option).

You can also display markers and data labels for outliers. Outliers are observations that are more extreme than the upper and lower fences (± 1.5 *IQR*). Outliers that are beyond upper and lower far fences (± 3 *IQR*) are called FAR OUTLIERS and can also be identified and labeled. From a graphical perspective, the location of fences along the axis are known, but there is no line or marker that displays a fence. (See DISPLAY=, LABELFAR=, and DATALABEL= options).

Finally, you can control the range represented by the whiskers. By default, the whiskers are drawn from the upper edge of the box to the MAX value, and from the lower edge of the box to the MIN value (see EXTREME= option).

Arguments

Y=*numeric-column* | *expression*
specifies the column for the Y values. This argument is required.

X=*column* | *expression*
specifies the column for the X values. This column is used to create a box plot for each unique X value and is optional.

Options

Statement Option	**Description**
BOXWIDTH	Specifies the width of a box as a ratio of the maximum possible width.

Statement Option	Description
CAPSHAPE	Specifies the shape at the ends of the whiskers.
CONNECT	Specifies that a connect line joins a statistic from box to box.
CONNECTATTRS	Specifies the properties of the line connecting multiple boxes.
DATALABEL	Specifies the labels of the outliers.
DATALABELATTRS	Specifies the color and font attributes of the outlier labels.
DATATRANSPARENCY	Specifies the degree of the transparency of the box outlines, box fill, whiskers, mean, median, caps, connect lines, and outliers, if displayed.
DISCRETEOFFSET	Specifies an amount to offset all boxes from the discrete X ticks.
DISPLAY	Specifies the box plot features to display.
EXTREME	Specifies whether the whiskers can extend beyond the fences.
FILLATTRS	Specifies the appearance of the interior fill area of the boxes.
FREQ	Specifies a numeric column that provides frequencies for each observation read.
LABELFAR on page 187	Specifies whether all outliers or only far outliers are labeled.
LEGENDLABEL	Specifies the label to be used in a legend.
MEANATTRS	Specifies the attributes of the maker that represents the mean values.
MEDIANATTRS	Specifies the properties of the line that represents the median values.
NAME	Assigns a name to a plot statement for reference in other template statements.
ORIENT	Specifies the orientation of the Y axis and of the boxes.
OUTLIERATTRS	Specifies the attributes of the outlier makers.
OUTLINEATTRS	Specifies the properties of the box outlines.

Statement Option	Description
PERCENTILE	Specifies one of five definitions used to calculate percentiles.
PRIMARY	Specifies that the data columns and plot type for this plot be used for determining default axis features.
SPREAD	Specifies whether outliers with the same value are spread out to avoid overlap.
WHISKERATTRS	Specifies the line properties of the whiskers and caps.
XAXIS	Specifies whether data are mapped to the primary X (bottom) axis or the secondary X2 (top) axis.
YAXIS	Specifies whether data are mapped to the primary Y (left) axis or the secondary Y2 (right) axis.

BOXWIDTH=*number*
specifies the width of a box as a ratio of the maximum possible width.

Default: .4

Range: 0 (narrowest) to 1 (widest)

CAPSHAPE=SERIF | LINE | BRACKET
specifies the shape at the ends of the whiskers.

Default: The GraphBox:CapStyle style reference.

SERIF
specifies a short line perpendicular to the whisker.

LINE
specifies a line perpendicular to whisker extending the width of the box.

BRACKET
specifies a line perpendicular to the whisker extending the width of the box with short extensions at the ends drawn in the direction of the box.

Interaction: The cap color and the thickness are specified by the WHISKERATTRS= option. The cap pattern is always solid.

Interaction: The DISPLAY= option must include CAPS for cap lines to be shown.

CONNECT= MEAN | MEDIAN | Q1 | Q3 | MIN | MAX
specifies that a connect line joins a statistic from box to box.

Default: The GraphBox:Connect style reference.

Requirement: The DISPLAY= option must contain the CONNECT suboption for the connect line to be displayed.

Interaction: This option only applies when the X= argument is used to generate multiple boxes.

CONNECTATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)

specifies the attributes of the lines connecting multiple boxes. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Line Options" on page 564 for available *line-options*.

Default: The GraphConnectLine style element.

Interaction: If there is only one box, this option is ignored.

DATALABEL=*column*

specifies the labels of the outliers. Either a numeric or a character column can be used.

Default: no default

Interaction: This option is ignored if EXTREME=TRUE or the DISPLAY= option does not display the outliers.

See also: LABELFAR= option

DATALABELATTRS=*style-element* | *style-element* (*text-options*) | (*text-options*)

specifies the color and font attributes of the outlier labels. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Text Options" on page 566 for available *text-options*.

Default: The GraphDataText style element.

Interaction: This option is ignored if EXTREME=TRUE or the DISPLAY= option does not display the outliers.

Interaction: If one or more label options are specified and they do not include all the font properties (color, family, size, weight, style), non-specified properties are derived from the GraphDataText style element.

DATATRANSPARENCY=*number*

specifies the degree of the transparency of the box outlines, box fill, whiskers, mean, median, caps, connect lines, and outliers, if displayed.

Default: 0

Range: 0 (opaque) to 1 (entirely transparent).

DISCRETEOFFSET=*number*

specifies an amount to offset all boxes from the discrete X ticks.

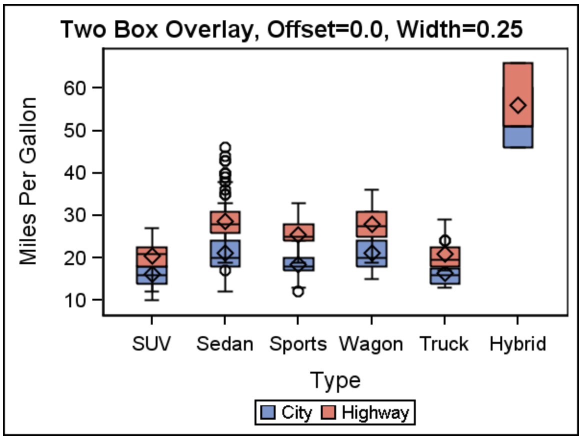
Note: This feature is for the third maintenance release of SAS 9.2 and later.

Default: 0 (no offset, all boxes are centered on the discrete ticks)

Range: -0.5 to +0.5, where .5 represents half the distance between discrete ticks. A positive offset is to the right when ORIENT=VERTICAL, and up when ORIENT=HORIZONTAL. (If the layout's axis options set REVERSE=TRUE, then the offset direction is also reversed.)

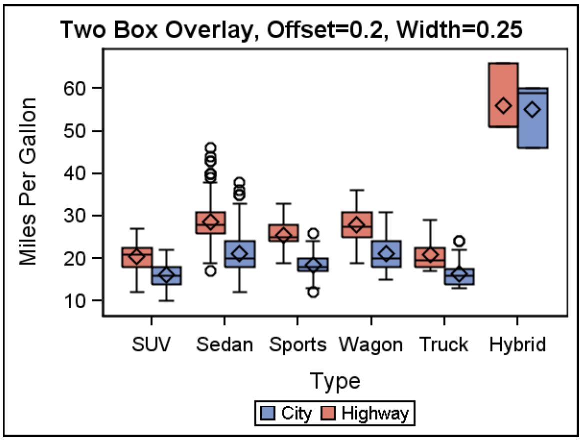
Details: This feature is useful for graphing multiple response variables side by side on a common axis. By default within an overlay-type layout, if multiple BOXPLOT statements are used with different analysis variables, the boxes for matching X values are centered on the ticks. Depending on the data, the boxes might be superimposed. The following code fragment shows the default box positioning:

```
layout overlay / cycleattrs=true
    yaxisopts=(label="Miles Per Gallon");

  boxplot x=type y=mpg_city    / name="City";
  boxplot x=type y=mpg_highway / name="Highway";
  discretelegend "City" "Highway";
endlayout;
```



To place the different response values side by side, you can assign a different offset to each BOXPLOT statement. The BOXWIDTH= option can be used in conjunction with the DISCRETEOFFSET= option to create narrower boxes when desired.

```
layout overlay / cycleattrs=true
    yaxisopts=(label="Miles Per Gallon");

  boxplot x=type y=mpg_city    / name="City"
    discreteoffset=.2 ;
  boxplot x=type y=mpg_highway / name="Highway"
    discreteoffset=-.2 ;

  discretelegend "City" "Highway";
endlayout;
```



DISPLAY=STANDARD | ALL | (*display-options*)

specifies which additional features of the box plot to display.

Default: The GraphBox:DisplayOpts style reference. If this style element does not exist, the default is STANDARD.

STANDARD

displays this combination of features (CAPS FILL MEAN MEDIAN OUTLIERS)

ALL

displays all features

(*display-options*)

a list of features, enclosed in parentheses, to be displayed. The list can include any of the following:

CAPS	displays caps at the ends of the whiskers
CONNECT	displays the line connecting multiple boxes
FILL	displays filled boxes
MEAN	displays the mean symbol within the box
MEDIAN	displays the median line within the box
NOTCHES	displays notched boxes
OUTLIERS	displays markers for the outliers

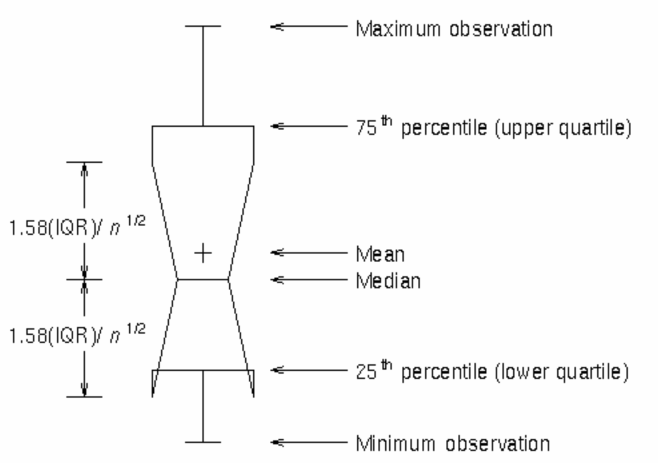
Interaction: If EXTREME=TRUE, then the OUTLIERS feature is ignored

To control the appearance of these features, use the CONNECTATTRS=, FILLATTRS=, MEANATTRS=, MEDIANATTRS=, OUTLIERATTRS=, and WHISKERATTRS= options. The WHISKERATTRS= option controls affects both CAPS and WHISKERS.

Details: The endpoints of the notches are at the following computed locations.

$$median \pm 1.58\ IQR\ /\ \sqrt{N}$$

In the equation, the IQR (IQR=Q3-Q1) is the interquartile range and N is the sample size.



EXTREME=*boolean*

specifies whether the whiskers can extend beyond the fences.

Default: FALSE

FALSE

specifies that whiskers be drawn from the upper edge of the box to the largest value within the upper fence, and from the lower edge of the box to the smallest value within the lower fence. This representation is sometime called a *schematic box and whisker plot* or *Tukey box and whisker plot*.

TRUE

specifies that whiskers be drawn to the largest and smallest data values, whether these values are inside or outside the fences. The outliers and far outliers are not displayed and are not labeled. This representation is sometime called a *skeletal box and whisker plot*.

Interaction: This option overrides the DATALABEL=, DATALABELATTRS=, LABELFAR=, OUTLIERATTRS=, and SPREAD= options.

Fences are locations above and below the box. The upper and lower fences are located at a distance 1.5 times the Interquartile Range (IQR) (IQR = Q3 - Q1). The upper and lower far fences are located at a distance 3 times the IQR (see "Example Program and Statement Details" on page 179).

FILLATTRS=*style-element* | *style-element* (*fill-options*) | (*fill-options*)

specifies the appearance of the interior fill area of the boxes. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Fill Options" on page 564 for available *fill-options*.

Default: The GraphDataDefault style element.

Interaction: For this option to have any effect, the fill must be enabled by the ODS style or the DISPLAY= option.

FREQ=*numeric-column* | *expression*

specifies a numeric column that provides frequencies for each observation read. If *n* is the value of the *numeric-column* for a given observation, then that observation is used *n* times for any statistical computation.

Default: Each observation is counted once.

Restriction: If the value of the *numeric-column* is missing or is less than 1, the observation is not used in the analysis. If the value is not an integer, only the integer portion is used.

LABELFAR=*boolean*

specifies whether all outliers or only far outliers are labeled. For more information about outliers, see the "Example Program and Statement Details" on page 179.

Default: FALSE

FALSE

the labels specified by the DATALABEL= option apply to both outliers and far outliers.

TRUE

the labels specified by the DATALABEL= option only apply to far outliers.

Interaction: This option is ignored if EXTREME=TRUE or the DISPLAY= option does not display the outliers.

LEGENDLABEL= *"string"*

specifies a label for use in a legend.

Default: The *string* specified on the NAME= option.

MEANATTRS=*style-element* | *style-element* (*marker-options*) | (*marker-options*)

specifies the attributes of the marker representing the mean within the box. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Marker Options" on page 565 for available *marker-options*.

Default: The GraphBoxMean style element.

Interaction: This option is ignored if the DISPLAY= option does not display the mean.

MEDIANATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)

specifies the appearance of the line representing the median within the box. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Line Options" on page 564 for available *line-options*.

Default: The GraphBoxMedian style element.

Interaction: This option is ignored if the DISPLAY= option does not display the median.

NAME=*"string"*

assigns a name to a plot statement for reference in other template statements.

Default: no default

Restriction: The *string* is case sensitive, cannot contain spaces, and must define a unique name within the template.

Interaction: The *string* is used as the default legend label if the LEGENDLABEL= option is not used.

The specified name is used primarily in legend statements to coordinate the use of colors and line patterns between the graph and the legend.

ORIENT= VERTICAL | HORIZONTAL

specifies the orientation of the Y axis and of the boxes.

Default: VERTICAL

OUTLIERATTRS=*style-element* | *style-element* (*marker-options*) | (*marker-options*)

specifies the attributes of the markers representing the outliers. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Marker Options" on page 565 for available *marker-options*.

Default: The GraphOutlier style element.

Interaction: This option is ignored if EXTREME=TRUE or the DISPLAY= option does not display the outliers.

OUTLINEATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)

specifies the appearance of the box outline. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Line Options" on page 564 for available *line-options*.

Default: The GraphOutlines style element.

PERCENTILE= 1 | 2 | 3 | 4 | 5

specifies one of five definitions used to calculate percentiles.

Default: 5 (empirical distribution function with averaging)

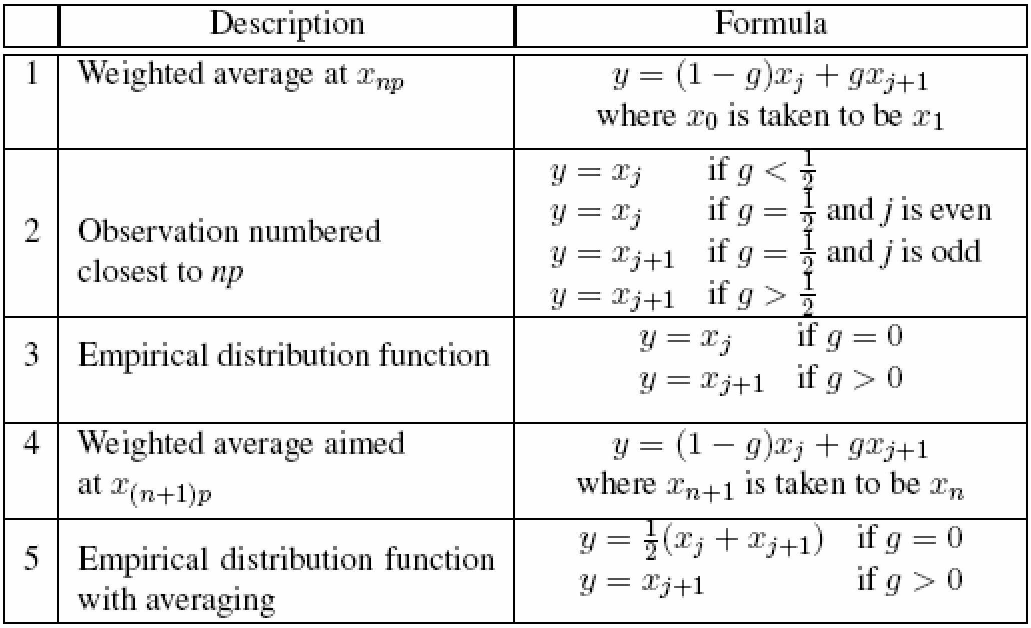
The percentile definitions and default are the same as used by PCTLDEF= option of PROC UNIVARIATE or the QNTLDEF= option of PROC SUMMARY.

Calculating Percentiles: You can specify one of five definitions for computing the percentiles with the PERCENTILE= option. Let n be the number of nonmissing values for a variable, and let X_1, X_2, ..., X_n represent the ordered values of the variable. X_1 is the smallest value, X_2 is the next smallest, and X_n is the largest value. Let the t th percentile be y, set $p = \frac{t}{100}$, and let

$np = j + g$ when PERCENTILE=1, 2, 3, or 5

$(n + 1)p = j + g$ when PERCENTILE=4

where j is the integer part of np, and g is the fractional part of np. Then the PERCENTILE= option defines the tth percentile, y, as described in the following table:



	Description	Formula
1	Weighted average at x_{np}	$y = (1-g)x_j + gx_{j+1}$ where x_0 is taken to be x_1
2	Observation numbered closest to *np*	$y = x_j \quad$ if $g < \frac{1}{2}$ $y = x_j \quad$ if $g = \frac{1}{2}$ and *j* is even $y = x_{j+1} \quad$ if $g = \frac{1}{2}$ and *j* is odd $y = x_{j+1} \quad$ if $g > \frac{1}{2}$
3	Empirical distribution function	$y = x_j \quad$ if $g = 0$ $y = x_{j+1} \quad$ if $g > 0$
4	Weighted average aimed at $x_{(n+1)p}$	$y = (1-g)x_j + gx_{j+1}$ where x_{n+1} is taken to be x_n
5	Empirical distribution function with averaging	$y = \frac{1}{2}(x_j + x_{j+1}) \quad$ if $g = 0$ $y = x_{j+1} \quad$ if $g > 0$

PRIMARY=*boolean*

specifies that the data columns for this plot and the plot type be used for determining default axis features.

Default: FALSE

Restriction: This option is ignored if the plot is placed under a GRIDDED or LATTICE layout block.

Details: This option is needed only when two or more plots within an overlay-type layout contribute to a common axis. For more information, see "When Plots Share Data and a Common Axis" on page 397

SPREAD=*boolean*

specifies whether outliers with the same value are spread out to avoid overlap. For vertical box plots this means offsetting the outliers horizontally. If this option is false, outliers with the same value are plotted in the same position. Thus, only one is visible

Default: FALSE

Interaction: This option is ignored if EXTREME=TRUE or the DISPLAY= option does not display the outliers.

WHISKERATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)

specifies the line properties of the whiskers and caps. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Line Options" on page 564 for available *line-options*.

Default: The GraphBoxWhisker style element.

XAXIS=X | X2

specifies whether data are mapped to the primary X (left) axis or to the secondary X2 (right) axis.

Default: X

Interaction: This option is ignored if the X= argument is not specified.

Interaction: The overall plot specification and the layout type determine the axis display. For more information, see "How Axis Features are Determined" on page 395.

YAXIS=Y | Y2

specifies whether data are mapped to the primary Y (bottom) axis or to the secondary Y2 (top) axis.

Default: Y

Interaction: The overall plot specification and the layout type determine the axis display. For more information, see "How Axis Features are Determined" on page 395.

Chapter 19
BOXPLOTPARM Statement

Description and Syntax

Creates side-by-side box-and-whisker plots specified by parameters.

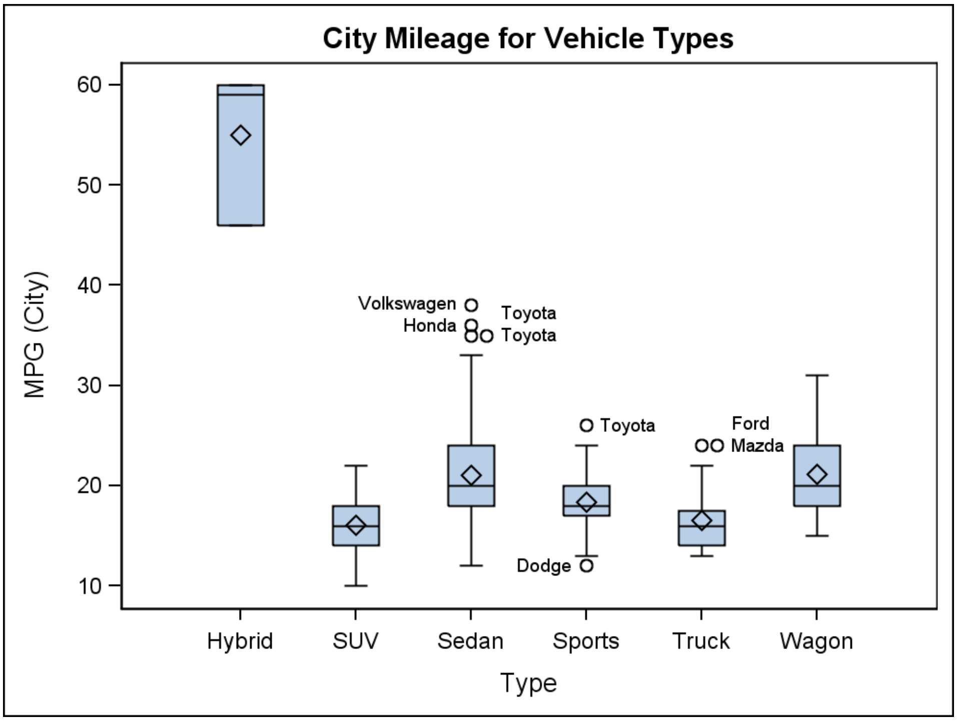
BOXPLOTPARM Y = *numeric-column* | *expression*
STAT = *string-column* </option(s)>;

BOXPLOTPARM X = *column* | *expression*
Y = *numeric-column* | *expression*
STAT = *string-column* <*/option(s)*>;

Example Program and Statement Details

Example Graph

The following graph was generated by the “Example Program” on page 192:



Example Program

```
proc template;
  define statgraph boxplotparm1;
    begingraph;
      entrytitle "City Mileage for Vehicle Types";
      layout overlay;
       boxplotparm y=value x=x stat=stat /
         datalabel=datalabel spread=true ;
      endlayout;
    endgraph;
  end;
run;

proc sgrender data=boxdata template=boxplotparm1;
run;
```

The following input data generated the box for Sedan in the graph. See Appendix A4, “Computing Input Data for BOXPLOTPARM,” on page 573 to see the code for creating all of the data.

```
STAT       X           VALUE    DATALABEL
 ...
N          Sedan         262
MEAN       Sedan     21.0840
MEDIAN     Sedan          20
Q1         Sedan          18
Q3         Sedan          24
STD        Sedan      4.2346
OUTLIER    Sedan          36    Honda
OUTLIER    Sedan          35    Toyota
OUTLIER   Sedan         35   Toyota
OUTLIER   Sedan         38   Volkswagen
MIN       Sedan         12
MAX       Sedan         33
 ...
```

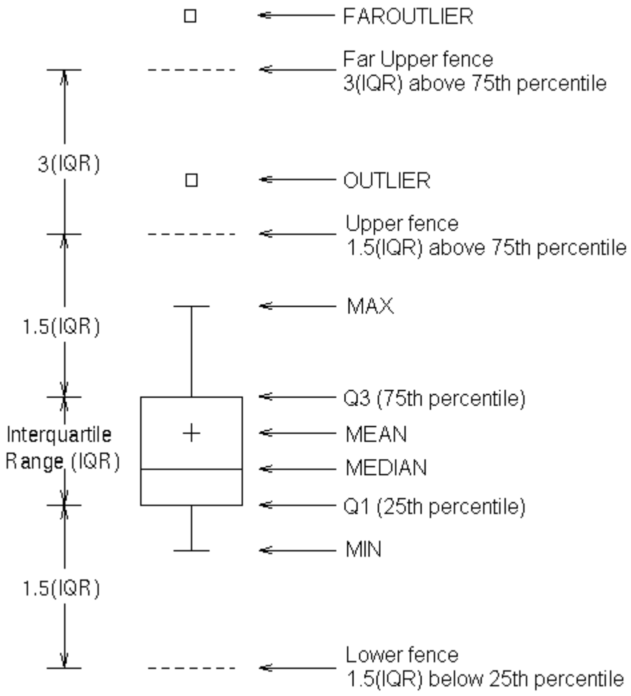
Statement Summary

The BOXPLOTPARM statement requires pre-computed input data. One reason to choose this statement over the BOXPLOT statement is that you can control the computational technique used to define the box plot components: the mean, quartiles, location of fences, outlier definition, and so on. See Appendix A4, "Computing Input Data for BOXPLOTPARM," on page 573 for examples of such computations using PROC SUMMARY and DATA steps.

The BOXPLOTPARM statement displays a single box if given just Y and a STAT argument. It displays multiple boxes if given both Y and X and a STAT argument and X has more than one unique value.

Two basic box plot representations can be drawn with the BOXPLOTPARM statement: a *schematic (Tukey) box plot* and a *skeletal box plot*. See the EXTREME= option for details.

The following figure illustrates the box plot elements:



As shown in the figure, the bottom and top edges of the box are located at the 1st quartile (25th percentile) and 3rd quartile (75th percentile) of the sample. Within the box you can display the median (50th percentile) as a line and the mean as a marker (see the DISPLAY= option).

You can also display markers and data labels for outliers. Outliers are observations that are more extreme than the upper and lower fences ($\pm 1.5\,IQR$). Outliers that are beyond upper and lower far fences ($\pm 3\,IQR$) are called FAR OUTLIERS and can also be identified and labeled. From a graphical perspective, the location of fences along the axis are known, but there is no line or marker that displays a fence. (See DISPLAY=, LABELFAR=, and DATALABEL= options).

Finally, you can control the range represented by the whiskers. By default, the whiskers are drawn from the upper edge of the box to the MAX value, and from the lower edge of the box to the MIN value (see EXTREME= option).

Input Data Requirements for the BOXPLOTPARM Statement

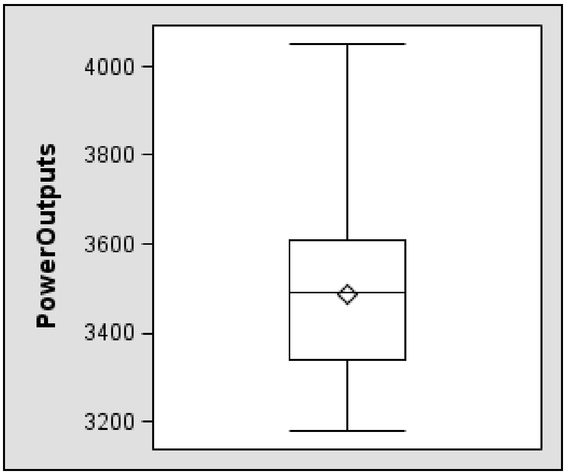
At a minimum, valid data for the BOXPLOTPARM statement must provide a numeric column (Y=) that contains calculated statistics for an analysis, and a string column (STAT=) that identifies each statistic. The Y column must contain nonmissing values for the Q1 (25th percentile) and Q3 (75th percentile) statistics. If Y values are missing or not supplied for other statistic values, then those statistics are not displayed in the plot, regardless of syntax requests to display them.

For example, a petroleum company uses a turbine to heat water into steam that is pumped into the ground to make oil more viscous and easier to extract. This process occurs 20 times daily, and the amount of power (in kilowatts) used to heat the water to the desired temperature is recorded. The following data show the statistics that are calculated for one day of this process:

PowerOutputs	Statistic
3180.00	MIN
3340.00	Q1
3487.40	MEAN
3490.00	MEDIAN
3610.00	Q3
4050.00	MAX
20.00	N

To plot the data from the preceding table, the following BOXPLOTPARM statement uses the Y= and STAT= arguments to generate a single box plot for the recorded statistics:

```
BOXPLOTPARM Y=PowerOutputs STAT=Statistic;
```

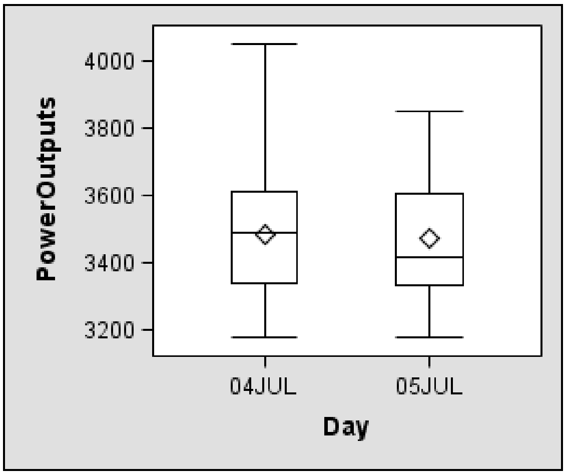


If the data contain statistics for multiple days of the process, a third column in the data must be present to identify the days that the statistics were recorded. For example, the following data show the statistics that are calculated for two days of this process:

Day	**PowerOutputs**	**Statistic**
04JUL	3180.00	MIN
04JUL	3340.00	Q1
04JUL	3487.40	MEAN
04JUL	3490.00	MEDIAN
04JUL	3610.00	Q3
04JUL	4050.00	MAX
04JUL	20.00	N
05JUL	3179.00	MIN
05JUL	3333.50	Q1
05JUL	3471.65	MEAN
05JUL	3419.50	MEDIAN
05JUL	3605.00	Q3
05JUL	3849.00	MAX
05JUL	20.00	N

To plot the data from the preceding table, the BOXPLOTPARM statement needs the Y=, STAT=, and X= arguments to generate a separate box plot for each day that the statistics were recorded:

```
BOXPLOTPARM Y=PowerOutputs STAT=Statistic X=Day;
```



See Appendix A4, “Computing Input Data for BOXPLOTPARM,” on page 573 for a more complete example of providing input data for BOXPLOTPARM.

Arguments

Y=*numeric-column* | *expression*
specifies the column for the Y values. The Y values must be the statistical values needed for the box plot. At a minimum, there must be nonmissing values for the 25th and 75th percentiles.

X=*column* | *expression*
specifies the column for the X values. The X values must qualify or classify the values in the Y column. This optional argument is used to create a plot box for each classifier.

STAT=*string-column*
specifies the statistic that is represented by the value in the Y column. Valid STAT= values include the following:

Q1
1st quartile (25th percentile). The data must contain a nonmissing value for this quartile.

Q3
3rd quartile (75th percentile). The data must contain a nonmissing value for this quartile.

MAX
maximum data value less than or equal to the upper fence.

MIN
minimum data value greater than or equal to the lower fence.

MEAN
data mean.

MEDIAN
data median.

OUTLIER
: an observation outside the lower and upper fences. The fences are located at a distance 1.5 times the Interquartile Range (IQR = Q3 - Q1) above and below the box. The outliers are labeled when the DATALABEL= option is used.

FAROUTLIER
: an observation outside the lower and upper far fences. The far fences are located at a distance 3 times the Interquartile Range (IQR = Q3 - Q1) above and below the box. The far outliers are labeled when the DATALABEL= option is used. Specify that LABELFAR=TRUE to label only the far outliers but not the outliers.

N
: subgroup sample size. The N value is not shown in the plot but is used to calculate notch locations when the DISPLAY= option displays notches.

STD
: data standard deviation.

Requirement: Other STAT values can be omitted or have missing Y values, but if present, must conform to the following rules for the plot to be displayed:

```
Q1 <= MEDIAN <= Q3
MIN <= MAX
STD >= 0
N > 0
```

Options

Statement Option	**Description**
BOXWIDTH	Specifies the width of a box as a ratio of the maximum possible width.
CAPSHAPE	Specifies the shape at the ends of the whiskers.
CONNECT	Specifies that a connect line joins a statistic from box to box.
CONNECTATTRS	Specifies the properties of the line connecting multiple boxes.
DATALABEL	Specifies the labels of the outliers.
DATALABELATTRS	Specifies the color and font attributes of the outlier labels.
DATATRANSPARENCY	Specifies the degree of the transparency of the box outlines, box fill, whiskers, mean, median, caps, connect lines, and outliers, if displayed.
DISCRETEOFFSET	Specifies an amount to offset all boxes from the discrete X ticks.
DISPLAY	Specifies the box plot features to display.

Statement Option	Description
EXTREME	Specifies whether the whiskers can extend beyond the fences.
FILLATTRS	Specifies the appearance of the interior fill area of the boxes.
LABELFAR	Specifies whether all outliers or only far outliers are labeled.
LEGENDLABEL	Specifies the label to be used in a legend.
MEANATTRS	Specifies the attributes of the maker that represents the mean values.
MEDIANATTRS	Specifies the properties of the line that represents the median values.
NAME	Assigns a name to a plot statement for reference in other template statements.
ORIENT	Specifies the orientation of the Y axis and of the boxes.
OUTLIERATTRS	Specifies the attributes of the outlier makers.
OUTLINEATTRS	Specifies the line properties of the box outlines.
PRIMARY	Specifies that the data columns and plot type for this plot be used for determining default axis features.
SPREAD	Specifies whether outliers with the same value are spread out to avoid overlap.
WHISKERATTRS	Specifies the line properties of the whiskers and caps.
XAXIS	Specifies whether data are mapped to the primary X (bottom) axis or the secondary X2 (top) axis.
YAXIS	Specifies whether data are mapped to the primary Y (left) axis or the secondary Y2 (right) axis.

BOXWIDTH=*number*
specifies the width of a box as a ratio of the maximum possible width.

Default: .4

Range: 0 (narrowest) to 1 (widest)

CAPSHAPE=SERIF | LINE | BRACKET
specifies the shape at the ends of the whiskers.

Default: The GraphBox:CapStyle style reference.

SERIF
specifies a short line perpendicular to the whisker.

LINE
specifies a line perpendicular to whisker extending the width of the box.

BRACKET
specifies a line perpendicular to the whisker extending the width of the box with short extensions at the ends drawn in the direction of the box.

Interaction: The cap color and the thickness are specified by the WHISKERATTRS= option. The cap pattern is always solid.

Interaction: The DISPLAY= option must include CAPS for cap lines to be shown.

CONNECT= MEAN | MEDIAN | Q1 | Q3 | MIN | MAX
specifies that a connect line joins a statistic from box to box.

Default: The GraphBox:Connect style reference.

Requirement: The DISPLAY= option must contain the CONNECT suboption for the connect line to be displayed.

Interaction: This option only applies when the X= argument is used to generate multiple boxes.

CONNECTATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)
specifies the attributes of the lines connecting multiple boxes. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Line Options" on page 564 for available *line-options*.

Default: The GraphConnectLine style element.

Interaction: If there is only one box, this option is ignored.

DATALABEL=*column*
specifies the labels of the values that are identified as outlier or faroutlier by the STAT= column. Either a numeric or a character column can be used.

Default: no default

Interaction: This option is ignored if EXTREME=TRUE or the DISPLAY= option does not display the outliers.

See also: LABELFAR= option

DATALABELATTRS=*style-element* | *style-element* (*text-options*) | (*text-options*)
specifies the color and font attributes of the outlier labels. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Text Options" on page 566 for available *text-options*.

Default: The GraphDataText style element.

Interaction: This option is ignored if EXTREME=TRUE or the DISPLAY= option does not display the outliers.

Interaction: If one or more specified label options does not include all the font properties (color, family, size, weight, style), non-specified properties are derived from the GraphDataText style element.

DATATRANSPARENCY=*number*
specifies the degree of the transparency of the box outlines, box fill, whiskers, mean, median, caps, connect lines, and outliers, if displayed.

Default: 0

Range: 0 (opaque) to 1 (entirely transparent).

DISCRETEOFFSET=*number*

specifies an amount to offset all boxes from the discrete X ticks.

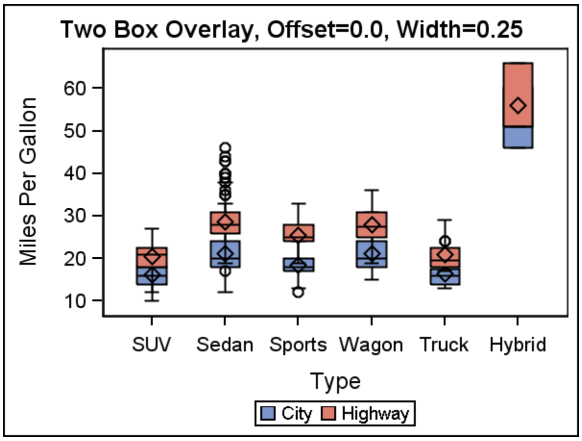
Note: This feature is for the third maintenance release of SAS 9.2 and later.

Default: 0 (no offset, all boxes are centered on the discrete ticks)

Range: -0.5 to +0.5, where .5 represents half the distance between discrete ticks. A positive offset is to the right when ORIENT=VERTICAL, and up when ORIENT=HORIZONTAL. (If the layout's axis options set REVERSE=TRUE, then the offset direction is also reversed.)

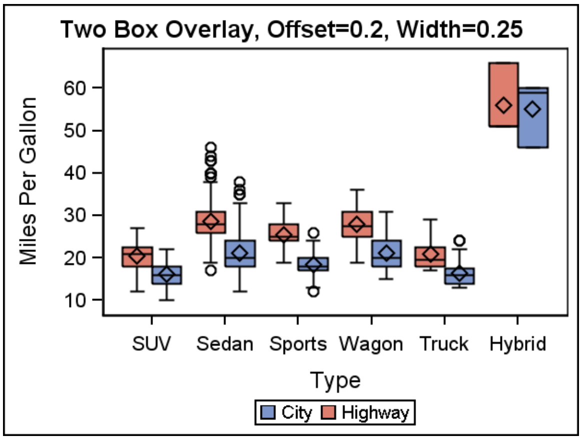
Details: This feature is useful for graphing multiple response variables side by side on a common axis. By default within an overlay-type layout, if multiple BOXPLOTPARM statements are used with different analysis variables, the boxes for matching X values are centered on the ticks. Depending on the data, the boxes might be superimposed. The following code fragment shows the default box positioning:

```
layout overlay / cycleattrs=true
    yaxisopts=(label="Miles Per Gallon");

  boxplotparm x=type y=mpg_city stat=y_stat    / name="City" ;
  boxplotparm x=type y=mpg_highway stat=y_stat / name="Highway" ;

  discretelegend "City" "Highway";
endlayout;
```



To place the different response values side by side, you can assign a different offset to each BOXPLOTPARM statement. The BOXWIDTH= option can be used in conjunction with the DISCRETEOFFSET= option to create narrower boxes when desired.

```
layout overlay / cycleattrs=true
    yaxisopts=(label="Miles Per Gallon");

  boxplotparm x=type y=mpg_city stat=y_stat    / name="City"
    discreteoffset=.2 ;
  boxplotparm x=type y=mpg_highway stat=y_stat / name="Highway"
    discreteoffset=-.2 ;

  discretelegend "City" "Highway";
endlayout;
```



DISPLAY=STANDARD | ALL | (*display-options*)

specifies which additional features of the box plot to display.

Default: The GraphBox:DisplayOpts style reference. If this style element does not exist, the default is STANDARD.

STANDARD

displays this combination of features (CAPS FILL MEAN MEDIAN OUTLIERS)

ALL

displays all features

display-options

a list of features to be displayed. The list must be enclosed in parentheses and can include any of the following:

CAPS	displays caps at the ends of the whiskers
CONNECT	displays the line connecting multiple boxes
FILL	displays filled boxes
MEAN	displays the mean symbol within the box
MEDIAN	displays the median line within the box
NOTCHES	displays notched boxes
OUTLIERS	displays markers for the outliers

Restriction: The display features requested can be displayed only if the input data includes this information.

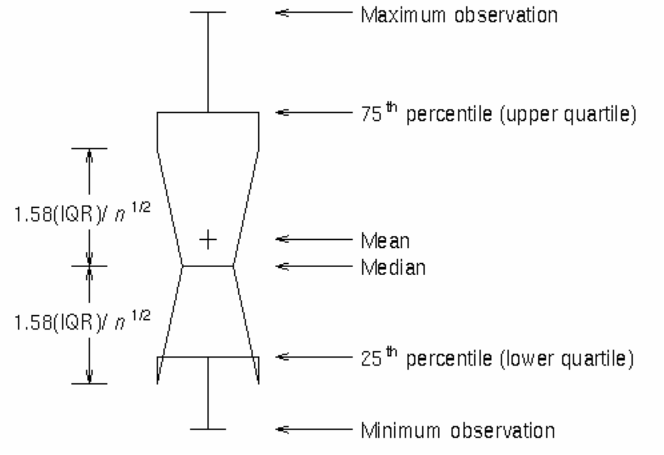
Interaction: If EXTREME=TRUE, then the OUTLIERS feature is ignored

To control the appearance of these features, use the CONNECTATTRS=, FILLATTRS=, MEANATTRS=, MEDIANATTRS=, OUTLIERATTRS=, and WHISKERATTRS= options. The WHISKERATTRS= option controls affects both CAPS and WHISKERS.

Details: The endpoints of the notches are at the following computed locations.

$$\textit{median} + \textit{ and} - 1.58(\textit{IQR} / \sqrt{N})$$

In the equation, the IQR is the interquartile range and N is the sample size.



EXTREME=*boolean*

specifies whether the whiskers can extend beyond the fences.

Default: FALSE

FALSE

specifies that whiskers be drawn from the upper edge of the box to the largest value within the upper fence, and from the lower edge of the box to the smallest value within the lower fence. This representation is sometime called a *schematic box and whisker plot* or *Tukey box and whisker plot*.

TRUE

specifies that whiskers be drawn to the largest and smallest data values, whether these values are inside or outside the fences. The outliers and far outliers are not displayed and are not labeled. This representation is sometime called a *skeletal box and whisker plot*.

Interaction: This option overrides the DATALABEL=, DATALABELATTRS=, LABELFAR=, OUTLIERATTRS=, and SPREAD= options.

Fences are locations above and below the box. The upper and lower fences are located at a distance 1.5 times the Interquartile Range (IQR) (IQR = Q3 - Q1). The upper and lower far fences are located at a distance 3 times the IQR (see "Example Program and Statement Details" on page 191).

FILLATTRS=*style-element* | *style-element* (*fill-options*) | (*fill-options*)

specifies the appearance of the interior fill area of the boxes. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Fill Options" on page 564 for available *fill-options*.

Default: The GraphDataDefault style element.

Interaction: For this option to have any effect, the fill must be enabled by the ODS style or the DISPLAY= option.

LABELFAR=*boolean*

specifies whether all outliers or only far outliers are labeled. For more information about outliers, see the "Example Program and Statement Details" on page 191.

Default: FALSE

FALSE

the labels specified by the DATALABEL= option apply to both outliers and far outliers.

TRUE
the labels specified by the DATALABEL= option only apply to far outliers.

Interaction: This option is ignored if EXTREME=TRUE or the DISPLAY= option does not display the outliers.

LEGENDLABEL= *"string"*
specifies a label for use in a legend.

Default: The *string* specified on the NAME= option.

MEANATTRS=*style-element* | *style-element* (*marker-options*) | (*marker-options*)
specifies the attributes of the marker representing the mean within the box. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Marker Options" on page 565 for available *marker-options*.

Default: The GraphBoxMean style element.

Interaction: This option is ignored if the DISPLAY= option does not display the mean.

MEDIANATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)
specifies the appearance of the line representing the median within the box. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Line Options" on page 564 for available *line-options*.

Default: The GraphBoxMedian style element.

Interaction: This option is ignored if the DISPLAY= option does not display the median.

NAME=*"string"*
assigns a name to a plot statement for reference in other template statements.

Default: no default

Restriction: The *string* is case sensitive, cannot contain spaces, and must define a unique name within the template.

Interaction: The *string* is used as the default legend label if the LEGENDLABEL= option is not used.

The specified name is used primarily in legend statements to coordinate the use of colors and line patterns between the graph and the legend.

ORIENT= VERTICAL | HORIZONTAL
specifies the orientation of the Y axis and of the boxes.

Default: VERTICAL

OUTLIERATTRS=*style-element* | *style-element* (*marker-options*) | (*marker-options*)
specifies the attributes of the markers representing the outliers. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Marker Options" on page 565 for available *marker-options*.

Default: The GraphOutlier style element.

Interaction: This option is ignored if EXTREME=TRUE or the DISPLAY= option does not display the outliers.

OUTLINEATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)
specifies the appearance of the box outline. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Line Options" on page 564 for available *line-options*.

Default: The GraphOutlines style element.

PRIMARY=*boolean*
specifies that the data columns for this plot and the plot type be used for determining default axis features.

Default: FALSE

Restriction: This option is ignored if the plot is placed under a GRIDDED or LATTICE layout block.

Details: This option is needed only when two or more plots within an overlay-type layout contribute to a common axis. For more information, see "When Plots Share Data and a Common Axis" on page 397

SPREAD=*boolean*
specifies whether outliers with the same value are spread out to avoid overlap. For vertical box plots this means offsetting the outliers horizontally. If this option is false, outliers with the same value are plotted in the same position. Thus, only one is visible

Default: FALSE

Interaction: This option is ignored if EXTREME=TRUE or the DISPLAY= option does not display the outliers.

WHISKERATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)
specifies the line properties of the whiskers and caps. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Line Options" on page 564 for available *line-options*.

Default: The GraphBoxWhisker style element.

XAXIS=X | X2
specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

Default: X

Interaction: This option is ignored if the X= argument is not specified.

Interaction: The overall plot specification and the layout type determine the axis display for the specified axis. For more information, see "How Axis Features are Determined" on page 395.

YAXIS=Y | Y2
specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

Default: Y

Interaction: The overall plot specification and the layout type determine the axis display for the specified axis. For more information, see "How Axis Features are Determined" on page 395.

Chapter 20

CONTOURPLOTPARM Statement

Description and Syntax

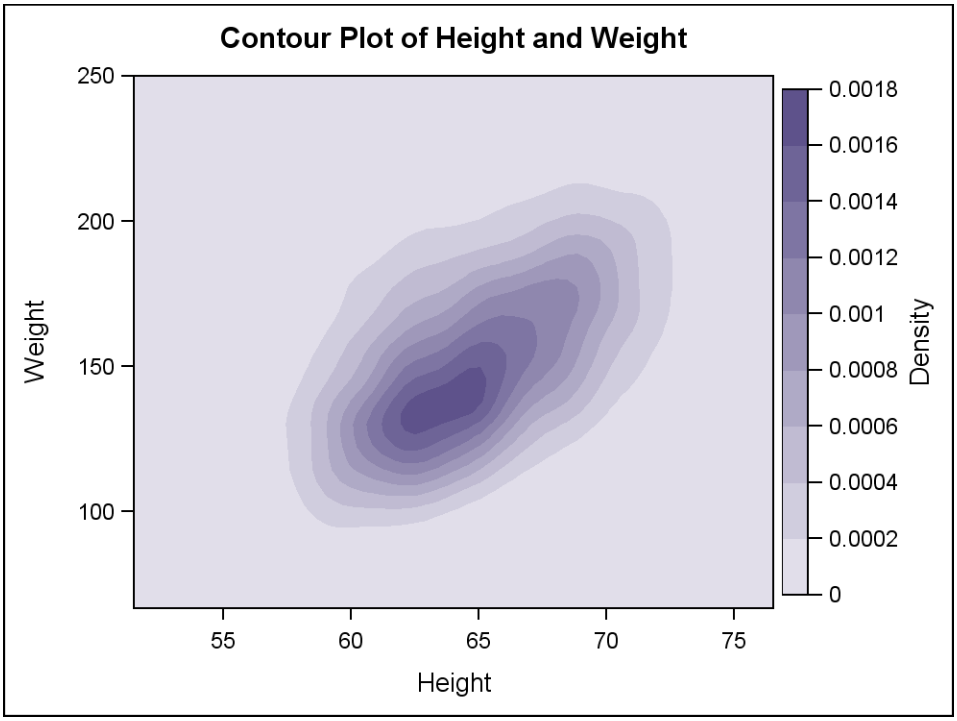
Creates a contour plot representing a response variable evaluated over a grid of X and Y values.

CONTOURPLOTPARM X = *numeric-column* | *expression*
Y = *numeric-column* | *expression*
Z = *numeric-column* | *expression* </*option(s)*>;

Example Program and Statement Details

Example Graph

The following graph was generated by the “Example Program” on page 206:



Example Program

```
proc template;
 define statgraph contourplotparm;
   begingraph;
     entrytitle "Contour Plot of Height and Weight";
     layout overlay /
         xaxisopts=(offsetmin=0 offsetmax=0
           linearopts=(thresholdmin=0 thresholdmax=0))
         yaxisopts=(offsetmin=0 offsetmax=0
           linearopts=(viewmax=250
                       thresholdmin=0 thresholdmax=0 ));
       contourplotparm x=height y=weight z=density /
         contourtype=fill nhint=12
           name="Contour" colormodel=twocolorramp;
       continuouslegend "Contour" / title="Density";
     endlayout;
   endgraph;
 end;
run;
proc sgrender data=sashelp.gridded template=contourplotparm;
run;
```

Statement Summary

A CONTOURPLOTPARM statement uses the CONTOURTYPE= option to specify the type of contour plot to generate. Contour types that display fills or gradients but no contour lines can use only a CONTINUOUSLEGEND statement to represent the contour level values in a legend. Contour types that display lines can use either a CONTINUOUSLEGEND or DISCRETELEGEND statement to identify contour level values.

By default, the CONTOURPLOTPARM statement assumes that the X-Y grid is complete. If the grid is not complete, then set the GRIDDED= option to FALSE so that the plot calculates the values needed to complete into a grid with a bounding rectangle.

For filled or gradient contour types, small gaps might be visible between the axes and the bounding box of the contour data. To eliminate these gaps, set the axis options of the LAYOUT OVERLAY statement as follows:

```
XAXISOPTS=(OFFSETMIN=0  OFFSETMAX=0
          LINEAROPTS=(THRESHOLDMIN=0 THRESHOLDMAX=0) )

YAXISOPTS=(OFFSETMIN=0  OFFSETMAX=0
          LINEAROPTS=(THRESHOLDMIN=0 THRESHOLDMAX=0 ))
```

Contour plots do not support the tooltips that are enabled by the IMAGEMAP= option in the ODS GRAPHICS statement.

Required Arguments

X=*numeric-column* | *expression*
specifies the X coordinates for the grid.

Y=*numeric-column* | *expression*
specifies the Y coordinates for the grid.

Z=*numeric-column* | *expression*
specifies the contour response values.

Options

Statement Option	Description
COLORMODEL	Specifies a style element that is used to determine the colors of filled or gradient contours.
CONTOURTYPE	Specifies how the contour is displayed.
GRIDDED	Specifies whether the grid formed by the X and Y values is complete.
LEGENDLABEL	Specifies the label to be used in a discrete legend.
LINEATTRS	Specifies the properties of the contour lines.
LINELABELATTRS	Specifies the color and font attributes of the contour line labels.
LINELABELBASELINE	Specifies the alignment of the contour line labels.
LINELABELFORMAT	Specifies the format of the contour line labels.
LINELABELPOSITION	Specifies the position of the contour line labels.

Statement Option	Description
NAME	Assigns a name to a plot statement for reference in other template statements.
NHINT	Specifies the suggested number of contour levels for the Z variable.
NLEVELS	Specifies the actual number of contour levels for the Z variable.
PRIMARY	Specifies that the data columns for this plot and the plot type be used for determining default axis features.
REVERSECOLORMODEL	Specifies whether to reverse the gradient defined by the COLORMODEL= on page 208 option.
XAXIS	Specifies whether data are mapped to the primary X (bottom) axis or the secondary X2 (top) axis.
YAXIS	Specifies whether data are mapped to the primary Y (left) axis or the secondary Y2 (right) axis.

COLORMODEL=*style-element*

specifies a style element that is used to determine the colors of filled or gradient contours.

Default: The ThreeColorRamp style element.

style-element

Name of a style element. The style element can contain these style attributes:

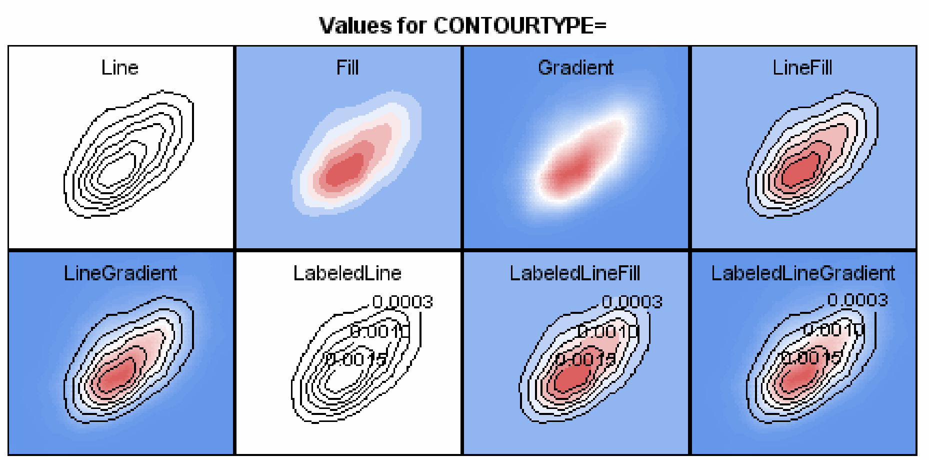
STARTCOLOR	Specifies a color for the smallest data value of the Z variable.
NEUTRALCOLOR	Specifies a color for the midpoint of the range of the Z variable.
ENDCOLOR	Specifies a color for the highest data value of the Z variable.

Interaction: The REVERSECOLORMODEL= option can be used to reverse the start and end colors of the ramp assigned to the color model.

Interaction: The NEUTRALCOLOR attribute is not used for two-color ramps.

CONTOURTYPE= LINE | LABELEDLINE | FILL | GRADIENT | LINEFILL | LINEGRADIENT | LABELEDLINEFILL | LABELEDLINEGRADIENT

specifies how the contour is displayed.



Default: The GraphContour:DisplayOpts style reference.

LINE
displays contour levels as unlabeled lines.

FILL
displays the area between the contour levels as filled. Each contour interval is filled with one color.

GRADIENT
displays a smooth gradient of color to represent contour levels.

LINEFILL
combines the LINE and FILL types. Each contour interval is filled with one color. Displays contour levels as unlabeled lines.

LINEGRADIENT
combines the LINE and GRADIENT types. Displays contour levels as unlabeled lines.

LABELEDLINE
adds labels to the LINE type, displaying contour levels as labeled lines.

LABELEDLINEFILL
adds labels to the LINEFILL type. Each contour interval is filled with one color. Displays contour levels as lines with labels showing contour level values.

LABELEDLINEGRADIENT
adds labels to the LINEGRADIENT type. Displays contour levels as lines with labels showing contour level values.

Interaction: The fill colors of the types that enable FILL or GRADIENT are controlled by the COLORMODEL= option.

Interaction: The line properties of the types that enable LINE or LABELEDLINE are controlled by the LINEATTRS= option.

Interaction: The label properties of the types that enable LABELEDLINE are controlled by the LINELABELATTRS= and LINELABELBASELINE= options.

Interaction: If a DISCRETELEGEND statement is associated with the contour, the legend is NOT displayed if CONTOURTYPE= is set to FILL or GRADIENT.

Interaction: If a CONTINUOUSLEGEND statement is associated with the contour, the legend is NOT displayed if CONTOURTYPE is set to LINE or LABELEDLINE.

GRIDDED=*boolean*
specifies whether the grid formed by the X and Y values is complete.

Default: TRUE

If set to FALSE, then additional calculations are performed.

For information about the algorithm used to calculate the grid, see Barnes, Stanley L., "Mesoscale Objective Map Analysis Using Weighted Time-Series Observations," United States National Oceanic and Atmospheric Administration, Environmental Research Labs, Norman, OK, *Technical Memorandum (NOAA TM ERLNSSL-62)*, March 1973.

LEGENDLABEL= *"string"*
specifies a label for use in a legend.

Default: The Z-variable label. If a label is not defined, the Z-variable name.

Restriction: This option applies only to an associated DISCRETELEGEND statement.

LINEATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)
specifies the attributes of the contour lines. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Line Options" on page 564 for available *line-options*.

Default: The GraphDataDefault style element.

Interaction: This option is honored only if the CONTOURTYPE= displays lines.

LINELABELATTRS=*style-element* | *style-element* (*text-options*) | (*text-options*)
specifies the color and font attributes of the contour line labels. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Text Options" on page 566 for available *text-options*.

Default: The GraphValueText style element.

Interaction: This option is honored only if the CONTOURTYPE= displays labels.

LINELABELBASELINE=HORIZONTAL | TANGENT
specifies the text alignment of the contour line labels. Each contour line has a precomputed label point.

Default: HORIZONTAL

HORIZONTAL
specifies that each label is parallel to the X-axis. The label intersects its contour line and is centered at the label point.

TANGENT
specifies that each label is drawn tangent to the contour line at the label point. This reduces intersection with the contour line and causes labels to be tilted at various angles in relation to the X-axis.

Interaction: This option is honored only if the CONTOURTYPE= displays labels.

LINELABELFORMAT=*format*
specifies the format to use for the contour line labels.

Default: The format associated with the Z variable or BEST6. if no format is assigned.

Interaction: This option is honored only if the CONTOURTYPE= displays labels.

LINELABELPOSITION=MIDDLE | BEGIN | END
specifies the position for the contour line labels.

Default: MIDDLE

Interaction: This option is honored only if the CONTOURTYPE= displays labels.

NAME=*"string"*
assigns a name to a plot statement for reference in other template statements.

Default: no default

Restriction: The *string* is case sensitive, cannot contain spaces, and must define a unique name within the template.

The specified name is used primarily in legend statements to indicate the colors associated with the Z values.

NHINT=*integer*
specifies the suggested number of contour levels for the Z variable.

Default: 7

The actual number of levels is adjusted to provide an appropriate number of levels for the data.

Interaction: This option is ignored if NLEVELS= is specified.

NLEVELS=*integer*
specifies the actual number of contour levels for the Z variable.

Default: The number of levels is determined internally, using the NHINT= value.

Interaction: This option overrides the NHINT= option.

Interaction: This option is ignored if CONTOURTYPE=GRADIENT.

PRIMARY=*boolean*
specifies that the data columns for this plot and the plot type be used for determining default axis features.

Default: FALSE

Restriction: This option is ignored if the plot is placed under a GRIDDED or LATTICE layout block.

Details: This option is needed only when two or more plots within an overlay-type layout contribute to a common axis. For more information, see "When Plots Share Data and a Common Axis" on page 397

REVERSECOLORMODEL=*boolean*
specifies whether to reverse the gradient (color ramp) defined by the COLORMODEL= option.

Default: FALSE

XAXIS=X | X2
specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

Default: X

Interaction: The overall plot specification and the layout type determine the axis display. For more information, see "How Axis Features are Determined" on page 395.

YAXIS=Y | Y2
specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

Default: Y

Interaction: The overall plot specification and the layout type determine the axis display. For more information, see "How Axis Features are Determined" on page 395.

Chapter 21
DENSITYPLOT Statement

Description and Syntax

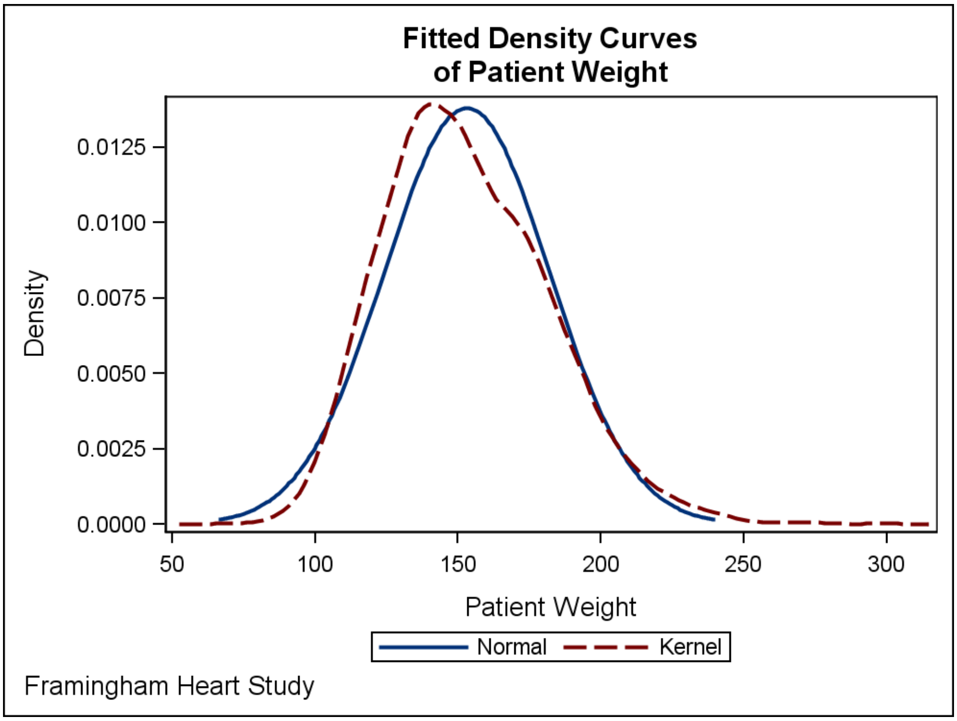
Creates a univariate probability density curve computed from input data.

DENSITYPLOT *numeric-column* | *expression* </ <*distribution-option*> </*option(s)*>>;

Example Program and Statement Details

Example Graph

The following graph was generated by the "Example Program" on page 214:

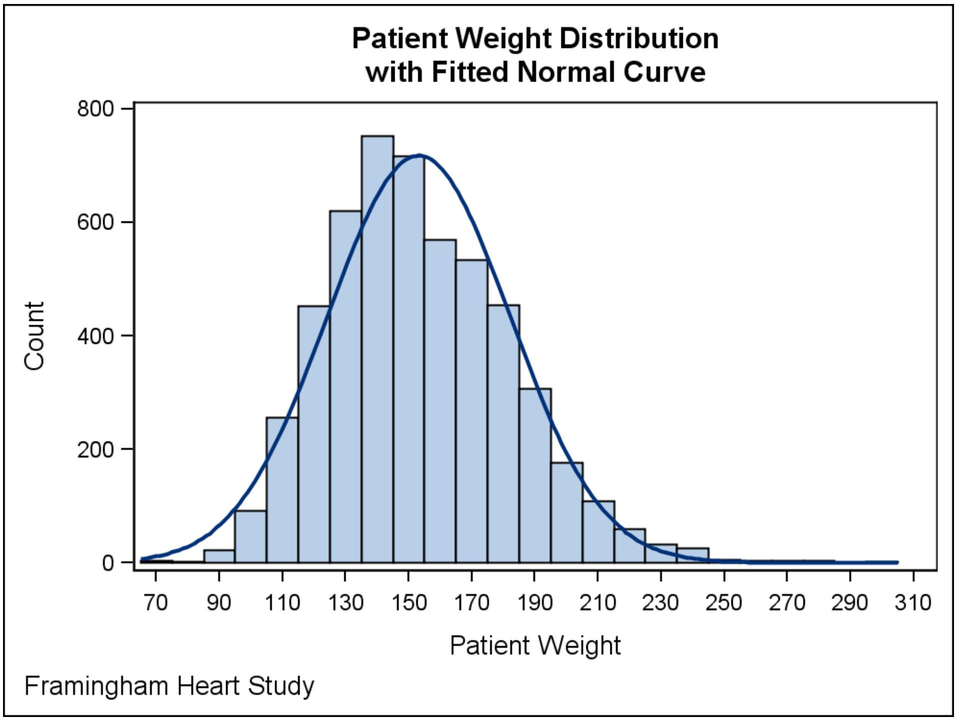


Example Program

When used as a stand-alone plot or overlaid with other DENSITYPLOTs, the dependent axis shows the computed density values.

```
proc template;
  define statgraph densityplot1;
    begingraph;
      entrytitle "Fitted Density Curves";
      entrytitle "of Patient Weight";
      entryfootnote halign=left "Framingham Heart Study";
      layout overlay;
        densityplot weight / normal()
         lineattrs=graphfit name="n" legendlabel="Normal";
        densityplot weight / kernel()
         lineattrs=graphfit2 name="k" legendlabel="Kernel";
        discretelegend "n" "k";
      endlayout;
    endgraph;
  end;
run;
proc sgrender data=sashelp.heart template=densityplot1;
  label weight="Patient Weight";
run;
```

When one or more DENSITYPLOTs are overlaid on a HISTOGRAM, the dependent axis shows the statistic indicated by the HISTOGRAM's SCALE= option. The area under each density curve is equal to the area of the HISTOGRAM.



```
proc template;
  define statgraph densityplot2;
    begingraph;
      entrytitle "Patient Weight Distribution";
      entrytitle "with Fitted Normal Curve";
      entryfootnote halign=left "Framingham Heart Study";
      layout overlay;
        histogram weight / primary=true scale=count;
        densityplot weight / normal() lineattrs=graphfit;
      endlayout;
    endgraph;
  end;
run;
proc sgrender data=sashelp.heart template=densityplot2;
  label weight="Patient Weight";
run;
```

Statement Summary

A typical DENSITYPLOT statement specifies either the NORMAL or the KERNEL distribution option. If no distribution option is specified, NORMAL() is used. The following syntax explicitly shows the default case:

```
DENSITYPLOT numeric-column / NORMAL()
```

To specify a KERNEL distribution, use the following plot syntax:

```
DENSITYPLOT numeric-column / KERNEL()
```

If more than one distribution option is specified, then the last distribution option specified is used.

Required Arguments

numeric-column

specifies a numeric column of data values that are used to calculate the parameters for the probability distribution.

expression

specifies an expression that calculates values when those values are not stored in the data.

Distribution Options

KERNEL (C=*number* WEIGHT=NORMAL | QUADRATIC | TRIANGULAR MAXPOINTS=*number*)

specifies a nonparametric kernel density estimate. The general form of the kernel density estimator is as follows.

$$\hat{f}_\lambda(x) = \frac{100h\%}{n\lambda} \sum_{i=1}^{n} K_0\left(\frac{x - x_i}{\lambda}\right)$$

In the equation, $K_0(\cdot)$ is the weight function, λ is the bandwidth, λ is the sample size, and x_i is the *i*th observation. You can use the C= suboption to specify the bandwidth and the WEIGHT= suboption to specify the weight function $K_0(\cdot)$.

For more information, see the discussion of Kernel Density Estimates for the UNIVARIATE procedure in the *Base SAS Procedures Guide* , Volume 3.

C=

specifies a positive number (0 < *number* <= 100) that represents the *standardized bandwidth*.

Default: Calculated from the data as the bandwidth that minimizes the approximate mean integrated square error (MISE).

The value of λ, referred to as the bandwidth parameter, determines the degree of smoothness in the estimated density function. You specify λ indirectly by specifying a standardized bandwidth *c* with the C= *kernel-option*. If *Q* is the interquartile range and *n* is the sample size, then *c* is related to λ by the following formula:

$$\lambda = cQn - \frac{1}{5}$$

WEIGHT=

specifies one of the weight functions NORMAL, QUADRATIC, or TRIANGULAR.

Default: NORMAL

The formulas for the weight functions are

Normal	$K_0(t) = \frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2}t^2)$	for $-\infty < t < \infty$
Quadratic	$K_0(t) = \frac{3}{4}(1 - t^2)$	for $\lvert t \rvert \leq 1$
Triangular	$K_0(t) = 1 - \lvert t \rvert$	for $\lvert t \rvert \leq 1$

MAXPOINTS=

specifies the maximum number of points generated for the curve.

Default: 512

NORMAL (MU=number SIGMA=number MAXPOINTS=number)

specifies a normal density estimate, with mean and standard deviation. The fitted density function equation is as follows:

$$p(x) = \frac{100h\%}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right) \quad \text{for } -\infty < x < \infty$$

In the equation, μ is the mean and σ is the standard deviation ($\sigma > 0$). You can specify μ with the MU= suboption and σ with the SIGMA= suboption. By default, ODS estimates μ with the sample mean and σ with the sample standard deviation.

For more information, see the discussion of Kernel Density Estimates for the UNIVARIATE procedure in the *Base SAS Procedures Guide* , Volume 3.

MU=

specifies the mean. If not specified, it is calculated from the data.

SIGMA=

specifies the standard deviation. If not specified, it is calculated from the data.

MAXPOINTS=

specifies the maximum number of points generated for the curve.

Default: 200

Options

Statement Option	Description
CURVELABEL	Specifies a label for the density curve.
CURVELABELATTRS	Specifies the color and font attributes of the density curve label.
CURVELABELLOCATION	Specifies the location of the curve label relative to the plot area.
CURVELABELPOSITION	Specifies the position of the density curve label relative to the curve line.
DATATRANSPARENCY	Specifies the degree of the transparency of the density curve.

Statement Option	Description
FREQ	Specifies a numeric column that provides frequencies for each observation read.
LEGENDLABEL	Specifies the label of the legend.
LINEATTRS	Specifies the properties of the density curve.
NAME	Assigns a name to a plot statement for reference in other template statements.
ORIENT	Specifies the orientation of the Y axis.
PRIMARY	Specifies that the data columns for this plot and the plot type be used for determining default axis features.
TIPFORMAT	Specifies display formats for the tooltips that are displayed when the cursor is positioned over the density curve.
TIPLABEL	Specifies display labels for the tooltips that are displayed when the cursor is positioned over the density curve.
XAXIS	Specifies whether data are mapped to the primary X (bottom) axis or the secondary X2 (top) axis.
YAXIS	Specifies whether data are mapped to the primary Y (left) axis or the secondary Y2 (right) axis.

CURVELABEL=*"string"*
specifies a label for the density curve.

Default: no default

The font and color attributes for the label are specified by the CURVELABELATTRS= option.

CURVELABELATTRS=*style-element* | *style-element* (*text-options*) | (*text-options*)
specifies the color and font attributes of the density curve label. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Text Options" on page 566 for available *text-options*.

Default: The GraphValueText style element.

Interaction: For this option to take effect, the CURVELABEL=curvelabel option must also be used.

CURVELABELLOCATION=INSIDE | OUTSIDE
specifies the location of the density curve label relative to the plot area.

Default: INSIDE

INSIDE
inside the plot area

OUTSIDE
outside the plot area

Restriction: OUTSIDE cannot be used when the DENSITYPLOT is used in multi-cell layouts such as LATTICE, DATAPANEL, or DATALATTICE, where axes can be external to the grid.

Interaction: This option is used in conjunction with the CURVELABELPOSITION= option to determine where the curve label appears. For more information, see "Location and Position of Curve Labels" on page 123.

CURVELABELPOSITION=AUTO | MAX | MIN | START | END
Specifies the position of the density curve label relative to the curve line.

Default: AUTO when CURVELABELLOCATION=OUTSIDE. END when CURVELABELLOCATION=INSIDE.

AUTO
Only used when CURVELABELPOSITION=OUTSIDE. The density label is positioned automatically near the end of the density curve along unused axes whenever possible (typically Y2 or X2) to avoid collision with tick values.

MAX
Forces the density label to appear near maximum density X- values (typically, to the right).

MIN
Forces the density label to appear near minimum density X- values (typically, to the left).

START
Only used when CURVELABELLOCATION=INSIDE. Forces density label to appear near the beginning of the curve. Particularly useful when the curve line has a spiral shape.

END
Only used when CURVELABELLOCATION=INSIDE. Forces density label to appear near the end of the curve. Particularly useful when the curve line has a spiral shape.

Restriction: The AUTO setting is not supported if CURVELABELLOCATION=INSIDE is specified. The START and END settings are not supported if CURVELABELLOCATION=OUTSIDE is specified.

Interaction: For this option to take effect, the CURVELABEL= option must also be specified.

Interaction: This option is used in conjunction with the CURVELABELLOCATION= option to determine where the density label appears. For more information, see "Location and Position of Curve Labels" on page 123.

DATATRANSPARENCY=*number*
specifies the degree of the transparency of the density curve.

Default: 0

Range: 0 (opaque) to 1 (entirely transparent)

FREQ=*numeric-column* | *expression*
specifies a numeric column that provides frequencies for each observation read. If *n* is the value of the *numeric-column* for a given observation, then that observation is used *n* times in computing the density curve.

Default: Each observation is counted once.

Restriction: If the value of the *numeric-column* is missing or is less than 1, the observation is not used in the analysis. If the value is not an integer, only the integer portion is used.

LEGENDLABEL= *"string"*

specifies a label to be used in a legend.

Default: The *string* specified on the NAME= option.

LINEATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)

specifies the attributes of the density curve. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Line Options" on page 564 for available *line-options*.

Default: The GraphFitLine style element.

NAME=*"string"*

assigns a name to a plot statement for reference in other template statements.

Default: no default

Restriction: The *string* is case sensitive, cannot contain spaces, and must define a unique name within the template.

Interaction: The *string* is used as the default legend label if the LEGENDLABEL= option is not used.

The specified name is used primarily in legend statements to coordinate the use of colors and line patterns between the graph and the legend.

ORIENT=VERTICAL | HORIZONTAL

specifies the orientation of the Y axis.

Default: VERTICAL

PRIMARY=*boolean*

specifies that the data columns for this plot and the plot type be used for determining default axis features.

Default: FALSE

Restriction: This option is ignored if the plot is placed under a GRIDDED or LATTICE layout block.

Details: This option is needed only when two or more plots within an overlay-type layout contribute to a common axis. For more information, see "When Plots Share Data and a Common Axis" on page 397

TIPFORMAT=(*role-format-list*)

specifies display formats for the tooltips that are displayed when the cursor is positioned over the density curve.

Default: The column format of the variable assigned to the role or BEST6. if no format is assigned to a numeric column.

(*role-format-list*)

a list of *role-name* = *format* pairs separated by blanks.

Use *role-name* X to format the tooltip values that map to the values along the X axis, and *role-name* Y to format the tooltip values that map to the values along the Y axis.

```
TIPFORMAT=(Y=6.2)
```

Tip: This option provides a way to control the formats of columns that appear in tooltips.

TIPLABEL=(*role-label-list*)

specifies display labels for the tooltips that are displayed when the cursor is positioned over the density curve.

Default: The column label or column name of the variable assigned to the role.

(*role-label-list*)

a list of *role-name = "string"* pairs separated by blanks.

Use *role-name* X to label the tooltip values that map to the values along the X axis, and *role-name* Y to label the tooltip values that map to the values along the Y axis.

```
TIPLABEL=(Y="Normal Approximation")
```

Tip: This option provides a way to control the labels of columns that appear in tooltips.

XAXIS=X | X2

specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

Default: X

Interaction: The overall plot specification and the layout type determine the axis display for the specified axis. For more information, see "How Axis Features are Determined" on page 395.

YAXIS=Y | Y2

specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

Default: Y

Interaction: The overall plot specification and the layout type determine the axis display for the specified axis. For more information, see "How Axis Features are Determined" on page 395.

Chapter 22
DROPLINE Statement

Description and Syntax

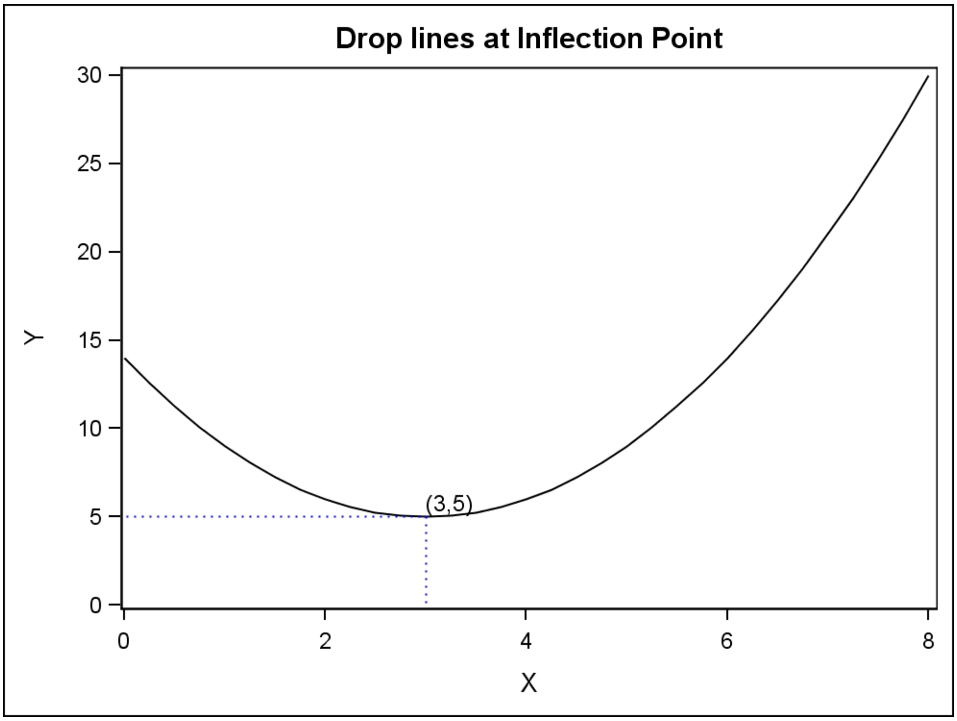
Creates a horizontal or vertical drop line from a point to an axis.

DROPLINE X = *x-axis-value* | *column* | *expression*
Y = *y-axis-value* | *column* | *expression* *</option(s)>*;

Example Program and Statement Details

Example Graph

The following graph was generated by the “Example Program” on page 224:



Example Program

```
proc template;
  define statgraph dropline;
    begingraph;
      entrytitle "Drop lines at Inflection Point";
      layout overlay / yaxisopts=(linearopts=(viewmin=0));
        seriesplot x=x y=y;
        dropline x=3 y=5 / dropto=x
         lineattrs=(color=blue pattern=dot) label="(3,5)";
        dropline x=3 y=5 / dropto=y
         lineattrs=(color=blue pattern=dot);
      endlayout;
   endgraph;
  end;
run;

data test;
  do X=0 to 8 by .25;
    Y=(x-3)*(x-3) + 5;
      output;
  end;
run;

proc sgrender data=test template=dropline;
run;
```

Statement Summary

A drop line is always drawn perpendicular from the specified point to the X (bottom), X2 (top), Y (left), or Y2 (right) axis. Axis offsets do not apply to drop lines, so they always meet the axis line.

The DROPTO= option controls whether a horizontal or vertical drop line is created. DROPTO=X specifies the horizontal axis for a vertical drop line, and DROPTO=Y specifies the vertical axis for a horizontal drop line.

A DROPLINE statement can be only used within a 2-D overlay-type layout (OVERLAY, OVERLAYEQUATED, PROTOTYPE). Another plot statement that is derived from data values that provide boundaries for the axis area must be included in the layout. For example, it can be used with a scatter plot or a histogram.

You can generate multiple drop lines by specifying a column for X and Y. The column type (numeric or string) must agree with the type of data presented on the axis.

To generate both a vertical and a horizontal drop line from a single point, use multiple DROPLINE statements. For example, the graph in the figure illustrates two DROPLINE statements originating from the same point (X=3, Y=5). One statement uses DROPTO=X and the other uses DROPTO=Y.

Required Arguments

X=*x-axis-value* | *column* | *expression*
specifies the X coordinate of the drop line(s).

Requirement: Values must agree in type with the X-axis data type. For example, you should use numeric SAS date or time values (or SAS date/time constants) for a time axis.

By default, if a specified value is outside of the X-axis data range, then the data range is extended to include the value. This behavior can be changed with the CLIP= option.

Y=*y-axis-value* | *column* | *expression*
specifies the Y coordinate of the drop line(s).

Requirement: Values must agree in type with the Y-axis data type.

By default, if a specified value is outside of the Y-axis data range, then the data range is extended to include the value. This behavior can be changed with the CLIP= option.

Options

Statement Option	**Description**
CLIP	Specifies whether the data for the drop line(s) are considered when determining the data ranges for the axes.
DATATRANSPARENCY	Specifies the degree of the transparency of the drop line(s).
DISCRETEOFFSET	Specifies an amount to offset all drop lines from discrete X values, or discrete Y values, or both.

Statement Option	Description
DROPTO	Specifies the axis to which the line is dropped.
LABEL	Specifies a label for the point(s) indicated by X= and Y= options.
LABELATTRS	Specifies the color and font attributes of the drop line label.
LEGENDLABEL	Specifies a label for use in a legend.
LINEATTRS	Specifies the properties of the drop line(s).
NAME	Assigns a name to a plot statement for reference in other template statements.
XAXIS	Specifies whether data are mapped to the primary X (bottom) axis or the secondary X2 (top) axis.
YAXIS	Specifies whether data are mapped to the primary Y (left) axis or the secondary Y2 (right) axis.

CLIP=*boolean*

specifies whether the data for the line are considered when determining the data ranges for the axes.

Default: FALSE

FALSE

The data for the line contribute to the data range for each axis. Each axis might be extended to force the display of the line.

TRUE

The data for the line are ignored when establishing axis scales. Each axis scale is determined by the other plots in the parent layout. This might result in the line not being displayed if its data range is not within the data ranges of the other plots.

DATATRANSPARENCY=*number*

specifies the degree of the transparency of the drop line.

Default: 0

Range: 0 (opaque) to 1 (entirely transparent)

DISCRETEOFFSET= *number*

specifies an amount to offset all drop lines from the X values for a discrete X axis, or from the Y values for a discrete Y axis.

Note: This feature is for the third maintenance release of SAS 9.2 and later.

Default: 0 (no offset, all drop lines are centered on X values for a discrete X axis, or Y values for a discrete Y axis)

Range: -0.5 to +0.5 where .5 represents half the distance between discrete ticks. If the X axis is discrete, a positive offset is to the right. If the Y axis is discrete, a positive offset is up. If REVERSE=TRUE on the X or Y axis, then the offset direction is also reversed.

DROPTO=X | Y

specifies the axis to which the line is dropped.

Default: X

X

Draws dropline(s) to an X axis. The XAXIS== option determines whether the X or X2 axis is the endpoint for the line.

Y

Draws dropline(s) to a Y axis. The YAXIS== option determines whether the Y or Y2 axis is the endpoint for the line.

LABEL=*"string"* | *column*

specifies a label for the point(s) indicated by the X= and Y= arguments.

Default: no default

Interaction: If drawing multiple droplines using *X=column* or *Y=column*, you can assign corresponding labels by using a *column* to define the labels.

The font and color attributes for the label are specified by the LABELATTRS= option.

LABELATTRS=*style-element* | *style-element* (*text-options*) | (*text-options*)

specifies the color and font attributes of the drop line label(s). See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Text Options" on page 566 for available *text-options*.

Default: The GraphValueText style element.

Interaction: For this option to have any effect, the LABEL= option must also be specified.

LEGENDLABEL= *"string"*

specifies a label for use in a legend.

Default: The **string** specified on the NAME= option.

LINEATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)

specifies the attributes of the drop line. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Line Options" on page 564 for available *line-options*.

Default: The GraphReference style element.

NAME=*"string"*

assigns a name to a plot statement for reference in other template statements.

Default: no default

Restriction: The *string* is case sensitive, cannot contain spaces, and must define a unique name within the template.

Interaction: The *string* is used as the default legend label if the LEGENDLABEL= option is not used.

The specified name is used primarily in legend statements to coordinate the use of colors and line patterns between the graph and the legend.

XAXIS=X | X2

specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

Default: X

Restriction: Another plot that establishes a data range for the designed axis must be included.

Interaction: The overall plot specification and the layout type determine the axis display. For more information, see "How Axis Features are Determined" on page 395.

YAXIS=Y | Y2

specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

Default: Y

Restriction: Another plot that establishes a data range for the designed axis must be included.

Interaction: The overall plot specification and the layout type determine the axis display. For more information, see "How Axis Features are Determined" on page 395.

Chapter 23
ELLIPSE Statement

Description and Syntax

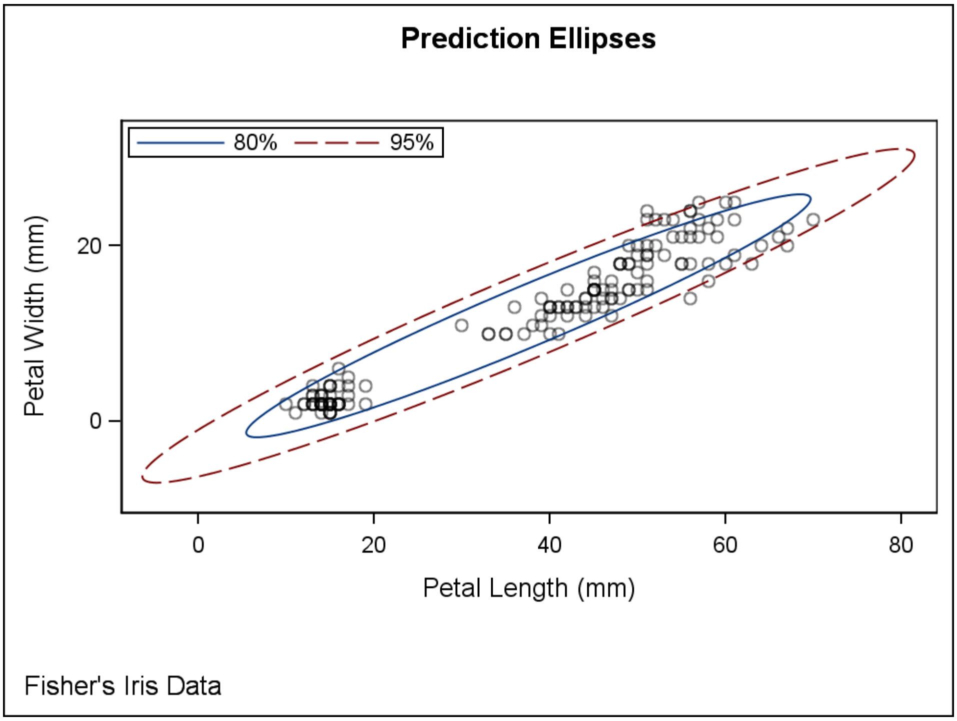
Creates a confidence ellipse computed from input data.

ELLIPSE X = *numeric-column* | *expression*
Y = *numeric-column* | *expression* <*/option(s)*>;

Example Program and Statement Details

Example Graph

The following graph was generated by the "Example Program" on page 230:



Example Program

This example overlays two ELLIPSE statements on a SCATTERPLOT of the same data.

- Both ELLIPSE statements use TYPE=PREDICTED.
- One ELLIPSE statement uses ALPHA=.2 and the other uses ALPHA=.05.

```
proc template;
  define statgraph ellipse;
    begingraph;
      entrytitle "Prediction Ellipses";
      layout overlayequated / equatetype=equate;
        scatterplot x=petallength y=petalwidth /
          datatransparency=.5;
        ellipse x=petallength y=petalwidth /
          type=predicted  alpha=.2
          name="p80" legendlabel="80%"
          outlineattrs=graphconfidence;
        ellipse x=petallength y=petalwidth /
          type=predicted  alpha=.05
          name="p95" legendlabel="95%"
          outlineattrs=graphconfidence2;
        discretelegend "p80" "p95" /
          location=inside autoalign=(topleft);
      endlayout;
      entryfootnote halign=left "Fisher's Iris Data";
    endgraph;
  end;
run;
proc sgrender data=sashelp.iris template=ellipse;
run;
```

Statement Summary

The ELLIPSE statement can be used only within 2-D overlay-type layouts. It computes an ellipse for a set of points specified by the X and Y columns and a confidence level specified by the ALPHA= option. Use the TYPE= option to control whether a predicted or confidence ellipse is generated.

Confidence and Prediction Ellipses

Two types of ellipses can be computed for the input data (where observations correspond to points in a scatter plot). One is a confidence ellipse for the population mean (TYPE=MEAN), and the other is a prediction ellipse for a new observation (TYPE=PREDICT). Both assume a bivariate normal distribution.

Let $\bar{Z}$ and S be the sample mean and sample covariance matrix of a random sample of size *n* from a bivariate normal distribution with mean μ and covariance matrix Σ. The variable $\bar{Z} - \mu$ is distributed as a bivariate normal variate with mean zero and covariance $(\frac{1}{n})\Sigma$, and it is independent of S. Using Hotelling's T^2 statistic, which is defined as

$$T^2 = n(\bar{Z} - \mu)'\, S^{-1}(\bar{Z} - \mu)$$

a $100(1 - \alpha)\ \%$ confidence ellipse for μ is computed from the equation

$$\frac{n}{n-1}(\bar{Z} - \mu)'\, S^{-1}(\bar{Z} - \mu) = \frac{2}{n-2}F_{2,n-2}(1 - \alpha)$$

where $F_{2,n-2}(1 - \alpha)$ is the $(1 - \alpha)$ critical value of an F distribution with degrees of freedom 2 and $n - 2$.

A prediction ellipse is a region for predicting a new observation in the population. It also approximates a region containing a specified percentage of the population.

Denote a new observation as the bivariate random variable Z_{new}. The variable

$$Z_{new} - \bar{Z} = (Z_{new} - \mu) - (\bar{Z} - \mu)$$

is distributed as a bivariate normal variate with mean zero (the zero vector) and covariance $(1 + \frac{1}{n})\Sigma$, and it is independent of S. A $100(1 - \alpha)\ \%$ prediction ellipse is then given by the equation

$$\frac{n}{n-1}(\bar{Z} - \mu)'\, S^{-1}(\bar{Z} - \mu) = \frac{2(n+1)}{n-2}F_{2,n-2}(1 - \alpha)$$

The family of ellipses generated by different critical values of the F distribution has a common center (the sample mean) and common major and minor axis directions.

The shape of an ellipse depends on the aspect ratio of the plot. The ellipse indicates the correlation between the two variables if the variables are standardized (by dividing the variables by their respective standard deviations). In this situation, the ratio between the major and minor axis lengths is

$$\sqrt{\frac{1 + |r|}{1 - |r|}}$$

In particular, if $r = 0$, the ratio is 1, which corresponds to a circular confidence contour and indicates that the variables are uncorrelated. A larger value of the ratio indicates a larger positive or negative correlation between the variables.

Required Arguments

X=*numeric-column* | *expression*
specifies the numeric column for the X values.

Y=*numeric-column* | *expression*
specifies the numeric column for the Y values.

Options

Statement Option	Description
ALPHA	Sets a significance value for the confidence level to compute for the ellipse.
CLIP	Specifies whether the data for the ellipse are considered when determining the data ranges for the axes.
DATATRANSPARENCY	Specifies the degree of the transparency of the ellipse fill color and outline.
DISPLAY	Specifies whether to display an outlined ellipse, a filled ellipse, or an outlined and filled ellipse.
FILLATTRS	Specifies the appearance of the interior fill of the ellipse.
FREQ	Specifies a numeric column that provides frequencies for each observation read.
LEGENDLABEL	Specifies the legend label.
NAME	Assigns a name to a plot statement for reference in other template statements.
OUTLINEATTRS	Specifies the properties of the ellipse outline.
TYPE	Specifies the type of ellipse.
XAXIS	Specifies whether data are mapped to the primary X (bottom) axis or the secondary X2 (top) axis.
YAXIS	Specifies whether data are mapped to the primary Y (left) axis or the secondary Y2 (right) axis.

ALPHA=*positive-number*
sets a significance value for the confidence level to compute for the ellipse.

Default: .05

Range: 0 < *number* < 1

ALPHA=.05 represents a 95% confidence level.

See also: TYPE= option

CLIP=*boolean*

specifies whether the data for the ellipse are considered when determining the data ranges for the axes.

Default: FALSE

FALSE

The data for the ellipse contribute to the data range for each axis. Each axis might be extended to force the display of the entire ellipse.

TRUE

The data for the ellipse are ignored when establishing axis scales. Each axis scale is determined by the other plots in the parent layout. This might result in the ellipse not being entirely displayed (clipped) if its data range is not within the data ranges of the other plots.

DATATRANSPARENCY=*number*

specifies the degree of the transparency of the ellipse fill color and outline.

Default: 0

Range: 0 (opaque) to 1 (entirely transparent)

DISPLAY=STANDARD | ALL | (*display-options*)

specifies whether to display an outlined ellipse, a filled ellipse, or an outlined and filled ellipse.

Default: GraphEllipse:DisplayOpts style reference.

STANDARD

displays an outlined, unfilled ellipse

ALL

displays an outlined, filled ellipse

(*display-options*)

a list of features to be displayed. The list must be enclosed in parentheses and must include one of the following:

OUTLINE—displays an outlined ellipse

FILL—displays a filled ellipse

Use FILLATTRS= and OUTLINEATTRS= to control the appearance of the ellipse.

FILLATTRS=*style-element* | *style-element* (*fill-options*) | (*fill-options*)

specifies the appearance of the interior fill area of the ellipse. See “General Syntax for Attribute Options” on page 563 for the syntax on using a *style-element* and “Fill Options” on page 564 for available *fill-options*.

Default: The GraphDataDefault style element.

Interaction: For this option to have any effect, the fill must be enabled by the ODS style or the DISPLAY= option.

FREQ=*numeric-column* | *expression*

Specifies a numeric column that provides frequencies for each observation read. If **n** is the value of the FREQ variable for a given observation, then that observation is counted **n** times in computing the ellipse.

Default: Each observation is counted once.

Restriction: If the value of the *numeric-column* is missing or is less than 1, the observation is not used in the analysis. If the value is not an integer, only the integer portion is used.

LEGENDLABEL= *"string"*
specifies the legend label.

Default: The *string* specified on the NAME= option.

NAME=*"string"*
assigns a name to a plot statement for reference in other template statements.

Default: no default

Restriction: The *string* is case sensitive, cannot contain spaces, and must define a unique name within the template.

Interaction: The *string* is used as the default legend label if the LEGENDLABEL= option is not used.

The specified name is used primarily in legend statements to coordinate the use of colors and line patterns between the graph and the legend.

OUTLINEATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)
specifies the attributes of the ellipse outline. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Line Options" on page 564 for available *line-options*.

Default: The GraphDataDefault style element.

Interaction: For this option to have any effect, the outlines must be enabled by the ODS style or the DISPLAY= option.

TYPE=MEAN | PREDICTED
specifies the type of ellipse.

Default: MEAN

MEAN
specifies a confidence ellipse of the mean

PREDICTED
specifies a prediction ellipse for a new observation

See also: ALPHA= option for specifying a confidence level.

See also: For statistical details about how the ellipse is calculated, see "Confidence and Prediction Ellipses " on page 231.

XAXIS=X | X2
specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

Default: X

Interaction: The overall plot specification and the layout type determine the axis display. For more information, see "How Axis Features are Determined" on page 395.

YAXIS=Y | Y2
specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

Default: Y

Interaction: The overall plot specification and the layout type determine the axis display. For more information, see “How Axis Features are Determined” on page 395.

Chapter 24
ELLIPSEPARM Statement

Description and Syntax

Creates an ellipse specified by slope, axis, and origin parameters.

ELLIPSEPARM SEMIMAJOR = *number* | *numeric-column* | *expression*

SEMIMINOR = *number* | *numeric-column* | *expression*

SLOPE = *number* | *numeric-column* | *expression*

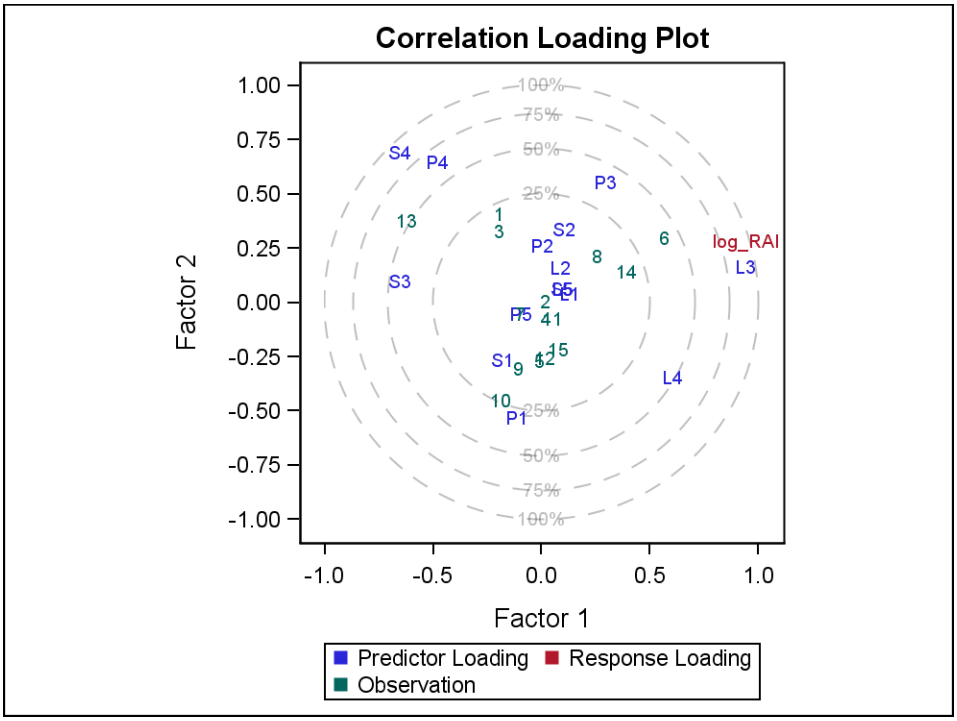
XORIGIN = *number* | *numeric-column* | *expression*

YORIGIN = *number* | *numeric-column* | *expression* </*option(s)*>;

Example Program and Statement Details

Example Graph

The following graph was generated by the “Example Program” on page 238:



Example Program

This is a simplified version of the CorrLoadPlot template for PROC PLS in the SAS/STAT product. It consists of overlaid scatter plots of the scores of the first two factors, the loadings of the model effects, and the loadings of the dependent variables. The loadings are scaled so that the amount of variation in the variables that is explained by the model is proportional to the distance from the origin; circles indicating various levels of explained variation are also overlaid.

The circles are drawn with ELLIPSEPARM statements by setting the SEMIMAJOR and SEMIMINOR lengths to be the same. Notice that all circles are concentric because they share the same origin (0,0). The radius of each circle is passed dynamically.

Also note that an OVERLAYEQUATED layout was used to force the length of unit intervals on both axes to be the same.

The input data shown is representative of that computed by PROC PLS for the Correlation Loadings Plot. For more details, see the first example for PROC PLS in the **MISSING Bookref** .

```
proc template;
 define statgraph ellipseparm;
  dynamic RADIUS1 RADIUS2 RADIUS3 RADIUS4 ;
  begingraph;
   entrytitle "Correlation Loading Plot";
   layout overlayequated / equatetype=square
     commonaxisopts=
      (tickvaluelist=(-1 -.75 -.5 -.25 0 .25 .5 .75 1)
        viewmin=-1 viewmax=1)
     xaxisopts=
      (label="Factor 1" offsetmin=0.05 offsetmax=0.05)
     yaxisopts=
      (label="Factor 2" offsetmin=0.05 offsetmax=0.05);
```

```
    ellipseparm semimajor=RADIUS1 semiminor=RADIUS1 slope=0
     xorigin=0 yorigin=0 / clip=true display=(outline)
     outlineattrs=(pattern=dash) datatransparency=.75;
    ellipseparm semimajor=RADIUS2 semiminor=RADIUS2 slope=0
     xorigin=0 yorigin=0 / clip=true display=(outline)
     outlineattrs=(pattern=dash) datatransparency=.75;
    ellipseparm semimajor=RADIUS3 semiminor=RADIUS3 slope=0
     xorigin=0 yorigin=0 / clip=true display=(outline)
     outlineattrs=(pattern=dash) datatransparency=.75;
    ellipseparm semimajor=RADIUS4 semiminor=RADIUS4 slope=0
     xorigin=0 yorigin=0 / clip=true display=(outline)
     outlineattrs=(pattern=dash) datatransparency=.75;

    scatterplot x=xcirclelabel y=ycirclelabel / primary=true
     markercharacter=circlelabel datatransparency=0.75 ;
    scatterplot x=corr1 y=corr2 / name="ScatterVars"
     group=corrgroup markercharacter=corrlabel;
    discretelegend "ScatterVars";
   endlayout;
  endGraph;
 end;
run;

data corrplot;
  infile cards missover;
  input Corr1 Corr2 CorrGroup &$18. CorrLabel :$7.
        xCircleLabel  yCircleLabel CircleLabel :$8.;
datalines;
-0.179 -0.268 Predictor Loading  S1 0  0.5 25%
 0.105  0.332 Predictor Loading  S2 0 -0.5 25%
-0.654  0.094 Predictor Loading  S3 0  0.707 50%
-0.653  0.685 Predictor Loading  S4 0 -0.707 50%
 0.096  0.059 Predictor Loading  S5 0  0.866 75%
 0.132  0.036 Predictor Loading  L1 0 -0.866 75%
 0.087  0.156 Predictor Loading  L2 0  1 100%
 0.940  0.160 Predictor Loading  L3 0 -1 100%
 0.607 -0.350 Predictor Loading  L4
 0.096  0.059 Predictor Loading  L5
-0.111 -0.534 Predictor Loading  P1
 0.003  0.256 Predictor Loading  P2
 0.293  0.551 Predictor Loading  P3
-0.480  0.643 Predictor Loading  P4
-0.096 -0.059 Predictor Loading  P5
 0.946  0.279 Response Loading  log_RAI
-0.196  0.403 Observation  1
 0.020 -0.001 Observation  2
-0.195  0.324 Observation  3
 0.021 -0.079 Observation  4
-0.009 -0.274 Observation  5
 0.567  0.294 Observation  6
-0.096 -0.059 Observation  7
 0.258  0.210 Observation  8
-0.104 -0.309 Observation  9
-0.187 -0.458 Observation  10
 0.051 -0.078 Observation  11
 0.017 -0.260 Observation  12
```

```
-0.621  0.372 Observation  13
 0.392  0.138 Observation  14
 0.080 -0.221 Observation  15
run;

proc sgrender data=corrplot template=ellipseparm;
  dynamic radius1=.50 radius2=.71 radius3=.87 radius4=1;
run;
```

Statement Summary

The ELLIPSEPARM statement plots an ellipse with specified semimajor and semiminor axis lengths, a specified slope for the major axis, and the ellipse center. The ellipse center is the point of intersection of the semimajor and semiminor axes.

An ELLIPSEPARM plot does not perform computations on input data to derive the location and shape of the ellipse. See "ELLIPSE Statement" on page 229 for information about computed ellipses.

An ELLIPSEPARM statement can be used only within a 2-D overlay-type layout (for example, an OVERLAY, OVERLAYEQUATED, or PROTOTYPE layout). Another plot that is derived from data values that provide boundaries for the axis area must be included in the layout. It is typically overlaid with a scatter plot.

You can generate a single ellipse by specifying a constant for each required argument. You can generate multiple ellipses by specifying a numeric column for any or all required arguments. If any of the SEMIMAJOR=, SEMIMINOR=, XORIGIN= or YORIGIN= constants or columns contains a missing value, no ellipse is drawn. To request a vertical major axis, specify **SLOPE=.** (missing value as a constant or column value).

If the GROUP= option is specified, then the group value must be a column (character or numeric). For each group value, there must be a numeric column value that does not contain missing values for SEMIMAJOR=, SEMIMINOR=, XORIGIN=, and YORIGIN=. The SLOPE= column might contain missing values. Under these circumstances, one or more ellipse might be drawn for each group value.

Required Arguments

SEMIMAJOR=*number* | *numeric-column* | *expression*
specifies half the length of the major axis for the ellipse.

Requirement: The SEMIMAJOR value must be positive.

Values are in the units of the data. The SEMIMAJOR value can be greater than, smaller than, or equal to the SEMIMINOR value.

Interaction: If a numeric column is specified and the column contains missing or negative values, no ellipse is drawn.

SEMIMINOR=*number* | *numeric-column* | *expression*
specifies half the length of the minor axis for the ellipse.

Requirement: The SEMIMINOR value must be positive.

Values are in the units of the data. The SEMIMINOR value can be greater than, smaller than, or equal to the SEMIMAJOR value.

Interaction: If a numeric column is specified and the column contains missing or negative values, no ellipse is drawn.

SLOPE=*number* | *numeric-column* | *expression*
specifies the slope of the major axis for the ellipse.

Slope can be positive or negative.

Setting **`SLOPE=0`** creates a major axis parallel to the X-axis. Setting **`SLOPE=.`** (missing value) creates a major axis parallel to the Y-axis.

The slope value is in the data space and might or might not be maintained in the screen space. Thus, setting **`SLOPE=1`** does not always generate a 45 degree line on the screen.

XORIGIN=*number* | *numeric-column* | *expression*
specifies the X coordinate of the center of the ellipse.

Values are in the units of the data.

By default, if the value specified for the XORIGIN= option is outside of the X-axis data range, then the data range is extended to include the specified point. This behavior can be changed with the CLIP= option.

YORIGIN=*number* | *numeric-column* | *expression*
specifies the Y coordinate of the center of the ellipse.

Values are in the units of the data.

By default, if the value specified for the YORIGIN= option is outside of the Y-axis data range, then the data range is extended to include the specified point. This behavior can be changed with the CLIP= option.

Options

Statement Option	Description
CLIP	Specifies whether the data for the ellipse are considered when determining the data ranges for the axes.
DATATRANSPARENCY	Specifies the degree of the transparency of the ellipse fill color and outline.
DISPLAY	Specifies whether to display an outlined ellipse, a filled ellipse, or an outlined and filled ellipse.
FILLATTRS	Specifies the appearance of the interior fill of the ellipse.
GROUP	Creates a separate ellipse for each unique group value of the specified column.
INDEX	Specifies indices for mapping line attributes (color and pattern) to one of the GraphData1 - GranphDataN style elements.
LEGENDLABEL	Specifies a label for use in a legend.
NAME	Assigns a name to a plot statement so that it can be referenced in other template statements.

Statement Option	Description
OUTLINEATTRS	Specifies the properties of the ellipse outline.
XAXIS	Specifies whether data are mapped to the primary X (bottom) axis or the secondary X2 (top) axis.
YAXIS	Specifies whether data are mapped to the primary Y (left) axis or the secondary Y2 (right) axis.

CLIP=*boolean*

specifies whether the data for the ellipse are considered when determining the data ranges for the axes.

Default: FALSE

FALSE

The data for the ellipse contribute to the data range for each axis. Each axis might be extended to force the display of the entire ellipse.

TRUE

The data for the ellipse are ignored when establishing axis scales. Each axis scale is determined by the other plots in the parent layout. This might result in the ellipse not being entirely displayed (clipped) if its data range is not within the data ranges of the other plots.

DATATRANSPARENCY=*number*

specifies the degree of the transparency of the ellipse fill color and outline.

Default: 0

Range: 0 (opaque) to 1 (entirely transparent)

DISPLAY=STANDARD | ALL | (*display-options*)

specifies whether to display an outlined ellipse, a filled ellipse, or an outlined and filled ellipse.

Default: GraphEllipse:DisplayOpts style reference.

STANDARD

displays an outlined, unfilled ellipse

ALL

displays an outlined, filled ellipse

(*display-options*)

a list of features to be displayed. The list must be enclosed in parentheses and must include one of the following:

OUTLINE—displays an outlined ellipse

FILL—displays a filled ellipse

Use FILLATTRS= and OUTLINEATTRS= to control the appearance of the ellipse.

FILLATTRS=*style-element* | *style-element* (*fill-options*) | (*fill-options*)

specifies the appearance of the interior fill area of the ellipse. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Fill Options" on page 564 for available *fill-options*.

Default:

- For non-grouped data, the GraphDataDefault style element.
- For grouped data, the GraphData1 - GraphDataN style elements.

Interaction: For this option to have any effect, the fill must be enabled by the ODS style or the DISPLAY= option.

GROUP=*column* | *expression*

creates a separate ellipse for each unique group value of the specified column.

Default:

- If DISPLAY=(OUTLINE), each distinct group value might be represented in the graph by a different combination of outline color and line pattern. Line color and pattern vary according to the ContrastColor and LineStyle attributes of the GraphData1 - GraphDataN style elements. Line thickness (for grouped and ungrouped data) is controlled by the OUTLINEATTRS= option.
- If DISPLAY=(FILL), each distinct group value might be represented in the graph by a different fill color defined by the GraphData1:Color - GraphDataN:Color style references.
- If DISPLAY=(FILL OUTLINE), each distinct group value might be represented in the graph by a different fill color, outline color, and outline pattern.

The representations that are used to identify the groups can be overridden individually. For example, each distinct group value might be represented by a different line pattern for the ellipses, but the PATTERN= suboption of the OUTLINEATTRS= option could be used to assign the same line pattern to all ellipse outlines, letting outline color distinguish group values.

Restriction: This option can be used only when a numeric column is specified for the SLOPE=, SEMIMAJOR=, and SEMIMINOR= arguments.

INDEX=*positive-integer-column* | *expression*

specifies indices for mapping line attributes (color and pattern) and fill attributes to one of the GraphData1 - GranphDataN style elements.

Default: no default

Interaction: For this option to have any effect, the GROUP= option must also be specified.

Interaction: All of the indexes for a specific group value must be the same. Otherwise, the results are unpredictable.

Interaction: If this option is not used, then the group values are mapped in the order of the data.

Interaction: The index values are 1-based indices. For the style elements in GraphData1 - GraphDataN, if the index value is greater than N, then a modulo operation remaps that index value to a number less than N to determine which style element to use.

Discussion: Indexing can be used to collapse the number of groups that are represented in a graph. For more information, see “Remapping Groups for Grouped Data” on page 121.

LEGENDLABEL= *"string"*

specifies a label for use in a legend.

Default: The *string* specified on the NAME= option.

Interaction: If the GROUP= option is specified, this option is ignored.

NAME=*"string"*

assigns a name to a plot statement for reference in other template statements.

Default: no default

Restriction: The *string* is case sensitive, cannot contain spaces, and must define a unique name within the template.

Interaction: The *string* is used as the default legend label if the LEGENDLABEL= option is not used.

The specified name is used primarily in legend statements to coordinate the use of colors and line patterns between the graph and the legend.

OUTLINEATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)

specifies the attributes of the ellipse outline. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Line Options" on page 564 for available *line-options*.

Default:

- For non-grouped data, the GraphDataDefault style element.
- For grouped data, the GraphData1 - GraphDataN style elements.

XAXIS=X | X2

specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

Default: X

Restriction: Another plot that establishes a data range for the designed axis must be included.

Interaction: The overall plot specification and the layout type determine the axis display. For more information, see "How Axis Features are Determined" on page 395.

YAXIS=Y | Y2

specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

Default: Y

Restriction: Another plot that establishes a data range for the designed axis must be included.

Interaction: The overall plot specification and the layout type determine the axis display. For more information, see "How Axis Features are Determined" on page 395.

Chapter 25

FRINGEPLOT Statement

Description and Syntax

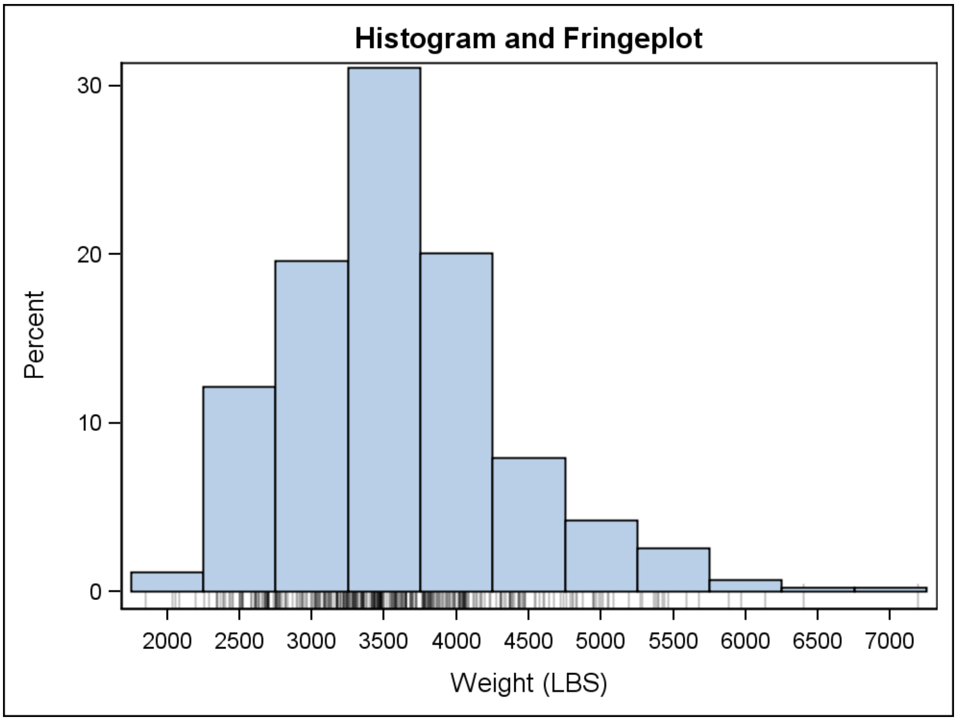
Creates a fringe plot on the X axis of an X-Y plot.

FRINGEPLOT *numeric-column* | *expression* <*/option(s)*>;

Example Program and Statement Details

Example Graph

The following graph was generated by the "Example Program" on page 246:



Example Program

```
proc template;
  define statgraph fringeplot;
    dynamic VAR VARLABEL;
    begingraph;
      entrytitle "Histogram and Fringeplot";
      layout overlay / xaxisopts=(label=VARLABEL)
                       yaxisopts=(offsetmin=.03);
        fringeplot VAR / datatransparency=.75
          fringeheight=3pct;
        histogram VAR;
      endlayout;
    endgraph;
  end;
run;

proc sgrender data=sashelp.cars template=fringeplot;
  dynamic var="weight" varlabel="Weight (LBS)";
run;
```

Statement Summary

In a FRINGEPLOT, each fringe line represents the location of the corresponding raw data value on the X axis. All fringe lines are of equal length.

Required Arguments

numeric-column
specifies a column that contains numeric values that provide the X coordinates of the data values.

expression
specifies an expression that calculates values when those values are not stored in the data.

Options

Statement Option	Description
DATATRANSPARENCY	Specifies the degree of the transparency of the fringe.
FRINGEHEIGHT	Specifies the height of the fringe lines.
LEGENDLABEL	Specifies a label for use in a legend.
LINEATTRS	Specifies the properties of the fringe lines.
NAME	Assigns a name to a plot statement for reference in other template statements.
ROLENAME	Specifies roles for information defined by data columns.
TIP	Specifies the information to display when the cursor is positioned over a fringe line.
TIPFORMAT	Specifies display formats for information defined by roles.
TIPLABEL	Specifies display labels for information defined by roles.
XAXIS	Specifies whether data are mapped to the primary X (bottom) axis or the secondary X2 (top) axis.

DATATRANSPARENCY=*number*
specifies the degree of the transparency of the fringe.

Default: 0

Range: 0 (opaque) to 1 (entirely transparent)

FRINGEHEIGHT=*dimension*
specifies the height of the fringe lines.

Default: 10 px

LEGENDLABEL= *"string"*

specifies a label for use in a legend.

Default: The X-variable label. If a label is not defined, the X-variable name.

LINEATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)

specifies the attributes of the fringe lines for the data points. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Line Options" on page 564 for available *line-options*.

Default: The GraphDataDefault style element.

NAME=*"string"*

assigns a name to a plot statement for reference in other template statements.

Default: no default

Restriction: The *string* is case sensitive, cannot contain spaces, and must define a unique name within the template.

The specified name is used primarily in legend statements to coordinate the use of colors and line patterns between the graph and the legend.

ROLENAME=(*role-name-list*)

specifies user-defined roles for information contained in data columns.

Default: no default

role-name-list

a blank-separated list of *rolename* = *column* pairs.

For example, ROLENAME= (TIP1=PCT) assigns the column PCT to the user-defined role TIP1.

Requirement: The role names that you choose must be unique and different from the pre-defined roles X and GROUP (reserved for future use).

This option provides a way to add to the data columns that appear in tooltips specified by the TIP= option.

TIP=(*role-list*)

specifies the information to display when the cursor is positioned over a fringe line. If this option is used, it replaces all the information displayed by default. Roles for columns that do not contribute to the fringe plot can be specified along with roles that do.

Default: The columns assigned to these roles are automatically included in the tooltip information: X and GROUP (reserved for future use).

role-list

an ordered, blank-separated list of unique FRINGEPLOT and user-defined roles. The FRINGEPLOT role is X and GROUP.

User-defined roles are defined with the ROLENAME= option.

The following example displays tooltips for the columns assigned to the roles X and Y, as well as the column PCT, which is not assigned to any pre-defined FringePlot role. The PCT column should appear first in the tooltip.

```
ROLENAME=(TIP1=PCT)
TIP=(X TIP1)
```

Requirement: To generate tooltips, you must include an ODS GRAPHICS ON statement that has the IMAGEMAP option specified, and write the graphs to the ODS HTML destination.

Interaction: The labels and formats for the TIP variables can be controlled with the TIPLABEL= and TIPFORMAT= options.

TIPFORMAT=(*role-format-list*)
specifies display formats for tip columns.

Default: The column format of the variable assigned to the role or BEST6. if no format is assigned to a numeric column.

role-format-list
a list of rolename = format pairs separated by blanks.

```
ROLENAME=(TIP1=PCT)
TIP=(X TIP1)
TIPFORMAT=( TIP1= PERCENT7.2 )
```

Requirement: This option provides a way to control the formats of columns that appear in tooltips. Only the roles that appear in the TIP= option are used. Columns must be assigned to the roles for this option to have any effect. See the ROLENAME= option.

TIPLABEL=(*role-format-list*)
specifies display labels for tip columns.

Default: The column label or column name of the variable assigned to the role.

role-label-list
a list of rolename = "string" pairs separated by blanks.

```
ROLENAME=(TIP1=PCT)
TIP=(X TIP1)
TIPLABEL=( TIP1="Percent" )
```

Requirement: This option provides a way to control the labels of columns that appear in tooltips. Only the roles that appear in the TIP= option are used. Columns must be assigned to the roles for this option to have any effect. See the ROLENAME= option.

XAXIS=X | X2
specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

Default: X

Interaction: The overall plot specification and the layout type determine the axis display. For more information, see "How Axis Features are Determined" on page 395.

The display of the fringe lines is always anchored on the X-axis (at the bottom of the plot area), even when the FRINGEPLOT's X-variable data range is mapped to the X2 axis.

Chapter 26
HISTOGRAM Statement

Description and Syntax

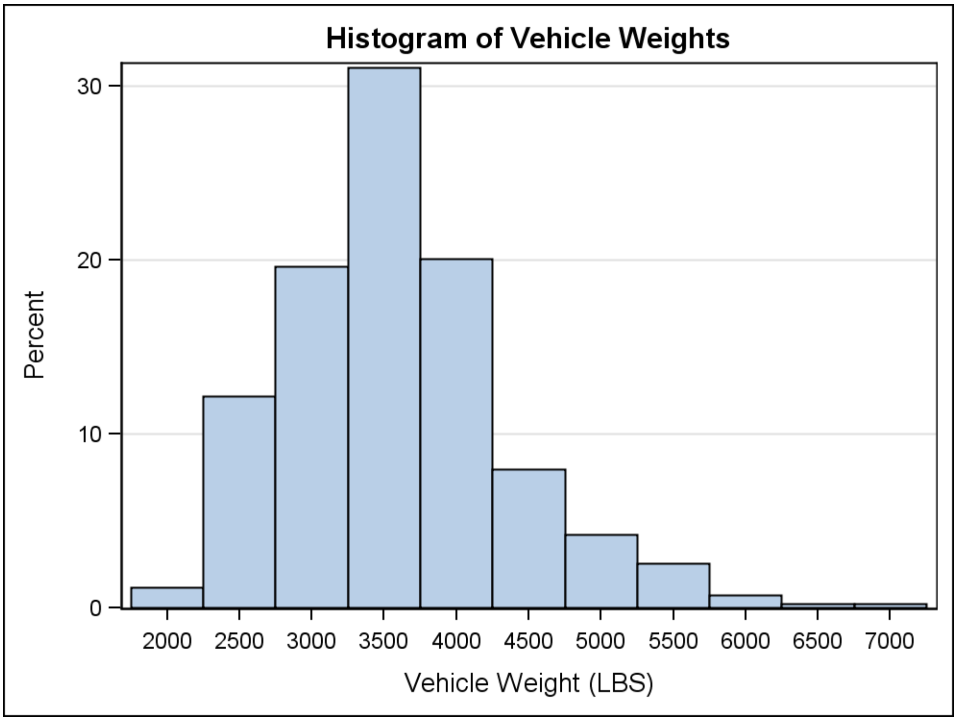
Creates a univariate histogram computed from input data.

HISTOGRAM *numeric-column* | *expression* </*option(s)*>;

Example Program and Statement Details

Example Graph

The following graph was generated by the "Example Program" on page 252:



Example Program

```
proc template;
  define statgraph histogram;
    begingraph;
      entrytitle "Histogram of Vehicle Weights";
      layout overlay /
        xaxisopts=(label="Vehicle Weight (LBS)")
        yaxisopts=(griddisplay=on);
        histogram weight;
      endlayout;
    endgraph;
  end;
run;

proc sgrender data=sashelp.cars template=histogram;
run;
```

Statement Summary

The Histogram’s default bin width is computed by using the number of observations and the range of the data. When a curve is overlaid on the histogram, the histogram’s bin width is used to scale the curve so that the area under the curve is equal to the area of the histogram.

Required Arguments

numeric-column

specifies a variable that contains numeric values, or a dynamic variable that refers to such a variable.

expression
specifies an expression that calculates values when those values are not stored in the data.

Options

Statement Option	Description
BINAXIS	Specifies whether to use bins as the basis for X axis tick marks.
BINSTART	Specifies the X coordinate of the first bin.
BINWIDTH	Specifies the bin width.
BOUNDARY	Specifies how a boundary is counted when it lies on the endpoint of a bin.
DATATRANSPARENCY	Specifies the degree of the transparency of the bars.
DISPLAY	Specifies whether to display outlined bars, filled bars, or outlined and filled bars.
ENDLABELS	Specifies whether axis ticks and value labels are drawn at the endpoints of the bins or the midpoints of the bins.
FILLATTRS	Specifies the appearance of the interior fill area of the bars.
FREQ	Specifies a column that indicates a frequency count for each observation of the input data object.
LEGENDLABEL	Specifies a label for use in a legend.
NAME	Assigns a name to a plot statement for reference in other template statements.
NBINS	Specifies the number of bins.
ORIENT	Specifies the orientation of the Y axis and the bars.
OUTLINEATTRS	Specifies the line properties of the bar outlines.
PRIMARY	Specifies that the data columns for this plot be used for determining default axis features.
SCALE	Specifies whether the Y axis displays the percentages between 0 and 100, the frequency counts, or the proportions between 0 and 1.

Statement Option	Description
TIPFORMAT	Specifies display formats for tip columns.
TIPLABEL	Specifies display labels for information defined by roles.
XAXIS	Specifies whether data are mapped to the primary X (bottom) axis or the secondary X2 (top) axis.
XVALUES	Specifies whether the X values represent lower endpoints, midpoints, or upper endpoints of the bins.
YAXIS	Specifies whether data are mapped to the primary Y (left) axis or the secondary Y2 (right) axis.

BINAXIS=*boolean*

Specifies whether to use bins as the basis for axis tick marks.

Default: TRUE

FALSE

The ENDLABELS= option is ignored. A standard axis is used, ignoring bin boundaries and midpoints.

TRUE

The ENDLABELS= option determines how the axis ticks and value labels are displayed. The axis ticks are in predetermined locations and cannot be changed with axis suboptions such TICKVALUELIST= and TICKVALUESEQUENCE=.

Interaction: When this option is set to TRUE, some X-axis options that are set on the parent layout might not apply.

Interaction: This option is ignored if this HISTOGRAM is not the primary plot (for more information about primary plots, see "When Plots Share Data and a Common Axis" on page 397).

BINSTART=*number*

specifies the X coordinate of the first bin. Use this option in conjunction with the BINWIDTH= or NBINS= options to specify bins. If neither BINWIDTH= nor the NBINS= option is specified, the system determines the number of bins. If the BINSTART value results in excluding the entire range of data, it is ignored and the default BINSTART value is used.

Default: Determined by the system.

See also: XVALUES= option

BINWIDTH=*number*

specifies the bin width. The system determines the number of bins. The bins always span the range of the data.

Default: Determined by the system.

Interaction: This option is ignored if the NBINS= option is also specified

BOUNDARY=UPPER | LOWER

specifies how a boundary is counted when it lies on the endpoint of a bin. If this option is set to UPPER, then the value is counted as one of the values in the upper bin (the bin to the right). Otherwise, it is counted in the lower bin.

Default: UPPER

DATATRANSPARENCY=*number*

specifies the degree of the transparency of the bars (outline and fill).

Default: 0

Range: 0 (opaque) to 1 (entirely transparent)

DISPLAY=STANDARD | ALL | (*display-options*)

specifies whether to display outlined bars, filled bars, or outlined and filled bars.

Default: The GraphHistogram:DisplayOpts style reference.

STANDARD

displays outlined, filled bars

ALL

displays outlined, filled bars

(*display-options*)

a list of comma-separated options enclosed in parentheses. Any of the options from the OUTLINEATTRS= and FILLATTRS= options can be used. The specified list must include one of the following:

OUTLINE—displays outlined bars

FILL—displays filled bars

Use the OUTLINEATTRS= and FILLATTRS= options to control the appearance of the bars.

ENDLABELS=*boolean*

specifies whether the axis ticks and value labels are drawn at the bin endpoints or bin midpoints.

Default: FALSE

Interaction: If this option is set to FALSE, then the axis ticks and value labels are drawn at the bin midpoints, whether the XVALUES= option identifies the X data as endpoint values or midpoint values.

Interaction: If BINAXIS=FALSE or this HISTOGRAM is not the primary plot, then this option is ignored.

FILLATTRS=*style-element* | *style-element* (*fill-options*) | (*fill-options*)

specifies the appearance of the interior fill area of the bars. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Fill Options" on page 564 for available *fill-options*.

Default: The GraphDataDefault style element

Interaction: For this option to have any effect, the fill must be enabled by the ODS style or the DISPLAY= option.

FREQ=*numeric-column* | *expression*

specifies a column that indicates a frequency count for each observation of the input data object. If *n* is the value of the FREQ variable for a given observation, then that observation is plotted *n* times.

Default: Each observation has a frequency count of 1.

Restriction: If the value of the *numeric-column* is missing or is less than 1, the observation is not used in the analysis. If the value is not an integer, only the integer portion is used.

LEGENDLABEL= *"string"*

specifies a label for use in a legend.

Default: The *string* specified on the NAME= option.

NAME= *"string"*

assigns a name to a plot statement for reference in other template statements.

Default: no default

Restriction: The *string* is case sensitive, cannot contain spaces, and must define a unique name within the template.

Interaction: The *string* is used as the default legend label if the LEGENDLABEL= option is not used.

The specified name is used primarily in legend statements to coordinate the use of colors and line patterns between the graph and the legend.

NBINS=*integer*

specifies the number of bins. The system determines the BINWIDTH= value. The bins always span the range of the data.

Default: Determined by the system.

ORIENT=VERTICAL | HORIZONTAL

specifies the orientation of the Y axis and the bars.

Default: VERTICAL

OUTLINEATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)

specifies the line properties of the bar outlines. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Line Options" on page 564 for available *line-options*.

Default: The GraphDataDefault style element.

Interaction: For this option to have any effect, the outlines must be enabled by the ODS style or the DISPLAY= option.

PRIMARY=*boolean*

specifies that the data columns for this plot be used for determining default axis features.

Default: FALSE

Restriction: This option is ignored if the plot is placed under a GRIDDED or LATTICE layout block.

Details: This option is needed only when two or more plots within an overlay-type layout contribute to a common axis. For more information, see **SECTION ON SHARED DATA**

SCALE=PERCENT | COUNT | PROPORTION

specifies whether the Y axis displays the percentages between 0 and 100, or the frequency counts, or the proportions between 0 and 1.

Default: PERCENT

TIPFORMAT=(*role-format-list*)

specifies display formats for tip columns.

Default: The column format of the variable assigned to the role or BEST6. if no format is assigned to a numeric column. The X role represents the binned value. The Y role represents the computed amount of X in the units specified by the SCALE= option.

(*role-format-list*)
a list of *rolename = format* pairs separated by blanks.

```
TIPFORMAT=( Y=PERCENT7.2)
```

Requirement: To generate tooltips, you must include an ODS GRAPHICS ON statement that has the IMAGEMAP option specified, and write the graphs to the ODS HTML destination.

TIPLABEL=(*role-label-list*)
specifies display labels for tip columns.

Default: The column label or column name of the variable assigned to the role. The X role represents the binned value. The Y role represents the computed amount of X in the units specified by the SCALE= option.

(*role-label-list*)
a list of *rolename = "string"* pairs separated by blanks.

```
TIPLABEL=( Y="Percent")
```

Requirement: To generate tooltips, you must include an ODS GRAPHICS ON statement that has the IMAGEMAP option specified, and write the graphs to the ODS HTML destination.

XAXIS=X | X2
specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

Default: X

Interaction: The overall plot specification and the layout type determine the axis display. For more information, see "How Axis Features are Determined" on page 395.

XVALUES=MIDPOINTS | LEFTPOINTS | RIGHTPOINTS
specifies whether the X values represent lower endpoints, midpoints, or upper endpoints of the bins. For example, if BINSTART=10 and BINWIDTH=10, using LEFTPOINTS would result in bins 10 - 20, 20 - 30, and so on. Using RIGHTPOINTS would result in bins 0 - 10, 10 - 20, ..., and using MIDPOINTS would result in bins 5 - 15, 15 - 25, ...

Default: MIDPOINTS

YAXIS=Y | Y2
specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

Default: Y

Interaction: The overall plot specification and the layout type determine the axis display. For more information, see "How Axis Features are Determined" on page 395.

Chapter 27
HISTOGRAMPARM Statement

Description and Syntax

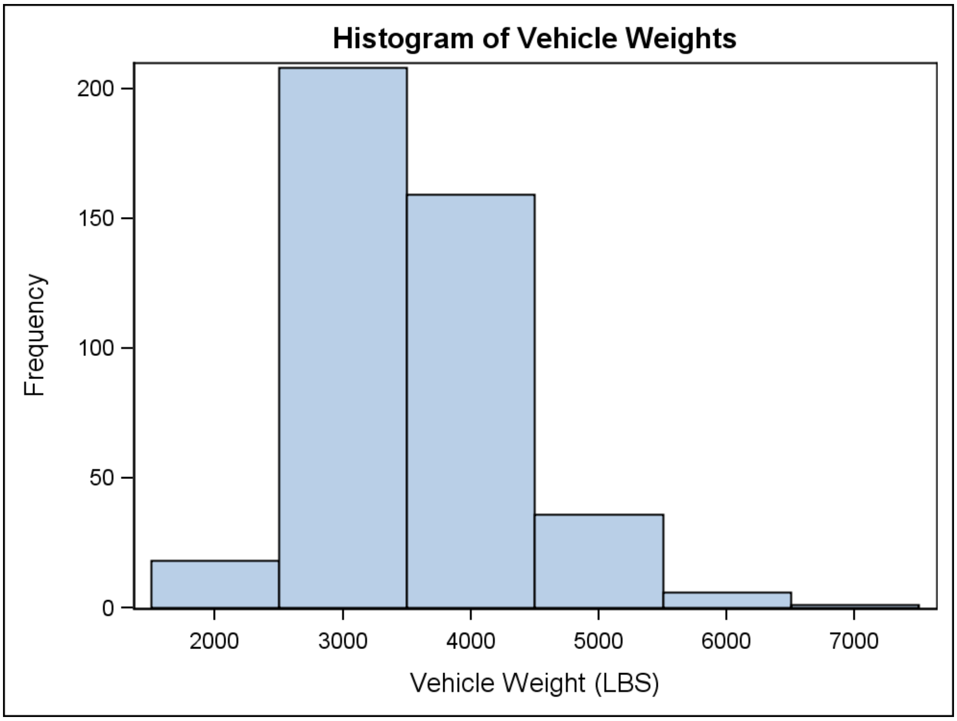
Creates a univariate histogram for specified values of bin midpoints and bin frequencies.

HISTOGRAMPARM X = *numeric-column* | *expression*
Y = *non-negative-numeric-column* | *expression* <*/option(s)*>;

Example Program and Statement Details

Example Graph

The following graph was generated by the "Example Program" on page 260:



Example Program

```
proc template;
  define statgraph histogramparm;
    begingraph;
      entrytitle "Histogram of Vehicle Weights";
      layout overlay;
        histogramparm  x=midpoint y=frequency;
      endlayout;
    endgraph;
  end;
run;

data bins;
  input Midpoint Frequency;
  label midpoint="Vehicle Weight (LBS)";
datalines;
2000 18
3000 208
4000 159
5000 36
6000 6
7000 1
;
run;

proc sgrender data=bins template=histogramparm;
run;
```

Statement Summary

When a curve is overlaid on the histogram, the histogram bin width is used to scale the curve so that the area under the curve is equal to the area of the histogram.

Required Arguments

X=*numeric-column* | *expression*
specifies the column for the midpoint values.

Requirement: The X values must be uniformly spaced.

Y=*non-negative-numeric-column* | *expression*
specifies the column for the frequency values. These values can be frequency counts, percentages, or proportions between 0 and 1.

Requirement: The Y values cannot be negative.

Options

Statement Option	Description
BINAXIS	Specifies whether to use bins as the basis for axis tick marks.
DATATRANSPARENCY	Specifies the degree of the transparency of the bars.
DISPLAY	Specifies whether to display outlined bars, filled bars, or outlined and filled bars.
ENDLABELS	Specifies whether axis ticks and value labels are drawn at the endpoints of the bins or the midpoints of the bins.
FILLATTRS	Specifies the appearance of the interior fill area of the bars.
LEGENDLABEL	Specifies a label for use in a legend.
NAME	Assigns a name to a plot statement for reference in other template statements.
ORIENT	Specifies the orientation of the Y axis and the bars.
OUTLINEATTRS	Specifies the properties of the bar outlines.
PRIMARY	Specifies that the data columns for this plot be used for determining default axis features.
ROLENAME	Specifies roles for information defined by data columns.

Statement Option	Description
TIP	Specifies the information to display when the cursor is positioned over a histogram bin.
TIPFORMAT	Specifies display formats for tip columns.
TIPLABEL	Specifies display labels for tip columns.
XAXIS	Specifies whether data are mapped to the primary X (bottom) axis or the secondary X2 (top) axis.
XVALUES	Specifies whether the X values represent midpoints, lower endpoints, or upper endpoints of the bins.
YAXIS	Specifies whether data are mapped to the primary Y (left) axis or the secondary Y2 (right) axis.

BINAXIS=*boolean*

Specifies whether to use bins as the basis for axis tick marks.

Default: TRUE

FALSE

The ENDLABELS= option is ignored. A standard axis is used, ignoring bin boundaries and midpoints.

TRUE

The ENDLABELS= option determines how the axis ticks and value labels are displayed. The axis ticks are in predetermined locations and cannot be changed with axis suboptions such TICKVALUELIST= and TICKVALUESEQUENCE=.

Interaction: When this option is set to TRUE, some X-axis options that are set on the parent layout might not apply.

Interaction: This option is ignored if this HISTOGRAMPARM is not the primary plot (for more information about primary plots, see "When Plots Share Data and a Common Axis" on page 397).

DATATRANSPARENCY= *number*

specifies the degree of the transparency of the bars (outline and fill).

Default: 0

Range: 0 (opaque) to 1 (entirely transparent)

DISPLAY=STANDARD | ALL | (*display-options*)

specifies whether to display outlined bars, filled bars, or outlined and filled bars.

Default: The GraphHistogram:DisplayOpts style reference.

STANDARD

displays outlined, filled bars

ALL

displays outlined, filled bars

(*display-options*)

a list of comma-separated options enclosed in parentheses. Any of the options from the OUTLINEATTRS= and FILLATTRS= options can be used. The specified list must include one of the following:

OUTLINE—displays outlined bars

FILL—displays filled bars

Use the OUTLINEATTRS= and FILLATTRS= options to control the appearance of the bars.

ENDLABELS=*boolean*

specifies whether axis ticks and value labels are drawn at the endpoints of the bins or midpoints of the bins.

Default: FALSE

Interaction: If this option is set to FALSE, then the axis ticks and value labels are drawn at the bin midpoints. This occurs whether the BINVALUES= option identifies the X data as endpoint values or midpoint values.

Interaction: If BINAXIS=FALSE or this HISTOGRAMPARM is not the primary plot, then this option is ignored.

FILLATTRS=*style-element* | *style-element* (*fill-options*) | (*fill-options*)

specifies the appearance of the interior fill area of the bars. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Fill Options" on page 564 for available *fill-options*.

Default: The GraphDataDefault style element

Interaction: For this option to have any effect, the fill must be enabled by the ODS style or the DISPLAY= option.

LEGENDLABEL= *"string"*

specifies a label for use in a legend.

Default: The *string* specified on the NAME= option.

NAME= *"string"*

assigns a name to a plot statement for reference in other template statements.

Default: no default

Restriction: The *string* is case sensitive, cannot contain spaces, and must define a unique name within the template.

Interaction: The *string* is used as the default legend label if the LEGENDLABEL= option is not used.

The specified name is used primarily in legend statements to coordinate the use of colors and line patterns between the graph and the legend.

ORIENT=VERTICAL | HORIZONTAL

specifies the orientation of the Y axis and the bars.

Default: VERTICAL

OUTLINEATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)

specifies the line properties of the bar outlines. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Line Options" on page 564 for available *line-options*.

Default: The GraphDataDefault style element.

Interaction: For this option to have any effect, the outlines must be enabled by the ODS style or the DISPLAY= option.

PRIMARY=*boolean*

specifies that the data columns for this plot be used for determining default axis features.

Default: FALSE

Restriction: This option is ignored if the plot is placed under a GRIDDED or LATTICE layout block.

Details: This option is needed only when two or more plots within an overlay-type layout contribute to a common axis. For more information, see "When Plots Share Data and a Common Axis" on page 397

ROLENAME=(*role-name-list*)

specifies user-defined roles for information contained in data columns.

Default: no default

(*role-name-list*)

a blank-separated list of *rolename* = *column* pairs.

For example, ROLENAME= (TIP1=PCT) assigns the column PCT to the user-defined role TIP1.

Requirement: The role names that you choose must be unique and different from the pre-defined roles X and Y.

This option provides a way to add to the data columns that appear in tooltips specified by the TIP= option.

TIP=(*role-list*)

specifies the information to display when the cursor is positioned over a histogram bin. If this option is used, it replaces all the information displayed by default. Roles for columns that do not contribute to the histogram can be specified along with roles that do.

Default: The columns assigned to these roles are automatically included in the tooltip information: X and Y.

(*role-list*)

an ordered, blank-separated list of unique HISTOGRAMPARM and user-defined roles. HISTOGRAMPARM roles include X and Y.

User-defined roles are defined with the ROLENAME= option.

The following example displays tooltips for the columns assigned to the roles X and Y, as well as the column PCT, which is not assigned to any pre-defined HISTOGRAMPARM role. The PCT column must first be assigned a role.:

```
ROLENAME=(TIP1=PCT)
TIP= (X Y TIP1)
```

Requirement: To generate tooltips, you must include an ODS GRAPHICS ON statement that has the IMAGEMAP option specified, and write the graphs to the ODS HTML destination.

Interaction: The labels and formats for the TIP variables can be controlled with the TIPLABEL= and TIPFORMAT= options.

TIPFORMAT=(*role-format-list*)

specifies display formats for tip columns.

Default: The column format of the variable assigned to the role or BEST6. if no format is assigned to a numeric column.

(*role-format-list*)
a list of *rolename = format* pairs separated by blanks.

```
ROLENAME=(TIP1=PCT)
TIP=(X Y TIP1)
TIPFORMAT=( TIP1=PERCENT7.2)
```

Requirement: This option provides a way to control the formats of columns that appear in tooltips. Only the roles that appear in the TIP= option are used. Columns must be assigned to the roles for this option to have any effect. See the ROLENAME= option.

TIPLABEL=(*role-label-list*)
specifies display labels for tip columns.

Default: The column label or column name of the variable assigned to the role.

(*role-label-list*)
a list of *rolename = "string"* pairs separated by blanks.

```
ROLENAME=(TIP1=PCT)
TIP=(X Y TIP1)
TIPLABEL=( TIP1="Percent")
```

Requirement: This option provides a way to control the labels of columns that appear in tooltips. Only the roles that appear in the TIP= option are used. Columns must be assigned to the roles for this option to have any effect. See the ROLENAME= option.

XAXIS=X | X2
specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

Default: X

Interaction: The overall plot specification and the layout type determine the axis display. For more information, see "How Axis Features are Determined" on page 395.

XVALUES=MIDPOINTS | LEFTPOINTS | RIGHTPOINTS
specifies whether the X values represent midpoints, lower endpoints, or upper endpoints of the bins.

Default: MIDPOINTS

YAXIS=Y | Y2
specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

Default: Y

Interaction: The overall plot specification and the layout type determine the axis display. For more information, see "How Axis Features are Determined" on page 395.

Chapter 28

LINEPARM Statement

Description and Syntax

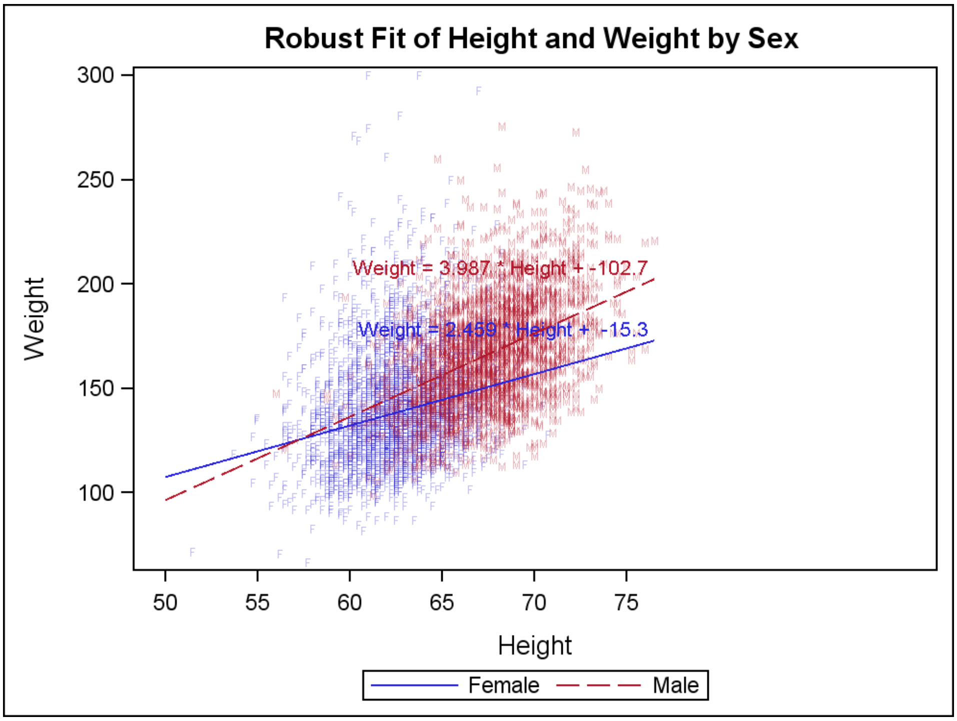
Creates a straight line specified by a point and a slope.

LINEPARM X = *number* | *numeric-column* | *expression*
Y = *number* | *numeric-column* | *expression*
SLOPE = *number* | *numeric-column* | *expression* </option(s)>;

Example Program and Statement Details

Example Graph

The following graph was generated by the “Example Program” on page 268:



Example Program

The LINEPARM statement draws a line based on a point and the slope of the line that passes through that point. You can use this statement to create a reference line with any slope or, in this example, to draw a fit from a linear regression. Many SAS/STAT procedures create output data sets containing a Y-intercept and slope and coefficient for the linear regression equation.

```
proc template;
 define statgraph lineparm;
  begingraph;
   entrytitle "Robust Fit of Height and Weight by Sex ";
   layout overlay / xaxisopts=(offsetmax=.35);
    scatterplot x=height y=weight / group=sex
     markercharacter=eval(substr(sex,1,1))
     markercharacterattrs=(size=5pt) datatransparency=.7;
    lineparm x=0 y=intercept slope=slope /
     name="Line" group=sex
     curvelabel=eval("Weight = "||put(slope,5.3)||
       " * Height + "||put(intercept,6.1))
     curvelabellocation=inside
     curvelabelattrs=(size=8pt);
    discretelegend "Line";
   endlayout;
  endgraph;
 end;
run;

proc sort data=sashelp.heart(keep=height weight sex)
     out=heart;
  by sex;
run;
```

```
ods exclude all;
proc robustreg data=heart method=m
     outest=stats(rename=(height=slope));
  by sex;
  model weight=height;
run;

data all;
  merge heart stats(keep=intercept slope sex);
run;

ods select all;
proc sgrender data=all template=lineparm;
run;
```

Statement Summary

The LINEPARM statement creates a straight line. You can generate a single line by specifying a constant for each required argument. You can generate multiple lines by specifying a numeric column for any or all required arguments. If any of the X= or Y= columns contains a missing value, no line is drawn. To request a vertical line, specify **`SLOPE= .`** (specify a missing value as a constant or column value).

A LINEPARM statement can be used in any layout except GRIDDED or OVERLAY3D layouts. The parent layout must include another plot statement that is derived from data values that establish a data range for the axes. For example, it can be used with a scatter plot or a histogram.

To draw vertical or horizontal reference lines, consider using the simpler REFERENCELINE statement.

Required Arguments

X=*number* | *numeric-column* | *expression*
specifies the X coordinate of a point.

By default, if the specified value is outside of the data range, then the data range is extended to include the specified intercept. This behavior can be changed with the CLIP= option. If a *numeric-column* is specified and the column contains missing values, no line is drawn for the missing values.

Values are displayed in the units of the data.

Y=*number* | *numeric-column* | *expression*
specifies the Y coordinate of a point.

By default, if the specified value is outside of the data range, then the data range is extended to include the specified intercept. This behavior can be changed with the CLIP= option. If a *numeric-column* is specified and the column contains missing values, no line is drawn for the missing values.

Values are displayed in the units of the data.

SLOPE=*number* | *numeric-column* | *expression*
specifies the slope of the line.

Slope can be positive or negative.

`SLOPE= 0` creates a line parallel to the X-axis. `SLOPE= .` (a missing value) creates a line parallel to the Y-axis.

Options

Statement Option	Description
CLIP	Specifies whether the data for the line are considered when determining the data ranges for the axes.
CURVELABEL	Specifies the label of the line.
CURVELABELATTRS	Specifies the color and font attributes of the line label.
CURVELABELLOCATION	Specifies the location of the line label relative to the plot area.
CURVELABELPOSITION	Specifies the position of the line label relative to the line end points.
DATATRANSPARENCY	Specifies the degree of the transparency of the lines.
EXTEND	Specifies whether the line is to be drawn to the area bounded by the axes, ignoring any specified offsets.
GROUP	Creates a separate parameterized line plot for each unique group value of the specified column.
INDEX	Specifies indices for mapping line attributes (color and pattern) to one of the GraphData1 - GranphDataN style elements.
LEGENDLABEL	Specifies a label for use in a legend.
LINEATTRS	Specifies the properties of the line.
NAME	Assigns a name to a plot statement for reference in other template statements.
XAXIS	Specifies whether data are mapped to the primary X (bottom) axis or the secondary X2 (top) axis.
YAXIS	Specifies whether data are mapped to the primary Y (left) axis or the secondary Y2 (right) axis.

CLIP= *boolean*

specifies whether the data for the line are considered when determining the data ranges for the axes.

Default: FALSE

FALSE

The data for the line contributes to data range for each axis. Each axis might be extended to force the display of the line.

TRUE

The data for the line are ignored when establishing axis scales. Each axis scale is determined by the other plots in the layout. This might result in the line not being displayed if its data range is not within the data ranges of the other plots.

CURVELABEL= *"string"* | *column* | *expression*

specifies a label for the line.

Default: no default

Interaction:

- For a single line, use *"string"*.
- For multiple lines, use a column to define the labels for each unique value.

The font and color attributes for the label are specified by the CURVELABELATTRS= option.

CURVELABELATTRS=*style-element* | *style-element* (*text-options*) | (*text-options*)

specifies the color and font attributes of the line label. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Text Options" on page 566 for available *text-options*.

Default:

- For non-grouped data, the GraphValueText style element.
- For grouped data, text color is derived from the GraphData1:ContrastColor - GraphDataN:ContrastColor style references. The font is derived from the GraphValueText style element.

Interaction: For this option to take effect, the CURVELABEL= option must also be used.

Interaction: This option's COLOR= setting overrides the colors indicated by the GROUP= option.

When the GROUP= option is used, each distinct group value might be represented by a different color. The line label that is associated with the group is assigned the group color. This option can be used to specify a single color for all line labels in a graph, without affecting the line colors.

CURVELABELLOCATION=INSIDE | OUTSIDE

specifies the location of the line label relative to the plot area.

Default: INSIDE

INSIDE

inside the plot area

OUTSIDE

outside the plot area

Restriction: OUTSIDE cannot be used when the LINEPARM is used in multi-cell layouts such as LATTICE, DATAPANEL, or DATALATTICE, where axes might be external to the grid.

Interaction: For this option to take effect, the CURVELABEL= option must also be specified.

Interaction: This option is used in conjunction with the CURVELABELPOSITION= option to determine where the line labels appear. For more information, see "Location and Position of Curve Labels" on page 123.

CURVELABELPOSITION=AUTO | MAX | MIN

specifies the position of the line label relative to the line end points.

Default: AUTO when CUVELABELLOCATION=OUTSIDE MAX when CURVELABELLOCATION=INSIDE

AUTO

Only used when CURVELABELLOCATION=OUTSIDE. The line label is positioned automatically near the line boundary along unused axes whenever possible (typically Y2 and X2) to avoid collision with tick values.

MAX

Forces the line label to appear near maximum line values (typically, upper right).

MIN

Forces the line label to appear near minimum line values (typically, lower left).

Restriction: The AUTO setting is not supported if CURVELABELLOCATION=INSIDE is specified.

Interaction: For this option to take effect, the CURVELABEL= option must also be specified.

Interaction: This option is used in conjunction with the CURVELABELLOCATION= option to determine where the line label appears. For more information, see "Location and Position of Curve Labels" on page 123.

DATATRANSPARENCY=*number*

specifies the degree of the transparency of the line.

Default: 0

Range: 0 (opaque) to 1 (entirely transparent)

EXTEND=*boolean*

specifies whether the line is to be drawn to the area bounded by the axes.

Default: FALSE

If this option is not specified, then there can be a small gap between the line and the axis. The gap is controlled by the axis offset. If the offset is set to 0, then there is no gap.

GROUP=*column* | *expression*

creates a separate parameterized line plot for each unique group value of the specified column.

Default: Each distinct group value might be represented in the graph by a different combination of line color and line pattern. Line colors vary according to the GraphData1:ContrastColor - GraphDataN:ContrastColor style references, and line patterns vary according to the GraphData1:LineStyle - GraphDataN:LineStyle style references.

Restriction: When this option is used, the X, Y, and SLOPE arguments must specify numeric columns.

Interaction: The group values are mapped in the order of the data, unless the INDEX= option is used to alter the default sequence of line colors and line patterns.

The LINEATTRS= option can be used to override the representations that are used to identify the groups. For example, LINEATTRS=(PATTERN=SOLID) can be used to assign the same pattern to all of the lines, letting the line color distinguish group values. Likewise, LINEATTRS=(COLOR=BLACK) can be used to assign the same color to all of the lines, letting the line pattern distinguish group values.

INDEX=*positive-integer-column* | *expression*
specifies indices for mapping line attributes (color and pattern) to one of the GraphData1 - GranphDataN style elements.

Default: no default

Interaction: For this option to take effect, the GROUP= option must also be specified.

Interaction: All of the indexes for a specific group value must be the same. Otherwise, the results are unpredictable.

Interaction: If this option is not used, then the group values are mapped in the order of the data.

Interaction: The index values are 1-based indices. For the style elements in GraphData1 - GraphDataN, if the index value is greater than N, then a modulo operation remaps that index value to a number less than N to determine which style element to use.

Discussion: Indexing can be used to collapse the number of groups that are represented in a graph. For more information, see "Remapping Groups for Grouped Data" on page 121.

LEGENDLABEL= *"string"*
specifies a label for use in a legend.

Default: The *string* specified on the NAME= option.

Interaction: If the GROUP= option is specified, this option is ignored.

LINEATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)
specifies the attributes of the line. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Line Options" on page 564 for available *line-options*.

Default:

- For non-grouped data, the GraphDataDefault style element.
- For grouped data, the ContrastColor, LineStyle, and LineThickness attributes of the GraphData1- GraphDataN style elements.

NAME=*"string"*
assigns a name to a plot statement for reference in other template statements.

Default: no default

Restriction: The *string* is case sensitive, cannot contain spaces, and must define a unique name within the template.

Interaction: The *string* is used as the default legend label if the LEGENDLABEL= option is not used.

The specified name is used primarily in legend statements to coordinate the use of colors and line patterns between the graph and the legend.

XAXIS=X | X2

specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

Default: X

Restriction: Another plot that establishes a data range for the axis must be included.

Interaction: The overall plot specification and the layout type determine the axis display. For more information, see "How Axis Features are Determined" on page 395.

YAXIS=Y | Y2

specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

Default: Y

Restriction: Another plot that establishes a data range for the axis must be included.

Interaction: The overall plot specification and the layout type determine the axis display. For more information, see "How Axis Features are Determined" on page 395.

Chapter 29

LOESSPLOT Statement

Description and Syntax

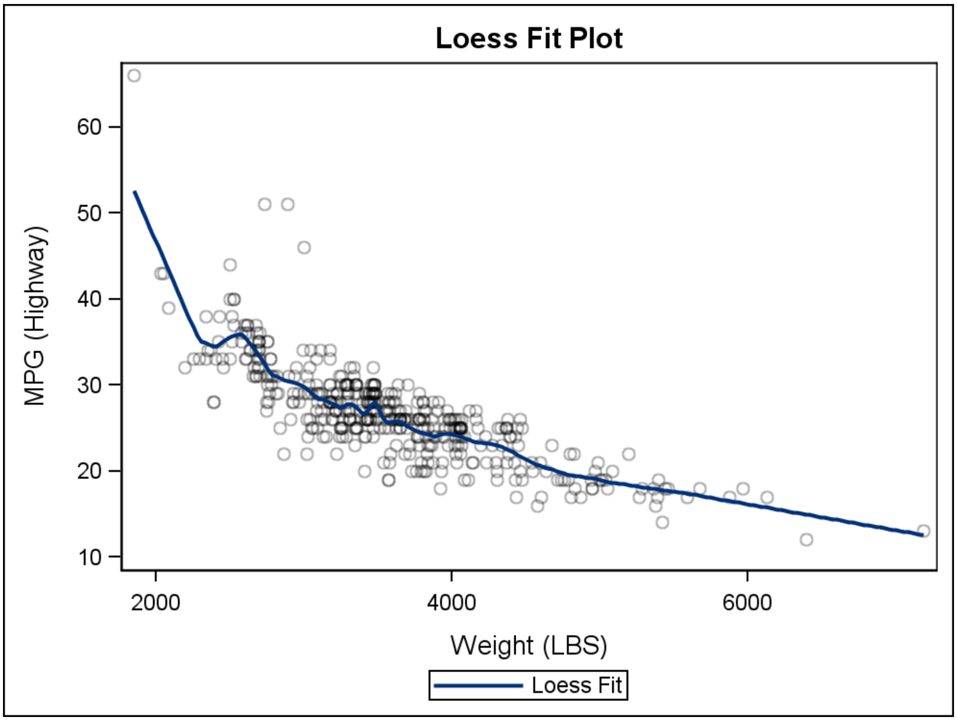
Creates a fitted loess curve computed from input data.

LOESSPLOT X = *numeric-column* | *expression*
Y = *numeric-column* | *expression*</ <*regression-options*> <*option(s)*> >;

Example Program and Statement Details

Example Graph

The following graph was generated by the “Example Program” on page 276:



Example Program

```
proc template;
  define statgraph loessplot;
    begingraph;
      entrytitle "Loess Fit Plot";
      layout overlay;
        scatterplot x=weight y=mpg_highway /
          datatransparency=.7;
        loessplot x=weight y=mpg_highway / name="fitline"
          alpha=.05 legendlabel="Loess Fit";
        discretelegend "fitline";
      endlayout;
    endgraph;
  end;
run;

proc sgrender data=sashelp.cars template=loessplot;
run;
```

Statement Summary

The LOESSPLOT statement only supports statistical models of one independent and one dependent variable. For more information about the fitting methodology, see the LOESS procedure in the **MISSING Bookref** .

In addition to the loess curve, the LOESSPLOT statement can compute confidence levels for the fitted line. To display the confidence levels,

1. use the CLM= option to declare a name for the confidence level of the mean

2. use a MODELBAND statement to refer this name. This statement draws a confidence band from this information. See "MODELBAND Statement" on page 283 for information about how to control the appearance of the confidence band.

Required Arguments

X=*numeric-column* | *expression*
specifies the column for the X values.

Y=*numeric-column* | *expression*
specifies the column for the Y values.

Loess Regression Options

ALPHA= *positive-number*
specifies the confidence level to compute.

Default: .05

Range: 0 < *number* < 1

Range: ALPHA=.05 represents a 95% confidence level.

CLM= *"name"*
produces confidence limits for a mean predicted value for each observation. The confidence level is set by the ALPHA= option.

Default: no default

Interaction: "*name*" is a unique name within the template that is case sensitive and cannot contain spaces. It must be assigned in order for the confidence limits to be computed. To display confidence limits, you must use this name as the required argument of a MODELBAND statement.

DEGREE=1 | 2
specifies the degree of the local polynomials to use for each local regression. The valid values are 1 for local linear fitting or 2 for local quadratic fitting.

Default: 1

INTERPOLATION=LINEAR | CUBIC
specifies the degree of the interpolating polynomials used for blending local polynomial fits at the kd tree vertices.

Default: LINEAR

CUBIC
cubic polynomials

LINEAR
linear polynomials

MAXPOINTS= *positive-integer*
specifies the maximum number of predicted points generated for the loess curve as well as confidence limits.

Default: 201

SMOOTH=AUTO | *positive-number*
specifies a regression parameter value.

Default: AUTO

REWEIGHT=NONE | *positive-integer*

specifies the number of iterative re-weighting steps to be done. Such iterations are appropriate when there are outliers in the data or when the error distribution is a symmetric long-tailed distribution.

Default: NONE

WEIGHT= *numeric-column*

specifies a variable in the input data set that contains values to be used as a priori weights for a loess fit. The values of the weight variable must be nonnegative. If an observation's weight is zero, negative, or missing, the observation is deleted from the analysis.

Default: no default

Options

Statement Option	Description
CURVELABEL	Specifies the label of the loess curve.
CURVELABELATTRS	Specifies the color and font attributes of the curve label.
CURVELABELLOCATION	Specifies the location of the curve label relative to the plot area.
CURVELABELPOSITION	Specifies the position of the curve label relative to the curve line.
DATATRANSPARENCY	Specifies the degree of the transparency of the loess curve.
GROUP	Creates a distinct set of curves from just the observations that correspond to each unique group value of the specified column.
INDEX	Specifies indices for mapping line attributes (color and pattern) to one of the GraphData1 - GranphDataN style elements.
LEGENDLABEL	Specifies a label for use in a legend.
LINEATTRS	Specifies the properties of the loess curve.
NAME	Assigns a name to a plot statement for reference in other template statements.
PRIMARY	Specifies that the data columns for this plot be used for determining default axis features.
TIPFORMAT	Specifies display formats for information defined by roles.
TIPLABEL	Specifies display labels for information defined by roles.

Statement Option	Description
XAXIS	Specifies whether data are mapped to the primary X (bottom) axis or the secondary X2 (top) axis.
YAXIS	Specifies whether data are mapped to the primary Y (left) axis or the secondary Y2 (right) axis.

CURVELABEL=*"string"*
specifies a label for the loess curve.

Default: no default

Interaction: If the GROUP= option is specified, this option is ignored.

The font and color attributes for the label are specified by the CURVELABELATTRS= option.

CURVELABELATTRS=*style-element* | *style-element* (*text-options*) | (*text-options*)
specifies the color and font attributes of the curve labels. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Text Options" on page 566 for available *text-options*.

Default: The GraphValueText style element.

Interaction: For this option to take effect, the CURVELABEL= option must also be specified.

Interaction: If the GROUP= option is specified, this option is ignored.

CURVELABELLOCATION=INSIDE | OUTSIDE
specifies the location of the curve label relative to the plot area.

Default: INSIDE

INSIDE
inside the plot area

OUTSIDE
outside the plot area

Restriction: OUTSIDE cannot be used when the LOESSPLOT is used in multicell layouts such as LATTICE, DATAPANEL, or DATALATTICE, where axes might be external to the grid.

Interaction: For this option to take effect, the CURVELABEL= option must also be specified.

Interaction: This option is used in conjunction with the CURVELABELPOSITION= option to determine where the curve labels appear. For more information, see "Location and Position of Curve Labels" on page 123.

CURVELABELPOSITION=AUTO | MAX | MIN | START | END
specifies the position of the curve label relative to the curve line.

Default: AUTO when CUVELABELLOCATION=OUTSIDE, and END when CURVELABELLOCATION=INSIDE

AUTO
Only used when CURVELABELPOSITION=OUTSIDE. The curve label is positioned automatically near the curve boundary along unused axes whenever possible (typically Y2 and X2) to avoid collision with tick values.

MAX
Forces the curve label to appear near maximum curve values (typically, upper right).

MIN
Forces the curve label to appear near minimum curve values (typically, lower left).

START
Only used when CURVELABELLOCATION=INSIDE. Forces the curve label to appear near the beginning of the curve. Particularly useful when the curve line has a spiral shape.

END
Only used when CURVELABELLOCATION=INSIDE. Forces the curve label to appear near the end of the curve. Particularly useful when the curve line has a spiral shape.

Interaction: For this option to take effect, the CURVELABEL= option must also be specified.

Interaction: The AUTO setting is not supported if CURVELABELLOCATION=INSIDE is specified. The START and END settings are not supported if CURVELABELLOCATION=OUTSIDE is specified.

Interaction: This option is used in conjunction with the CURVELABELLOCATION= option to determine where the curve label appears. For more information, see "Location and Position of Curve Labels" on page 123.

DATATRANSPARENCY=*number*
specifies the degree of the transparency of the loess curve.

Default: 0

Range: 0 (opaque) to 1 (entirely transparent)

GROUP=*column* | *expression*
creates a distinct set of curves from just the observations that correspond to each unique group value of the specified column.

Default: Each distinct group value might be represented in the graph by a different combination of color and line pattern. Line colors vary according to the GraphData1:ContrastColor - GraphDataN:ContrastColor style references, and line patterns vary according to the GraphData1:LineStyle - GraphDataN:LineStyle style references.

Restriction: The input data must be sorted by the GROUP= column.

Interaction: The group values are mapped in the order of the data, unless the INDEX= option is used to alter the default sequence of line colors and line patterns.

The LINEATTRS= option can be used to override the representations that are used to identify the groups. For example, LINEATTRS=(PATTERN=SOLID) can be used to assign the same pattern to all of the loess curves, letting the line color distinguish group values. Likewise, LINEATTRS=(COLOR=BLACK) can be used to assign the same color to all of the curves, letting the line pattern distinguish group values.

INDEX=*positive-integer-column* | *expression*
specifies indices for mapping line attributes (color and pattern) to one of the GraphData1 - GranphDataN style elements.

Default no default

Interaction: For this option to take effect, the GROUP= option must also be specified.

Interaction: All of the indexes for a specific group value must be the same. Otherwise, the results are unpredictable.

Interaction: If this option is not used, then the group values are mapped in the order of the data.

Interaction: The index values are 1-based indices. For the style elements in GraphData1 - GraphDataN, if the index value is greater than N, then a modulo operation remaps that index value to a number less than N to determine which style element to use.

Discussion: Indexing can be used to collapse the number of groups that are represented in a graph. For more information, see "Remapping Groups for Grouped Data" on page 121.

LEGENDLABEL= *"string"*
specifies a label to be used in a legend.

Default: The *string* specified on the NAME= option.

Interaction: If the GROUP= option is specified, this option is ignored.

LINEATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)
specifies the attributes of the loess curve. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Line Options" on page 564 for available *line-options*.

Default: The GraphFit style element.

NAME=*"string"*
assigns a name to a plot statement for reference in other template statements.

Default: no default

Restriction: The *string* is case sensitive, cannot contain spaces, and must define a unique name within the template.

Interaction: The *string* is used as the default legend label if the LEGENDLABEL= option is not used.

The specified name is used primarily in legend statements to coordinate the use of colors and line patterns between the graph and the legend.

PRIMARY=*boolean*
specifies that the data columns for this plot and the plot type be used for determining default axis features.

Default: FALSE

Restriction: This option is ignored if the plot is placed under a GRIDDED or LATTICE layout block.

Details: This option is needed only when two or more plots within an overlay-type layout contribute to a common axis. For more information, see "When Plots Share Data and a Common Axis" on page 397

TIPFORMAT=(*role-format-list*)
specifies display formats for tip columns.

Default: The column format of the variable assigned to the role or BEST6. if no format is assigned to a numeric column.

