THE POWER TO KNOW.

SAS® 9.3 Graph Template Language Reference

Third Edition

SAS® Documentation

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2012. *SAS® 9.3 Graph Template Language: Reference, Third Edition*. Cary, NC: SAS Institute Inc.

SAS® 9.3 Graph Template Language: Reference, Third Edition

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

1st electronic book, December 2012
2nd electronic book, May 2013

SAS® Publishing provides a complete selection of books and electronic products to help customers use SAS software to its fullest potential. For more information about our e-books, e-learning products, CDs, and hard-copy books, visit the SAS Publishing Web site at **support.sas.com/publishing** or call 1-800-727-3228.

Contents

What's New in the SAS 9.3 Graph Template Language

Overview

New and enhanced statements for Graph Template Language (GTL) extend the versatility of the language and introduce a number of new plot types. The changes include the following:

- new layout statements
- new plot statements
- new legend statements
- new features for general use
- enhancements to SAS 9.2 statements

New Layout Statements

The following layout statements are new:

- LAYOUT REGION provides a container for plots that do not use axes (for example, a pie chart).
- LAYOUT GLOBALLEGEND creates a compound legend containing multiple discrete legends.

New Plot Statements

The following plot statements are new:

- BUBBLEPLOT creates a bubble plot of the input data, using X and Y columns to locate bubble centers and a SIZE column to control the bubble radius.
- DENDROGRAM creates a tree diagram that is typically used to display the results of a hierarchical clustering analysis.
- HEATMAPPARM creates a two-dimensional plot that represents the values of pre-binned, three-dimensional data.

- HIGHLOWPLOT creates floating vertical or horizontal lines or bars that connect the minimum and maximum response values for each value of a categorical variable. The vertical version is typically used in the financial industry to plot stock values over time. The horizontal version is typically used in the Health and Life Sciences industry to display over time the duration of adverse events or of adverse reactions to medication.
- PIECHART creates a pie chart that is computed from input data.
- WATERFALLCHART creates a waterfall chart that is computed from input data. A waterfall chart is typically used to show credit and debit transactions or successive changes to a given state.

In the second maintenance release of SAS 9.3, the MOSAICPLOTPARM statement is new. It creates a mosaic plot from pre-summarized categorical data.

New Legend Statements

The following legend statements are new:

- LEGENDITEM creates the definition for a legend item that can be included in a legend. The item is independent of the data and enables you to customize the legend to enhance or replace a standard legend.
- MERGEDLEGEND consolidates legend entries when a graph displays grouped data for two plots.
- For discrete axes that might appear crowded with too many tick values, the new AXISLEGEND statement can clean up the clutter. AXISLEGEND generates consecutive integers for display as axis-tick values in the graph. It also creates a legend that correlates those integers with the actual tick values that they represent.

New Features for General Use

Attribute Maps

The GTL now provides discrete and range “attribute maps” that enable you to map visual attributes to input data values.

- Discrete attribute maps enable you to map discrete data to specific visual attributes (for example, color). This feature could be used to represent in the legend a data value with a specific visual attribute, regardless of the order of the values in the data or whether the value exists in the data. For example, when using gender as a grouping category for plots, you could set an attribute map for marker symbols. You could define the map so that all plots that are generated by a template use red circles to represent the values for females and blue diamonds to represent the values for males. Relevant statements: DISCRETEATTRMAP and DISCRETEATTRVAR.
- Range attribute maps enable you to control color assignment, regardless of the current data range for a plot. This feature could be used to make the color assignments independent of the actual data range for consistency. For example, for

temperature data, you could set blue for 0 and red for 100, even if the values 0 and 100 are not in the data. Relevant statements: RANGEATTRMAP and RANGEATTRVAR.

Draw Statements

A set of new draw statements enables you to draw lines, arrows, ovals, rectangles, and other shapes in your graph. You can also draw text and images. A global drawing space and drawing units for all of the draw statements is set in the BEGINGRAPH statement's new DRAWSPACE= option. The individual draw statements provide options that enable you to set the drawing space and drawing units for that individual statement. The following draw statements are available:

- BEGINPOLYGON
- BEGINPOLYLINE
- DRAWARROW
- DRAWIMAGE
- DRAWLINE
- DRAWOVAL
- DRAWRECTANGLE
- DRAWTEXT

In the second maintenance release of SAS 9.3:

- DRAWRECTANGLE:
  - CORNERRADIUS= specifies the radius of the rectangle corners.

General Enhancements Supported by Many of the Plots

The following new features that are supported by many of the plot statements are worth highlighting. The individual plot statements that support these features are identified in "Plot Enhancements" on page xv.

- For plots that support a grouping variable:
  - A grouping variable can now display groups in clusters. To do so, use GROUP= (new for some of the plots) to specify the grouping variable and set the new GROUPDISPLAY= option to CLUSTER. The new INCLUDEMISSINGGROUP= option specifies whether missing values of the group variable are included in the plot. GROUPORDER= (new for some of the plots) specifies the order of the grouped plot elements (for example, bars) for each category value. The new CLUSTERWIDTH= option specifies the width of the group clusters as a fraction of the midpoint spacing on a discrete axis or a fraction of the minimum interval between adjacent data values on an interval axis.
  - In the first maintenance release of SAS 9.3:
    - New fill colors are generated automatically when the number of group values exceeds the number of GraphData1–GraphDataN style elements. The new fill colors are generated by repeating the GraphData1–GraphDataN colors and varying the shade of each original color on each cycle. The shade variations alternate between one shade lighter and one shade darker on each cycle.

    - The group value style attributes are now independent of group order and are always assigned in data order.
    - The attributes of the missing group value are now determined by the GraphMissing style element except when the MISSING= system option is used to specify a missing character other than the default, or a user-defined format is applied to the missing group value. In those cases, the attributes of the missing group value are determined by a GraphData1–GraphDataN style element instead of the GraphMissing style element.
- Data skins, which can be applied with the new DATASKIN= option, enhance the visual appearance of a graph. For example, a data skin can be used to apply a glossy, three-dimensional appearance to the filled bars of a bar chart or the filled slices of a pie chart. Data skins are available only for bars, pie slices, bubbles, and scatter plot markers.
- Transparency for the filled elements of some graph types can now be managed independently of the other transparent elements in the graph. For example, one transparency level can be set for the filled bars of a bar chart and a different transparency level can be set for the bar outlines. As in SAS 9.2, DATATRANSPARENCY= sets the transparency level for fills and lines. A different transparency level can now be set for area fills, using the new fill option TRANSPARENCY=. The new fill option is available in any option that sets area fills (for example, a FILLATTRS= option).

General Enhancements to the Output Delivery System

The following enhancements have been made to the Output Delivery System (ODS):

- When running SAS in windowing mode for the Windows and UNIX operating environments, the default output destination has changed from LISTING to HTML.
- When running SAS in windowing mode for the Windows and UNIX operating environments and directing output to the HTML destination, the default style has changed from DEFAULT to HTMLBLUE. The new all-color HTMLBLUE style is ideal for ODS graphics because it provides superior color coordination between graphs and tables, using colors to differentiate between groups.
- The LISTING, HTML, and PRINTER destinations now support Scalable Vector Graphics (SVG).

For more information about these and other ODS enhancements, consult *SAS Language Reference: Concepts* and *SAS Output Delivery System: User's Guide*.

Enhancements to SAS 9.2 Statements

Layout Enhancements

LAYOUT DATALATTICE and LAYOUT DATAPANEL:

- These lattice-type layouts now support independent X2 (top) and Y2 (right) axes. In SAS 9.2, the X2 axis could mirror only the X axis, and the Y2 axis could mirror only the Y axis.

- INCLUDEMISSINGCLASS= specifies whether to include grid cells for crossings of the class variables that contain a missing value.
- INSETOPTS= now provides TITLE= for setting an inset title and TITLEATTRS= for setting the text attributes of that title.
- SHRINKFONTS= specifies whether fonts in the multi-cell layout are scaled when appropriate, depending on the nesting levels of the layouts that have SHRINKFONTS=TRUE.
- SPACEFILL= (in the SIDEBAR statement) specifies whether the sidebar's contents should expand to fill the entire sidebar area, up to its boundaries.

LAYOUT LATTICE:

- The LATTICE layout now supports independent X2 (top) and Y2 (right) axes. In SAS 9.2, the X2 axis could mirror only the X axis, and the Y2 axis could mirror only the Y axis.
- SHRINKFONTS= specifies whether fonts in a multi-cell layout are scaled when appropriate, depending on the nesting levels of the layouts that have SHRINKFONTS=TRUE.
- SPACEFILL= (in the SIDEBAR statement) specifies whether the sidebar's contents should expand to fill the entire sidebar area, up to its boundaries.

LAYOUT GRIDDED:

- SHRINKFONTS= specifies whether fonts in a multi-cell layout are scaled when appropriate, depending on the nesting levels of the layouts that have SHRINKFONTS=TRUE.

LAYOUT OVERLAY:

- The INNERMARGIN statement generates one or more "inner margins" within a LAYOUT OVERLAY container. An inner margin is a nested region at the top or bottom of the OVERLAY container.
- ASPECTRATIO= specifies the aspect ratio of the rectangle that forms the bounds of the plot wall.

LAYOUT PROTOTYPE:

- ASPECTRATIO= specifies the aspect ratio of the plot wall area.

Plot Enhancements

BANDPLOT:

- INCLUDEMISSINGGROUP= specifies whether missing values of the group variable are included in the plot.
- Transparency can be managed separately for the filled bands and the band outline. As in SAS 9.2, DATATRANSPARENCY= sets the transparency level for both the fills and the lines. The new FILLATTRS=(TRANSPARENCY=*number*) can set a different transparency level for just the band fill.

BARCHART and BARCHARTPARM:

- Both statements support the general plot enhancements that were listed previously. For more information, see "General Enhancements Supported by Many of the Plots" on page xiii.

- FILLPATTERNATTRS= specifies the attributes for fill patterns, which are useful for displaying monochrome graphs (typically the black and white graphs that are printed in journal articles).
- BASELINEINTERCEPT= specifies the response axis intercept for the baseline.
- TARGET= specifies a numeric column that provides target values to display on the chart bars as a small triangle with a line extending from it across the bar.