(*role-format-list*)
a list of *role-name* = *format* pairs separated by blanks.

```
TIPFORMAT=(Y=6.2)
```

Requirement: To generate tooltips, you must include an ODS GRAPHICS ON statement that has the IMAGEMAP option specified, and write the graphs to the ODS HTML destination.

The columns assigned to the X, Y, and GROUP (if assigned) roles are automatically included in the tooltip information.

TIPLABEL=(*role-label-list*)
specifies display labels for tip columns.

Default: The column label or column name of the variable assigned to the role.

(*role-label-list*)
a list of *role-name* = *"string"* pairs separated by blanks.

```
TIPLABEL=(Y="Loess Fit")
```

Requirement: To generate tooltips, you must include an ODS GRAPHICS ON statement that has the IMAGEMAP option specified, and write the graphs to the ODS HTML destination.

The columns assigned to the X, Y, and GROUP (if assigned) roles are automatically included in the tooltip information.

XAXIS=X | X2
specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

Default: X

Interaction: The overall plot specification and the layout type determine the axis display for the specified axis. For more information, see "How Axis Features are Determined" on page 395.

YAXIS=Y | Y2
specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

Default: Y

Interaction: The overall plot specification and the layout type determine the axis display for the specified axis. For more information, see "How Axis Features are Determined" on page 395.

Chapter 30

MODELBAND Statement

Description and Syntax

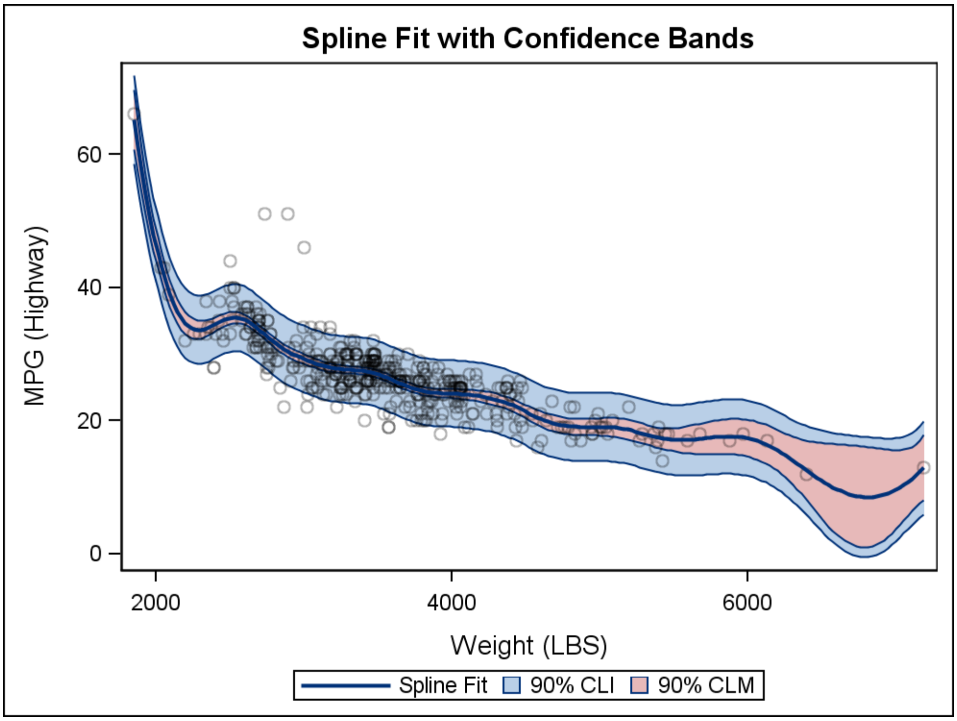
Creates a band showing confidence limits for an associated smoother plot.

MODELBAND "*confidence-name*" <*/option(s)*>;

Example Program and Statement Details

Example Graph

The following graph was generated by the "Example Program" on page 284:



Example Program

```
proc template;
 define statgraph modelband;
  begingraph;
   entrytitle "Spline Fit with Confidence Bands";
   layout overlay;
     modelband "cliband" / name="confband1" display=all
       legendlabel="90% CLI" fillattrs=GraphConfidence;
     modelband "clmband" / name="confband2" display=all
       legendlabel="90% CLM" fillattrs=GraphConfidence2;
     scatterplot x=weight y=mpg_highway /
       datatransparency=.7;
     pbsplineplot x=weight y=mpg_highway / name="fitline"
       clm="clmband" cli="cliband"
       alpha=.1 legendlabel="Spline Fit";
     discretelegend "fitline" "confband1" "confband2";
   endlayout;
  endgraph;
 end;
run;

proc sgrender data=sashelp.cars template=modelband;
run;
```

Statement Summary

A MODELBAND statement must be associated with a smoother statement (LOESSPLOT, REGRESSIONPLOT, or PBSPLINEPLOT) that specifies a fitted model and a type of confidence level to compute. The example program uses a PBSPLINEPLOT statement.

Required Arguments

"*confidence-name*"

specifies the case-sensitive name assigned to a confidence option in a smoother plot statement.

Requirement:*confidence-name* must have been assigned to the CLM= or CLI= option on a smoother plot statement such as LOESSPLOT, REGRESSIONPLOT, or PBSPLINEPLOT.

Options

Option	Description
CURVELABELATTRS	Specifies the color and font attributes of the upper and lower band labels.
CURVELABELLOWER	Specifies a label for the lower band limit.
CURVELABELUPPER	Specifies a label for the upper band limit.
CURVELABELLOCATION	Specifies the location of the band labels relative to the plot area.
CURVELABELPOSITION	Specifies the position of the band label relative to the band line.
DATATRANSPARENCY	Specifies the degree of the transparency of the band.
DISPLAY	Specifies whether to display an outlined band area, a filled band area, or an outlined and filled band area.
FILLATTRS	Specifies the appearance of the band area.
LEGENDLABEL	Specifies a label for use in a legend.
NAME	Assigns a name to a plot statement for reference in other template statements.
OUTLINEATTRS	Specifies the properties of the band lines.
TIPFORMAT	Specifies display formats for information defined by roles.
TIPLABEL	Specifies display labels for information defined by roles.
XAXIS	Specifies whether data are mapped to the primary X (bottom) axis or the secondary X2 (top) axis.

Option	Description
YAXIS	Specifies whether data are mapped to the primary Y (left) axis or the secondary Y2 (right) axis.

CURVELABELATTRS=*style-element* | *style-element* (*text-options*) | (*text-options*)
specifies the color and font attributes of the upper and lower band labels. See “General Syntax for Attribute Options” on page 563 for the syntax on using a *style-element* and “Text Options” on page 566 for available *text-options*.

Default: The GraphValueText style element.

Interaction: For this option to take effect, the CURVELABELLOWER= or CURVELABELUPPER= option must also be specified.

Interaction: If the smoother statement’s GROUP= option is specified, this option is ignored.

CURVELABELLOWER=*"string"*
specifies a label for the lower band limit.

Default: no default

Interaction: If the smoother statement’s GROUP= option is specified, this option is ignored.

The font and color attributes for the label are specified by the CURVELABELATTRS= option.

CURVELABELUPPER=*"string"*
specifies a label for the upper band limit.

Default: no default

Interaction: If the smoother statement’s GROUP= option is specified, this option is ignored.

The font and color attributes for the label are specified by the CURVELABELATTRS= option.

CURVELABELLOCATION=INSIDE | OUTSIDE
specifies the location of the band labels relative to the plot area.

Default: INSIDE

INSIDE
inside the plot area

OUTSIDE
outside the plot area

Restriction: OUTSIDE cannot be used when the MODELBAND is used in multi-cell layouts such as LATTICE, DATAPANEL, or DATALATTICE, where axes might be external to the grid.

Interaction: This option is used in conjunction with the CURVELABELPOSITION= option to determine where the band labels appear. For more information, see “Location and Position of Curve Labels” on page 123.

Interaction: For this option to take effect, the CURVELABELLOWER= or CURVELABELUPPER= option must also be specified.

CURVELABELPOSITION=AUTO | MAX | MIN | START | END

specifies the position of the band label relative to the band line.

Default: AUTO when CUVELABELLOCATION=OUTSIDE END when CURVELABELLOCATION=INSIDE

AUTO

Only used when CURVELABELPOSITION=OUTSIDE. Band labels are positioned automatically near the band boundary along unused axes whenever possible (typically Y2 and X2).

MAX

Forces the band label to appear near maximum band values (typically, upper right)

MIN

Forces the band label to appear near minimum band values (typically, lower left)

START

Only used when CURVELABELLOCATION=INSIDE. Forces the band label to appear near the beginning of the band. Particularly useful when the curve line has a spiral shape.

END

Only used when CURVELABELLOCATION=INSIDE. Forces the band label to appear near the end of the band. Particularly useful when the curve line has a spiral shape.

Interaction: For this option to take effect, the CURVELABELLOWER= or CURVELABELUPPER= option must also be specified.

Restriction: The AUTO setting is not supported if CURVELABELLOCATION=INSIDE is specified. The START and END settings are not supported if CURVELABELLOCATION=OUTSIDE is specified.

Interaction: This option is used in conjunction with the CURVELABELLOCATION= option to determine where the band label appears. For more information, see "Location and Position of Curve Labels" on page 123.

DATATRANSPARENCY=*number*

specifies the degree of the transparency of the band. This option affects both the fill and outline.

Default: 0

Range: 0 (opaque) to 1 (entirely transparent)

DISPLAY=STANDARD | ALL | (*display-options*)

specifies whether to display an outlined area, a filled area, or an outlined and filled modelband area.

Default: The GraphBand:DisplayOpts style reference.

STANDARD

displays a filled band with no outlined

ALL

displays an outlined, filled band

(*display-options*)

a list of options, enclosed in parentheses, that must include one of the following:

OUTLINE—displays an outlined band

FILL—displays a filled band

Use the OUTLINEATTRS= and FILLATTRS= options to control the appearance of the band.

FILLATTRS=*style-element* | *style-element* (*fill-options*) | (*fill-options*)
specifies the appearance of the filled modelband area. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Fill Options" on page 564 for available *fill-options*.

Default:

- For non-grouped data, the GraphConfidence:Color style reference.
- For grouped data, the GraphData1:Color - GraphDataN:Color style references.

Interaction: For this option to have any effect, the fill must be enabled by the ODS style or the DISPLAY= option.

LEGENDLABEL= *"string"*
specifies a label to be used in a legend.

Default: The *string* specified on the NAME= option.

Interaction: The smoother statement's GROUP= option overrides this option.

NAME=*"string"*
assigns a name to a plot statement for reference in other template statements.

Default: no default

Restriction: The *string* is case sensitive, cannot contain spaces, and must define a unique name within the template.

Interaction: The *string* is used as the default legend label if the LEGENDLABEL= option is not used.

The specified name is used primarily in legend statements to coordinate the use of colors and line patterns between the graph and the legend.

OUTLINEATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)
specifies the attributes of the modelband outlines. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Line Options" on page 564 for available *line-options*.

Default:

- For non-grouped data, the GraphConfidence style element.
- For grouped data, the GraphData1: ContrastColor - GraphDataN:ContrastColor style references.

Interaction: If DISPLAY=(FILL), this option has no effect.

TIPFORMAT=(*role-format-list*)
specifies display formats for tip columns.

Default: The column format of the variable assigned to the role or BEST6. if no format is assigned to a numeric column.

(*role-format-list*)
a list of *role-name* = *format* pairs separated by blanks.

The *role-names* X, LIMITLOWER, LIMITUPPER, GROUP, and INDEX are available to indicate which tooltip values to format.

```
TIPFORMAT=(LIMITUPPER=5.3
 LIMITLOWER=5.3)
```

Requirement: To generate tooltips, you must include an ODS GRAPHICS ON statement that has the IMAGEMAP option specified, and write the graphs to the ODS HTML destination.

TIPLABEL=(*role-label-list*)
specifies display labels for tip columns.

Default: The column label or column name of the variable assigned to the role.

(*role-label-list*)
a list of *role-name* = *"string"* pairs separated by blanks.

The *role-names* X, LIMITLOWER, LIMITUPPER, GROUP, and INDEX are available to indicate which tooltip values to label.

```
TIPLABEL=(LIMITUPPER="90% CLM"
LIMITLOWER="90% CLM")
```

Requirement: To generate tooltips, you must include an ODS GRAPHICS ON statement that has the IMAGEMAP option specified, and write the graphs to the ODS HTML destination.

XAXIS=X | X2
specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

Default: X

Interaction: The overall plot specification and the layout type determine the axis display. For more information, see “How Axis Features are Determined” on page 395.

The setting for this option should be the same as for the smoother statement referenced by “"confidence-name" ”.

YAXIS=Y | Y2
specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

Default: Y

Interaction: The overall plot specification and the layout type determine the axis display. For more information, see “How Axis Features are Determined” on page 395.

The setting for this option should be the same as for the smoother statement referenced by “"confidence-name" ”.

Chapter 31

NEEDLEPLOT Statement

Description and Syntax

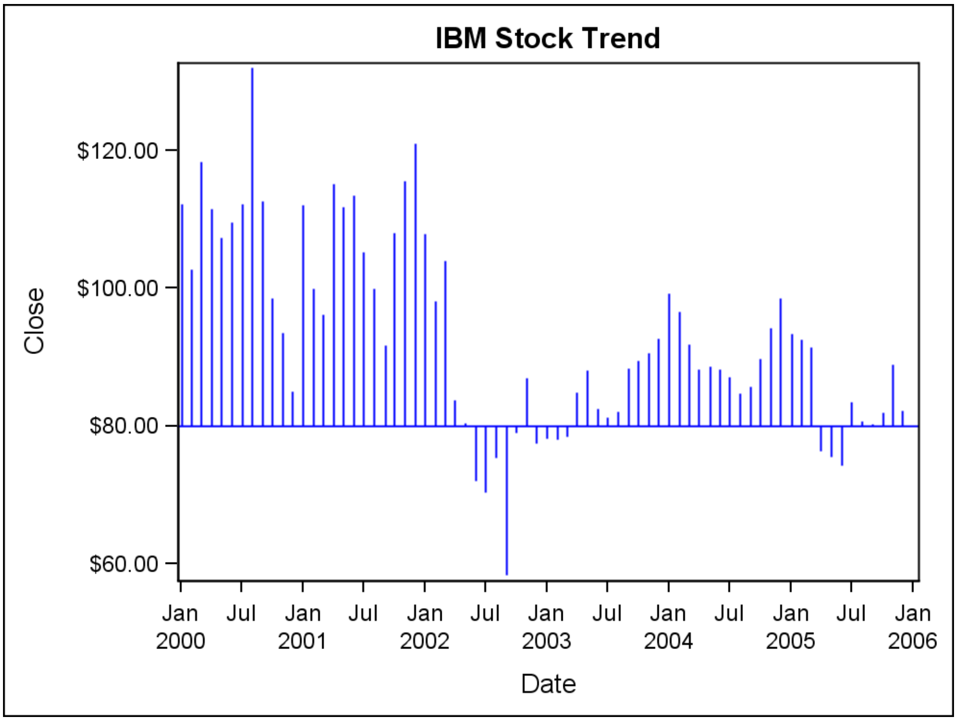
Creates a plot of observations as points connected to a baseline by vertical line segments.

NEEDLEPLOT X = *column* | *expression*
Y = *numeric-column* | *expression* </option(s)>;

Example Program and Statement Details

Example Graph

The following graph was generated by the "Example Program" on page 292:



Example Program

```
proc template;
  define statgraph needleplot;
    begingraph;
      entrytitle "IBM Stock Trend";
      layout overlay;
        needleplot x=date y=close /
          baselineintercept=80 lineattrs=(color=blue);
      endlayout;
    endgraph;
  end;
run;

proc sgrender data=sashelp.stocks template=needleplot;
  where stock="IBM" and date > "31dec1999"d;
run;
```

Statement Summary

In the NEEDLEPLOT statement, the X variable can specify character or numeric values. The Y variable must specify numeric values. For character columns, the X-axis is always of TYPE=DISCRETE. For numeric columns, the X-axis is of TYPE=LINEAR by default.

The Y-axis is of TYPE=LINEAR by default.

Required Arguments

X=*column* | *expression*
  specifies a column or expression for the X values.

Y=*numeric-column* | *expression*
specifies a numeric column or numeric expression for the Y values.

Options

Statement Option	Description
BASELINEINTERCEPT	Specifies the Y-intercept for the baseline.
DATALABEL	Specifies labels at the data points.
DATALABELATTRS	Specifies the color and font attributes of the data labels.
DATATRANSPARENCY	Specifies the degree of the transparency of the needles and the markers, if displayed.
DISPLAY	Specifies whether to display needle lines with or without markers.
GROUP	Creates a separate needle plot for each unique group value of the specified column.
INDEX	Specifies indices for mapping needle attributes (color, marker symbol, and line pattern) to one of the GraphData1 - GranphDataN style elements.
LEGENDLABEL	Specifies a label for use in a legend.
LINEATTRS	Specifies the properties of the needles.
MARKERATTRS	Specifies the attributes of the data markers.
NAME	Assigns a name to a plot statement for reference in other template statements.
PRIMARY	Specifies that the data columns for this plot be used for determining default axis features.
ROLENAME	Specifies roles for information defined by data columns.
TIP	Specifies the information to display when the cursor is positioned over a needle line or marker.
TIPFORMAT	Specifies display formats for information defined by roles.
TIPLABEL	Specifies display labels for information defined by roles.

Statement Option	Description
URL	Specifies an HTML page to display when a needle line is selected.
XAXIS	Specifies whether data are mapped to the primary X (bottom) axis or the secondary X2 (top) axis.
YAXIS	Specifies whether data are mapped to the primary Y (left) axis or the secondary Y2 (right) axis.

BASELINEINTERCEPT=*number* | RELATIVE
specifies the Y-intercept for the baseline.

Default: 0

number
The Y-intercept value to use for the baseline.

RELATIVE
The baseline is placed at the Y-axis tick mark closest to the minimum of the range for the needle data points.

DATALABEL=*column*
specifies labels at the data points.

Default: no default

The position of the labels is adjusted to prevent the labels from overlapping.

DATALABELATTRS=*style-element* | *style-element* (*text-options*) | (*text-options*)
specifies the color and font attributes of the data labels. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Text Options" on page 566 for available *text-options*.

Default:

- For non-grouped data, the GraphValueText style element.
- For grouped data, the GraphData1:ContrastColor - GraphDataN:ContrastColor style references.

Interaction: For this option to have any effect, the DATALABEL= option must also be specified.

DATATRANSPARENCY=*number*
specifies the degree of the transparency of the needles and markers, if displayed.

Default: 0

Range: 0 (opaque) to 1 (entirely transparent)

DISPLAY=STANDARD | ALL | *display-options*
specifies whether to display needle lines with or without markers.

Default: STANDARD

STANDARD
displays needle lines without markers.

ALL
displays needle lines with markers.

(*display-options*)
a list of options, enclosed in parentheses. Currently only the following option is available:

MARKERS—displays needle lines with markers

Use the MARKERATTRS= and LINEATTRS= options to control the appearance of the line and markers.

GROUP=*column* | *expression*
creates a distinct set of needles, markers, and data labels for each unique group value of the specified column.

Default: Each distinct group value might be represented in the graph by a different combination of color, line pattern, and marker symbol. These vary according to the ContrastColor, LineStyle, and MarkerSymbol attributes of the GraphData1 - GraphDataN style elements.

Interaction: The group values are mapped in the order of the data, unless the INDEX= option is used to alter the default sequence of marker symbols, colors, and line patterns.

Interaction: The marker size is set by the MARKERATTRS= option.

The representations that are used to identify the groups can be overridden. For example, each distinct group value might be represented by a different line pattern, but the LINEATTRS= (PATTERN=*pattern*) option could be used to assign the same line pattern to all of the plot's line patterns, letting line color indicate group values. Likewise, LINEATTRS= (COLOR= *color*) could be used to assign the same color to all lines, letting line pattern indicate group values.

INDEX=*positive-integer-column*
specifies indices for mapping needle attributes (color, marker symbol, and line pattern) to one of the GraphData1 - GranphDataN style elements.

Default: no default

Interaction: For this option to have any effect, the GROUP= option must also be specified.

Interaction: All of the indexes for a specific group value must be the same. Otherwise, the results are unpredictable.

Interaction: If this option is not used, then the group values are mapped in the order of the data.

Interaction: The index values are 1-based indices. For the style elements in GraphData1 - GraphDataN, if the index value is greater than N, then a modulo operation remaps that index value to a number less than N to determine which style element to use.

Discussion: Indexing can be used to collapse the number of groups that are represented in a graph. For more information, see "Remapping Groups for Grouped Data" on page 121.

LEGENDLABEL= *"string"*
Specifies a label for use in a legend.

Default: The Y-variable label. If a label is not defined, the Y-variable name.

Interaction: If the GROUP= option is specified, this option is ignored.

LINEATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)

specifies the attributes of the needles for the data points. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Line Options" on page 564 for available *line-options*.

Default:

- For non-grouped data, the GraphDataDefault style element.
- For grouped data, the ContrastColor, LineStyle and LineThickness attributes of the GraphData1 - GraphDataN style elements.

MARKERATTRS=*style-element* | *style-element* (*marker-options*) | (*marker-options*)

specifies the attributes of the data markers. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Marker Options" on page 565 for available *marker-options*.

Default:

- For non-grouped data, the GraphDataDefault style element.
- For grouped data, the MarkerSymbol, MarkerSize, and ContrastColor attributes of the GraphData1 - GraphDataN style elements.

Interaction: The DISPLAY= option must display markers for this option to have any effect.

NAME=*"string"*

assigns a name to a plot statement for reference in other template statements.

Default: no default

Restriction: The *string* is case sensitive, cannot contain spaces, and must define a unique name within the template.

The specified name is used primarily in legend statements to coordinate the use of colors, marker symbols, and line patterns between the graph and the legend.

PRIMARY=*boolean*

specifies that the data columns for this plot be used for determining default axis features.

Default: FALSE

Restriction: This option is ignored if the plot is placed under a GRIDDED or LATTICE layout block.

Details: This option is needed only when two or more plots within an overlay-type layout contribute to a common axis. For more information, see "When Plots Share Data and a Common Axis" on page 397

ROLENAME=(*role-name-list*)

specifies user-defined roles for information contained in data columns.

Default: no default

(*role-name-list*)

a blank-separated list of *rolename* = *column* pairs.

For example, ROLENAME= (TIP1=OBS) assigns the column OBS to the user-defined role TIP1.

Requirement: The role names that you choose must be unique and different from the pre-defined roles X, Y, DATALABEL, INDEX, and GROUP.

This option provides a way to add to the data columns that appear in tooltips specified by the TIP= option.

TIP=(*role-list*)

specifies the information to display when the cursor is positioned over a needle line or marker. If this option is used, it replaces all the information displayed by default. Roles for columns that do not contribute to the needle plot can be specified along with roles that do.

Default: The columns assigned to these roles are automatically included in the tooltip information: X, Y, DATALABEL, and GROUP.

(*role-list*)

an ordered, blank-separated list of unique NEEDLEPLOT and user-defined roles. NEEDLEPLOT roles include X, Y, DATALABEL, and GROUP.

User-defined roles are defined with the ROLENAME= option.

The following example displays tooltips for the columns assigned to the roles X and Y as well as the column OBS, which is not assigned to any pre-defined NeedlePlot role. The OBS column must first be assigned a role.

```
ROLENAME=(TIP1=OBS)
TIP= (TIP1 X Y)
```

Requirement: To generate tooltips, you must include an ODS GRAPHICS ON statement that has the IMAGEMAP option specified, and write the graphs to the ODS HTML destination.

Interaction: The labels and formats for the TIP variables can be controlled with the TIPLABEL= and TIPFORMAT= options.

TIPFORMAT=(*role-format-list*)

specifies display formats for tip columns.

Default: The column format of the variable assigned to the role or BEST6. if no format is assigned to a numeric column.

(*role-format-list*)

a list of *rolename* = *format* pairs separated by blanks.

```
ROLENAME=(TIP1=OBS)
TIP=(TIP1 X Y)
TIPFORMAT=(TIP1=4.)
```

Requirement: This option provides a way to control the formats of columns that appear in tooltips. Only the roles that appear in the TIP= option are used. Columns must be assigned to the roles for this option to have any effect. See the ROLENAME= option.

TIPLABEL=(*role-label-list*)

specifies display labels for tip columns.

Default: The column label or column name of the variable assigned to the role.

(*role-label-list*)

a list of *rolename* = *"string"* pairs separated by blanks.

```
ROLENAME=(TIP1=OBS)
TIP=(TIP1 X Y)
TIPLABEL=(TIP1="Observation #")
```

Requirement: This option provides a way to control the labels of columns that appear in tooltips. Only the roles that appear in the TIP= option are used. Columns must be assigned to the roles for this option to have any effect. See the ROLENAME= option.

URL=*character-column*

specifies an HTML page to display when a needle or marker is selected.

Default: no default

character-column

each value of the column must be a valid HTML page reference (HREF). For example, *http://www.sas.com/technologies/analytics/index.html*

Requirement: To generate selectable needles, you must include an ODS GRAPHICS ON statement that has the IMAGEMAP option specified, and write the graphs to the ODS HTML destination.

The URL value can be blank for some X and Y pairs, meaning that no action is taken when the corresponding needle or marker is selected. The URL value can be the same for any X and Y pairs. In that case, the same action is taken when the needle or marker is selected for those X and Y pairs.

XAXIS=X | X2

specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

Default: X

Interaction: The overall plot specification and the layout type determine the axis display. For more information, see "How Axis Features are Determined" on page 395.

YAXIS=Y | Y2

specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

Default: Y

Interaction: The overall plot specification and the layout type determine the axis display. For more information, see "How Axis Features are Determined" on page 395.

Chapter 32
PBSPLINEPLOT Statement

Description and Syntax

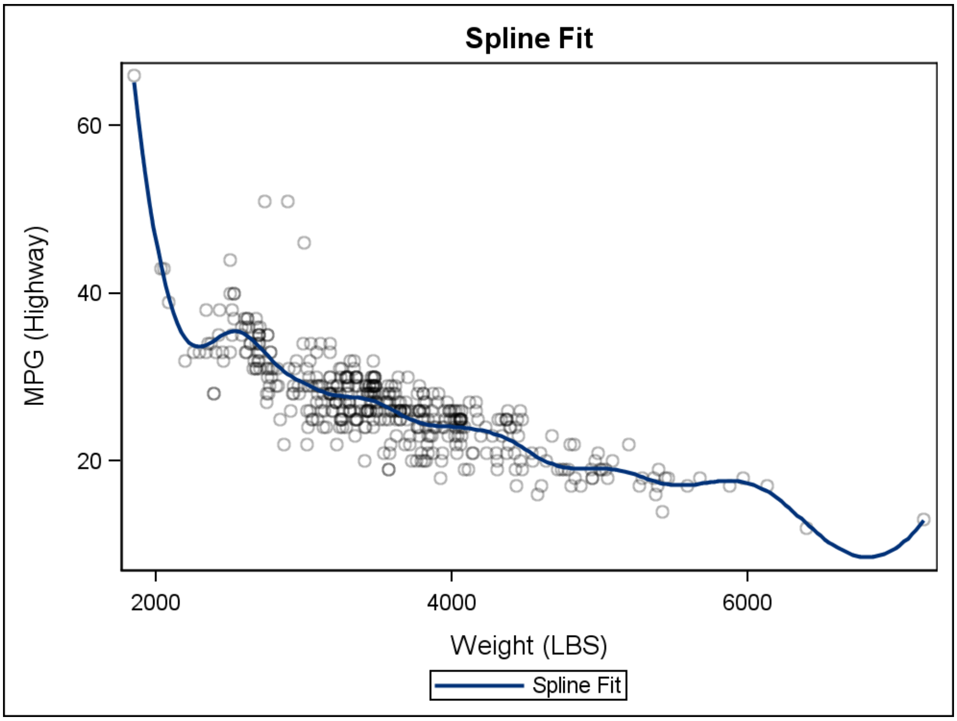
Creates a fitted penalized B-spline curve computed from input data.

PBSPLINEPLOT X = *numeric-column* | *expression*
Y = *numeric-column* | *expression* </ <*regression-options*> <*option(s)*>>;

Example Program and Statement Details

Example Graph

The following graph was generated by the “Example Program” on page 300:



Example Program

```
proc template;
 define statgraph pbsplineplot;
  begingraph;
    entrytitle "Spline Fit";
    layout overlay;
      scatterplot x=weight y=mpg_highway /
        datatransparency=.7;
      pbsplineplot x=weight y=mpg_highway / name="fitline"
        alpha=.05 legendlabel="Spline Fit";
      discretelegend "fitline";
    endlayout;
  endgraph;
 end;
run;

proc sgrender data=sashelp.cars template=pbsplineplot;
run;
```

Statement Summary

The PBSPLINEPLOT statement only supports models of one independent and one dependent variable. For more information about the fitting methodology, see the TRANSREG procedure in the **MISSING Bookref** .

In addition to the penalized B-spline, the PBSPLINEPLOT statement can compute confidence levels for the fitted line. To display the confidence levels,

1. use the CLI= or CLM= option to declare a name for the confidence level

2. use a MODELBAND statement to refer to this name. This statement draws a confidence band from this information. See "MODELBAND Statement" on page 283 for information about how to control the appearance of the confidence band.

Required Arguments

X=*numeric-column* | *expression*
specifies the column for the X values.

Y=*numeric-column* | *expression*
specifies the column for the Y values.

PBSPLINE Regression Options

ALPHA= *positive-number*
specifies the confidence level to compute.

Default: .05

Range: 0 - 1

ALPHA=.05 represents a 95% confidence level.

CLI= *"name"*
produces confidence limits for individual predicted values for each observation.

Default: no default

Interaction: *name* is a unique name within the template that is case sensitive and cannot contain spaces. It must be assigned in order for the confidence limits to be computed. To display confidence limits, you must use this name as the required argument of a MODELBAND statement. See the example in the section "Example Program and Statement Details" on page 299.

CLM= *"name"*
produces confidence limits for a mean predicted value for each observation.

Default: no default

Interaction: *name* is a unique name within the template that is case sensitive and cannot contain spaces. It must be assigned in order for the confidence limits to be computed. To display confidence limits, you must use this name as the required argument of a MODELBAND statement. See the example in the section "Example Program and Statement Details" on page 299.

DEGREE= *non-negative-integer*
specifies the degree of B-spline.

Default: 3

FREQ= *numeric-column*
specifies a variable in the input data set that represents the frequency of occurrence of the current observation, essentially treating the data set as if each observation appeared *n* times, where *n* is the value of the FREQ variable for the observation. Noninteger values of the FREQ variable are truncated to the largest integer less than the FREQ value. The observation is used in the analysis only if the value of the FREQ variable is greater than or equal to 1.

Default: no default

MAXPOINTS= *positive-integer*
specifies the maximum number of predicted points generated for the spline curve as well as any confidence limits.

Default: 201

NKNOTS= *non-negative-integer*
specifies the number of evenly spaced internal knots.

Default: 100

By default, a large number of knots (100) is specified, which allows for an extreme lack of smoothness in the results. However, the final function is typically much smoother due to the penalty. See the section and example on “Penalized B-Splines” in PROC TRANSREG. When SMOOTH=0 is specified, you should typically ask for many fewer knots than the default, since there is no penalty for lack of smoothness. For example, ten or fewer knots is usually enough to follow the functional form found in most data. See “Using Splines and Knots” and “Specifying the Number of Knots” in PROC TRANSREG.

SMOOTH=AUTO | *non-negative-number*
specifies a regression parameter value.

Default: AUTO

With SMOOTH=AUTO, a regression parameter is automatically selected that minimizes a lack-of-smoothness penalty.

You can specify SMOOTH=0 to get an ordinary B-spline fit.

WEIGHT= *numeric-column*
specifies a variable in the input data set that contains values to be used as a priori weights for a penalized B-spline fit. If an observation’s weight is zero, negative, or missing, the observation is deleted from the analysis.

Default: no default

Options

Statement Option	Description
CURVELABEL	Specifies the label of the regression curve.
CURVELABELATTRS	Specifies the color and font attributes of the regression curve label.
CURVELABELLOCATION	Specifies the location of the regression curve label relative to the plot area.
CURVELABELPOSITION	Specifies the position of the regression curve label relative to the curve line.
DATATRANSPARENCY	Specifies the degree of the transparency of the regression curve.
GROUP	Creates a distinct set of lines from just the observations that correspond to each unique group value of the specified column.

Statement Option	Description
INDEX	Specifies indices for mapping line attributes (color and pattern) to one of the GraphData1 - GranphDataN style elements.
LEGENDLABEL	Specifies a label for use in a legend.
LINEATTRS	Specifies the line properties of the regression curve.
NAME	Assigns a name to a plot statement for reference in other template statements.
PRIMARY	Specifies that the data columns for this plot be used for determining default axis features.
TIPFORMAT	Specifies display formats for information defined by roles.
TIPLABEL	Specifies display labels for information defined by roles.
XAXIS	Specifies whether data are mapped to the primary X (bottom) axis or the secondary X2 (top) axis.
YAXIS	Specifies whether data are mapped to the primary Y (left) axis or the secondary Y2 (right) axis.

CURVELABEL=*"string"*

specifies a label for the regression curve.

Default: no default

Interaction: If the GROUP= option is specified, this option is ignored.

The font and color attributes for the label are specified by the CURVELABELATTRS= option.

CURVELABELATTRS=*style-element* | *style-element* (*text-options*) | (*text-options*)

specifies the color and font attributes of the regression curve labels. See “General Syntax for Attribute Options” on page 563 for the syntax on using a *style-element* and “Text Options” on page 566 for available *text-options*.

Default: The GraphValueText style element.

Interaction: For this option to take effect, the CURVELABEL= option must also be specified.

Interaction: If the GROUP= option is specified, this option is ignored.

CURVELABELLOCATION=INSIDE | OUTSIDE

specifies the location of the regression curve label relative to the plot area.

Default: INSIDE

INSIDE

inside the plot area

OUTSIDE
outside the plot area

Restriction: OUTSIDE cannot be used when the PBSPLINEPLOT is used in multicell layouts such as LATTICE, DATAPANEL, or DATALATTICE, where axes might be external to the grid.

Interaction: For this option to take effect, the CURVELABEL= option must also be specified.

Interaction: This option is used in conjunction with the CURVELABELPOSITION= option to determine where the curve labels appear. For more information, see "Location and Position of Curve Labels" on page 123.

CURVELABELPOSITION=AUTO | MAX | MIN | START | END
specifies the position of the regression curve label relative to the curve line.

Default: AUTO when CUVELABELLOCATION=OUTSIDE END when CURVELABELLOCATION=INSIDE

AUTO
Only used when CURVELABELPOSITION=OUTSIDE. The regression curve label is positioned automatically near the curve boundary along unused axes whenever possible (typically Y2 and X2) to avoid collision with tick values.

MAX
Forces the regression curve label to appear near maximum curve values (typically, upper right)

MIN
Forces the regression curve label to appear near minimum curve values (typically, lower left)

START
Only used when CURVELABELLOCATION=INSIDE. Forces the regression curve label to appear near the beginning of the curve. Particularly useful when the curve line has a spiral shape.

END
Only used when CURVELABELLOCATION=INSIDE. Forces the regression curve label to appear near the end of the curve. Particularly useful when the curve line has a spiral shape.

Interaction: For this option to take effect, the CURVELABEL= option must also be specified.

Interaction: The AUTO setting is not supported if CURVELABELLOCATION=INSIDE is specified. The START and END settings are not supported if CURVELABELLOCATION=OUTSIDE is specified.

Interaction: This option is used in conjunction with the CURVELABELLOCATION= option to determine where the regression curve label appears. For more information, see "Location and Position of Curve Labels" on page 123.

DATATRANSPARENCY=*number*
specifies the degree of the transparency of the curve.

Default: 0

Range: 0 (opaque) to 1 (entirely transparent)

GROUP=*column* | *expression*
creates a distinct set of curves from just the observations that correspond to each unique group value of the specified column.

Default: Each distinct group value might be represented in the graph by a different combination of color and line pattern. Line colors vary according to the GraphData1:ContrastColor - GraphDataN:ContrastColor style references, and line patterns vary according to the GraphData1:LineStyle - GraphDataN:LineStyle style references.

Restriction: The input data must be sorted by the GROUP= column.

Interaction: The group values are mapped in the order of the data, unless the INDEX= option is used to alter the default sequence of line colors and line patterns.

The LINEATTRS= option can be used to override the representations that are used to identify the groups. For example, LINEATTRS=(PATTERN=SOLID) can be used to assign the same pattern to all of the loess curves, letting the line color distinguish group values. Likewise, LINEATTRS=(COLOR=BLACK) can be used to assign the same color to all of the curves, letting the line pattern distinguish group values.

INDEX=*positive-integer-column* | *expression*

specifies indices for mapping line attributes (color and pattern) to one of the GraphData1 - GranphDataN style elements.

Default: no default

Interaction: For this option to take effect, the GROUP= option must also be specified.

Interaction: All of the indexes for a specific group value must be the same. Otherwise, the results are unpredictable.

Interaction: If this option is not used, then the group values are mapped in the order of the data.

Interaction: The index values are 1-based indices. For the style elements in GraphData1 - GraphDataN, if the index value is greater than N, then a modulo operation remaps that index value to a number less than N to determine which style element to use.

Discussion: Indexing can be used to collapse the number of groups that are represented in a graph. For more information, see "Remapping Groups for Grouped Data" on page 121.

LEGENDLABEL= *"string"*

specifies a label for use in a legend.

The *string* specified on the NAME= option.

Interaction: If the GROUP= option is specified, this option is ignored.

LINEATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)

specifies the line attributes of the regression curve. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Line Options" on page 564 for available *line-options*.

Default: The GraphFit style element.

NAME=*"string"*

assigns a name to a plot statement for reference in other template statements.

Default: no default

Restriction: The *string* is case sensitive, cannot contain spaces, and must define a unique name within the template.

Interaction: The *string* is used as the default legend label if the LEGENDLABEL= option is not used.

The specified name is used primarily in legend statements to coordinate the use of colors and line patterns between the graph and the legend.

PRIMARY=*boolean*

specifies that the data columns for this plot and the plot type be used for determining default axis features.

Default: FALSE

Restriction: This option is ignored if the plot is placed under a GRIDDED or LATTICE layout block.

Details: This option is needed only when two or more plots within an overlay-type layout contribute to a common axis. For more information, see "When Plots Share Data and a Common Axis" on page 397

TIPFORMAT=(*role-format-list*)

specifies display formats for tip columns.

Default: The column format of the variable assigned to the role or BEST6. if no format is assigned to a numeric column.

(*role-format-list*)

a list of *role-name* = *format* pairs separated by blanks.

```
TIPFORMAT=(Y=6.2)
```

Requirement: To generate tooltips, you must include an ODS GRAPHICS ON statement that has the IMAGEMAP option specified, and write the graphs to the ODS HTML destination.

The columns assigned to the X, Y, and GROUP (if assigned) roles are automatically included in the tooltip information.

TIPLABEL=(*role-label-list*)

specifies display labels for tip columns.

Default: The column label or column name of the variable assigned to the role.

(*role-label-list*)

a list of *role-name* = *"string"* pairs separated by blanks.

```
TIPLABEL=(Y="Spline Regression")
```

Requirement: To generate tooltips, you must include an ODS GRAPHICS ON statement that has the IMAGEMAP option specified, and write the graphs to the ODS HTML destination.

The columns assigned to the X, Y, and GROUP (if assigned) roles are automatically included in the tooltip information.

XAXIS=X | X2

specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

Default: X

Interaction: The overall plot specification and the layout type determine the axis display. For more information, see "How Axis Features are Determined" on page 395.

YAXIS=Y | Y2

specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

Default: Y

Interaction: The overall plot specification and the layout type determine the axis display. For more information, see "How Axis Features are Determined" on page 395.

Chapter 33
REFERENCELINE Statement

Description and Syntax

Creates a horizontal or vertical reference line.

Syntax for creating a line perpendicular to the X-axis at an X-intercept:

REFERENCELINE X = *x-axis-value* | *column* | *expression* *</option(s)>*;

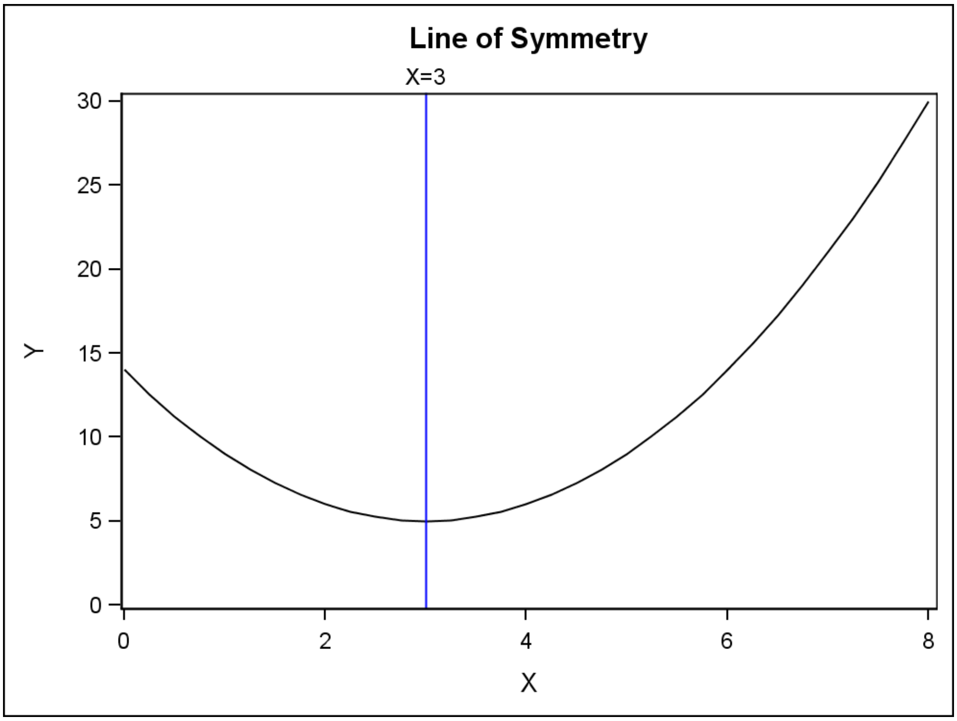
Syntax for creating a line perpendicular to the Y-axis at a Y-intercept:

REFERENCELINE Y = *y-axis-value* | *column* | *expression* *</option(s)>*;

Example Program and Statement Details

Example Graph

The following graph was generated by the “Example Program” on page 310:



Example Program

```
proc template;
  define statgraph referenceline;
    begingraph;
      entrytitle "Line of Symmetry";
      layout overlay / yaxisopts=(linearopts=(viewmin=0));
        seriesplot x=x y=y;
        referenceline x=3 /
          lineattrs=(color=blue) curvelabel="X=3";
      endlayout;
   endgraph;
  end;
run;

data test;
  do X=0 to 8 by .25;
    Y=(x-3)*(x-3) + 5;
    output;
  end;
run;

proc sgrender data=test template=referenceline;
run;
```

Statement Summary

Reference lines are always drawn perpendicular to the axes. They are drawn from one axis boundary to the companion boundary (X to X2 or Y to Y2). Axis offsets do not apply to reference lines.

A REFERENCELINE statement can be used only within 2-D overlay-type layouts (OVERLAY, OVERLAYEQUATED, or PROTOTYPE). A stand-alone plot statement that provides a sufficient data range for determining axis extents must be included in the layout. For example, a REFERENCELINE statement can be used with a scatter plot or a histogram.

If a column is used to generate multiple reference lines, the column type (numeric or string) must agree with the type of data presented on the axis.

Arguments

Either the X= or the Y= argument must be used in the REFERENCELINE statement.

X=*x-axis-value* | *column* | *expression*
specifies the X intercept of the reference line(s).

Requirement: If not specified, then Y must be specified.

By default, if the value specified for the X= argument is outside of the data range, then the data range is extended to include the specified intercept. This behavior can be changed with the CLIP== option

Values must agree in type with the data type of the X axis. For example, you should use numeric SAS date or time values (or SAS date/time constants) for a time axis.

Y=*y-axis-value* | *column* | *expression*
specifies the Y intercept of the reference line(s).

Requirement: If not specified, then X must be specified.

By default, if the value specified for the Y= argument is outside of the data range, then the data range is extended to include the specified intercept. This behavior can be changed with the CLIP= option.

Values must agree in type with the data type of the Y axis.

Options

Statement Option	**Description**
CLIP	Specifies whether the data for the reference line(s) are considered when determining the data ranges for the axes.
CURVELABEL	Specifies a label for the reference line(s).
CURVELABELATTRS	Specifies the color and font attributes of the reference line label(s).
CURVELABELLOCATION	Specifies the location of the reference line label relative to the plot area.
CURVELABELPOSITION	Specifies the position of the reference line label relative to the reference line.
DATATRANSPARENCY	Specifies the degree of the transparency of the reference line(s).

Statement Option	Description
DISCRETEOFFSET	Specifies an amount to offset all reference lines from discrete X or Y values.
LEGENDLABEL	Specifies a label for use in a legend.
LINEATTRS	Specifies the properties of the reference line(s).
NAME	Assigns a name to a plot statement for reference in other template statements.
XAXIS	Specifies whether data are mapped to the primary X (bottom) axis or the secondary X2 (top) axis.
YAXIS	Specifies whether data are mapped to the primary Y (left) axis or the secondary Y2 (right) axis.

CLIP=*boolean*

specifies whether the data for the line are considered when determining the data ranges for the axes.

Default: FALSE

FALSE

The data for the line contributes to data range for each axis. Each axis might be extended to force the display of the line.

TRUE

The data for the line are ignored when establishing axis scales. Each axis scale is determined by the other plots in the layout. This might result in the line not being displayed if its data range is not within the data ranges of the other plots.

CURVELABEL=*"string"* | *column* | *expression*

specifies a label for the reference line(s).

Default: no default

Interaction:

- If the X or Y argument specifies a value, use *"string"*.
- If the X or Y argument specifies a column, use a *column* to define the label for each value.

The font and color attributes for the label are specified by the CURVELABELATTRS= option.

CURVELABELATTRS=*style-element* | *style-element* (*text-options*) | (*text-options*)

specifies the color and font attributes of the reference line label(s). See “General Syntax for Attribute Options” on page 563 for the syntax on using a *style-element* and “Text Options” on page 566 for available *text-options*.

Default: The GraphValueText style element.

Interaction: For this option to take effect, the CURVELABEL= option must also be used.

CURVELABELLOCATION=INSIDE | OUTSIDE

specifies the location of the reference line label relative to the plot area.

Default: INSIDE

INSIDE

inside the plot area

OUTSIDE

outside the plot area

Restriction: OUTSIDE cannot be used when the REFERENCELINE is used in multicell layouts such as LATTICE, DATAPANEL, or DATALATTICE, where axes might be external to the grid.

Interaction: For this option to take effect, the CURVELABEL= option must also be specified.

Interaction: This option is used in conjunction with the CURVELABELPOSITION= option to determine where the line labels appear. For more information, see “Location and Position of Curve Labels” on page 123.

CURVELABELPOSITION=AUTO | MAX | MIN

specifies the position of the reference line label relative to the reference line.

Default: AUTO when CUVELABELLOCATION=OUTSIDE MAX when CURVELABELLOCATION=INSIDE

AUTO

Only used when CURVELABELLOCATION=OUTSIDE. The line label is positioned automatically near the line boundary along unused axes whenever possible (typically Y2 and X2) to avoid collision with tick values.

MAX

Forces the line label to appear near maximum line values (typically, the top or right).

MIN

Forces the line label to appear near minimum line values (typically, the bottom or left).

Restriction: The AUTO setting is not supported if CURVELABELLOCATION=INSIDE is specified.

Interaction: For this option to take effect, the CURVELABEL= option must also be specified.

Interaction: This option is used in conjunction with the CURVELABELLOCATION= option to determine where the line label appears. For more information, see “Location and Position of Curve Labels” on page 123.

DATATRANSPARENCY=*number*

specifies the degree of the transparency of the reference line.

Default: 0

Range: 0 (opaque) to 1 (entirely transparent)

DISCRETEOFFSET= *number*

specifies an amount to offset all reference lines from X values on a discrete X axis, or from Y values on a discrete Y axis.

Note: This feature is for the third maintenance release of SAS 9.2 and later.

Default: 0 (no offset, all reference lines are centered on the X values for a discrete X axis, or on the Y values for a discrete Y axis)

Range: -0.5 to +0.5 where .5 represents half the distance between discrete ticks. A positive offset is to the right for a vertical reference line and up for a horizontal reference line. If the layout's axis options set REVERSE=TRUE, then the offset direction is also reversed.

LEGENDLABEL= *"string"*
specifies a label for use in a legend.

Default: The *string* specified on the NAME= option.

LINEATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)
specifies the attributes of the reference line. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Line Options" on page 564 for available *line-options*.

Default: The GraphReference style element.

NAME=*"string"*
assigns a name to a plot statement for reference in other template statements.

Default: no default

Restriction: The *string* is case sensitive, cannot contain spaces, and must define a unique name within the template.

Interaction: The *string* is used as the default legend label if the LEGENDLABEL= option is not used.

The specified name is used primarily in legend statements to coordinate the use of colors and line patterns between the graph and the legend.

XAXIS=X | X2
specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

Default: X

Restriction: Another plot that establishes a data range for the axis must be included.

Interaction: This option is ignored if the X= argument is not specified.

Interaction: The overall plot specification and the layout type determine the axis display. For more information, see "How Axis Features are Determined" on page 395.

YAXIS=Y | Y2
specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

Default: Y

Restriction: Another plot that establishes a data range for the axis must be included.

Interaction: This option is ignored if the Y= argument is not specified.

Interaction: The overall plot specification and the layout type determine the axis display. For more information, see "How Axis Features are Determined" on page 395.

Chapter 34
REGRESSIONPLOT Statement

Description and Syntax

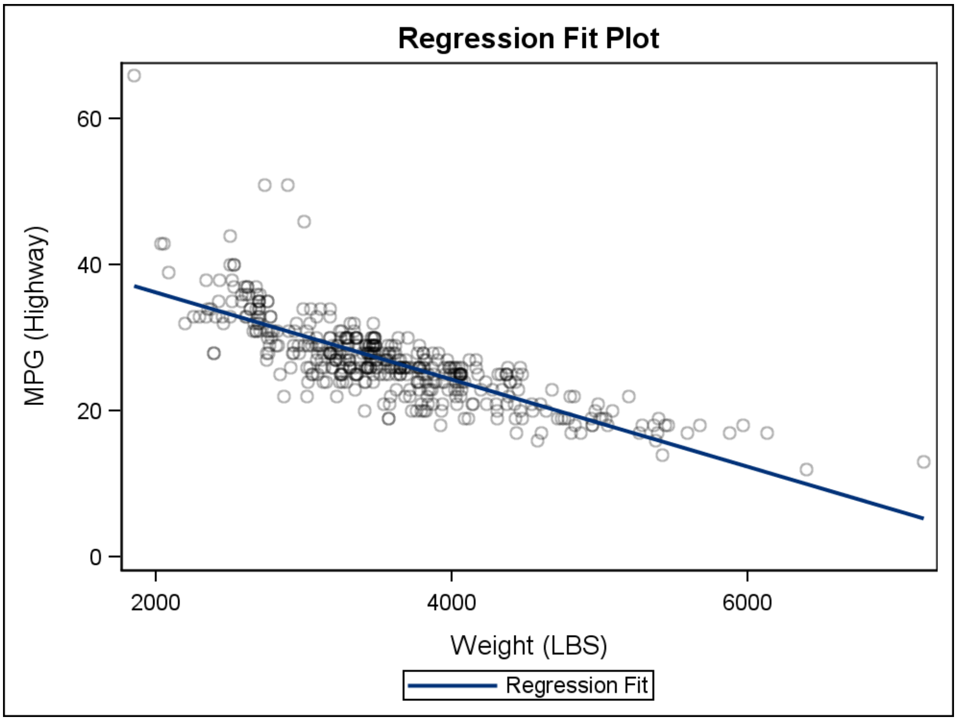
Creates a fitted regression line or curve computed from input data.

REGRESSIONPLOT X = *numeric-column* | *expression*
Y = *numeric-column* | *expression* </ <*regression-options*> <*option(s)*>>;

Example Program and Statement Details

Example Graph

The following graph was generated by the "Example Program" on page 316:



Example Program

```
proc template;
  define statgraph regressionplot;
    begingraph;
      entrytitle "Regression Fit Plot";
      layout overlay;
        scatterplot x=weight y=mpg_highway /
          datatransparency=.7;
        regressionplot x=weight y=mpg_highway /
          name="fitline"
          alpha=.05 legendlabel="Regression Fit";
        discretelegend "fitline";
      endlayout;
   endgraph;
  end;
run;

proc sgrender data=sashelp.cars template=regressionplot;
run;
```

Statement Summary

The REGRESSIONPLOT statement only supports models of one independent and one dependent variable. For more information about the fitting methodology, see the TRANSREG procedure in the **MISSING Bookref** .

In addition to the regression line, the REGRESSIONPLOT statement can compute confidence levels for the fitted line. To display the confidence levels,

1. use the CLI= or CLM= regression option(s) to declare a name for each confidence level
2. use MODELBAND statements to refer to the name(s) and draw a confidence band(s) from this information.

Required Arguments

X=*numeric-column* | *expression*
specifies the column for the X values.

Y=*numeric-column* | *expression*
specifies the column for the Y values.

Regression Options

ALPHA= *positive-number*
specifies the confidence level to compute.

Default: .05

Range: 0 - 1

ALPHA=.05 represents a 95% confidence level.

CLI= *"name"*
produces confidence limits for individual predicted values for each observation. The confidence level is set by the ALPHA= option.

Default: no default

Interaction: *name* is a unique name within the template that is case sensitive and cannot contain spaces. It must be assigned in order for the confidence limits to be computed. To display confidence limits, you must use this name as the required argument of a MODELBAND statement. See the example in "Example Program and Statement Details" on page 315.

CLM= *"name"*
produces confidence limits for a mean predicted value for each observation. The confidence level is set by the ALPHA= option.

Default: no default

Interaction: *name* is a unique name within the template that is case sensitive and cannot contain spaces. It must be assigned in order for the confidence limits to be computed. To display confidence limits, you must use this name as the required argument of a MODELBAND statement. See the example in "Example Program and Statement Details" on page 315.

DEGREE= *non-negative-integer*
specifies the degree of the polynomial.

Default: 1

DEGREE=1 produces a linear fit, DEGREE=2 produces a quadratic fit, DEGREE=3 produces a cubic fit, and so on.

The value of the DEGREE= *d* option corresponds to either of the following PROC TRANSREG specifications for the independent variable: SPLINE(X / DEGREE=*d*) or PBSPLINE(X / DEGREE=*d* LAMBDA=0).

FREQ= *numeric-column*

specifies a variable in the input data set that represents the frequency of occurrence of the current observation, essentially treating the data set as if each observation appeared *n* times, where *n* is the value of the FREQ variable for the observation. Noninteger values of the FREQ variable are truncated to the largest integer less than the FREQ value. The observation is used in the analysis only if the value of the FREQ variable is greater than or equal to 1.

Default: no default

MAXPOINTS= *positive-integer*

specifies the maximum number of predicted points generated for the regression curve as well as any confidence limits.

Default: 201

WEIGHT= *numeric-column*

specifies a variable in the input data set that contains values to be used as a priori weights for a regression fit. If an observation's weight is zero, negative, or missing, the observation is deleted from the analysis.

Default: no default

Options

Statement Option	Description
CURVELABEL	Specifies the label of the regression line.
CURVELABELATTRS	Specifies the color and font attributes of the regression line label.
CURVELABELLOCATION	Specifies the location of the regression line label relative to the plot area.
CURVELABELPOSITION	Specifies the position of the regression line label relative to the regression line.
DATATRANSPARENCY	Specifies the degree of the transparency of the regression line.
GROUP	Creates a distinct set of regression lines from just the observations that correspond to each unique group value of the specified column.
INDEX	Specifies indices for mapping line attributes (color and pattern) to one of the GraphData1 - GranphDataN style elements.
LEGENDLABEL	Specifies a label for use in a legend.
LINEATTRS	Specifies the properties of the regression line.
NAME	Assigns a name to a plot statement for reference in other template statements.

Statement Option	Description
PRIMARY	Specifies that the data columns for this plot be used for determining default axis features.
TIPFORMAT	Specifies display formats for information defined by roles.
TIPLABEL	Specifies display labels for information defined by roles.
XAXIS	Specifies whether data are mapped to the primary X (bottom) axis or the secondary X2 (top) axis.
YAXIS	Specifies whether data are mapped to the primary Y (left) axis or the secondary Y2 (right) axis.

CURVELABEL=*"string"*
specifies a label for the regression line.

Default: no default

Interaction: If the GROUP= option is specified, this option is ignored.

The font and color attributes for the label are specified by the CURVELABELATTRS= option.

CURVELABELATTRS=*style-element* | *style-element* (*text-options*) | (*text-options*)
specifies the color and font attributes of the regression line labels. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Text Options" on page 566 for available *text-options*.

Default: The GraphValueText style element.

Interaction: For this option to take effect, the CURVELABEL= option must also be used.

Interaction: If the GROUP= option is specified, this option is ignored.

CURVELABELLOCATION=INSIDE | OUTSIDE
specifies the location of the regression line label relative to the plot area.

Default: INSIDE

INSIDE
inside the plot area

OUTSIDE
outside the plot area

Restriction: OUTSIDE cannot be used when the REGRESSIONPLOT is used in multi-cell layouts such as LATTICE, DATAPANEL, or DATALATTICE, where axes might be external to the grid.

Interaction: For this option to take effect, the CURVELABEL= option must also be specified.

Interaction: This option is used in conjunction with the CURVELABELPOSITION= option to determine where the line labels appear. For more information, see "Location and Position of Curve Labels" on page 123.

CURVELABELPOSITION=AUTO | MAX | MIN | START | END

specifies the position of the regression line label relative to the regression line.

Default: AUTO when CUVELABELLOCATION=OUTSIDE END when CURVELABELLOCATION=INSIDE

AUTO

Only used when CURVELABELLOCATION=OUTSIDE. The line label is positioned automatically near the line boundary along unused axes whenever possible (typically Y2 and X2) to avoid collision with tick values.

MAX

Forces the line label to appear near maximum line values (typically, upper right).

MIN

Forces the line label to appear near minimum line values (typically, lower left).

START

Only used when CURVELABELLOCATION=INSIDE. Forces the line label to appear near the beginning of the regression line. Particularly useful when the curve line has a spiral shape.

END

Only used when CURVELABELLOCATION=INSIDE. Forces the line label to appear near the end of the regression line. Particularly useful when the curve line has a spiral shape.

Restriction: The AUTO setting is not supported if CURVELABELLOCATION=INSIDE is specified. The START and END settings are not supported if CURVELABELLOCATION=OUTSIDE is specified.

Interaction: For this option to take effect, the CURVELABEL= option must also be specified.

Interaction: This option is used in conjunction with the CURVELABELLOCATION= option to determine where the line label appears. For more information, see “Location and Position of Curve Labels” on page 123.

DATATRANSPARENCY=*number*

specifies the degree of the transparency of the regression line.

Default: 0

Range: 0 (opaque) to 1 (entirely transparent)

GROUP=*column* | *expression*

creates a distinct set of regression lines from just the observations that correspond to each unique group value of the specified column.

Default: Each distinct group value might be represented in the graph by a different combination of line color and line pattern. Line colors vary according to the GraphData1:ContrastColor - GraphDataN:ContrastColor style references, and line patterns vary according to the GraphData1:LineStyle - GraphDataN:LineStyle style references.

Restriction: The input data must be sorted by the GROUP= column.

Interaction: The group values are mapped in the order of the data, unless the INDEX= option is used to alter the default sequence of line colors and line patterns.

The LINEATTRS= option can be used to override the representations that are used to identify the groups. For example, LINEATTRS=(PATTERN=SOLID) can be used to assign the same pattern to all of the lines, letting the line color distinguish group values.

Likewise, LINEATTRS=(COLOR=BLACK) can be used to assign the same color to all of the lines, letting the line pattern distinguish group values.

INDEX=*positive-integer-column* | *expression*

specifies indices for mapping line attributes (color and pattern) to one of the GraphData1 - GranphDataN style elements.

Default: no default

Interaction: For this option to take effect, the GROUP= option must also be specified.

Interaction: All of the indexes for a specific group value must be the same. Otherwise, the results are unpredictable.

Interaction: If this option is not used, then the group values are mapped in the order of the data.

Interaction: The index values are 1-based indices. For the style elements in GraphData1 - GraphDataN, if the index value is greater than N, then a modulo operation remaps that index value to a number less than N to determine which style element to use.

Discussion: Indexing can be used to collapse the number of groups that are represented in a graph. For more information, see "Remapping Groups for Grouped Data" on page 121.

LEGENDLABEL= *"string"*

specifies a label for use in a legend.

Default: The *string* specified on the NAME= option.

Interaction: If the GROUP= option is specified, this option is ignored.

LINEATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)

specifies the attributes of the regression line. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Line Options" on page 564 for available *line-options*.

Default: The GraphFit style element.

NAME=*"string"*

assigns a name to a plot statement for reference in other template statements.

Default: no default

Restriction: The *string* is case sensitive, cannot contain spaces, and must define a unique name within the template.

Interaction: The *string* is used as the default legend label if the LEGENDLABEL= option is not used.

The specified name is used primarily in legend statements to coordinate the use of colors, marker symbols, and line patterns between the graph and the legend.

PRIMARY=*boolean*

specifies that the data columns for this plot and the plot type be used for determining default axis features.

Default: FALSE

Restriction: This option is ignored if the plot is placed under a GRIDDED or LATTICE layout block.

Details: This option is needed only when two or more plots within an overlay-type layout contribute to a common axis. For more information, see "When Plots Share Data and a Common Axis" on page 397

TIPFORMAT=(*role-format-list*)
specifies display formats for tip columns.

Default: The column format of the variable assigned to the role or BEST6. if no format is assigned to a numeric column.

(*role-format-list*)
a list of *role-name* = *format* pairs separated by blanks.

```
TIPFORMAT=(Y=6.2)
```

Requirement: To generate tooltips, you must include an ODS GRAPHICS ON statement that has the IMAGEMAP option specified, and write the graphs to the ODS HTML destination.

The columns assigned to the roles X, Y, and GROUP (if assigned) are automatically included in the tooltip information.

TIPLABEL=(*role-label-list*)
specifies display labels for tip columns.

Default: The column label or column name of the variable assigned to the role.

(*role-label-list*)
a list of *role-name* = *"string"* pairs separated by blanks.

```
TIPLABEL=(Y="Regression Fit")
```

Requirement: To generate tooltips, you must include an ODS GRAPHICS ON statement that has the IMAGEMAP option specified, and write the graphs to the ODS HTML destination.

The columns assigned to the roles X, Y, and GROUP (if assigned) are automatically included in the tooltip information.

XAXIS=X | X2
specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

Default: X

Interaction: The overall plot specification and the layout type determine the axis display. For more information, see "How Axis Features are Determined" on page 395.

YAXIS=Y | Y2
specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

Default: Y

Interaction: The overall plot specification and the layout type determine the axis display. For more information, see "How Axis Features are Determined" on page 395.

Chapter 35

SCATTERPLOT Statement

Description and Syntax

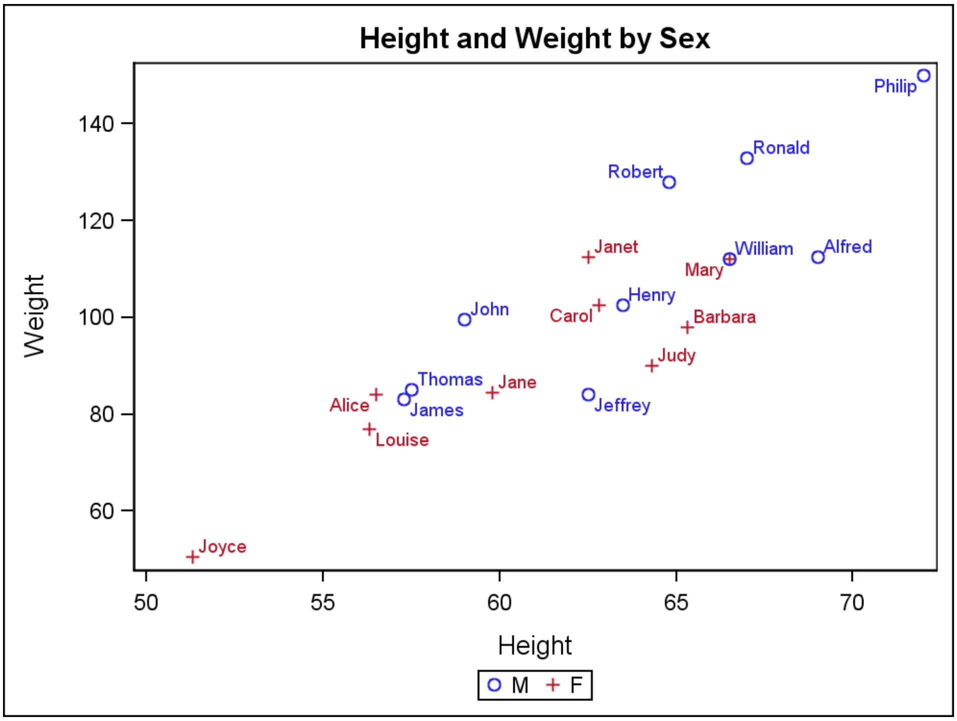
Creates a scatter plot of input data.

SCATTERPLOT X = *column* | *expression*
Y = *column* | *expression* <*/option(s)*>;

Example Program and Statement Details

Example Graph

The following graph was generated by the "Example Program" on page 324:



Example Program

```
proc template;
  define statgraph scatterplot;
    begingraph;
      entrytitle "Height and Weight by Sex";
      layout overlay;
        scatterplot x=height y=weight /
          group=sex name="scatter" datalabel=name;
        discretelegend "scatter";
      endlayout;
    endgraph;
  end;
run;

proc sgrender data=sashelp.class template=scatterplot;
run;
```

Required Arguments

X=*column* | *expression*
specifies the column for the X values.

Y=*column* | *expression*
specifies the column for the Y values.

Options

Statement Option	Description
COLORMODEL	Specifies a style element to be used with the MARKERCOLORGRADIENT= option.
DATALABEL	Specifies a column for marker labels.
DATALABELATTRS	Specifies the color and font attributes of the data labels.
DATATRANSPARENCY	Specifies the degree of the transparency of the markers, data labels, and error bars, when displayed.
DISCRETEOFFSET	Specifies an amount to offset all markers from discrete X values, or discrete Y values, or both.
ERRORBARATTRS	specifies the attributes of any error bars associated with the data points.
FREQ	Specifies a column that indicates a frequency count for each observation of the input data object.
GROUP	Creates a separate marker type for each unique group value of the specified column.
INDEX	Specifies indices for mapping marker attributes (color and symbol) to one of the GraphData1 - GranphDataN style elements.
MARKERATTRS	Specifies the attributes of the data markers.
MARKERCHARACTER	Specifies a column that defines strings to be used instead of marker symbols.
MARKERCHARACTERATTRS	Specifies the color and font attributes of the marker character specified on the MARKERCHARACTER= option.
MARKERCOLORGRADIENT	Specifies the column that is used to map marker colors to a continuous gradient.
NAME	Assigns a name to a plot statement for reference in other template statements.
PRIMARY on page 331	Specifies that the data columns for this plot be used for determining default axis features.
REVERSECOLORMODEL	Specifies whether to reverse a gradient defined by the COLORMODEL= option.

Statement Option	Description
ROLENAME	Specifies roles for information defined by data columns.
TIP	Specifies the information to display when the cursor is positioned over the scatter points.
TIPFORMAT	Specifies display formats for information defined by roles.
TIPLABEL	Specifies display labels for information defined by roles.
URL	Specifies an HTML page to display when a point is selected.
XAXIS	Specifies whether data are mapped to the primary X (bottom) axis or the secondary X2 (top) axis.
XERRORLOWER	Specifies values for the lower endpoints on the X error bars.
XERRORUPPER	Specifies values for the upper endpoints on the X error bars.
YAXIS	Specifies whether data are mapped to the primary Y (left) axis or the secondary Y2 (right) axis.
YERRORLOWER	Specifies values for the lower endpoints on the Y error bars.
YERRORUPPER	Specifies values for the upper endpoints on the Y error bars.

COLORMODEL=*style-element*

specifies a style element to be used with the MARKERCOLORGRADIENT= option.

Default: The ThreeColorAltRamp style element.

style-element

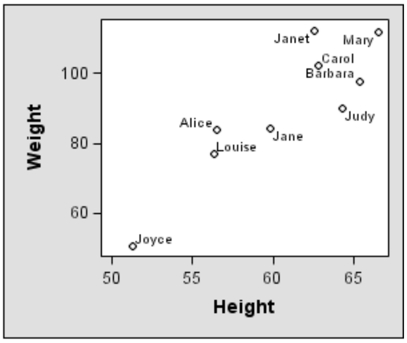
Name of a style element. The style element should contain these style attributes:

STARTCOLOR	color for the smallest data value of the column that is specified on the MARKERCOLORGRADIENT= option
NEUTRALCOLOR	color for the midpoint of the range of the column that is specified on the MARKERCOLORGRADIENT= option
ENDCOLOR	color for the highest data value of the column that is specified on the MARKERCOLORGRADIENT= option

Interaction: For this option to take effect, the MARKERCOLORGRADIENT= option must also be specified.

The REVERSECOLORMODEL= option can be used to reverse the start and end colors of the ramp assigned to the color model.

DATALABEL=*column* | *expression*
specifies a column for marker labels. The label positions are adjusted to prevent them from overlapping.



Default: no default

Interaction: If a numeric *column* is specified and the column has no format, a BEST6. format is applied.

Interaction: This option is ignored if the MARKERCHARACTER= option is used, which displays labels instead of the markers.

DATALABELATTRS=*style-element* | *style-element* (*text-options*) | (*text-options*)
specifies the color and font attributes of the data labels. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Text Options" on page 566 for available *text-options*.

Default:

- For non-grouped data, the GraphDataText style element.
- For grouped data, the GraphData1:ContrastColor - GraphDataN:ContrastColor style references.

Interaction: For this option to take effect, the DATALABEL= option must also be specified.

Interaction: This option is ignored if the MARKERCHARACTER= option is specified.

DATATRANSPARENCY=*number*
specifies the degree of the transparency of the markers, data labels, and error bars, when displayed.

Default: 0

Range: 0 (opaque) to 1 (entirely transparent)

DISCRETEOFFSET=*number*
specifies an amount to offset all markers from X values for a discrete X axis, or from Y values for a discrete Y axis. If both axes are discrete, the offset applies to both axes.

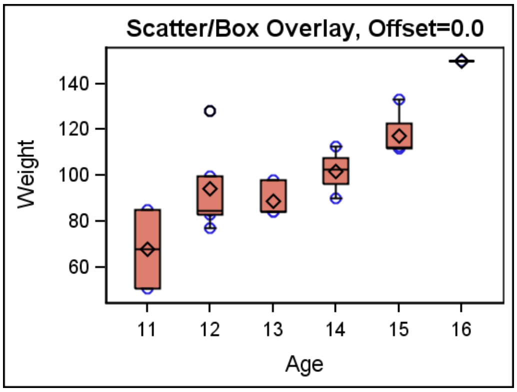
Note: This feature is for the third maintenance release of SAS 9.2 and later.

Default: 0 (no offset, all markers are centered on the X values for a discrete X axis, or on the Y values for a discrete Y axis)

Range: -0.5 to +0.5, where .5 represents half the distance between discrete ticks. A positive offset is to the right on discrete X values and up on discrete Y values. If the layout's axis options set REVERSE=TRUE, the offset direction is also reversed.

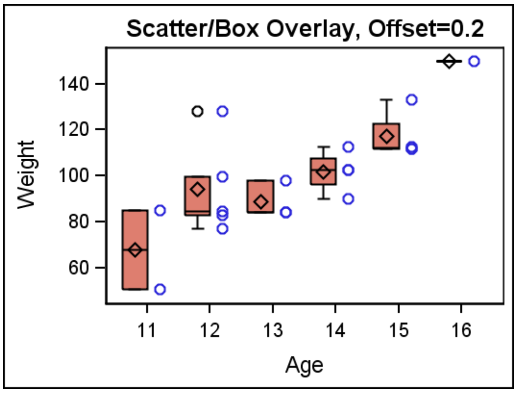
Details: This feature is useful for graphing multiple response variables side by side on a common axis. By default within an overlay-type layout, if a SCATTERPLOT is used with other plots with a discrete axis, the markers are centered on the discrete X values, or discrete Y values, or both. Depending on the data, the markers might be superimposed over other graph data. The following code fragment shows the default positioning when a SCATTERPLOT is used with a BOXPLOT:

```
layout overlay / cycleattrs=true
     xaxisopts=(type=discrete);

  scatterplot x=age y=weight;
  boxplot x=age y=weight;

endlayout;
```



To avoid superimposed plots, you can assign a different offset to each plot statement:

```
layout overlay / cycleattrs=true
        xaxisopts=(type=discrete);
  scatterplot x=age y=weight /
    discreteoffset=0.2;
  boxplot x=age y=weight /
    discreteoffset=-0.2;
endlayout;
```



ERRORBARATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)
specifies the attributes of the error bars that are associated with the data points. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Line Options" on page 564 for available *line-options*.

Default:

- For non-grouped data, the GraphError style element.

- For grouped data, the LineStyle and LineThickness attributes of the GraphError style element and the ContrastColor attribute of the GraphData1 - GraphDataN style elements. (The LineStyle does not apply to the "serif" parts of the error bars.)

Interaction: For this option to take effect, error bars must be displayed by the XERRORLOWER=, XERRORUPPER=, YERRORLOWER=, or YERRORUPPER= options.

FREQ=*numeric-column* | *expression*

specifies a column that indicates a frequency count for each observation of the input data object. If *n* is the value of the FREQ variable for a given observation, then that observation is plotted *n* times.

Default: Each observation is plotted once.

Restriction: If the value of the *numeric-column* is missing or is less than 1, the observation is not used in the analysis. If the value is not an integer, only the integer portion is used.

GROUP=*column* | *expression*

creates a separate marker type for each unique group value of the specified column.

Default: Each distinct group value might be represented in the graph by a different combination of color and marker symbol. Markers vary according to the ContrastColor and MarkerSymbol attributes of the GraphData1 - GraphDataN style elements.

Interaction: The group values are mapped in the order of the data, unless the INDEX= option is used to alter the default sequence of markers and colors.

Interaction: The marker size is set by the MARKERATTRS= option.

Interaction: If the MARKERCHARACTER= and MARKERCOLORGRADIENT= options are used, their settings override the group settings for marker symbol and marker color.

Interaction: The representations that are used to identify the groups can be overridden. For example, each distinct group value is represented by a different marker symbol, but the MARKERATTRS= (SYMBOL= *marker*) option could be used to assign the same symbol to all of the plot's marker symbols, letting marker color indicate group values. Likewise, MARKERATTRS= (COLOR= *color*) could be used to assign the same color to all markers, letting marker symbol indicate group values.

INDEX=*positive-integer-column* | *expression*

specifies indices for mapping marker attributes (color and symbol) to one of the GraphData1 - GranphDataN style elements.

Default: no default

Interaction: For this option to take effect, the GROUP= option must also be specified.

Interaction: All of the indexes for a specific group value must be the same. Otherwise, the results are unpredictable.

Interaction: If this option is not used, then the group values are mapped in the order of the data.

Interaction: If the MARKERCHARACTER= and MARKERCOLORGRADIENT= options are used, their settings override the group settings for marker symbol and marker color.

Interaction: The index values are 1-based indices. For the style elements GraphData1 - GraphDataN, if the index value is greater than N, then a modulo operation remaps that index value to a number less than N to determine which style element to use.

Discussion: Indexing can be used to collapse the number of groups that are represented in a graph. For more information, see "Remapping Groups for Grouped Data" on page 121.

LEGENDLABEL= *"string"*
specifies a label to be used in a legend.

Default: The Y-variable label. If a label is not defined, the Y-variable name.

Interaction: If the GROUP= option is specified, this option is ignored.

MARKERATTRS=*style-element* | *style-element* (*marker-options*) | (*marker-options*)
specifies the attributes of the data markers. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Marker Options" on page 565 for available *marker-options*.

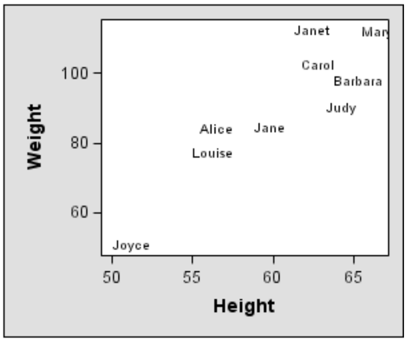
Default:

- For non-grouped data, the GraphDataDefault style element.
- For grouped data, the MarkerSymbol and ContrastColor attributes of the GraphData1 - GraphDataN style elements, and the GraphDataDefault:MarkerSize style reference.

Interaction: If the MARKERCOLORGRADIENT= option is specified, this option's COLOR= setting is ignored.

Interaction: If the MARKERCHARACTER= option is specified, its SYMBOL= and WEIGHT= settings are ignored.

MARKERCHARACTER=*column* | *expression*
specifies a column that defines strings to be used instead of marker symbols.



Default: no default

Interaction: This option overrides the DATALABEL= option.

Interaction: If the GROUP= option is also used, color is displayed for a DISCRETE legend, but the character is not displayed in the legend.

If the GROUP= option is also specified, the same colors are applied to the text strings as would have been applied to markers.

If a numeric column is used, its values are converted to strings using the format associated with the column or BEST6. if no format is defined.

Each string is centered horizontally and vertically at the data point. The data point positions are not adjusted to prevent text overlap.

MARKERCHARACTERATTRS=*style-element* | *style-element* (*text-options*) | (*text-options*)
specifies the color and font attributes of the marker character specified on the MARKERCHARACTER= option. See "General Syntax for Attribute Options" on page

563 for the syntax on using a *style-element* and "Text Options" on page 566 for available *text-options*.

Default:

- For non-grouped data, the GraphDataText style element.
- For grouped data, GraphData1:ContrastColor - GraphDataN:ContrastColor style references.

Interaction: For this option to take effect, the MARKERCHARACTER= option must also be used.

When the GROUP= option is also specified, each distinct group value might be represented by a different color (depending on the ODS style setting or the setting on the INDEX= option). The marker character that is associated with the group is assigned the group color. This option's COLOR= suboption can be used to specify a single color for all marker characters in a graph, without affecting items that have the group color, such as error bars and marker symbols.

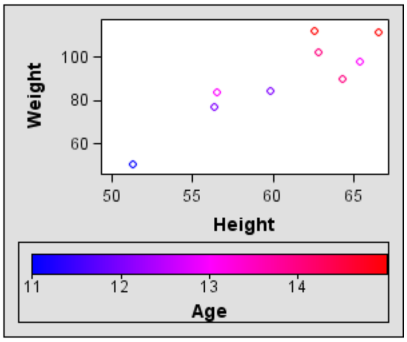
MARKERCOLORGRADIENT=*numeric-column* | *expression*
specifies the column that is used to map marker colors to a continuous gradient.

Default: no default

Restriction: To display a legend with this option in effect, you must use a CONTINUOUSLEGEND statement, not a DISCRETELEGEND statement.

Interaction: This option overrides the COLOR= setting of the MARKERATTRS= or MARKERCHARACTERATTRS= option.

This option can be used to add a second response variable to an analysis. For example, in an analysis of weight by height, an age column might be specified by the MARKERCOLORGRADIENT= option so that the change in the gradient color of the markers reflects the change in age.



By default, the color range is determined by the ThreeColorAltRamp style element. The COLORMODEL= option allows a different color range to be used.

NAME=*"string"*
assigns a name to a plot statement for reference in other template statements.

Default: no default

Restriction: The *string* is case sensitive, cannot contain spaces, and must define a unique name within the template.

The specified name is used primarily in legend statements to coordinate the use of colors and marker symbols between the graph and the legend.

PRIMARY=*boolean*
specifies that the data columns for this plot and the plot type be used for determining default axis features.

Default: FALSE

Restriction: This option is ignored if the plot is placed under a GRIDDED or LATTICE layout block.

Details: This option is needed only when two or more plots within an overlay-type layout contribute to a common axis. For more information, see "When Plots Share Data and a Common Axis" on page 397

REVERSECOLORMODEL=*boolean*

specifies whether to reverse a gradient (color ramp) defined by the COLORMODEL= option.

Default: FALSE

ROLENAME=(*role-name-list*)

specifies user-defined roles for information contained in data columns.

Default: no default

(*rolename-list*)

a blank-separated list of *rolename* = *column* pairs.

For example, ROLENAME= (TIP1=OBS) assigns the column OBS to the user-defined role TIP1.

Requirement: The role names that you choose must be unique and different from the pre-defined roles X, Y, DATALABEL, MARKERCHARACTER, MARKERCOLORGRADIENT, XERRORLOWER, XERRORUPPER, YERRORLOWER, YERRORUPPER, GROUP, and INDEX.

Interaction: For this option to take effect, the TIP= option must also be used.

This option provides a way to add to the data columns that appear in tooltips specified by the TIP= option.

TIP=(*role-list*)

specifies the information to display when the cursor is positioned over the scatter points. If this option is used, it replaces all the information displayed by default. Roles for columns that do not contribute to the scatter plot can be specified along with roles that do.

Default: The columns assigned to these roles are automatically included in the tooltip information: X, Y, DATALABEL, MARKERCHARACTER, MARKERCOLORGRADIENT, XERRORLOWER, XERRORUPPER, YERRORLOWER, YERRORUPPER, FREQ, and GROUP.

(*role-list*)

an ordered, blank-separated list of unique SCATTERPLOT and user-defined roles. SCATTERPLOT roles include X on page 324, Y, DATALABEL, MARKERCHARACTER, MARKERCOLORGRADIENT, XERRORLOWER, XERRORUPPER, YERRORLOWER, YERRORUPPER, GROUP, and INDEX.

User-defined roles are defined with the ROLENAME= option.

The following example displays tooltips for the columns assigned to the roles X, XERRORUPPER and XERRORLOWER, as well as the column OBS, which is not assigned to any pre-defined ScatterPlot role. The OBS column must first be assigned a role.

```
ROLENAME=(TIP1=OBS)
TIP= (TIP1 X XERRORUPPER XERRORLOWER)
```

Requirement: To generate tooltips, you must include an ODS GRAPHICS ON statement that has the IMAGEMAP option specified, and write the graphs to the ODS HTML destination.