BLOCKPLOT:

- BLOCKPLOT supports the general plot enhancements (excluding data skins) that were listed previously. For more information, see "General Enhancements Supported by Many of the Plots" on page xiii.
- EXTENDBLOCKONMISSING= specifies whether a missing value in the BLOCK column starts a new block or reverts to the previous nonmissing value.
- INCLUDEMISSINGCLASS= specifies whether missing values of the class variable are included in the plot.

BOXPLOT and BOXPLOTPARM:

- Both statements now support the GROUP= option and clustered groups. For more information, see "General Enhancements Supported by Many of the Plots" on page xiii. For overriding default grouping, the new INDEX= option can specify indices for mapping line and fill attributes (color and pattern) to one of the GraphData1–GraphDataN style elements.
- Box plots now support an independent, numeric axis. By default, box plot requests a discrete category axis. If your data corresponding to the independent axis is numeric (or time), you can set TYPE= to LINEAR, TIME, or LOG in the layout's axis options. With a numeric axis in effect, you can also use the new INTERVALBOXWIDTH= option to specify the box width.
- Box plots now support tooltips. The TIP= and OUTLIERTIP= options can be used to set the tips for the boxes and the outliers. The TIPFORMAT = and TIPLABEL= options can be used to format and label the tips. BOXPLOTPARM also supports the ROLENAME= option (not available for BOXPLOT), which specifies user-defined roles that can be used to display additional information in the tooltips.
- BOXPLOTPARM supports the DISPLAYSTATS= options (not available for BOXPLOT and horizontal BOXPLOTPARM), which specifies the statistics to be displayed for each box.
- In the second maintenance release for SAS 9.3, BOXPLOT supports the DISPLAYSTATS= option (not available for horizontal BOXPLOT), which specifies the statistics to be displayed for each box.

BUBBLEPLOT:

- In the second maintenance release for SAS 9.3:
  - RELATIVESCALETYPE= specifies the type of scaling that is to be applied to the SIZE= column values.
  - SIZETHRESHOLDMAX= specifies a SIZE= column value threshold at which bubble size is clamped to the BUBBLERADIUSMAX= option value.

ELLIPSE and ELLIPSEPARM:

- Transparency can be managed separately for the filled ellipse and the ellipse outline. DATATRANSPARENCY= sets the transparency level for both the fill and the line. FILLATTRS=(TRANSPARENCY=*number*) can set a different transparency level for just the fill.

- INCLUDEMISSINGGROUP= specifies whether missing values of the group variable are included in the plot. (This option is available for ELLIPSEPARM but not for ELLIPSE.)

FRINGEPLOT, LINEPARM, LOESSPLOT, PBSPLINEPLOT, and REGRESSIONPLOT:

- Each of these statements has a new INCLUDEMISSINGGROUP= options, which specifies whether missing values of the group variable are included in the plot.

HIGHLOWPLOT:

- In the second maintenance release for SAS 9.3:
  - ENDCAPDISPLAYPOLICY= specifies whether to display outlined colored regions or just colored regions.

HISTOGRAMPARM:

- In the second maintenance release for SAS 9.3:
  - DATALABEL= specifies a column for the bar labels.
  - DATALABELATTRS= specifies the color and font attributes of the bar labels.

NEEDLEPLOT:

- NEEDLEPLOT supports clustered groups. For more information, see "General Enhancements Supported by Many of the Plots" on page xiii. It does not support the data skins or dual transparency levels.
- DATALABEPOSITION= specifies the location of the data labels relative to the needle lines and markers.
- DISCRETEOFFSET= specifies an amount to offset all needle lines and markers from discrete X values when graphing multiple response variables side by side on a common axis.

PIECHART:

- In the first maintenance release for SAS 9.3:
  - The STAT= option value PERCENT is changed to PCT in order to make it consistent with other plots.
  - The DATALABELCONTENT= option default is changed in order to reduce the information that is displayed by default and to customize the information for the STAT= option value. The new default depends on the STAT= option value as follows:
    - For STAT=PCT, the DATALABELCONTENT= default is (CATEGORY PERCENT).
    - For all other values of the STAT= option, the DATALABELCONTENT= default is STANDARD.

SCATTERPLOT:

- SCATTERPLOT supports clustered groups and data skins. For more information, see "General Enhancements Supported by Many of the Plots" on page xiii.
- DATALABEPOSITION= specifies the location of the data labels relative to the markers.
- USEDISCRETESIZE= specifies that the marker size should be based on fraction of the midpoint spacing. DISCRETEMARKERSIZE= specifies the fractional size to use for that marker size.

- MARKERSIZERESPONSE= specifies a column to use for varying marker sizes with response values. MARKERSIZEMAX= and MARKERSIZEMIN= enable you to manage the range of the marker-size variation when MARKERSIZERESPONSE= is used.

SCATTERPLOTMATRIX:

- DATALABEPOSITION= specifies the location of the data labels relative to the markers.
- INCLUDEMISSINGGROUP= specifies whether missing values of the group variable are included in the plot.
- INSETOPTS= now provides TITLE= for setting an inset title and TITLEATTRS= for setting the text attributes of that title.

SERIESPLOT:

- SERIESPLOT supports clustered groups. For more information, see "General Enhancements Supported by Many of the Plots" on page xiii.
- DATALABEPOSITION= specifies the location of the data labels relative to the series line and markers.
- SMOOTHCONNECT= specifies that the plot vertices be connected by a smooth line.

STEPPLOT:

- STEPPLOT supports clustered groups. For more information, see "General Enhancements Supported by Many of the Plots" on page xiii.
- DATALABEPOSITION= specifies the location of the data labels relative to the series line and markers.

VECTORPLOT:

- DATALABEPOSITION= specifies the location of the data labels relative to the vector lines and arrow heads.
- INCLUDEMISSINGGROUP= specifies whether missing values of the group variable are included in the plot.

Axis Enhancements

LAYOUT OVERLAY:

- NAME= assigns a name to an axis for reference in other statements.
- DISCRETEOPTS= supports the following new features for discrete axes:
  - COLORBANDS= specifies the display of alternating wall-color bands corresponding to the discrete axis bins. COLORBANDSATTRS= specifies the appearance of the alternating wall-color band.
  - TICKTYPE= specifies the position of the axis tick mark.
  - TICKVALUEFITPOLICY= specifies a policy for avoiding tick value collision on an axis.

LAYOUT OVERLAYEQUATED:

- In the second maintenance release of SAS 9.3:
  - REVERSE= specifies whether the tick values on the X or Y axis should appear in the reverse order.

LAYOUT LATTICE:

- NAME= assigns a name to an axis for reference in an AXISLEGEND statement.
- REVERSE= specifies whether the axis origin should be reversed.
- DISCRETEOPTS= supports the following new features for discrete axes:
  - TICKTYPE= specifies the position of the axis tick mark.
  - TICKVALUEFITPOLICY= specifies a policy for avoiding tick value collision on an axis.

Attribute Map Enhancements

RANGEATTRMAP:

- In the first maintenance release of SAS 9.3, for the RANGE statement:
  - RANGEALTCOLOR= specifies a single contrast color to represent the defined value range.
  - RANGEALTCOLORMODEL= specifies either a style element or a list of one or more specific contrast colors to represent the defined value range.

Deprecated Statement Options

In the first maintenance release of SAS 9.3:

- The DRAWTEXT statement BACKGROUNDATTRS= option is deprecated.
- The LAYOUT GLOBALLEGEND statement BACKGROUNDATTRS= option is deprecated.

Part 1

Fundamentals

Chapter 1

Overview

Graph Template Language (GTL)

GTL and the Output Delivery System (ODS)

The SAS *Graph Template Language* (GTL) is an extension to the Output Delivery System (ODS) that enables you to create sophisticated graphics. 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. It also includes 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.

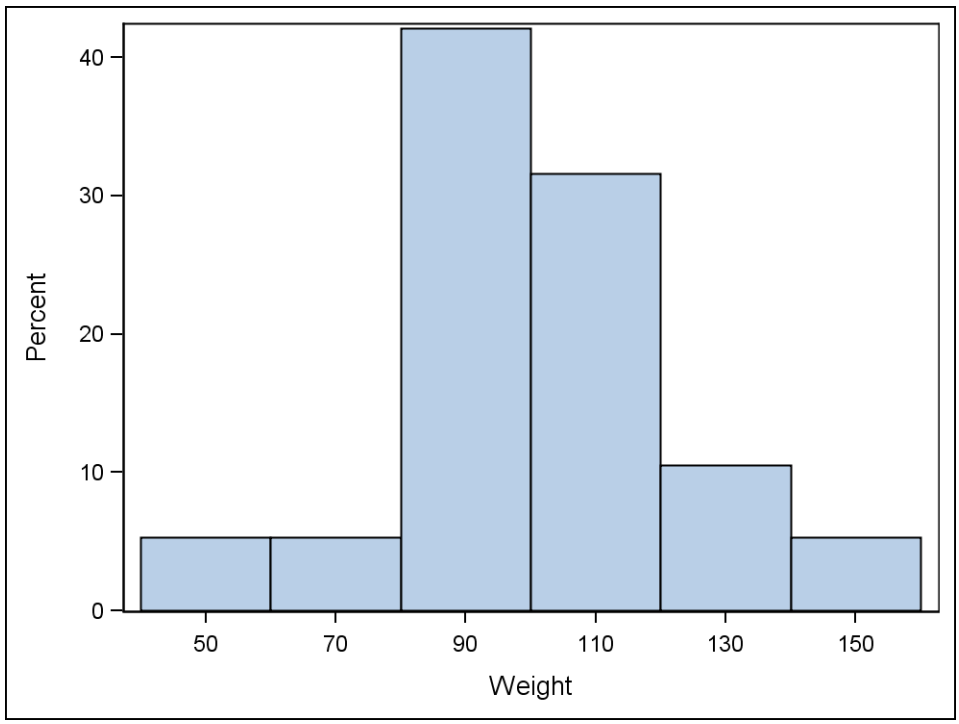
- The GTL templates are rendered using the SGRENDER procedure, which specifies a data source that contains appropriate data values and the template to use for rendering the graph.
- You can also modify predefined GTL templates that the SAS System delivers for use on the SAS statistical procedures. For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

Note to SAS/GRAPH Users. If you are also a SAS/GRAPH user, you might want to consult the *SAS Graph Template Language: User's Guide* to learn about some of the distinctions between ODS Graphics and SAS/GRAPH.