The labels and formats for the TIP variables can be controlled with the TIPLABEL= and TIPFORMAT= options.

TIPFORMAT=(*role-format-list*)
specifies display formats for tip columns.

Default: The column format of the variable assigned to the role or BEST6. if no format is assigned to a numeric column.

(*role-format-list*)
a list of *rolename* = *format* pairs separated by blanks.

```
ROLENAME=(TIP1=OBS)
TIP=(TIP1 X XERRORUPPER XERRORLOWER)
TIPFORMAT=(XERRORUPPER=5.3 XERRORLOWER=5.3)
```

Requirement: Columns must be assigned to the roles for this option to have any effect. See the ROLENAME= option.

This option provides a way to control the formats of columns that appear in tooltips. Only the roles that appear in the TIP= option are used.

TIPLABEL=(*role-label-list*)
specifies display labels for tip columns.

Default: The column label or column name of the variable assigned to the role.

(*role-label-list*)
a list of *rolename* = *"string"* pairs separated by blanks.

```
ROLENAME=(TIP1=PCT)
TIP=(TIP1 X XERRORUPPER XERRORLOWER)
TIPLABEL=(XERRORUPPER="1 STD ERROR"
XERRPRLOWER="-1 STD ERROR")
```

Requirement: Columns must be assigned to the roles for this option to have any effect. See the ROLENAME= option.

This option provides a way to control the labels of columns that appear in tooltips. Only the roles that appear in the TIP= option are used.

URL=*character-column*
specifies an HTML page to display when a point is selected.

Default: no default

character-column
each value of the column must be a valid HTML page reference (HREF). For example, *http://www.sas.com/technologies/analytics/index.html*.

Requirement: To generate selectable markers, you must include an ODS GRAPHICS ON statement that has the IMAGEMAP option specified, and write the graphs to the ODS HTML destination.

The URL value can be blank for some X and Y pairs, meaning that no action is taken when the corresponding point is selected. The URL value can be the same for any X and Y pairs. In that case, the same action is taken when the points for those X and Y pairs are selected.

XAXIS=X | X2
specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

Default: X

Interaction: This option is ignored if the X= argument is not specified.

Interaction: The overall plot specification and the layout type determine the axis display for the specified axis. For more information, see "How Axis Features are Determined" on page 395.

XERRORLOWER=*numeric-column* | *expression*
specifies values for the lower endpoints on the X error bars.

Default: The lower segment of the error bars is not drawn.

The values are actual values, not relative values.

The appearance of the error bars is controlled by the ERRORBARATTRS= option.

XERRORUPPER=*numeric-column* | *expression*
specifies values for the upper endpoints on the X error bars.

Default: The upper segment of the error bars is not drawn.

The values are actual values, not relative values.

The appearance of the error bars is controlled by the ERRORBARATTRS= option.

YAXIS=Y | Y2
specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

Default: Y

Interaction: This option is ignored if the Y= argument is not specified.

Interaction: The overall plot specification and the layout type determine the axis display for the specified axis. For more information, see "How Axis Features are Determined" on page 395.

YERRORLOWER=*numeric-column* | *expression*
specifies values for the lower endpoints on the Y error bars.

Default: The lower segment of the error bars is not drawn.

The values are actual values, not relative values.

The appearance of the error bars is controlled by the ERRORBARATTRS= option.

YERRORUPPER=*numeric-column* | *expression*
specifies values for the upper endpoints on the Y error bars.

Default: The upper segment of the error bars is not drawn.

The values are actual values, not relative values.

The appearance of the error bars is controlled by the ERRORBARATTRS= option.

Chapter 36

SCATTERPLOTMATRIX Statement

Description and Syntax

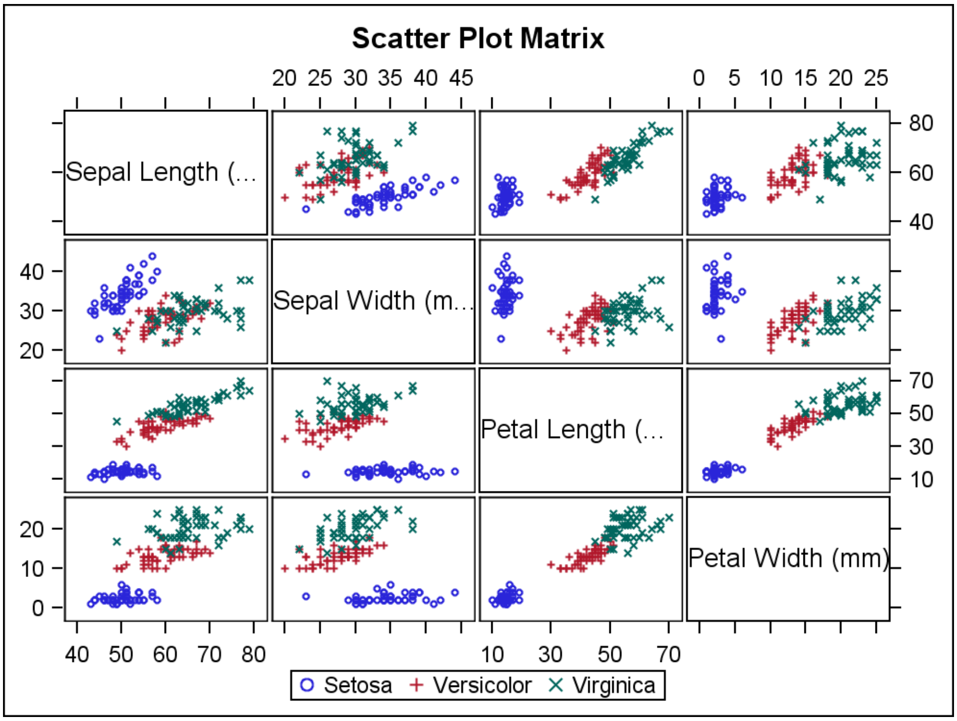
Creates a matrix of all pairwise scatter plots of the specified variables.

SCATTERPLOTMATRIX *numeric-column-list <option(s)>*

Example Program and Statement Details

Example Graph

The following graph was generated by the "Example Program" on page 336:



Example Program

```
proc template;
  define statgraph scatterplotmatrix;
    begingraph;
      entrytitle "Scatter Plot Matrix";
      layout gridded;
        scatterplotmatrix
          sepallength sepalwidth petallength petalwidth /
          group=species name="matrix";
        discretelegend "matrix";
      endlayout;
    endgraph;
  end;
run;

proc sgrender data=sashelp.iris template=scatterplotmatrix;
run;
```

Statement Summary

By default, the SCATTERPLOTMATRIX statement produces a symmetric scatter plot matrix. For *n* columns, it produces an *n* columns by *n* rows matrix of scatter plots. By default, the columns of the matrix are in the same left-to-right order as the order of the *numeric-column-list*. The rows of the matrix are in the same bottom-to-top order as the *numeric-column-list*. You can reverse the direction of the diagonal by setting START=TOPLEFT.

To produce a rectangular matrix of scatter plots, use the ROWVARS= option. Specifying *n* columns in the SCATTERPLOTMATRIX statement and *m* columns on the ROWVARS= option produces an n-columns by m-rows matrix of scatter plots. For example, the following statement specifies 2 columns on SCATTERPLOTMATRIX and 3 columns on the ROWVARS= option to produce the 2-columns by 3-rows matrix:

```
SCATTERPLOTMATRIX Height Weight
   / ROWVARS=(Age Height Weight);
```

The SCATTERPLOTMATRIX statement cannot appear within an overlay-type layout. It generates its own matrix of plots and is typically placed in a LAYOUT GRIDDED block.

If there are missing values in a column or a row, all of the points that can be plotted are plotted in each scatter plot.

Required Arguments

numeric-column-list

specifies a list of numeric columns to plot. There must be at least two columns to produce a useful matrix.

The default width is 640px, and the default height is 480px. The graph size is not automatically adjusted to accommodate a large number of columns.

To change the graph size for the current template, use the DESIGNHEIGHT= and DESIGNWIDTH= options in the BEGINGRAPH statement. To change the graph size for all templates in the current SAS session, use the HEIGHT= and WIDTH= options in the ODS GRAPHICS statement. Size settings in the ODS GRAPHICS statement override size settings in the BEGINGRAPH statement.

You can also limit the number of columns in the matrix (perhaps to seven in each dimension, for example) so that the resulting graphs are not too small to be useful.

Options

Statement Option	Description
COLORMODEL	Specifies a style element to be used with the MARKERCOLORGRADIENT= option.
CORROPTS	Specifies options for computing measures of association between pairs of variables.
DATALABEL	Specifies a column for marker labels.
DATALABELATTRS	Specifies the color and font attributes of the data labels.
DATATRANSPARENCY	Specifies the degree of the transparency of the markers.
DIAGONAL	Specifies whether the diagonal cells of the matrix are labeled with the labels (names) of the required arguments, or with a graph. The graph for each diagonal cell consists of an overlay combination of a histogram, normal, or kernel curves.

Statement Option	Description
ELLIPSE	Specifies that a confidence ellipse be included in each cell containing a scatter plot.
FREQ	Specifies a column that indicates a frequency count for each observation of the input data object.
GROUP	Creates a distinct set of scatter markers, error bars, and data labels for each unique group value of the specified column.
INDEX	Specifies indices for mapping marker attributes (color and symbol) to one of the GraphData1 - GranphDataN style elements.
INSET	Specifies what information is displayed in an inset.
INSETOPTS	Specifies the location and appearance options for the inset information.
MARKERATTRS	Specifies the attributes of the data markers.
MARKERCHARACTER	Specifies a column that defines strings to be used instead of marker symbols.
MARKERCHARACTERATTRS	Specifies the color and font attributes of the marker character specified on the MARKERCHARACTER= option.
MARKERCOLORGRADIENT	Specifies the column that is used to map marker colors to a continuous gradient.
NAME	Assigns a name to a plot statement for reference in other template statements.
REVERSECOLORMODEL	Specifies whether to reverse a gradient defined by the COLORMODEL= option.
ROLENAME	Specifies roles for information defined by data columns.
ROWVARS	Specifies a secondary list of columns to be paired with the required column list that is specified by the SCATTERPLOTMATRIX statement.
START	Specifies whether to start populating the rows of the matrix from the top left or the bottom left corner.
TIP	Specifies the information to display when the cursor is positioned over the scatter points.

Statement Option	Description
TIPFORMAT	Specifies display formats for information defined by roles.
TIPLABEL	Specifies display labels for information defined by roles.
WALLCOLOR	Specifies the fill color of the plot wall area.
WALLDISPLAY	Specifies whether the plot’s wall and wall outline are displayed.

COLORMODEL=*style-element*

specifies a style element to be used with the MARKERCOLORGRADIENT= option.

Default: The ThreeColorAltRamp style element.

style-element

Name of a style element. The style element should contain these style attributes:

STARTCOLOR	color for the smallest data value of the column that is specified on the MARKERCOLORGRADIENT= option
NEUTRALCOLOR	color for the midpoint of the range of the column that is specified on the MARKERCOLORGRADIENT= option
ENDCOLOR	color for the highest data value of the column that is specified on the MARKERCOLORGRADIENT= option

Interaction: For this option to take effect, the MARKERCOLORGRADIENT= option must also be specified.

Interaction: The REVERSECOLORMODEL= option can be used to reverse the start and end colors of the ramp assigned to the color model.

CORROPTS=(*correlation-options*)

specifies options for computing measures of association between pairs of variables.

The following *correlation-options* are available:

EXCLNPWGT = FALSE | TRUE

specifies whether observations with non-positive weight values are excluded (TRUE) from the analysis.

Default: FALSE (observations with negative weights are treated like those with zero weights and counted in the total number of observations).

NOMISS = FALSE | TRUE

specifies whether observations with missing values are excluded (TRUE) from the analysis.

Default: FALSE (correlation statistics are computed using all of the nonmissing pairs of variables).

Using NOMISS=TRUE is computationally more efficient.

WEIGHT = *numeric-column*

specifies a weighting variable to use in the calculation of Pearson weighted product-moment correlation. The observations with missing weights are excluded from the analysis.

Default: For observations with non-positive weights, the weights are set to zero and the observations are included in the analysis.

You can include EXCLNPWGT among the *correlation-options* to exclude observations with negative or zero weights from the analysis. If you use this WEIGHT *correlation-option*, consider which value of the VARDEF= *correlation-option* is appropriate.

VARDEF=DF | N | WDF | WEIGHT

specifies the variance divisor in the calculation of variances and covariances.

Default: DF

DF	Degrees of Freedom (N - 1)
N	number of observations
WDF	sum of weights minus 1 (WEIGHT - 1)
WEIGHT	sum of weights

Interaction: This option has effect only when the INSET= option is also used.

See PROC CORR in the *Base SAS Procedures Guide: Statistical Procedures* for statistical and computational details of these options.

DATALABEL=*column*

specifies a column for marker labels. The label positions are adjusted to prevent the labels from overlapping.

Default: no default

Interaction: If a numeric *column* is specified and the column has no format, a BEST6. format is applied.

Interaction: This option is ignored if the MARKERCHARACTER= option is used.

DATALABELATTRS=*style-element* | *style-element* (*text-options*) | (*text-options*)

specifies the color and font attributes of the data labels. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Text Options" on page 566 for available *text-options*.

Default:

- For non-grouped data, the GraphDataText style element.
- For grouped data, the GraphData1:ContrastColor - GraphDataN:ContrastColor style references.

Interaction: For this option to take effect, the DATALABEL= option must also be specified.

Interaction: This option is ignored if the MARKERCHARACTER= option is specified.

DATATRANSPARENCY=*number*

specifies the degree of the transparency of the markers.

Default: 0

Range: 0 (opaque) to 1 (entirely transparent)

DIAGONAL= LABEL | (*graph-list*)

specifies whether the diagonal cells of the matrix are labeled with the labels (names) of the required arguments, or with a graph. The graph for each diagonal cell consists of an overlay combination of a histogram, normal, or kernel curves.

Default: Variable labels (or names) are displayed in the diagonal cells.

The *graph-list* can specify one or more of the following:

HISTOGRAM

specifies a histogram

NORMAL

specifies a normal density curve

KERNEL

specifies a kernel density estimate.

Requirement: When specifying multiple graphs in the *graph-list*, you must separate the values with a space. For example, the following specification requests both a histogram and a normal density curve in each diagonal cell:

```
DIAGONAL=(HISTOGRAM NORMAL)
```

Interaction: The computation for HISTOGRAM, NORMAL, and KERNEL is always computed on all the data for the current variable (including the FREQ= variable, if used). The GROUP= option is not considered in any of these computations.

Interaction: This option is ignored if the ROWVARS= option is used.

When this option is specified, the labels are drawn around the outside of the matrix, and the matrix axes are dropped.

ELLIPSE=(<*ellipse-suboptions*>)

specifies that a confidence ellipse be included in each cell containing a scatter plot. The ellipse is always drawn behind the scatter points.

The *ellipse-suboptions* include the following:

TYPE=MEAN | PREDICTED

specifies the type of ellipse.

Default: MEAN

See also: For statistical details about how the ellipse is calculated, see "ELLIPSE Statement" on page 229.

MEAN	specifies a confidence ellipse of the mean
PREDICTED	specifies a prediction ellipse of the data

ALPHA=*positive-number*

specifies the confidence level to compute for each ellipse.

Default: .05

Range: 0 < *number* < 1

ALPHA=.05 represents a 95% confidence level.

Default: TYPE=MEAN ALPHA=.05 You can set defaults by specifying the option without arguments: ELLIPSE=( ).

Interaction: The ellipse might be clipped by the data range for the scatter points.

Interaction: The ellipse is always computed on all the data for the current pair of X and Y variables (including the FREQ= variable, if used). The GROUP= option is not considered when computed the ellipse.

The display properties of each ellipse are controlled by the style elements:

- The GraphDataDefault element controls the outline and fill properties.
- The GraphEllipse element controls whether the outline, fill, or both are shown.

FREQ= *numeric-column* | *expression*

specifies a column that indicates a frequency count for each observation of the input data object. If *n* is the value of the FREQ variable for a given observation, then that observation is plotted *n* times.

Default: Each observation is plotted once.

Restriction: If the value of the *numeric-column* is missing or is less than 1, the observation is not used in the analysis. If the value is not an integer, only the integer portion is used.

GROUP=*column* | *expression*

creates a distinct set of scatter markers, error bars, and data labels for each unique group value of the specified column.

Default: Each distinct group value might be represented in the graph by a different combination of color and marker symbol. Markers vary according to the ContrastColor and MarkerSymbol attributes of the GraphData1 - GraphDataN style elements.

Interaction: The group values are mapped in the order of the data, unless the INDEX= option is used to alter the default sequence of markers and colors.

Interaction: The marker size is set by the MARKERATTRS= option.

Interaction: If the MARKERCHARACTER= and MARKERCOLORGRADIENT= options are used, their settings override the group settings for marker symbol and marker color.

Interaction: The representations that are used to identify the groups can be overridden. For example, each distinct group value is represented by a different marker symbol, but the MARKERATTRS=(SYMBOL=*marker*) option could be used to assign the same symbol to all of the plot's marker symbols, letting marker color indicate group values. Likewise, MARKERATTRS=(COLOR=*color*) could be used to assign the same color to all markers, letting marker symbol indicate group values.

INDEX=*positive-integer-column* | *expression*

specifies indices for mapping marker attributes (color and symbol) to one of the GraphData1 - GranphDataN style elements.

Default: no default

Interaction: For this option to take effect, the GROUP= option must also be specified.

Interaction: All of the indexes for a specific group value must be the same. Otherwise, the results are unpredictable.

Interaction: If this option is not used, then the group values are mapped in the order of the data.

Interaction: If the MARKERCHARACTER= and MARKERCOLORGRADIENT= options are used, their settings override the group settings for marker symbol and marker color.

Interaction: The index values are 1-based indices. For the style elements GraphData1 - GraphDataN, if the index value is greater than N, then a modulo operation remaps that index value to a number less than N to determine which style element to use.

Discussion: Indexing can be used to collapse the number of groups that are represented in a graph. For more information, see "Remapping Groups for Grouped Data" on page 121.

INSET= (*info-options*)
specifies what information is displayed in an inset. Insets appear in all cells of the matrix except the diagonal and are displayed as a small table of *name-value* pairs.

Default: no default

The following *info-options* are available:

NOBS
Total number of observations where both the X and Y variables have nonmissing values. If the FREQ= option is used, this number is adjusted accordingly. The value of NOBS can be further adjusted by the use of the NOMISS=, WEIGHT=, and EXCLNPWGT= suboptions of the CORROPTS= option.

PEARSON
The Pearson product-moment correlation. The computation of the correlation is affected by the FREQ= and CORROPTS= options. The computation is not done on a per group value when GROUP= is used.

PEARSONPVAL
The probability value for the Pearson product-moment correlation.

The location and appearance of the inset is controlled by the INSETOPTS= option.

Discussion: A typical inset looks like this:

```
N      150
r 0.96287
p(r)  <.0001
```

In this example,

NOBS
is represented by N

PEARSON
is represented by r

PEARSONPVAL
is represented by p(r)

See PROC CORR in the *Base SAS Procedures Guide: Statistical Procedures* for statistical and computational details.

INSETOPTS = (*appearance-options*)
specifies location and appearance options for the inset information.

The *appearance-options* can be any one or more of the settings that follow. The options must be enclosed in parentheses, and each option is specified as a *name* = *value* pair.

AUTOALIGN=NONE | AUTO | (*location-list*)
specifies whether the inset is automatically aligned within the layout.

Default: NONE

NONE — Do not automatically align the inset. The inset's position is therefore set by the

HALIGN= and VALIGN= *appearance-options*.

AUTO
Attempt to center this inset in the area that is farthest from any surrounding markers. Data cells might have different inset placements.

(*location-list*)
Restrict this inset's possible locations to those locations in the specified *location-list*, and use the *location-list* position that least collides with the data cell's other graphics features. The *location-list* is blank-separated and can contain any of these locations: TOPLEFT TOP TOPRIGHT LEFT CENTER RIGHT BOTTOMLEFT BOTTOM BOTTOMRIGHT. Example: AUTOALIGN = (TOPRIGHT TOPLEFT)

Interaction: When AUTOALIGN=AUTO or (*location-list*), the enclosing layout statement's HALIGN= and VALIGN= *appearance-options* are ignored.

BACKGROUNDCOLOR= *style-reference* | *color*
specifies the color of the inset background

Default: GraphWalls:Color style reference

style-reference
A reference of the form *style-element : style-attribute*. Only the style-attribute named COLOR is used.

BORDER= *boolean*
specifies whether a border is displayed around the inset.

Default: FALSE

HALIGN=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the inset.

Default: LEFT

Interaction: This option is ignored unless AUTOALIGN=NONE.

OPAQUE= *boolean*
specifies whether the inset background is opaque (TRUE) or transparent (FALSE).

Default: FALSE

Interaction: When this option is set to FALSE, the background color is not used.

TEXTATTRS=*style-element* | *style-element* (*text-options*) | (*text-options*)
specifies the text properties of the entire inset. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Text Options" on page 566 for available *text-options*.

Default: The GraphDataText style element.

VALIGN=TOP | CENTER |BOTTOM
specifies the vertical alignment of the inset.

Default: TOP

Interaction: This option is ignored unless AUTOALIGN=NONE.

MARKERATTRS=*style-element* | *style-element* (*marker-options*) | (*marker-options*)

specifies the attributes of the data markers. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Marker Options" on page 565 for available *marker-options*.

Default:

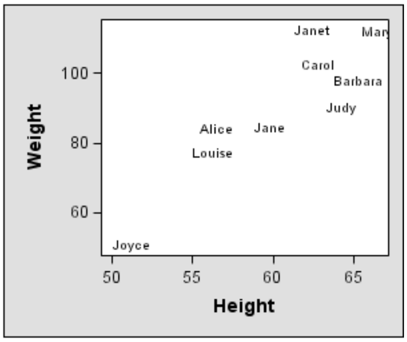
- For non-grouped data, the GraphDataDefault style element.
- For grouped data, the MarkerSymbol and ContrastColor attributes of the GraphData1 - GraphDataN style elements, and the GraphDataDefault:MarkerSize style reference.

Interaction: If the MARKERCOLORGRADIENT= option is specified, this option's COLOR= setting is ignored.

Interaction: If the MARKERCHARACTER= option is specified, its SYMBOL= and WEIGHT= settings are ignored.

MARKERCHARACTER=*column* | *expression*

specifies a column that defines strings to be used instead of marker symbols.



Default: no default

Interaction: This option overrides the DATALABEL= option.

Interaction: If the GROUP= option is also used, color is displayed for a DISCRETE legend, but the character is not displayed in the legend.

If the GROUP= option is also specified, the same colors are applied to the text strings as would have been applied to markers.

If a numeric column is used, its values are converted to strings using the format associated with the column or BEST6. if no format is defined.

Each string is centered horizontally and vertically at the data point. The data point positions are not adjusted to prevent text overlap.

MARKERCHARACTERATTRS=*style-element* | *style-element* (*text-options*) | (*text-options*)

specifies the color and font attributes of the marker characters. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Text Options" on page 566 for available *text-options*.

Default:

- For non-grouped data, the GraphDataText style element.
- For grouped data, GraphData1:ContrastColor - GraphDataN:ContrastColor style references.

Interaction: For this option to take effect, the MARKERCHARACTER= option must also be used.

When the GROUP= option is also specified, each distinct group value might be represented by a different color (depending on the ODS style setting or the setting on the INDEX= option). The marker character that is associated with the group is assigned the group color. This option's COLOR= suboption can be used to specify a single color for all marker characters in a graph, without affecting items that have the group color, such as error bars and marker symbols.

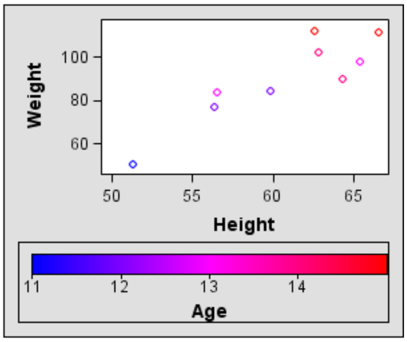
MARKERCOLORGRADIENT=*numeric-column* | *expression*
specifies the column that is used to map marker colors to a continuous gradient.

Default: no default

Restriction: To display a legend with this option in effect, you must use a CONTINUOUSLEGEND statement, not a DISCRETELEGEND statement.

Interaction: This option overrides the COLOR= setting of the MARKERATTRS= or MARKERCHARACTERATTRS= option.

This option can be used to add a second response variable to an analysis. For example, in an analysis of weight by height, an age column might be specified by the MARKERCOLORGRADIENT= option so that the change in the gradient color of the markers reflects the change in age.



By default, the color range is determined by the ThreeColorAltRamp style element. The COLORMODEL= option allows a different color range to be used.

NAME=*"string"*
assigns a name to a plot statement for reference in other template statements.

Default: no default

Restriction: The *string* is case sensitive, cannot contain spaces, and must define a unique name within the template.

The specified name is used primarily in legend statements to coordinate the use of colors and marker symbols between the graph and the legend.

REVERSECOLORMODEL=*boolean*
specifies whether to reverse a gradient (color ramp) defined by the COLORMODEL= option.

Default: FALSE

ROLENAME=(*role-name-list*)
specifies user-defined roles for information contained in data columns.

Default: no default

(*role-name-list*)
a blank-separated list of *rolename* = *column* pairs.

The following example assigns column ID to the user-defined role TIP1, and columns AGE, HEIGHT, WEIGHT to the user-defined roles TIP2, TIP3, and TIP4.

```
ROLENAME=(TIP1=ID TIP2=AGE TIP3=HEIGHT TIP4=WEIGHT)
```

Requirement: The role names that you choose must be unique and different from the pre-defined roles X, Y, DATALABEL, MARKERCHARACTER, MARKERCOLORGRADIENT, GROUP, and INDEX.

Interaction: For this option to take effect, the TIP= option must also be used.

This option provides a way to add to the data columns that appear in tooltips specified by the TIP= option.

ROWVARS = (*column-list*)
specifies a secondary list of columns to be paired with the required column list that is specified by the SCATTERPLOTMATRIX statement.

Default: no default

Interaction: When this option is specified, the DIAGONAL= option is ignored.

The labels for the variables appear vertically on the left side of the matrix.

START=TOPLEFT | BOTTOMLEFT
specifies whether to start populating the matrix from the top left or bottom left corner.

Default: TOPLEFT

TIP=(*role-list*)
specifies the information to display when the cursor is positioned over the scatter points. If this option is used, it replaces all the information displayed by default. Roles for columns that do not contribute to the scatter plot can be specified along with roles that do.

Default: The columns assigned to these roles are automatically included in the tooltip information: current X, current Y, DATALABEL, MARKERCHARACTER, MARKERCOLORGRADIENT, and GROUP.

(*role-list*)
an ordered, blank-separated list of unique SCATTERPLOTMATRIX and user-defined roles. SCATTERPLOT roles include: X, Y, GROUP, DATALABEL, MARKERCHARACTER, and MARKERCOLORGRADIENT.

User-defined roles are defined with the ROLENAME= option.

The following example displays tooltips for the columns assigned to the roles TIP1, TIP2, TIP3, and TIP4.

```
ROLENAME=(TIP1=ID TIP2=AGE TIP3=HEIGHT TIP4=WEIGHT)
TIP= (TIP1 TIP2 TIP3 TIP4)
```

Requirement: To generate tooltips, you must include an ODS GRAPHICS ON statement that has the IMAGEMAP option specified, and write the graphs to the ODS HTML destination.

Interaction: The labels and formats for the TIP variables can be controlled with the TIPLABEL= and TIPFORMAT= options.

TIPFORMAT=(*role-format-list*)
specifies display formats for tip columns.

Default: The column format of the variable assigned to the role or BEST6. if no format is assigned to a numeric column.

(*role-format-list*)
a list of *rolename* = *format* pairs separated by blanks.

```
ROLENAME=(TIP1=ID TIP2=AGE TIP3=HEIGHT TIP4=WEIGHT)
TIP=(TIP1 TIP2 TIP3 TIP4)
TIPFORMAT=(TIP3= 4.1)
```

Requirement: Columns must be assigned to the roles for this option to have any effect. See the ROLENAME= option.

This option provides a way to control the formats of columns that appear in tooltips. Only the roles that appear in the TIP= option are used.

TIPLABEL=(*role-label-list*)
specifies display labels for tip columns.

Default: The column label or column name of the variable assigned to the role.

(*role-label-list*)
a list of *rolename = "string"* pairs separated by blanks.

```
ROLENAME=(TIP1=ID TIP2=AGE TIP3=HEIGHT TIP4=WEIGHT)
TIP=(TIP1 TIP2 TIP3 TIP4)
TIPLABEL=(TIP3="Height in Inches"
TIP4="Weight in Pounds")
```

Requirement: Columns must be assigned to the roles for this option to have any effect. See the ROLENAME= option.

This option provides a way to control the labels of columns that appear in tooltips. Only the roles that appear in the TIP= option are used.

WALLCOLOR=*style-reference* | *color*
specifies the fill color of the plot wall area.

Default: The GraphWalls:Color style reference.

style-reference
A reference in the form *style-element:style-attribute*. Only the style- attribute named COLOR is used.

Interaction: This option is ignored if WALLDISPLAY=NONE or WALLDISPLAY=(OUTLINE).

WALLDISPLAY=STANDARD | ALL | NONE | *(display-options)*
specifies whether the plot's wall and wall outline are displayed.

Default: STANDARD

STANDARD
Displays a filled wall. The setting of the FRAMEBORDER= ON | OFF attribute of the GraphWalls style element determines whether the wall outline is displayed.

ALL
Displays a filled, outlined wall.

NONE
Displays no wall, no wall outline.

(*display-options*)
These options must include one of the following:

OUTLINE	displays the wall outline.
FILL	displays a filled wall area.

Use the WALLCOLOR= option to control the fill color of the wall.

The appearance attributes of the wall outline are set by the GraphAxisLine style element.

Chapter 37
SERIESPLOT Statement

Description and Syntax

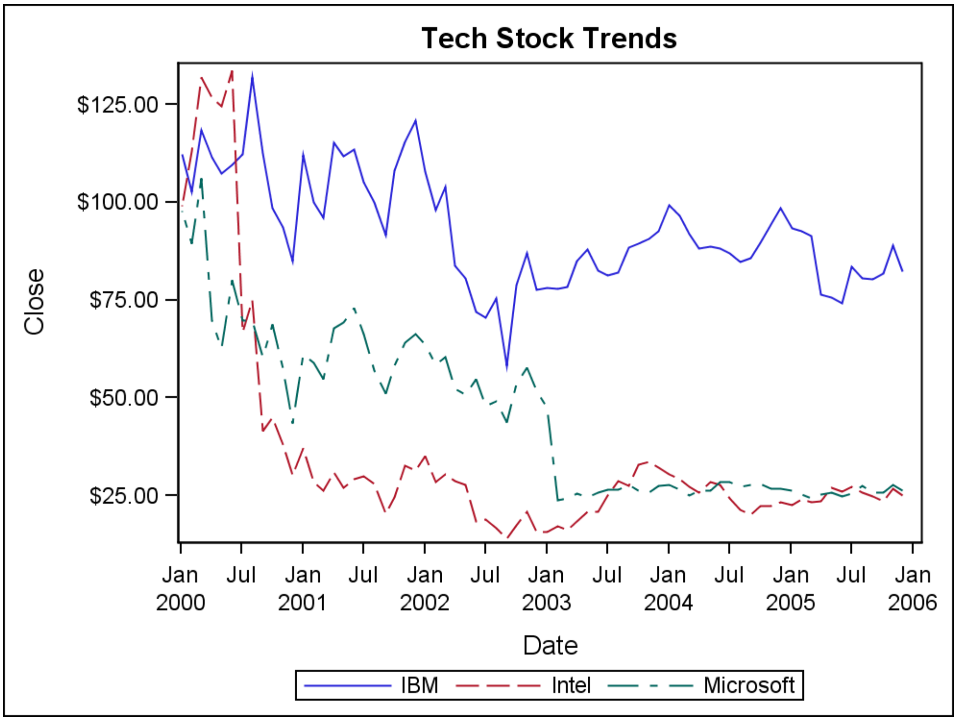
Displays a series of line segments that connect observations of input data.

SERIESPLOT X = *column* | *expression*
Y = *column* | *expression* <*/option(s)*>;

Example Program and Statement Details

Example Graph

The following graph was generated by the “Example Program” on page 352:



Example Program

```
proc template;
 define statgraph seriesplot;
  begingraph;
   entrytitle "Tech Stock Trends";
   layout overlay;
     seriesplot x=date y=close / group=stock name="stocks";
     discretelegend "stocks";
   endlayout;
  endgraph;
 end;
run;

proc sgrender data=sashelp.stocks template=seriesplot;
  where date > "31dec1999"d;
run;
```

Statement Summary

The SERIESPLOT statement is typically used to show time-dependent data.

Required Arguments

X=*column* | *expression*
specifies the column for the X values.

Y=*column* | *expression*
specifies the column for the Y values.

Options

Statement Option	Description
BREAK	Breaks the plot line at the occurrences of the missing values of the X or Y variable.
CONNECTORDER	Specifies how to connect the data points to form the series line.
CURVELABEL	Specifies a label for the series line.
CURVELABELATTRS	Specifies the color and font attributes of the series line label.
CURVELABELLOCATION	Specifies the location of the series line label relative to the plot area.
CURVELABELPOSITION	Specifies the position of the series label relative to the series line.
DATALABEL	Specifies a column that supplies values for the data point labels.
DATALABELATTRS	Specifies the color and font attributes of the data labels.
DATATRANSPARENCY	Specifies the degree of the transparency of the line and markers, when displayed.
DISCRETEOFFSET	Specifies an amount to offset all series lines and markers from discrete X values, or discrete Y values, or both.
DISPLAY	Specifies additional feature to display with the series line.
GROUP	Creates a separate series plot for each unique group value of the specified column.
INDEX	Specifies indices for mapping series-line attributes (color, marker symbol, and line pattern) to one of the GraphData1 - GranphDataN style elements.
LEGENDLABEL	Specifies a label to be used in a legend.
LINEATTRS	Specifies the line attributes of the series line.
MARKERATTRS	Specifies the attributes of the data markers.
NAME	Assigns a name to a plot statement for reference in other template statements.

Statement Option	Description
PRIMARY	Specifies that the data columns for this plot be used for determining default axis features.
ROLENAME	Specifies roles for information defined by data columns.
TIP	Specifies the information to display when the cursor is positioned over the series line.
TIPFORMAT	Specifies display formats for information defined by roles.
TIPLABEL	Specifies display labels for information defined by roles.
URL	Specifies an HTML page to display when a series line segment is selected.
XAXIS	Specifies whether data are mapped to the primary X (bottom) axis or the secondary X2 (top) axis.
YAXIS	Specifies whether data are mapped to the primary Y (left) axis or the secondary Y2 (right) axis.

BREAK=*boolean*
breaks the plot line at missing values of the X or Y variable.

Default: FALSE

When this option is set to FALSE, missing values are skipped and a continuous line is drawn.

CONNECTORDER=XVALUES | XAXIS
specifies how to connect the data points to form the series line.

Default: XVALUES

XVALUES
Connects data points in the order read from the X variable.

XAXIS
Connects data points as they occur min-to-max along the X axis.

For certain types of series lines (for example, time series) when the input data might not be sorted by the X variable, set this option to XAXIS to assure the expected connect order.

CURVELABEL=*"string"* | *column* | *expression*
specifies a label for the series line.

Default: no default

Interaction:

- For non-grouped data, use *"string"*.
- For grouped data, use a column to define the labels for each group value.

The font and color attributes for the label are specified by the CURVELABELATTRS= option.

CURVELABELATTRS=*style-element* | *style-element* (*text-options*) | (*text-options*)

specifies the color and font attributes of the series line labels. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Text Options" on page 566 for available *text-options*.

Default:

- For non-grouped data, the GraphValueText style element.
- For grouped data, text color is derived from the GraphData1:ContrastColor - GraphDataN:ContrastColor style references. The font is derived from the GraphValueText style element.

Interaction: For this option to take effect, the CURVELABEL=curvelabel option must also be used.

Interaction: This option's COLOR= setting overrides the colors indicated by the GROUP= option.

When the GROUP= option is used, each distinct group value might be represented by a different color. The series label that is associated with the group is assigned the group color. This option can be used to specify a single color for all series labels in a graph, without affecting items that have the group color, such as lines and marker symbols.

CURVELABELLOCATION=INSIDE | OUTSIDE

specifies the location of the series line label relative to the plot area.

Default: INSIDE

INSIDE

inside the plot area

OUTSIDE

outside the plot area

Restriction: OUTSIDE cannot be used when the SERIESPLOT is used in multicell layouts such as LATTICE, DATAPANEL, or DATALATTICE, where axes might be external to the grid.

Interaction: For this option to take effect, the CURVELABEL= option must also be specified.

Interaction: This option is used in conjunction with the CURVELABELPOSITION= option to determine where the curve labels appear. For more information, see "Location and Position of Curve Labels" on page 123.

CURVELABELPOSITION=AUTO | MAX | MIN | START | END

specifies the position of the series label relative to the series line.

Default: AUTO when CURVELABELLOCATION=OUTSIDE END when CURVELABELLOCATION=INSIDE

AUTO

Only used when CURVELABELLOCATION=OUTSIDE. The series label is positioned automatically near the end series line along unused axes whenever possible (typically Y2 or X2) to avoid collision with tick values.

MAX

Forces the series label to appear near maximum series values (typically, to the right)

MIN

Forces the series label to appear near minimum series values (typically, to the left)

START

Only used when CURVELABELLOCATION=INSIDE. Forces the series label to appear near the beginning of the curve. Particularly useful when the curve line has a spiral shape.

END

Only used when CURVELABELLOCATION=INSIDE. Forces the series label to appear near the end of the curve. Particularly useful when the curve line has a spiral shape.

Restriction: The AUTO setting is not supported if CURVELABELLOCATION=INSIDE is specified. The START and END settings are not supported if CURVELABELLOCATION=OUTSIDE is specified.

Interaction: For this option to take effect, the CURVELABEL= option must also be specified.

Interaction: This option is used in conjunction with the CURVELABELLOCATION= option to determine where the series label appears. For more information, see "Location and Position of Curve Labels" on page 123.

DATALABEL=*column* | *expression*

specifies a column that supplies values for the data point labels.

Default: no default

The label positions are adjusted to prevent the labels from overlapping.

DATALABELATTRS=*style-element* | *style-element* (*text-options*) | (*text-options*)

specifies the color and font attributes of the data labels. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Text Options" on page 566 for available *text-options*.

Default:

- For non-grouped data, the GraphDataText style element.
- For grouped data, the GraphData1:ContrastColor - GraphDataN:ContrastColor style references.

Interaction: For this option to take effect, the DATALABEL= option must also be specified.

Interaction: This option's COLOR= setting overrides the colors indicated by the GROUP= option.

When the GROUP= option is used, each distinct group value might be represented by a different color. The data label that is associated with the group is assigned the group color. This option can be used to specify a single color for all data labels in a graph, without affecting items that have the group color, such as error bars and marker symbols.

DATATRANSPARENCY=*number*

specifies the degree of the transparency of the line and markers, when displayed.

Default: 0

Range: 0 (opaque) to 1 (entirely transparent)

DISCRETEOFFSET=*number*

specifies an amount to offset all series lines and markers from X values for a discrete X axis, or from Y values for a discrete Y axis. If both axes are discrete, the offset applies to both axes.

Note: This feature is for the third maintenance release of SAS 9.2 and later.

Default: 0 (no offset, all series lines and markers are centered on the X values for a discrete X axis, or on the Y values for a discrete Y axis)

Range: -0.5 to +0.5, where .5 represents half the distance between discrete ticks. A positive offset is to the right on discrete X values and up on discrete Y values. If the layout's axis options set REVERSE=TRUE, then the offset direction is also reversed.

DISPLAY=STANDARD | ALL | *display-options*

specifies additional feature to display with the series line.

Default: STANDARD

STANDARD

displays a series line without markers.

ALL

displays a series line with markers.

(display-options)

a list of options enclosed in parentheses. Currently, only the following option is available for the list:

MARKERS—displays a series line with markers

Use the MARKERATTRS= and LINEATTRS= options to control the appearance of the line and markers.

GROUP=*column* | *expression*

creates a separate series plot for each unique group value of the specified column.

Default: Each distinct group value might be represented in the graph by a different combination of color, line pattern, and marker symbol. Lines and markers vary according to the ContrastColor, LineStyle, and MarkerSymbol attributes of the GraphData1 - GraphDataN style elements. Line thickness (for grouped and ungrouped data) is controlled by the LINEATTRS= option.

Interaction: The group values are mapped in the order of the data, unless the INDEX= option is used to alter the default sequence of marker symbols, colors, and line patterns.

Interaction: The marker size is set by the MARKERATTRS= option.

The representations that are used to identify the groups can be overridden. For example, each distinct group value is often represented by a different line pattern, but the LINEATTRS= (PATTERN=*pattern*) option could be used to assign the same line pattern to all of the plot's line patterns, letting line color indicate group values. Likewise, LINEATTRS= (COLOR= color) could be used to assign the same color to all lines, letting line pattern indicate group values.

INDEX=*positive-integer-column* | *expression*

specifies indices for mapping series-line attributes (color, marker symbol, and line pattern) to one of the GraphData1 - GranphDataN style elements.

Default: no default

Interaction: For this option to take effect, the GROUP= option must also be specified.

Interaction: All of the indexes for a specific group value must be the same. Otherwise, the results are unpredictable.

Interaction: If this option is not used, then the group values are mapped in the order of the data.

Interaction: The index values are 1-based indices. For the style elements in GraphData1 - GraphDataN, if the index value is greater than N, then a modulo operation remaps that index value to a number less than N to determine which style element to use.

Discussion: Indexing can be used to collapse the number of groups that are represented in a graph. For more information, see "Remapping Groups for Grouped Data" on page 121.

LEGENDLABEL= *"string"*
specifies a label to be used in a legend.

Default: The Y-variable label. If a label is not defined, the Y-variable name.

Interaction: If the GROUP= option is specified, this option is ignored.

LINEATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)
specifies the attributes of the series line. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Line Options" on page 564 for available *line-options*.

Default:

- For non-grouped data, the GraphDataDefault style element.
- For grouped data, the ContrastColor and LineStyle attributes of the GraphData1 - GraphDataN style elements, and the GraphDataDefault:LineThickness style reference.

MARKERATTRS=*style-element* | *style-element* (*marker-options*) | (*marker-options*)
specifies the attributes of the data markers. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Marker Options" on page 565 for available *marker-options*.

Default:

- For non-grouped data, GraphDataDefault style element.
- For grouped data, the MarkerSymbol and ContrastColor attributes of the GraphData1 - GraphDataN style elements, and the GraphDataDefault:MarkerSize style reference.

Interaction: DISPLAY=ALL or DISPLAY=(MARKERS) must be set for this option to have any effect.

Interaction: This option's COLOR= suboption overrides the default behavior for grouped data: all markers the same color and the marker symbol alone distinguishes the markers.

Interaction: This option's SYMBOL= suboption overrides the default behavior for grouped data: it holds the marker constant across group values and marker color alone distinguishes the markers.

NAME=*"string"*
assigns a name to a plot statement for reference in other template statements.

Default: no default

Restriction: The *string* is case sensitive, cannot contain spaces, and must define a unique name within the template.

The specified name is used primarily in legend statements to coordinate the use of colors, marker symbols, and line patterns between the graph and the legend.

PRIMARY=*boolean*
specifies that the data columns for this plot and the plot type be used for determining default axis features.

Default: FALSE

Restriction: This option is ignored if the plot is placed under a GRIDDED or LATTICE layout block.

Details: This option is needed only when two or more plots within an overlay-type layout contribute to a common axis. For more information, see "When Plots Share Data and a Common Axis" on page 397

ROLENAME=(*role-name-list*)
specifies user-defined roles for information contained in data columns.

Default: no default

(*role-name-list*)
a blank-separated list of *rolename* = *column* pairs.

For example, ROLENAME= (TIP1=OBS) assigns the column OBS to the user-defined role TIP1.

Requirement: The role names that you choose must be unique and different from the pre-defined roles X, Y, CURVELABEL, DATALABEL, GROUP, and INDEX.

Interaction: For this option to take effect, the TIP= option must also be used.

This option provides a way to add to the data columns that appear in tooltips specified by the TIP= option.

TIP=(*role-list*)
specifies the information to display when the cursor is positioned over the series line. If this option is used, it replaces all the information displayed by default. Roles for columns that do not contribute to the series plot can be specified along with roles that do.

Default: The columns assigned to these roles are automatically included in the tooltip information: X, Y, DATALABEL, and GROUP.

(*role-list*)
an ordered, blank-separated list of unique SERIESPLOT and user-defined roles. SERIESPLOT roles include X, Y, CURVELABEL, DATALABEL, and GROUP.

Note: CURVELABEL is considered a role only when it is assigned a column of values. It is not considered a role and does not display tooltips when assigned a string.

User-defined roles are defined with the ROLENAME= option.

The following example displays tooltips for the columns assigned to the roles X and Y as well as the column OBS, which is not assigned to any pre-defined SeriesPlot role. The OBS column must first be assigned a role.

```
ROLENAME=(TIP1=OBS)
TIP= (TIP1 X Y)
```

Requirement: To generate tooltips, you must include an ODS GRAPHICS ON statement that has the IMAGEMAP option specified, and write the graphs to the ODS HTML destination.

Interaction: The labels and formats for the TIP variables can be controlled with the TIPLABEL= and TIPFORMAT= options.

TIPFORMAT=(*role-format-list*)
specifies display formats for tip columns.

Default: The column format of the variable assigned to the role or BEST6. if no format is assigned to a numeric column.

(*role-format-list*)
a list of *rolename* = *format* pairs separated by blanks.

```
ROLENAME=(TIP1=OBS)
TIP=(TIP1 X Y)
TIPFORMAT=(TIP1=4.)
```

Requirement: This option provides a way to control the formats of columns that appear in tooltips. Only the roles that appear in the TIP= option are used. Columns must be assigned to the roles for this option to have any effect. See the ROLENAME= option.

TIPLABEL=(*role-label-list*)
specifies display labels for tip columns.

Default: The column label or column name of the variable assigned to the role.

(*role-label-list*)
a list of *rolename* = *"string"* pairs separated by blanks.

```
ROLENAME=(TIP1=OBS)
TIP=(TIP1 X Y)
TIPLABEL=(TIP1="Observation #")
```

Requirement: This option provides a way to control the labels of columns that appear in tooltips. Only the roles that appear in the TIP= option are used. Columns must be assigned to the roles for this option to have any effect. See the ROLENAME= option.

URL=*character-column*
specifies an HTML page to display when a point or a segment of the curve is selected.

Default: no default

character-column
each value of the column must be a valid HTML page reference (HREF). For example, *http://www.sas.com/technologies/analytics/index.html*.

Requirement: To generate selectable bars, you must include an ODS GRAPHICS ON statement that has the IMAGEMAP option specified, and write the graphs to the ODS HTML destination.

XAXIS=X | X2
specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

Default: X

Interaction: The overall plot specification and the layout type determine the axis display for the specified axis. For more information, see "How Axis Features are Determined" on page 395.

YAXIS=Y | Y2
specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

Default: Y

Interaction: The overall plot specification and the layout type determine the axis display for the specified axis. For more information, see "How Axis Features are Determined" on page 395.

Chapter 38

STEPPLOT Statement

Description and Syntax

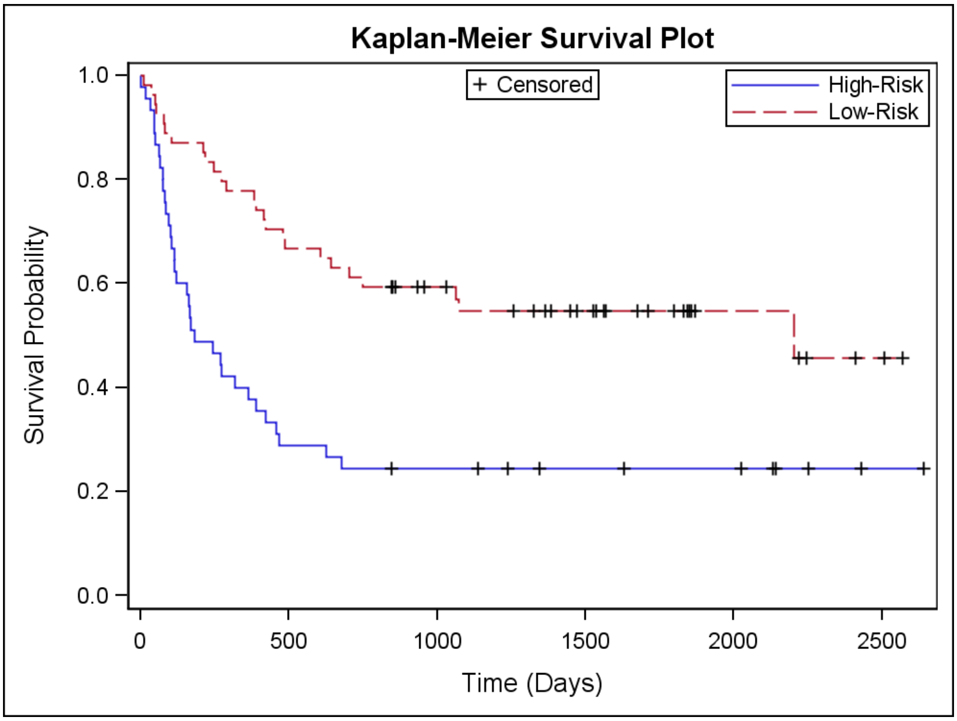
Displays a series of horizontal and vertical line segments that connect observations of input data.

STEPPLOT X = *column* | *expression*
Y = *numeric-column* | *expression* *</option(s)>*;

Example Program and Statement Details

Example Graph

The following graph was generated by the “Example Program” on page 362:



Example Program

```
proc template;
  define statgraph stepplot;
    begingraph;
      entrytitle "Kaplan-Meier Survival Plot";
      layout overlay /
        yaxisopts=(linearopts=(viewmin=0 viewmax=1));
        stepplot x=Time y=Survival /
          group=Stratum name="step";
        scatterplot x=Time y=Censored /  name="scat"
          legendlabel="Censored" markerattrs=(symbol=plus);
        discretelegend "step" / location=inside
          halign=right valign=top across=1;
        discretelegend "scat" /
          location=inside halign=center valign=top;
       endlayout;
    endgraph;
  end;
run;

data Study;
input Group : $10. Time Status @@;
label Time="Time (Days)";
datalines;

 ... [datalines shown below]

run;

ods graphics;
ods exclude all;
ods output survivalplot=plotdata;
proc lifetest data=Study plots=(survival);
   time Time * Status(0);
   survival;
   strata Group;
run;
ods select all;

proc sgrender data=plotdata template=stepplot;
run;
```

Here are the data lines for the example program:

```
Low-Risk  2569 0 Low-Risk  2506 0 Low-Risk  2409 0
Low-Risk  2218 0 Low-Risk  1857 0 Low-Risk  1829 0
Low-Risk  1562 0 Low-Risk  1470 0 Low-Risk  1363 0
Low-Risk  1030 0 Low-Risk   860 0 Low-Risk  1258 0
Low-Risk  2246 0 Low-Risk  1870 0 Low-Risk  1799 0
Low-Risk  1709 0 Low-Risk  1674 0 Low-Risk  1568 0
Low-Risk  1527 0 Low-Risk  1324 0 Low-Risk   957 0
Low-Risk   932 0 Low-Risk   847 0 Low-Risk   848 0
Low-Risk  1850 0 Low-Risk  1843 0 Low-Risk  1535 0
Low-Risk  1447 0 Low-Risk  1384 0 Low-Risk   414 1
Low-Risk  2204 1 Low-Risk  1063 1 Low-Risk   481 1
Low-Risk   105 1 Low-Risk   641 1 Low-Risk   390 1
Low-Risk   288 1 Low-Risk   421 1 Low-Risk    79 1
Low-Risk   748 1 Low-Risk   486 1 Low-Risk    48 1
Low-Risk   272 1 Low-Risk  1074 1 Low-Risk   381 1
Low-Risk    10 1 Low-Risk    53 1 Low-Risk    80 1
Low-Risk    35 1 Low-Risk   248 1 Low-Risk   704 1
Low-Risk   211 1 Low-Risk   219 1 Low-Risk   606 1
High-Risk 2640 0 High-Risk 2430 0 High-Risk 2252 0
High-Risk 2140 0 High-Risk 2133 0 High-Risk 1238 0
High-Risk 1631 0 High-Risk 2024 0 High-Risk 1345 0
High-Risk 1136 0 High-Risk  845 0 High-Risk  422 1
High-Risk  162 1 High-Risk   84 1 High-Risk  100 1
High-Risk    2 1 High-Risk   47 1 High-Risk  242 1
High-Risk  456 1 High-Risk  268 1 High-Risk  318 1
High-Risk   32 1 High-Risk  467 1 High-Risk   47 1
High-Risk  390 1 High-Risk  183 1 High-Risk  105 1
High-Risk  115 1 High-Risk  164 1 High-Risk   93 1
High-Risk  120 1 High-Risk   80 1 High-Risk  677 1
High-Risk   64 1 High-Risk  168 1 High-Risk   74 1
High-Risk   16 1 High-Risk  157 1 High-Risk  625 1
High-Risk   48 1 High-Risk  273 1 High-Risk   63 1
High-Risk   76 1 High-Risk  113 1 High-Risk  363 1
```

Statement Summary

For character columns, the X-axis is always of TYPE=DISCRETE. For numeric columns, both the X- and the Y-axis are of TYPE=LINEAR by default.

You can change the axis type for numeric axes with the XAXISOPTS= and YAXISOPTS= options of the containing overlay layout.

The STEPPLOT statement uses X values in data order, unless the CONNECTORDER= option is used to change the order.

Required Arguments

X=*column* | *expression*
specifies the column of the X values.

Y=*numeric-column* | *expression*
specifies the numeric column of the Y values.

Options

Statement Option	Description
BREAK	Breaks the plot line at the occurrences of the missing values of the Y variable.
CONNECTORDER	Specifies how to connect the data points to form the step line.
CURVELABEL	Specifies a label for the step line.
CURVELABELATTRS	Specifies the color and font attributes of the step line label.
CURVELABELLOCATION	Specifies the location of the step label relative to the plot area.
CURVELABELPOSITION	Specifies the position of the step line labels relative to the step line.
DATALABEL	Specifies a column that supplies values for the data point labels.
DATALABELATTRS	Specifies the color and font attributes of the data labels.
DATATRANSPARENCY	Specifies the degree of the transparency of the step lines, markers, and error bars, when displayed.
DISCRETEOFFSET	Specifies an amount to offset all step lines and markers from discrete X values.
DISPLAY	Specifies whether to display markers on the step line.
ERRORBARATTRS	Specifies the attributes of the error bars that are associated with the data points.
ERRORLOWER	Specifies the values of the lower endpoints on the Y error bars.

Statement Option	**Description**
ERRORUPPER	Specifies the values of the upper endpoints of the Y error bars.
GROUP	Creates a separate step plot for each unique group value of the specified column.
INDEX	Specifies indices for mapping step-line attributes (color, marker symbol, and line pattern) to one of the GraphData1 - GranphDataN style elements.
JOIN	Specifies whether the steps are connected.
JUSTIFY	Specifies the location of the data point relative to the step.
LEGENDLABEL	Specifies a label for use in a legend.
LINEATTRS	Specifies the line attributes of the step line.
MARKERATTRS	Specifies the attributes of the data markers.
NAME	Assigns a name to a plot statement for reference in other template statements.
PRIMARY	Specifies that the data columns for this plot be used for determining default axis features.
ROLENAME	Specifies roles for information defined by data columns.
TIP	Specifies the information to display when the cursor is positioned over the step line.
TIPFORMAT	Specifies display formats for information defined by roles.
TIPLABEL	Specifies display labels for information defined by roles.
URL	Specifies an HTML page to display when a series line segment is selected.
XAXIS	Specifies whether data are mapped to the primary X (bottom) axis or the secondary X2 (top) axis.
YAXIS	Specifies whether data are mapped to the primary Y (left) axis or the secondary Y2 (right) axis.

BREAK=*boolean*

determines whether the plot line should show breaks at occurrences of missing values of the Y variable.

Default: FALSE

When this option is set to FALSE, missing values are skipped and the line continues through the missing value and to the next point.

CONNECTORDER=XVALUES | XAXIS

specifies how to connect the data points to form the step line.

Default: XVALUES

XVALUES

Connects data points in the data order of the X variable.

XAXIS

Connects data points sorted by their X values.

When the input data for the step lines is not sorted by the X variable, set this option to XAXIS to assure the expected connect order.

CURVELABEL=*"string"* | *column* | *expression*

specifies a label for the step line.

Default: no default

Interaction:

- For non-grouped data, use *"string"*.
- For grouped data, use a column to define the labels for each group value.

The font and color attributes for the label are specified by the CURVELABELATTRS= option.

CURVELABELATTRS=*style-element* | *style-element* (*text-options*) | (*text-options*)

specifies the color and font attributes of the step line labels. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Text Options" on page 566 for available *text-options*.

Default:

- For non-grouped data, the GraphValueText style element.
- For grouped data, text color is derived from the GraphData1:ContrastColor - GraphDataN:ContrastColor style references. The font is derived from the GraphValueText style element.

Interaction: For this option to take effect, the CURVELABEL= option must also be used.

Interaction: This option's COLOR= setting overrides the colors indicated by the GROUP= option.

When the GROUP= option is used, each distinct group value might be represented by a different color. The series label that is associated with the group is assigned the group color. This option can be used to specify a single color for all series labels in a graph, without affecting items that have the group color, such as lines and marker symbols.

CURVELABELLOCATION=INSIDE | OUTSIDE

specifies the location of the step label relative to the plot area.

Default: INSIDE

INSIDE
: inside the plot area

OUTSIDE
: outside the plot area

Restriction: OUTSIDE cannot be used when the STEPPLOT is used in multicell layouts such as LATTICE, DATAPANEL, or DATALATTICE, where axes are external to the grid.

Interaction: For this option to take effect, the CURVELABEL= option must also be specified.

Interaction: This option is used in conjunction with the CURVELABELPOSITION= option to determine where the curve labels appear. For more information, see "Location and Position of Curve Labels" on page 123.

CURVELABELPOSITION=AUTO | MAX | MIN | START | END
: specifies the position of the step line labels relative to the step line.

Default: AUTO when CUVELABELLOCATION=OUTSIDE END when CURVELABELLOCATION=INSIDE

AUTO
: Only used when CURVELABELPOSITION=OUTSIDE. The step label is positioned automatically near the step boundary along unused axes whenever possible (typically Y2 and X2) to avoid collision with tick values.

MAX
: Forces the step label to appear near maximum step values (typically, upper right)

MIN
: Forces the step label to appear near minimum step values (typically, lower left)

START
: Only used when CURVELABELLOCATION=INSIDE. Forces the step label to appear near the beginning of the steps. Particularly useful when the step line "spirals" around.

END
: Only used when CURVELABELLOCATION=INSIDE. Forces the step label to appear near the end of the steps. Particularly useful when the step line "spirals" around.

Restriction: The AUTO setting is not supported if CURVELABELLOCATION= INSIDE is specified. The START and END settings are not supported if CURVELABELLOCATION=OUTSIDE is specified.

Interaction: For this option to take effect, the CURVELABEL= option must also be specified.

Interaction: This option is used in conjunction with the CURVELABELLOCATION= option to determine where the series label appears. For more information, see "Location and Position of Curve Labels" on page 123.

DATALABEL=*column* | *expression*
: specifies a column that supplies values for the data point labels.

Default: no default

The label positions are adjusted to prevent the labels from overlapping.

DATALABELATTRS=*style-element* | *style-element* (*text-options*) | (*text-options*)
specifies the color and font attributes of the data labels. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Text Options" on page 566 for available *text-options*.

Default:

- For non-grouped data, the GraphDataText style element.
- For grouped data, the GraphData1:ContrastColor - GraphDataN:ContrastColor style references.

Interaction: For this option to take effect, the DATALABEL= option must also be specified.

Interaction: This option's COLOR= setting overrides the colors indicated by the GROUP= option.

When the GROUP= option is used, each distinct group value might be represented by a different color. The data label that is associated with the group is assigned the group color. This option can be used to specify a single color for all data labels in a graph, without affecting items that have the group color, such as error bars and marker symbols.

DATATRANSPARENCY=*number*
specifies the degree of the transparency of the step lines, markers, and error bars, when displayed.

Default: 0

Range: 0 (opaque) to 1 (entirely transparent)

DISCRETEOFFSET=*number*
specifies an amount to offset all step lines and markers from the X values for a discrete X axis.

Note: This feature is for the third maintenance release of SAS 9.2 and later.

Default: 0 (no offset, all step lines and markers are centered on the discrete X values)

Range: -0.5 to +0.5, where .5 represents half the distance between discrete ticks. A positive offset is to the right. If the layout's axis options set REVERSE=TRUE, then the offset direction is also reversed.

DISPLAY=STANDARD | ALL | (*display-options*)
specifies whether to display markers on the step line.

Default: STANDARD

STANDARD
displays a step line without markers.

ALL
displays a step line with markers.

(*display-options*)
a list of options enclosed in parentheses. Currently, only the following option is available for the list:

MARKERS—displays a step line with markers

Use the MARKERATTRS= and LINEATTRS= options to control the appearance of the line and markers.

ERRORBARATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)
specifies the attributes of the error bars that are associated with the data points. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Line Options" on page 564 for available *line-options*.

Default:

- For non-grouped data, the GraphError style element.
- For grouped data, the LineStyle and LineThickness attributes of the GraphError style element and the ContrastColor attribute of the GraphData1 - GraphDataN style elements. (The LineStyle does not apply to the "serif" parts of the error bars.)

Interaction: For this option to take effect, error bars must be displayed by the ERRORLOWER= or ERRORUPPER= options.

ERRORLOWER=*numeric-column* | *expression*
specifies the values of the lower endpoints on the Y error bars.

Default: The lower segment of the error bars is not drawn.

Interaction: If the GROUP= option is specified, this option is ignored.

The error bar values must be absolute data values, not data values relative to the value of the bar.

The appearance of the error bars is controlled by the ERRORBARATTRS= option.

ERRORUPPER=*numeric-column* | *expression*
specifies the values of the upper endpoints on the Y error bars.

Default: The upper segment of the error bars is not drawn.

Interaction: If the GROUP= option is specified, this option is ignored.

The error bar values must be absolute data values, not data values relative to the value of the bar.

The appearance of the error bars is controlled by the ERRORBARATTRS= option.

GROUP=*column* | *expression*
creates a distinct set of lines, markers, and data labels for each unique group value of the specified column.

Default: Each distinct group value might be represented in the graph by a different combination of color, line pattern, and marker symbol. Lines and markers vary according to the ContrastColor, LineStyle, and MarkerSymbol attributes of the GraphData1 - GraphDataN style elements. Line thickness (for grouped and ungrouped data) is controlled by the LINEATTRS= option.

Interaction: The group values are mapped in the order of the data, unless the INDEX= option is used to alter the default sequence of marker symbols, colors, and line patterns.

Interaction: The marker size is set by the MARKERATTRS= option.

The representations that are used to identify the groups can be overridden. For example, each distinct group value is often represented by a different line pattern, but the LINEATTRS= (PATTERN=*pattern*) option could be used to assign the same line pattern to all of the plot's line patterns, letting line color indicate group values. Likewise, LINEATTRS= (COLOR= *color*) could be used to assign the same color to all lines, letting line pattern indicate group values.

INDEX=*positive-integer-column* | *expression*
specifies indices for mapping step-line attributes (color, marker symbol, and line pattern) to one of the GraphData1 - GranphDataN style elements.

Default: no default

Interaction: For this option to take effect, the GROUP= option must also be specified.

Interaction: All of the indexes for a specific group value must be the same. Otherwise, the results are unpredictable.

Interaction: If this option is not used, then the group values are mapped in the order of the data.

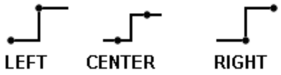
Interaction: The index values are 1-based indices. For the style elements in GraphData1 - GraphDataN, if the index value is greater than N, then a modulo operation remaps that index value to a number less than N to determine which style element to use.

Discussion: Indexing can be used to collapse the number of groups that are represented in a graph. For more information, see "Remapping Groups for Grouped Data" on page 121.

JOIN=*boolean*
specifies whether the steps are connected.

Default: TRUE

JUSTIFY= (LEFT | CENTER | RIGHT)
specifies the location of the data point relative to the step.



Default: LEFT

LEGENDLABEL= *"string"*
specifies a label for use in a legend.

Default: The Y-variable label. If a label is not defined, the Y-variable name.

Interaction: If the GROUP= option is specified, this option is ignored.

LINEATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)
specifies the attributes of the step line connecting the data points. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Line Options" on page 564 for available *line-options*.

Default:

- For non-grouped data, the GraphDataDefault style element.
- For grouped data, the ContrastColor and LineStyle attributes of the GraphData1 - GraphDataN style elements, and the GraphDataDefault:LineThickness style reference.

MARKERATTRS=*style-element* | *style-element* (*marker-options*) | (*marker-options*)
specifies the attributes of the data markers. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Marker Options" on page 565 for available *marker-options*.

Default:

- For non-grouped data, GraphDataDefault style element.
- For grouped data, the MarkerSymbol and ContrastColor attributes of the GraphData1 - GraphDataN style elements, and the GraphDataDefault:MarkerSize style reference.

Interaction: DISPLAY=ALL or DISPLAY=(MARKERS) must be set for this option to have any effect.

Interaction: This option's COLOR= suboption overrides the default behavior for grouped data: all markers the same color and the marker symbol alone distinguishes the markers.

Interaction: This option's SYMBOL= suboption overrides the default behavior for grouped data: it holds the marker constant across group values and marker color alone distinguishes the markers.

NAME=*"string"*
assigns a name to a plot statement for reference in other template statements.

Default: no default

Restriction: The *string* is case sensitive, cannot contain spaces, and must define a unique name within the template.

The specified name is used primarily in legend statements to coordinate the use of colors, marker symbols, and line patterns between the graph and the legend.

PRIMARY=*boolean*
specifies that the data columns for this plot be used for determining axis types and labels.

Default: FALSE

Restriction: This option is ignored if the plot is placed under a GRIDDED or LATTICE layout block.

Details: This option is needed only when two or more plots within an overlay-type layout contribute to a common axis. For more information, see "When Plots Share Data and a Common Axis" on page 397

ROLENAME=(*rolename-list*)
specifies user-defined roles for information contained in data columns.

Default: no default

(*rolename-list*)
a blank-separated list of *rolename* = *column* pairs.

For example, ROLENAME= (TIP1=OBS) assigns the column OBS to the user-defined role TIP1.

Requirement: The role names that you choose must be unique and different from the pre-defined roles X, Y, CURVELABEL, DATALABEL, ERRORLOWER, ERRORUPPER, GROUP, and INDEX.

Interaction: For this option to take effect, the TIP= option must also be used.

This option provides a way to add to the data columns that appear in tooltips specified by the TIP= option.

TIP=(*role-list*)
specifies the information to display when the cursor is positioned over the step line. If this option is used, it replaces all the information displayed by default. Roles for columns that do not contribute to the step plot can be specified along with roles that do.

Default: The columns assigned to these roles are automatically included in the tooltip information: X, Y, DATALABEL, ERRORLOWER, ERRORUPPER, and GROUP.

(*role-list*)

an ordered, blank-separated list of unique STEPPLOT and user-defined roles. STEPPLOT roles include X, Y, CURVELABEL, DATALABEL, ERRORLOWER, ERRORUPPER, and GROUP.

Note: CURVELABEL is considered a role only when it is assigned a column of values. It is not considered a role and does not display tooltips when assigned a string.

User-defined roles are defined with the ROLENAME= option.

The following example displays tooltips for the columns assigned to the roles X and Y as well as the column OBS, which is not assigned to any pre-defined StepPlot role. The OBS column must first be assigned a role.

```
ROLENAME=(TIP1=OBS)
TIP= (TIP1 X Y)
```

Requirement: To generate tooltips, you must include an ODS GRAPHICS ON statement that has the IMAGEMAP option specified, and write the graphs to the ODS HTML destination.

Interaction: The labels and formats for the TIP variables can be controlled with the TIPLABEL= and TIPFORMAT= options.

TIPFORMAT=(*role-format-list*)

specifies display formats for tip columns.

Default: The column format of the variable assigned to the role or BEST6. if no format is assigned to a numeric column.

(*role-format-list*)

a list of *rolename* = *format* pairs separated by blanks.

```
ROLENAME=(TIP1=OBS)
TIP= (TIP1 X ERRORUPPER ERRORLOWER)
TIPFORMAT=(ERRORUPPER=5.3 ERRORLOWER=5.3)
```

Requirement: This option provides a way to control the formats of columns that appear in tooltips. Only the roles that appear in the TIP= option are used. Columns must be assigned to the roles for this option to have any effect. See the ROLENAME= option.

TIPLABEL=(*role-format-list*)

specifies display labels for tip columns.

Default: The column label or column name of the variable assigned to the role.

(*role-label-list*)

a list of *rolename* = *"string"* pairs separated by blanks.

```
ROLENAME=(TIP1=OBS)
TIP= (TIP1 X ERRORUPPER ERRORLOWER)
TIPLABEL=(ERRORUPPER="1 STD ERROR"
ERRORLOWER="-1 STD ERROR")
```

Requirement: This option provides a way to control the labels of columns that appear in tooltips. Only the roles that appear in the TIP= option are used. Columns must be assigned to the roles for this option to have any effect. See the ROLENAME= option.

URL=*character-column*

specifies an HTML page to display when a step line segment is selected.

Default: no default

character-column
each value of the column must be a valid HTML page reference (HREF). For example, *http://www.sas.com/technologies/analytics/index.html*.

Requirement: To generate selectable bars, you must include an ODS GRAPHICS ON statement that has the IMAGEMAP option specified, and write the graphs to the ODS HTML destination.

XAXIS=X | X2
specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

Default: X

Interaction: The overall plot specification and the layout type determine the axis display for the specified axis. For more information, see "How Axis Features are Determined" on page 395.

YAXIS=Y | Y2
specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

Default: Y

Interaction: The overall plot specification and the layout type determine the axis display for the specified axis. For more information, see "How Axis Features are Determined" on page 395.

Chapter 39

SURFACEPLOTPARM Statement

Description and Syntax

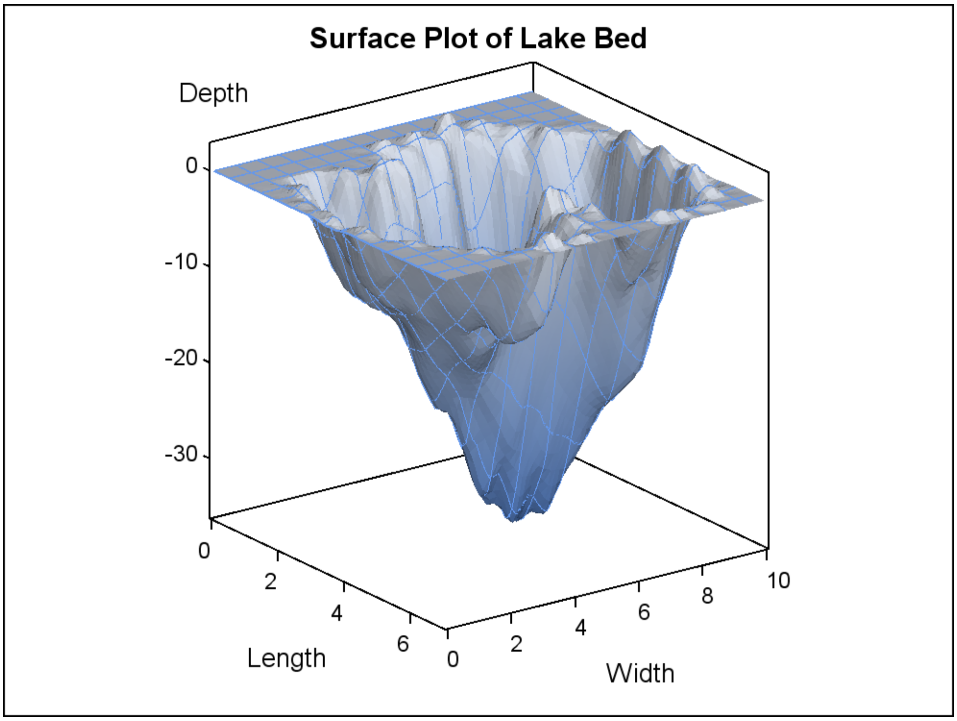
Creates a three-dimensional surface representing a response variable evaluated over a grid of X and Y values.

SURFACEPLOTPARM X = *numeric-column* | *expression*
Y = *numeric-column* | *expression*
Z = *numeric-column* | *expression* <*/option(s)*>;

Example Program and Statement Details

Example Graph

The following graph was generated by the "Example Program" on page 376:



Example Program

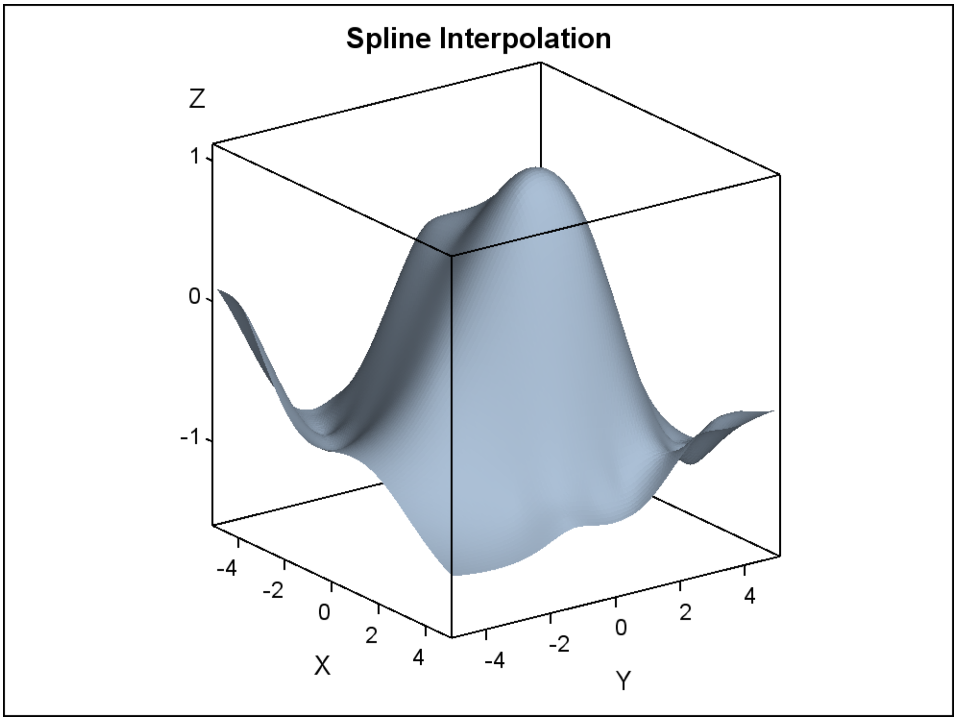
```
proc template;
  define statgraph surfaceplotparm;
    begingraph;
      entrytitle "Surface Plot of Lake Bed";
      layout overlay3d / cube=false;
        surfaceplotparm x=length y=width z=depth /
          reversecolormodel=true
          surfacecolorgradient=depth
          colormodel=twocoloraltramp;
      endlayout;
    endgraph;
  end;
run;
/* create gridded data for surface */
proc g3grid data=sashelp.lake out=gridded;
  grid width*length = depth / naxis1=75 naxis2=75;
run;

proc sgrender data= gridded template=surfaceplotparm;
run;
```

Statement Summary

The SURFACEPLOTPARM statement assumes that the Z response values have been provided for a uniform X-Y grid. Missing Z values leave a “hole” in the surface. The observations in the input data set should form an evenly spaced grid of horizontal (X and Y) values and one vertical (Z) value for each of these combinations. The observations should be in sorted order of Y and X to obtain an accurate graph.

You can use the G3GRID procedure to interpolate the necessary values to produce a data set with nonmissing Z values for every combination of X and Y. The G3GRID procedure can also smooth data with spline interpolations. For further details, see the documentation for PROC G3GRID in the *SAS/GRAPH: Reference* .

Using PROC G3GRID, the following code performs a Spline interpolation and generates this figure:

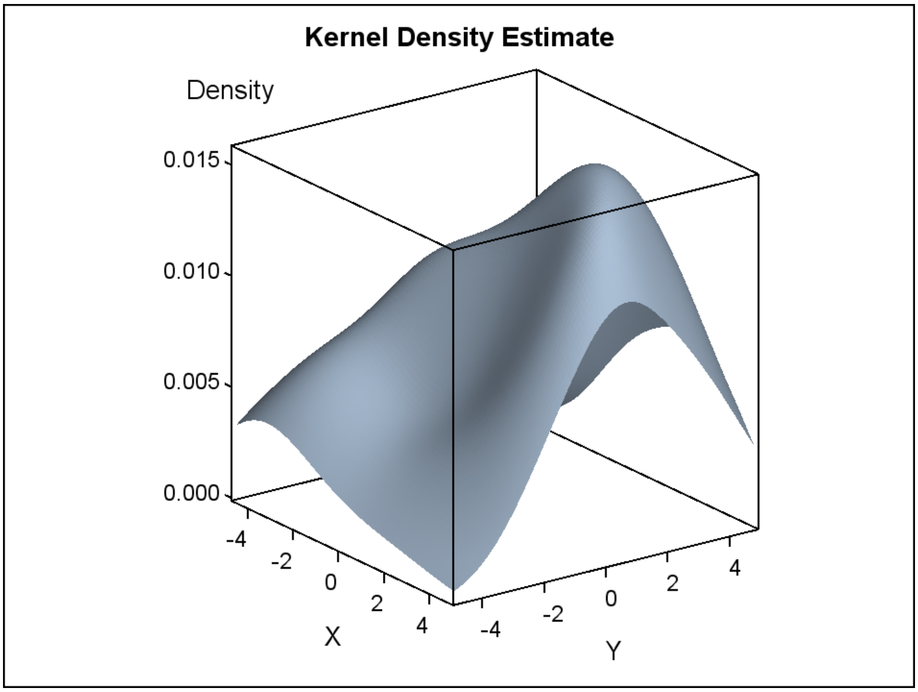


```
data nums;
  do i=1 to 30;
     X=10*ranuni(33)-5;
     Y=10*ranuni(35)-5;
     Z=sin(sqrt(x*x+y*y));
     output;
  end;
run;
proc g3grid data=nums out=gridded;
  grid y*x=z / spline
               axis1=-5 to 5 by .1
               axis2=-5 to 5 by .1;
run;
proc sort data=gridded; by y x; run;

proc template;
  define statgraph g3grid_surface;
   begingraph;
    entrytitle "Spline Interpolation";
    layout overlay3d;
      surfaceplotparm x=x y=y z=z /
        surfacetype=fill;
    endlayout;
   endgraph;
  end;
```

```
run;

proc sgrender data=gridded template=g3grid_surface;
run;
```

The KDE procedure can produce an output data set of gridded X-Y values where the Z value is computed to be a Kernel Density Estimate of the distribution of X and Y. For further details, see the documentation for PROC KDE in **MISSING Bookref** .

Using PROC KDE on the *nums* data generated in the previous example, the following code computes a Kernel Density Estimate and generates this figure:



```
/* use the nums data generated in the previous example */
proc kde data=nums;
  bivar x y / ngrid=100
    out=binned(rename=(value1=X value2=Y));
run;
proc sort data=binned; by y x;
  label x="X" y="Y";
run;

proc template;
  define statgraph kde_surface;
   begingraph;
     entrytitle "Kernel Density Estimate";
     layout overlay3d;
       surfaceplotparm x=x y=y z=density /
         surfacetype=fill;
     endlayout;
   endgraph;
  end;
run;
```

```
proc sgrender data=binned template=kde_surface;
run;
```

The SURFACEPLOTPARM does not support the tooltips that are enabled by the IMAGEMAP= option in the ODS GRAPHICS statement.

Required Arguments

X=*numeric-column* | *expression*
specifies the X coordinates of the grid.

Y=*numeric-column* | *expression*
specifies the Y coordinates of the grid.

Z=*numeric-column* | *expression*
specifies the height of response values.

Note: The input data should be sorted by both 1) the Y column and 2) the X column. The sort direction for Y should be ascending. The sort direction of X be either ascending or descending.

Options

Statement Option	Description
COLORMODEL	Specifies a style element that is used to determine the colors of the contour lines or filled areas.
DATATRANSPARENCY	Specifies the degree of the transparency of the surface.
FILLATTRS	Specifies the appearance of the filled surface or the wire-frame mesh.
LEGENDLABEL	Specifies a label for use in a legend.
NAME	Assigns a name to a plot statement for reference in other template statements.
PRIMARY	Specifies that the data columns for this plot and the plot type be used for determining default axis features.
REVERSECOLORMODEL	Specifies whether to reverse a gradient defined by the COLORMODEL= option.
SURFACECOLORGRADIENT	Specifies a column that is used to map surface colors to a continuous gradient.
SURFACETYPE	Specifies how the surface is displayed.

COLORMODEL=*style-element*
specifies a style element that is used to determine gradient surface colors.

Default: The ThreeColorRamp style element.

style-element
Name of a style element. The style element should contain these style attributes:

STARTCOLOR	specifies a color for the smallest data value of the SURFACECOLORGRADIENT variable
NEUTRALCOLOR	specifies a color for the midpoint of the range Of the SURFACECOLORGRADIENT variable
ENDCOLOR	specifies a color for the highest data value of the SURFACECOLORGRADIENT variable

Interaction: For this option to have any effect, the SURFACECOLORGRADIENT= option must also be used.

The REVERSECOLORMODEL= option can be used to reverse the start and end colors of the ramp assigned to the color model.

DATATRANSPARENCY=*number*
specifies the degree of the transparency of the surface.

Default: 0

Range: 0 (opaque) to 1 (entirely transparent)

FILLATTRS=*style-element* | *style-element* (*fill-options*) | (*fill-options*)
specifies the color of the filled surface or the wire-frame mesh. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Fill Options" on page 564 for available *fill-options*.

Default: The GraphDataDefault:Color style reference.

Interaction: The SURFACECOLORGRADIENT= option is ignored if this option is specified.

LEGENDLABEL= *"string"*
specifies a label for use in a legend.

Default: The Z-variable label. If a label is not defined, the Z-variable name.

NAME=*"string"*
assigns a name to a plot statement for reference in other template statements.

Default: no default

Restriction: The *string* is case sensitive, cannot contain spaces, and must define a unique name within the template.