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 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;
```

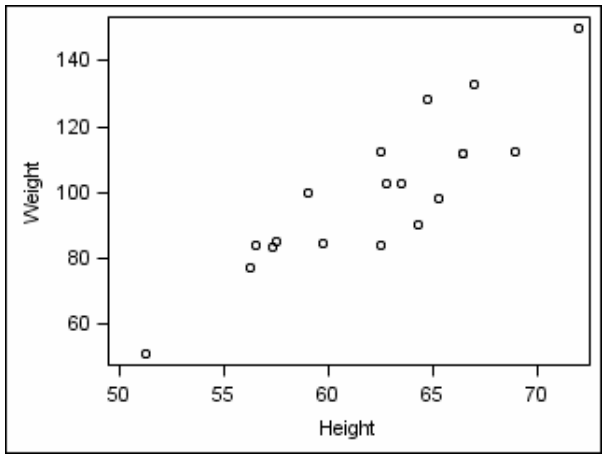
When this code is submitted, the statement keywords and options are parsed, just as with any other procedure. If no syntax error is detected, an output template named SCATTER is created and stored in the default template folder SASUSER.TEMPLAT. No graph is produced. Note the following:

- 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;
```



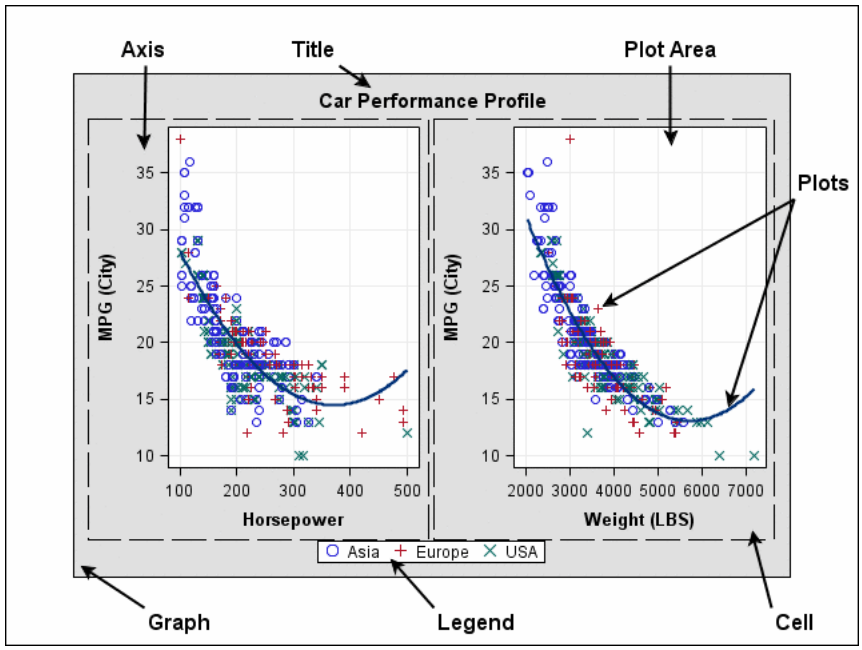
Generally, an ODS data object is constructed by comparing the template references to column names with variables that exist in the current data set. In the current example, SASHELP.CLASS contains variables named HEIGHT and WEIGHT. Because these variable names match the variables that are named on template 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

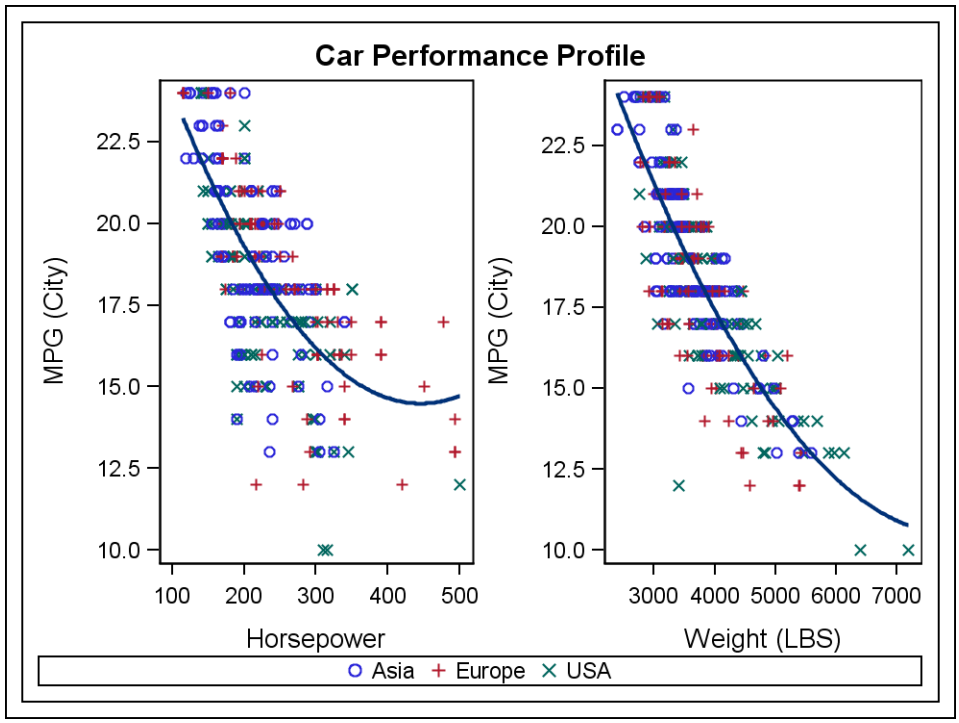
One of most powerful features of the GTL is the syntax built around hierarchical statement blocks called “layouts.” The outermost layout block determines

- 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.
REGION	General purpose layout for displaying single-cell graphs that does not use 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.
GLOBALLEGEND	Specialized layout for creating a compound legend that contains multiple discrete legends.

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 on. The main distinction between such plots is the nature of the input data that they accept:

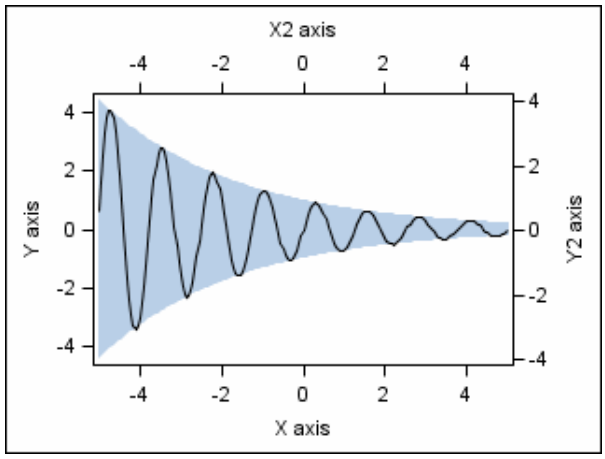
- The “non-parm” version (for example, BARCHART) computes its values from raw, unsummarized data. For example, a BARCHART computes the summary values it needs for the bars in the chart. Such plots are often referred to as “computed plots.”
- The “parm” version (for example, BARCHARTPARM) does not summarize 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 14, “Key Concepts for Using Plots,” on page 137 discusses general concepts that apply across plot types. For detailed information about a particular plot, see the chapter for that plot.

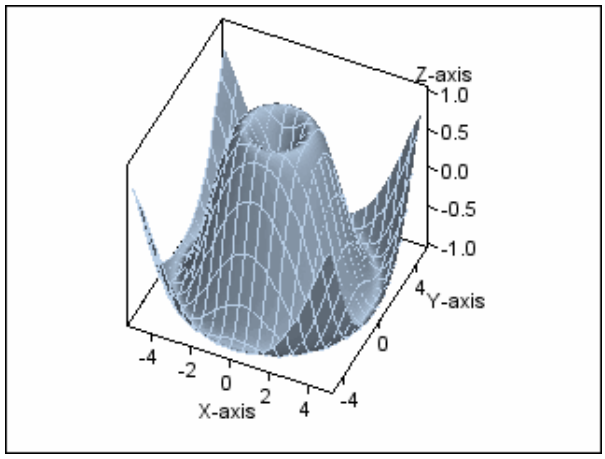
Axes

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 50, "Axis Features in Layouts," on page 557.

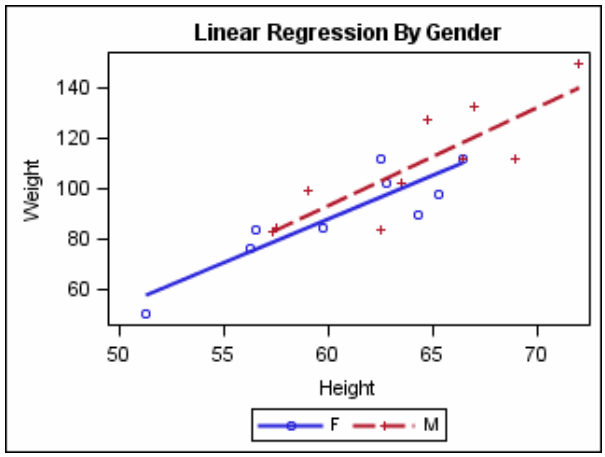
Legends

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, MERGEDLEGEND).

```
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";
     mergedlegend "scat" "reg" / border=true;
   endlayout;
  endgraph;
 end;
run;
```



For more information about managing legends in GTL, see *SAS 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 75, “Expressions,” on page 831. For more information about using functions, see Chapter 76, “Functions,” on page 835.

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:

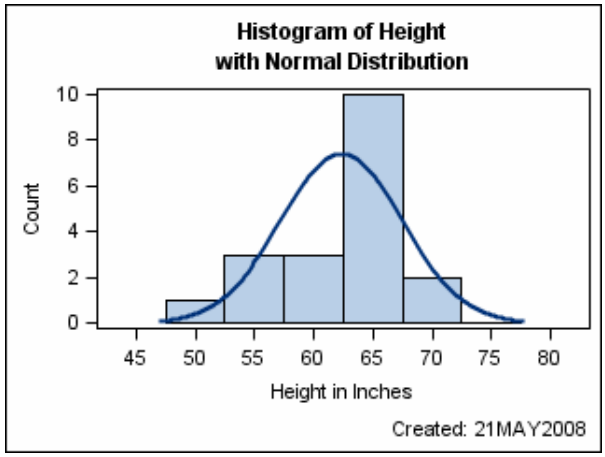
Template Statement	Purpose	Value supplied by...
DYNAMIC	defines one or more dynamic variables	either of the following: • DYNAMIC= suboption of ODS= option of FILE PRINT • DYNAMIC statement of PROC SGRENDER
MVAR	defines one or more macro variables	%LET or CALL SYMPUT( )
NMVAR	defines one or more macro variables that resolve to a number or numbers	%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 74, "Dynamics and Macro Variables," on page 827.

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;
```

In 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 size is adjusted to fill the remaining space.

For more information about using conditional logic, see Chapter 77, "Conditional Logic," on page 839.

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 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;
```

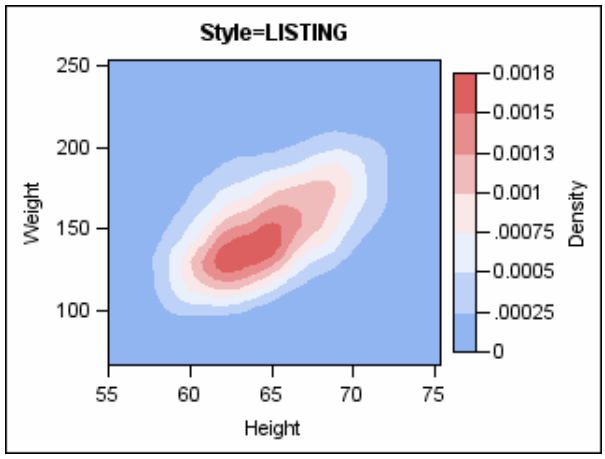
Support for ODS styles is highly integrated into GTL syntax. By default, the graphical appearance features of most plot and text statements are mapped to corresponding style elements and associated attributes. Because of this, your output tables and graphs always have a reasonable overall appearance.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

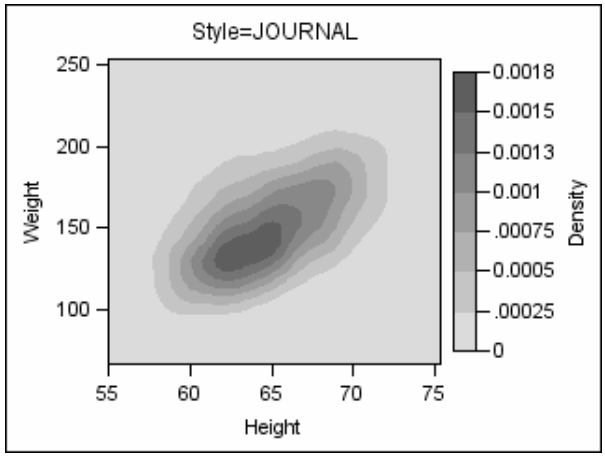
The following figures show how a graph’s appearance can be changed by using references to style elements to set the graph’s appearance options. This technique permits changes in graph appearance by style modification instead of graphical template modification. The graphs in the figures are generated 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 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. We used the LISTING destination for all of the graphs. If you generate the example graphs in the LISTING destination, they will typically be rendered in 640 pixel by 480 pixel size. Some of the examples show you how to change the graph size, and of course, those graphs will be rendered in the specified 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 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

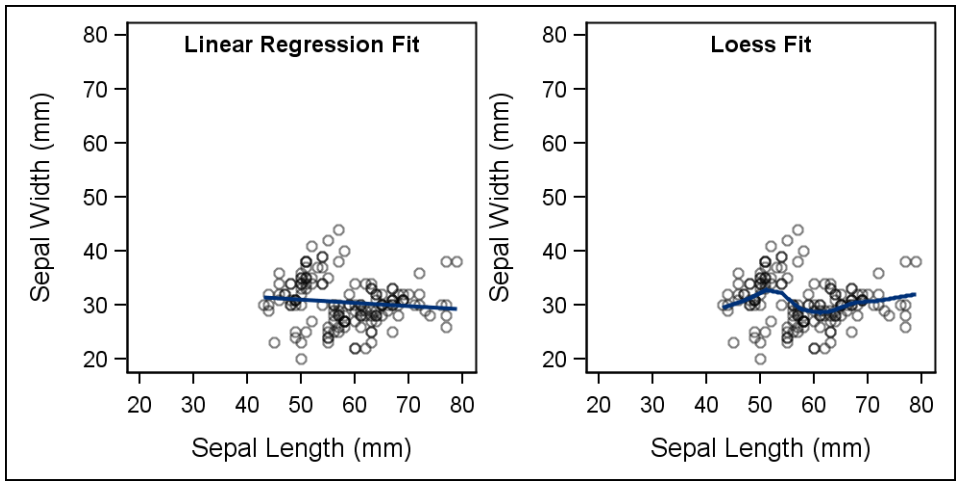
Defines the outermost container for a graph template that is defined with *GTL-statements*.

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 of its purposes is to support options that apply to the entire graph. For example, the default graph size that a template produces is typically 640x480 pixels. If you need a different size, you can declare the alternative size on this statement. 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 textattrs=(weight=bold);
        scatterplot x=XVAR y=YVAR / datatransparency=.5;
        regressionplot x=XVAR y=YVAR;
      endlayout;
      layout overlayequated / equatetype=square;
        entry "Loess Fit" /
          valign=top textattrs=(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, one and only one GTL layout block is required. It can be a LATTICE, GRIDDED, OVERLAY, OVERLAYEQUATED, OVERLAY3D, REGION, DATALATTICE, or DATAPANEL layout block. This layout block can contain other nested layout blocks and should contain at least one plot statement.

The GTL global statements apply to the entire and can include such statements as title and footnote statements, attribute maps, draw statements, conditional statements, and so on. 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 output size for a single graph, use the DESIGNWIDTH= and DESIGNHEIGHT= options in the BEGINGRAPH statement for that graph. 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. In this instance, the setting renders each graph cell as a 320px by 320px square. (The cells are square in this case, but the resulting cell size depends on the graph definition and would not be the same for all graphs.)

Note: To change the graph sizes for all templates in the current SAS session, you can 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 in another ODS GRAPHICS statement. You can also use WIDTH= and HEIGHT= settings in the graph style to modify the graphs sizes across template definitions. Be aware, however, that if you explicitly manage the graph output size, the graph elements might be scaled so that the size specification is honored.

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:

Note: A "square graph" means that the output graph's width and height are equal. That does not imply that the X and Y axis lengths are equal if the graph contains only one cell.

```
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 render width or render 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.
DRAWSPACE	Specifies a global drawing space and drawing units for all of the draw statements within this BEGINGRAPH block.
INCLUDEMISSINGDISCRETE	Specifies whether missing values are displayed on a discrete axis.
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: The ODS GRAPHICS statement BORDER= option setting, which is TRUE by default.

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 849 for the syntax on using a *style-element* and "Line Options" on page 850 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.

Restriction: The minimum *dimension* value that you can set is 2 pixels. If a smaller setting is specified, the default design height is used.

Interaction: This height can be overridden at run time with a render height that is specified with the HEIGHT= option in the ODS GRAPHICS statement (external to the template). Also, the ODS destination statement's IMAGE_DPI= option can affect the height.

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. Also, changing the height setting in the graph style would affect the height of all templates that use that style.

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.

Restriction: The minimum *dimension* value that you can set is 2 pixels. If a smaller setting is specified, the default design width is used.

Interaction: This width can be overridden at run time with a render width that is specified with the WIDTH= option in the ODS GRAPHICS statement (external to the template). Also, the ODS destination statement's IMAGE_DPI= option can affect the width.

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. Also, changing the width setting in the graph style would affect the width of all templates that use that style.

DRAWSPACE= GRAPHPERCENT | GRAPHPIXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALUE
specifies a global drawing space and drawing units for all of the draw statements within this BEGINGRAPH block.

Default: LAYOUTPERCENT

Tip: Individual draw statements within this BEGINGRAPH block can override this global setting.

See Also: “About the Drawing Space and Drawing Units” on page 736 .

INCLUDEMISSINGDISCRETE=*boolean*
specifies whether missing values are displayed on a discrete axis.

Default: FALSE

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

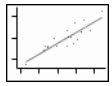
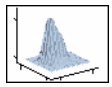
Summary of Layout Statements

Overview

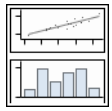
Layout blocks always begin with the LAYOUT keyword followed by a keyword indicating the purpose of the layout. All layout blocks end with an ENDLAYOUT statement.

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

Single-cell 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	
REGION (Single Cell)	2-D (1 cell)	General purpose layout for displaying single-cell graphs that does not use axes	
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.	

Legend Layout

Layout	Cells Produced	Comments
GLOBALLEGEND	1 cell for a legend	Specialized layout for creating a compound legend that contains multiple discrete legends.

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;

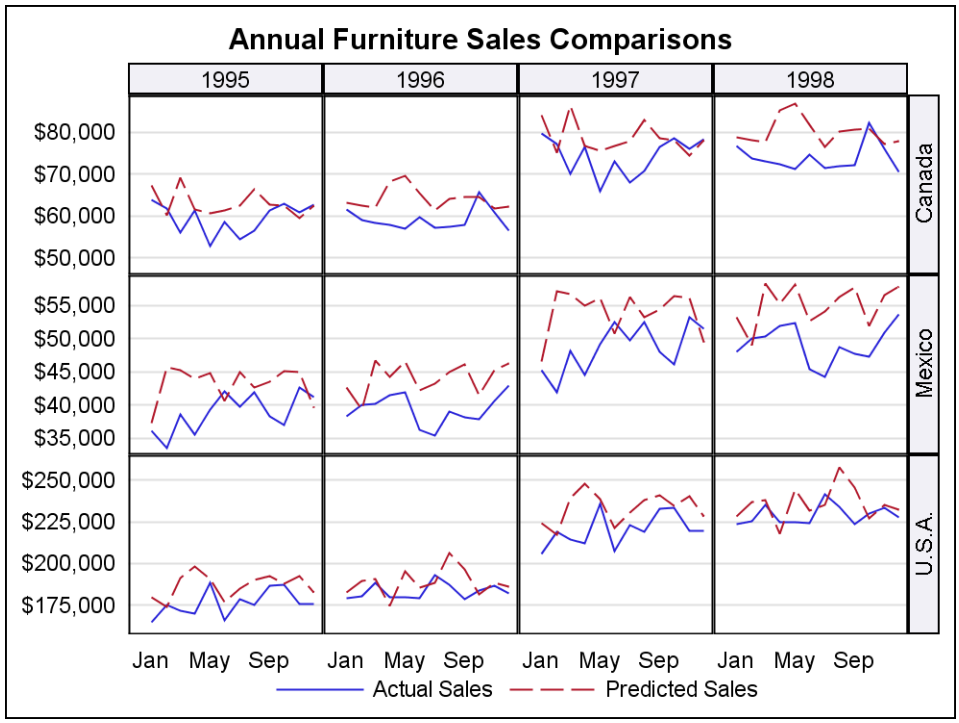
<*sidebar-statement-block(s)*;>

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

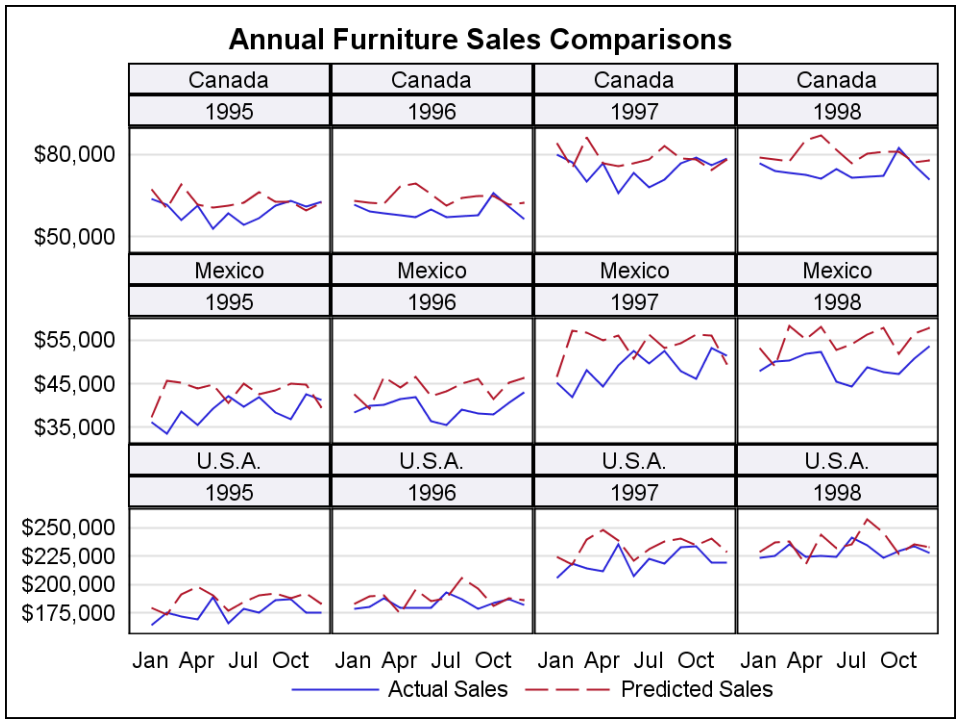
The LAYOUT DATALATTICE statement makes it easy to create a grid of graphs, based on the values of one or two classifications variables. To create a grid that is based on more than two classification variables, or to have more control over the grid layout, use LAYOUT DATAPANEL instead.

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 or 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.

If the class variable is of type character, its values are returned in data order. To control the ordering of the values, you can sort the input data by the classification variables. If the class variable is of type numeric, the values are displayed in ordinal order.

Formats can be assigned to class variables to create classification levels (for example, an AGEGROUPFMT. format for numeric AGE). In this case, the classification is performed after the format is applied. For numeric data, the order is ordinal, based on the first value in each class.

Use the INCLUDEMISSINGCLASS option to control whether cells are displayed when any value crossing contains a missing value.

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 that 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;

Note: You can specify only one LAYOUT PROTOTYPE block in the LAYOUT DATALATTICE block. If you specify more than one, only the last prototype block specified is honored. The remaining prototype blocks are ignored.

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 125.

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 </ *option(s)*>;

GTL-statement(s);

ENDSIDEBAR;

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;
```

SIDEBAR Options

ALIGN=BOTTOM | TOP | LEFT | RIGHT
specifies the sidebar's location within the layout.

Default: BOTTOM

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.

SPACEFILL=*boolean*
specifies whether to fill all the area of the sidebar with its contents.

Default: TRUE

Tip: To prevent a layout block within the sidebar from expanding to the sidebar boundaries, set this option to FALSE.

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 X-axis options for all columns.
COLUMN2AXISOPTS	Specifies X2-axis options for all columns.
COLUMNDATARANGE	Specifies how the X-axes of instances of the *graph-prototype* are scaled.
COLUMN2DATARANGE	Specifies how the X2-axes of instances of the *graph-prototype* are scaled.

Statement Option	Description
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.
INCLUDEMISSINGCLASS	Specifies whether to include grid cells for crossings of the class variables that contain a missing value.
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.
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 Y-axis options for all rows.
ROW2AXISOPTS	Specifies Y2-axis options for all rows.
ROWDATARANGE	Specifies how the Y-axes of instances of the *graph-prototype* are scaled.
ROW2DATARANGE	Specifies how the Y2-axes of instances of the *graph-prototype* are scaled.

Statement Option	Description
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.
SHRINKFONTS	Specifies whether fonts in the layout are scaled, depending on the nesting levels of the layouts that are used.
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 in order 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 849 for the syntax on using a *style-element* and "Line Options" on page 850 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.

COLUMN2AXISOPTS=(*axis-options*)
specifies X2-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.

Details: This option is needed only if you use a plot statement that supports a secondary X2 axis. If you do not use that statement's XAXIS= option to map data to the X2 axis, this option is ignored. For more information about how data are mapped to the axes, see "Plot Data Are Mapped to a Designated Axis" on page 558

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.

Tip: Use the COLUMNAXISOPTS= option to control shared axis features.

See Also: The PANELNUMBER= option for a discussion of how to create multiple panels.

COLUMN2DATARANGE=UNIONALL | UNION
specifies how the X2-axes of instances of the *graph-prototype* are scaled.

Default: UNIONALL

UNIONALL
scales the X2-axis data ranges across all layout columns and panels (when PANELNUMBER= is in effect).

UNION
scales the X2-axis data ranges separately for each column on a per-panel basis. The scaling does not span across multiple panels.

Tip: Use the COLUMN2AXISOPTS= option to control shared axis features.

Details: This option is needed only if you use a plot statement that supports a secondary X2 axis. If you do not use that statement's XAXIS= option to map data to the X2 axis, this option is ignored. For more information about how data are mapped to the axes, see "Plot Data Are Mapped to a Designated Axis" on page 558

See Also: 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 in order 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 849 for the syntax on using a *style-element* and "Text Options" on page 852 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.

INCLUDEMISSINGCLASS=*boolean*
specifies whether to include grid cells for crossings of the ROWVAR and COLUMNVAR variables that contain a missing value.

Default: TRUE. Any crossing of the class variables that includes a missing value produces a row or column of cells in the grid.

Tip: If this option is set to FALSE, any crossing of the class variables that contains a missing value does not produce a row or columns of cells in the grid.

Discussion: By default, missing class values are included in the classification levels for the panel. When the data contains missing classification values, cells are created in the panel for the missing classes. The classification headers for the missing values are either blank for missing string values or a dot for missing numeric values. You can use the INCLUDEMISSINGCLASS=FALSE option to exclude the missing values. If you want to keep the missing values, you can create a format that specifies more meaningful headings for the missing classes. For example, here is a format that specifies descriptive headers for missing product name and branch number classes.

```
proc format;
  value $prdfmt    " "="Missing Product";
```

```
  value branchfmt .="Missing Branch";
run;
```

This format can be applied to the classification columns in the PROC SGRENDER statement. In that case, if the classification columns contain missing values, the labels specified in the format statement are used as the headings for the missing classes.

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 *SAS 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 the population order of the classification panel's cells, 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, excluding the title. See "General Syntax for Attribute Options" on page 849 for the syntax on using a *style-element* and "Text Options" on page 852 for available *text-options*.

Default: The GraphDataText style element.

TITLE= "*string*"
specifies a title for the inset. The title is added at the top of the inset and spans the full inset width.

Default: no default, and space is not reserved for the title when it is not set

Tip: Text properties for the title string can be set with TITLEATTRS=.

TITLEATTRS=*style-element* | *style-element* (*text-options*) | (*text-options*)
specifies the text properties of the inset's title string. See "General Syntax for Attribute Options" on page 849 for the syntax on using a *style-element* and "Text Options" on page 852 for available *text-options*.

Default: The GraphValueText 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 639.

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

ROW2AXISOPTS=(*axis-options*)
specifies Y2-axis options for all rows. For a list of options, see "Axis Options for LAYOUT DATALATTICE/DATAPANEL" on page 639.

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

Details: This option is needed only if you use a plot statement that supports a secondary Y2 axis. If you do not use that statement's YAXIS= option to map data to the Y2 axis, this option is ignored. For more information about how data are mapped to the axes, see "Plot Data Are Mapped to a Designated Axis" on page 558

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-panel basis. The scaling does not span across multiple panels.