The specified name is used primarily in legend statements to coordinate the use of colors and line patterns between the graph and the legend.

PRIMARY=*boolean*
specifies that the data columns for this plot and the plot type be used for determining default axis features.

Default: FALSE

Details: This option is needed only when two or more plots within an OVERLAY3D layout contribute to a common axis. For more information, see "When Plots Share Data and a Common Axis" on page 397

REVERSECOLORMODEL=*boolean*
specifies whether to reverse a gradient (color ramp) defined by the COLORMODEL= option.

Default: FALSE

SURFACECOLORGRADIENT=*numeric-column*
specifies a column that is used to map surface colors to a continuous gradient.

Default: no default

Interaction: This option is ignored if the FILLATTRS== option is specified.

SURFACETYPE=FILLGRID | FILL | WIREFRAME
specifies how the surface is displayed.

Default: FILLGRID

FILLGRID
a filled surface with superimposed grid lines

FILL
a filled surface without grid lines

WIREFRAME
an unfilled surface with grid lines

Chapter 40
VECTORPLOT Statement

Description and Syntax

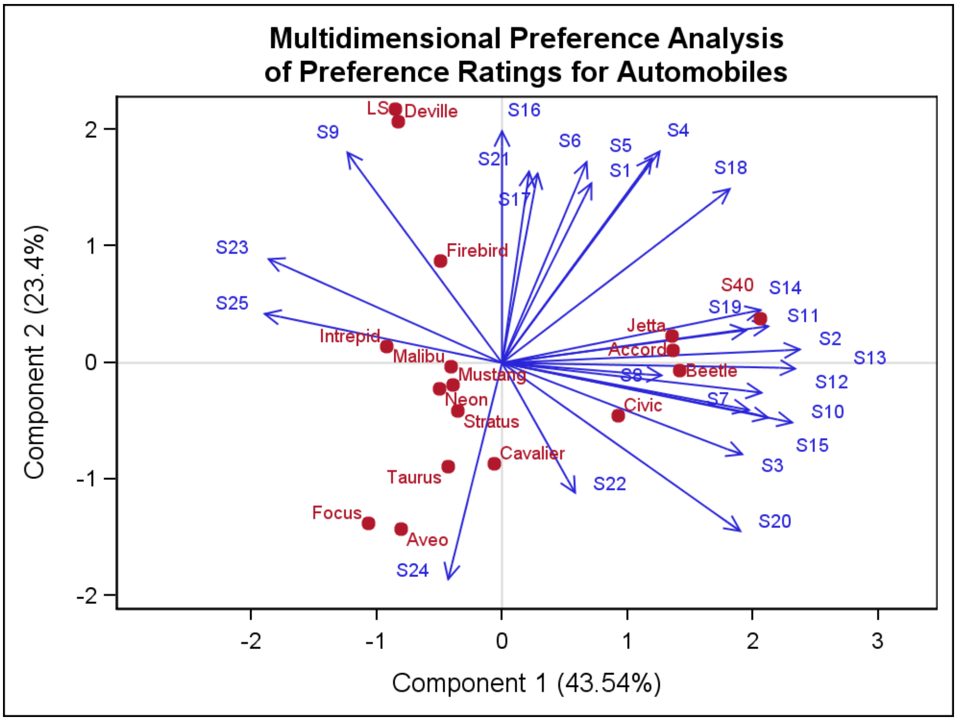
Creates a plot of vectors (directed line segments).

VECTORPLOT X = *numeric-column* | *expression*
Y = *numeric-column* | *expression*
XORIGIN = *numeric-constant* | *numeric-column* | *expression*
YORIGIN = *numeric-constant* | *numeric-column* | *expression* </*option(s)*>;

Example Program and Statement Details

Example Graph

The following graph was generated by the "Example Program" on page 384:



Example Program

```
data CarPref;
  input Make $12. Model $13. (S1-S25) (1.);
datalines;
Cadillac   Deville      8007990491240508971093809
Chevrolet  Aveo         0051200423451043003515698
Chevrolet  Cavalier     4053305814161643544747795
Chevrolet  Malibu       6027400723121345545668658
Dodge      Intrepid     7006000434101107333458708
Dodge      Stratus      3005005635461302444675655
Dodge      Neon         4005003614021602754476555
Ford       Taurus       2024006715021443530648655
Ford       Mustang      5007197705021101850657555
Ford       Focus        0021000303030201500514078
Honda      Accord       5956897609699952998975078
Honda      Civic        4836709507488852567765075
Lincoln    LS           7008990592230409962091909
Pontiac    Firebird     0107895613201206958265907
Volkswagen Jetta        4858696508877795377895000
Volkswagen Beetle       4858509709695795487885000
Volvo      S40          9989998909999987989919000
;
run;
* Compute Two Component Model;
ods graphics;
ods exclude all;
ods output mdprefplot=plotdata;
proc prinqual data=CarPref n=2 replace mdpref method=mgv;
  id model;
  transform monotone(S1-S25);
```

```
run;
ods select all;

proc template;
 define statgraph vectorplot;
  begingraph;
   entrytitle "Multidimensional Preference Analysis";
   entrytitle "of Preference Ratings for Automobiles";
   layout overlayequated / equatetype=fit cycleattrs=true;
    referenceline y=0 / datatransparency=.7;
    referenceline x=0 / datatransparency=.7;
    vectorplot y=vec2 x=vec1 xorigin=0 yorigin=0 /
      datalabel=label2var shaftprotected=false;
    scatterplot y=prin2 x=prin1 /
      datalabel=idlab1 primary=true
      markerattrs=(symbol=circlefilled);
   endlayout;
  endgraph;
 end;
run;

proc sgrender data=plotdata template=vectorplot;
run;
```

Statement Summary

By default in a VECTORPLOT, each vector starts at 0, 0 in the data space. To specify alternative coordinates for the starting point, use the XORIGIN= and YORIGIN= arguments.

Required Arguments

X=*numeric-column* | *expression*
specifies the column for the X values of the vector endpoints.

Y=*numeric-column* | *expression*
specifies the column for the Y values of the vector endpoints.

XORIGIN=*numeric-constant* | *numeric-column* | *expression*
Specifies the X data coordinate of the vector origin.

YORIGIN=*numeric-constant* | *numeric-column* | *expression*
Specifies the Y data coordinate of the vector origin.

Options

Statement Option	Description
ARROWDIRECTION	Specifies the direction of the arrowhead at the end of a vector.
ARROWHEADS	Specifies whether arrowheads are displayed on the vectors.

Statement Option	Description
ARROWHEADSHAPE	Specifies the shape of the arrowheads.
CLIP	Specifies whether the origin is considered when determining the data ranges for the axes.
DATALABEL	Specifies the labels at the ends of the vectors.
DATALABELATTRS	Specifies the color and font attributes of the data labels.
DATATRANSPARENCY	Specifies the degree of the transparency of the vector line and arrow.
GROUP	Creates a separate vector plot for each unique group value of the specified column.
INDEX	Specifies indices for mapping line attributes (color and pattern) to one of the GraphData1 - GranphDataN style elements.
LEGENDLABEL	Specifies a label for use in a legend.
LINEATTRS	Specifies the properties of the series line.
NAME	Assigns a name to a plot statement for reference in other template statements.
PRIMARY	Specifies that the data columns for this plot and the plot type be used for determining default axis features.
ROLENAME	Specifies roles for information defined by data columns.
SCALE	Specifies the scale factor of the vector length.
TIP	Specifies the information to display when the cursor is positioned over a vector line.
TIPFORMAT	Specifies display formats for information defined by roles.
TIPLABEL	Specifies display labels for information defined by roles.
XAXIS	Specifies whether data are mapped to the primary X (bottom) axis or the secondary X2 (top) axis.
YAXIS	Specifies whether data are mapped to the primary Y (left) axis or the secondary Y2 (right) axis.

ARROWDIRECTION=OUT | IN |BOTH

specifies the placement of the arrowhead(s) at the end of the vector.

Default: OUT

OUT

specifies a single arrowhead, pointing away from the origin, at the end of the vector away from the origin.

IN

specifies a single arrowhead, pointing toward the origin, at the end of the vector near the origin.

BOTH

specifies two arrowheads. One arrowhead points away from the origin, at the end of the vector opposite from the origin. The other arrowhead points toward the origin, at the end of the vector near the origin.

Interaction: If ARROWHEADS=FALSE, this option is ignored.

Use the ARROWHEADSHAPE= option to control arrowhead appearance.

ARROWHEADS=*boolean*

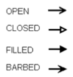
specifies whether arrowheads are displayed on the vectors.

Default: TRUE

Interaction: When this option is set to FALSE, the ARROWDIRECTION= and ARROWHEADSHAPE= options are ignored and all vectors are displayed as undirected line segments.

ARROWHEADSHAPE=OPEN | CLOSED | FILLED | BARBED

specifies the shape of the arrowheads.



Default: OPEN

Interaction: This option is ignored if ARROWHEADS=FALSE.

Use the ARROWDIRECTION= option to control arrow direction.

CLIP=*boolean*

specifies whether the origin is considered when determining the data ranges for the axes.

Default: FALSE.

FALSE

The origin contributes to data range for each axis. Each axis might be extended to force the display of the origin.

TRUE

The origin is ignored when establishing axis scales. Each axis scale is determined by the other plots in the overlay. This might result in the origin not being displayed if its data range is not within the data ranges of tips of the vectors.

DATALABEL=*column* | *expression*

specifies the labels at the ends of the vectors.

Default: no default

The label positions are automatically adjusted to prevent the labels from colliding with other labels and other arrows.

DATALABELATTRS=*style-element* | *style-element* (*text-options*) | (*text-options*)

specifies the color and font attributes of the data labels. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Text Options" on page 566 for available *text-options*.

Default:

- For non-grouped data, the GraphDataText style element.
- For grouped data, the GraphData1:ContrastColor - GraphDataN:ContrastColor style references.

Interaction: For this option to take effect, the DATALABEL= option must also be specified.

DATATRANSPARENCY=*number*

specifies the degree of the transparency of the vector line and arrow.

Default: 0

Range: 0 (opaque) to 1 (entirely transparent)

GROUP=*column* | *expression*

creates a distinct set of vector lines and data label colors for each unique group value of the specified column.

Default: Each distinct group value might be represented in the graph by a different combination of color and line pattern. Lines vary according to the ContrastColor and LineStyle attributes of the GraphData1 - GraphDataN style elements.

Interaction: The group values are mapped in the order of the data, unless the INDEX= option is used to alter the default sequence of line patterns and colors.

The LINEATTRS= option can be used to override the representations that are used to identify the groups. For example, LINEATTRS=(PATTERN=SOLID) can be used to assign the same pattern to all of the lines, letting the line color distinguish group values. Likewise, LINEATTRS=(COLOR=BLACK) can be used to assign the same color to all of the lines, letting the line pattern distinguish group values.

INDEX=*positive-integer-column* | *expression*

specifies indices for mapping line attributes (color and pattern) to one of the GraphData1 - GranphDataN style elements.

Default: no default

Interaction: For this option to take effect, the GROUP= option must also be specified.

Interaction: All of the indexes for a specific group value must be the same. Otherwise, the results are unpredictable.

Interaction: If this option is not used, then the group values are mapped in the order of the data.

Interaction: The index values are 1-based indices. For the style elements GraphData1 - GraphDataN, if the index value is greater than N, then a modulo operation remaps that index value to a number less than N to determine which style element to use.

Discussion: Indexing can be used to collapse the number of groups that are represented in a graph. For more information, see "Remapping Groups for Grouped Data" on page 121.

LEGENDLABEL= *"string"*

Specifies a label for use in a legend.

Default: The *string* specified on the NAME= option.

Interaction: If the GROUP= option is specified, this option is ignored.

LINEATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)
specifies the attributes of the vector line and arrowhead. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Line Options" on page 564 for available *line-options*.

Default:

- For non-grouped data, the GraphDataDefault style element.
- For grouped data, the ContrastColor, LineStyle, and LineThickness attributes of the GraphData1 - GraphDataN style elements.

NAME=*"string"*
assigns a name to a plot statement for reference in other template statements.

Default: no default

Restriction: The *string* is case sensitive, cannot contain spaces, and must define a unique name within the template.

Interaction: The *string* is used as the default legend label if the LEGENDLABEL= option is not used.

The specified name is used primarily in legend statements to coordinate the use of colors and line patterns between the graph and the legend.

PRIMARY=*boolean*
specifies that the data columns for this plot and the plot type be used for determining default axis features.

Default: FALSE

Restriction: This option is ignored if the plot is placed under a GRIDDED or LATTICE layout block.

Details: This option is needed only when two or more plots within an overlay-type layout contribute to a common axis. For more information, see "When Plots Share Data and a Common Axis" on page 397

ROLENAME=(*role-name-list*)
specifies user-defined roles for information contained in data columns.

Default: no default

(*role-name-list*)
a blank-separated list of *rolename* = *column* pairs.

For example, ROLENAME= (TIP1=OBS) assigns the column OBS to the user-defined role TIP1.

Requirement: The role names that you choose must be unique and different from the pre-defined roles X, Y, DATALABEL, XORIGIN, YORIGIN, GROUP, and INDEX.

Interaction: For this option to take effect, the TIP= option must also be used.

This option provides a way to add to the data columns that appear in tooltips specified by the TIP= option.

SCALE=*number*
specifies the scale factor of the vector length.

Default: 1.0

Restriction: The number specified must be greater than 0.

TIP=(*role-list*)

specifies the information to display when the cursor is positioned over a vector line. If this option is used, it replaces all the information displayed by default. Roles for columns that do not contribute to the vector plot can be specified along with roles that do.

Default: The columns assigned to these roles are automatically included in the tooltip information: X, Y, DATALABEL, XORIGIN, YORIGIN, and GROUP.

(*role-list*)

an ordered, blank-separated list of unique VECTORPLOT and user-defined roles. VECTORPLOT roles include X, Y, DATALABEL, XORIGIN, YORIGIN, GROUP, and INDEX.

User-defined roles are defined with the ROLENAME= option.

The following example displays tooltips for the columns assigned to the roles X, Y, GROUP, and the column OBS, which is not assigned to any pre-defined VectorPlot role. The OBS column must first be assigned a role.

```
ROLENAME=(TIP1=OBS)
TIP= (TIP1 X Y GROUP)
```

Requirement: To generate tooltips, you must include an ODS GRAPHICS ON statement that has the IMAGEMAP option specified, and write the graphs to the ODS HTML destination.

Interaction: The labels and formats for the TIP variables can be controlled with the TIPLABEL= and TIPFORMAT= options.

TIPFORMAT=(*role-format-list*)

specifies display formats for tip columns.

Default: The column format of the variable assigned to the role.

(*role-format-list*)

a list of *rolename* = *format* pairs separated by blanks.

```
ROLENAME=(TIP1=OBS)
TIP=(TIP1 X Y GROUP)
TIPFORMAT=(TIP1=3.)
```

Requirement: Columns must be assigned to the roles for this option to have any effect. See the ROLENAME= option.

This option provides a way to control the formats of columns that appear in tooltips. Only the roles that appear in the TIP= option are used.

TIPLABEL=(*role-format-list*)

specifies display labels for tip columns.

Default: The column label or column name of the variable assigned to the role.

(*role-label-list*)

a list of *rolename* = *"string"* pairs separated by blanks.

```
ROLENAME=(TIP1=OBS)
TIP=(TIP1 X Y GROUP)
TIPLABEL=(TIP1="Observation #")
```

Requirement: Columns must be assigned to the roles for this option to have any effect. See the ROLENAME= option.

This option provides a way to control the labels of columns that appear in tooltips. Only the roles that appear in the TIP= option are used.

XAXIS=X | X2
specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

Default: X

Interaction: The overall plot specification and the layout type determine the axis display. For more information, see "How Axis Features are Determined" on page 395.

YAXIS=Y | Y2
specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

Default: Y

Interaction: The overall plot specification and the layout type determine the axis display. For more information, see "How Axis Features are Determined" on page 395.

Part 5

Plot Axes

Chapter 41
Axis Features in Layouts

Overview

GTL graphs are specified within layout blocks that enable you to control the graph display features, including the display of the axes for the plots within the layout. For example, the LAYOUT OVERLAY statement has XAXISOPTS= and YAXISOPTS= options that enable you to specify axis features for the plots within the layout.

The following sections explain how the axis features are determined in a layout. The sections also discuss the issues you must consider when setting the axis type or adjusting the appearance of the axis display.

How Axis Features are Determined

Overview

The GTL uses various criteria to determine the displayed axis features for a graph. Generally, axis features are based on the layout type, the order of plot statements in the layout and the options specified on those statements, the use of primary and secondary axes on the plots (when secondary axes are supported), the plot type, the column(s) of data that contribute to defining the axis range, and the data formats for the contributing data columns.

While the default axis features depend on a combination of the factors above, it is useful to understand how the axis features are determined in the templates that you build:

- how the data are mapped to the plot axes
- how the various layout types manage the axes.

Plot Data Are Mapped to a Designated Axis

Depending on the layout type and the plots that you specify within the layout, you can manage up to four axes for 2-D plots:

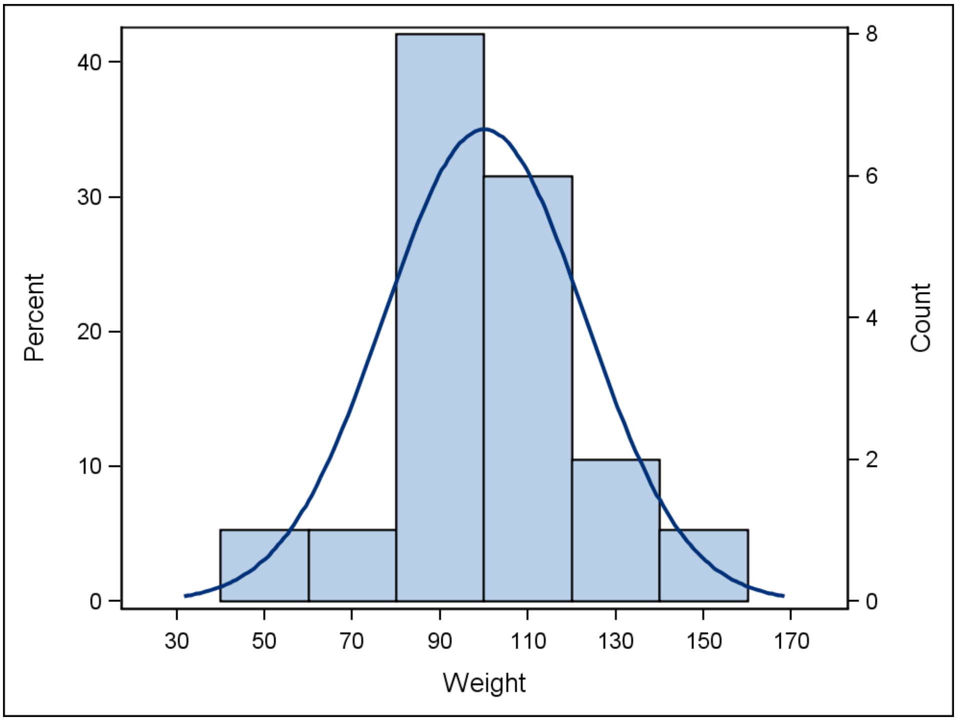
- a primary X (bottom) axis
- a primary Y (left) axis
- a secondary X axis (X2 or “top” axis)
- a secondary Y axis (Y2 or “right” axis)

By default, plot data are mapped to the primary axes. To enable you to override the default, plot statements that support a secondary X2 axis provide an XAXIS= option that can map data to the X or X2 axis. Plot statements that support a Y axis provide a YAXIS= option that can map data to the Y or Y2 axis.

To determine the axis features within a layout, the GTL must first determine what data must be mapped to a particular axis. Thus, your use of primary and secondary axes on plot specifications affects the GTL’s determination of default axis features for the layout.

For example, the plot statements in the following template specify Y-data mappings to the Y2 and Y axes:

```
proc template;
  define statgraph y2axis;
    begingraph;
      layout overlay;
        histogram weight / scale=count yaxis=y2;
        histogram weight / scale=percent yaxis=y;
        densityplot weight / normal();
      endlayout;
    endgraph;
  end;
run;
proc sgrender data=sashelp.class template=y2axis;
run;
```



In this example, the first HISTOGRAM maps its Y-axis data to the Y2 axis, and the second HISTOGRAM maps its Y-axis data to the Y axis. The DENSITY plot does not explicitly map its Y-axis data, so the default Y axis is used. None of the plots explicitly map X-data, so the default X axis is used for all three plots. Thus, the GTL must manage any interactions that result from representing multiple plots on the X and Y axes. For example, on the X axis, it must determine an appropriate data range for representing the data values of all three plots.

When establishing axis features for each axis, the GTL determines which plot specifications map data to the axis. The GTL also collects the data for all of the plots that must be represented and maps that data to the designated axis. The next section discusses the criteria the GTL uses to determine the axis features for the axes after this mapping has been done for each axis.

Note: When layout statements are nested, the complete nesting structure must be taken into account. For example, if multiple plot statements are specified within an overlay-type layout that is in turn nested in a lattice-type layout, each layout's axis restrictions apply to the graph.

When Plots Share Data and a Common Axis

Overview

If a layout block contains multiple plots that share data and a common axis, the plot settings often interact in ways that affect the axis features. Axis features include the axis type, axis label, tick-mark layout, and so on. The GTL resolves these interactions in ways that vary according to the layout block and plot statements.

Note: Axis interactions might not occur if other settings in the template prevent them. As discussed in in "Plot Data Are Mapped to a Designated Axis" on page 396, if two plot statements are within an OVERLAY layout, one of them might map its data to the X axis and the other might map its data to the X2 (top) axis, thus avoiding the interactions that might occur if they both mapped their data to the X axis.

Axis Features in Overlay-type Layouts

Overlay-type layouts (OVERLAY, OVERLAYEQUATED, and PROTOTYPE, for example) build a composite from one or more *GTL-statements*.

Within overlay-type layouts, if you do not explicitly set axis features in your template statements, the GTL automatically determines them. It sets the axis features based on the layout(s) and plot(s) in the layout block and the data that are associated with the template at run time.

If only one plot statement within an overlay-type layout generates an axis, then determining axis features is straight forward: the features are derived directly from the plot type and the columns that are used for the plot data. For example, if a LAYOUT OVERLAY block contains a single SCATTERPLOT and the X variable specifies a numeric column of children's weights, the default X-axis type is LINEAR. The default X-axis label is the column label of the WEIGHT variable. If the WEIGHT variable has no defined label, the column name is used as a label.

When an overlay-type layout contains multiple plots that generate axes, the GTL can determine default axis features for the shared axes. Alternatively, you can use the PRIMARY= option on one of the plot statements to specify which plot you want the GTL to use. The following code fragment explicitly specifies that the SCATTERPLOT of children's weights be used to determine axis features within the layout:

```
layout overlay;
  scatterplot x=weight ... / primary=true;
  ...
```

- If no plot in an overlay-type layout is designated as primary, the data columns associated with the first plot that generates an axis are considered primary on a per-axis basis.
- If PRIMARY=TRUE for a plot within an overlay-type layout, that plot's data columns and data type determine the default axis features. An explicitly specified primary plot determines the default axis features regardless of where that plot statement occurs within the layout block.
- Only one plot can be primary on a per-axis basis. If multiple plots specify PRIMARY=TRUE for the same axis, the last one encountered is considered primary.

The following SCATTERPLOT specifies a string variable on the X= argument:

```
layout overlay;
  scatterplot x=name ... / primary=true;
  ...
```

In this case, the default X-axis type is DISCRETE and the X-axis label is the column label of NAME, or the column name if no label exists.

If a SCATTERPLOT's X= argument specifies a column that has a SAS DATETIME format, the default X-axis type is TIME and the default X-axis label is the column label or name of the datetime column:

```
layout overlay;
  scatterplot x=date ... / primary=true;
  ...
```

For some plot types, the default axis type does not directly correlate to the specified column's data type. For example, the following code fragment specifies a BARCHART for the numeric column AGE:

```
layout overlay;
  barchart x=age ... / primary=true;
  ...
```

Because a BARCHART requires a discrete X axis, the default X-axis type in this case is DISCRETE, in spite of the fact that column AGE is numeric. The X-axis label is the column label of AGE, or the column name if no label exists.

Finally, consider a HISTOGRAM that is set as the primary plot in the layout and that bins data values:

```
layout overlay;
  histogram weight / binaxis=true primary=true;
  ...
```

In this case, the default X-axis type is LINEAR, but the histogram's data bins are used by default as the basis for the axis tick marks.

Axis Features in Data Panel and Data Lattice Layouts

The criteria discussed in"Axis Features in Overlay-type Layouts" on page 398 apply to determining the default axis features for the plots within DATAPANEL and DATALATTICE layouts. Both of these layout types nest a LAYOUT PROTOTYPE statement within their layout blocks. In both cases, the plot statements within the LAYOUT PROTOTYPE block—an overlay-type layout—determine the axis features for the plot display.

However, one distinction is worth noting. The DATAPANEL and DATALATTICE layouts do not support the display of secondary X2 (top) or secondary Y2 (right) axes. Thus, you cannot use secondary axes within these layout types to avoid competition among layout cells for determining axis features in the plot display.

Axis Features in Lattice-type Layouts

The LAYOUT LATTICE statement can create a grid of graphs that automatically aligns plot areas, data display areas, labels, and headers across the columns and rows in the layout. The layout gives you the option of unifying the scale of the data ranges that are displayed in the graphs.

If a LAYOUT LATTICE specification generates only one cell, then no competition exists between cells for determining axis features in the display. In this case, the axis features are derived directly from the plot type and the columns used for the plot data.

Similarly, for multi-cell displays, if either or both of the options COLUMNDATARANGE= or ROWDATARANGE= uses the DATA setting to scale axis data ranges separately for each cell in the layout, then the layout cells are independent. The independent cells do not interact with each other for determining the axis features in the display.

Axes are shared in the layout when either or both of the options COLUMNDATARANGE= or ROWDATARANGE= is used to unite axis data ranges for layout cells. By default in those cases, the first cell that is drawn (by default, the top left cell) determines the axis features in the display. The same features are used in all of the grid's layout cells. If you specify external axes for the columns or rows in the layout, you can specify desired axis features on the appropriate COLUMNAXIS or ROWAXIS statements used in the layout. In that case, any XAXIS= or YAXIS= options on individual plot statements are ignored.

For an example LATTICE layout with external axes, see the Example Program inChapter 7, "LAYOUT LATTICE Statement," on page 73.

Note: LATTICE layouts support stand-alone plots, which are single plot statements that are specified directly in the LATTICE layout block and are not nested within an overlay-type layout. Stand-alone plots within a LATTICE layout cannot support a secondary axis, so the layout in those cases ignores any XAXIS= or YAXIS= options on the plot statement.

Axis Features in Gridded Layouts

In a GRIDDED layout the layout cells are independent of one another. Plot statements within the layout cells do not share data and are not represented on a common axis. Thus, no competition exists among layout cells for determining the axis features.

Note: GRIDDED layouts support "stand-alone plots," which are single plot statements that are specified directly in the GRIDDED layout block and are *not* nested within an overlay-type layout. Stand-alone plots within a GRIDDED layout cannot support a secondary axis, so the layout in those cases ignores any XAXIS= or YAXIS= options on the plot statement.

Plot Axis Types Must Agree on Common Axes

The GTL is extremely flexible and enables you to generate a wide variety of plot displays. However, if you request incompatible plot displays within the same layout, the results are unpredictable.

"When Plots Share Data and a Common Axis" on page 397 discusses the criteria GTL uses to determine the default axis features. After the axis type has been determined, the GTL expects that all plots that share that axis will have the assigned axis type. The expectation applies whether you specify axis features in your template or let GTL determine default features.

For example, a BOXPLOT cannot be overlaid by a LINEPARM: the two types of plot display are incompatible within the same set of axes. Thus, if you were to use both a BOXPLOT statement and a LINEPARM statement within a LAYOUT OVERLAY block, only one of them can be displayed. The GTL therefore displays the primary plot (the first specified plot by default, or the plot designated as primary by setting PRIMARY=TRUE). The other plot is not displayed.

Similarly, a BARCHART requires a discrete X-axis, whereas a HISTOGRAM cannot be displayed on a discrete axis. If you specify both a BARCHART and a HISTOGRAM within the same overlay-type layout, only the primary plot is displayed and the other plot is rejected from the display.

Axis types must also be the same for plots that must share an axis across the columns or rows in a multi-cell layout. For example, in a LAYOUT LATTICE, the GTL expects that plots have the same axis type and data ranges if they are to share an external axis. Otherwise, the external axis cannot be displayed for that row or column.

Controlling Axis Features

Overview

To enable you to control axis features within each of the layout types, there are different sets of axis options for the different types of axes:

Option Category	Used With ...
2-D Axis Options	LAYOUT OVERLAY statement
3-D Axis Options	LAYOUT OVERLAY3D statement

Option Category	Used With ...
Equated Axis Options	LAYOUT OVERLAYEQUATED statement
Lattice Axis Options	LAYOUT LATTICE statement
DataLattice/DataPanel Axes	LAYOUT DATALATTICE and LAYOUT DATAPANEL statements

The options that are available for each layout are documented separately, but it is worth discussing the following tasks in general for all of the layout types:

- Setting the Axis Type
- Adjusting the Axis View
- Adjusting Axis Thresholds
- Adjusting Axis Offsets

Setting the Axis Type

Within any given layout in the graph display, each plot axis is always of a particular type. In the default cases, the axis type is always LINEAR, DISCRETE, or TIME.

The axis options for each layout statement include a TYPE= option that enables you to specify an axis type that overrides the default selection mechanisms. When you override the default axis type, you must be sure to specify the correct axis type for the plot(s) that you are defining. For every plot in the template language, the documentation indicates what axis types it supports.

Each axis type has features specific to that type, and the following axis options enable you to specify features for the different types:

```
LINEAROPTS = ( linear-suboptions )
DISCRETEOPTS = ( discrete-suboptions )
TIMEOPTS= ( time-suboptions )
LOGOPTS = ( log-suboptions )
```

One or more of these options can be specified for an axis, but the specified settings are applied only to the axis type that supports them.

For example, a BarChart has two axes – a TYPE=DISCRETE axis for the X axis and a TYPE=LINEAR axis for the Y axis. If a numeric column (for example, Age) is assigned to the X role, this column’s values are always treated as discrete values, never as a continuous range of values. You cannot request another axis type for the X axis, but you can request a different axis type for the Y axis.

Sometimes you want a specialized axis type depending on the nature of the data. For example, if the data have a very large range of values (orders of magnitude apart), you could request that the values be displayed on a logarithmic scale. To set a logarithmic scale, use the TYPE=LOG axis option.

Time series data benefit from displaying the X axis with a TYPE=TIME axis. A TIME axis type requires that the column values are SAS Date, Time, or Datetime values.

3-D plots such as BIHISTOGRAM3DPARM and SURFACEPLOTPARM always use TYPE=LINEAR for X, Y, and Z axes.

Note: Certain plot types or layouts might impose restrictions on what type of axis can be assigned. The documentation for each plot and layout type identifies any restrictions that might apply to the axes.

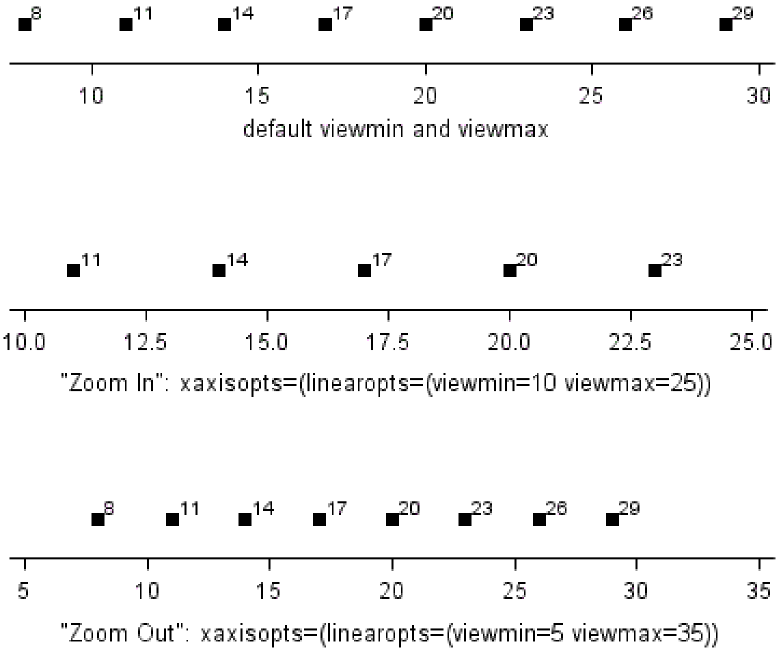
Adjusting the Axis View

The VIEWMIN= and VIEWMAX= axis options can be used to adjust the view of an axis. You can specify minimum data values to include in the display, maximum data values, or both (the specified values might be adjusted by the threshold calculation). By default, the VIEWMIN= value is the minimum data value for the specified axis and the VIEWMAX= value is the maximum data value for the specified axis.

A VIEWMIN= value that is greater than the data minimum or a VIEWMAX= value that is less than the data maximum acts like a “zoom in” operation. The adjusted view reduces the range of values represented on the axis and can sometimes exclude markers, lines, or fills that would normally appear.

A VIEWMIN= value that is less than the data minimum or a VIEWMAX= value that is greater than the data maximum acts like a “zoom out” operation. The adjusted view extends the range of values represented on the axis and sometimes compresses the markers, lines, or fills into a smaller area.

The following figure shows how the view settings can affect the tick and data displays.



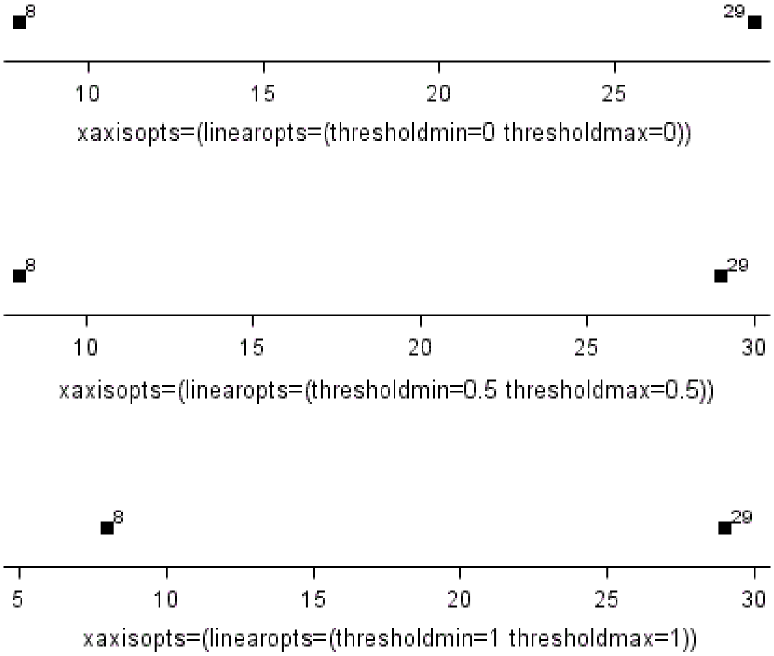
Adjusting Axis Thresholds

The THRESHOLDMIN = and THRESHOLDMAX = axis options can be used to set a bias for including one more tick mark outside of either end of the data range (or VIEWMIN/ VIEWMAX range) on a continuous, linear axis. The threshold range is from 0 (do not include the tick mark) to 1 (include the tick mark). The default is 0.30. The bias at the minimum end of the axis is calculated using the THRESHOLDMIN= value and the minimum data value (by default) or the VIEWMIN= value (if set).

The bias at the maximum end of the axis is calculated using the THRESHOLDMAX= value and the maximum data value (by default) or the VIEWMAX= value (if set).

Specifying THRESHOLDMIN=0 and THRESHOLDMAX=0 prevents the tick marks from extending beyond the data range. Specifying THRESHOLDMIN=1 and THRESHOLDMAX=1 ensures that the data range is bounded by tick marks.

The following figure shows how the threshold settings can affect the tick display on an axis. In the figure, 8 is the minimum value for the display and 29 is the maximum value.

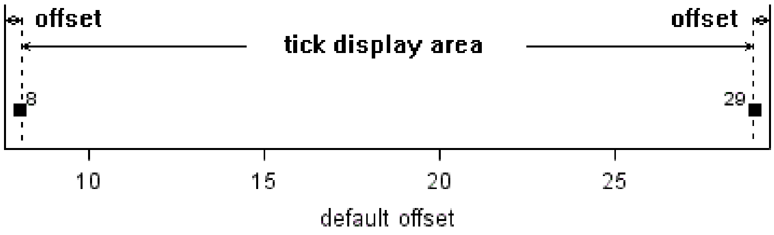

8
29
10
15
20
25
xaxisopts=(linearopts=(thresholdmin=0 thresholdmax=0))
8
29
10
15
20
25
30
xaxisopts=(linearopts=(thresholdmin=0.5 thresholdmax=0.5))
8
29
5
10
15
20
25
30
xaxisopts=(linearopts=(thresholdmin=1 thresholdmax=1))

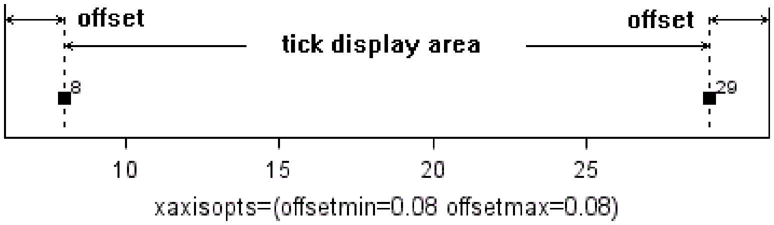
Adjusting Axis Offsets

The OFFSETMIN = and OFFSETMAX = axis options can be used to reserve an area at the minimum end of an axis, the maximum end, or both ends. No tick marks are displayed in the reserved areas.

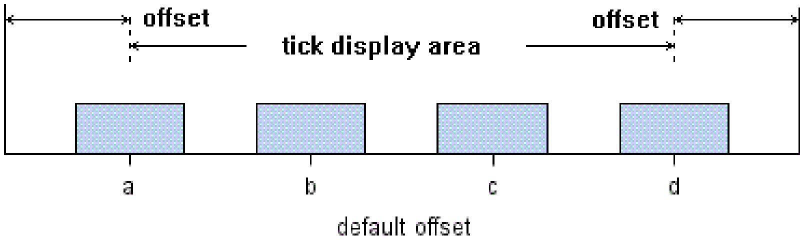
The offset range is from 0 to 1, and the specified value is used to calculate the offset as a percentage of the full axis length. The larger the offset area that is reserved, the less space is available for the tick display area. The default offset reserves just enough area to fully display markers and other graphical features near the ends of an axis.

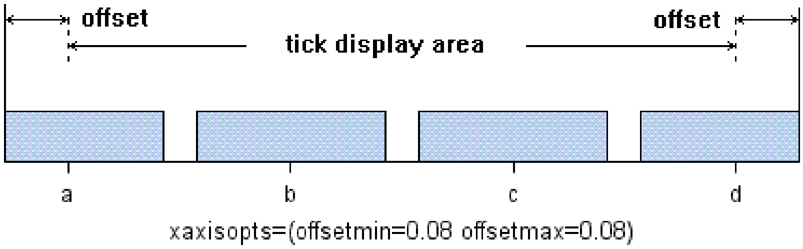
The following figure shows how offset values of 0.08 might compare with the default offsets for a continuous axis.





This next figure shows how offset values might affect the discrete axis of a bar chart.





Chapter 42

Axis Options for LAYOUT OVERLAY

Description and Syntax

Axis options for the plots within an OVERLAY layout are specified with the following options on a LAYOUT OVERLAY statement:

XAXISOPTS= (*axis-options*)

YAXISOPTS= (*axis-options*)

X2AXISOPTS= (*axis-options*)

Y2AXISOPTS= (*axis-options*)

Example Program and Statement Details

Overview

The LAYOUT OVERLAY statement provides the XAXISOPTS=, YAXISOPTS=, X2AXISOPTS=, Y2AXISOPTS= options that enable you to manage the axis display separately for the X, Y, X2, and Y2 axes. The following example template uses the YAXISOPTS= option to manage the grid lines, tick marks, and tick values on a Y axis:

```
begingraph;
  layout overlay /
    yaxisopts=(
      griddisplay=on
      display=(ticks tickvalues)
    );
    seriesplot x=month y=predict;
  endlayout;
endgraph;
```

Within an OVERLAY layout block, each plot axis is always of a particular type. In the default cases, the axis type is always DISCRETE, LINEAR, or TIME. The TYPE= option enables you to specify an axis type that overrides the default. For example, when appropriate for the data, you can request a LOG axis. When you override the default axis type, you must be sure to specify the correct axis type for the plot(s) that you are defining.

Each axis type has features specific to that type, and the following axis options enable you to specify features for the different types: DISCRETEOPTS=, LINEAROPTS=, LOGOPTS= , and TIMEOPTS= . One or more of these options can be specified for an axis, but the specified settings are applied only to the axis type that supports them.

Unless otherwise indicated in an option description, each axis option is available for the X, Y, X2, and Y2 axis.

Interaction. The OVERLAY's axis options are ignored when the LAYOUT OVERLAY statement is nested within another layout type that has external axes in effect. For example, the axis options are ignored when the OVERLAY is nested in a LAYOUT LATTICE with a COLUMNAXIS= or ROWAXIS= option in effect.

General Options for All Axes in an Overlay

The options that are documented in this section can be used with any of the axis types that are supported within an OVERLAY layout. Subsequent sections in the chapter document the axis options that are available only for specific axis types: discrete, linear, log, or time axes.

Statement Option	Description
DISCRETE	Specifies options for a discrete X or X2 axis. This option is not available for a Y or Y2 axis.
DISPLAY	Controls which axis features are displayed on the primary axis.
DISPLAYSECONDARY	Controls which axis features are displayed on the secondary axis.
GRIDATTRS	Specifies the attributes of the grid lines.
GRIDDISPLAY	Specifies whether axis grid lines are displayed.
LABEL	Specifies the axis label.
LABELATTRS	Specifies the color and font attributes of the axis label.
LINEAROPTS	Specifies features for a standard numeric interval axis.
LOGOPTS	Specifies features for a log axis.

Statement Option	Description
OFFSETMAX	Reserves an area at the maximum end of the axis. No tick marks are displayed in the reserved area.
OFFSETMIN	Reserves an area at the minimum end of the axis. No tick marks are displayed in the reserved area.
REVERSE	Specifies whether the tick values should appear in the reverse order.
SHORTLABEL	Specifies an alternate axis label to use if the default or specified axis label is too long for the axis length.
TICKSTYLE	Specifies the placement of tick marks in relation to the axis line.
TICKVALUEATTRS	Specifies the color and font attributes of the axis tick values.
TIMEOPTS	Specifies features for a TIME axis.
TYPE	Specifies the type of axis to use.

DISCRETEOPTS=(*discrete-axis-options*)
specifies one or more options for a discrete X or X2 axis. See "Options for Discrete X or X2 Axes Only" on page 412.

DISPLAY=STANDARD | ALL | NONE | (*display-options*)
controls which axis features are displayed on the primary axis.

Default: STANDARD

STANDARD
specifies that the LABEL, LINE, TICKS, and TICKVALUES are displayed

ALL
specifies that LABEL, LINE, TICKS, and TICKVALUES are displayed

NONE
specifies that no axis features are displayed

(*display-options*)
a list of space-delimited options enclosed in parentheses. The list must include one or more of the following:

LABEL	displays the axis label
LINE	displays the axis line
TICKS	displays the tick marks
TICKVALUES	displays the values that are represented by the major tick marks

The default line attributes for the axis line and axis tick marks are defined in the GraphAxisLine style element.

See GRIDDISPLAY= and GRIDATTRS= for setting axis grid lines.

DISPLAYSECONDARY=NONE | ALL | STANDARD | (*display-options*)

controls which axis features are displayed on the secondary axis. When data are mapped to the X or Y axis but not to the X2 or Y2, you can display X2 or Y2 (secondary) axis using this option. The secondary axis is a duplicate of the X or Y axis (or X2 or Y2, if data are mapped only to X2 or Y2) but can have different display options.

Default: NONE

NONE

specifies that no axis features are displayed

STANDARD

specifies that the LABEL, LINE, TICKS, and TICKVALUES are displayed on the secondary axis

ALL

specifies that LABEL, LINE, TICKS, and TICKVALUES are displayed on the secondary axis

(*display-options*)

a list of space-delimited options enclosed in parentheses. The list must include one or more of the following:

LABEL	displays the axis label
LINE	displays the axis line
TICKS	displays the tick marks
TICKVALUES	displays the values that are represented by the major tick marks

This option is ignored if data are mapped to both X and X2 axes or Y and Y2 axes.

GRIDATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)

specifies the attributes of the grid lines. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Line Options" on page 564 for available *line-options*.

Default: The GraphGridLines style element.

Interaction: This option is ignored if the GRIDDISPLAY= option does not display the grid lines.

GRIDDISPLAY=AUTO_OFF | AUTO_ON | ON | OFF

specifies whether axis grid lines are displayed.

Default: AUTO_OFF

AUTO_OFF

specifies that grid lines are not displayed unless the GraphGridLines element in the current style contains DisplayOpts="ON."

AUTO_ON

specifies that grid lines are displayed unless the GraphGridLines element in the current style contains DisplayOpts="OFF."

ON

specifies that grid lines are always displayed. The current style has no override.

OFF

specifies that grid lines are never displayed. The current style has no override.

This option enables the template to absolutely control the display of grid lines or to allow interaction with the current style to decide whether grid lines are displayed.

Supplied styles use DisplayOpts="AUTO," which means that the style has no "preference" about grid lines and the graphics template setting for grid lines is always used.

Discussion: The following table shows the end results for various combinations of the GRIDDISPLAY= option and the DisplayOpts= attribute of the GraphGridLines style element. Most supplied templates use the default setting AUTO_OFF to indicate a preference for not displaying grid lines, but allowing the style to override.

GRIDDISPLAY= option	DisplayOpts= style attribute	Grid Lines Shown?
AUTO_OFF	AUTO	no
AUTO_OFF	ON	yes
AUTO_OFF	OFF	no
AUTO_ON	AUTO	yes
AUTO_ON	ON	yes
AUTO_ON	OFF	no
ON	any value	yes
OFF	any value	no

LABEL=*"string"* | (*"string"* ...*"string"*)

specifies the axis label. The *string* can be either a string literal or a dynamic. The list form implies that all included string literals or dynamics will be concatenated.

Default: The default label is derived from the primary plot in the layout. For more information, see "When Plots Share Data and a Common Axis" on page 397.

Interaction: If the axis label is too long to fit along the axis, it is truncated by default. Use the SHORTLABEL= option to specify an alternate axis label to be used whenever truncation would normally occur.

Interaction: This option is ignored if the DISPLAY= option does not display the axis label.

LABELATTRS=*style-element* | *style-element* (*text-options*) | (*text-options*)

specifies the color and font attributes of the axis label. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Text Options" on page 566 for available *text-options*.

Default: The GraphLabelText style element.

Interaction: This option is ignored if the DISPLAY= option does not display the axis label.

LINEAROPTS=(*linear-axis-options*)

specifies one or more *linear-axis-options* for a numeric interval axis. See "Options for Linear Axes Only" on page 412.

LOGOPTS=(*log-axis-options*)

specifies one or more *log-axis-options* for a log axis. See "Options for Log Axes Only" on page 417.

OFFSETMAX=AUTO | AUTOCOMPRESS | *number*

reserves an area at the maximum end of the axis. No tick marks are displayed in the reserved area. For more information, see “Adjusting Axis Offsets” on page 403.

Default: AUTO

Range: 0 - 1. The sum of OFFSETMAX= and OFFSETMIN= should not be more than 1.

AUTO

Reserves just enough area to fully display markers and other graphical features near the maximum end of an axis.

AUTOCOMPRESS

Automatic offset applied that prevents axis labels and tick values from extending beyond the axis length.

number

The offset is expressed as a decimal proportion of the full axis length.

OFFSETMIN=AUTO | AUTOCOMPRESS | *number*

reserves an area at the minimum end of the axis. No tick marks are displayed in the reserved area. For more information, see “Adjusting Axis Offsets” on page 403.

Default: AUTO

Range: 0 - 1. The sum of OFFSETMAX= and OFFSETMIN= should not be more than 1.

AUTO

Reserves just enough area to fully display markers and other graphical features near the maximum end of an axis.

AUTOCOMPRESS

Automatic offset applied that prevents axis labels and tick values from extending beyond the axis length.

number

The offset is expressed as a decimal proportion of the full axis length.

REVERSE=*boolean*

specifies whether tick values should appear in the reverse order.

Default: FALSE

SHORTLABEL=*"string"*

specifies an alternate axis label. This label is used when the default axis label or label specified by the LABEL= option is too long for the axis length.

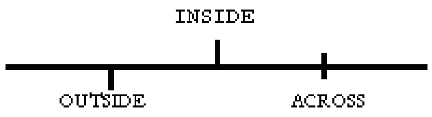
Default: no default

Interaction: This option is ignored if the DISPLAY= option does not display the axis label.

If the specified label is itself too long for the axis, it is truncated in the display.

TICKSTYLE=OUTSIDE | INSIDE | ACROSS

specifies the placement of tick marks in relation to the axis line. The figure shows the tick display on an X axis.



Default: The GraphAxisLines:TickDisplay style reference.

OUTSIDE
displays tick marks outside of the axis frame.

INSIDE
displays tick marks inside of the axis frame.

ACROSS
displays tick marks across the axis line.

Interaction: This option is ignored if the DISPLAY= option does not display tick marks.

This option has no affect on the placement of the tick values, which are always outside the axis frame.

This option applies to both major ticks and minor ticks.

TICKVALUEATTRS=*style-element* | *style-element* (*text-options*) | (*text-options*)
specifies the color and font attributes of the axis tick values. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Text Options" on page 566 for available *text-options*.

Default: The GraphValueText style element.

Interaction: This option is ignored if the DISPLAY= option does not display tick values.

TIMEOPTS=(*time-axis-options*)
specifies one or more *time-axis-options* for a time axis. See "Options for Time Axes Only" on page 419.

TYPE=AUTO | DISCRETE | LINEAR | TIME | LOG
specifies the type of axis to use.

Default: AUTO

AUTO
Requests that the axis type be automatically determined, based on the overlay contents. For more information, see "When Plots Share Data and a Common Axis" on page 397.

DISCRETE
Use a DISCRETE axis if possible. The data for discrete axes can be character or numeric. For an X or X2 axis, you can add a DISCRETEOPTS=( ) option list to customize this axis type. DISCRETEOPTS= is not supported on a Y or Y2 axis.

LINEAR
Use a LINEAR axis if possible. You can add a LINEAROPTS=( ) option list to customize this axis type.

TIME
Use a TIME axis if possible. Data for this axis must be SAS time, SAS date, or SAS datetime values. You can add a TIMEOPTS= ( ) option list to customize this axis type.

LOG
Use a LOG axis if possible. You can add a LOGOPTS= ( ) option list to customize this axis type.

Interaction: If this option is set to anything other than AUTO, plots within the layout are dropped from the display if their data types or data ranges do not match the axis type requirements. For more information, see "Plot Axis Types Must Agree on Common Axes" on page 400.

Interaction: After the axis type is determined (whether you set a specific type or AUTO is in effect), only options supported by that axis type can be used. For example, if TYPE=TIME, only the general OVERLAY axis options and those available on TIMEOPTS= are supported.

Options for Discrete X or X2 Axes Only

DISCRETEOPTS=(*discrete-axis-options*)

specifies one or more options for a discrete X or X2 axis. This option is not available for a Y or Y2 axis. Options must be enclosed in parentheses. Each option is specified as a *name* = *value* pair and each pair is space separated.

Interaction: This option is ignored if the axis type is not DISCRETE.

TICKVALUEFITPOLICY=ROTATE | *other-policy*

specifies a policy for avoiding tick value collision on an X or X2 axis.

Default: ROTATE

The collision-avoidance policy that is most effective depends on the number of tick values, their length, and the length of the axis. The policy can be one of the following:

ROTATE	Tick values are rotated 45 degrees.
ROTATETHIN	Attempt ROTATE policy and then THIN policy.
STAGGER	Tick values alternate between two rows.
STAGGERROTATE	Attempt STAGGER policy and then ROTATE policy.
STAGGERTHIN	Attempt STAGGER policy and then THIN policy.
STAGGERTRUNCATE	Attempt STAGGER policy and then TRUNCATE policy.
TRUNCATE	Tick values are shortened when they exceed a certain number of characters.
TRUNCATEROTATE	Attempt TRUNCATE policy and then ROTATE policy.
TRUNCATESTAGGER	Attempt TRUNCATE policy and then STAGGER policy.
TRUNCATETHIN	Attempt TRUNCATE policy and then THIN policy.
THIN	Some tick values are removed.

Options for Linear Axes Only

LINEAROPTS=(*linear-axis-options*)

specifies one or more linear-axis-options for a numeric interval axis. Options must be enclosed in parentheses. Each option is specified as a *name* = *value* pair and each pair is space separated.

Interaction: This option is ignored if the axis type is not LINEAR.

Linear-Axis-Option	Description
INTEGER	Specifies that evenly spaced integer values are used for tick marks.
ORIGIN	Specifies that the axis perpendicular to the current axis be drawn at the indicated data value.
THRESHOLDMAX	Specifies a bias for including one more tick mark at the maximum end of the axis.
THRESHOLDMIN	Specifies a bias for including one more tick mark at the minimum end of the axis.
TICKVALUEFITPOLICY	Specifies a policy for avoiding tick value collision. Only the default policy (THIN) is available for a Y or Y2 axis.
TICKVALUEFORMAT	Specifies how to format the values for major tick marks.
TICKVALUELIST	Specifies the order of the tick values for a linear axis as list.
TICKVALUEPRIORITY	Specifies whether an axis tick specification can extend the axis data range.
TICKVALUESEQUENCE	Specifies the tick values for a linear axis by start, end, and increment.
VIEWMAX	Specifies the maximum data value to include in the display.
VIEWMIN	Specifies the minimum data value to include in the display.

INTEGER=*boolean*

specifies that evenly spaced integer values are used for tick marks.

Default: FALSE

Interaction: This option is overridden by the TICKVALUELIST= or TICKVALUESEQUENCE= option.

Interaction: This option overrides the MAXDECIMALS= and PREFERREDDECIMALS= suboptions of the TICKVALUEFORMAT= option

ORIGIN=*number*

specifies that the axis perpendicular to the current axis be drawn at the indicated data value.

Default: no default

Interaction: If the specified value is outside the data range for the current axis, the data range is extended to include the value.

The axis line, ticks, and tick values of the “perpendicular” axis move to the location indicated by the origin. The axis label is not moved.

This option is often used to create Cartesian axes (axes centered at ORIGIN=0).

Discussion: For managing origin settings, the GTL treats the X and Y axes as a pair, and the X2 and Y2 axes as a separate pair. Thus, if you set the Y-axis origin to 200, the X axis is drawn from that origin point. If the graph also displays an X2 axis, it is unaffected and does not move. Similarly, if you set an origin for the Y2 axis, the X2 axis moves to that origin point and the X axis is unaffected.

If you set an origin for the Y2 axis and there is no X2 axis, then the origin setting for Y2 does not affect the graph display. That is, the X axis does not move to that origin point.

If you set an origin for an axis and the axis has a tick value at that origin value, the tick value is not displayed. Suppressing the tick value at the origin prevents the value from colliding with the axis value on the "perpendicular" axis. However, it is possible that the tick values on the orthogonal axes will collide.

THRESHOLDMAX= *number*
specifies a bias for including one more tick mark at the maximum end of the axis. For more information, see “Adjusting Axis Thresholds” on page 402.

Default: .30

Range: 0 - 1

Interaction: This option is ignored if the TICKVALUELIST= or TICKVALUESEQUENCE= option is used.

THRESHOLDMIN= *number*
specifies a bias for including one more tick mark at the minimum end of the axis. For more information, see “Adjusting Axis Thresholds” on page 402.

Default: .30

Range: 0 - 1

Interaction: This option is ignored if the TICKVALUELIST= or TICKVALUESEQUENCE= option is used.

TICKVALUEFITPOLICY=THIN | *other-policy*
specifies a policy for avoiding tick value collision. Only the default setting (THIN) is available for a Y or Y2 axis.

Default: THIN

The collision-avoidance policy that is most effective depends on the number of tick values, their length, and the length of the axis. For an X or X2 axis, the policy can be one of the following:

THIN
Eliminate alternate tick values.

ROTATE
Tick values are rotated 45 degrees.

ROTATETHIN
Attempt ROTATE policy and then THIN policy.

STAGGER
Tick values alternate between two rows.

STAGGERROTATE
Attempt STAGGER policy and then ROTATE policy.

STAGGERTHIN
Attempt STAGGER policy and then THIN policy.

TICKVALUEFORMAT=(*format-options*) | DATA | *format*

specifies how to format the values for major tick marks.

Default: (MAXWIDTH=8, MAXDECIMALS=6, PREFERREDDECIMALS=2, EXTRACTSCALE=FALSE)

(*format-options*)

specifies one or more major tick value formatting options. Together, these options provide parameters for determining an optimal format (*w.d*, E*w*., BEST*w*.) for displaying major tick values.

MAXWIDTH = *integer*	Specifies the maximum width for displayed tick values. Values might be rounded or converted to E-notation to fit into this width.
MAXDECIMALS = *integer*	Specifies the maximum number of decimals for displayed tick values. Values might be rounded or converted to E-notation to fit into this width. MAXWIDTH > MAXDECIMALS
PREFERREDDECIMALS = *integer*	Specifies the number of decimal places desired for most values. The actual number might vary based on other constraints.
EXTRACTSCALE = *boolean*	Specifies whether to extract some factor of ten from all tick values in order to reduce the overall width of the tick values and improve legibility. The extracted factor is concatenated to the existing axis label. Example of revised axis label: *Number Sold (x1000)*

DATA

use the format that has been assigned to the variable contributing to the axis (or BEST6. if no format is assigned) to control the formatting of the major tick values.

format

specifies a format to apply to the major tick values.

Restriction: The *format-options* are used only when no format has been assigned to the variable(s) that contribute to the axis or when the assigned format(s) are *w.d*, E*w*. or BEST*w*.

Restriction: GTL currently honors most but not every SAS format. For details, see "SAS Formats Not Supported" on page 569.

Interaction: If the axis label is not displayed, the EXTRACTSCALE=TRUE *format-option* is ignored.

TICKVALUELIST=(*numeric-list*)

specifies the tick values for a linear axis as list.

Default: An internal algorithm determines the tick marks based on the actual axis data range or the data range established by the VIEWMIN= and VIEWMAX= options. By default when this option is used, the only tick values that appear are the tick values in *numeric-list* that fall within the explicit data range (set by VIEWMIN= and VIEWMAX=) or implicit data range (set by the actual data minimum and data maximum).

Requirement: The *numeric-list* must be enclosed in parentheses and each item separated by a blank.

Interaction: This option overrides the INTEGER= option.

Interaction: This option is ignored if the TICKVALUESEQUENCE= option is specified, or if the DISPLAY= option does not display tick values.

Interaction: The VIEWMIN= and VIEWMAX= options alter the axis data range. If the VIEWMIN= option is set to the minimum tick list value and the VIEWMAX= option is set to the maximum tick list value, all ticks in the tick list are displayed.

Interaction: If the TICKVALUEPRIORITY= is set to TRUE, 1) the VIEWMIN= and VIEWMAX= options are ignored if they are fully enclosed by the *numeric-list* 2) the tick *numeric-list* can extend the implicit data range of the axis, but never reduce it.

The values in the list are formatted according to the setting for the TICKVALUEFORMAT= option.

TICKVALUEPRIORITY=*boolean*
specifies whether an axis tick specification (TICKVALUELIST= or TICKVALUESEQUENCE=) can extend the axis data range.

Default: FALSE

FALSE
the only tick values that appear are the user-specified tick values that fall within the explicit data range (set by VIEWMIN= and VIEWMAX=) or implicit data range (set by the actual data minimum and data maximum).

TRUE
the axis data range can be extended (but not reduced) to include the minimum and maximum specified by either the TICKVALUELIST= or TICKVALUESEQUENCE= option. If minimum and maximum of the user-specified values are within the data range, this option has no effect.

Interaction: When this option is set to TRUE, the VIEWMIN= and VIEWMAX= options are ignored if they are fully enclosed by value range set by the TICKVALUELIST= or TICKVALUESEQUENCE= option.

Interaction: This option is ignored if the DISPLAY= option does not display tick marks.

TICKVALUESEQUENCE=(*sequence-options*)
specifies the tick values for a linear axis by start, end, and increment.

Default: An internal algorithm determines the tick marks based on the actual axis data range or the data range established by the VIEWMIN= and VIEWMAX= options. By default when this option is used, the only tick values that appear are those that fall within the explicit data range (set by VIEWMIN= and VIEWMAX=) or implicit data range (set by the actual data minimum and data maximum).

(sequence-options)
Three settings that control major tick values. All three options must be provided.

START = *number*	Specifies the value for the first tick mark.
END = *number*	Specifies the value for the last tick mark.
INCREMENT = *number*	Specifies the increment for intermediate tick marks between the first and last tick marks. The END value always controls the last tick mark. The interval between the last tick mark and the previous tick

mark might not necessarily be the INCREMENT value.

Interaction: This option overrides the INTEGER= option.

Interaction: The VIEWMIN= and VIEWMAX= options alter the axis data range. If the VIEWMIN= option is set to the START= option value and the VIEWMAX= option is set to the END= option value, all ticks in the tick sequence are displayed.

Interaction: If the TICKVALUEPRIORITY= option is set to TRUE, the tick sequence might extend the explicit data range of the axis, but never reduce it.

Interaction: This option is ignored if the DISPLAY= option does not display tick marks.

The values in the sequence are formatted according to the setting for the TICKVALUEFORMAT= option.

See also: TICKVALUELIST= option as an alternative for customizing tick marks.

VIEWMAX= *number*

specifies the maximum data value to include in the display (the value might be adjusted by the threshold calculation). For more information, see “Adjusting the Axis View” on page 402.

Default: The maximum value in the data for the specified axis.

Interaction: This option does not determine the maximum axis tick value displayed. The THRESHOLDMAX= value is used to determine the maximum tick value.

Interaction: This option has no effect if the TICKVALUEPRIORITY= is set to TRUE.

Setting a VIEWMAX= or VIEWMIN= value does not alter the original data or any calculations on it.

VIEWMIN= *number*

specifies the minimum data value to include in the display (the value might be adjusted by the threshold calculation). For more information, see “Adjusting the Axis View” on page 402.

Default: The minimum value in the data for the specified axis.

Interaction: This option does not determine the minimum axis tick value displayed. The THRESHOLDMIN= value is used to determine the maximum tick value.

Interaction: This option has no effect if the TICKVALUEPRIORITY= is set to TRUE.

Setting a VIEWMAX= or VIEWMIN= value does not alter the original data or any calculations on it.

Options for Log Axes Only

LOGOPTS=(*log-axis-options*)

specifies one or more *log-axis-options* for a log axis. Options must be enclosed in parentheses. Each option is specified as a *name* = *value* pair and each pair is space separated.

Interaction: This option is ignored if the axis type is not LOG.

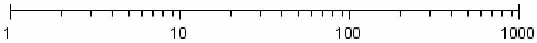
Log-Axis-Option	**Description**
BASE	Specifies the base of the logarithmic scale for the axis values.

Log-Axis-Option	Description
MINORTICKS	Specifies whether minor ticks are displayed.
TICKINTERVALSTYLE	Specifies how to scale and format the values for major tick marks.
VIEWMAX	Specifies the maximum data value to include in the display.
VIEWMIN	Specifies the minimum data value to include in the display.

BASE=10 | 2 | E
specifies the base of the logarithmic scale for the axis values.

Default: 10

MINORTICKS=*boolean*
specifies whether minor ticks are displayed:



Default: FALSE

Restriction: Minor ticks can be displayed only when BASE=10 and TICKINTERVALSTYLE= is LOGEXPAND or LOGEXPONENT.

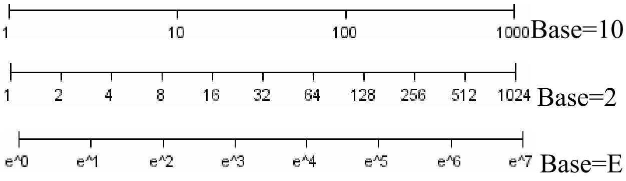
If this option is set to TRUE, the number of minor ticks is automatically determined.

TICKINTERVALSTYLE=AUTO | LOGEXPAND | LOGEXPONENT | LINEAR
specifies how to scale and format the values for major tick marks.

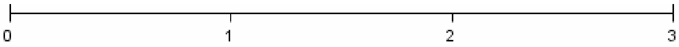
Default: AUTO

AUTO
A LOGEXPAND, LOGEXPONENT, or LINEAR representation is chosen automatically based on the range of the data. When the data range is small (within an order of magnitude), a LINEAR representation is typically used. Data ranges that encompass several orders of magnitude typically use the LOGEXPAND or LOGEXPONENT representation.

LOGEXPAND
Major ticks are placed at uniform intervals at integer powers of the base. The tick values are expanded as follows:

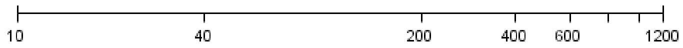


LOGEXPONENT
Major ticks are placed at uniform intervals at integer powers of the base. The tick values are only the integer exponents for all bases.



LINEAR

Major tick marks are placed at non-uniform intervals that cover the range of the data.



Restriction: For LOGEXPONENT, formats on data columns contributing to the axis are ignored. For LOGEXPAND, formats on data columns contributing to the axis are ignored, although any "named format" on the column is retained. For LINEAR, ticks values are automatically formatted when the column format is not assigned or one of w.d, Ew. or BESTw. Other formats (SAS defined or user-defined) are used if specified.

Restriction: GTL currently honors most but not every SAS format. For details, see "SAS Formats Not Supported" on page 569.

Interaction: When BASE=10 and LOGEXPAND or LOGEXPONENT is used, an intermediate tick is added whenever the axis data range is less than or equal to 1.5 powers of 10.

If you use TICKINTERVALSTYLE=LOGEXPONENT, you might want to include information in the axis label about which base is used.

VIEWMAX= *number*

specifies the maximum data value to include in the display. For more information, see "Adjusting the Axis View" on page 402.

Default: The maximum value in the data for the specified axis.

Setting a VIEWMAX= or VIEWMIN= value does not alter the original data or any calculations on it.

VIEWMIN= *number*

specifies the minimum data value to include in the display. For more information, see "Adjusting the Axis View" on page 402.

Default: The minimum value in the data for the specified axis.

Setting a VIEWMAX= or VIEWMIN= value does not alter the original data or any calculations on it.

Options for Time Axes Only

TIMEOPTS=(*time-axis-options*)

specifies one or more *time-axis-options* for a time axis.

Variables associated with a time axis must be in SAS time, SAS date, or SAS datetime units and have an associated SAS time, date, or datetime format.

Options must be enclosed in parentheses. Each option is specified as a *name = value* pair and each pair is space separated.

Interaction: This option is ignored if the axis type is not TIME.

Time-Axis-Option	Description
INTERVAL	Specifies the time interval between major tick marks.
MINORTICKS	Specifies whether minor ticks are displayed.

Time-Axis-Option	Description
SPLITTICKVALUE	Specifies whether to split the tick values on an X or X2 axis, if possible. This option is not available on a Y or Y2 axis.
TICKVALUEFITPOLICY	Specifies a policy for avoiding tick value collision on an X or X2 axis. This option is not available on a Y or Y2 axis.
TICKVALUEFORMAT	Specifies how to format the values for major tick marks.
TICKVALUELIST	Specifies the order of the tick values for a time axis as list.
TICKVALUEPRIORITY	Specifies whether an axis tick specification can extend the axis data range.
VIEWMAX	Specifies the maximum data value to include in the display.
VIEWMIN	Specifies the minimum data value to include in the display.

INTERVAL=*interval*

specifies the time interval between major ticks. Valid *interval* keywords are AUTO, SECOND, MINUTE, HOUR, DAY, TENDAY, WEEK, SEMIMONTH, MONTH, QUARTER, SEMIYEAR, YEAR.

The data column(s) mapped to a time axis must be in the same duration units: TIME, DATE, or DATETIME. The selection of an interval must be consistent with the duration unit. For example, if the data are in time units, you can select only AUTO, SECOND, MINUTE, HOUR.

Default: AUTO. An appropriate interval is chosen based on the data and the column date/datetime/time format.

Interaction: This option is ignored if the TICKVALUELIST= option is used.

INTERVAL	Unit	Tick interval	Default tick value format
AUTO	DATE, TIME, or DATETIME	automatically chosen	automatically chosen
SECOND	TIME or DATETIME	second	TIME8.
MINUTE	TIME or DATETIME	minute	TIME8.
HOUR	TIME or DATETIME	hour	TIME8.

INTERVAL	Unit	Tick interval	Default tick value format
DAY	DATE or DATETIME	day	DATE9.
TENDAY	DATE or DATETIME	10 days	DATE9.
WEEK	DATE or DATETIME	7 days	DATE9.
SEMIMONTH	DATE or DATETIME	1st and 16th of each month	DATE9.
MONTH	DATE or DATETIME	month	MONYY7.
QUARTER	DATE or DATETIME	3 months	YYQC6.
SEMIYEAR	DATE or DATETIME	6 months	MONYY7.
YEAR	DATE or DATETIME	year	YEAR4.

MINORTICKS=*boolean*

specifies whether minor ticks are displayed.

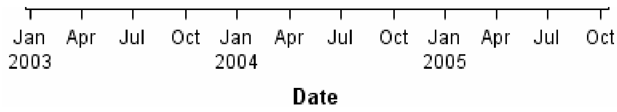
Default: FALSE

Interaction: The number of minor ticks is dependent on the value of the INTERVAL= option.

Interaction: This option is ignored if the DISPLAY= option does not display tick marks, or if the TICKVALUELIST= option is used.

SPLITTICKVALUE=*boolean*

specifies whether to split the tick values on an X or X2 axis, if possible. This option is not available for a Y or Y2 axis. For example, with INTERVAL= MONTH, this is how tick values are split:



Default: TRUE

TRUE

axis tick values are split into two lines allowing more tick values to appear

FALSE

typically, fewer tick values fit, causing thinning, rotation, or staggering of the values. See the TICKVALUEFITPOLICY= option.

Interaction: This option is ignored if the TICKVALUELIST= or TICKVALUEFORMAT= option is used.

TICKVALUEFITPOLICY=THIN | ROTATE | ROTATETHIN | STAGGER | STAGGERROTATE | STAGGERTHIN

specifies a policy for avoiding tick value collision on an X or X2 axis. This option is not available for a Y or Y2 axis.

Default: THIN

The collision-avoidance policy that is most effective depends on the number of tick values, their length, and the length of the axis. For an X or X2 axis, the policy can be one of the following:

THIN	Some tick values are removed.
ROTATE	Tick values are rotated 45 degrees.
ROTATETHIN	Attempt ROTATE policy and then THIN policy.
STAGGER	Tick values alternate between two rows.
STAGGERROTATE	Attempt STAGGER policy and then ROTATE policy.
STAGGERTHIN	Attempt STAGGER policy and then THIN policy.

Interaction: When SPLITTICKVALUE= TRUE, this option is ignored and only the THIN policy is used.

TICKVALUEFORMAT=*format* | DATA

specifies how to format the values for major tick marks.

Default: The default format used by the INTERVAL= option. The default does not apply if TICKVALUELIST= is specified.

format

A SAS date, time, or datetime format to control how the major tick values are displayed. This format must be in the same duration units as the data column(s) mapped to a time axis: TIME, DATE, or DATETIME and should be appropriate for the value of the INTERVAL= option. For example, if INTERVAL=MONTH and there are two years of data displayed on the axis, choosing TICKVALUEFORMAT=YEAR. would result in several ticks having the same year value.

DATA

Specifies that the SAS date, time, or datetime format associated with the data column assigned to the axis be used to control how the major tick values are displayed.

Restriction: GTL currently honors most but not every SAS format. For details, see "SAS Formats Not Supported" on page 569.

Interaction: If this option is specified, the SPLITTICKVALUE= option is ignored.

TICKVALUELIST=(*time-constant-list* | *date-constant-list* | *datetime-constant-list* | *numeric-list*)

specifies the tick values for a time axis as list.

Default: An internal algorithm determines the tick values.

Requirement: The list must be enclosed in parentheses and each item separated by a blank. The items in the list must be in the same duration units as the data mapped to the axis: TIME, DATE, or DATETIME. The values can be expressed as SAS TIME, DATE, or DATETIME constants (for example, "13:23"T, "11MAY06"D, or "11MAY06:13:23"DT) or their numeric equivalents.

Restriction: If TICKVALUEPRIORITY= is set to FALSE, this option does not extend the data range of the axis. If the values fall within the default data range or that specified by the VIEWMIN= or VIEWMAX= options, they are used.

The values in the list are formatted according to the format specified on the TICKVALUEFORMAT= option. If TICKVALUEFORMAT= is not used, the values are formatted according to the column format (the default TICKVALUEFORMAT value is not applied to these values).

TICKVALUEPRIORITY=*boolean*

specifies whether an axis tick specification (TICKVALUELIST=) can extend the axis data range.

Default: FALSE

FALSE

the only tick values that appear are the user-specified tick values that fall within the explicit data range (set by VIEWMIN= and VIEWMAX=) or implicit data range (set by the actual data minimum and data maximum).

TRUE

the axis data range might be extended (but not reduced) to include the minimum and maximum specified by the TICKVALUELIST= option. If minimum and maximum of the user-specified values are within the data range, this option has no effect.

Interaction: When this option is set to TRUE, the VIEWMIN= or VIEWMAX= options might have no effect.

Interaction: This option is ignored if the DISPLAY= option does not display tick marks.

VIEWMAX= *number*

specifies the maximum data value to include in the display. For more information, see “Adjusting the Axis View” on page 402.

Default: The maximum value in the data for the specified axis.

Setting a VIEWMAX= or VIEWMIN= value does not alter the original data or any calculations on it.

VIEWMIN= *number*

specifies the minimum data value to include in the display. For more information, see “Adjusting the Axis View” on page 402.

Default: The minimum value in the data for the specified axis.

Setting a VIEWMAX= or VIEWMIN= value does not alter the original data or any calculations on it.

Chapter 43

Axis Options for LAYOUT OVERLAY3D

Description and Syntax

Axis options for the plots within an OVERLAY3D layout are specified with the following options on a LAYOUT OVERLAY3D statement:

XAXISOPTS= (*axis-options*)

YAXISOPTS= (*axis-options*)

ZAXISOPTS= (*axis-options*)

Example Program and Statement Details

Overview

The LAYOUT OVERLAY3D statement provides XAXISOPTS=, YAXISOPTS=, and ZAXISOPTS= options that enable you to manage the axis display separately for the X, Y, and Z axes. The following example template uses the YAXISOPTS= option to manage the display of grid lines, tick marks, and tick values on a Y axis:

```
begingraph;
  layout overlay3d /
    yaxisopts=(
      griddisplay=on
      display=(ticks tickvalues)
    );
    bihistogram3dparm x=rater y=customer z=percent;
  endlayout;
endgraph;
```

Within an OVERLAY3D layout block, each plot axis is always either a linear or a time axis. The default axis-type setting is AUTO, which specifies that the axis type be automatically determined by the plot or the overlay contents. The TYPE= option enables you to explicitly specify either a LINEAR or a TIME axis type. When you override the default axis type, you must be sure to specify the correct axis type for the plot(s) that you are defining.

Each axis type has features specific to that type, and the axis options LINEAROPTS= and TIMEOPS= enable you to specify features for a linear or a time axis. You can combine one or more general axis options with the options for the specific axis type. However, specified settings are applied only to the axis type that supports them. For example, if you specify general axis options with time axis options and the generated graph produces a linear axis type, the time axis options are ignored.

Unless otherwise indicated in an option description, each axis option is available for the X, Y, and Z axis.

General Options for All Axes in an Overlay3D

The options that are documented in this section can be used with either axis type that is supported within an OVERLAY3D layout. Subsequent sections in the chapter document the axis options that are available only for the specific axis type: linear or time.

Statement Option	Description
DISPLAY	Controls which axis features are displayed.
GRIDATTRS	Specifies the attributes of the grid lines.
GRIDDISPLAY	Specifies whether axis grid lines are displayed.
LABEL	Specifies the axis label.
LABELATTRS	Specifies the color and font attributes of the axis label.
LINEAROPTS	Specifies options for a standard numeric interval axis.
OFFSETMAX	Reserves an area at the maximum end of the axis. No tick marks are displayed in the reserved area.
OFFSETMIN	Reserves an area at the minimum end of the axis. No tick marks are displayed in the reserved area.
TICKVALUEATTRS	Specifies the color and font attributes of the axis tick values.
TIMEOPTS	Specifies options for a TIME axis.
TYPE	Specifies the type of axis to use.

DISPLAY=STANDARD | ALL | NONE | (*display-options*)
controls which axis features are displayed.

Default: STANDARD

STANDARD
specifies that the LABEL, LINE, TICKS, and TICKVALUES are displayed

ALL
specifies that LABEL, LINE, TICKS, and TICKVALUES are displayed

NONE
specifies that no axis features are displayed

(*display-options*)
a list of space-delimited options, enclosed in parentheses. The list must include one or more of the following:

LABEL	displays the axis label
LINE	displays the axis line
TICKS	displays the tick marks
TICKVALUES	displays the values that are represented by the major tick marks

The default line attributes for the axis line and axis tick marks are defined in the GraphAxisLine style element.

See GRIDDISPLAY= and GRIDATTRS= for setting axis grid lines.

GRIDATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)
specifies the attributes of the grid lines. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Line Options" on page 564 for available *line-options*.

Default: The GraphGridLines style element.

Interaction: This option is ignored if the GRIDDISPLAY= option does not display the grid lines.

GRIDDISPLAY=AUTO_OFF | AUTO_ON | ON | OFF
specifies whether axis grid lines are displayed.

Default: AUTO_OFF

AUTO_OFF
specifies that grid lines are not displayed unless the GraphGridLines element in the current style contains DisplayOpts="ON."

AUTO_ON
specifies that grid lines are displayed unless the GraphGridLines element in the current style contains DisplayOpts="OFF."

ON
specifies that grid lines are always displayed. The current style has no override.

OFF
specifies that grid lines are never displayed. The current style has no override.

This option enables the template to absolutely control the display of grid lines or to allow interaction with the current style to decide whether grid lines are displayed.

Supplied styles use DisplayOpts="AUTO," which means that the style has no "preference" about grid lines and the graphics template setting for grid lines is always used.

Discussion: The following table shows the end results for various combinations of the GRIDDISPLAY= option and the DisplayOpts= attribute of the GraphGridLines style element. Most supplied templates use the default setting AUTO_OFF to indicate a preference for not displaying grid lines, but allowing the style to override.

GRIDDISPLAY= option	DisplayOpts= style attribute	Grid Lines Shown?
AUTO_OFF	AUTO	no
AUTO_OFF	ON	yes
AUTO_OFF	OFF	no
AUTO_ON	AUTO	yes
AUTO_ON	ON	yes
AUTO_ON	OFF	no
ON	any value	yes
OFF	any value	no

LABEL=*"string"* | (*"string"* ... *"string"*)
specifies the axis label. The *string* can be either a string literal or a dynamic. The list form implies that all included string literals or dynamics will be concatenated.

Default: The default label is derived from the primary plot in the layout. For more information, see “When Plots Share Data and a Common Axis” on page 397.

Interaction: If the axis label is too long to fit along the axis, it is truncated.

Interaction: This option is ignored if the DISPLAY= option does not display the axis label.

LABELATTRS=*style-element* | *style-element* (*text-options*) | (*text-options*)
specifies the color and font attributes of the axis label. See “General Syntax for Attribute Options” on page 563 for the syntax on using a *style-element* and “Text Options” on page 566 for available *text-options*.

Default: The GraphLabelText style element.

Interaction: This option is ignored if the DISPLAY= option does not display the axis label.

LINEAROPTS=(*linear-axis-options*)
specifies one or more *linear-axis-options* for a numeric interval axis. See “Options for Linear Axes Only” on page 430.

OFFSETMAX=AUTO | *number*
reserves an area at the maximum end of the axis. No tick marks are displayed in the reserved area. For more information, see “Adjusting Axis Offsets” on page 403.

Default: AUTO

Range: 0 - 1. The sum of OFFSETMAX= and OFFSETMIN= should not be more than 1.

AUTO

Reserves just enough area to fully display markers and other graphical features near the maximum end of an axis.

number

The offset is expressed as a decimal proportion of the full axis length. For a continuous axis, the offset follows the highest data value or highest tick value, whichever is greater.

OFFSETMIN=AUTO | *number*

reserves an area at the minimum end of the axis. No tick marks are displayed in the reserved area. For more information, see "Adjusting Axis Offsets" on page 403.

Default: AUTO

Range: 0 - 1. The sum of OFFSETMAX= and OFFSETMIN= should not be more than 1.

AUTO

Reserves just enough area to fully display markers and other graphical features near the minimum end of an axis.

number

The offset is expressed as a decimal proportion of the full axis length. For a continuous axis, the offset precedes the lowest data value or lowest tick value, whichever is less.

TICKVALUEATTRS=*style-element* | *style-element* (*text-options*) | (*text-options*)

specifies the color and font attributes of the axis tick values. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Text Options" on page 566 for available *text-options*.

Default: The GraphValueText style element.

Interaction: This option is ignored if the DISPLAY= option does not display tick values.

TIMEOPTS=(*time-axis-options*)

specifies one or more *time-axis-options* for a time axis. See "Options for Time Axes Only" on page 432.

TYPE=AUTO | LINEAR | TIME

specifies the type of axis to use.

Default: AUTO

AUTO

Requests that the axis type be automatically determined by the plot or the overlay contents.

LINEAR

Use a LINEAR axis if possible. You can add a LINEAROPTS= ( ) option list to customize this axis type.

TIME

Use a TIME axis if possible. Data for this axis must be SAS time, SAS date, or SAS datetime values. You can add a TIMEOPS= ( ) option list to customize this axis type.

Interaction: If this option is set to anything other than AUTO, plots within the layout are dropped from the display if their data types or data ranges do not match the axis type requirements. For more information, see "Plot Axis Types Must Agree on Common Axes" on page 400.

Interaction: After the axis type is determined (whether you set a specific type or AUTO is in effect), only options supported by that axis type can be used. For example, if TYPE=TIME, only the general OVERLAY3D axis options and those available on TIMEOPS= are supported.