Tip: Use the ROWAXISOPTS= option to control shared axis features.

See Also: The PANELNUMBER= option for a discussion of how to create multiple panels.

ROW2DATARANGE=UNIONALL | UNION
specifies how the Y2-axes of instances of the *graph-prototype* are scaled.

Default: UNIONALL

UNIONALL
scales the Y2-axis data ranges across all layout rows and panels (when PANELNUMBER= is in effect).

UNION
scales the Y2-axis data ranges separately for each row in the layout on a per-panel basis. The scaling does not span across multiple panels.

Tip: Use the ROW2AXISOPTS= option to control shared axis features.

Details: This option is needed only if you use a plot statement that supports a secondary Y2 axis. If you do not use that statement's YAXIS= option to map data to the Y2 axis, this option is ignored. For more information about how data are mapped to the axes, see "Plot Data Are Mapped to a Designated Axis" on page 558

See Also: 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 how the rows are populated.

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

SHRINKFONTS=*boolean*
specifies whether fonts in the layout are scaled, depending on the nesting levels of the layouts that are used.

Default: FALSE. Fonts maintain their size regardless of the specifications in the nested layouts.

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

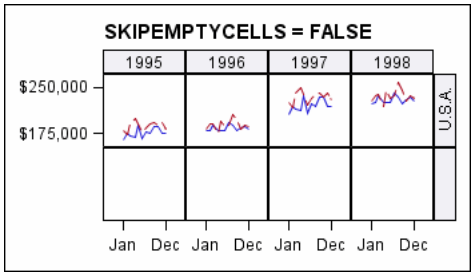
Default: FALSE

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

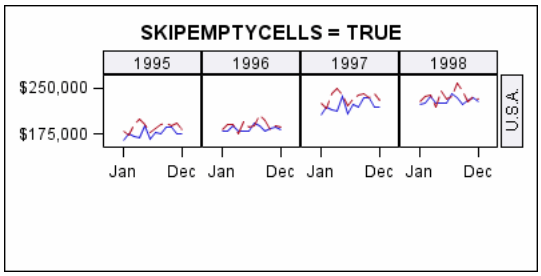
TRUE
External axes skip empty cells and "snap" to the nearest data cell, both vertically and horizontally. Though the empty cells are not displayed, the data cells in the grid are not enlarged to fill the area.

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

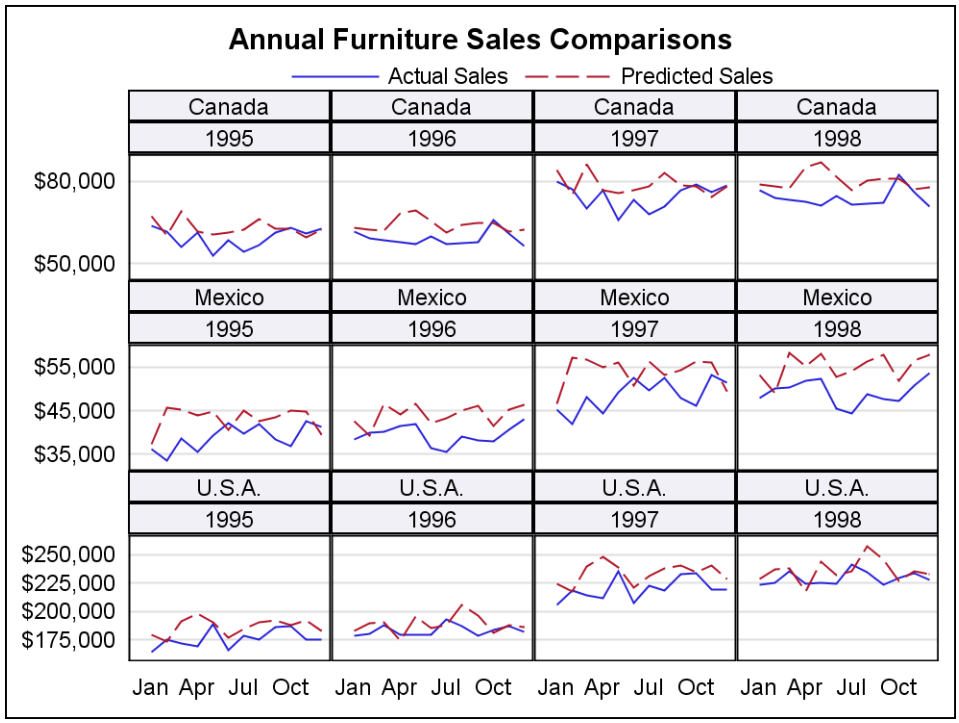
Creates a grid of graphs based on one or more classification variables and a graphical prototype. By default, a separate instance of the prototype (a data cell) is created for each actual combination of the classification variables.

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 52:



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 or 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

The CLASSVARS= argument is required to specify a list of classification variables, using the following syntax:

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

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 or 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.

If the class variable is of type character, its values are returned in data order. To control the ordering of the values, you can sort the input data by the classification variables. If the class variable is of type numeric, the values are displayed in ordinal order.

Formats can be assigned to class variables to create classification levels (for example, an AGEGROUPFMT. format for numeric AGE). In this case, the classification is performed after the format is applied. For numeric data, the order is ordinal, based on the first value in each class.

Use the INCLUDEMISSINGCLASS option to control whether cells are displayed when any value crossing contains a missing value.

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 the ROWS= , COLUMNS= , and PANELNUMBER= options to partition your data so that 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;

Note: You can specify only one LAYOUT PROTOTYPE block in the LAYOUT DATAPANEL block. If you specify more than one, only the last prototype block specified is honored. The remaining prototype blocks are ignored.

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 125.

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 </ *option(s)*>;

GTL-statement(s);

ENDSIDEBAR;