Options for Linear Axes Only

LINEAROPTS=(*linear-axis-options*)

specifies one or more options for a numeric interval axis. Options must be enclosed in parentheses. Each option is specified as a *name* = *value* pair and each pair is space separated.

Interaction: This option is ignored if the axis type is not LINEAR.

Linear-Axis-Option	Description
INTEGER	Specifies that evenly spaced integer values are used for tick marks.
THRESHOLDMAX	Specifies a bias for including one more tick mark at the maximum end of the axis.
THRESHOLDMIN	Specifies a bias for including one more tick mark at the minimum end of the axis.
TICKVALUEFORMAT	Specifies how to format the values for major tick marks.
TICKVALUELIST	Specifies the order of the tick values for a linear axis as list.
TICKVALUESEQUENCE	Specifies the tick values for a linear axis by start, end, and increment.

INTEGER=*boolean*

specifies that evenly spaced integer values are used for tick marks.

Default: FALSE

Interaction: This option is overridden by the TICKVALUELIST= or TICKVALUESEQUENCE= option.

Interaction: This option overrides the MAXDECIMALS= and PREFERREDDECIMALS= suboptions of the TICKVALUEFORMAT= option

THRESHOLDMAX= *number*

specifies a bias for including one more tick mark at the maximum end of the axis. For more information, see "Adjusting Axis Thresholds" on page 402.

Default: .30

Range: 0 - 1

Interaction: This option is ignored if the TICKVALUELIST= or TICKVALUESEQUENCE= option is used.

THRESHOLDMIN= *number*

specifies a bias for including one more tick mark at the minimum end of the axis. For more information, see "Adjusting Axis Thresholds" on page 402.

Default: .30

Range: 0 - 1

Interaction: This option is ignored if the TICKVALUELIST= or TICKVALUESEQUENCE= option is used.

TICKVALUEFORMAT=(*format-options*) | DATA | *format*
specifies how to format the values for major tick marks.

Default: (MAXWIDTH=8, MAXDECIMALS=6, PREFERREDDECIMALS=2, EXTRACTSCALE=FALSE)

(*format-options*)
specifies one or more major tick value formatting options. Together, these options provide parameters for determining an optimal format (*w.d*, E*w*., BEST*w*.) for displaying major tick values.

MAXWIDTH = *integer*	Specifies the maximum width for displayed tick values. Values might be rounded or converted to E-notation to fit into this width.
MAXDECIMALS = *integer*	Specifies the maximum number of decimals for displayed tick values. Values might be rounded or converted to E-notation to fit into this width. MAXWIDTH > MAXDECIMALS
PREFERREDDECIMALS = *integer*	Specifies the number of decimal places desired for most values. The actual number might vary based on other constraints.
EXTRACTSCALE = *boolean*	Specifies whether to extract some factor of ten from all tick values in order to reduce the overall width of the tick values and improve legibility. The extracted factor is concatenated to the existing axis label. Example of revised axis label: *Number Sold (x1000)*

DATA
use the format that has been assigned to the variable(s) contributing to the axis (or BEST6. if no format is assigned) to control the formatting of the major tick values.

format
use this format to control how the major tick values are formatted.

Restriction: The *format-option* are used only when no format has been assigned to the variable(s) that contribute to the axis or when the assigned format(s) are *w.d*, E*w*. or BEST*w*.

Restriction: GTL currently honors most but not every SAS format. For details, see "SAS Formats Not Supported" on page 569.

Interaction: If the axis label is not displayed, the EXTRACTSCALE=TRUE is *format-option* ignored.

TICKVALUELIST=(*numeric-list*)
specifies the tick values for a linear axis as list.

Default: An internal algorithm determines the tick values.

Requirement: The *numeric-list* must be enclosed in parentheses and each item separated by a blank.

Interaction: This option overrides the INTEGER= option.

Interaction: The values in the list are formatted according to the setting for the TICKVALUEFORMAT= option.

If ticks are specified outside of the axis range, then the axis range is extended to display the ticks.

See also: TICKVALUESEQUENCE= option as an alternative for customizing tick marks.

TICKVALUESEQUENCE=(*sequence-options*)

specifies the tick values for a linear axis by start, end, and increment.

Default: An internal algorithm determines the tick marks.

(sequence-options)

Three options that control major tick values. All three options must be provided.

START = *number*	Specifies the value for the first tick mark.
END = *number*	Specifies the value for the last tick mark.
INCREMENT = *number*	Specifies the increment for intermediate tick marks between the first and last tick marks. The END value always controls the last tick mark. The interval between the last tick mark and the previous tick mark might not necessarily be the INCREMENT value.

Interaction: This option overrides the INTEGER= option.

Interaction: This option is ignored if the DISPLAY= option does not display tick marks.

Interaction: The values in the sequence are formatted according to the setting for the TICKVALUEFORMAT= option.

If ticks are specified outside of the axis range, then the axis range is extended to display the ticks.

See also: TICKVALUELIST= option as an alternative for customizing tick marks.

Options for Time Axes Only

TIMEOPTS=(*time-axis-options*)

specifies one or more options for a time axis.

Variables associated with a time axis must be in SAS time, SAS date, or SAS datetime units and have an associated SAS time, date, or datetime format.

Options must be enclosed in parentheses. Each option is specified as a *name = value* pair and each pair is space separated.

Interaction: This option is ignored if the axis type is not TIME.

Time-Axis-Option	Description
INTERVAL	Specifies the time interval between major tick marks.
MINORTICKS	Specifies whether minor ticks are displayed.
TICKVALUEFORMAT	Specifies how to format the values for major tick marks.
TICKVALUELIST	Specifies the order of the tick values for a time axis as list.

INTERVAL=*interval*

specifies the time interval between major ticks. Valid *interval* keywords are AUTO, SECOND, MINUTE, HOUR, DAY, TENDAY, WEEK, SEMIMONTH, MONTH, QUARTER, SEMIYEAR, YEAR.

The data column(s) mapped to a time axis must be in the same duration units: TIME, DATE, or DATETIME. The selection of an interval must be consistent with the duration unit. For example, if the data are in time units, you can select only AUTO, SECOND, MINUTE, HOUR.

Default: AUTO. An appropriate interval is chosen based on the data and the column date/datetime/time format.

Interaction: This option is ignored if the TICKVALUELIST= option is used.

INTERVAL	Unit	Tick interval	Default tick value format
AUTO	DATE, TIME, or DATETIME	automatically chosen	automatically chosen
SECOND	TIME or DATETIME	second	TIME8.
MINUTE	TIME or DATETIME	minute	TIME8.
HOUR	TIME or DATETIME	hour	TIME8.
DAY	DATE or DATETIME	day	DATE9.
TENDAY	DATE or DATETIME	10 days	DATE9.
WEEK	DATE or DATETIME	7 days	DATE9.
SEMIMONTH	DATE or DATETIME	1st and 16th of each month	DATE9.

INTERVAL	Unit	Tick interval	Default tick value format
MONTH	DATE or DATETIME	month	MONYY7.
QUARTER	DATE or DATETIME	3 months	YYQC6.
SEMIYEAR	DATE or DATETIME	6 months	MONYY7.
YEAR	DATE or DATETIME	year	YEAR4.

MINORTICKS=Boolean

specifies whether minor ticks are displayed.

Default: FALSE

Interaction: The number of minor ticks is dependent on the value of the INTERVAL= option. For example, if INTERVAL=YEAR, there would be four minor ticks (1 per quarter).

Interaction: This option is ignored if the DISPLAY= option does not display tick marks, or if the TICKVALUELIST= option is used.

TICKVALUEFORMAT=*format* | DATA

specifies how to format the values for major tick marks.

Default: The default format used by the INTERVAL= option. The default does not apply if TICKVALUELIST= is specified.

format

A SAS date, time, or datetime format to control how the major tick values are displayed. This format must be in the same duration units as the data column(s) mapped to a time axis: TIME, DATE, or DATETIME and should be appropriate for the value of the INTERVAL= option. For example, if INTERVAL=MONTH and there are two years of data displayed on the axis, choosing TICKVALUEFORMAT=YEAR. would result in several ticks having the same year value.

DATA

Specifies that the SAS date, time, or datetime format associated with the data column assigned to the axis be used to control how the major tick values are displayed.

Restriction: GTL currently honors most but not every SAS format. For details, see "SAS Formats Not Supported" on page 569.

TICKVALUELIST=(*time-constant-list* | *date-constant-list* | *datetime-constant-list* | *numeric-list*)

specifies the tick values for a time axis as list.

Default: An internal algorithm determines the tick values.

Requirement: The list must be enclosed in parentheses and each item separated by a blank. The items in the list must be in the same duration units as the data mapped to the axis: TIME, DATE, or DATETIME. The values can be expressed as SAS TIME, DATE, or DATETIME constants (for example, "13:23"T, "11MAY06"D, or "11MAY06:13:23"DT) or their numeric equivalents.

Restriction: This option does not extend the data range of the axis. If the values fall within the default data range, they are used.

Interaction: The values in the list are formatted according to the setting for the TICKVALUEFORMAT= option. If TICKVALUEFORMAT= is not used, the values are formatted according to the column format (the default TICKVALUEFORMAT value is not applied to these values).

Chapter 44
Axis Options for LAYOUT LATTICE

Description and Syntax

Axis options for the plots within a LATTICE layout are specified with the following statements within a LAYOUT LATTICE statement block:

COLUMNAXIS /*external-axis-options*

ROWAXIS /*external-axis-options*

Example Program and Statement Details

Overview

The LAYOUT LATTICE statement creates a grid of graphs that automatically aligns plot areas, data display areas, axis labels, and headers across the columns and rows of the layout. The axis data ranges can be scaled, and the axes for individual cells in the layout can be managed by row and by column using COLUMNAXIS and ROWAXIS statements. COLUMNAXIS statements are used within a COLUMNAXES block to externalize column axes for the layout. Similarly, ROWAXIS statements are used within a ROWAXES block to externalize row axes for the layout. For the specifications to take effect,

- UNION or UNIONALL data scaling must be set for the affected columns and rows. The data scaling is set with COLUMNDATARANGE= and ROWDATARANGE= options.
- Within the COLUMNAXES block, the number of COLUMNAXIS statements must match the number of columns. Within the ROWAXES block, the number of ROWAXIS statements must match the number of rows. If you do not specify enough of either

statement, the last statement is used on remaining cells. If you specify too many statements, the extra statements are ignored.

The following example shows COLUMNAXIS statements for a lattice with two columns:

```
layout lattice / columns=2 columndatarange=union;
  columnaxes;
    columnaxis / griddisplay=on displaysecondary=(ticks tickvalues);
    columnaxis / griddisplay=on displaysecondary=(ticks tickvalues);
  endcolumnaxes;

/* rest of lattice definition */

endlayout;
```

COLUMNAXIS and ROWAXIS statements are similar to the XAXISOPTS= and YAXISOPTS= options for LAYOUT OVERLAY, with the following differences:

- COLUMNAXIS and ROWAXIS specify a secondary axis by using the DISPLAYSECONDARY= option, which shows a secondary axis on the opposite side of the lattice.
- When COLUMNAXIS and ROWAXIS are used, any axis options specified on plots within the affected columns or rows are ignored. All axis features for the external axes must be specified on the COLUMNAXIS or ROWAXIS statement.
- When COLUMNAXIS and ROWAXIS are used, any LAYOUT OVERLAYEQUATED layouts specified for cells in the affected columns or rows are implemented as LAYOUT OVERLAY layouts. Equated axes are not supported on external axes.

In the default cases for each plot in the layout, the axis type is always DISCRETE, LINEAR, or TIME. The TYPE= option enables you to specify an axis type that overrides the default. For example, when appropriate for the data, you can request a LOG axis. When you override the default axis type, you must be sure to specify the correct axis type for the plot(s) that you are defining.

Each axis type has features specific to that type, and the following axis options enable you to specify features for the different types: DISCRETEOPTS= , LINEAROPTS= , LOGOPTS= , and TIMEOPTS= . One or more of these options can be specified for an axis, but the specified settings are applied only to the axis type that supports them.

General Options for All Axes in a Lattice

The options that are documented in this section can be used with any of the axis types that are supported within a LATTICE layout. Subsequent sections in the chapter document the axis options that are available only for specific axis types: discrete, linear, log, or time axes.

Statement Option	Description
DISCRETEOPTS	Specifies features for a discrete column axis. This option is not available in the ROWAXIS statement.
DISPLAY	Controls which axis features are displayed on the primary axis.
DISPLAYSECONDARY	Controls which axis features are displayed on the secondary axis.

Statement Option	Description
GRIDATTRS	Specifies the attributes of the grid lines.
GRIDDISPLAY	Specifies whether axis grid lines are displayed.
LABEL	Specifies the axis label.
LABELATTRS	Specifies the color and font attributes of the axis label.
LINEAROPTS	Specifies features for a standard numeric interval axis.
LOGOPTS	Specifies features for a log axis.
OFFSETMAX	Reserves an area at the maximum end of the axis. No tick marks are displayed in the reserved area.
OFFSETMIN	Reserves an area at the minimum end of the axis. No tick marks are displayed in the reserved area.
SHORTLABEL	Specifies an alternate axis label to use if the default or specified axis label is too long for the axis length.
TICKVALUEATTRS	Specifies the color and font attributes of the axis tick values.
TIMEOPTS	Specifies features for a TIME axis.
TYPE	Specifies the type of axis to use.

DISCRETEOPTS=(*discrete-axis-options*)

specifies one or more options for a discrete X or X2 axis. See "Options for Discrete Column Axes Only" on page 443.

DISPLAY=STANDARD | ALL | NONE | (*display-options*)

controls which axis features are displayed on the primary axis.

Default: STANDARD

STANDARD

specifies that the LABEL, LINE, TICKS, and TICKVALUES are displayed

ALL

specifies that the LABEL, LINE, TICKS, and TICKVALUES are displayed

NONE

specifies that no axis features are displayed

(*display-options*)

a list of space-delimited options, enclosed in parentheses. The list must include one or more of the following:

LABEL	displays the axis label

LINE	displays the axis line
TICKS	displays the tick marks
TICKVALUES	displays the values that are represented by the major tick marks

The default line attributes for the axis line and axis tick marks are defined in the GraphAxisLine style element.

See GRIDDISPLAY= and GRIDATTRS= for setting axis grid lines.

DISPLAYSECONDARY=NONE | ALL | STANDARD | (*display-options*)
controls which axis features are displayed on the secondary axis.

Default: NONE

NONE
specifies that no secondary axis features are displayed

STANDARD
specifies that the LABEL, LINE, TICKS, and TICKVALUES are displayed on the secondary axis

ALL
specifies that the LABEL, LINE, TICKS, and TICKVALUES are displayed on the secondary axis

(*display-options*)
a list of space-delimited options, enclosed in parentheses. The list must include one or more of the following:

LABEL	displays the axis label
LINE	displays the axis line
TICKS	displays the tick marks
TICKVALUES	displays the values that are represented by the major tick marks

The secondary axis is a duplicate of the primary axis (the axis to which the data are mapped). It is on the opposite side of the lattice from the primary axis.

GRIDATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)
specifies the attributes of the grid lines. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Line Options" on page 564 for available *line-options*.

Default: The GraphGridLines style element.

Interaction: This option is ignored if the GRIDDISPLAY= option does not display the grid lines.

GRIDDISPLAY=AUTO_OFF | AUTO_ON | ON | OFF
specifies whether axis grid lines are displayed.

Default: AUTO_OFF

AUTO_OFF
specifies that grid lines are not displayed unless the GraphGridLines element in the current style contains DisplayOpts="ON".

AUTO_ON
specifies that grid lines are displayed unless the GraphGridLines element in the current style contains DisplayOpts="OFF".

ON

specifies that grid lines are always displayed. The current style has no override.

OFF

specifies that grid lines are never displayed. The current style has no override.

This option enables the template to absolutely control the display of grid lines or to allow interaction with the current style to decide whether grid lines are displayed. When displayed, the grids appear in all cells.

Supplied styles use DisplayOpts="AUTO," which means that the style has no "preference" about grid lines and the graphics template setting for grid lines is always used.

Discussion: The following table shows the end results for various combinations of the GRIDDISPLAY= option and the DisplayOpts= attribute of the GraphGridLines style element. Most supplied templates use the default setting AUTO_OFF to indicate a preference for not displaying grid lines, but allowing the style to override.

GRIDDISPLAY= option	DisplayOpts= style attribute	Grid Lines Shown?
AUTO_OFF	AUTO	no
AUTO_OFF	ON	yes
AUTO_OFF	OFF	no
AUTO_ON	AUTO	yes
AUTO_ON	ON	yes
AUTO_ON	OFF	no
ON	any value	yes
OFF	any value	no

LABEL=*"string"* | (*"string"* ... *"string"*)

specifies the axis label. The *string* can be either a string literal or a dynamic. The list form implies that all included string literals or dynamics will be concatenated

Default: The default label is derived from the primary plot in the layout. For more information, see "When Plots Share Data and a Common Axis" on page 397.

Interaction: If the axis label is too long to fit along the axis, it is truncated by default. Use the SHORTLABEL= option to specify an alternate axis label to be used whenever truncation would normally occur.

Interaction: This option is ignored if the DISPLAY= or DISPLAYSECONDARY= option does not display the axis label.

LABELATTRS=*style-element* | *style-element* (*text-options*) | (*text-options*)

specifies the color and font attributes of the axis label. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Text Options" on page 566 for available *text-options*.

Default: The GraphLabelText style element.

Interaction: This option is ignored if the DISPLAY= or DISPLAYSECONDARY= option does not display the axis label.

LINEAROPTS=(*linear-axis-options*)
specifies one or more *linear-axis-options* for a numeric interval axis. See "Options for Linear Axes Only" on page 444.

LOGOPTS=(*log-axis-options*)
specifies one or more *log-axis-options* for a log axis. See "Options for Log Axes Only" on page 449.

OFFSETMAX=AUTO | AUTOCOMPRESS | *number*
Reserves an area at the maximum end of the axis. No tick marks are displayed in the reserved area. For more information, see "Adjusting Axis Offsets" on page 403.

Default: AUTO

Range: 0 - 1. The sum of OFFSETMAX= and OFFSETMIN= should not be more than 1.

AUTO
Reserves just enough area to fully display markers and other graphical features near the maximum end of an axis.

AUTOCOMPRESS
Automatic offset applied that prevents axis labels and tick values from extending beyond the axis length.

number
The offset is expressed as a decimal proportion of the full axis length.

OFFSETMIN=AUTO | AUTOCOMPRESS | *number*
Reserves an area at the minimum end of the axis. No tick marks are displayed in the reserved area. For more information, see "Adjusting Axis Offsets" on page 403.

Default: AUTO

Range: 0 - 1. The sum of OFFSETMAX= and OFFSETMIN= should not be more than 1.

AUTO
Reserves just enough area to fully display markers and other graphical features near the maximum end of an axis.

AUTOCOMPRESS
Automatic offset applied that prevents axis labels and tick values from extending beyond the axis length.

number
The offset is expressed as a decimal proportion of the full axis length.

SHORTLABEL=*"string"*
specifies an alternate axis label. This label is used when the default axis label or label specified by the LABEL= option is too long for the axis length.

Default: no default

Interaction: This option is ignored if the DISPLAY= or DISPLAYSECONDARY= option does not display the axis label.

If the specified label is itself too long for the axis, it is truncated in the display.

TICKVALUEATTRS=*style-element* | *style-element* (*text-options*) | (*text-options*)

specifies the color and font attributes of the axis tick values. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Text Options" on page 566 for available *text-options*.

Default: The GraphValueText style element.

Interaction: This option is ignored if the DISPLAY= or DISPLAYSECONDARY= option does not display the axis label.

TIMEOPTS=(*time-axis-options*)

specifies one or more *time-axis-options* for a time axis. See "Options for Time Axes Only" on page 450.

TYPE=AUTO | DISCRETE | LINEAR | TIME | LOG

specifies the type of axis to use.

Default: AUTO

AUTO

Requests that the axis type be automatically determined, based on the overlay contents. For more information, see "When Plots Share Data and a Common Axis" on page 397.

DISCRETE

Use a DISCRETE X or X2 axis if possible. This setting is not available on a ROWAXIS statement. The data for discrete axes can be character or numeric. For a column axis, you can add a DISCRETEOPTS= ( ) option list to customize this axis type.

LINEAR

Use a LINEAR axis if possible. You can add a LINEAROPTS= ( ) option list to customize this axis type.

TIME

Use a TIME axis if possible. Data for this axis must be SAS time, SAS date, or SAS datetime values. You can add a TIMEOPTS= ( ) option list to customize this axis type.

LOG

Use a LOG axis if possible. You can add a LOGOPTS= ( ) option list to customize this axis type.

Options for Discrete Column Axes Only

DISCRETEOPTS=(*discrete-axis-options*)

specifies one or more options for a discrete X or X2 axis. This option is not available in the ROWAXIS statement. Options must be enclosed in parentheses. Each option is specified as a *name* = *value* pair and each pair is space separated.

Interaction: This option is ignored if the axis type is not DISCRETE.

TICKVALUEFITPOLICY=ROTATE | other-policy

specifies a policy for avoiding tick value collision on a column axis.

Default: ROTATE

The following fit policies are available:

ROTATE	Tick values are rotated 45 degrees.
ROTATETHIN	Attempt ROTATE policy and then THIN policy.

STAGGER	Tick values alternate between two rows.
STAGGERROTATE	Attempt STAGGER policy and then ROTATE policy.
STAGGERTHIN	Attempt STAGGER policy and then THIN policy.
STAGGERTRUNCATE	Attempt STAGGER policy and then TRUNCATE policy.
TRUNCATE	Tick values are shortened when they exceed a certain number of characters.
TRUNCATEROTATE	Attempt TRUNCATE policy and then ROTATE policy.
TRUNCATESTAGGER	Attempt TRUNCATE policy and then STAGGER policy.
TRUNCATETHIN	Attempt TRUNCATE policy and then THIN policy.
THIN	Some tick values are removed.

Options for Linear Axes Only

LINEAROPTS=(*linear-axis-options*)

specifies one or more *linear-axis-options* for a numeric interval axis. Options must be enclosed in parentheses. Each option is specified as a *name* = *value* pair and each pair is space separated.

Interaction: This option is ignored if the axis type is not LINEAR.

Linear-Axis-Option	Description
INTEGER	Specifies that evenly spaced integer values are used for tick marks.
THRESHOLDMAX	Specifies a bias for including one more tick mark at the maximum end of the axis.
THRESHOLDMIN	Specifies a bias for including one more tick mark at the minimum end of the axis.
TICKVALUEFITPOLICY	Specifies a policy for avoiding tick value collision on an X or X2 axis. This option is not available in the ROWAXIS statement.
TICKVALUEFORMAT	Specifies how to format the values for major tick marks.
TICKVALUELIST	Specifies the order of the tick values for a linear axis as list.
TICKVALUEPRIORITY	Specifies whether an axis tick specification can extend the axis data range.
TICKVALUESEQUENCE	Specifies the tick values for a linear axis by start, end, and increment.

Linear-Axis-Option	**Description**
VIEWMAX	Specifies the maximum data value to include in the display.
VIEWMIN	Specifies the minimum data value to include in the display.

INTEGER=*boolean*

specifies that evenly spaced integer values are used for tick marks.

Default: FALSE

Interaction: This option is overridden by the TICKVALUELIST= or TICKVALUESEQUENCE= option.

Interaction: This option overrides the MAXDECIMALS= and PREFERREDDECIMALS= suboptions of the TICKVALUEFORMAT= option

THRESHOLDMAX= *number*

specifies a bias for including one more tick mark at the maximum end of the axis. For more information, see "Adjusting Axis Thresholds" on page 402.

Default: .30

Range: 0 - 1

Interaction: This option is ignored if the TICKVALUELIST= or TICKVALUESEQUENCE= option is used.

THRESHOLDMIN= *number*

specifies a bias for including one more tick mark at the minimum end of the axis. For more information, see "Adjusting Axis Thresholds" on page 402.

Default: .30

Range: 0 - 1

Interaction: This option is ignored if the TICKVALUELIST= or TICKVALUESEQUENCE= option is used.

TICKVALUEFITPOLICY=THIN | *other-policy*

specifies an ordered list of strategies to avoid tick value collision on an X or X2 axis. This option is not available in the ROWAXIS statement.

Default: THIN

The fit policy can be any one of the following:

THIN	Eliminate alternate tick values. Available for the COLUMNAXIS but not for the ROWAXIS.
ROTATE	Tick values are rotated 45 degrees.
ROTATETHIN	Attempt ROTATE policy and then THIN policy.
STAGGER	Tick values alternate between two columns.

STAGGERROTATE	Attempt STAGGER policy and then ROTATE policy.
STAGGERTHIN	Attempt STAGGER policy and then THIN policy.

If more than one strategy is used, the values in the list are formatted according to the TICKVALUEFORMAT= option.

TICKVALUEFORMAT=(*format-options*) | DATA | *format*
specifies how to format the values for major tick marks.

Default: (MAXWIDTH=8, MAXDECIMALS=6, PREFERREDDECIMALS=2, EXTRACTSCALE=FALSE)

(*format-options*)
specifies one or more major tick value formatting options. Together, these options provide parameters for determining an optimal format (*w.d*, E*w*., BEST*w*.) for displaying major tick values.

MAXWIDTH = *integer*	Specifies the maximum width for displayed tick values. Values might be rounded or converted to E-notation to fit into this width.
MAXDECIMALS = *integer*	Specifies the maximum number of decimals for displayed tick values. Values might be rounded or converted to E-notation to fit into this width. MAXWIDTH > MAXDECIMALS
PREFERREDDECIMALS = *integer*	Specifies the number of decimal places desired for most values. The actual number might vary based on other constraints.
EXTRACTSCALE = *boolean*	Specifies whether to extract some factor of ten from all tick values in order to reduce the overall width of the tick values and improve legibility. The extracted factor is concatenated to the existing axis label. Example of revised axis label: *Number Sold (x1000)*

DATA
use the format that has been assigned to the variable(s) contributing to the axis (or BEST6. if no format is assigned) to control the formatting of the major tick values.

format
use this format to control how the major tick values are formatted.

Restriction: The *format-options* are used only when no format has been assigned to the variable(s) that contribute to the axis or when the assigned format(s) are *w.d*, E*w*. or BEST*w*.

Restriction: GTL currently honors most but not every SAS format. For details, see "SAS Formats Not Supported" on page 569.

Interaction: If the axis label is not displayed, the EXTRACTSCALE=TRUE *format-option* is ignored.

TICKVALUELIST=(*numeric-list*)

specifies the tick values for a linear axis as list.

Default: An internal algorithm determines the tick marks based on the actual axis data range or the data range established by the VIEWMIN= and VIEWMAX= options. By default when this option is used, the only tick values that appear are those in *numeric-list* that fall within the explicit data range (set by VIEWMIN= and VIEWMAX=) or implicit data range (set by the actual data minimum and data maximum).

Requirement: The *numeric-list* must be enclosed in parentheses and each item separated by a blank.

Interaction: This option overrides the INTEGER=

Interaction: This option is ignored if the TICKVALUESEQUENCE= option is specified, or if the DISPLAY= or DISPLAYSECONDARY= option does not display the tick values.

Interaction: The VIEWMIN= and VIEWMAX= options alter the axis data range. If the VIEWMIN= option is set to the minimum tick list value and the VIEWMAX= option is set to the maximum tick list value, all ticks in the tick list are displayed.

Interaction: If the TICKVALUEPRIORITY= is set to TRUE, 1) the VIEWMIN= and VIEWMAX= options are ignored if they are fully enclosed by the *numeric-list*, and 2) the tick *numeric-list* can extend the implicit data range of the axis, but never reduce it.

The values in the list are formatted according to the setting for the TICKVALUEFORMAT= option.

TICKVALUEPRIORITY=*boolean*

specifies whether an axis tick specification (TICKVALUELIST= or TICKVALUESEQUENCE=) can extend the axis data range.

Default: FALSE.

FALSE

the only tick values that appear are the user-specified tick values that fall within the explicit data range (set by VIEWMIN= and VIEWMAX=) or implicit data range (set by the actual data minimum and data maximum).

TRUE

the axis data range might be extended (but not reduced) to include the minimum and maximum specified by either the TICKVALUELIST= or TICKVALUESEQUENCE= option. If minimum and maximum of the user-specified values are within the data range, this option has no effect.

Interaction: When this option is set to TRUE, the VIEWMIN= and VIEWMAX= options are ignored if they are fully enclosed by value range set by the TICKVALUELIST= or TICKVALUESEQUENCE= option.

Interaction: This option is ignored if the DISPLAY= or DISPLAYSECONDARY= option does not display the tick values.

TICKVALUESEQUENCE=(*sequence-options*)

specifies the tick values for a linear axis by start, end, and increment.

Default: An internal algorithm determines the tick marks based on the actual axis data range or the data range established by the VIEWMIN= and VIEWMAX= options. By default when this option is used, the only tick values that appear are those that fall within the explicit data range (set by VIEWMIN= and VIEWMAX=) or implicit data range (set by the actual data minimum and data maximum).

(*sequence-options*)

Three settings that control major tick values. All three must be provided.

START = *number*	Specifies the value for the first tick mark.
END = *number*	Specifies the value for the last tick mark.
INCREMENT = *number*	Specifies the increment for intermediate tick marks between the first and last tick marks. The END value always controls the last tick mark. The interval between the last tick mark and the previous tick mark might not necessarily be the INCREMENT value.

Interaction: This option overrides the INTEGER= option.

Interaction: The VIEWMIN= and VIEWMAX= options alter the axis data range. If the VIEWMIN= option is set to the START= option value and the VIEWMAX= option is set to the END= option value, all ticks in the tick sequence are displayed.

Interaction: If the TICKVALUEPRIORITY= option is set to TRUE, the tick sequence might extend the explicit data range of the axis, but never reduce it.

Interaction: This option is ignored if the DISPLAY= or DISPLAYSECONDARY= option does not display tick marks.

The values in the sequence are formatted according to the setting for the TICKVALUEFORMAT= option.

See also: TICKVALUELIST= option as an alternative for customizing tick marks.

VIEWMAX= *number*

specifies the maximum data value to include in the display (the value might be adjusted by the threshold calculation). For more information, see "Adjusting the Axis View" on page 402.

Default: The maximum value in the data for the specified axis.

Interaction: This option does not determine the maximum axis tick value displayed. The THRESHOLDMAX= value is used to determine the maximum tick value.

Interaction: This option has no effect if the TICKVALUEPRIORITY= is set to TRUE.

Setting a VIEWMAX= or VIEWMIN= value does not alter the original data or any calculations on it.

VIEWMIN= *number*

specifies the minimum data value to include in the display (the value might be adjusted by the threshold calculation). For more information, see "Adjusting the Axis View" on page 402.

Default: The minimum value in the data for the specified axis.

Interaction: This option does not determine the minimum axis tick value displayed. The THRESHOLDMIN= value is used to determine the maximum tick value.

Interaction: This option has no effect if the TICKVALUEPRIORITY= is set to TRUE.

Setting a VIEWMAX= or VIEWMIN= value does not alter the original data or any calculations on it.

Options for Log Axes Only

LOGOPTS=(*log-axis-options*)

specifies one or more *log-axis-options* for a log axis. Options must be enclosed in parentheses. Each option is specified as a *name* = *value* pair and each pair is space separated.

Interaction: This option is ignored if the axis type is not LOG.

Log-Axis-Option	Description
BASE	Specifies the base of the logarithmic scale for the axis values.
MINORTICKS	Specifies whether minor ticks are displayed.
TICKINTERVALSTYLE	Specifies how to scale and format the values for major tick marks.
VIEWMAX	Specifies the maximum data value to include in the display.
VIEWMIN	Specifies the minimum data value to include in the display.

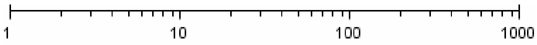
BASE=10 | 2 | E

specifies the base of the logarithmic scale for the axis values.

Default: 10

MINORTICKS=*boolean*

specifies whether minor ticks are displayed:



Default: FALSE

Restriction: Minor ticks can be displayed only when BASE=10 and TICKINTERVALSTYLE= is LOGEXPAND or LOGEXPONENT.

If this option is set to TRUE, the number of minor ticks is automatically determined.

TICKINTERVALSTYLE=AUTO | LOGEXPAND | LOGEXPONENT | LINEAR

specifies how to scale and format the values for major tick marks.

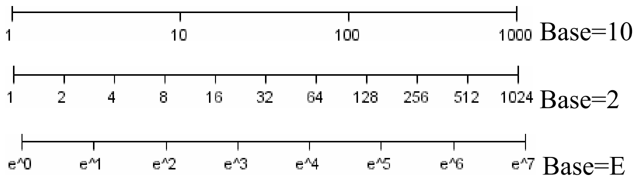
Default: AUTO

AUTO

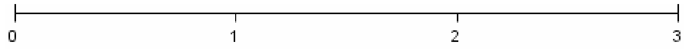
A LOGEXPAND, LOGEXPONENT, or LINEAR representation is chosen automatically based on the range of the data. When the data range is small (within an order of magnitude), a LINEAR representation is typically used. Data ranges that encompass several orders of magnitude typically use the LOGEXPAND or LOGEXPONENT representation.

LOGEXPAND

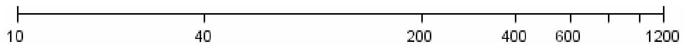
Major ticks are placed at uniform intervals at integer powers of the base. The tick values are expanded as follows:



LOGEXPONENT
Major ticks are placed at uniform intervals at integer powers of the base. The tick values are only the integer exponents for all bases.



LINEAR
Major tick marks are placed at non-uniform intervals that cover the range of the data.



Restriction: For LOGEXPAND and LOGEXPONENT, formats on data columns contributing to the axis are ignored. For LINEAR, ticks values are automatically formatted when the column format is not assigned or one of *w.d*, E*w*. or BEST*w*. Other formats (SAS defined or user-defined) are used if specified.

Restriction: GTL currently honors most but not every SAS format. For details, see "SAS Formats Not Supported" on page 569.

Interaction: When BASE=10 and LOGEXPAND or LOGEXPONENT is used, an intermediate tick is added whenever the axis data range is less than or equal to 1.5 powers of 10.

If you use TICKINTERVALSTYLE=LOGEXPONENT, you might want to include information in the axis label about which base is used.

VIEWMAX= *number*
specifies the maximum data value to include in the display. For more information, see "Adjusting the Axis View" on page 402.

Default: The maximum value in the data for the specified axis.

Setting a VIEWMAX= or VIEWMIN= value does not alter the original data or any calculations on it.

VIEWMIN= *number*
specifies the minimum data value to include in the display. For more information, see "Adjusting the Axis View" on page 402.

Default: The minimum value in the data for the specified axis.

Setting a VIEWMAX= or VIEWMIN= value does not alter the original data or any calculations on it.

Options for Time Axes Only

TIMEOPTS=(*time-axis-options*)
specifies one or more *time-axis-options* for a time axis.

Variables associated with a time axis must be in SAS time, SAS date, or SAS datetime units and have an associated SAS time, date, or datetime format.

Options must be enclosed in parentheses. Each option is specified as a *name = value* pair and each pair is space separated.

Interaction: This option is ignored if the axis type is not TIME.

Time-Axis-Option	Description
INTERVAL	Specifies the time interval between major tick marks.
MINORTICKS	Specifies whether minor tick marks are displayed.
SPLITTICKVALUE	Specifies whether to split the tick values on column axes, if possible. This option is not available in the ROWAXIS statement.
TICKVALUEFITPOLICY	Specifies a policy for avoiding tick value collision on column axes. This option is not available in the ROWAXIS statement.
TICKVALUEFORMAT	Specifies how to format the values for major tick marks.
TICKVALUELIST	Specifies the order of the tick values for a time axis as list.
TICKVALUEPRIORITY	Specifies whether an axis tick specification can extend the axis data range.
VIEWMAX	Specifies the maximum data value to include in the display.
VIEWMIN	Specifies the minimum data value to include in the display.

INTERVAL=*interval*

specifies the time interval between major ticks. Valid *interval* keywords are AUTO, SECOND, MINUTE, HOUR, DAY, TENDAY, WEEK, SEMIMONTH, MONTH, QUARTER, SEMIYEAR, YEAR.

The data column(s) mapped to a time axis must be in the same duration units: TIME, DATE, or DATETIME. The selection of an interval must be consistent with the duration unit. For example, if the data are in time units, you can select only AUTO, SECOND, MINUTE, HOUR.

Default: AUTO. An appropriate interval is chosen based on the data.

Interaction: This option is ignored if the TICKVALUELIST= option is used.

INTERVAL	Unit	Tick interval	Default tick value format
AUTO	DATE, TIME, or DATETIME	automatically chosen	automatically chosen
SECOND	TIME or DATETIME	second	TIME8.

INTERVAL	Unit	Tick interval	Default tick value format
MINUTE	TIME or DATETIME	minute	TIME8.
HOUR	TIME or DATETIME	hour	TIME8.
DAY	DATE or DATETIME	day	DATE9.
TENDAY	DATE or DATETIME	10 days	DATE9.
WEEK	DATE or DATETIME	7 days	DATE9.
SEMIMONTH	DATE or DATETIME	1st and 16th of each month	DATE9.
MONTH	DATE or DATETIME	month	MONYY7.
QUARTER	DATE or DATETIME	3 months	YYQC6.
SEMIYEAR	DATE or DATETIME	6 months	MONYY7.
YEAR	DATE or DATETIME	year	YEAR4.

MINORTICKS=*boolean*

specifies whether minor tick marks are displayed.

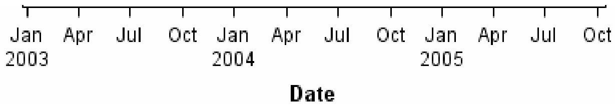
Default: FALSE

Interaction: The number of minor ticks is dependent on the value of the INTERVAL= option. For example, on a row axis, if INTERVAL=YEAR, there would be four minor ticks (one per quarter).

Interaction: This option is ignored if the TICKVALUELIST= option is used, or if the DISPLAY= or DISPLAYSECONDARY= option does not display the tick marks.

SPLITTICKVALUE=*boolean*

specifies whether to split the tick values on column axes, if possible (this option is not available in the ROWAXIS statement). For example, with INTERVAL=MONTH, this is how tick values are split:



Default: TRUE

TRUE

axis tick values are split into two lines allowing more tick values to appear

FALSE

typically, fewer tick values fit, causing thinning, rotation, or staggering of the values. See the TICKVALUEFITPOLICY= option.

Interaction: This option is ignored if the TICKVALUELIST= or TICKVALUEFORMAT= option is specified.

TICKVALUEFITPOLICY=THIN | ROTATE | ROTATETHIN | STAGGER | STAGGERROTATE | STAGGERTHIN

specifies a policy for avoiding tick value collision on column axes. This option is not available in the ROWAXIS statement.

Default: THIN

The fit policy can be any one of the following policies:

THIN	Some tick values are removed.
ROTATE	Tick values are rotated 45 degrees.
ROTATETHIN	Attempt ROTATE policy and then THIN policy.
STAGGER	Tick values alternate between two columns.
STAGGERROTATE	Attempt STAGGER policy and then ROTATE policy.
STAGGERTHIN	Attempt STAGGER policy and then THIN policy.

Interaction: When SPLITTICKVALUE= TRUE, this option is ignored and only the THIN policy is used.

TICKVALUEFORMAT=*format* | DATA

specifies how to format the values for major tick marks.

Default: The default format used by the INTERVAL= option. The default does not apply if TICKVALUELIST= is specified.

format

A SAS date, time, or datetime format to control how the major tick values are displayed. This format should be in the same duration units as the data column(s) mapped to a time axis: TIME, DATE, or DATETIME, and should be appropriate for the value of the INTERVAL= option. For example, if INTERVAL=MONTH and there are two years of data displayed on the axis, choosing TICKVALUEFORMAT=YEAR. would result in several ticks having the same year value.

DATA

Use the SAS date, time, or datetime format associated with the data column assigned to the axis to control how the major tick values are displayed.

Restriction: GTL currently honors most but not every SAS format. For details, see “SAS Formats Not Supported” on page 569.

Interaction: If this option is specified, the SPLITTICKVALUE= option is ignored.

TICKVALUELIST=(*time-constant-list* | *date-constant-list* | *datetime-constant-list* | *numeric-list*)

specifies the tick values for a time axis as list.

Default: An internal algorithm determines the tick values.

Requirement: The list must be enclosed in parentheses and each item separated by a blank. The items in the list must be in the same duration units as the data mapped to the axis: TIME, DATE, or DATETIME. The values can be expressed as SAS TIME, DATE, or DATETIME constants (for example, "13:23"T, "11MAY06"D, or "11MAY06:13:23"DT) or their numeric equivalents.

Restriction: If TICKVALUEPRIORITY= is set to FALSE, this option does not extend the data range of the axis. If the values fall within the default data range or that specified by the VIEWMIN= or VIEWMAX= options, they are used.

The values in the list are formatted according to the setting for the TICKVALUEFORMAT= option. If TICKVALUEFORMAT= is not used, the values are formatted according to the column format (the default TICKVALUEFORMAT value is not applied to these values).

See also: TICKVALUESEQUENCE= option as an alternate to customizing tick marks.

TICKVALUEPRIORITY=*boolean*

specifies whether an axis tick specification (TICKVALUELIST=) can extend the axis data range.

Default: FALSE

FALSE

the only tick values that appear are the user-specified tick values that fall within the explicit data range (set by VIEWMIN= and VIEWMAX=) or implicit data range (set by the actual data minimum and data maximum).

TRUE

the axis data range might be extended (but not reduced) to include the minimum and maximum specified by the TICKVALUELIST= option. If minimum and maximum of the user-specified values are within the data range, this option has no effect.

Interaction: When this option is set to TRUE, the VIEWMIN= or VIEWMAX= options might have no effect.

Interaction: This option is ignored if the DISPLAY= or DISPLAYSECONDARY= option does not display tick marks.

VIEWMAX= *number*

specifies the maximum data value to include in the display. For more information, see "Adjusting the Axis View" on page 402.

Default: The maximum value in the data for the specified axis.

Setting a VIEWMAX= or VIEWMIN= value does not alter the original data or any calculations on it.

VIEWMIN= *number*

specifies the minimum data value to include in the display. For more information, see "Adjusting the Axis View" on page 402.

Default: The minimum value in the data for the specified axis.

Setting a VIEWMAX= or VIEWMIN= value does not alter the original data or any calculations on it.

Chapter 45

Axis Options for LAYOUT OVERLAYEQUATED

Description and Syntax

Axis options for the plots within an OVERLAYEQUATED layout are specified with the following options on a LAYOUT OVERLAYEQUATED statement:

COMMONAXISOPTS= (*common-equated-axis-options*)

XAXISOPTS= (*equated-axis-options*)

YAXISOPTS= (*equated-axis-options*)

Example Program and Statement Details

Overview

The LAYOUT OVERLAYEQUATED statement is used for equated layouts where the X and Y axes always have equal increments between tick values. Because the axes within the equated layout are so closely correlated, some axis adjustments cannot be applied to one axis without applying them to the other.

For example, the INTEGER= option sets evenly spaced integer values for the axis tick marks. That setting must be applied to both axes if the correlation between them is to be maintained. The LAYOUT OVERLAYEQUATED statement provides the COMMONAXISOPTS= option for specifying the INTEGER= option and other options whose settings apply in common to both axes.

Despite the close correlation between the axes, some axis adjustments can be made to one axis without affecting the other. For example, displaying grid lines on one axis has no impact on the other. The XAXISOPTS= and YAXISOPTS= options are available for applying settings separately to the X and Y axes.

The following example template sets evenly spaced integer values for the axis tick marks of both axes. It also specifies the display of grid lines, tick marks, and tick values for the Y axis:

```
begingraph;
  layout overlayequated /
    commonaxisopts=(integer=true);
    yaxisopts=(griddisplay=on display=(ticks tickvalues));
    seriesplot x=var1 y=var2;
  endlayout;
endgraph;
```

Interaction. The OVERLAYEQUATED's axis options are ignored when the LAYOUT OVERLAYEQUATED statement is nested within another layout type that has external axes in effect. For example, the axis options are ignored when the statement is nested within a LAYOUT LATTICE with a COLUMNAXIS= or ROWAXIS= option in effect.

Options That Apply in Common to Both Equated Axes

The options that are documented in this section are applied to both axes and cannot be applied separately to an X or Y axis. See "Options That Apply Separately to an X or Y Equated Axis" on page 459 for a list of options that can be applied to a single axis.

COMMONAXISOPTS= (*common-equated-axis-options*)

specifies one or more options to apply to both the X and Y equated axes. Options must be enclosed in parentheses. Each option is specified as a *name* = *value* pair and each pair is space separated.

Equated-Axis-Option	Description
INTEGER	Specifies that evenly spaced integer values are used for tick marks for all axes.
TICKSTYLE	Specifies the placement of tick marks in relation to the axis line.
TICKVALUELIST	Specifies the order of the tick values as list.
TICKVALUEPRIORITY	Specifies whether an axis tick specification can extend the axis data range.
TICKVALUESEQUENCE	Specifies the tick values by start, end, and increment.
VIEWMAX	Specifies the maximum data value to include in the display.
VIEWMIN	Specifies the minimum data value to include in the display.

INTEGER=*boolean*

specifies that evenly spaced integer values are used for tick marks for all axes.

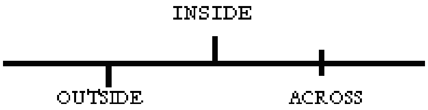
Default: FALSE

Interaction: This option is overridden by the TICKVALUELIST= or TICKVALUESEQUENCE= option.

Interaction: This option overrides the MAXDECIMALS= and PREFERREDDECIMALS= suboptions of the TICKVALUEFORMAT= option on XAXISOPTS= or YAXISOPTS=.

TICKSTYLE=OUTSIDE | INSIDE | ACROSS

specifies the placement of tick marks in relation to the axis line. The figure shows the tick display on an X axis.



Default: OUTSIDE

OUTSIDE

displays tick marks outside of the axis line.

INSIDE

displays tick marks inside of the axis line.

ACROSS

displays tick marks across the axis line.

Interaction: This option is ignored if the ticks are not displayed. See the DISPLAY= and DISPLAYSECONDARY= suboptions of XAXISOPTS= and YAXISOPTS=.

This option has no affect on the placement of the tick values, which are always outside the axis line.

TICKVALUELIST=(*numeric-list*)

specifies the tick values as a list.

Default: An internal algorithm determines the tick marks based on the actual axis data range or the data range established by the VIEWMIN= and VIEWMAX= options. By default when this option is used, the only tick values that appear are the *numeric-list* tick values that fall within the explicit data range (set by VIEWMIN= and VIEWMAX=) or implicit data range (set by the actual data minimum and data maximum).

Requirement: The *numeric-list* must be enclosed in parentheses and each item separated by a blank.

Interaction: This option overrides the INTEGER= option.

Interaction: This option is ignored if the LAYOUT OVERLAYEQUATED statement specifies EQUATETYPE=FIT (the default).

Interaction: This option is ignored if the TICKVALUESEQUENCE= option is specified, or if the DISPLAY= option does not display tick values.

Interaction: The VIEWMIN= and VIEWMAX= options alter the axis data range. If the VIEWMIN= option is set to the minimum tick list value and the VIEWMAX= option is set to the maximum tick list value, all ticks in the tick list are displayed.

Interaction: If the TICKVALUEPRIORITY= is set to TRUE, 1) the VIEWMIN= and VIEWMAX= options are ignored if they are fully enclosed by the *numeric-list* 2) the tick *numeric-list* can extend the implicit data range of the axis, but never reduce it.

The values in the list are formatted according to the setting for the TICKVALUEFORMAT= option on XAXISOPTS= or YAXISOPTS=.

TICKVALUEPRIORITY=*boolean*

specifies whether an axis tick specification (TICKVALUELIST= or TICKVALUESEQUENCE=) can extend the axis data range.

Default: FALSE

FALSE
the only tick values that appear are the user-specified tick values that fall within the explicit data range (set by VIEWMIN= and VIEWMAX=) or implicit data range (set by the actual data minimum and data maximum).

TRUE
the axis data range might be extended (but not reduced) to include the minimum and maximum specified by either the TICKVALUELIST= or TICKVALUESEQUENCE= option. If minimum and maximum of the user-specified values are within the data range, this option has no effect.

Interaction: When this option is set to TRUE, the VIEWMIN= and VIEWMAX= options are ignored if they are fully enclosed by value range set by the TICKVALUELIST= or TICKVALUESEQUENCE= option.

Interaction: This option is ignored if the DISPLAY= option does not display tick marks.

TICKVALUESEQUENCE=(*sequence-options*)
specifies the tick values by start, end, and increment.

Default: An internal algorithm determines the tick marks based on the actual axis data range or the data range established by the VIEWMIN= and VIEWMAX= options. By default when this option is used, the only tick values that appear are those that fall within the explicit data range (set by VIEWMIN= and VIEWMAX=) or implicit data range (set by the actual data minimum and data maximum).

(*sequence-options*)
Three options that control tick values. All three options must be provided.

START = *number*	Specifies the value for the first tick mark.
END = *number*	Specifies the value for the last tick mark.
INCREMENT = *number*	Specifies the increment for intermediate tick marks between the first and last tick marks. The END value always controls the last tick mark. The interval between the last tick mark and the previous tick mark might not necessarily be the INCREMENT value.

Interaction: This option is ignored if the LAYOUT OVERLAYEQUATED statement specifies EQUATETYPE=FIT (the default), or if the DISPLAY= option does not display tick marks.

Interaction: This option overrides the INTEGER= option.

Interaction: The VIEWMIN= and VIEWMAX= options alter the axis data range. If the VIEWMIN= option is set to the START= option value and the VIEWMAX= option is set to the END= option value, all ticks in the tick sequence are displayed.

Interaction: If the TICKVALUEPRIORITY= option is set to TRUE, the tick sequence might extend the explicit data range of the axis, but never reduce it.

The values in the sequence are formatted according to the setting for the TICKVALUEFORMAT= option on XAXISOPTS= or YAXISOPTS=.

See also: TICKVALUELIST= option as an alternative for customizing tick marks.

VIEWMAX= *number*

specifies the maximum data value to include in the display (the value might be adjusted by the threshold calculation). For more information, see "Adjusting the Axis View" on page 402.

Default: The maximum value in the data for the specified axis.

Interaction: This option does not determine the maximum axis tick value displayed. The THRESHOLDMAX= value is used to determine the maximum tick value.

Interaction: This option has no effect if the TICKVALUEPRIORITY= is set to TRUE.

Setting a VIEWMAX= or VIEWMIN= value does not alter the original data or any calculations on it.

VIEWMIN= *number*

specifies the minimum data value to include in the display (the value might be adjusted by the threshold calculation). For more information, see "Adjusting the Axis View" on page 402.

Default: The minimum value in the data for the specified axis.

Interaction: This option does not determine the minimum axis tick value displayed. The THRESHOLDMIN= value is used to determine the maximum tick value.

Interaction: This option has no effect if the TICKVALUEPRIORITY= is set to TRUE.

Setting a VIEWMAX= or VIEWMIN= value does not alter the original data or any calculations on it.

Options That Apply Separately to an X or Y Equated Axis

The options that are documented in this section can be applied to an X axis with the XAXISOPTS= option, or to the Y axis with the YAXISOPTS= option. See "Options That Apply in Common to Both Equated Axes" on page 456 for a list of options that apply in common to both axes.

XAXISOPTS= (*equated-axis-options*), YAXISOPTS= (*equated-axis-options*)

each option specifies one or more options for an equated axis. Options must be enclosed in parentheses. Each option is specified as a *name* = *value* pair and each pair is space separated.

Equated-Axis-Option	Description
DISPLAY	Controls which axis features are displayed on the primary axis.
DISPLAYSECONDARY	Controls which axis features are displayed on the secondary axis.
GRIDATTRS	Specifies the attributes of the grid lines.
GRIDDISPLAY	Specifies when axis grid lines are displayed.
LABEL	Specifies the axis label.
LABELATTRS	Specifies the color and font attributes of the axis label.

Equated-Axis-Option	Description
OFFSETMAX	Reserves an area at the maximum end of the axis. No tick marks are displayed in the reserved area.
OFFSETMIN	Reserves an area at the minimum end of the axis. No tick marks are displayed in the reserved area.
SHORTLABEL	Specifies an alternate axis label to use if the default or specified axis label is too long for the axis length.
THRESHOLDMAX	Specifies a bias for including one more tick mark at the maximum end of the axis.
THRESHOLDMIN	Specifies a bias for including one more tick mark at the minimum end of the axis.
TICKVALUEATTRS	Specifies the color and font attributes of the axis tick values.
TICKVALUEFITPOLICY	Specifies a policy for avoiding tick value collision. Only the default policy (THIN) is available for a Y or Y2 axis.
TICKVALUEFORMAT	Specifies how to format the values for tick marks.

DISPLAY=STANDARD | ALL | NONE | (*display-options*)

controls which axis features are displayed on the primary axis.

Default: STANDARD

STANDARD

specifies that the LABEL, LINE, TICKS, and TICKVALUES are displayed

ALL

specifies that LABEL, LINE, TICKS, and TICKVALUES are displayed

NONE

specifies that no axis features are displayed

(*display-options*)

a list of space-delimited options, enclosed in parentheses. The list must include one or more of the following:

LABEL	displays the axis label
LINE	displays the axis line
TICKS	displays the tick marks
TICKVALUES	displays the values that are represented by the tick marks

The default line attributes for the axis line and axis tick marks are defined in the GraphAxisLine style element.

See GRIDDISPLAY= and GRIDATTRS= for setting axis grid lines.

DISPLAYSECONDARY=NONE | ALL | STANDARD | (*display-options*)

controls which axis features are displayed on the secondary axis. When data are mapped to the X or Y axis, you can display an X2 or Y2 (secondary) axis using this option. The secondary axis is a duplicate of the X or Y axis but can have different display options.

Default: NONE

NONE

specifies that no axis features are displayed

STANDARD

specifies that the LABEL, LINE, TICKS, and TICKVALUES are displayed on the secondary axis

ALL

specifies that LABEL, LINE, TICKS, and TICKVALUES are displayed on the secondary axis

(*display-options*)

a list of space-delimited options, enclosed in parentheses. The list must include one or more of the following:

LABEL	displays the axis label
LINE	displays the axis line
TICKS	displays the tick marks
TICKVALUES	displays the values that are represented by the tick marks

GRIDATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)

specifies the attributes of the grid lines. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Line Options" on page 564 for available *line-options*.

Default: The GraphGridLines style element.

Interaction: This option is ignored if the GRIDDISPLAY= option does not display the grid lines.

GRIDDISPLAY=AUTO_OFF | AUTO_ON | ON | OFF

specifies when axis grid lines are displayed.

Default: AUTO_OFF

AUTO_OFF

specifies that grid lines are not displayed unless the GraphGridLines element in the current style contains DisplayOpts="ON."

AUTO_ON

specifies that grid lines are displayed unless the GraphGridLines element in the current style contains DisplayOpts="OFF."

ON

specifies that grid lines are always displayed. The current style has no override.

OFF

specifies that grid lines are never displayed. The current style has no override.

This option enables the template to absolutely control the display of grid lines or to allow interaction with the current style to decide whether grid lines are displayed.

Supplied styles use DisplayOpts="AUTO," which means that the style has no "preference" about grid lines and the graphics template setting for grid lines is always used.

Discussion: The following table shows the end results for various combinations of the GRIDDISPLAY= option and the DisplayOpts= attribute of the GraphGridLines style element. Most supplied templates use the default setting AUTO_OFF to indicate a preference for not displaying grid lines, but allowing the style to override.

GRIDDISPLAY= option	DisplayOpts= style attribute	Grid Lines Shown?
AUTO_OFF	AUTO	no
AUTO_OFF	ON	yes
AUTO_OFF	OFF	no
AUTO_ON	AUTO	yes
AUTO_ON	ON	yes
AUTO_ON	OFF	no
ON	any value	yes
OFF	any value	no

LABEL=*"string"* | (*"string"* ... *"string"*)

specifies the axis label. The *string* can be either a string literal or a dynamic. The list form implies that all included string literals or dynamics will be concatenated

Default: The default label is derived from the primary plot in the layout. For more information, see "When Plots Share Data and a Common Axis" on page 397.

Interaction: If the axis label is too long to fit along the axis, it is truncated by default. Use the SHORTLABEL= option to specify an alternate axis label to be used whenever truncation would normally occur.

Interaction: This option is ignored if the DISPLAY= option does not display the axis label.

LABELATTRS=*style-element* | *style-element* (*text-options*) | (*text-options*)

specifies the color and font attributes of the axis label. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Text Options" on page 566 for available *text-options*.

Default: The GraphLabelText style element.

Interaction: This option is ignored if the DISPLAY= option does not display the axis label.

OFFSETMAX=AUTO | AUTOCOMPRESS | *number*

reserves an area at the maximum end of the axis. No tick marks are displayed in the reserved area. For more information, see "Adjusting Axis Offsets" on page 403.

Default: AUTO

Range: 0 - 1. The sum of OFFSETMAX= and OFFSETMIN= should not be more than 1.

AUTO

Reserves just enough area to fully display markers and other graphical features near the maximum end of an axis.

AUTOCOMPRESS
Automatic offset applied that prevents axis labels and tick values from extending beyond the axis length.

number
The offset is expressed as a decimal proportion of the full axis length.

OFFSETMIN=AUTO | AUTOCOMPRESS | *number*
reserves an area at the minimum end of the axis. No tick marks are displayed in the reserved area. For more information, see "Adjusting Axis Offsets" on page 403.

Default: AUTO

Range: 0 - 1. The sum of OFFSETMAX= and OFFSETMIN= should not be more than 1.

AUTO
Reserves just enough area to fully display markers and other graphical features near the maximum end of an axis.

AUTOCOMPRESS
Automatic offset applied that prevents axis labels and tick values from extending beyond the axis length.]

number
The offset is expressed as a decimal proportion of the full axis length.

SHORTLABEL=*"string"*
specifies an alternate axis label. This label is used when the default axis label or label specified by the LABEL= option is too long for the axis length.

Default: no default

Interaction: This option is ignored if the DISPLAY= option does not display the axis label.

If the specified label is itself too long for the axis, it is truncated in the display.

THRESHOLDMAX= *number*
specifies a bias for including one more tick mark at the maximum end of the axis. For more information, see "Adjusting Axis Thresholds" on page 402.

Default: .30

Range: 0 - 1

Interaction: This option is ignored if the TICKVALUELIST= or TICKVALUESEQUENCE= option is used.

THRESHOLDMIN= *number*
specifies a bias for including one more tick mark at the minimum end of the axis. For more information, see "Adjusting Axis Thresholds" on page 402.

Default: .30

Range: 0 - 1

Interaction: This option is ignored if the TICKVALUELIST= or TICKVALUESEQUENCE= option is used.

TICKVALUEATTRS=*style-element* | *style-element* (*text-options*) | (*text-options*)
specifies the color and font attributes of the axis tick values. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Text Options" on page 566 for available *text-options*.

Default: The GraphValueText style element.

Interaction: This option is ignored if the DISPLAY= option does not display tick values.

TICKVALUEFITPOLICY=*policy*

specifies an ordered list of strategies to avoid tick value collision. Only the default setting (THIN) is available for a Y or Y2 axis.

Default: THIN

The collision-avoidance policy that is most effective depends on the number of tick values, their length, and the length of the axis. For an X or X2 axis, the policy can be one of the following:

THIN	Eliminate alternate tick values.
ROTATE	Tick values are rotated 45 degrees.
ROTATETHIN	Attempt ROTATE policy and then THIN policy.
STAGGER	Tick values alternate between two rows.
STAGGERROTATE	Attempt STAGGER policy and then ROTATE policy.
STAGGERTHIN	Attempt STAGGER policy and then THIN policy.

TICKVALUEFORMAT=(*format-options*) | DATA | *format*

specifies how to format the values for tick marks.

Default: (MAXWIDTH=8, MAXDECIMALS=6, PREFERREDDECIMALS=2, EXTRACTSCALE=FALSE)

(*format-options*)

specifies one or more tick value formatting options. Together, these options provide parameters for determining an optimal format (*w.d*, E*w*., BEST*w*.) for displaying tick values.

MAXWIDTH = *integer*	Specifies the maximum width for displayed tick values. Values might be rounded or converted to E-notation to fit into this width.
MAXDECIMALS = *integer*	Specifies the maximum number of decimals for displayed tick values. Values might be rounded or converted to E-notation to fit into this width. MAXWIDTH > MAXDECIMALS
PREFERREDDECIMALS = *integer*	Specifies the number of decimal places desired for most values. The actual number might vary based on other constraints.
EXTRACTSCALE = *boolean*	Specifies whether to extract some factor of ten from all tick values in order to reduce the overall width of the tick values and improve legibility. The extracted factor is concatenated to the existing axis

label. Example of revised axis label: *Number Sold (x1000)*

DATA

use the format that has been assigned to the variable contributing to the axis (or BEST6. if no format is assigned) to control the formatting of the tick values.

format

specifies a format that controls how the tick values are formatted.

Restriction: The *format-options* are used only when no format has been assigned to the variable that contributes to the axis or when the assigned format(s) are *w.d*, E*w*. or BEST*w*.

Restriction: GTL currently honors most but not every SAS format. For details, see "SAS Formats Not Supported" on page 569.

Interaction: If the axis label is not displayed, the EXTRACTSCALE=TRUE *format-option* is ignored.

Chapter 46

Axis Options for LAYOUT DATALATTICE/DATAPANEL

Description and Syntax

Axis options for the plots within DATALATTICE and DATAPANEL layouts are specified with the following options:

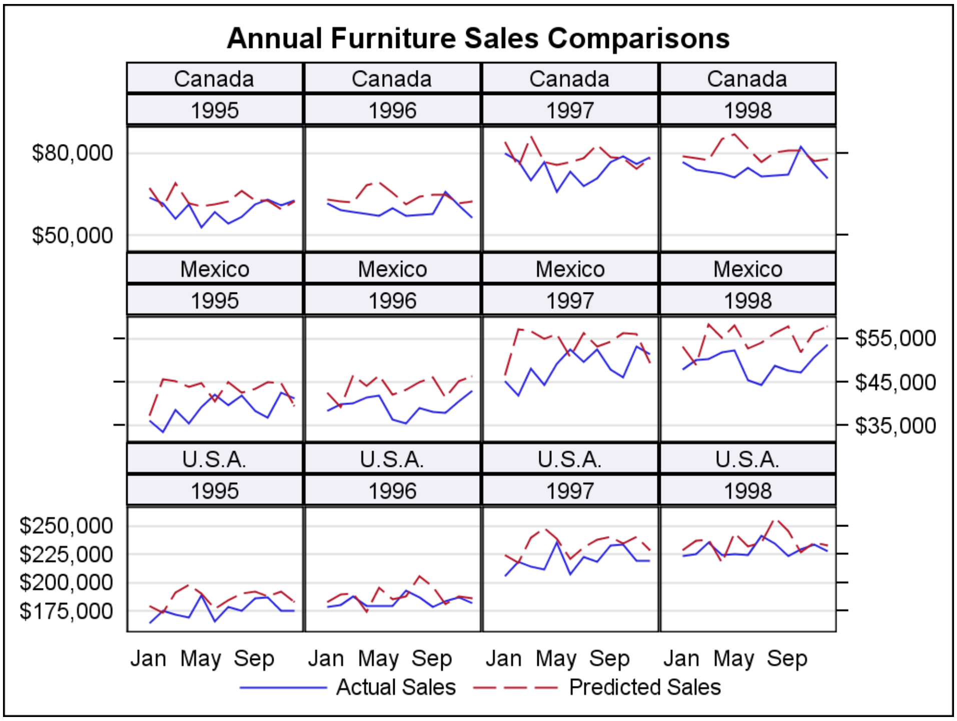
COLUMNAXISOPTS= (*axis-option(s)*)

ROWAXISOPTS= (*axis-option(s)*)

Example Program and Statement Details

Example Graph

The following graph was generated by the "Example Program" on page 468:



Example Program

This example shows how axis attributes can be managed separately for even and odd columns and rows in the layout grid. In this case, the ROWAXISOPTS= option is used to stagger the Y-axes attributes:

- On the primary (left) Y axis, DISPLAY= displays TICKS and TICKVALUES on the first and third rows, while ALTDISPLAY= displays just TICKS on the second row.
- On the secondary (right) Y axis, DISPLAYSECONDARY= displays just TICKS on the first and third rows, while ALTDISPLAYSECONDARY= displays TICKS and TICKVALUES on the second row.
- This alternating pattern could also have been set for the column axes. The pattern is independent of the number of rows and columns.

```
proc template;
 define statgraph layoutdatalattice;
  begingraph;
    entrytitle "Annual Furniture Sales Comparisons";
    layout datalattice rowvar=country columnvar=year /
      rowdatarange=union
      headerlabellocation=inside
      headerlabeldisplay=value
      headerbackgroundcolor=GraphAltBlock:color
      rowaxisopts=(griddisplay=on
        display=(tickvalues)
        altdisplay=(ticks)
        displaysecondary=(ticks)
        altdisplaysecondary=(ticks tickvalues)
        linearopts=(tickvalueformat=dollar12.))
      columnaxisopts=(display=(tickvalues)
        timeopts=(tickvalueformat=monname3.));
      layout prototype / cycleattrs=true;
        seriesplot x=month y=TotalActual / name="Actual";
        seriesplot x=month y=TotalPredict / name="Predict";
      endlayout;
      sidebar / align=bottom;
        discretelegend "Actual" "Predict" / border=false;
      endsidebar;
    endlayout;
  endgraph;
 end;
run;

proc summary data=sashelp.prdsal2 nway;
  class country year month;
  var actual predict;
  output out=prdsal2 sum=TotalActual TotalPredict;
run;

proc sgrender data=prdsal2 template=layoutdatalattice;
run;
```

Statement Summary

The LAYOUT DATALATTICE and LAYOUT DATAPANEL statements each create a grid of graphs based on the values of one or more classification variables. In the grid, the axes are always external and, by default, display information about the left and bottom axes, called the “primary” axes.

Both the LAYOUT DATALATTICE and LAYOUT DATAPANEL statements have COLUMNAXISOPTS= and ROWAXISOPTS= options that manage the axis display separately for columns and rows. The display settings that are available can manage odd and even columns and rows separately, enabling you to simplify the axis display within the grid.

- To manage the first, third, and odd occurrences of a primary axis, use the DISPLAY= option.
- To manage the second, fourth, and even occurrences of a primary axis, use the ALTDISPLAY= option.

To display “secondary” axes on the right or top of the grid, use the DISPLAYSECONDARY= and ALTDISPLAYSECONDARY= options.

In the default cases for the plots within the layout, the axis type is always DISCRETE, LINEAR, or TIME. The TYPE= option enables you to specify an axis type that overrides the default. For example, when appropriate for the data, you can request a LOG axis. When you override the default axis type, you must be sure to specify the correct axis type for the plot(s) that you are defining.

Each axis type has features specific to that type, and the following axis options enable you to specify features for the different types: DISCRETEOPTS= , LINEAROPTS= , LOGOPTS= , and TIMEOPTS= . One or more of these options can be specified for an axis, but the specified settings are applied only to the axis type that supports them.

Note: Certain plot types or layouts sometimes impose restrictions on what type of axis can be assigned. See the plot or layout documentation for default axis types and any restrictions that might apply.

General Options for All Axes in the Layout

The options that are documented in this section can be used with any of the axis types that are supported within a DATALATTICE or DATAPANEL layout. Subsequent sections in the chapter document the axis options that are available only for specific axis types: discrete, linear, log, or time axes.

Statement Option	Description
ALTDISPLAY	Controls which axis features are displayed on second, fourth, and other even row or column occurrences of the primary axis.
ALTDISPLAYSECONDARY	Controls which features are displayed on second, fourth, and other even row or column occurrences of the secondary axis.
DISCRETEOPTS	Specifies options for a discrete axis.
DISPLAY	Controls which axis features are displayed on first, third, and other odd row or column occurrences of the primary axis.
DISPLAYSECONDARY	Controls which axis features are displayed on first, third, and other odd row or column occurrences the secondary axis.
GRIDATTRS	Specifies the attributes of the grid lines.
GRIDDISPLAY	Specifies whether axis grid lines are displayed.
LABEL	Specifies the axis label.
LABELATTRS	Specifies the color and font attributes of the axis label.
LINEAROPTS	Specifies options for a standard numeric interval axis.
LOGOPTS	Specifies options for a log axis.
OFFSETMAX	Reserves an area at the maximum end of the axis. No tick marks are displayed in the reserved area.
OFFSETMIN	Reserves an area at the minimum end of the axis. No tick marks are displayed in the reserved area.
REVERSE	Specifies whether the tick values should appear in the reverse order.
SHORTLABEL	Specifies an alternate axis label.

Statement Option	Description
TICKVALUEATTRS	Specifies the color and font attributes of the axis tick value labels.
TIMEOPTS	Specifies options for a TIME axis.
TYPE	Specifies the type of axis to use.

ALTDISPLAY=STANDARD | ALL | NONE | (*display-options*)

controls which axis features are displayed on second, fourth, and other even row or column occurrences of the primary axis.

Default: The settings on the DISPLAY= option.

STANDARD

specifies that the LABEL, LINE, TICKS, and TICKVALUES are displayed

ALL

specifies that LABEL, LINE, TICKS, and TICKVALUES are displayed

NONE

specifies that no axis features are displayed

(*display-options*)

a list of space-delimited options, enclosed in parentheses. The list must include one or more of the following:

LABEL	Displays the axis label. A common label is displayed at the center of the grid, and the label applies to all the primary axes in the row or column. This label is overridden by a label specified on the DISPLAY= option.
LINE	Displays the axis line.
TICKS	Displays the tick marks.
TICKVALUES	Displays the values that are represented by the major tick marks.

The default line attributes for the axis line and axis tick marks are defined in the GraphAxisLine style element.

See GRIDDISPLAY= and GRIDATTRS= for setting axis grid lines.

This option can be used to obtain the alternating axis information as seen in a ScatterPlotMatrix.

ALTDISPLAYSECONDARY=NONE | ALL | STANDARD | (*display-options*)

controls which features are displayed on second, fourth, and other even row or column occurrences of the secondary axis.

Default: The settings on the DISPLAYSECONDARY= option.

NONE

specifies that no axis features are displayed

STANDARD

specifies that the LABEL, LINE, TICKS, and TICKVALUES are displayed on the secondary axis

ALL

specifies that LABEL, LINE, TICKS, and TICKVALUES are displayed on the secondary axis

(*display-options*)

a list of space-delimited options, enclosed in parentheses. The list must include one or more of the following:

LABEL	Displays the axis label. A common label is displayed at the center of the grid, and the label applies to all the secondary axes in the row or column. This label is overridden by a label specified on the DISPLAY= option.
LINE	Displays the axis line.
TICKS	Displays the tick marks.
TICKVALUES	Displays the values that are represented by the major tick marks.

The secondary axis is a duplicate of the primary axis (the axis to which the data are mapped). It is on the opposite side of the lattice from the primary axis.

The default line attributes for the axis line and axis tick marks are defined in the GraphAxisLine style element.

See GRIDDISPLAY= and GRIDATTRS= for setting axis grid lines.

This option can be used to obtain the alternating axis information as seen in a ScatterPlotMatrix.

DISCRETEOPTS=(*discrete-axis-options*)

specifies one or more options for a discrete X or X2 axis. See "Options for Discrete X or X2 Axes Only" on page 477.

DISPLAY=STANDARD | ALL | NONE | (*display-options*)

controls which axis features are displayed on first, third, and other odd row or column occurrences of the primary axis.

Default: STANDARD

STANDARD

specifies that the LABEL, LINE, TICKS, and TICKVALUES are displayed

ALL

specifies that LABEL, LINE, TICKS, and TICKVALUES are displayed

NONE

specifies that no axis features are displayed

(*display-options*)

a list of space-delimited options, enclosed in parentheses. The list must include one or more of the following:

LABEL	Displays the axis label. A common label is displayed at the center of the grid, and the label applies to all the primary axes in the row or column. This label overrides a label specified on the ALTDISPLAY= option.
LINE	Displays the axis line.
TICKS	Displays the tick marks.

TICKVALUES	Displays the values that are represented by the major tick marks.

The default line attributes for the axis line and axis tick marks are defined in the GraphAxisLine style element.

See GRIDDISPLAY= and GRIDATTRS= for setting axis grid lines.

DISPLAYSECONDARY=NONE | ALL | STANDARD | (*display-options*)
controls which axis features are displayed on first, third, and other odd row or column occurrences of the secondary axis.

Default: NONE

NONE
specifies that no axis features are displayed

STANDARD
specifies that the LABEL, LINE, TICKS, and TICKVALUES are displayed on the secondary axis

ALL
specifies that LABEL, LINE, TICKS, and TICKVALUES are displayed on the secondary axis

(*display-options*)
a list of space-delimited options, enclosed in parentheses. The list must include one or more of the following:

LABEL	Displays the axis label. A common label is displayed at the center of the grid, and the label applies to all the secondary axes in the row or column. This label overrides a label specified on the ALTDISPLAYSECONDARY= option.
LINE	Displays the axis line.
TICKS	Displays the tick marks.
TICKVALUES	Displays the values that are represented by the major tick marks.

The secondary axis is a duplicate of the primary axis (the axis to which the data are mapped). It is on the opposite side of the lattice from the primary axis.

The default line attributes for the axis line and axis tick marks are defined in the GraphAxisLine style element.

See GRIDDISPLAY= and GRIDATTRS= for setting axis grid lines.

GRIDATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)
specifies the attributes of the grid lines. See “General Syntax for Attribute Options” on page 563 for the syntax on using a *style-element* and “Line Options” on page 564 for available *line-options*.

Default: The GraphGridLines style element.

Interaction: For this option to have any effect, the grid lines must be enabled by the ODS style or the GRIDDISPLAY= option.

GRIDDISPLAY=AUTO_OFF | AUTO_ON | ON | OFF
specifies whether axis grid lines are displayed.

Default: AUTO_OFF

AUTO_OFF
specifies that grid lines are not displayed unless the GraphGridLines element in the current style contains DisplayOpts="ON."

AUTO_ON
specifies that grid lines are displayed unless the GraphGridLines element in the current style contains DisplayOpts="OFF."

ON
specifies that grid lines are always displayed. The current style has no override.

OFF
specifies that grid lines are never displayed. The current style has no override.

This option enables the template to absolutely control the display of grid lines or to allow interaction with the current style to decide whether grid lines are displayed.

Supplied styles use DisplayOpts="AUTO," which means that the style has no "preference" about grid lines and the graphics template setting for grid lines is always used.

Discussion: The following table shows the end results for various combinations of the GRIDDISPLAY= option and the DisplayOpts= attribute of the GraphGridLines style element. Most supplied templates use the default setting AUTO_OFF to indicate a preference for not displaying grid lines, but allowing the style to override.

GRIDDISPLAY= option	DisplayOpts= style attribute	Grid Lines Shown?
AUTO_OFF	AUTO	no
AUTO_OFF	ON	yes
AUTO_OFF	OFF	no
AUTO_ON	AUTO	yes
AUTO_ON	ON	yes
AUTO_ON	OFF	no
ON	any value	yes
OFF	any value	no

LABEL=*"string"* | (*"string"* ... *"string"*)
specifies the axis label. The *string* can be either a string literal or a dynamic. The list form implies that all included string literals or dynamics will be concatenated

Default: The default label is derived from the primary plot in the layout. For more information, see "When Plots Share Data and a Common Axis" on page 397.

Interaction: If the axis label is too long to fit along the axis, it is truncated by default. Use the SHORTLABEL= option to specify an alternate axis label to be used whenever truncation would normally occur.

Interaction: For this option to have any effect, the axis label must be enabled by the ODS style or the DISPLAY= or DISPLAYSECONDARY= option.

LABELATTRS=*style-element* | *style-element* (*text-options*) | (*text-options*)

specifies the color and font attributes of the axis label. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Text Options" on page 566 for available *text-options*.

Default: The GraphLabelText style element.

Interaction: For this option to have any effect, the axis label must be enabled by the ODS style or the DISPLAY= or DISPLAYSECONDARY= option.

LINEAROPTS=(*linear-axis-options*)

specifies one or more *linear-axis-options* for a numeric interval axis. See "Options for Linear Axes Only" on page 477.

LOGOPTS=(*log-axis-options*)

specifies one or more *log-axis-options* for a log axis. See "Options for Log Axes Only" on page 482.

OFFSETMAX=AUTO | AUTOCOMPRESS | *number*

reserves an area at the maximum end of the axis. No tick marks are displayed in the reserved area. For more information, see "Adjusting Axis Offsets" on page 403.

Default: AUTOCOMPRESS

Range: 0 - 1. The sum of OFFSETMAX= and OFFSETMIN= should not be more than 1.

AUTO

Reserves just enough area to fully display markers and other graphical features near the maximum end of an axis.

AUTOCOMPRESS

Automatic offset applied that prevents axis labels and tick values from extending beyond the axis length.

number

The offset is expressed as a decimal proportion of the full axis length. For a continuous axis, the offset follows the highest data value or highest tick value, whichever is greater.

OFFSETMIN=AUTO | AUTOCOMPRESS | *number*

reserves an area at the minimum end of the axis. No tick marks are displayed in the reserved area. For more information, see "Adjusting Axis Offsets" on page 403.

Default: AUTOCOMPRESS

Range: 0 - 1. The sum of OFFSETMAX= and OFFSETMIN= should not be more than 1.

AUTO

Reserves just enough area to fully display markers and other graphical features near the minimum end of an axis.

AUTOCOMPRESS

Automatic offset applied that prevents axis labels and tick values from extending beyond the axis length.

number

The offset is expressed as a decimal proportion of the full axis length. For a continuous axis, the offset precedes the lowest data value or lowest tick value, whichever is less.

REVERSE=*boolean*

specifies whether tick values should appear in the reverse order.

Default: FALSE

SHORTLABEL=*"string"*

specifies an alternate axis label. This label is used when the default axis label or label specified by the LABEL= option is too long for the grid length or the grid width.

Default: no default

Interaction: For this option to have any effect, the axis label must be enabled by the ODS style or the DISPLAY= or DISPLAYSECONDARY= option.

If the specified label is itself too long for the grid length or the grid width, it is truncated in the display.

TICKVALUEATTRS=*style-element* | *style-element* (*text-options*) | (*text-options*)

specifies the color and font attributes of the axis tick value labels. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Text Options" on page 566 for available *text-options*.

Default: The GraphValueText style element.

Interaction: For this option to have any effect, the tick values must be enabled by the ODS style or the DISPLAY= or DISPLAYSECONDARY= option.

TIMEOPTS=(*time-axis-options*)

specifies one or more *time-axis-options* for a time axis. See "Options for Time Axes Only" on page 484.

TYPE=AUTO | DISCRETE | LINEAR | TIME | LOG

specifies the type of axis to use.

Default: AUTO

AUTO

Requests that the axis type be automatically determined by the plot or the overlay contents.

DISCRETE

Use a DISCRETE axis if possible. The data for discrete axes can be character or numeric. You can add a DISCRETEOPTS= ( ) option list to customize this axis type.

LINEAR

Use a LINEAR axis if possible. You can add a LINEAROPTS= ( ) option list to customize this axis type.

TIME

Use a TIME axis if possible. Data for this axis must be SAS time, SAS date, or SAS datetime values. You can add a TIMEOPTS= ( ) option list to customize this axis type.

LOG

Use a LOG axis if possible. You can add a LOGOPTS= ( ) option list to customize this axis type.

Interaction: If this option is set to anything other than AUTO, plots within the layout are dropped from the display if their data types or data ranges do not match the axis type requirements. For more information, see "Plot Axis Types Must Agree on Common Axes" on page 400.

Interaction: After the axis type is determined (whether you set a specific type or AUTO is in effect), only options supported by that axis type can be used. For example, if TYPE=TIME, only the general OVERLAY axis options and those available on TIMEOPTS= are supported.

Options for Discrete X or X2 Axes Only

All settings for discrete axes are set as options on the DISCRETEOPTS= option.

DISCRETEOPTS=(*discrete-axis-options*)

specifies one or more *discrete-axis-options* for a discrete axis. The *discrete-axis-options* can be set only for an X or X2 axis. A Y or Y2 axis uses the default THIN fit policy. Options must be enclosed in parentheses. Each option is specified as a *name* = *value* pair and each pair is space separated.

Interaction: This option is ignored if the axis type is not DISCRETE.

TICKVALUEFITPOLICY=*policy*

specifies a policy for avoiding tick value collision.

Default: ROTATE

The *policy* can be any one of the following:

THIN	Some tick values are removed.
ROTATE	Tick values are rotated 45 degrees.
ROTATETHIN	Attempt ROTATE policy and then THIN policy.
STAGGER	Tick values alternate between two rows.
STAGGERROTATE	Attempt STAGGER policy and then ROTATE policy.
STAGGERTHIN	Attempt STAGGER policy and then THIN policy.
STAGGERTRUNCATE	Attempt STAGGER policy and then TRUNCATE policy.
TRUNCATE	Tick values are shortened when they exceed a certain number of characters.
TRUNCATEROTATE	Attempt TRUNCATE policy and then ROTATE policy.
TRUNCATESTAGGER	Attempt TRUNCATE policy and then STAGGER policy.
TRUNCATETHIN	Attempt TRUNCATE policy and then THIN policy.

Options for Linear Axes Only

LINEAROPTS=(*linear-axis-options*)

specifies one or more options for a numeric interval axis. Options must be enclosed in parentheses. Each option is specified as a *name* = *value* pair and each pair is space separated.

Interaction: This option is ignored if the axis type is not LINEAR.

Linear-Axis-Option	**Description**
INTEGER	Specifies that evenly spaced integer values are used for tick marks.

Linear-Axis-Option	Description
THRESHOLDMAX	Specifies a bias for including one more tick mark at the maximum end of the axis.
THRESHOLDMIN	Specifies a bias for including one more tick mark at the minimum end of the axis.
TICKVALUEFITPOLICY	Specifies a policy for avoiding tick value collision. Only the default policy (THIN) is available for a Y or Y2 axis.
TICKVALUEFORMAT	Specifies how to format the values for major tick marks.
TICKVALUELIST	Specifies the order of the tick values for a linear axis as list.
TICKVALUEPRIORITY	Specifies whether an axis tick specification can extend the axis data range.
TICKVALUESEQUENCE	Specifies the tick values for a linear axis by start, end, and increment.
VIEWMAX	Specifies the maximum data value to include in the display.
VIEWMIN	Specifies the minimum data value to include in the display.

INTEGER=*boolean*

specifies that evenly spaced integer values are used for tick marks.

Default: FALSE

Interaction: This option is overridden by the TICKVALUELIST= or TICKVALUESEQUENCE= option.

Interaction: This option overrides the MAXDECIMALS= and PREFERREDDECIMALS= suboptions of the TICKVALUEFORMAT= option

THRESHOLDMAX= *number*

specifies a bias for including one more tick mark at the maximum end of the axis. For more information, see "Adjusting Axis Thresholds" on page 402.

Default: .30

Range: 0 - 1

Interaction: This option is ignored if the TICKVALUELIST= or TICKVALUESEQUENCE= option is used.

THRESHOLDMIN= *number*

specifies a bias for including one more tick mark at the minimum end of the axis. For more information, see "Adjusting Axis Thresholds" on page 402.

Default: .30

Range: 0 - 1

Interaction: This option is ignored if the TICKVALUELIST= or TICKVALUESEQUENCE= option is used.

TICKVALUEFITPOLICY=THIN | *policy*
specifies a policy for avoiding tick value collision. Only the default setting (THIN) is available for a Y or Y2 axis.

Default: THIN

Restriction: THIN is the only policy supported for the Y or Y2 axis.

If more than one policy is used, the values in the list are formatted according to the TICKVALUEFORMAT= option.

The *policy* can be any one of the following policies:

THIN	Eliminate alternate tick values. Only available on COLUMNAXIS.
ROTATE	Tick values are rotated 45 degrees.
ROTATETHIN	Attempt ROTATE policy and then THIN policy.
STAGGER	Tick values alternate between two rows.
STAGGERROTATE	Attempt STAGGER policy and then ROTATE policy.
STAGGERTHIN	Attempt STAGGER policy and then THIN policy.

TICKVALUEFORMAT=(*format-options*) | DATA | *format*
specifies how to format the values for major tick marks.

Default: (MAXWIDTH=8, MAXDECIMALS=6, PREFERREDDECIMALS=2, EXTRACTSCALE=FALSE)

(*format-options*)
specifies one or more major tick value formatting options. Together, these options provide parameters for determining an optimal format (*w.d*, E*w*., BEST*w*.) for displaying major tick values.

MAXWIDTH = *integer*	Specifies the maximum width for displayed tick values. Values might be rounded or converted to E-notation to fit into this width.
MAXDECIMALS = *integer*	Specifies the maximum number of decimals for displayed tick values. Values might be rounded or converted to E-notation to fit into this width. MAXWIDTH > MAXDECIMALS
PREFERREDDECIMALS = *integer*	Specifies the number of decimal places desired for most values. The actual number might vary based on other constraints.
EXTRACTSCALE = *boolean*	Specifies whether to extract some factor of ten from all tick values in order to

reduce the overall width of the tick values and improve legibility. The extracted factor is concatenated to the existing axis label. Example of revised axis label: *Number Sold (x1000)*

DATA
use the format that has been assigned to the variable(s) contributing to the axis (or BEST6. if no format is assigned) to control the formatting of the major tick values.

format
use this format to control how the major tick values are formatted.

Restriction: The *format-options* are used only when no format has been assigned to the variable(s) that contribute to the axis or when the assigned format(s) are *w.d*, E*w*. or BEST*w*.

Restriction: GTL currently honors most but not every SAS format. For details, see "SAS Formats Not Supported" on page 569.

Interaction: If the axis label is not displayed, the EXTRACTSCALE=TRUE *format-option* is ignored.

TICKVALUELIST=(*numeric-list*)
specifies the tick values for a linear axis as list.

Default: An internal algorithm determines the tick marks based on the actual axis data range or the data range established by the VIEWMIN= and VIEWMAX= options. By default, when this option is used, only tick values that appear are those in *numeric-list* that fall within the explicit data range (set by VIEWMIN= and VIEWMAX=) or implicit data range (set by the actual data minimum and data maximum).

Requirement: The *numeric-list* must be enclosed in parentheses and each item separated by a blank.

Interaction: This option overrides the INTEGER= option.

Interaction: This option is ignored if the TICKVALUESEQUENCE= option is specified, or if the tick values are not enabled by the ODS style or the DISPLAY= or DISPLAYSECONDARY= option.

Interaction: The VIEWMIN= and VIEWMAX= options alter the axis data range. If the VIEWMIN= option is set to the minimum tick list value and the VIEWMAX= option is set to the maximum tick list value, all ticks in the tick list are displayed. This might result in some data not being displayed. For example, data might not be displayed when the VIEWMIN= value is greater than the actual data minimum, or when the VIEWMAX= value is less than actual data maximum.

Interaction: If the TICKVALUEPRIORITY= is set to TRUE, 1) the VIEWMIN= and VIEWMAX= options are ignored if they are fully enclosed by the *numeric-list* 2) the tick *numeric-list* can extend the implicit data range of the axis, but never reduce it.

The values in the list are formatted according to the setting for the TICKVALUEFORMAT= option.

TICKVALUEPRIORITY=*boolean*
specifies whether an axis tick specification (TICKVALUELIST= or TICKVALUESEQUENCE=) can extend the axis data range.

Default: FALSE

FALSE

the only tick values to appear are the user-specified tick values that fall within the explicit data range (set by VIEWMIN= and VIEWMAX=) or implicit data range (set by the actual data minimum and data maximum).

TRUE

the axis data range might be extended (but not reduced) to include the minimum and maximum specified by either the TICKVALUELIST= or TICKVALUESEQUENCE= option. If minimum and maximum of the user-specified values are within the data range, this option has no effect.

Interaction: When this option is set to TRUE, the VIEWMIN= and VIEWMAX= options are ignored if they are fully enclosed by value range set by the TICKVALUELIST= or TICKVALUESEQUENCE= option.

Interaction: For this option to have any effect, the tick marks must be enabled by the ODS style or the DISPLAY= or DISPLAYSECONDARY= option.

TICKVALUESEQUENCE=(*sequence-options*)

specifies the tick values for a linear axis by start, end, and increment.

Default: An internal algorithm determines the tick marks based on the actual axis data range or the data range established by the VIEWMIN= and VIEWMAX= options. By default when this option is used, the only tick values to appear are those that fall within the explicit data range (set by VIEWMIN= and VIEWMAX=) or implicit data range (set by the actual data minimum and data maximum).

(*sequence-options*)

Three options that control major tick values. All three options must be provided.

START = *number*	Specifies the value for the first tick mark.
END = *number*	Specifies the value for the last tick mark.
INCREMENT = *number*	Specifies the increment for intermediate tick marks between the first and last tick marks. The END value always controls the last tick mark. The interval between the last tick mark and the previous tick mark might not necessarily be the INCREMENT value.

Interaction: This option overrides the INTEGER= option.

Interaction: The VIEWMIN= and VIEWMAX= options alter the axis data range. If the VIEWMIN= option is set to the START= option value and the VIEWMAX= option is set to the END= option value, all ticks in the tick sequence are displayed.

Interaction: If the TICKVALUEPRIORITY= option is set to TRUE, the tick sequence might extend the explicit data range of the axis, but never reduce it.

Interaction: For this option to have any effect, the tick marks must be enabled by the ODS style or the DISPLAY= or DISPLAYSECONDARY= option.

The values in the sequence are formatted according to the setting for the TICKVALUEFORMAT= option.

See also: TICKVALUELIST= option as an alternative for customizing tick marks.

VIEWMAX= *number*

specifies the maximum data value to include in the display (the value might be adjusted by the threshold calculation). For more information, see “Adjusting the Axis View” on page 402.

Default: The maximum value in the data for the specified axis.

Interaction: This option does not determine the maximum axis tick value displayed. The THRESHOLDMAX= value is used to determine the maximum tick value.

Interaction: This option has no effect if the TICKVALUEPRIORITY= is set to TRUE.

Setting a VIEWMAX= or VIEWMIN= value does not alter the original data or any calculations on it.

VIEWMIN= *number*

specifies the minimum data value to include in the display (the value might be adjusted by the threshold calculation). For more information, see "Adjusting the Axis View" on page 402.

Default: The minimum value in the data for the specified axis.

Interaction: This option does not determine the minimum axis tick value displayed. The THRESHOLDMIN= value is used to determine the maximum tick value.

Interaction: This option has no effect if the TICKVALUEPRIORITY= is set to TRUE.

Setting a VIEWMAX= or VIEWMIN= value does not alter the original data or any calculations on it.

Options for Log Axes Only

LOGOPTS=(*log-axis-options*)

specifies one or more options for a log axis. Options must be enclosed in parentheses. Each option is specified as a *name* = *value* pair separated by a blank.

Interaction: This option is ignored if the axis type is not LOG.

Log-Axis-Option	Description
BASE	Specifies the base of the logarithmic scale for the axis values.
MINORTICKS	Specifies whether minor ticks are displayed.
TICKINTERVALSTYLE	Specifies how to scale and format the values for major tick marks.
VIEWMAX	Specifies the maximum data value to include in the display.
VIEWMIN	Specifies the minimum data value to include in the display.

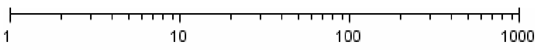
BASE=10 | 2 | E

specifies the base of the logarithmic scale for the axis values.

Default: 10

MINORTICKS=*boolean*

specifies whether minor ticks are displayed:



Default: FALSE

Restriction: Minor ticks can be displayed only when BASE=10 and TICKINTERVALSTYLE= is LOGEXPAND or LOGEXPONENT.

If minor ticks are displayed, the number of minor tick marks is automatically determined.

TICKINTERVALSTYLE=AUTO | LOGEXPAND | LOGEXPONENT | LINEAR
specifies how to scale and format the values for major tick marks.

Default: AUTO

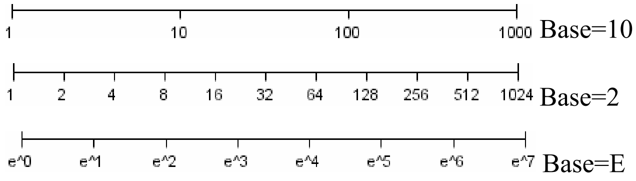
Restriction: For LOGEXPAND and LOGEXPONENT, formats on data columns contributing to the axis are ignored. For LINEAR, ticks values are automatically formatted when the column format is not assigned or one of *w.d*, *Ew.*, or *BESTw*. Other formats (SAS defined or user-defined) are used if specified.

Restriction: GTL currently honors most but not every SAS format. For details, see "SAS Formats Not Supported" on page 569.

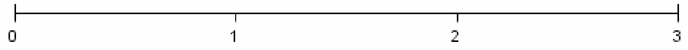
If you use TICKINTERVALSTYLE=LOGEXPONENT, you might want to include information in the axis label about which base is used.

AUTO
A LOGEXPAND, LOGEXPONENT, or LINEAR representation is chosen automatically based on the range of the data. When the data range is small (within an order of magnitude), a LINEAR representation is typically used. Data ranges that encompass several orders of magnitude typically use the LOGEXPAND or LOGEXPONENT representation.

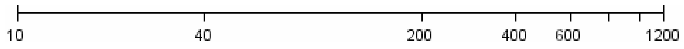
LOGEXPAND
Major ticks are placed at uniform intervals at integer powers of the base. The tick values are expanded as follows:



LOGEXPONENT
Major ticks are placed at uniform intervals at integer powers of the base. The tick values are only the integer exponents for all bases.



LINEAR
Major tick marks are placed at non-uniform intervals that cover the range of the data.



Restriction: For LOGEXPONENT, formats on data columns contributing to the axis are ignored. For LOGEXPAND, formats on data columns contributing to the axis are ignored, although any "named format" on the column is retained. For LINEAR, ticks values are automatically formatted when the column format is not assigned or one of *w.d*, *Ew.* or *BESTw*. Other formats (SAS defined or user-defined) are used if specified.

Restriction: GTL currently honors most but not every SAS format. For details, see "SAS Formats Not Supported" on page 569.

Interaction: When BASE=10 and LOGEXPAND or LOGEXPONENT is used, an intermediate tick is added whenever the axis data range is less than or equal to 1.5 powers of 10.

If you use TICKINTERVALSTYLE=LOGEXPONENT, you might want to include information in the axis label about which base is used.

VIEWMAX= *number*

specifies the maximum data value to include in the display. For more information, see "Adjusting the Axis View" on page 402.

Default: The maximum value in the data for the specified axis.

Setting a VIEWMAX= or VIEWMIN= value does not alter the original data or any calculations on it.

VIEWMIN= *number*

specifies the minimum data value to include in the display. For more information, see "Adjusting the Axis View" on page 402.

Default: The minimum value in the data for the specified axis.

Setting a VIEWMAX= or VIEWMIN= value does not alter the original data or any calculations on it.

Options for Time Axes Only

TIMEOPTS=(*time-axis-options*)

specifies one or more options for a time axis.

Variables associated with a time axis must be in SAS time, SAS date, or SAS datetime units and have an associated SAS time, date, or datetime format.

Options must be enclosed in parentheses. Each option is specified as a *name = value* pair separated by a blank.

Interaction: This option is ignored if the axis type is not TIME.

Time-Axis-Option	Description
INTERVAL	Specifies that evenly spaced integer values are used for tick marks.
MINORTICKS	Specifies whether minor ticks are displayed.
SPLITTICKVALUE	Specifies whether to split the tick values on an X or X2 axis, if possible. This option is not available on a Y or Y2 axis.
TICKVALUEFITPOLICY	Specifies a policy for avoiding tick value collision on an X or X2 axis. This option is not available on a Y or Y2 axis.
TICKVALUEFORMAT	Specifies how to format the values for major tick marks.
TICKVALUELIST	Specifies the order of the tick values for a time axis as list.

Time-Axis-Option	Description
VIEWMAX	Specifies the maximum data value to include in the display.
VIEWMIN	Specifies the minimum data value to include in the display.

INTERVAL=*interval*

specifies the time interval between major ticks. Valid *interval* keywords are AUTO, SECOND, MINUTE, HOUR, DAY, TENDAY, WEEK, SEMIMONTH, MONTH, QUARTER, SEMIYEAR, YEAR.

The data column(s) mapped to a time axis must be in the same duration units: TIME, DATE, or DATETIME. The selection of an interval must be consistent with the duration unit. For example, if the data are in time units, you can select only AUTO, SECOND, MINUTE, HOUR.

Default: AUTO. An appropriate interval is chosen based on the data and the column date/datetime/time format.

Interaction: This option is ignored if the TICKVALUELIST= option is used.

INTERVAL	Unit	Tick interval	Default tick value format
AUTO	DATE, TIME, or DATETIME	automatically chosen	automatically chosen
SECOND	TIME or DATETIME	second	TIME8.
MINUTE	TIME or DATETIME	minute	TIME8.
HOUR	TIME or DATETIME	hour	TIME8.
DAY	DATE or DATETIME	day	DATE9.
TENDAY	DATE or DATETIME	10 days	DATE9.
WEEK	DATE or DATETIME	7 days	DATE9.
SEMIMONTH	DATE or DATETIME	1st and 16th of each month	DATE9.
MONTH	DATE or DATETIME	month	MONYY7.
QUARTER	DATE or DATETIME	3 months	YYQC6.

INTERVAL	**Unit**	**Tick interval**	**Default tick value format**
SEMIYEAR	DATE or DATETIME	6 months	MONYY7.
YEAR	DATE or DATETIME	year	YEAR4.

MINORTICKS=Boolean

specifies whether minor ticks are displayed.

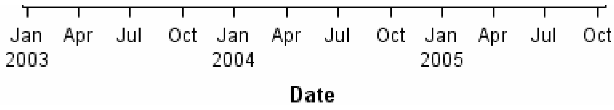
Default: FALSE

Interaction: The number of minor ticks is dependent on the value of the INTERVAL= option.

Interaction: This option is ignored if the TICKVALUELIST= option is used or if the tick marks are not enabled by the DISPLAY= or DISPLAYSECONDARY= option.

SPLITTICKVALUE=Boolean

specifies whether to split the tick values on an X or X2 axis, if possible. This option is not available for a Y or Y2 axis. For example, with INTERVAL= MONTH, this is how tick values are split:



Default: TRUE

TRUE

axis tick values are split into two lines allowing more tick values to appear

FALSE

typically, fewer tick values fit, causing thinning, rotation, or staggering of the values. See the TICKVALUEFITPOLICY= option.

Interaction: This option is ignored if the TICKVALUELIST= or TICKVALUEFORMAT= option is used.

TICKVALUEFITPOLICY=THIN | ROTATE | ROTATETHIN | STAGGER | STAGGERROTATE | STAGGERTHIN

specifies a policy for avoiding tick value collision on an X or X2 axis. This option is not available for a Y or Y2 axis.

Default: THIN

The collision-avoidance policy that is most effective depends on the number of tick values, their length, and the length of the axis. For an X or X2 axis, the policy can be one of the following:

THIN	Some tick values are removed.
ROTATE	Tick values are rotated 45 degrees.
ROTATETHIN	Attempt ROTATE policy and then THIN policy.
STAGGER	Tick values alternate between two rows.

STAGGERROTATE	Attempt STAGGER policy and then ROTATE policy.
STAGGERTHIN	Attempt STAGGER policy and then THIN policy.

Interaction: When SPLITTICKVALUE= TRUE, this option is ignored and only the THIN policy is used.

TICKVALUEFORMAT=*format* | DATA

specifies how to format the values for major tick marks.

Default: The default format used by the INTERVAL= option. The default does not apply if TICKVALUELIST= is specified.

format

A SAS date, time, or datetime format to control how the major tick values are displayed. This format must be in the same duration units as the data column(s) mapped to a time axis: TIME, DATE, or DATETIME and should be appropriate for the value of the INTERVAL= option. For example, if INTERVAL=MONTH and there are two years of data displayed on the axis, choosing TICKVALUEFORMAT=YEAR. would result in several ticks having the same year value.

DATA

Specifies that the SAS date, time, or datetime format associated with the data column assigned to the axis be used to control how the major tick values are displayed.

Restriction: GTL currently honors most but not every SAS format. For details, see "SAS Formats Not Supported" on page 569.

Interaction: If this option is specified, the SPLITTICKVALUE= option is ignored.

TICKVALUELIST=(*time-constant-list* | *date-constant-list* | *datetime-constant-list* | *numeric-list*)

specifies the tick values for a time axis as list.

Default: An internal algorithm determines the tick values.

Requirement: The list must be enclosed in parentheses and each item separated by a blank. The items in the list must be in the same duration units as the data mapped to the axis: TIME, DATE, or DATETIME. The values can be expressed as SAS TIME, DATE, or DATETIME constants (for example, "13:23"T, "11MAY06"D, or "11MAY06:13:23"DT) or their numeric equivalents.

Restriction: This option does not extend the data range of the axis. If the values fall within the default data range or that specified by the VIEWMIN= or VIEWMAX= options, they are used.

The values in the list are formatted according to the setting for the TICKVALUEFORMAT= option. If TICKVALUEFORMAT= is not used, the values are formatted according to the column format (the default TICKVALUEFORMAT value is not applied to these values).

See also: TICKVALUESEQUENCE= option as an alternate to customizing tick marks.

VIEWMAX= *number*

specifies the maximum data value to include in the display. For more information, see "Adjusting the Axis View" on page 402.

Default: The maximum value in the data for the specified axis.

Setting a VIEWMAX= or VIEWMIN= value does not alter the original data or any calculations on it.

VIEWMIN= *number*

specifies the minimum data value to include in the display. For more information, see "Adjusting the Axis View" on page 402.

Default: The minimum value in the data for the specified axis.

Setting a VIEWMAX= or VIEWMIN= value does not alter the original data or any calculations on it.

Part 6

Legend Statements

Chapter 47

CONTINUOUSLEGEND Statement

Description and Syntax

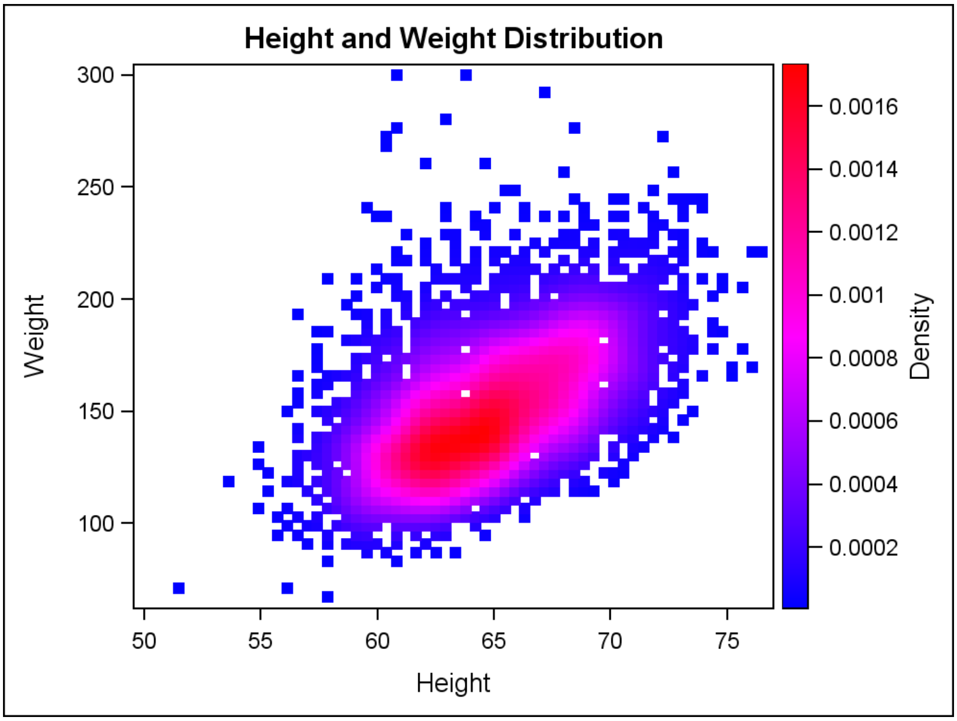
Creates a legend with a color ramp corresponding to a range of values.

CONTINUOUSLEGEND “*graph-name*” *</option(s)>*;

Example Program and Statement Details

Example Graph

The following graph was generated by the “Example Program” on page 492:



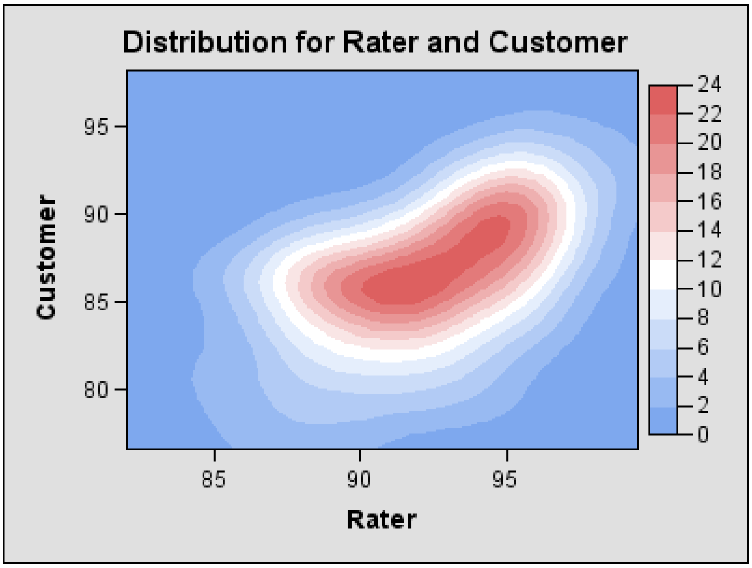
Example Program

```
proc template;
  define statgraph continuouslegend;
    begingraph;
      entrytitle "Height and Weight Distribution";
        layout overlay;
          scatterplot x=height y=weight /
            markercolorgradient=density
            markerattrs=(symbol=squarefilled size=6px)
            name="scatter";
          continuouslegend "scatter" / orient=vertical
            location=outside valign=center halign=right
            valuecounthint=10 title="Density";
        endlayout;
    endgraph;
  end;
run;
proc sgrender data=sashelp.gridded(where=(count>0))
  template=continuouslegend;
run;
```

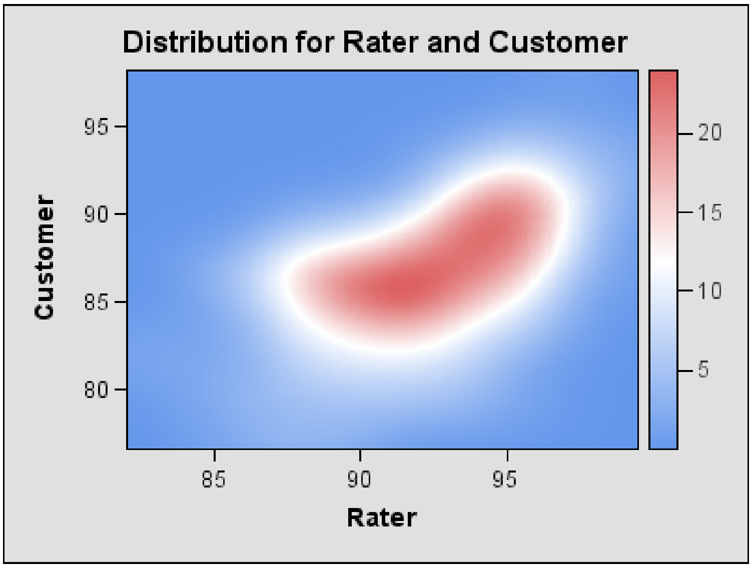
Statement Summary

A continuous legend consists of a color ramp and a numeric scale indicating color values.

In the following figure, the continuous legend references a contour plot with a fixed number of levels. The color ramp and legend values automatically reflect these discrete levels. The legend options VALUECOUNT= and VALUECOUNTHINT= have no effect.



In this next figure, the continuous legend references a contour plot with a continuous gradient. The number of legend values displayed is automatically determined by the legend, or decided by the contributing plot, such as the CONTOURPLOTPARM with CONTOURTYPE=FILL. For plots with a continuous gradient, you can use the legend options VALUECOUNT= or VALUECOUNTHINT= to control how many legend values appear (these options are ignored if there is no gradient).



Within an overlay-type layout, when a continuous legend is placed inside the plot area with LOCATION=INSIDE,

- It is always placed on top of plot lines and markers.
- By default, its background is fully transparent (OPAQUE=FALSE), meaning that underlying lines, markers, and data labels show through the legend.
- Its position can be controlled with the AUTOALIGN= option, or with the HALIGN= and VALIGN= options.

Within an overlay-type layout, when a continuous legend is placed outside the plot area with LOCATION=OUTSIDE,

- By default, its background is fully opaque (OPAQUE=TRUE).
- Its position can be controlled with the HALIGN= and VALIGN= options.

Required Argument

"*graph-name*"
specifies the plot to be represented by the legend. The plot is identified by the name that is assigned to it on the plot statement's NAME= option.

Requirement: *graph-name* must be enclosed in quotation marks.

Note: Unlike the DISCRETELEGEND statement, only a single *graph-name* can be specified.

Options

Statement Option	Description
AUTOALIGN	Specifies whether the legend is automatically aligned within its parent layout when nested within an overlay-type layout.
BACKGROUNDCOLOR	Specifies the color of the legend background.
BORDER	Specifies whether a border is displayed around the legend.
BORDERATTRS	Specifies the properties of the border line around the legend.
HALIGN	specifies the horizontal alignment of the legend within its parent layout when nested within an overlay-type layout.
LOCATION	Specifies whether the legend appears inside or outside the plot area.
OPAQUE	Specifies whether the legend background is opaque.
ORIENT	Specifies the orientation of the legend.
PAD	Specifies the amount of extra space that is added inside the legend perimeter.
TITLE	Specifies the title of the legend.
TITLEATTRS	Specifies the color and font attributes of the legend title.
VALIGN	Specifies the vertical alignment of the legend within its parent layout when nested within an overlay-type layout.

Statement Option	Description
VALUEATTRS	Specifies the color and font attributes of the legend values.
VALUECOUNT	Specifies the number of values that the continuous legend must use to label the data range. Available only for plots that display a gradient.
VALUECOUNTHINT	Recommends a number of values for the continuous legend to use to label the data range. Available only for plots that display a gradient.

AUTOALIGN=NONE | AUTO | *(location-list)*

specifies whether the legend is automatically aligned within its parent layout when nested within an overlay-type layout. For more information about how child positions are determined in an overlay-type layout, see the LAYOUT OVERLAY's "Example Program and Statement Details" on page 89.

Default: NONE

NONE

Do not automatically align the legend within its parent layout. The legend's position is set by the HALIGN= and VALIGN= options.

AUTO

Available only if the parent layout contains a scatter plot; ignored otherwise. Within the parent layout, attempt to center the legend in the area that is farthest from any surrounding data point markers.

(*location-list*)

Within the parent layout, restrict the legend's possible locations to those locations in the specified *location-list*, and use the *location-list* position that least collides with the parent layout's other graphics features. The *location-list* is blank-separated and can contain any of these locations: TOPLEFT, TOP, TOPRIGHT, LEFT, CENTER, RIGHT, BOTTOMLEFT, BOTTOM, and BOTTOMRIGHT.

Interaction: This option has no effect unless LOCATION=INSIDE.

Interaction: If LOCATION=INSIDE, this option overrides HALIGN= and VALIGN=.

BACKGROUNDCOLOR=*style-reference* | *color*

specifies the color of the legend background.

Default: The GraphLegendBackground:Color style reference.

style-reference

A reference of the form *style-element:style-attribute*. Only the style-attribute named COLOR is used.

Interaction: OPAQUE=TRUE must be in effect for the color to be seen. By default, OPAQUE=FALSE.

BORDER=*boolean*

specifies whether a border is displayed around the legend.

Default: FALSE

BORDERATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)

specifies the attributes of the border line around the legend. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Line Options" on page 564 for available *line-options*.

Default: The GraphBorderLines style element.

Interaction: BORDER=TRUE must be set for this option to have any effect.

HALIGN= RIGHT | CENTER | LEFT

specifies the horizontal alignment of the legend within its parent layout when nested within an overlay-type layout. For more information about how child positions are determined in an overlay-type layout, see the LAYOUT OVERLAY's "Example Program and Statement Details" on page 89.

Default: RIGHT if LOCATION=OUTSIDE CENTER if LOCATION=INSIDE

Restriction: If LOCATION=OUTSIDE, the HALIGN= and VALIGN= options cannot both be set to CENTER.

Restriction: This option is available only when this statement is nested within an overlay-type layout. It is ignored if the parent layout is not an overlay-type layout.

Interaction: If LOCATION=INSIDE and the AUTOALIGN= option is enabled, this option is ignored.

LOCATION=OUTSIDE | INSIDE

specifies whether the legend appears inside or outside the plot area.

Default: OUTSIDE

Restriction: This option has effect only when the CONTINUOUSLEGEND statement appears within a 2-D overlay-type layout and there is at least one stand-alone plot statement with XY axes that is referenced by the legend. For more information about how child positions are determined in an overlay-type layout, see the LAYOUT OVERLAY's "Example Program and Statement Details" on page 89.

Interaction: The actual position is determined by the settings for the LOCATION=, AUTOALIGN=, HALIGN= , and VALIGN= options.

Interaction: Within an overlay-type layout, if the ORIENT= option is not set, the orientation changes depending on the actual position. If LOCATION=OUTSIDE and the legend is right or left of the plot, the orientation is vertical. If LOCATION=OUTSIDE and the legend is above or below the plot, the orientation is horizontal.

OPAQUE=*boolean*

specifies whether the legend background is opaque (TRUE) or transparent (FALSE).

Default: FALSE

Interaction: When this option is set to FALSE, the background color is not used.

ORIENT=VERTICAL | HORIZONTAL

specifies the orientation of the legend.

Default: VERTICAL

PAD=*dimension* | *(pad-options)*

specifies the amount of extra space that is reserved inside the legend perimeter.

Default: The default padding for all sides is 0. Values without units are in pixels (px). A unit must be provided if other than pixels.

dimension

Specifies a dimension to use for the extra space at the left, right, top, and bottom of the legend perimeter.

(*pad-options*)

Enables separate settings for the left, right, top, and bottom padding dimensions. Use the *pad-options* to create non-uniform padding. These options must be enclosed in parentheses. Each option is specified as a *name* = *value* pair. Sides not assigned padding are padded with the default amount.

Values without units are in pixels (px). A unit must be provided if other than pixels.

LEFT=*dimension*	specifies the amount of extra space added to the left side.
RIGHT=*dimension*	specifies the amount of extra space added to the right side.
TOP=*dimension*	specifies the amount of extra space added to the top.
BOTTOM=*dimension*	specifies the amount of extra space added to the bottom.

TITLE= *"string"*

specifies the title of the legend.

Default: no default

Requirement: string must be enclosed in quotation marks.

TITLEATTRS=*style-element* | *style-element* (*text-options*) | (*text-options*)

specifies the color and font attributes of the legend title. See “General Syntax for Attribute Options” on page 563 for the syntax on using a *style-element* and “Text Options” on page 566 for available *text-options*.

Default: The GraphLabelText style element.

Interaction: For this option to have any effect, the TITLE= option must also be specified.

VALIGN=CENTER | TOP | BOTTOM

specifies the vertical alignment of the legend within its parent layout when nested within an overlay-type layout. For more information about how child positions are determined in an overlay-type layout, see the LAYOUT OVERLAY’s “Example Program and Statement Details” on page 89.

Default: CENTER

Restriction: If LOCATION=OUTSIDE, the VALIGN= and HALIGN= options cannot both be set to CENTER.

Restriction: This option is available only when this statement is nested within an overlay-type layout. It is ignored if the parent layout is not an overlay-type layout.

Interaction: If LOCATION=INSIDE and the AUTOALIGN= option is enabled, this option is ignored.

VALUEATTRS=*style-element* | *style-element* (*text-options*) | (*text-options*)

specifies the color and font attributes of the legend values. See “General Syntax for Attribute Options” on page 563 for the syntax on using a *style-element* and “Text Options” on page 566 for available *text-options*.

Default: The GraphValueText style element.

VALUECOUNT=*positive-integer*

specifies the number of values that the continuous legend must use to label the data range.

Default: The VALUECOUNTHINT= setting is used.

Restriction: The associated plot must be displayed with smooth gradient for this option to have any effect. For example, in a contour plot, CONTOURTYPE must be set to GRADIENT or LINEGRADIENT.

The legend uses even intervals to label the range.

VALUECOUNTHINT=*positive-integer*

recommends a number of values for the continuous legend to use to label the data range.

Default: 6

Restriction: The associated plot must be displayed with smooth gradient for this option to have any effect. For example, in a contour plot, CONTOURTYPE must be set to GRADIENT or LINEGRADIENT.

The legend uses even intervals to label the range.

Chapter 48

DISCRETELEGEND Statement

Description and Syntax

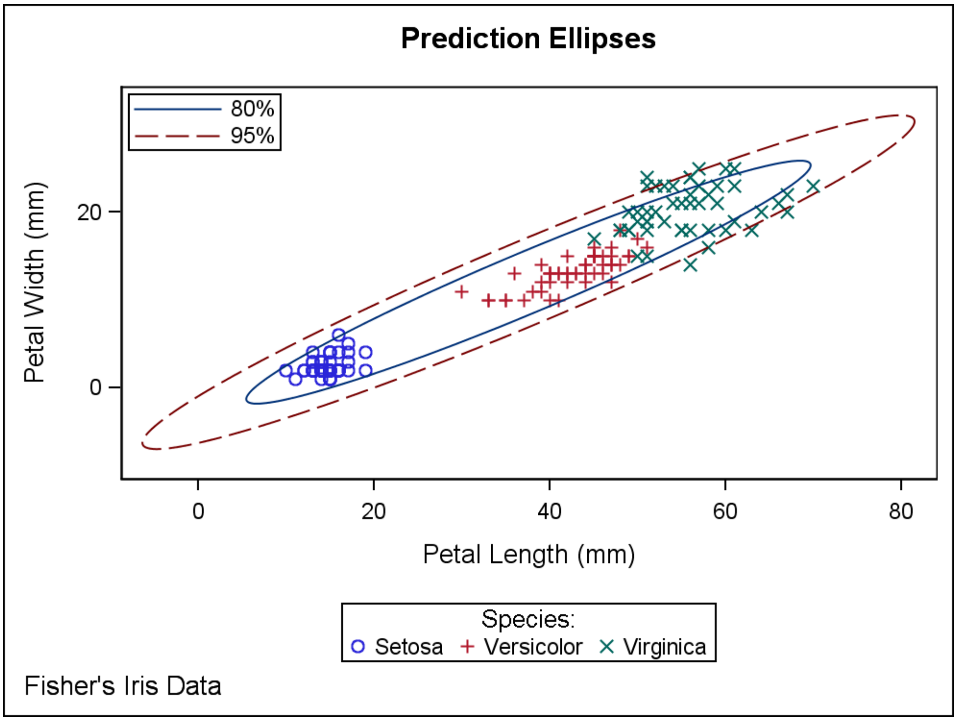
Creates a legend with entries that refer to plots, or group values, or both.

DISCRETELEGEND "*graph-name*" <... "*graph-name-n*"> </*option(s)*>;

Example Program and Statement Details

Example Graph

The following graph was generated by the "Example Program" on page 500:



Example Program

```
proc template;
  define statgraph discretelegend;
    begingraph;
     entrytitle "Prediction Ellipses";
     layout overlayequated / equatetype=equate;
      scatterplot x=petallength y=petalwidth /
        group=species name="s";
      ellipse x=petallength y=petalwidth /
        type=predicted  alpha=.2
        name="p80" legendlabel="80%"
        outlineattrs=graphconfidence;
      ellipse x=petallength y=petalwidth /
        type=predicted  alpha=.05
        name="p95" legendlabel="95%"
        outlineattrs=graphconfidence2;
      discretelegend "s" / title="Species:";
      discretelegend "p80" "p95" / across=1
        autoalign=(topleft) location=inside;
     endlayout;
     entryfootnote halign=left "Fisher's Iris Data";
    endgraph;
  end;
run;

proc sgrender data=sashelp.iris template=discretelegend;
run;
```

Statement Summary

A discrete legend consists of one or more units called legend entries. Each legend entry consists of a legend symbol and a legend value. The legend symbol is typically a marker, line, or fill that represents a specific area in the plot. The legend value is descriptive text derived from group values or assigned with the plot's LEGENDLABEL= option.

Within an overlay-type layout, when a discrete legend is placed inside the plot area with LOCATION=INSIDE,

- It is always placed on top of plot lines and markers.
- By default, its background is fully transparent (OPAQUE=FALSE), meaning that underlying lines, markers, and data labels show through the legend.
- Its position can be controlled with the AUTOALIGN= option, or with the HALIGN= and VALIGN= options.

Within an overlay-type layout, when a discrete legend is placed outside the plot area with LOCATION=OUTSIDE,

- By default, its background is fully opaque (OPAQUE=TRUE).
- Its position can be controlled with the HALIGN= and VALIGN= options.

When a discrete legend is placed within nested layouts, it might be necessary to use the ACROSS= and ORDER=ROWMAJOR options or the DOWN= and ORDER=COLUMNMAJOR options to obtain the desired legend organization. A legend might be dropped if the total legend area in the graph exceeds the percentage set by the MAXLEGENDAREA= option on an ODS GRAPHICS ON statement that is in effect for the output destination.

Required Arguments

"*graph-name*"
: references one or more unique names that are specified by the NAME= option of a plot statement. These names control which plots contribute to the legend, and the order in which the legend entries occur. If a contributing plot uses a GROUP= option, then there is a legend entry for each group value.

  - Each *graph-name* must be enclosed in quotation marks.
  - Each plot that is referenced must be able to be identified by the legend entries. For example, a filled CONTOURPLOTPARM plot *cannot* be specified in the DISCRETELEGEND statement because it requires a continuous legend.

Options

Statement Option	**Description**
ACROSS	Specifies the number of legend entries that are placed horizontally before the next row begins.
AUTOALIGN	Specifies whether the legend is automatically aligned within its parent layout when nested within an overlay-type layout.

Statement Option	Description
BACKGROUNDCOLOR	Specifies the color of the legend background.
BORDER	Specifies whether a border is displayed around the legend.
BORDERATTRS	Specifies the properties of the border line around the legend.
DISPLAYCLIPPED	Specifies whether the legend is displayed when any portion of the legend is not entirely rendered.
DOWN	Specifies the number of legend entries that are placed vertically before the next column begins.
HALIGN on page 504	Specifies the horizontal alignment of the legend within its parent layout when nested within an overlay-type layout.
LOCATION on page 504	Specifies whether the legend appears inside or outside the plot area when the legend is specified within an overlay-type layout.
OPAQUE	Specifies whether the legend background is opaque.
ORDER	Specifies the order of the legend entries.
PAD	Specifies the amount of extra space that is added inside the legend perimeter.
TITLE	Specifies the title of the legend.
TITLEATTRS	Specifies the color and font attributes of the legend title.
TITLEBORDER	Specifies a border around the legend title that separates the title from the legend body.
VALIGN	Specifies the vertical alignment of the legend within its parent layout when nested within an overlay-type layout.
VALUEATTRS	Specifies the color and font attributes of the legend values.

ACROSS=*positive-integer*

specifies the number of legend entries that are placed horizontally before the next row begins.

Default: The entries are placed to best fit the available area. This “best fit” approach only works when the legend is nested in the template’s outermost layout.

Restriction: This option is ignored if ORDER=COLUMNMAJOR

A legend entry typically consists of two parts, such as a marker symbol and an associated value.

AUTOALIGN=NONE | AUTO | *(location-list)*

specifies whether the legend is automatically aligned within its parent layout when nested within an overlay-type layout. For more information about how child positions are determined in an overlay-type layout, see the LAYOUT OVERLAY's "Example Program and Statement Details" on page 89.

Default: NONE

NONE

Do not automatically align the legend within its parent layout. The legend's position is set by the HALIGN= and VALIGN= options.

AUTO

Available only if the parent layout contains a scatter plot; ignored otherwise. Within the parent layout, attempt to center the legend in the area that is farthest from any surrounding data point markers.

(*location-list*)

Within the parent layout, restrict the legend's possible locations to those locations in the specified *location-list*, and use the *location-list* position that least collides with the parent layout's other graphics features. The *location-list* is blank-separated and can contain any of these locations: TOPLEFT, TOP, TOPRIGHT, LEFT, CENTER, RIGHT, BOTTOMLEFT, BOTTOM, and BOTTOMRIGHT.

Interaction: This option has no effect unless LOCATION=INSIDE.

Interaction: If LOCATION=INSIDE, this option overrides HALIGN= and VALIGN=.

BACKGROUNDCOLOR=*style-reference* | *color*

specifies the color of the legend background.

Default: The GraphLegendBackground:Color style reference.

style-reference

A reference of the form *style-element:style-attribute*. Only the style-attribute named COLOR is used.

Interaction: OPAQUE=TRUE must be in effect for the color to be seen. By default, OPAQUE=FALSE.

BORDER=*boolean*

specifies whether a border is displayed around the legend.

Default: TRUE

BORDERATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)

specifies the attributes of the border line around the legend. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Line Options" on page 564 for available *line-options*.

Default: The GraphBorderLines style element.

Interaction: BORDER=TRUE must be set for this option to have any effect.

DISPLAYCLIPPED=*boolean*

specifies whether the legend is displayed when any portion of the legend is not entirely rendered.

Default: FALSE

When the graph size is reduced, parts of the legend (title, legend symbol, or legend value) might be clipped (truncated). When clipping occurs and DISPLAYCLIPPED is FALSE, the entire legend is removed from the graph and the space for it is reclaimed by the remainder of the graph. When DISPLAYCLIPPED is TRUE, the legend always appears, even if some parts of have been clipped.

DOWN=*positive-integer*

specifies the number of legend entries that are placed vertically before the next column begins.

Default: The entries are placed to best fit the available area. This “best fit” approach only works when the legend is nested in the template’s outermost layout.

Restriction: This option is ignored if ORDER=ROWMAJOR

A legend entry typically consists of two parts, such as a marker symbol and an associated value.

HALIGN=CENTER | LEFT | RIGHT

specifies the horizontal alignment of the legend within its parent layout when nested within an overlay-type layout. For more information about how child positions are determined in an overlay-type layout, see the LAYOUT OVERLAY’s “Example Program and Statement Details” on page 89.

Default: CENTER

Restriction: If LOCATION=OUTSIDE, the HALIGN= and VALIGN= options cannot both be set to CENTER.

Restriction: This option is available only when this statement is nested within an overlay-type layout. It is ignored if the parent layout is not an overlay-type layout.

Interaction: If LOCATION=INSIDE and the AUTOALIGN= option is enabled, this option is ignored.

LOCATION=OUTSIDE | INSIDE

specifies whether the legend appears inside or outside the plot area.

Default: OUTSIDE

Restriction: This option has effect only when the DISCRETELEGEND statement appears within an overlay-type layout and at least one stand-alone plot statement with XY axes is referenced by the legend. For more information about how child positions are determined in an overlay-type layout, see the LAYOUT OVERLAY’s “Example Program and Statement Details” on page 89.

Interaction: The actual position is determined by the settings for the LOCATION=, AUTOALIGN=, HALIGN=, and VALIGN= options.

OPAQUE=*boolean*

specifies whether the legend background is opaque (TRUE) or transparent (FALSE).

Default: TRUE when LOCATION=OUTSIDE FALSE when LOCATION=INSIDE

Interaction: When this option is set to FALSE, the background color is not used.

ORDER=ROWMAJOR | COLUMNMAJOR

specifies whether legend entries are added as rows or as columns.

Default: ROWMAJOR

Interaction: If ORDER=ROWMAJOR, use the ACROSS= option to limit the number of entries in a row. If ORDER=COLUMNMAJOR, use the DOWN= option to limit the number of entries in a column.

PAD=*dimension* | *(pad-options)*

specifies the amount of extra space that is reserved inside the legend perimeter.

Default: The default padding for all sides is 0. Values without units are in pixels (px). A unit must be provided if other than pixels.

dimension

Specifies a dimension to use for the extra space at the left, right, top, and bottom of the legend perimeter.

(*pad-options*)

Enables separate settings for the left, right, top, and bottom padding dimensions. Use the *pad-options* to create non-uniform padding. These options must be enclosed in parentheses. Each option is specified as a *name* = *value* pair. Sides not assigned padding are padded with the default amount.

Values without units are in pixels (px). A unit must be provided if other than pixels.

LEFT=*dimension*	specifies the amount of extra space added to the left side.
RIGHT=*dimension*	specifies the amount of extra space added to the right side.
TOP=*dimension*	specifies the amount of extra space added to the top.
BOTTOM=*dimension*	specifies the amount of extra space added to the bottom.

TITLE= *"string"*

specifies the title of the legend.

Default: no default

Requirement: string must be enclosed in quotation marks.

TITLEATTRS=*style-element* | *style-element* (*text-options*) | (*text-options*)

specifies the color and font attributes of the legend title. See “General Syntax for Attribute Options” on page 563 for the syntax on using a *style-element* and “Text Options” on page 566 for available *text-options*.

Default: The GraphLabelText style element.

Interaction: For this option to have any effect, the TITLE= option must also be specified.

TITLEBORDER=*boolean*

specifies a border around the legend title that separates it from the legend body.

Default: FALSE

Interaction: For this option to have any effect, the TITLE= option must also be specified.

The line attributes of the title border are set by the BORDERATTRS= options.

VALIGN=CENTER | TOP | BOTTOM

specifies the vertical alignment of the legend within its parent layout when nested within an overlay-type layout. For more information about how child positions are determined in an overlay-type layout, see the LAYOUT OVERLAY’s “Example Program and Statement Details” on page 89.

Default: CENTER

Restriction: If LOCATION=OUTSIDE, the VALIGN= and HALIGN= options cannot both be set to CENTER.

Restriction: This option is available only when this statement is nested within an overlay-type layout. It is ignored if the parent layout is not an overlay-type layout.

Interaction: If LOCATION=INSIDE and the AUTOALIGN= option is enabled, this option is ignored.

VALUEATTRS=*style-element* | *style-element* (*text-options*) | (*text-options*)
specifies the color and font attributes of the legend values. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Text Options" on page 566 for available *text-options*.

Default: The GraphValueText style element.

Part 7

Text Statements

Chapter 49

Managing Text Items

Overview

The ENTRY, ENTRYTITLE, and ENTRYFOOTNOTE statements all use the same syntax for specifying one or more pieces of text called *text-items*. For example, here is the ENTRYTITLE syntax:

ENTRYTITLE *text-item* <...<*text-item*>> </*option(s)*>;

Each *text-item* can be specified using the following syntax:

<*prefix-option*...<*prefix-option*>> "*string*" | *dynamic* | *character-expression* | {*text-command*}

The simplest specification for each statement is to specify a string in quotes. For example, you might specify a graph title as follows:

```
entrytitle "Height and Weight by Sex";
```

In this example, the string *Height and Weight by Sex* is formatted as the graph title and displayed in the title position. If multiple strings are specified, they are concatenated into a single line of text. This second specification generates the same title:

```
entrytitle "Height " "and " "Weight " "by " "Sex";
```

To provide control over the text, multiple prefix options can precede each *text-item*, and the *text-item* can be a string literal, a dynamic, or a text command. All *text-items* with the same HALIGN= setting are concatenated into one string. Up to three strings with different horizontal alignment can result for each ENTRY statement. Leading and trailing blanks in the concatenated string are always used.

- a *string* must be enclosed in quotation marks.
- a *character-expression* must be enclosed in an EVAL function.
- a *text-command* must be enclosed in braces (see "Using Text Commands" on page 512).

Using Prefix Options

Available Prefix Options

The following prefix options are available on ENTRY, ENTRYTITLE, and ENTRYFOOTNOTE statements:

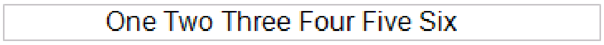
HALIGN
: Specifies the horizontal alignment of a *text-item*.

TEXTATTRS
: Specifies font attributes for a *text-item*.

When used, a prefix option applies not only to immediately following piece of text but also to ALL subsequent text strings and text-commands. If the same prefix option appears more than once, each use overrides the last used prefix of the same name.

Managing Horizontal Alignment

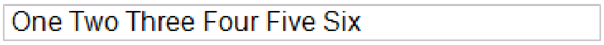
For the ENTRY, ENTRYTITLE, and ENTRYFOOTNOTE statements, the default horizontal alignment is CENTER.



One Two Three Four Five Six

```
entry "One" " Two" " Three" " Four" " Five" " Six";
```

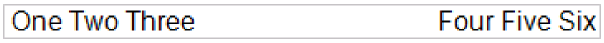
To change the default alignment, you can use the HALIGN= option, which can specify LEFT, CENTER, or RIGHT alignment. The following specification left-justifies the text:



One Two Three Four Five Six

```
entry halign=left
 "One" " Two" " Three" " Four" " Five" " Six";
```

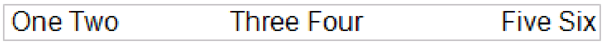
Text items are positionally concatenated by alignment area. For example, the following specification left-justifies the first three strings and right-justifies the last three strings:



One Two Three   Four Five Six

```
entry halign=left "One" " Two" " Three"
      halign=right "Four" " Five" " Six";
```

Even if the HALIGN= specifications are jumbled, the final text is nevertheless positionally concatenated by alignment area:



One Two Three Four Five Six

```
entry halign=right "Five" halign=left "One"
      halign=right " Six" halign=center "Three"
      halign=left " Two"  halign=center " Four";
```

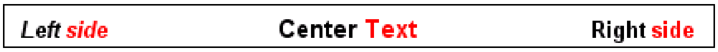
When long strings are used or the bounding container is constrained, the alignment areas might overlap.

Managing Font Attributes

For the ENTRY, ENTRYTITLE, and ENTRYFOOTNOTE statements, the default font attributes are determined by the active ODS style. To change the default font attributes, you can use the TEXTATTRS= option, which has the following syntax (see "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Text Options" on page 566 for available *text-options*):

TEXTATTRS=*style-element* | *style-element(text-options)* | (*text-options*)

For example, the following ENTRYTITLE statement uses prefix options to create this title line:



Left side **Center Text** **Right side**

```
entrytitle textattrs=(color=black) "Center"
  textattrs=(color=red) " Text"
  halign=right textattrs=(color=black size=10pt) "Right "
    textattrs=(color=red size=10pt) "side"
  halign=left
    textattrs=(color=black style=italic size=10pt) "Left"
    textattrs=(color=red style=italic size=10pt) " side";
```

- The TEXTATTRS= options are reset each time a new TEXTATTRS= appears—there is no "carry over" or accumulation of the settings.
- Blanks (spaces) must be provided as needed to achieve the desired concatenation. If you are concatenating dynamics that are stripped of leading and trailing blanks, you might need to introduce a literal space to separate them, as shown in the following example:

```
entry _DYN1 " " _DYN2;
```

Alternatively, you can specify font attributes by overriding style element defaults. The following specification overrides the settings of the GraphTitleText style element, which sets the default attributes for ENTRYTITLE text:

```
entrytitle
halign=left
```

```
 textattrs=GraphTitleText(color=black style=italic) "Left"
 textattrs=GraphTitleText(color=red style=italic) " side"
halign=center
 textattrs=(color=black) "Center"
 textattrs=(color=red) " Text"
halign=right
 textattrs=GraphTitleText(color=black weight=bold) "Right "
 textattrs=GraphTitleText(color=red weight=bold) "side" ;
```

You can also use dynamics in the *text-items*. In the following ENTRYTITLE statement, assume that _DEPLABEL and _MODELLABEL are dynamics that are specified on PROC TEMPLATE's DYNAMIC statement:

```
entrytitle "Residual by Predicted for " _DEPLABEL
  halign=left textattrs=GraphTitleText _MODELLABEL /
   pad=(bottom=5);
```

- The default style element for ENTRYTITLE is GraphTitleText, so all three text items (one literal and two dynamics) get these font properties as a starting point.
- The text *"Residual by Predicted for " _DEPLABEL* is center-aligned by default.
- The prefix options HALIGN= and TEXTATTRS= override the center alignment and font properties for the text _MODELLABEL.

Notice that this could have been coded as follows:

```
entrytitle halign=left
  textattrs=GraphTitleText _MODELLABEL
  halign=center textattrs=()
    "Residual by Predicted for " _DEPLABEL;
```

In this example, the second HALIGN= and TEXTATTRS=() were necessary to reset alignment and font properties to their defaults.

Note that the string length of the resolved dynamic _MODELLABEL does not affect the placement of the center-aligned text.

Using Text Commands

Available Text Commands

Text commands on ENTRY, ENTRYTITLE, and ENTRYFOOTNOTE statements are special in-line instructions that either modify the appearance of the text or script special characters. The following text commands are available:

{SUB}
: Specifies that the string or dynamic appears as a subscript.

{SUP}
: Specifies that the string or dynamic appears as a superscript.

{UNICODE}
: Specifies a glyph (graphical character) to be displayed using its Unicode specification or keyword equivalent.

The general form of a text command is

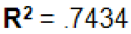
{command argument(s)}

The opening and closing braces are required to denote the scope of the command.

Subscripting and Superscripting Text

The {SUB} and {SUP} text commands are used to subscript and superscript text. Each of these text commands accepts a *string* or a *dynamic* for its argument(s).

In the following example, if _RSQUARE is an existing dynamic that resolves to the value *.7434*, then the following ENTRY specification superscripts the string value "2" to generate this text:



R^2 = .7434

```
entry textattrs=(weight=bold) "R" {sup "2"}
      textattrs=() "=" _RSQUARE;
```

Notice that textattrs=() cancels all style overrides and reverts to the default text properties.

Using UNICODE Text

The {UNICODE} text command places special characters into the text and accepts any of the following for its argument(s):

- a hexadecimal Unicode Code Point for a character (for example, *"03B1"x*)
- a reserved keyword for a commonly used code point (for example, *beta*)
- a dynamic that resolves to a hexadecimal value or keyword.

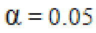
See the tables in "Reserved Keywords and Unicode Values" on page 514 for a list of the reserved keywords and Unicode values (the tables are not complete, but give a good idea about what is possible).

Multiple arguments can be used within the scope of a single UNICODE text command. For example, the following two specifications are equivalent:

```
{unicode "03b1"x beta}
{unicode "03b1"x} {unicode beta}
```

The default formatting for the UNICODE text is derived from the GraphUnicodeText style element.

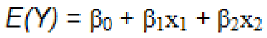
In the following example, if _ALPHAVAL is an existing dynamic that resolves to the value *0.05*, then the following ENTRY specification generates this text:



$\alpha = 0.05$

```
entry {unicode alpha} " = " _ALPHAVAL;
```

By combining the TEXTATTRS= prefix option with the {SUB} and {UNICODE} text commands, you can generate the following text:



$E(Y) = \beta_0 + \beta_1 x_1 + \beta_2 x_2$

```
Entry textattrs=(style=italic) "E(Y)" textattrs=() " = "
      {unicode beta} {sub "0"} " + "
      {unicode beta} {sub "1"} "x" {sub "1"}
      " + " {unicode beta} {sub "2"} "x" {sub "2"};
```

Reserved Keywords and Unicode Values

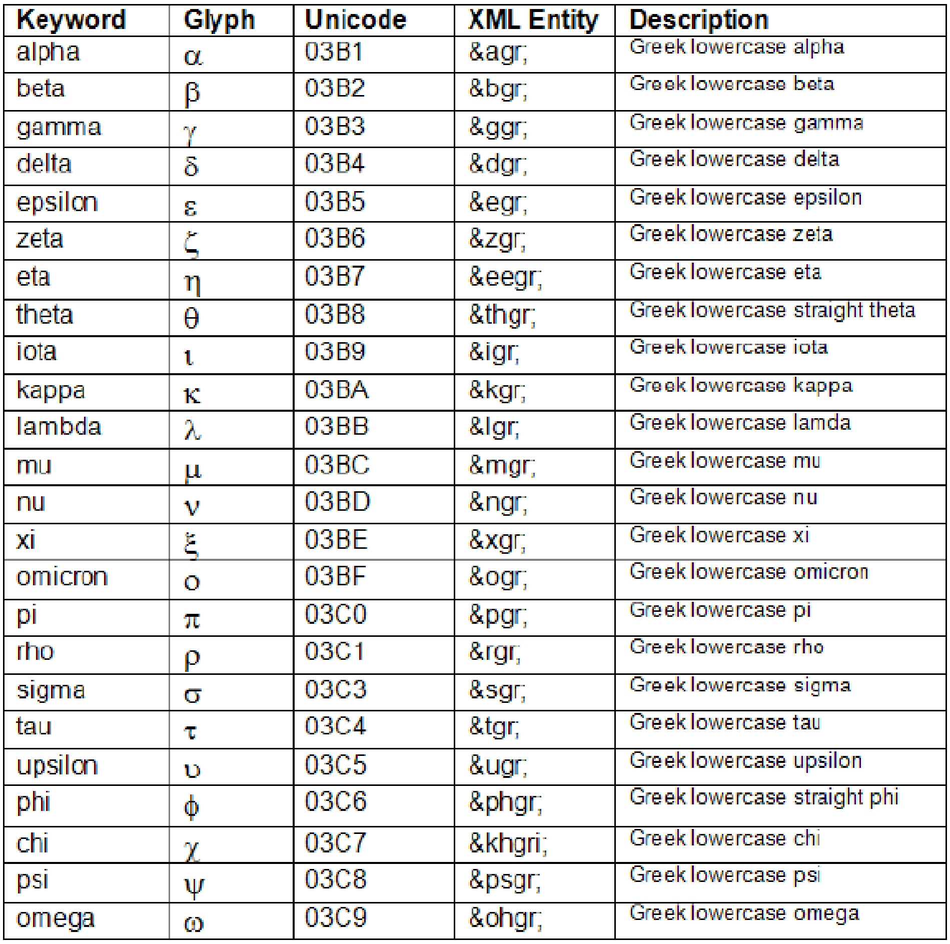
Overview

The tables that follow show some of the reserved keywords and Unicode values that can be used with the UNICODE text command.

Keywords and Unicode values are not case-sensitive: "03B1"x is the same code point as "03b1"x.

The word **blank** is the keyword for a blank space.

Lower Case Greek Letters



Keyword	Glyph	Unicode	XML Entity	Description
alpha	α	03B1	&agr;	Greek lowercase alpha
beta	β	03B2	&bgr;	Greek lowercase beta
gamma	γ	03B3	&ggr;	Greek lowercase gamma
delta	δ	03B4	&dgr;	Greek lowercase delta
epsilon	ε	03B5	&egr;	Greek lowercase epsilon
zeta	ζ	03B6	&zgr;	Greek lowercase zeta
eta	η	03B7	&eegr;	Greek lowercase eta
theta	θ	03B8	&thgr;	Greek lowercase straight theta
iota	ι	03B9	&igr;	Greek lowercase iota
kappa	κ	03BA	&kgr;	Greek lowercase kappa
lambda	λ	03BB	&lgr;	Greek lowercase lamda
mu	μ	03BC	&mgr;	Greek lowercase mu
nu	ν	03BD	&ngr;	Greek lowercase nu
xi	ξ	03BE	&xgr;	Greek lowercase xi
omicron	ο	03BF	&ogr;	Greek lowercase omicron
pi	π	03C0	&pgr;	Greek lowercase pi
rho	ρ	03C1	&rgr;	Greek lowercase rho
sigma	σ	03C3	&sgr;	Greek lowercase sigma
tau	τ	03C4	&tgr;	Greek lowercase tau
upsilon	υ	03C5	&ugr;	Greek lowercase upsilon
phi	ϕ	03C6	&phgr;	Greek lowercase straight phi
chi	χ	03C7	&khgri;	Greek lowercase chi
psi	ψ	03C8	&psgr;	Greek lowercase psi
omega	ω	03C9	&ohgr;	Greek lowercase omega

Upper Case Greek Letters

Keyword	Glyph	Unicode	XML Entity	Description
alpha_u	Α	0391	&Agr;	Greek uppercase alpha
beta_u	Β	0392	&Bgr;	Greek uppercase beta
gamma_u	Γ	0393	&Ggr;	Greek uppercase gamma
delta_u	Δ	0394	&Dgr;	Greek uppercase delta
epsilon_u	Ε	0395	&Egr;	Greek uppercase epsilon
zeta_u	Ζ	0396	&Zgr;	Greek uppercase zeta
eta_u	Η	0397	&EEgr;	Greek uppercase eta
theta_u	Θ	0398	&THgr;	Greek uppercase theta
iota_u	Ι	0399	&Igr;	Greek uppercase iota
kappa_u	Κ	039A	&Kgr;	Greek uppercase kappa
lambda_u	Λ	039B	&Lgr;	Greek uppercase lambda
mu_u	Μ	039D	&Mgr;	Greek uppercase mu
nu_u	Ν	039C	&Ngr;	Greek uppercase nu
xi_u	Ξ	039E	&Xgr;	Greek uppercase xi
omicron_u	Θ	039F	&Ogr;	Greek uppercase omicron
pi_u	Π	03A0	&Pgr;	Greek uppercase pi
rho_u	Ρ	03A1	&Rgr;	Greek uppercase rho
sigma_u	Σ	03A3	&Sgr;	Greek uppercase sigma
tau_u	Τ	03A4	&Tgr;	Greek uppercase theta
upsilon_u	Υ	03A5	&Ugr;	Greek uppercase upsilon
phi_u	Φ	03A6	&PHgr;	Greek uppercase phi
chi_u	Χ	03A7	&KHgr;	Greek uppercase chi
psi_u	Ψ	03A8	&PSgr;	Greek uppercase psi
omega_u	Ω	03A9	&OHgr;	Greek uppercase omega

Chapter 50
ENTRY Statement

Description and Syntax

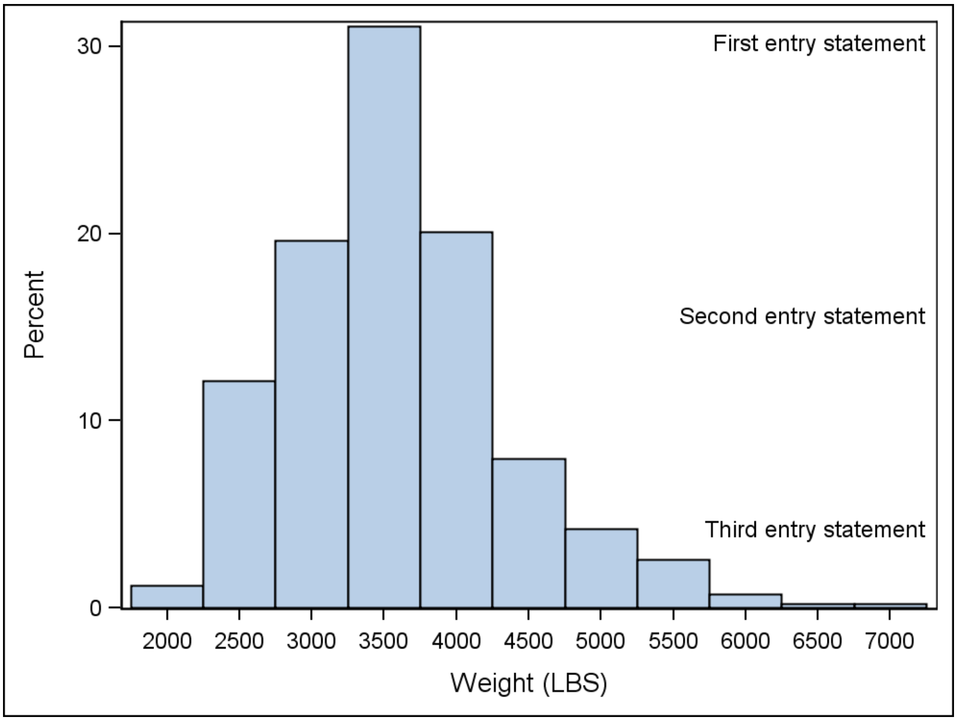
Displays a line of text in the plot area.

ENTRY *text-item <…<text-item>> </option(s)>*;

Example Program and Statement Details

Example Graph

The following graph was generated by the "Example Program" on page 518:

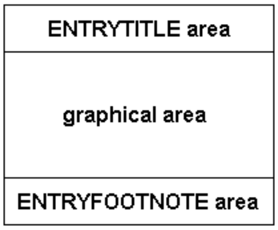


Example Program

```
proc template;
  define statgraph entry;
    begingraph;
      layout overlay;

        entry halign=right "First entry statement" /
          valign=top;

        histogram weight;

        entry halign=right "Second entry statement";

        entry halign=right "Third entry statement" /
          valign=bottom pad=(bottom=40px);

      endlayout;
    endgraph;
  end;
run;

proc sgrender data=sashelp.cars template=entry;
run;
```

Statement Summary

An ENTRY statement creates one line of text in the plot area. The statement must be specified within a LAYOUT, HEADER, SIDEBAR, or CELL statement block. It cannot be specified outside of one of these blocks, where global statements like ENTRYTITLE and ENTRYFOOTNOTE are used.



The text line specified on an ENTRY statement can be made of several pieces of the text called *text-items*. Statement options that are used establish properties for the entire line of text (that is, all *text-items*). These properties can be overridden with *prefix-options* that are specific to one or more *text-items*. See "Required Arguments" on page 519 for more information.

You can specify an ENTRY statement inside or outside of a nested statement block:

- When you specify an ENTRY statement inside of a nested LAYOUT, HEADER, SIDEBAR, or CELL statement block, then, by default, the text is placed inside the bounding area of the results that the nested statement block creates.
- When you specify an ENTRY statement outside of a nested LAYOUT, HEADER, SIDEBAR, or CELL statement block, then the text is placed outside of the bounding area of the results that the nested statement block creates.

Required Arguments

text-item <…<text-item>>
specifies one or more pieces of text for the entry. Each *text-item* has the following form:

<prefix-option …<prefix-option>>"*string*" | *dynamic* | *character-expression* | {*text-command*}

Each piece of text can have multiple prefix settings that precede the piece of text. A piece of text is either a string literal, a dynamic, or a text command. All *text-items* with the same HALIGN= setting are concatenated into one string. Up to three strings can result for each ENTRY statement. Leading and trailing blanks in the concatenated string are always used.

When used, a prefix option applies not only to immediately following piece of text but also to ALL subsequent text strings and text-commands. If the same prefix option appears more than once, it has the effect of overriding the last used prefix of the same name.

Requirement:

- *string* must be enclosed in quotation marks.
- *character-expression* must be enclosed in an EVAL function.
- *text-command* must be enclosed in braces.

See **LINK TO GENERIC ENTRY STRINGS DOC** for more information and several examples.

Options and Text Commands

prefix-option	**Description**
HALIGN	Specifies the horizontal alignment of a *text-item*.
TEXTATTRS	Specifies font attributes for a *text-item*.

text-command	**Description**
{SUB}	Specifies that the string or dynamic appears as a subscript.
{SUP}	Specifies that the string or dynamic appears as a superscript.
{UNICODE}	Specifies a glyph (graphical character) to be displayed using its Unicode specification or keyword equivalent.

Statement Option	**Description**
AUTOALIGN	Specifies whether the entry is automatically aligned within its parent when nested within an overlay-type layout.
BACKGROUNDCOLOR	Specifies the color of the entry background.
BORDER	Specifies whether a border is displayed around the entry.
BORDERATTRS	Specifies the properties of the border line.
OPAQUE	Specifies whether the entry background is opaque.
PAD	Specifies the amount of extra space that is reserved inside the entry border.
ROTATE	Specifies the angle of text rotation.
TEXTATTRS	Specifies the font attributes of the text.
VALIGN	Specifies the vertical alignment of the entry.

AUTOALIGN=NONE | AUTO | (*location-list*)

specifies whether the entry is automatically aligned within its parent when nested within an overlay-type layout. For more information about how child positions are determined in an overlay-type layout, see the LAYOUT OVERLAY's "Example Program and Statement Details" on page 89.

Default: NONE

NONE
Do not automatically align the entry within the area. Alignment is set with HALIGN= and VALIGN= options.

AUTO
Available only if the parent layout contains a scatter plot; ignored otherwise. Within the parent layout, attempt to center the entry in the area that is farthest from any surrounding data point markers.

(*location-list*)
Within the available area, restrict the entry's possible locations to those locations in the specified *location-list*, and use the *location-list* position that least collides with the other graphics features in the area. The *location-list* is blank-separated and can contain any of these locations: TOPLEFT, TOP, TOPRIGHT, LEFT, CENTER, RIGHT, BOTTOMLEFT, BOTTOM, and BOTTOMRIGHT.

Example: (TOPRIGHT TOPLEFT)

BACKGROUNDCOLOR=*style-reference* | *color*
specifies the color of the entry background.

Default: The GraphBackground:Color style reference.

style-reference
A reference of the form *style-element:style-attribute*. Only the style-attribute named COLOR is used.

Interaction: OPAQUE=TRUE must be in effect for the color to be seen. By default, OPAQUE=FALSE.

BORDER=*boolean*
specifies whether a border is displayed around the entry.

Default: FALSE

BORDERATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)
specifies the attributes of the border line around the text. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Line Options" on page 564 for available *line-options*.

Default: The GraphBorderLines style element.

Interaction: BORDER=TRUE must be set for this option to have any effect.

HALIGN=CENTER | LEFT | RIGHT
prefix-option that specifies the horizontal alignment of a *text-item*.

Default: CENTER

Each *text-item* has a horizontal alignment, and *text-items* with the same alignment are always grouped together.

OPAQUE=*boolean*
specifies whether the entry background is opaque (TRUE) or transparent (FALSE).

Default: FALSE

Interaction: When this option is set to FALSE, the background color is not used.

PAD=*dimension* | *(pad-options)*
specifies the amount of extra space that is reserved inside the entry border.

Default: (LEFT=3 RIGHT=3 TOP=0 BOTTOM=0)

dimension

Specifies a dimension to use for the extra space at the left, right, top, and bottom of the entry border.

(*pad-options*)

Enables separate settings for the left, right, top, and bottom padding dimensions. Use the *pad-options* to create non-uniform padding. These options must be enclosed in parentheses. Each option is specified as a *name* = *value* pair. Sides not assigned padding are padded with the default amount.

Values without units are in pixels (px). A unit must be provided if other than pixels.

LEFT=*dimension*	specifies the amount of extra space added to the left side.
RIGHT=*dimension*	specifies the amount of extra space added to the right side.
TOP=*dimension*	specifies the amount of extra space added to the top.
BOTTOM=*dimension*	specifies the amount of extra space added to the bottom.

ROTATE=0 | 90 | 180 | 270

specifies the angle of text rotation measured in degrees. The angle is measured from a horizontal line passing through the middle of the bounding box of the text, counter-clockwise starting at the center of the bounding box.

Default: 0. No rotation is performed.

Restriction: Only angles of 0, 90, 180, or 270 degrees are allowed.

Interaction: The bounding box is the determined by the size of the text in the current font plus any horizontal and vertical padding. See TEXTATTRS= and PAD=.

{ SUB *"string"* | *dynamic* }

text-command that specifies that the string or dynamic is to appear as subscript text.

Default: no default

Example: entry "y = " b{sub "0"} " + b" {sub "1"} "x";

{ SUP *"string"* | *dynamic* }

text-command that specifies that the string or dynamic is to appear as superscript text.

Default: no default

Example: entry "R" {sup "2"} " = " {format (6.4) RSQUARED} ;

TEXTATTRS=*style-element* | *style-element* (*text-options*) | (*text-options*)

as a *statement option*, specifies the text properties of the entire entry text. As a *prefix-option*, specifies the text properties of individual *text-items*. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Text Options" on page 566 for available *text-options*.

Default: The GraphValueText style element.

This option can be used as both a prefix option and a statement option. When used as a prefix option, it overrides the statement option.

When used as a prefix option, TEXTATTRS=( ) cancels the last used TEXTATTRS= prefix option. It resets all text options to those set by 1) the TEXTATTRS= statement option or 2) the default style element for the statement (GraphValueText) if no TEXTATTRS= statement option is used.

{ UNICODE *"hex-string"x* | *keyword* | *dynamic* }

text-command that specifies a glyph (character) to be displayed using its Unicode specification or keyword equivalent.

Default: no default

*"hex-string"*x

A four-byte hexadecimal constant that represents a UNICODE character in the current font. For a complete listing, see **`http://unicode.org/charts/charindex.html`**.

keyword

A SAS keyword for a UNICODE character. For a listing of SAS supplied keywords, see **LINK TO GENERIC ENTRY STRINGS**.

dynamic

The dynamic must resolve to either *"hex-string"x* or a keyword for a UNICODE character.

Examples:

```
entry {unicode alpha} "=" CONF;

entry {unicode "03B1"x} "=" CONF;
```

This text command attempts to access a UNICODE value in the current font. Not all fonts support accessing characters via their UNICODE value. Some fonts only support a limited set of UNICODE values. If the UNICODE value is not accessible, the command might be ignored or an unprintable character might be substituted.

VALIGN=CENTER | TOP | BOTTOM

specifies the vertical alignment of the entry.

Default: CENTER

Chapter 51
ENTRYFOOTNOTE Statement

Description and Syntax

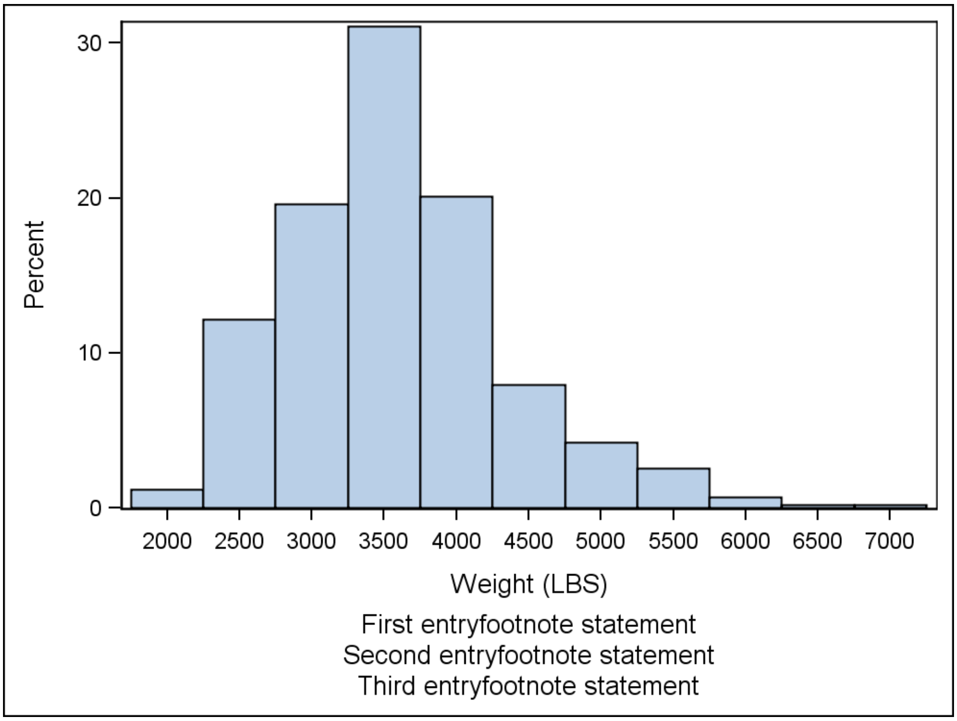
Displays a footnote.

ENTRYFOOTNOTE *text-item* <…<*text-item*>> *</option(s)>*;

Example Program and Statement Details

Example Graph

The following graph was generated by the “Example Program” on page 526:

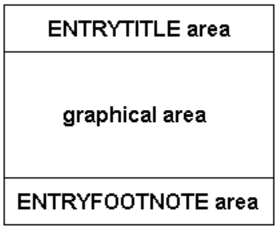


Example Program

```
proc template;
  define statgraph entryfootnote;
    begingraph;

      entryfootnote "First entryfootnote statement" ;

        layout overlay;
          histogram weight;
        endlayout;

      entryfootnote "Second entryfootnote statement" ;
      entryfootnote "Third entryfootnote statement" ;

    endgraph;
  end;
run;

proc sgrender data=sashelp.cars template=entryfootnote;
run;
```

Statement Summary

The ENTRYFOOTNOTE statement places footnote text below the graphical area. More than one ENTRYFOOTNOTE statement can be used. Footnotes appear in the order of the ENTRYFOOTNOTE statements within the template.



When adding an ENTRYFOOTNOTE statement to a template definition, the statement must be located within the BeginGraph/EndGraph block but outside of the outermost layout block.

- All ENTRYFOOTNOTE statements that appear in the template are gathered and placed in the ENTRYFOOTNOTE area.
- The placement of an ENTRYFOOTNOTE statement is relevant only in relation to other ENTRYFOOTNOTE statements.
- As the number of ENTRYFOOTNOTE statements increases the size of the graphical area is reduced.

Footnotes always span the entire width of the output. By default, footnotes are “center-aligned,” based on the type of the outermost layout. The meaning of “center-aligned” varies by layout type and the number of columns in the layout:

Layout Type	**Default horizontal centering of footnotes**
GRIDDED OVERLAY3D LATTICE (COLUMNS > 1) DATAPANEL (COLUMNS > 1) DATALATTICE (COLUMNS > 1)	Centered on width of entire graph
OVERLAY OVERLAYEQUATED LATTICE (COLUMNS = 1) DATAPANEL (COLUMNS = 1) DATALATTICE (COLUMNS = 1)	Centered on the plot area

Required Arguments

text-item <…<*text-item*>>
specifies one or more pieces of text for the entry. Each *text-item* has the following form:

<*prefix-option* …<*prefix-option*>>"*string*" | *dynamic* | *character-expression* | {*text-command*}

Each piece of text can have multiple prefix options that precede the piece of text. A piece of text is either a string literal, a dynamic, or a text command. All *text-items* with the same HALIGN= setting are concatenated into one string. Up to three strings can result for each ENTRY statement. Leading and trailing blanks in the concatenated string are always used.

When used, a prefix option applies not only to immediately following piece of text but also to ALL subsequent text strings and text-commands. If the same prefix option appears more than once, it has the effect of overriding the last used prefix of the same name.

Requirement:

- *string* must be enclosed in quotation marks.
- *character-expression* must be enclosed in an EVAL function.
- *text-command* must be enclosed in braces.

See **LINK TO GENERIC ENTRY STRINGS DOC** for more information and several examples.

Options and Text Commands

prefix-option	Description
HALIGN	Specifies the horizontal alignment of a **text-item**.
TEXTATTRS	Specifies font attributes for a **text-item**.

text-command	Description
{SUB}	Specifies that the string or dynamic appears as a subscript.
{SUP}	Specifies that the string or dynamic appears as a superscript.
{UNICODE}	Specifies a glyph (graphical character) to be displayed using its Unicode specification or keyword equivalent.

Statement Option	Description
BACKGROUNDCOLOR	Specifies the color of the footnote background.
BORDER	Specifies whether a border is displayed around the footnote.
BORDERATTRS	Specifies the properties of the border line.
OPAQUE	Specifies whether the footnote background is opaque.
PAD	Specifies the amount of extra space that is reserved inside the footnote border.

Statement Option	Description
SHORTTEXT	Specifies an alternate footnote to use if the specified footnote is too long for the output width.
TEXTATTRS	Specifies the text properties of the entire footnote text.
TEXTFITPOLICY	Specifies how to handle a footnote that is too long to fit in the output width.

BACKGROUNDCOLOR=*style-reference* | *color*
specifies the color of the footnote background.

Default: The GraphBackground:Color style reference.

style-reference
A reference of the form *style-element:style-attribute*. Only the style-attribute named COLOR is used.

Interaction: OPAQUE=TRUE must be in effect for the color to be seen. By default, OPAQUE=FALSE.

BORDER=*boolean*
specifies whether a border is displayed around the footnote.

Default: FALSE

BORDERATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)
specifies the attributes of the border line around the footnote. See “General Syntax for Attribute Options” on page 563 for the syntax on using a *style-element* and “Line Options” on page 564 for available *line-options*.

Default: The GraphBorderLines style element.

Interaction: BORDER=TRUE must be set for this option to have any effect.

HALIGN=CENTER | LEFT | RIGHT
prefix-option that specifies the horizontal alignment of a *text-item*.

Default: CENTER

Each *text-item* has a horizontal alignment, and *text-items* with the same alignment are always concatenated together.

OPAQUE=*boolean*
specifies whether the footnote background is opaque (TRUE) or transparent (FALSE).

Default: FALSE

Interaction: When this option is set to FALSE, the background color is not used.

PAD=*dimension* | *(pad-options)*
specifies the amount of extra space that is reserved inside the footnote border.

Default: (LEFT=3 RIGHT=3 TOP=0 BOTTOM=0)

dimension
Specifies a dimension to use for the extra space at the left, right, top, and bottom of the footnote border.

(*pad-options*)

Enables separate settings for the left, right, top, and bottom padding dimensions. Use the *pad-options* to create non-uniform padding. These options must be enclosed in parentheses. Each option is specified as a *name* = *value* pair. Sides not assigned padding are padded with the default amount.

Values without units are in pixels (px). A unit must be provided if other than pixels.

LEFT=*dimension*	specifies the amount of extra space added to the left side.
RIGHT=*dimension*	specifies the amount of extra space added to the right side.
TOP=*dimension*	specifies the amount of extra space added to the top.
BOTTOM=*dimension*	specifies the amount of extra space added to the bottom.