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;
```

SIDEBAR Options

ALIGN=BOTTOM | TOP | LEFT | RIGHT
specifies the sidebar's location within the layout.

Default: BOTTOM

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.

SPACEFILL=*boolean*
specifies whether to fill all the area of the sidebar with its contents.

Default: TRUE

To prevent a layout block within the sidebar from expanding to the sidebar boundaries, set this option to FALSE.

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 X-axis options for all columns.
COLUMN2AXISOPTS	Specifies X2-axis options for all columns.
COLUMNDATARANGE	Specifies how the X-axes of instances of the *graph-prototype* are scaled.

Statement Option	Description
COLUMN2DATARANGE	Specifies how the X2-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.
INCLUDEMISSINGCLASS	Specifies whether to include grid cells for crossings of the class variables that contain a missing value.
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.
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 Y-axis options for all rows.
ROW2AXISOPTS	Specifies Y2-axis options for all rows.
ROWDATARANGE	Specifies how the Y-axes of instances of the *graph-prototype* are scaled.
ROW2DATARANGE	Specifies how the Y2-axes of instances of the *graph-prototype* are scaled.
ROWGUTTER	Specifies amount of empty space between the rows.

Statement Option	Description
ROWS	Specifies the number of rows in the layout.
SHRINKFONTS	Specifies whether fonts in the layout are scaled, depending on the nesting levels of the layouts that are used.
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 in order 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 849 for the syntax on using a *style-element* and "Line Options" on page 850 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 639.

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

COLUMN2AXISOPTS=(*axis-options*)
specifies X2-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.

Details: This option is needed only if you use a plot statement that supports a secondary X2 axis. If you do not use that statement's XAXIS= option to map data to the X2 axis, this option is ignored. For more information about how data are mapped to the axes, see "Plot Data Are Mapped to a Designated Axis" on page 558

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.

Tip: Use the COLUMNAXISOPTS= option to control shared axis features.

See Also: The PANELNUMBER= option for a discussion of how to create multiple panels.

COLUMN2DATARANGE=UNIONALL | UNION
specifies how the X2-axes of instances of the *graph-prototype* are scaled.

Default: UNIONALL

UNIONALL
scales the X2-axis data ranges across all layout columns and panels (when PANELNUMBER= is in effect).

UNION
scales the X2-axis data ranges separately for each column on a per-panel basis. The scaling does not span across multiple panels.

Tip: Use the COLUMN2AXISOPTS= option to control shared axis features.

Details: This option is needed only if you use a plot statement that supports a secondary X2 axis. If you do not use that statement's XAXIS= option to map data to the X2 axis, this option is ignored. For more information about how data are mapped to the axes, see "Plot Data Are Mapped to a Designated Axis" on page 558

See Also: 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 are 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 in order 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 849 for the syntax on using a *style-element* and "Text Options" on page 852 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.

INCLUDEMISSINGCLASS=*boolean*
specifies whether to include grid cells for crossings of the CLASSVARS variables that contain a missing value.

Default: TRUE. Any crossing of the class variables that includes a missing value produces a cell in the grid.

If this option is set to FALSE, any crossing of the class variables that contains a missing value does not produce a cell in the grid.

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 *SAS 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 849 for the syntax on using a *style-element* and "Text Options" on page 852 for available *text-options*.

Default: The GraphDataText style element.

TITLE= "*string*"
specifies a title for the inset. The title is added at the top of the inset and spans the full inset width.

Default: no default, and space is not reserved for the title when it is not set

Tip: Text properties for the title string can be set with TITLEATTRS=.

TITLEATTRS=*style-element* | *style-element* (*text-options*) | (*text-options*)
specifies the text properties of the inset's title string. See "General Syntax for Attribute Options" on page 849 for the syntax on using a *style-element* and "Text Options" on page 852 for available *text-options*.

Default: The GraphValueText 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 639.

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

ROW2AXISOPTS=(*axis-options*)
specifies Y2-axis options for all rows. For a list of options, see "Axis Options for LAYOUT DATALATTICE/DATAPANEL" on page 639.

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

Details: This option is needed only if you use a plot statement that supports a secondary Y2 axis. If you do not use that statement's YAXIS= option to map data to the Y2 axis, this option is ignored. For more information about how data are mapped to the axes, see "Plot Data Are Mapped to a Designated Axis" on page 558

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-panel basis. The scaling does not span across multiple panels.

Use the ROWAXISOPTS= option to control shared axis features.

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

ROW2DATARANGE=UNIONALL | UNION
specifies how the Y2-axes of instances of the *graph-prototype* are scaled.

Default: UNIONALL

UNIONALL
scales the Y2-axis data ranges across all layout rows and panels (when PANELNUMBER= is in effect).

UNION
scales the Y2-axis data ranges separately for each row in the layout on a per-panel basis. The scaling does not span across multiple panels.

Tip: Use the ROW2AXISOPTS= option to control shared axis features.

Details: This option is needed only if you use a plot statement that supports a secondary Y2 axis. If you do not use that statement's YAXIS= option to map data to the Y2 axis, this option is ignored. For more information about how data are mapped to the axes, see "Plot Data Are Mapped to a Designated Axis" on page 558

See Also: 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 are 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 how the rows are populated.

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

SHRINKFONTS=*boolean*
specifies whether fonts in the layout are scaled, depending on the nesting levels of the layouts that are used.

Default: FALSE. Fonts maintain their size regardless of the specifications in the nested layouts.

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

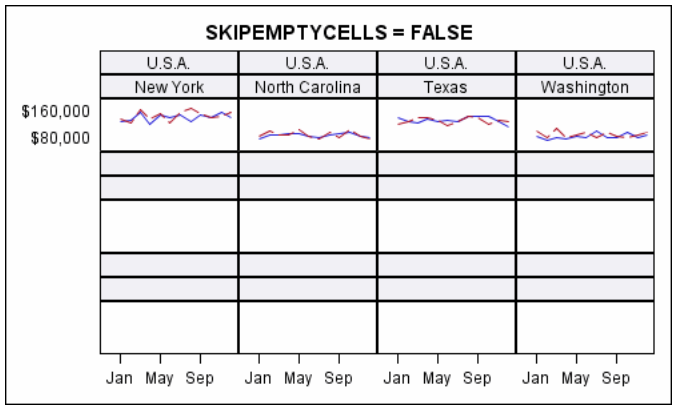
Default: FALSE

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

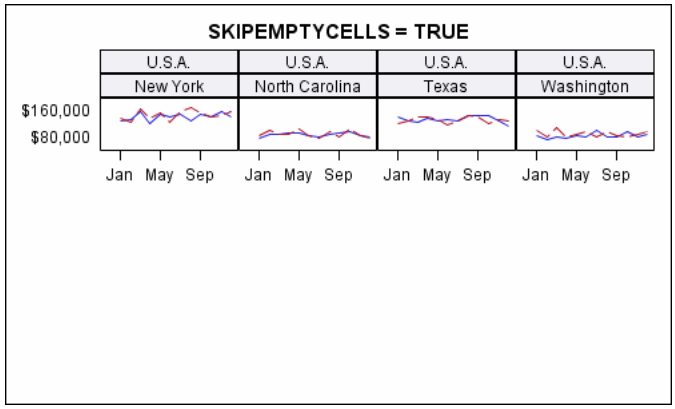
TRUE
External axes skip empty cells and "snap" to the nearest data cell, both vertically and horizontally. Though the empty cells are not displayed, the data cells in the grid are not enlarged to fill the area.

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

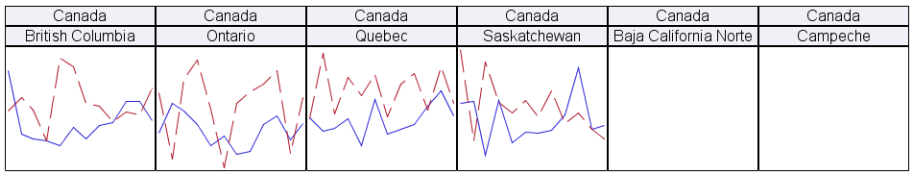
The number of data cells is the product of the unique values for each classification variable.

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.

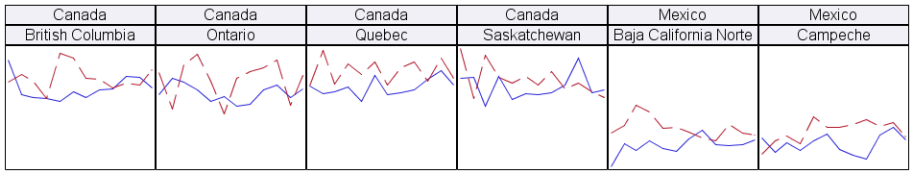
Assume that a data panel layout is created with COLUMNS=6 and SPARSE=TRUE, meaning to display all possible crossings. This is what the first row would look like. Blank data cells are added whenever there are no data values for a crossing:

SPARSE = TRUE



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

SPARSE = FALSE



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 GLOBALLEGEND Statement

Description and Syntax

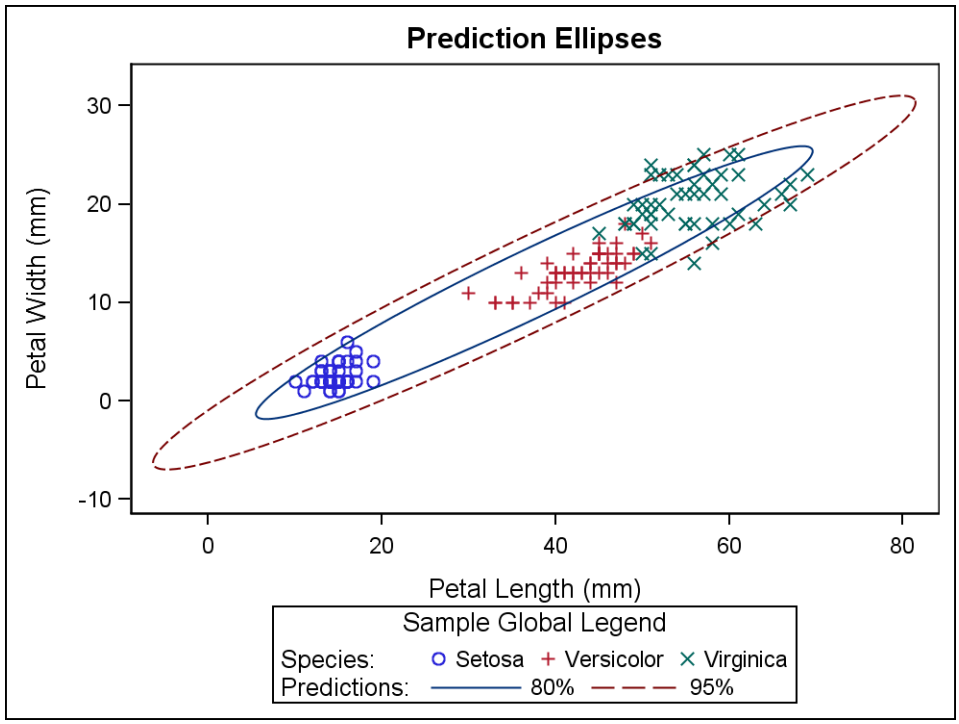
Creates a compound legend containing multiple discrete legends positioned at the bottom of a graph.

LAYOUT GLOBALLEGEND </*option(s)*>;
DiscreteLegend-statements | *MergedLegend-statements*;
ENDLAYOUT;

Example Program and Statement Details

Example Graph

The following graph was generated by the "Example Program" on page 70:



Example Program

```
proc template;
  define statgraph globallegend;
    begingraph;
      entrytitle "Prediction Ellipses";
      layout overlay;
        scatterplot x=petallength y=petalwidth / group=species name="sp";
        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;
      endlayout;
      layout globalLegend / type=column title="Sample Global Legend";
        discretelegend "sp" / title="Species:";
        discretelegend "p80" "p95" / title="Predictions:";
      endLayout;
    endgraph;
  end;
run;