SHORTTEXT= (*text-item* <...<*text-item*>>)

specifies an alternate footnote to use if the specified footnote is too long for the output width. If the shortened text is itself too long, it is truncated.

Default: no default

Interaction: This option has no effect unless TEXTFITPOLICY=SHORT.

Interaction: This option is ignored if any *text-items* include an HALIGN= prefix option.

Interaction: The horizontal alignment of the shortened text is derived from the horizontal alignment of the footnote to be shortened.

{ SUB *"string"* | *dynamic* }

text-command that specifies that the string or dynamic is to appear as subscript text.

Default: no default

Example: entryfootnote "y = " b{sub "0"} " + b" {sub "1"} "x";

{ SUP *"string"* | *dynamic* }

text-command that specifies that the string or dynamic is to appear as superscript text.

Default: no default

Example: entryfootnote "R" {sup "2"} " = " RSQUARED;

TEXTATTRS=*style-element* | *style-element* (*text-options*) | (*text-options*)

as a *statement option*, specifies the text properties of the entire entry text. As a *prefix-option*, specifies the text properties of individual *text-items*. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Text Options" on page 566 for available *text-options*.

Default: The GraphFootnoteText style element.

This option can be used as both a prefix option and a statement option. When used as a prefix option, it overrides the statement option.

When used as a prefix option, TEXTATTRS=( ) cancels the last used TEXTATTRS= prefix option. It resets all text options to those set by 1) the TEXTATTRS= statement option or 2) the default style element for the statement (GraphFootnoteText) if no TEXTATTRS= statement option is used.

TEXTFITPOLICY=WRAP | SHORT | TRUNCATE

specifies how to handle a footnote that is too long to fit in the output width.

Default: WRAP

WRAP
specifies that the text wraps to the next line(s).

SHORT
specifies that the text indicated by the SHORTTEXT= option be substituted when the footnote does not fit.

TRUNCATE
specifies that the text is truncated to make it fit.

Interaction: If you specify SHORT for this option, you should assign a shortened version of the footnote with the SHORTTEXT= option. If the shortened footnote is itself too long, it is truncated.

{ UNICODE *"hex-string"x* | *keyword* | *dynamic* }
text-command that specifies a glyph (character) to be displayed using its Unicode specification or keyword equivalent.

Default: no default

*"hex-string"*x
A four-byte hexadecimal constant that represents a UNICODE character in the current font. For a complete listing, see **`http://unicode.org/charts/charindex.html`**.

keyword
A SAS keyword for a UNICODE character. For a listing of SAS supplied keywords, see "Reserved Keywords and Unicode Values" on page 514.

dynamic
The dynamic must resolve to either *"hex-string"x* or a keyword for a UNICODE character.

Examples:

```
entryfootnote {unicode alpha} "=" CONF;

entryfootnote {unicode "03B1"x} "=" CONF;
```

This text command attempts to access a UNICODE value in the current font. Not all fonts support accessing characters via their UNICODE value. Some fonts only support a limited set of UNICODE values. If the UNICODE value is not accessible, the command might be ignored or an unprintable character might be substituted.

Chapter 52
ENTRYTITLE Statement

Description and Syntax

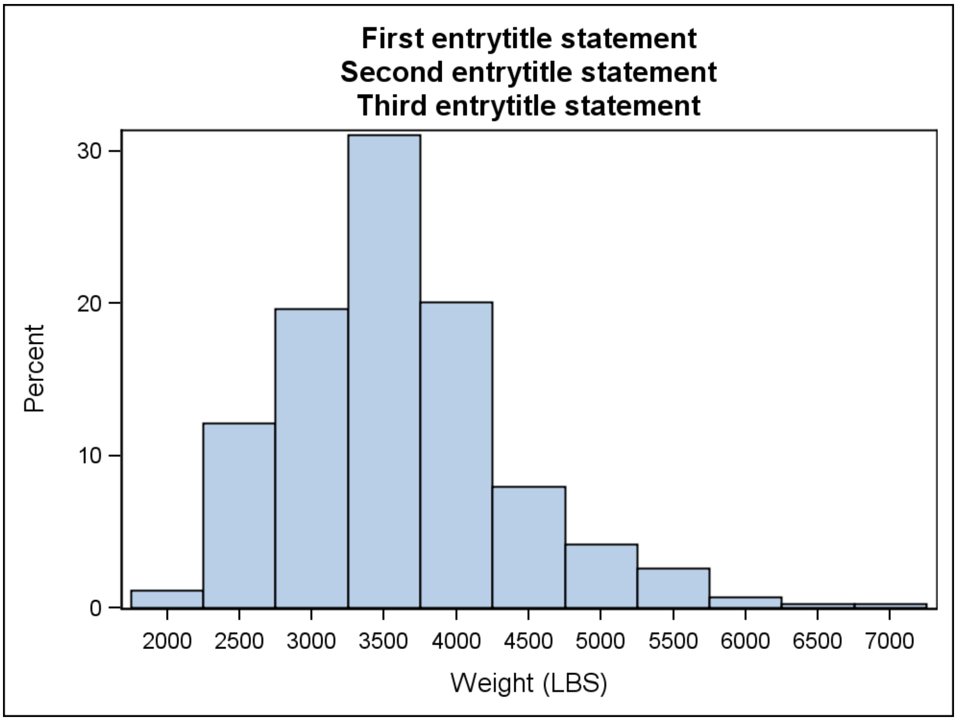
Displays a title.

ENTRYTITLE *text-item* <…<*text-item*>> <*/option(s)*>;

Example Program and Statement Details

Example Graph

The following graph was generated by the "Example Program" on page 534:

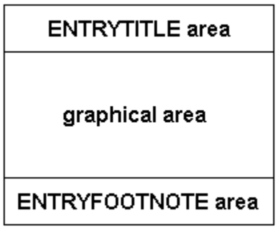


Example Program

```
proc template;
  define statgraph entrytitle;
    begingraph;

      entrytitle "First entrytitle statement" ;

        layout overlay;
          histogram weight;
        endlayout;

      entrytitle "Second entrytitle statement" ;
      entrytitle "Third entrytitle statement" ;

    endgraph;
  end;
run;

proc sgrender data=sashelp.cars template=entrytitle;
run;
```

Statement Summary

The ENTRYTITLE statement places title text above the graphical area. More than one ENTRYTITLE statement can be used. Titles appear in the order of the ENTRYTITLE statements within the template.



When adding an ENTRYTITLE statement to a template definition, the statement must be located within the BeginGraph/EndGraph block but outside of the outermost layout block.

- All ENTRYTITLE statements that appear in the template are gathered and placed in the ENTRYTITLE area.
- The placement of an ENTRYTITLE statement is relevant only in relation to other ENTRYTITLE statements.
- As the number of ENTRYTITLE statements increases the size of the graphical area is reduced.

Titles always span the entire width of the output. By default, titles are "center-aligned," based on the type of the outermost layout. The meaning of "center-aligned" varies by layout type and the number of columns in the layout:

Layout Type	**Default horizontal centering of titles**
GRIDDED OVERLAY3D LATTICE (COLUMNS > 1) DATAPANEL (COLUMNS > 1) DATALATTICE (COLUMNS > 1)	Centered on width of entire graph
OVERLAY OVERLAYEQUATED LATTICE (COLUMNS = 1) DATAPANEL (COLUMNS = 1) DATALATTICE (COLUMNS = 1)	Centered on the plot area

Required Arguments

text-item <…<*text-item*>>
specifies one or more pieces of text for the entry. Each *text-item* has the following form:

<*prefix-option* …<*prefix-option*>>"*string*" | *dynamic* | *character-expression* | {*text-command*}

Each piece of text can have multiple prefix options that precede the piece of text. A piece of text is either a string literal, a dynamic, or a text command. All *text-items* with the same HALIGN= setting are concatenated into one string. Up to three strings can result for each ENTRY statement. Leading and trailing blanks in the concatenated string are always used.

When used, a prefix option applies not only to immediately following piece of text but also to ALL subsequent text strings and text-commands. If the same prefix option appears more than once, it has the effect of overriding the last used prefix of the same name.

Requirement:

- *string* must be enclosed in quotation marks.
- *character-expression* must be enclosed in an EVAL function.
- *text-command* must be enclosed in braces.

See **LINK TO GENERIC ENTRY STRINGS DOC**for more information and several examples.

Options and Text Commands

prefix-option	**Description**
HALIGN	Specifies the horizontal alignment of a *text-item*.
TEXTATTRS	Specifies font attributes for a *text-item*.

text-command	**Description**
{SUB}	Specifies that the string or dynamic appears as a subscript.
{SUP}	Specifies that the string or dynamic appears as a superscript.
{UNICODE}	Specifies a glyph (graphical character) to be displayed using its Unicode specification or keyword equivalent.

Statement Option	**Description**
BACKGROUNDCOLOR	Specifies the color of the title background.
BORDER	Specifies whether a border is displayed around the title.
BORDERATTRS	Specifies the properties of the border line.
OPAQUE	Specifies whether the title background is opaque.
PAD	Specifies the amount of extra space that is reserved inside the title border.
SHORTTEXT	Specifies an alternate title to use if the specified title is too long for the output width.

Statement Option	**Description**
TEXTATTRS	Specifies the text properties of the entire title text.
TEXTFITPOLICY	Specifies how to handle a title that is too long to fit in the output width.

BACKGROUNDCOLOR=*style-reference* | *color*

specifies the color of the title background.

Default: The GraphBackground:Color style reference.

style-reference

A reference of the form *style-element:style-attribute*. Only the style-attribute named COLOR is used.

Interaction: OPAQUE=TRUE must be in effect for the color to be seen. By default, OPAQUE=FALSE.

BORDER=*boolean*

specifies whether a border is displayed around the title.

Default: FALSE

BORDERATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)

specifies the attributes of the border line around the title. See "General Syntax for Attribute Options" on page 563 for the syntax on using a *style-element* and "Line Options" on page 564 for available *line-options*.

Default: The GraphBorderLines style element.

Interaction: BORDER=TRUE must be set for this option to have any effect.

HALIGN=CENTER | LEFT | RIGHT

prefix-option that specifies the horizontal alignment of a *text-item*.

Default: CENTER

Each *text-item* has a horizontal alignment, and *text-items* with the same alignment are always concatenated together.

OPAQUE=*boolean*

specifies whether the title background is opaque (TRUE) or transparent (FALSE).

Default: FALSE

Interaction: When this option is set to FALSE, the background color is not used.

PAD=*dimension* | *(pad-options)*

specifies the amount of extra space that is reserved inside the title border.

Default: (LEFT=3 RIGHT=3 TOP=0 BOTTOM=0)

dimension

Specifies a dimension to use for the extra space at the left, right, top, and bottom of the title border.

(*pad-options*)

Enables separate settings for the left, right, top, and bottom padding dimensions. Use the *pad-options* to create non-uniform padding. These options must be enclosed in parentheses. Each option is specified as a *name* = *value* pair. Sides not assigned padding are padded with the default amount.

Values without units are in pixels (px). A unit must be provided if other than pixels.

LEFT=*dimension*	specifies the amount of extra space added to the left side.
RIGHT=*dimension*	specifies the amount of extra space added to the right side.
TOP=*dimension*	specifies the amount of extra space added to the top.
BOTTOM=*dimension*	specifies the amount of extra space added to the bottom.

SHORTTEXT= (*text-item* <...<*text-item*>>)

specifies an alternate title to use if the specified title is too long for the output width. If the shortened text is itself too long, it is truncated.

Default: no default

Interaction: This option has no effect unless TEXTFITPOLICY=SHORT.

Interaction: This option is ignored if any *text-items* include an HALIGN= prefix option.

Interaction: The horizontal alignment of the shortened text is derived from the horizontal alignment of the title to be shortened.

{ SUB *"string"* | *dynamic* }

text-command that specifies that the string or dynamic is to appear as subscript text.

Default: no default

Example: entrytitle "y = " b{sub "0"} " + b" {sub "1"} "x";

{ SUP *"string"* | *dynamic* }

text-command that specifies that the string or dynamic is to appear as superscript text.

Default: no default

Example: entrytitle "R" {sup "2"} " = " RSQUARED;

TEXTATTRS=*style-element* | *style-element* (*text-options*) | (*text-options*)

as a *statement option*, specifies the text properties of the entire entry text. As a *prefix-option*, specifies the text properties of individual *text-items*. See “General Syntax for Attribute Options” for the syntax on using a *style-element* and “Text Options” on page 566 for available *text-options*.

Default: The GraphTitleText style element.

This option can be used as both a prefix option and a statement option. When used as a prefix option, it overrides the statement option.

When used as a prefix option, TEXTATTRS=( ) cancels the last used TEXTATTRS= prefix option. It resets all text options to those set by 1) the TEXTATTRS= statement option or 2) the default style element for the statement (GraphTitleText) if no TEXTATTRS= statement option is used.

TEXTFITPOLICY=WRAP | SHORT | TRUNCATE

specifies how to handle a title that is too long to fit in the output width.

Default: WRAP

WRAP

specifies that the text wraps to the next line(s).

SHORT
specifies that the text indicated by the SHORTTEXT= option be substituted when the title does not fit.

TRUNCATE
specifies that the text is truncated to make it fit.

Interaction: If you specify SHORT for this option, you should assign a shortened version of the title with the SHORTTEXT= option. If the shortened title is itself too long, it is truncated.

{ UNICODE *"hex-string"x* | *keyword* | *dynamic* }
text-command that specifies a glyph (character) to be displayed using its Unicode specification or keyword equivalent.

Default: no default

*"hex-string"*x
A four-byte hexadecimal constant that represents a UNICODE character in the current font. For a complete listing, see **`http://unicode.org/charts/charindex.html`**.

keyword
A SAS keyword for a UNICODE character. For a listing of SAS supplied keywords, see "Reserved Keywords and Unicode Values" on page 514.

dynamic
The dynamic must resolve to either *"hex-string"x* or a keyword for a UNICODE character.

Examples:

```
entrytitle {unicode alpha} "=" CONF;
```

```
entrytitle {unicode "03B1"x} "=" CONF;
```

This text command attempts to access a UNICODE value in the current font. Not all fonts support accessing characters via their UNICODE value. Some fonts only support a limited set of UNICODE values. If the UNICODE value is not accessible, the command might be ignored or an unprintable character might be substituted.

Part 8

Runtime Programming Features

Chapter 53
Dynamics and Macro Variables

Template Types on PROC TEMPLATE

PROC TEMPLATE supports different template types such as COLUMN, TABLE, HEADER, FOOTER, and STATGRAPH. All of these template types support run time variable substitution via dynamics or macro variables. For STATGRAPH templates such variables should be declared within the scope of the template definition before the BEGINGRAPH block.

```
PROC TEMPLATE;
  DEFINE STATGRAPH template-name ;

   DYNAMIC variable-1<"text-1"> <... variable-n<"text-n">>;
   MVAR variable-1<"text-1"> <... variable-n<"text-n">>;
   NMVAR variable-1<"text-1"> <... variable-n<"text-n">>;
   NOTES "text";

   BEGINGRAPH;
     GTL statements;
   ENDGRAPH;
  END;
RUN;
```

DYNAMIC, MVAR, and NMVAR Statements

Each of the DYNAMIC, MVAR, and NMVAR statements can define multiple variables and an optional text-string denoting its purpose or usage. For example:

```
DYNAMIC YVAR "required" YLABEL "optional";
```

```
MVAR LOCATE "INSIDE or OUTSIDE" SYSDATE;
NMVAR TRANS "transparency factor";
```

Note: For template readability, it is helpful to adopt a naming convention for these variables to distinguish them from actual option values or column names. Common conventions include capitalization, or adding leading or trailing underscores to their names.

Dynamics and macro variables can be referenced within the template definition as

- argument or option values. For example:

  ```
  seriesplot x=date y=YVAR / curvelabel=YLABEL
    curvelabellocation=LOCATE datatransparency=TRANS;
  ```

- parts of some text strings. For example:

  ```
  entrytitle "Time Series for " YLABEL;
  entryfootnote "Created on " SYSDATE;
  ```

Dynamics and run-time macro variable references cannot resolve to statement or option keywords.

Note that macro variable references should not be prefaced with an ampersand (&) if you want them to resolve at run time.

Macro variables defined by MVAR are strings when they resolve, as with SYMGET() in the DATA step.

Macro variables defined by NMVAR are converted to numeric tokens when they resolve, as with SYMGETN() in the DATA step.

Dynamics Compared to Macro Variables

The main difference between dynamics and macro variables is how they are initialized.

For dynamics, use the DYNAMIC statement with PROC SGRENDER. For example,

```
proc sgrender data=sashelp.class template=timeseries;
  dynamic yvar="inflation" ylabel="Inflation Rate";
run;
```

Values for dynamics that resolve to column names or strings should be quoted. Numeric values should not be quoted.

For macro variables, use the current symbol table (local or global) to look up the macro variable values at run time. For example,

```
%let locate=inside;
%let trans=.3;

proc sgrender data=sashelp.class template=timeseries;
  dynamic yvar="inflation" ylabel="Inflation Rate";
run;
```

No initialization is needed for system macro variables like SYSDATE.

Chapter 54
Expressions

Overview

In GTL, as in Base SAS, an expression is an arithmetic or logical expression that consists of a sequence of operators, operands, and functions. An operand is a dynamic, a macro variable, a column, a function, or a constant. An operator is a symbol that requests a comparison, logical operation, or arithmetic calculation. In GTL, the expression must be enclosed in EVAL.

Expressions can be used set an option value that is

- a constant
- a column
- part of the text for ENTRYTITLE, ENTRYFOOTNOTE, and ENTRY statements.

Example of computing constants:

```
/* create reference lines at computed positions */
referenceline y=eval(mean(height)+2*std(height)) /
  curvelabel="+2 STD";
referenceline y=eval(mean(height));
referenceline y=eval(mean(height)-2*std(height)) /
  curvelabel="-2 STD";
```

Example of creating a new column:

```
/* create a new column as a log transformation */
scatterplot x=date y=eval(log10(amount));
```

Example of building a text string:

```
/* create a date and time stamp as a footnote */
entryfootnote  eval(put(today(),date9.)||" : "||
  put(time(),timeampm8.));
```

GTL Expressions Compared to WHERE Expressions

Valid GTL expressions are identical to valid WHERE expressions. See the WHERE statement documentation in Base SAS for a comprehensive list of operators and operands. However, GTL expressions do not perform subset operations as WHERE expressions do. The major difference in the result of a logical GTL expression on a column is that a Boolean value is returned for each observation without changing the number of observations.

For example, in the following line of code, the expression for the Y= argument does not reduce the number of observations plotted.

```
scatterplot x=name y=eval(height between 40 and 60);
```

Instead, the computed numeric column for the Y= argument consists of 0s and 1s, based on whether each observation's HEIGHT value is between 40 and 60. Whenever expressions are used to create new columns, a new column name is internally manufactured so it does not collide with other columns in use.

An *expression* in Statement Syntax

Throughout GTL documentation, you see *expression* used in statement documentation:

BOXPLOT X= *column* | *expression*
Y= *numeric-column* | *expression* < /*option(s)*>;

For the X= argument, *expression* means any EVAL(*expression*) that results in either a numeric or character column. An expression that yields a constant is not valid because the X= argument does not accept constants.

Similarly, for the Y= argument, expression means any EVAL(*expression*) that results in a numeric column. The expression cannot result in a character column or any constant because the Y= argument only accepts a numeric column.

On the following REFERENCELINE statement, the X= argument can be a constant (single line) or a column (multiple lines) that has the same data type as the axis. This means that EVAL(*expression*) can result in a numeric or character column or constant that agrees with the axis type.

REFERENCELINE X= *x-axis-value* | *column* | *expression* </*option(s)*>;

Automatic Type Conversion. Although expressions that are used in a DATA step perform automatic type conversion, GTL expression evaluation does not. Thus, you must use function(s) to perform required type conversions in an expression. Otherwise, the expression generates an error condition without warning when the template is executed.

For example, consider the following GTL expression:

```
if(substr(value, 1, 2) = "11")
```

This expression uses the SUBSTR function to determine whether the first two characters from VALUE evaluate to the string value "11". If VALUE is a string, the expression works fine. However, if VALUE is numeric, the expression generates an error condition. For a numeric, you must convert the value to a string before passing it to the SUBSTR function. The following modification uses the CATS function to perform the type conversion when necessary:

```
if(substr(cats(value, 1, 2)) = "11")
```

Chapter 55
Functions

Overview

GTL supports a large number of functions

- SAS functions that can be used in the context of a WHERE expression
- functions defined only in GTL.

SAS functions that can be used in a WHERE expression include

- character handling functions
- date and time functions
- mathematical and statistical functions.

Note: Not all SAS functions are available in WHERE expressions. Call routines and other DATA-step-only functions (for example, LAG, VNAME, OPEN) are some examples of functions that cannot be used.

Functions Used Only in GTL

The following table shows some functions that are used only in GTL. As with other functions, these must be enclosed within an EVAL. In all these functions, *column* can be either the name of a column in the input data set or a dynamic / macro variable that resolves to such a column.

Function Name	Description
COLNAME(*column*)	returns the case-sensitive name of the *column*.

Function Name	Description
COLLABEL(*column*)	returns the case-sensitive label of the *column*. If no label is defined for the *column*, the case-sensitive name of the *column* is returned.
EXISTS(*item*)	returns 1 if specified *item* exists, 0 otherwise. If *item* is a column, it tests for the presence of the column in the input data set. If *item* is a dynamic / macro variable, it tests whether there has been a run-time initialization of the variable.
EXPAND(*numeric-column*, *freq-column*)	creates a new column as (*numeric-column* * *frequency-column*)
ASORT(*column*, RETAIN=ALL)	sorts all columns of the data object by the values of *column* in ascending order. SORT is an alias for ASORT. **Warning:** if the RETAIN=ALL argument is not included, *column* alone is sorted, not the other columns, causing rowwise information to be lost.
DSORT(*column*, RETAIN=ALL)	sorts all columns of the data object by the values of *column* in descending order. **Warning:** if the RETAIN=ALL argument is not included, *column* alone is sorted, not the other columns, causing rowwise information to be lost.
NUMERATE(*column*)	returns a *column* that contains the ordinal position of each observation in the input data set (similar to an OBS column)

Examples:

```
/* arrange bars in descending order of response values */
barchartparm x=region y=eval(dsort(amount,retain=all));

/* label outliers with their position in the data set */
/* it does not matter which column is used for NUMERATE() */
boxplot x=age y=weight / datalabel=eval(numerate(age));

/* add information about the column being processed,
   which is passed by a dynamic */
entrytitle "Distribution for " eval(colname(DYNVAR));
```

GTL Summary Statistic Functions

The following functions return a numeric constant, based on a summary operation on a numeric column. The results are the same as if the corresponding statistics were requested with PROC SUMMARY. These functions take a single argument that resolves to the name of a numeric column. These functions take precedence over similar multi-argument DATA step functions.

number = EVAL(function-name(numeric-column))	
Function Name	**Description**
CSS	Corrected sum of squares
CV	Coefficient of variation
KURTOSIS	Kurtosis
LCLM	One-sided confidence limit below the mean
MAX	Largest (maximum) value
MEAN	Mean
MEDIAN	Median (50th percentile)
MIN	Smallest (minimum) value
N	Number of nonmissing values
NMISS	Number of missing values
P1	1st percentile
P5	5th percentile
P25	25th percentile
P50	50th percentile
P75	75th percentile
P90	90th percentile
P95	95th percentile
P99	99th percentile
PROBT	p-value for Student's t statistic
Q1	First quartile
Q3	Third quartile
QRANGE	Interquartile range
RANGE	Range

number = EVAL(function-name(numeric-column))	
Function Name	**Description**
SKEWNESS	Skewness
STDDEV	Standard deviation
STDERR	Standard error of the mean
SUM	Sum
SUMWGT	Sum of weights
T	Student’s t statistic
UCLM	One-sided confidence limit above the mean
USS	Uncorrected sum of squares
VAR	Variance

Chapter 56

Conditional Logic

Overview

GTL supports conditional logic that enables you to include or exclude one or more GTL statements at run time:

IF (*condition*)

GTL-statement(s);

ELSE

GTL-statement(s);

ENDIF;

The IF statement requires an ENDIF statement. The IF block can be placed anywhere within the BEGINGRAPH / ENDGRAPH block.

The *condition* is an expression that evaluates to a numeric constant, where all numeric constants other than 0 and MISSING are true. There is an implied EVAL(*condition*), so it is not necessary to include an EVAL as part of the condition.

Examples:

```
/* test a computed value */
if (weekday(today()) in (1 7))
   entrytitle "Run during the work week";
else
   entrytitle "Run during the weekend";
endif;

/* test for the value a numeric dynamic */
if ( ADDREF > 0 )
   referenceline y=1;
   referenceline y=0;
   referenceline y=-1;
endif;

/* test for the value a character dynamic */
```

```
   if ( upcase(ADDREF) =: "Y")
      referenceline y=1;
      referenceline y=0;
      referenceline y=-1;
   endif;

   /* test whether a dynamic is initialized */
   if (exists(ADDREF))
      referenceline y=1;
      referenceline y=0;
      referenceline y=-1;
   endif;
```

Conditional Logic Determines Statement Rendering

The GTL conditional logic is used only to determine which statements are rendered, not to control what is in the data object. In the following example, the data object contains columns for DATE, AMOUNT, and LOG(AMOUNT), but only one scatter plot is created.

```
if ( LOGFLAG )
  scatterplot x=date y=amount;
else
  scatterplot x=date y=log(amount);
endif;
```

Also, it is seldom necessary to test for the existence of option values set by columns or dynamics. Consider the following statement:

```
scatterplot x=date y=amount / group=GROUPVAR;
```

This SCATTERPLOT statement is equivalent to the following code because option values set by columns that do not exist or dynamics that are uninitialized simply “drop out” at run time and do not produce errors or warnings:

```
if ( exists(GROUPVAR) )
  scatterplot x=date y=amount / group=GROUPVAR;
else
  scatterplot x=date y=amount;
endif;
```

The GTL code that is conditional must be complete statements, or complete blocks of statements, or both. The following IF block produces a compile error because there are more LAYOUT statements than ENDLAYOUT statements:

```
/* this IF block produces a compile error */
if ( exists(SQUAREPLOT) )
  layout overlayequated / equatetype=square;
else
  layout overlay;
endif;
```

```
    scatterplot x=XVAR y=YVAR;
endlayout;
```

This is the correct conditional construct:

```
if ( exists(SQUAREPLOT) )
  layout overlayequated / equatetype=square;
    scatterplot x=XVAR y=YVAR;
  endlayout;
else
  layout overlay;
    scatterplot x=XVAR y=YVAR;
  endlayout;
endif;
```

GTL Does Not Provide ELSE IF Syntax

The GTL does not provide ELSE IF syntax, but you can create a nested IF/ELSE block as follows:

IF (*condition-1*)
  GTL-statement(s);
ELSE
  IF (*condition-2*)
    GTL-statement(s);
  ELSE
    GTL-statement(s);
  ENDIF;
ENDIF;

Part 9

Appendices

Appendix 1

Syntax Conventions and Argument Value Types

Syntax Conventions

< > ==> optional

a | b ==> either a or b

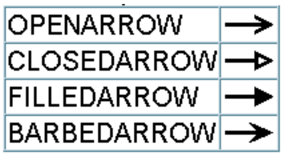
Value Types for Statement Options

The default value of an option depends on the template definition that is in use, and the implementation of that option depends on the ODS destination that formats the output. In addition, if you are creating HTML output, the implementation of an attribute depends on the browser that you use.

This section describes the value types that are available for GTL statement options.

arrow

specifies a string value of an arrow shape, a dynamic variable that contains an arrow shape, or a style reference to an arrow shape.



OPENARROW	→
CLOSEDARROW	⇾
FILLEDARROW	➔
BARBEDARROW	➝

boolean

specifies a string value that is true or false, or a style reference to a Boolean value.

For all attributes that support a value of ON, the following forms are equivalent:

```
ATTRIBUTE-NAME
ATTRIBUTE-NAME=ON
```

For all attributes that support a value of *column*, *column* can be any variable that you declare in the GTL template with the DYNAMIC, MVAR, or NMVAR statement. If the attribute is a Boolean value, then the value of variable should resolve to either true or false. The following values represent a true or false value.

Values for True	**Values for False**
True	False
ON	OFF
_ON_	_OFF_
TRUE	FALSE
YES	NO
YES	_NO_
1	0

color

specifies a string that identifies a color. A color can be one of the following:

- any of the color names that are supported by SAS/GRAPH. These names include
  - a predefined SAS color (for example, blue or VIYG)
  - a red/green/blue (RGB) value (for example, CX0023FF or #0023FF)
  - a hue/light/saturation (HLS) value (for example, H14E162D)
  - a gray-scale value (for example, GRAYBB).
- one of the colors that exists in the SAS session when the style definition is used, such as DMSBLACK or DMSCYAN. (Use these color specifications only if you are running SAS in the windowing environment.)
- an English description of an Hue/Light/Saturation (HLS) value. Such descriptions use a combination of words to describe the lightness, the saturation, and the hue (in that order). You can use the Color Naming System to form a color by doing one of the following:
  - combining a chromatic hue with a lightness, a saturation, or both
  - combining the achromatic hue gray with a lightness
  - combining the achromatic hue black or white without qualifiers.
  - combining words to form a wide variety of colors, such as light vivid green, dark vivid orange, or light yellow.
- specify hues that are intermediate between two neighboring colors. To do so, combine one of the following adjectives with one of its neighboring colors: brownish, greenish, purplish, or yellowish. For example, bluish purple or reddish orange.

For information about using SAS/GRAPH colors, see*SAS/GRAPH: Reference* .

column

specifies a column variable that contains either double-precision values or string values, or a dynamic variable that refers to such a column variable.

See also: *integer-column*, *numeric-column*, and *string-column*.

dimension

specifies a nonnegative number. The number can be followed by one of the following optional units of measure:

Unit	Description
%	percentage
PX	pixels
PT	point size, calculated at 100 dots per inch

The default value for the Printer destination is units of 1/150 of an inch.

expression

specifies a selective, relational, or logical program structure that calculates values when those values are not stored in the data. The expression must be specified as an EVAL() argument. The following shows the structure of an EVAL() argument:

```
x = EVAL(expression)
```

The expression returns a number and can be formed with consonants, data columns, dynamic variables, functions, or other expressions. The following example uses the data column Time and the SGE functions MEAN and ACF:

```
EVAL( MEAN( Time ) + ACF( Time, NLags=10 ))
```

For more information about expressions, see Chapter 54, “Expressions,” on page 545.

format

specifies a SAS format or a user-defined format.

integer, integer-column

specifies a member of the set of positive whole numbers, negative whole numbers, and zero.

An *integer-column* specifies a column variable that contains integer values, or a dynamic variable that refers to such a column variable.

line-pattern-name, line-pattern-number

specifies a string value of a line pattern, a numeric value of a line pattern, a dynamic variable that contains such a string or number, or a style reference to a line pattern. Line patterns are chosen for discriminability. Because of different densities, equal weighting is impossible for lines of the same thickness. Instead, line patterns are ordered to provide a continuum of weights, which is useful when displaying confidence bands.

For details about line attributes, see “Line Options” on page 564.

marker-name

specifies a string value of a maker symbol, a dynamic variable that contains a marker symbol, or a style reference to a marker symbol.

For details about marker attributes, see “Marker Options” on page 565.

number, numeric-column

specifies a value, a dynamic variable that contains a double-precision value, an expression that resolves to a double-precision value, or a style reference to a double-precision value.

A *numeric-column* specifies a column variable that contains double-precision values, or a dynamic variable that refers to such a column variable.

string, string-column

specifies a quoted character string.

A *string-column* specifies a column variable that contains string values, or a dynamic variable that refers to such a column variable.

style-reference

specifies a reference to an attribute that is defined in a style element.

In the ODS Graphics templates that SAS provides, options for plot features are specified with a style reference in the form *style-element:attribute*, rather than a specific value. For example, the symbol, color, and size of markers for a basic scatter plot is specified in a SCATTERPLOT statement as follows:

```
scatterplot x=X y=Y /
  markersymbol=GraphDataDefault:markersymbol
  markercolor=GraphDataDefault:contrastcolor
  markersize=GraphDataDefault:markersize
```

The above style references guarantee a common appearance for markers used in all basic scatter plots. For non-grouped data, the marker appearance is controlled by the GraphDataDefault style element in the style definition that you specify.

In order to create your own style definition, or to modify a style definition to use with ODS Graphics, you need to understand the relationship between style elements and graph features. For more information, see the usage guide.

Appendix 2
Display Attributes

General Syntax for Attribute Options

Most statements provide options that enable you to specify attributes for the fills, lines, data markers, or text that is used in the display. For example, many plots provide a DATALABELATTRS= option that specifies the attributes of the data labels. This appendix discusses the general syntax for those options and the valid values for they accept.

A statement's attribute options use the following general syntax:

OPTIONATTRS = *style-element* | *style-element (options)* | *(options)*

style-element
Name of a style element. Only style attributes relevant for rendering the fill, line, data marker, or text are used. Example:

```
DATALABELATTRS=GRAPHVALUETEXT
```

style-element (options)
Name of a style element, plus individual *options* to be used as style overrides. Any *options* not specified are derived from the specified *style-element*.Example:

```
DATALABELATTRS=GRAPHVALUETEXT(SIZE=10pt)
```

(options)
Individual *options*. Any *options* not specified are derived from the default style element. Examples:

```
DATALABELATTRS=(FAMILY="Arial" SIZE=10pt)
DATALABELATTRS=(FAMILY=GraphValueText:FontFamily)
```

Depending on the attribute option used, the *options* might be fill-options on page 564, line-options on page 564, marker-options on page 565, or text-options on page 566.

In general, any relevant attribute that is not specified default to some internal value, typically those from the default style element of the *style-element* you specify for the attributes. When choosing a style element, you should use an element of the correct type. See*SAS Output Delivery System: Procedures Guide* for a list of style elements and their types.

Attributes Available for the Attribute Options

Depending on the attribute option used on a statement, the available attributes might be fill-options on page 564, line-options on page 564, marker-options on page 565, or text-options on page 566.

Fill Options

When specifying the attributes for an area fill, the COLOR *fill-option* is the only option that is currently available. The option must be enclosed in parentheses and specified as a *name* = *value pair*. The *value* can be a *style-reference* in the form *style-element:style-attribute*.

COLOR=*style-reference* | *color*

specifies the fill color. If you use a *style-reference*, the *style-attribute* should be a valid attribute such as COLOR, CONTRASTCOLOR, STARTCOLOR, NEUTRAL, ENDCOLOR. The convention is to use the COLOR attribute for fill areas.

If you use a *color*, SAS/GRAPH accepts color names, such as RED, or color codes, such as CXFF0000 or #FF0000. Color names must not exceed 64 characters. Color codes must not exceed 8 characters and must be in a valid SAS color-naming scheme, such as RGB, CMYK, HLS, or HSV (HSB).

For more information about using colors in SAS/GRAPH, see the discussion on color in *SAS/GRAPH: Reference* .

Line Options

When specifying the attributes for a line, the available *line-options* can be any one or more of the following settings. The options must be enclosed in parentheses, and each option is specified as a *name* = *value pair*. In all cases, the *value* can be a *style-reference* in the form *style-element:style-attribute*.

COLOR=*style-reference* | *color*

specifies the line color. If you use a *style-reference*, the *style-attribute* should be a valid attribute such as COLOR, CONTRASTCOLOR, STARTCOLOR, NEUTRAL, ENDCOLOR. The convention is to use CONTRASTCOLOR for lines. If you specify a style element that does not have a CONTRASTCOLOR attribute, the element's COLOR attribute is used.

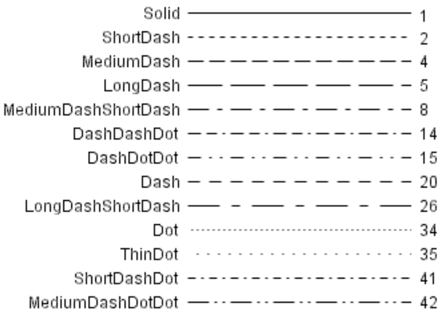
If you use a *color*, SAS/GRAPH accepts color names, such as RED, or color codes, such as CXFF0000 or #FF0000. Color names must not exceed 64 characters. Color codes must not exceed 8 characters and must be in a valid SAS color-naming scheme, such as RGB, CMYK, HLS, or HSV (HSB).

For more information about using colors in SAS/GRAPH, see the discussion on color in *SAS/GRAPH: Reference* .

PATTERN=*style-reference* | *line-pattern-name* | *line-pattern-number*

specifies the line pattern. If you use a *style-reference*, the *style-attribute* should be LINESTYLE.

Line patterns can be specified as a pattern name or pattern number.



Valid pattern numbers range from 1 to 46. Not all pattern numbers have names. See "Available Line Patterns" on page 566 for a list of all possible line patterns. We recommend that you use the named patterns because they have been optimized to provide good discriminability when used in the same plot.

THICKNESS=*style-reference* | *dimension*

specifies the line thickness. If you use a *style-reference*, the *style-attribute* should be LINETHICKNESS.

Marker Options

When specifying the attributes for a data marker, the available *marker-options* can be any one or more of the following settings. The options must be enclosed in parentheses, and each option is specified as a *name = value pair*. In all cases, the *value* can be a *style-reference* in the form *style-element:style-attribute*.

COLOR=*style-reference* | *color*

specifies the color of the marker. If you use a *style-reference*, the *style-attribute* should be a valid attribute such as COLOR, CONTRASTCOLOR, STARTCOLOR, NEUTRAL, ENDCOLOR. The convention is to use CONTRASTCOLOR for markers. For grouped data, this option keeps all markers the same color and the marker symbol alone distinguishes the group values.

If you use a *color*, SAS/GRAPH accepts color names, such as RED, or color codes, such as CXFF0000 or #FF0000. Color names must not exceed 64 characters. Color codes must not exceed 8 characters and must be in a valid SAS color-naming scheme, such as RGB, CMYK, HLS, or HSV (HSB).

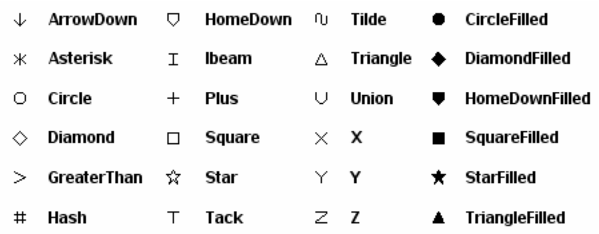
For more information about using colors in SAS/GRAPH, see the discussion on color in *SAS/GRAPH: Reference* .

SIZE=*style-reference* | *dimension*

specifies the marker size (both width and height). If you use a *style-reference*, the *style-attribute* should be MARKERSIZE.

SYMBOL=*style-reference* | *marker-name*

specifies the name of the marker. If you use a *style-reference*, the *style-attribute* should be MARKERSYMBOL. The following symbols are supported:



↓	ArrowDown	▽	HomeDown	∿	Tilde	●	CircleFilled
⁎	Asterisk	I	Ibeam	△	Triangle	◆	DiamondFilled
○	Circle	+	Plus	∪	Union	▼	HomeDownFilled
◇	Diamond	□	Square	×	X	■	SquareFilled
>	GreaterThan	☆	Star	Y	Y	★	StarFilled
#	Hash	T	Tack	Z	Z	▲	TriangleFilled

WEIGHT=NORMAL | BOLD
specifies the marker weight.

Text Options

When specifying the attributes for text, the available *text-options* can be any one or more of the following settings. The options must be enclosed in parentheses, and each option is specified as a *name = value pair*. In all cases, the *value* can be a *style-reference* in the form *style-element:style-attribute*.

COLOR=*style-reference* | *color*
specifies the color of the text. If you use a *style-reference*, the *style-attribute* should be a valid attribute such as COLOR, CONTRASTCOLOR, STARTCOLOR, NEUTRAL, ENDCOLOR. The convention is to use COLOR for text.

If you use a *color*, SAS/GRAPH accepts color names, such as RED, or color codes, such as CXFF0000 or #FF0000. Color names must not exceed 64 characters. Color codes must not exceed 8 characters and must be in a valid SAS color-naming scheme, such as RGB, CMYK, HLS, or HSV (HSB).

For more information about using colors in SAS/GRAPH, see the discussion on color in *SAS/GRAPH: Reference* .

FAMILY =*style-reference* | *"string"*
specifies the font family of the text. If you use a *style-reference*, the *style-attribute* should be FONTFAMILY.

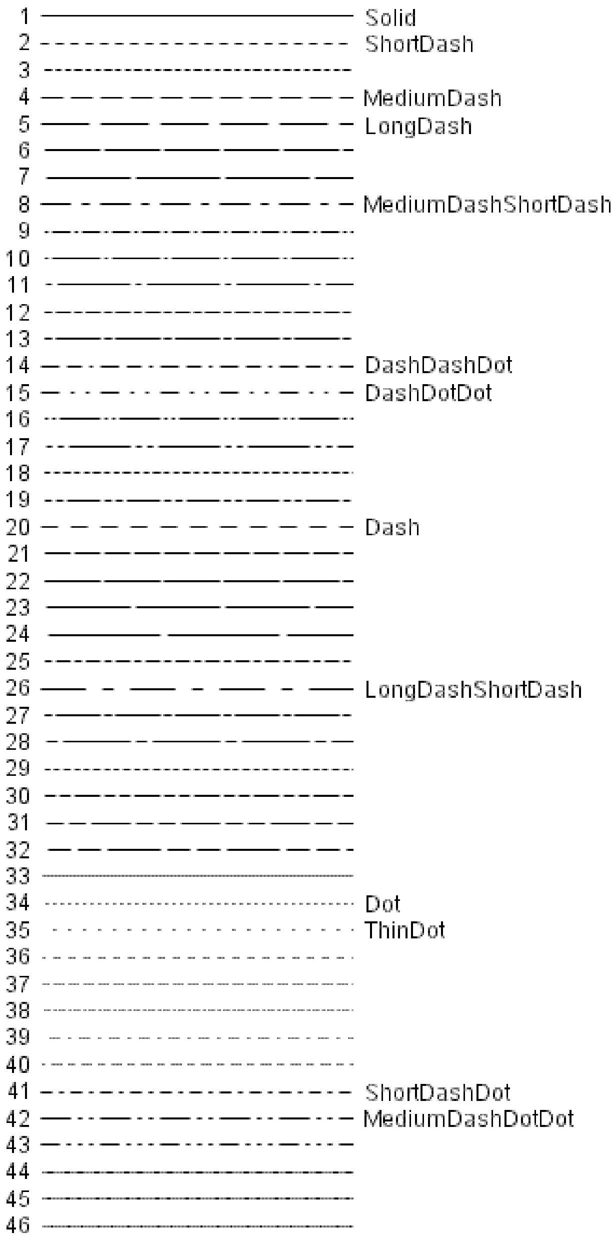
SIZE=*style-reference* | *dimension*
specifies the font size of the text. If you use a *style-reference*, the *style-attribute* should be FONTSIZE.

STYLE=*style-reference* | NORMAL | ITALIC
specifies the font style of the text. If you use a *style-reference*, the *style-attribute* should be FONTSTYLE.

WEIGHT=*style-reference* | NORMAL | BOLD
specifies the font weight of the text. If you use a *style-reference*, the *style-attribute* should be FONTWEIGHT.

Available Line Patterns

The following line patterns can be used with the Graphics Template Language. A line pattern can be specified by its number or name. Not all patterns have names. We recommend that you use the named patterns because they have been optimized to provide good discriminability when used in the same plot.


1 Solid
2 ShortDash
3
4 MediumDash
5 LongDash
6
7
8 MediumDashShortDash
9
10
11
12
13
14 DashDashDot
15 DashDotDot
16
17
18
19
20 Dash
21
22
23
24
25
26 LongDashShortDash
27
28
29
30
31
32
33
34 Dot
35 ThinDot
36
37
38
39
40
41 ShortDashDot
42 MediumDashDotDot
43
44
45
46

Appendix 3
SAS Formats Not Supported

Using SAS Formats

SAS formats can be assigned to input data columns with the FORMAT statement of the SGRENDER procedure. Additionally, several GTL statement options enable a SAS format as an option value. Examples include the TICKVALUEFORMAT= option for formatting axis tick values, and the TIPFORMAT= option for formatting data tips.

Not all SAS formats are supported in the GTL or with the SGPLOT, SGSCATTER, SGPANEL, and SGRENDER procedures. The tables in the following sections show the character and numeric SAS formats that are not supported.

When the GTL encounters an unsupported format, a note similar to the following is written to the SAS log:

```
TICKVALUEFORMAT=bestx. is invalid. The format is invalid or unsupported. The
default will be used.
```

Unsupported Numeric Formats

The following numeric formats are not supported in the GTL:

BESTD	BESTX	D	FLOAT	FRACT
FREE	IB	IBR	IEEE	IEEER
ODDSR	PCPIB	PD	PIB	PIBR

PK	RB	SSN	WORDF	WORDS
Z	ZD			

Unsupported Date and Time Formats Related to IS0 8601

The following date and time formats are not supported in the GTL:

$N8601B	$N8601BA	$N8601E	$N8601EA	$N8601EH
$N8601EX	$N8601H	$N8601X	B8601DA	B8601DN
B8601DT	B8601DZ	B8601LZ	B8601TM	B8601TZ
E8601DA	E8601DN	E8601DT	E8601DZ	E8601LZ
E8601TM	E8601TZ	IS8601DA	IS8601DN	IS8601DT
IS8601DZ	IS8601LZ	IS8601TM	IS8601TZ	

Other Unsupported Date and Time Formats

The following date and time formats are not supported in the GTL:

HDATE	HEBDATE	JDATEMDW	JDATEMNW	JDATEWK
JDATEYDW	JDATEYM	JDATEYMD	JDATEYMW	JDATEYT
JDATEYTW	JNENGO	JNENGOT	JNENGOTW	JNENGOW
JTIMEH	JTIMEHM	JTIMEHMS	JTIMEHW	JTIMEMW
JTIMESW	MDYAMPM	MINGUO	NENGO	NLDATEYQ
NLDATEYR	NLDATEYW	NLDATMYQ	NLDATMYR	NLDATMYW
NLSTRMON	NLSTRQTR	NLSTRWK	PDJULG	PDJULI
TWMDY	XYYMMDD	YYQZ		

Unsupported Currency Formats

The following currency formats are not supported in the GTL:

EURFRATS	EURFRBEF	EURFRCHF	EURFRCZK	EURFRDEM
EURFRDKK	EURFRESP	EURFRFIM	EURFRFRF	EURFRGBP
EURFRGRD	EURFRHUF	EURFRIEP	EURFRITL	EURFRLUF
EURFRNLG	EURFRNOK	EURFRPLZ	EURFRPTE	EURFRROL
EURFRRUR	EURFRSEK	EURFRSIT	EURFRTRL	EURFRYUD
EURTOATS	EURTOBEF	EURTOCHF	EURTOCZK	EURTODEM
EURTODKK	EURTOESP	EURTOFIM	EURTOFRF	EURTOGBP
EURTOGRD	EURTOHUF	EURTOIEP	EURTOITL	EURTOLUF
EURTONLG	EURTONOK	EURTOPLZ	EURTOPTE	EURTOROL
EURTORUR	EURTOSEK	EURTOSIT	EURTOTRL	EURTOYUD

Appendix 4

Computing Input Data for BOXPLOTPARM

The following SAS code is a generalized macro for computing input data for BOXPLOTPARM.

```
%macro boxcompute(indsn=,x=,y=,outdsn=boxdata,datalabel=,
                  qntldef=5,table=no);
/* NOTE: INDSN, X and Y are required parameters, where
   INDSN = input SAS data set
   X = categorical variable (num or char)
   Y = response variable (num)
   OUTDSN = output dataset. It contains these variables:
      STAT: Statistic names for BOXPLOTPARM
      VALUE: values for STAT type
      X: X variable values
      DATALABEL: outlier labels from the DATALABEL= variable
      N, Mean, Median, Std if TABLE=YES
   DATALABEL= variable used to label outliers (num or char)
   QNTLDEF = 1|2|3|4|5
     (how to compute quantiles - see PROC SUMMARY)
   TABLE = YES | NO
     (add additional data to build table of statistics)
 */
  %macro varinfo(dsid,varname,role,rc);
   /* utility macro for obtaining variable info */
    %local varnum;
    %if %length(&varname)=0 %then %do;
      %let &rc=0;  %return;
    %end;
    %let varnum=%sysfunc(varnum(&dsid,&varname));
    %if &varnum > 0 %then %do;
      %let &role.label=%sysfunc(varlabel(&dsid,&varnum));
      %if  %length(&&&role.label)=0 %then
         %let &role.label=%sysfunc(varname(&dsid,&varnum));
      %let &role.fmt=%sysfunc(varfmt(&dsid,&varnum));
      %let &rc=0;
    %end;
    %else %do;
      %put ERROR: %upcase(&role) variable &varname not found.;
      %let &rc=1;
    %end;
  %mend varinfo;

/* validate dataset and variables */
%local dsid ylabel xlabel datalabellabel
       yfmt xfmt datalabelfmt rc_y rc_x rc_d;
```

```
%let dsid=%sysfunc(open(&indsn));
%if &dsid %then %do;
   %varinfo(&dsid,&y,Y,rc_y)
   %varinfo(&dsid,&x,X,rc_x)
   %if %length(&datalabel) %then
      %varinfo(&dsid,&datalabel,DATALABEL,rc_d);
   %else %let rc_d=0;
   %let dsid=%sysfunc(close(&dsid));
   %if &rc_y or &rc_x or &rc_d %then %return;
%end;
%else %do;
   %put ERROR: Input dataset &indsn not found.;
   %return;
%end;

/* compute basic summary statistics */
proc summary data=&indsn(rename=(&y=VALUE &x=X))
             nway qntldef=&qntldef;
  class x;
  var value;
  output out=summary(drop=_type_ _freq_) n=N mean=Mean
             median=Median q1=Q1 q3=Q3 std=STD / noinherit;
run;
proc sort data=&indsn(keep=&x &y &datalabel)
  %if %length(&datalabel) %then
    out=sorted(rename=(&x=X &y=VALUE &datalabel=DATALABEL));
  %else out=sorted(rename=(&x=X &y=VALUE));
;  by &x;
run;
/* compute fences, MIN, MAX and any outliers for X values */
data outliers;
  length STAT $10;
  %if %length(&datalabel) %then
   %do;
    keep STAT X VALUE DATALABEL;
    label VALUE="&ylabel" X="&xlabel"
          DATALABEL="&datalabellabel";
    format VALUE &yfmt X &xfmt DATALABEL &datalabelfmt;
   %end;
  %else
   %do;
    keep STAT X VALUE;
    label VALUE="&ylabel" X="&xlabel";
    format VALUE &yfmt X &xfmt;
   %end;
  retain lowerFence upperFence farLowerFence farUpperFence
         tempmin tempmax;
  merge sorted summary; by x;

  /* perform computations for each X value */
  if first.X then do;
     lowerFence=q1-((q3-q1)*1.5);
     upperFence=q3+((q3-q1)*1.5);
     farLowerFence=q1-((q3-q1)*3);
     farUpperFence=q3+((q3-q1)*3);
     /* these computations for MIN and MAX result
```

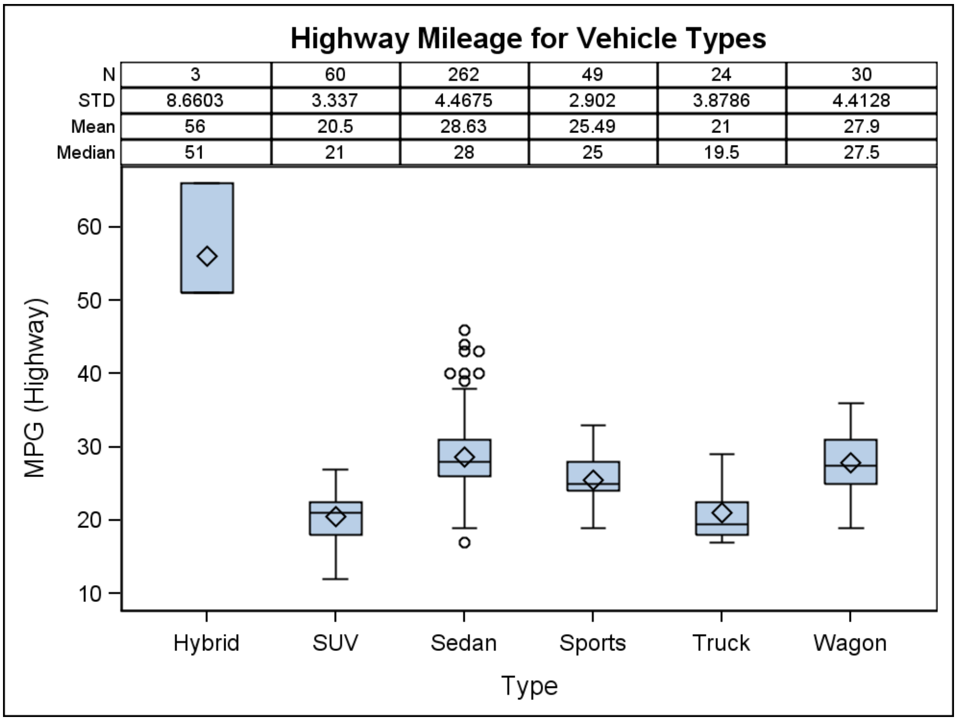
```
         in the same values produced by the BOXPLOT
         statement, however they can be modified to
         satify other statistical definitions */
      if value <= upperFence then tempmax=value;
      else tempmax=.;
      if value => lowerFence then tempmin=value;
      else tempmin=.;
   end;
   /* recompute MAX and MIN for each obs */
   if 0 <= sum(upperFence,-value) then
      tempmax=max(tempmax,value);
   if 0 <= sum(value,-lowerFence) then
      tempmin=min(tempmin,value);
   /* write out both types of outliers */
   if value < farLowerFence or value > farUpperFence then do;
      stat="FAROUTLIER"; output;
   end;
   else if value < LowerFence or value > UpperFence then do;
      stat="OUTLIER"; output;
   end;
   /* write out MIN and MAX for each X value */
   if last.X then do;
     value=tempmin; stat="MIN"; output;
     value=tempmax; stat="MAX"; output;
   end;
 run;
 /* transpose the stats into the columns
    required by BOXPLOTPARM */
 data transpose(keep=x stat value);
   length STAT $10;
   set summary;
   array stats{*} n--std;
   do i=1 to dim(stats);
     stat=upcase(vname(stats{i}));
     VALUE=stats{i};
     output;
   end;
 run;
 /* interleave the obs by the X variable */
 data &outdsn;
   set transpose outliers; by X;
 run;
 /* merge the output stats for building a stat table */
 %if %upcase(&table)=YES %then %do;
   data &outdsn;
     merge &outdsn summary; by X;
   run;
 %end;
 %mend boxcompute;
```

Here is the macro invocation to produce the data for the graph shown in the section “Example Program and Statement Details” on page 191 of “BOXPLOTPARM Statement” on page 191 .

```
%boxcompute(indsn=sashelp.cars,x=type,y=mpg_city,
            datalabel=make);
```

```
proc template;
  define statgraph boxplotparm1;
    begingraph;
      entrytitle "City Mileage for Vehicle Types";
      layout overlay;
        boxplotparm y=value x=x stat=stat /
         datalabel=datalabel spread=true ;
      endlayout;
    endgraph;
  end;
run;

proc sgrender data=boxdata template=boxplotparm1;
run;
```

The following figure and code show an example of a “table” of statistics with BLOCKPLOT statements.



Highway Mileage for Vehicle Types

N	3	60	262	49	24	30
STD	8.6603	3.337	4.4675	2.902	3.8786	4.4128
Mean	56	20.5	28.63	25.49	21	27.9
Median	51	21	28	25	19.5	27.5

```
%boxcompute(indsn=sashelp.cars,x=type,y=mpg_highway,
            outdsn=boxdata2,table=yes);

proc template;
  define statgraph boxplotparm2;
    begingraph;
      entrytitle "Highway Mileage for Vehicle Types";
      layout lattice / rowweights=(.04 .04 .04 .04 .84);
        blockplot x=x block=n /
          display=(values label outline) valuehalign=center
          labelattrs=graphdatatext valueattrs=graphdatatext;
        blockplot x=x block=std /
          display=(values label outline) valuehalign=center
          labelattrs=graphdatatext valueattrs=graphdatatext;
        blockplot x=x block=mean /
```

```
          display=(values label outline) valuehalign=center
          labelattrs=graphdatatext valueattrs=graphdatatext;
        blockplot x=x block=median /
          display=(values label outline) valuehalign=center
          labelattrs=graphdatatext valueattrs=graphdatatext;
        boxplotparm y=value x=x stat=stat /
          datalabel=datalabel spread=true;
      endlayout;
    endgraph;
  end;
run;

proc sgrender data=boxdata2 template=boxplotparm2;
run;
```

Appendix 5

Memory Management for ODS Graphics

SAS Options Affecting Memory

ODS Graphics uses Java technology to produce its graphs. Most of the time this fact is transparent to you because the required Java Runtime Environment (JRE) and Jar files are included with SAS software installation. Also, the Java environment is automatically started and stopped for you. When Java is started, it allocates a fixed amount of memory. The memory can grow up to the value set for the -Xmx suboption in the JREOPTIONS option (discussed in a moment). This memory is independent of the memory limit that SAS sets for the SAS session with its MEMSIZE= option.

Normally, the memory limit for Java is sufficient for most ODS Graphics applications. However, some tasks are very memory intensive and might exhaust all available Java memory, resulting in an OutOfMemoryError condition. You might encounter Java memory limitations when

- the product of the output size and the DPI setting results in very large output
- a classification panel has a very large number of classifier crossings
- a scatter plot matrix has a large number of variables
- creating 3-D plots and 2-D contours, which are memory intensive to generate
- a plot has a very large number of marker labels
- a plot uses many character variables or has a large number of GROUP values
- using the SG Editor to edit a graph with a large amount of data.

Managing a Java Out Of Memory Error

If you encounter a Java OutOfMemoryError, you can try executing your program again by restarting SAS and specifying a larger amount of memory for Java at SAS invocation.

To determine what the current Java memory settings are, you can submit a PROC OPTIONS statement that shows the value of the JREOPTIONS option:

```
proc options option=jreoptions;
run;
```

After you submit this procedure code, a list of JREOPTIONS settings is displayed in the SAS log. The JREOPTIONS option has many suboptions that configure the SAS Java environment. Many of the suboptions are installation and host specific and should not be modified, especially the ones that provide installed file locations. For managing memory, look for the -Xmx and -Xms suboptions:

```
JREOPTIONS=(  /* other Java suboptions */ -Xmx128m -Xms128m  )
```

-Xms
: Use this option to set the minimum Java memory (heap) size, in bytes. Set this value to a multiple of 1024 greater than 1MB. Append the letter k or K to indicate kilobytes, or m or M to indicate megabytes. The default is 2MB. Examples:

```
-Xms6291456
-Xms6144k
-Xms6m
```

-Xmx
: Use this option to set the maximum size, in bytes, of the memory allocation pool. Set this value to a multiple of 1024 greater than 2MB. Append the letter k or K to indicate kilobytes, or m or M to indicate megabytes. The default is 64MB. Examples:

```
-Xmx83886080
-Xmx81920k
-Xmx80m
```

As a general rule, you should set the minimum heap size (-Xms) equal to the maximum heap size (-Xmx) to minimize garbage collections.

Typically, SAS sets both -Xms and -Xmx to be about 1/4 of the total available memory or a maximum of 128M. However, you can set a more aggressive maximum memory (heap) size, but it should never be more than 1/2 of physical memory.

You should be aware of the maximum amount of physical memory your computer has available. Let us assume that doubling the Java memory allocation is feasible. So when you start SAS from a system prompt, you can add the following option:

```
-jreoptions (-Xmx256m -Xms256m)
```

Alternatively, you might need to specify the setting in quotation marks:

```
-jreoptions '(-Xmx256m -Xms256m)'
```

The exact syntax varies for specifying Java options, depending on your operating system, and the amount of memory that you can allocate varies from system to system. The set of JRE options must be enclosed in parentheses. If you specify multiple JREOPTIONS system options, SAS appends JRE options to JRE options that are currently defined. Incorrect JRE options are ignored.

If you choose to create a custom configuration file, you would simply replace the existing -Xms and -Xmx suboption values in the JREOPTIONS=(*all Java options*) portion of the configuration file.

For more information, see the SAS Companion for your operating system.

Glossary

anti-aliasing

a rendering technique for improving the appearance of text and curved lines in a graph by blurring the jagged edges normally present. The degree of improvement is relative to the nature of the graphical content (for example, vertical and horizontal lines do not benefit from anti-aliasing). Extra processing is required to perform anti-aliasing.

attribute bundle

a common collection of visual properties associated with a graphical primitive such as a line, marker, or text. For example, all lines have visual properties of pattern, thickness, and color. All markers have visual properties of symbol, size, weight, and color. Attribute bundles can be associated with style elements in order to indirectly assign visual properties.

axis

a graphical element used to locate or identify the values of other graphical elements, such as points or bars. An axis consists of an axis line with tick marks, tick values, and a descriptive label. Not all parts of the axis need to be displayed. An axis is typically an interval axis (linear, log, or time) or a discrete axis. A two-dimensional graph can have up to four independent axes: X, Y, X2, and Y2. A three-dimensional graph has three independent axes: X, Y, and Z.

axis offset

the gaps that normally appear at the ends of an axis line. The gaps enable markers, bars, and other graphic primitives that are drawn at extreme data values to be rendered without clipping. An offset can also be used to add extra space between an axis line and visual elements in the graph. An offset distance is expressed as a value from 0 to 1, which represents a percentage of axis length to the axis data minimum or axis data maximum. An offset can be specified for either end of any axis.

axis threshold

a numerical bias from 0 to 1 that determines whether an extra tick is added at either end of a non-discrete, interval axis. If the minimum and maximum thresholds are set to 0, then no ticks are added beyond the actual data range. If both minimum and maximum thresholds are set to 1, then the data range is completely bounded by the first and last ticks.

axis tick

a short line segment perpendicular to the axis line. A tick can cross the axis line, or be drawn from the axis inside or outside the wall.

axis tick value

a formatted data value represented by a tick.

axis type

a keyword denoting axis functionality. For example, the axis type of interval axes can be LINEAR, TIME, or LOG. The axis type of a discrete axis is DISCRETE.

axis viewport

the range of values displayed on an interval axis. This range can be larger or smaller than the actual data range of the axis. An axis viewport that is larger than the data range effectively zooms out from the plot or plots. An axis viewport that is smaller than the data range zooms in on the plot or plots.

band plot

a plot that draws a horizontal band using two Y values for each X value, or that draws a vertical band using two X values for each Y value. A band plot is typically used to show confidence, error, prediction, or control limits. The points on the upper and lower band boundaries can be joined to create two outlines, or the area between the boundaries can be filled.

binned data

data that has been summarized or transformed in some way to facilitate its rendering by a parameterized plot. Continuous numeric data is typically binned by setting a bin width (interval size) and then computing the number of bins, or by setting the number of bins and computing the bin width. A histogram is often used to represent binned data.

bins

numeric intervals into which continuous numeric data can be categorized.

block plot

a plot that displays one or more rectangles (blocks) along an axis, where each rectangle identifies a block of consecutive observations having the same value for a specified block variable. The first block begins at the start of the axis (mapped to the values of a specified variable), and represents the first observation s block value, and continues through consecutive observations having the same block value. A change in the block variable s value ends the first block and starts the second, which continues through consecutive observations until the block value changes again. The last block extends to the end of the axis.

border

the outermost outline of a graph, a layout container, or a graphical element (for example, a legend or text).

category variable

a classification variable with a finite number of distinct (discrete) values. These variables are typically used to split data into subsets. For example, in a bar chart, each unique value is displayed as a bar on a DISCRETE axis.

cell

in ODS graphics, a distinct rectangular subregion of a graph that can contain plots, text, or legends. For example, a graph defined with the OVERLAY layout contains only one cell, whereas a graph defined with the GRIDDED layout can contain several cells.

cell block
a block beginning with a CELL statement and ending with an ENDCELL statement that defines the graphical content of a cell. The cell block is available only within a LATTICE layout.

cell header
a graphical element (typically text or a legend) that is aligned at the top of a cell and provides information about the cell contents. A cell header is defined within a cell block, which is available only within a LATTICE layout.

child block
a block that is contained within another block when two or more blocks are nested. For example, a CELLHEADER block is always a child of a CELL block.

classification level
for a single classification variable, each unique value is regarded as a classification level. For two or more variables, a classification level is one of the unique combinations (crossings) of the unique values of each variable. For example, if three variables have four, two, and three distinct values, there are 24 classification levels.

classification panel
a multi-cell graph created by a layout, such as DATAPANEL, in which the number of cells is determined by the number of classification levels of one or more classification variables. Each cell displays a common plot based on subsets of the input data.

classification variable
a variable whose values are used to group (or classify) the observations in a data set into different groups that are meaningful for analysis. A classification variable can have either character or numeric values. Classification variables include group, subgroup, category, and BY variables.

clip
to truncate a plot or graphical element (such as a line, marker, or band) when it reaches a boundary such as a plot wall.

column
a vertical component of a table. Each column has a unique name, contains data of a specific type, and has certain attributes. A column is analogous to a variable in SAS terminology. In the Graph Template Language a column can also be a set of layout cells, stacked vertically and sharing the same alignment.

column axis
an external axis appearing above or below a column of cells and serving as a common reference for the column of a multi-cell layout, such as a LATTICE, DATAPANEL, or DATALATTICE layout.

column gutter
the space between columns of cells in a multi-cell layout.

column header
text that labels the column contents in a multi-cell layout. This text can be aligned above or below the cells in a column. In a LATTICE layout, the column header is not restricted to text (it can contain a plot or a legend, for example).

column major order

an order for populating cells of a layout or entries in a legend when the number of rows is specified. By default, cells or entries are filled starting from the top left and moving down. When the bottom row of the first column is filled, a new column begins filling to the right of the previous column, and so on until all content items have been placed in cells or entries. There might be empty cells or entries in the last column.

column weight

in a LATTICE layout, the proportion of width allotted to a specific column of the layout. The sum of all column weights is 1.

computed plot

a plot in which input data is internally summarized or otherwise transformed to create new data that is actually rendered by the plot. Examples of computed plot statements are BARCHART, BOXPLOT, HISTOGRAM, ELLIPSE, and REGRESSIONPLOT.

conditional logic

syntax that enables one set of statements or an optional alternate set of statements to execute at run time. In the Graph Template Language, an IF/ENDIF block defines conditional logic: IF (condition) statements; ELSE statements; ENDIF; The ELSE statement is not required.

continuous legend

a legend that shows a mapping between a color ramp or color segments and corresponding numeric values. Plots that support a COLORMODEL= option can use this type of legend.

crossing

a combination of the unique values of one or more classification variables. See also classification level.

cube

in three-dimensional graphics, the outlines formed by the intersection of three pairs of parallel planes; each pair is orthogonal to the primary X, Y, and Z axes. The display of the cube is optional.

data object

a transient version of a SAS data set created by ODS. When an input SAS data set is bound to a compiled graph template, an ODS data object is created, based on all the columns requested in the template definition and any new columns that have been directly or indirectly computed. A data object can persist when used with the ODS OUTPUT statement.

data tip

data or other detailed information that is displayed when a user positions a mouse pointer over an element in a graph. For example, a data tip typically displays the data value that is represented by a bar, a plot point, or some other element.

define block

in the TEMPLATE procedure, a define block (beginning with a DEFINE statement and ending with an END statement) creates various types of templates, including STATGRAPH, STYLE, and TABLE.

dependent plot

a plot that cannot be rendered by itself. Dependent plots must be overlaid with a stand-alone plot. Dependent plots do not provide data ranges to establish axes.

REFERENCELINE, DROPLINE, and LINEPARM statements produce dependent plots. See also stand-alone plot.

dependent variable
a variable that is observed to change in response to the independent variables. In a function y=f(x), the value of the dependent variable y is a function of the independent variable x.

design size
the intended size of a graph that is specified in the graph template definition. The DESIGNHEIGHT and DESIGNWIDTH options of the BEGINGRAPH statement set the intended height and width, which are used to determine the scale factors when the graph is resized. The intended height and width are used unless overridden by the ODS Graphics statement HEIGHT or WIDTH options when the template is executed.

device-based graphic
a graph created with traditional SAS/GRAPH software that requires DEVICE specification. ODS graphics (template-based graphics) do not use device technology.

discrete axis
an axis for categorical data values. The distance between ticks has no significance. A bar chart always has a discrete axis.

discrete legend
a legend that provides values or descriptive information about graphical elements in a grouped or overlaid plot.

dots per inch
a measure of the graph resolution by its dot density. Short form: DPI.

DPI
See dots per inch.

drop line
a line drawn from a point in the plot area perpendicular to an axis.

dynamic variable
a variable defined in a template with the DYNAMIC statement that can be initialized at template run time.

equated axes
in two-dimensional plots, axes that use the same drawing scale (ratio of display distance to data interval) on both axes. For example, an interval of 2 on the X axis maps to the same display distance as an interval of 2 on the Y axis. The aspect ratio of the plot display equals the aspect ratio of the plot data. In other words, a 45-degree slope in data will be represented by a 45-degree slope in the display. Equated axes are always of TYPE=LINEAR. The number of intervals displayed on each axis does not have to be the same.

external axis
an axis that is outside all cells of a layout. An external axis represents a common scale for all plots in a row or column of a multi-cell layout.

fill

to apply a color within a bounded area. Many plots, such bar charts and band plots, have bounded areas that can be filled or unfilled. When filled, a color is applied. When unfilled, the areas are transparent.

fit policy

one of several algorithms for avoiding tick-value collision when space allotted to a predefined area does not permit all the text to fit. For example, an axis might have a THIN policy that eliminates the display of tick values for alternate ticks. A ROTATE policy would turn the tick values at a 45-degree angle. A TRUNCATE policy would truncate all long tick values to a fixed length and add an ellipsis (. . .) at the end to imply truncation. A STAGGER policy would create two rows of tick values with consecutive tick values alternating between rows. A compound policy such as STAGGERROTATE could be used to automatically choose the best fit policy for the situation.

footnote area

the region below the graph area where text produced by ENTRYFOOTNOTE statements appears.

frequency variable

in an input data set, a non-negative and non-zero integer variable that represents the frequency of occurrence of the current observation, essentially treating the data set as if each observation appeared <userSuppliedValue>n</userSuppliedValue> times, where <userSuppliedValue>n</userSuppliedValue> is the value of the FREQ variable for the observation.

fringe plot

a plot consisting of short, equal-length line segments drawn from and perpendicular to an axis. Each observation of a numeric variable corresponds to the location for a line segment.

glyph

a letter, character, punctuation mark, pictogram, or symbol that is rendered in the context of some written language. A typeface (font) consists of a coordinated set of glyphs. See also Unicode.

graph

a visualization created by SAS/GRAPH software. A graph that is created by the ODS Graphics system can contain titles, footnotes, legends, and one or more cells, and is typically saved as an image or an SGE file. A generic term for final graphical output without regard to content or format.

graph(ical) area

the region where the visualization displays between the title area and footnote area. The graphical area consists of one or more cells. See also title area and footnote area.

graph(ics) template

See ODS Graphics template.

grid

rows and columns of a multi-cell layout.

gridded data

input that contains at least three numeric variables. Two of the variables are treated as X and Y variables and the third variable Z is treated as if it were a function of X and Y. The X and Y variable values occur at uniformly spaced intervals (although the size and number of intervals might be different for X and Y). All X,Y pairs are unique, and Z values are interpolated so that every X,Y pair has a Z value. Raw data that has at least three numeric variables can be converted to gridded data with the G3GRID procedure (in SAS/GRAPH). The procedure offers both bivariate and spline interpolation methods for computing Z values.

group index

a numeric variable with positive integer values that correspond to values of a group variable. The index values are used to associate GraphData1 GraphData<userSuppliedValue>N</userSuppliedValue> style elements with group values.

group variable

a variable in the input data set used to categorize chart variable values into groups. A group variable enables the data for each distinct group value to be rendered in a visually different manner. For example, a grouped scatter plot displays a distinct marker and color for each group value.

gutter

the space between columns or rows of cells in a multi-cell layout.

image format

a file format that displays a graphical representation. PNG, GIF, TIFF, and JPEG are examples of image formats, each with different characteristics.

image map

in an HTML file, the information contained in the map tag. This information can be used by a browser to display data tips on the image.

independent variable

a variable that persists and affects dependent variables. For a function y=f(x), the value of the dependent variable y is a function of the independent variable x. For example in a REGRESSIONPLOT statement, the X= variable is the independent variable.

inset

a graphical element such as a legend, line of text, or a table of text that is embedded inside of a graph's plot area.

interval axis

an axis where the distance between tick marks represents monotonically increasing or decreasing numeric units of some scale (like a ruler). The standard interval axis is called a LINEAR axis. Specialized interval axes include a TIME axis and a LOG axis.

layout

a generic term for a rectangular container that lays out the positions and sizes of its child components.

layout block

a block beginning with a LAYOUT statement and ending with an ENDLAYOUT statement.

layout grid

a multi-cell layout arranged as a grid of cells in rows and columns.

layout type
a keyword indicating the functionality of the layout. For example OVERLAY, LATTICE, and DATAPANEL are layout types.

legend entry
a combination of a graphical element such as a marker or line along with text describing the value or use of the graphical element. A discrete legend can have several legend entries.

legend title
text that explains how to interpret the legend.

line property
a value that defines the pattern, thickness, or color of a line. By default, the value for a line property is derived from a style element in the current style.

linear axis
an interval axis with ticks placed on a linear scale.

loess plot
a curved line showing a loess fit for a set of points.

log axis
an interval axis with ticks placed on a log scale with a base of 10, 2, or e. A log axis can show ticks 1) that are uniformly spaced representing integer powers of the base, 2) that are uniformly spaced representing the expanded integer powers of the base, or 3) that are uniformly spaced in the data domain (100, 200, 300), but may be shown in a non-uniform spacing on the axis.

macro variable reference
in a template definition, a reference to a macro variable that has been declared with MVAR or NMVAR statements. These references are meant to be resolved at template run time and should not be preceded with an ampersand. If a standard macro variable reference (a name preceded with an ampersand) appears in a template definition, it is resolved at template compile time.

marker
a symbol such as a dot, a cross, a diamond, a circle, or a triangle that is used to indicate the location of, or annotate, a data point in a plot or graph.

marker property
a value that defines the symbol used as a marker, or its size, weight, or color. By default, the value for a marker property is derived from a style element in the current style.

multi-cell layout
a layout that supports a rectangular grid of cells, each of which can contain a graphical element, such as a plot, a legend, a nested layout, and so on.

nested layout
a layout block that appears within the scope of another layout block.

ODS
See Output Delivery System.

ODS Graphics
an extension to ODS that is used to create analytical graphs using the Graph Template Language.

ODS Graphics Editor
an interactive application that can be used to edit and annotate ODS Graphics output.

ODS Template
a description of how output should appear when it is formatted. ODS templates are stored as compiled entries in a template store (item store). Common template types include STATGRAPH, STYLE, CROSSTABS, TAGSET, and TABLE.

opaque
a property of a background. Opaque backgrounds are filled with a color. Non-opaque backgrounds are transparent.

outlier
a data point that differs from the general trend of the data by more than is expected by chance alone. An outlier might be an erroneous data point or one that is not from the same sampling model as the rest of the data.

Output Delivery System
a component of SAS software that can produce output in a variety of formats such as markup languages (HTML, XML), PDF, listing, RTF, PostScript, and SAS data sets. Short form: ODS.

overlay
a plot that can be superimposed on another plot when specified within an overlay-type layout. A common overlay combination is a fit line on a scatter plot.

overlay layout
a type of layout that supports the superimposition of graphical components, such as plots, legends, and nested layouts.

padding
space added inside the border of a graphical component, such as a layout or a legend.

panel
a graph with multiple cells.

parameterized plot
a non-computed plot that requires parameterized data. The Graph Template Language offers several plots in both computed and parameterized versions, for example, BARCHART and BARCHARTPARM. Some computed plots such as REGRESSIONPLOT can be emulated with a SERIESPLOT if the input data represented points on a fit line.

parent block
when two or more blocks are nested, any layout block that contains one or more layout blocks is a parent of the contained blocks.

plot
a visual representation of data such as a scatter plot, a series line, or a histogram. In the ODS Graphics context, plot is a generic term for the graphical element or elements drawn by a plot statement. Multiple plots can be overlaid in a cell to create a graph.

plot area

the space, bounded by the axes, where a visual representation of data, such as a scatter plot, a series line, or a histogram, is drawn.

plot type

a plot family such as bar chart (which would include horizontal, vertical, and grouped bar charts), or a classification scheme for plots based on some useful criteria, such as whether the plots are computed or parameterized.

primary axis

the X or Y axis contrasted to the X2 or Y2 secondary axis.

primary plot

the plot in an overlay that determines axis features, such as axis type and axis label.

prototype layout

an overlay plot composite that appears in each cell of a classification panel. Each instance of the prototype represents a different subset (classification level) of the data.

regression plot

a straight or curved line showing a linear or higher order regression fit for a set of points.

required argument

a variable or constant that must be specified in order to evaluate an expression or render a plot, legend, text, or a layout. For example, a scatter plot has two required arguments: X=column and Y=column.

response variable

See dependent variable.

rich text

a generic term for text that can have different font characteristics (color, family, size, weight, style) on a character-by-character basis and can also be used as a superscript or subscript. All text statements in GTL support rich text.

role

a description of the purpose that a variable serves in a plot. For example, a series plot has predefined roles named for X , Y, GROUP, and CURVELABEL.

row

a set of layout cells that are side-by-side and share the same alignment.

row axis

an external axis appearing on the left or right of a row of cells in a multi-cell layout.

row gutter

space between rows of cells of a multi-cell layout.

row header

typically, the text that identifies the row contents in a multi-cell layout. This text can be aligned to the right or left of the cells in a row. The row header is not restricted to text (it can contain a plot or a legend, for example).

row major order

an order for populating cells of a layout or entries of a legend when the number of columns is specified. For example, in the default case: Start at the top left and fill cells or entries left-to-right. When the right-most column is filled, begin a new row below the previous row. Continue this until all content items have been placed in cells or entries. There might be empty cells/entries in the last row.

row weight
in a LATTICE layout, the proportion of height allotted to a specific row of the layout. The sum of all row weights is 1.

secondary axis
the X2 or Y2 axis as contrasted to the X or Y primary axis.

SGE file
a file created in the ODS Graphics environment that contains an editable graph. Such files have a .SGE file extension and can be edited only with the ODS Graphics Editor. You can edit SGE files from the SAS Results window or by opening the SGE file from within the ODS Graphics editor.

sidebar
an area of certain multi-cell layouts external to the grid of cells where text or other graphical elements can appear. The LATTICE, DATAPANEL, and DATALATTICE layout support four sidebar areas (TOP, BOTTOM, LEFT, and RIGHT).

single-cell layout
a layout type that supports only one cell. The OVERLAY, OVERLAY3D, and OVERLAYEQUATED layouts are examples of single-cell layouts.

sparse data
in classification panels with two or more classifiers, some crossings of the classification values might not be present in the input data. Such input data is called sparse data. By default, a DATAPANEL layout does not generate cells for sparse data, but if requested, it can produce empty cells as place holders for the non-existent crossings.

stand-alone plot
a plot that has its own data range and can therefore appear by itself in a layout.

statement block
a group of statements that has both a logical beginning and ending statement. For example, a LAYOUT statement along with its ENDLAYOUT statement and all contained statements are a block. Some blocks can be nested within other blocks.

style
an ODS template that can be used to control the visual aspects (colors, fonts, lines, markers, and so on) of a graph. A style consists of many style elements, and each style element consists of style attributes.

style attribute
a visual property such as a color, line pattern, or font property that has a reserved name. For example, COLOR, FONTFAMILY, FONTSIZE, FONTWEIGHT, and FONTSTYLE are all attributes of style elements such as GraphTitleText, GraphLabelText, and so on. Style attributes are collectively referenced by a style element within a style definition.

style element
a named collection of style attributes that affects specific parts of ODS output. For example, the GraphTitleText style element specifies the color and font properties of title text and possibly other text in the graph.

style reference
a part of the Graph Template Language syntax that indicates the current value of a specific attribute of a specific style element. For example, SIZE=GraphTitleText:FontSize means to assign to SIZE the value of the FontSize attribute of the GraphTitleText style element from the current style.

surface plot
a three-dimensional graph that displays values of a vertical Z variable based on gridded X and Y variables.

template compile time
the phase when the source program of a template definition is submitted. The syntax of the definition is evaluated for correctness. If no errors are detected, the definition is converted to a binary format and stored for later access.

template definition
the TEMPLATE procedure source program that creates a template. A template definition can be generated from a compiled template. Also called the template source.

template run time
the actions performed when a compiled template is bound to a data object and then rendered to produce a graph. Run-time errors can occur that prevent a graph from being produced.

template source
See template definition.

template store
an item store that contains definitions that were created by the TEMPLATE procedure. Definitions that SAS provides are in the item store Sashelp.Tmplmst. You can store definitions that you create in any template store to which you have write access.

template-based graphic
SAS/GRAPH output where a compiled ODS template of the type STATGRAPH is used to produce graphical output, that is, a graph produced within the ODS graphics environment as opposed to a graph produced in the traditional device-based environment.

text properties
a common set of characteristics that can be specified for any text string: COLOR, FAMILY, SIZE, WEIGHT, and STYLE. By default, values for these properties are derived from a style element in the current style.

time axis
an axis type that displays only SAS date, time, or datetime values. Axis tick value increments can be specified as time or date intervals, such as MINUTE, HOUR, DAY, WEEK, MONTH, QUARTER, or YEAR.

title area
the region above the graph area where text produced by ENTRYTITLE statements appears.

transparency
the degree to which a graphic element (such as a marker or filled area) is opaque or transparent. Transparency is indicated with a number from 0 (completely opaque) to 1 (completely transparent).

Unicode

an encoding system that provides a single comprehensive mapping of all characters (glyphs) in all languages to unique numeric values called code points.

viewport

See axis viewport.

wall

the area bounded by orthogonal axis pairs. In two-dimensional graphs, there is one wall bounded by the XY axes. In three-dimensional graphs, there are three walls, bounded by the XY, YZ, and XZ axes. A wall has an optional outline and can be opaque or transparent.

weight variable

a positive numeric variable in the input data set that represents a weight to be applied to the current observation.

Index

Your Turn

We welcome your feedback.

- If you have comments about this book, please send them to **`yourturn@sas.com`**. Include the full title and page numbers (if applicable).
- If you have comments about the software, please send them to **`suggest@sas.com`**.