proc sgrender data=sashelp.iris template=globallegend;
run;
```

Statement Summary

A global legend layout can contain multiple discrete or merged legends. Continuous legends are not supported inside the global legend block.

A global legend is placed at the bottom of the graph, just above the footnote(s). All of the discrete or merged legend statements that are nested within the global legend block

are arranged into a single row or column, depending on the setting for the TYPE= option.

Depending on the outermost layout type and the legend content, the legend is centered on the graph wall area or on the graph output area. For example, if the outermost layout is an overlay layout, when positioning the legend, the GLOBALLEGEND statement first attempts to center the legend on the graph wall area. If that position causes the legend to be clipped, it then attempts to center the legend on the entire output area instead. In that case, the legend might appear to be slightly off-center with respect to the graph.

Only one global legend block is permitted in a graph. The block must be located within the BEGINGRAPH block, but outside of the outermost layout block.

When a global legend block is used, only legend statements within the block are displayed in the graph. Any legend statements that are specified outside of the global legend block are ignored.

Options

Statement Option	Description
BORDER	Specifies whether a border is drawn around the legend.
BORDERATTRS	Specifies the properties of the border line around the legend.
DISPLAYCLIPPED	Specifies whether the global legend is displayed when any portion of its nested legends cannot be fully rendered because of space constraints.
GUTTER	Specifies the gap between nested legends.
HALIGN	Specifies the legend's horizontal alignment within the graph area that is defined by the BEGINGRAPH block.
LEGENDTITLEPOSITION	Specifies the position of each nested legend's title.
PAD	Specifies the amount of extra space that is added inside the legend border.
TITLE	Specifies a title for the global legend.
TITLEATTRS	Specifies the color and font attributes of the global legend title.
TYPE	Specifies whether nested legends are arranged into a single row or column.
WEIGHTS	Specifies the preferred space allocation for the nested legends.

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

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 legend. See "General Syntax for Attribute Options" on page 849 for the syntax on using a *style-element* and "Line Options" on page 850 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 global legend is displayed when any portion of its nested legends cannot be fully rendered because of space constraints.

Default: FALSE.

Interaction: This option overrides any DISPLAYCLIPPED option that is set on its nested legend statements.

Details: When the graph size is reduced, parts of a nested legend (title, legend symbol, or legend value) might be clipped (truncated). When clipping occurs and this option is FALSE, the entire global legend is removed from the graph and the space for it is reclaimed by the remainder of the graph. When this option is TRUE, the global legend always appears, even if some parts of the nested legends have been clipped.

GUTTER= *dimension*
specifies the gap between nested legends.

Default: 0

HALIGN=CENTER | LEFT | RIGHT
specifies the legend's horizontal alignment within the graph area that is defined by the BEGINGRAPH block.

Default: CENTER

Tip: When CENTER is in effect and the outermost layout is an overlay-type layout, the global legend is centered below the wall area if it can fit within the wall width.

LEGENDTITLEPOSITION= LEFT | TOP
specifies the position of each nested legend's title. Specifying LEFT places each title to the left of the legend items for that legend. Specifying TOP places each title above the legend items for that legend.

Default: LEFT

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

(*pad-options*)
Enables separate settings for the left, right, top, and bottom padding dimensions. Use *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 a title for the global legend.

Default: no title is displayed for the global legend

Restriction: The *string* must be enclosed in quotation marks.

Tip: The title for the global legend is independent of the titles for its nested legends.

TITLEATTRS=*style-element* | *style-element* (*text-options*) | (*text-options*)
specifies the color and font attributes of the global legend title. See "General Syntax for Attribute Options" on page 849 for the syntax on using a *style-element* and "Text Options" on page 852 for available *text-options*.

Default: The GraphLabelText style element.

Interaction: For this option to have any effect, the TITLE= option must also be specified.

TYPE= ROW | COLUMN
specifies whether nested legends are arranged into a single row or column.

Default: ROW

Interaction: When this option is set to ROW, the relative width of each legend is determined by the setting for the WEIGHTS= option.

WEIGHTS= UNIFORM | PREFERRED | (*weight-list*)
Specifies the preferred space allocation for the nested legends.

Default: UNIFORM

UNIFORM	All nested legends get an equal amount of space.
PREFERRED	All nested legends get their preferred amount of space.
(*weight-list*)	A space-delimited list of preferred space allocations, enclosed in parentheses. The list can combine numbers with the keyword PREFERRED. Each number is a proportional weight for the corresponding nested legend (the weights do not have to sum to 1.0). Keyword PREFERRED specifies that the corresponding nested legend should be allocated its preferred space. The order of the weights that are specified in the list should correspond to the order of the legend statements that are nested in the GLOBALLEGEND layout.

Restriction: The option is supported only for TYPE=ROW.

Tip: When a *weight-list* is specified, all the legends using PREFERRED get their preferred space. Any remaining space is divided among the legends, in proportion to the numeric values specified in the *weight-list*.

Chapter 7

LAYOUT GRIDDED Statement

Description and Syntax

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

LAYOUT GRIDDED *</option(s)>*

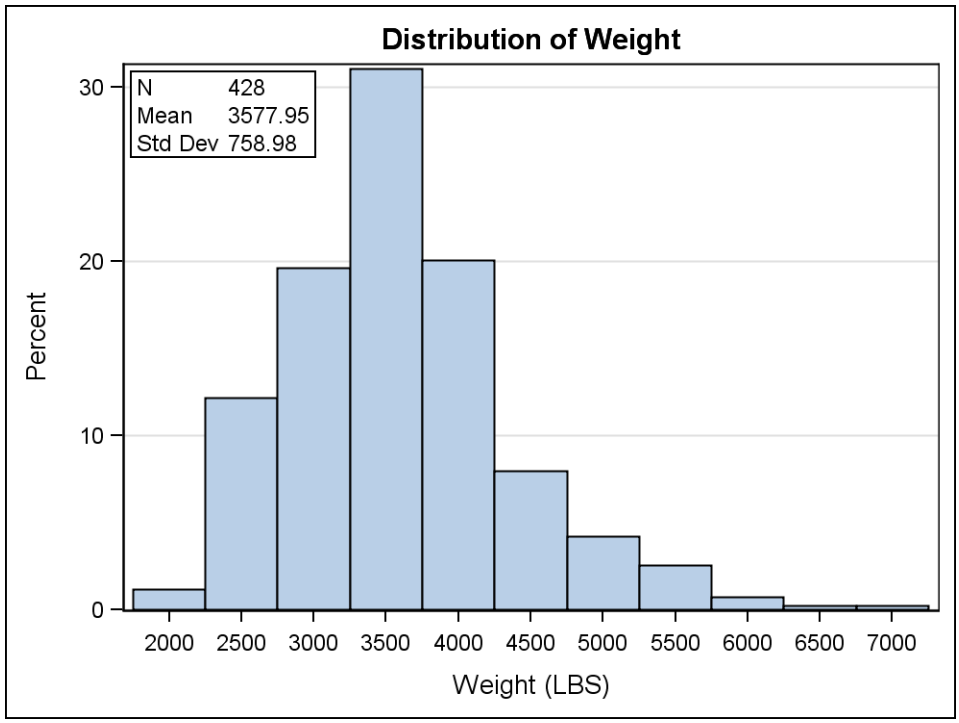
GTL-statements;

ENDLAYOUT;

Example Program and Statement Details

Example Graph

The following graph was generated by the "Example Program" on page 76:



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 or region 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 or region 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.
SHRINKFONTS	Specifies whether fonts in the layout are scaled when the size of the output is reduced.
VALIGN	Specifies this layout's vertical alignment within its parent when nested within an overlay-type or region layout.

AUTOALIGN=NONE | AUTO | *(location-list)*
specifies whether this layout is automatically aligned within its parent when nested within an overlay-type or region layout. For more information about how child positions are determined in an overlay-type or region layout, see the LAYOUT OVERLAY's "Statement Summary" on page 105 .

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 or region layout. This option is ignored if this statement is the outermost layout or if the parent layout is not an overlay-type or region 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 849 for the syntax on using a *style-element* and "Line Options" on page 850 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 | *number*
specifies this layout's horizontal alignment within its parent when nested within an overlay-type or region layout. For more information about how child positions are determined in an overlay-type or region layout, see the LAYOUT OVERLAY's "Statement Summary" on page 105 .

Default: CENTER

Range: A *number* specification can range from 0 to 1. The *number* represents a fraction of the parent container's width, where 0 is all the way to the left and 1 is all the way to the right.

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

Interaction: For a *number* setting to take effect, LOCATION=INSIDE must be set. A *number* setting is invalid on this option when LOCATION=OUTSIDE.

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 103.

Interaction: If this option is set to OUTSIDE, the HALIGN= and VALIGN= options must specify a keyword (LEFT, RIGHT, or CENTER). The *number* setting for the alignment is invalid when the layout is positioned outside of the plot area.

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;
```

SHRINKFONTS=*boolean*
specifies whether fonts in the layout are scaled when the size of the output is reduced.

Default: FALSE. Fonts maintain their size regardless of the output size.

VALIGN=CENTER | TOP | BOTTOM | *number*
specifies this layout's vertical alignment within its parent when nested within an overlay-type or region layout. For more information about how child positions are determined in an overlay-type or region layout, see the LAYOUT OVERLAY's "Statement Summary" on page 105 .

Default: CENTER

Range: A *number* specification can range from 0 to 1. The *number* represents a fraction of the parent container's height, where 0 is on the bottom and 1 is on the top.

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

Interaction: For a *number* setting to take effect, LOCATION=INSIDE must be set. A *number* setting is invalid on this option when LOCATION=OUTSIDE.

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

Chapter 8

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)*;>

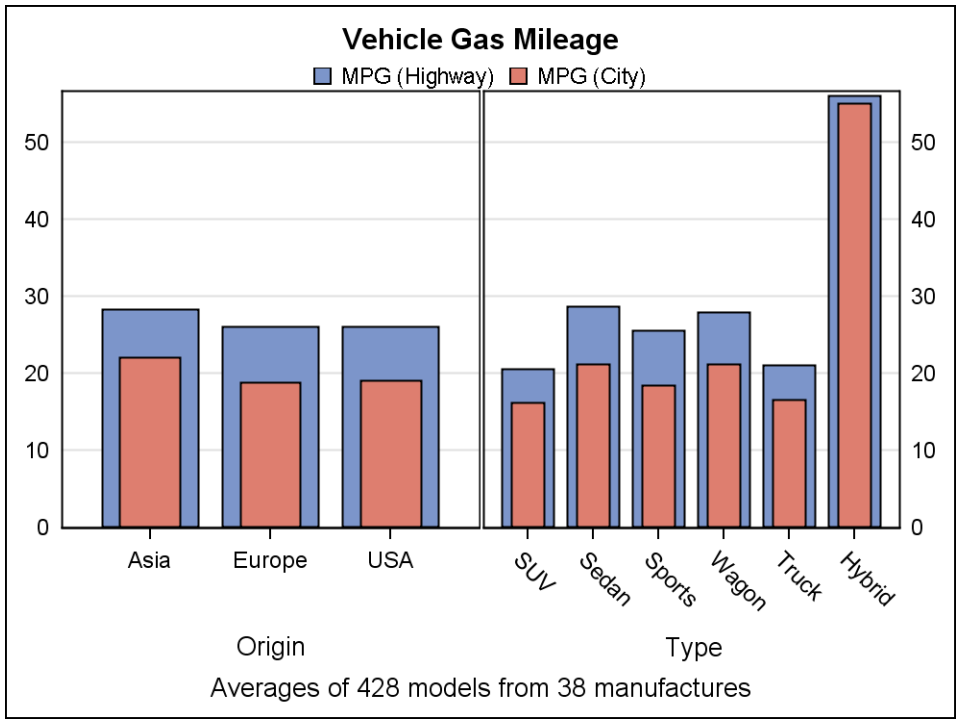
<*sidebar-statement-block(s)*;>

ENDLAYOUT;

Example Program and Statement Details

Example Graph

The following graph was generated by the "Example Program" on page 84:



Example Program

This example shows a two-cell lattice layout (two columns, one row). A ROWAXES block is used in the example to make the Y axes external to both cells.

- 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.
- Because the ROWAXIS statement within the ROWAXES block uses the DISPLAYSECONDARY= option, a secondary Y axis is displayed on the right. The secondary Y axis is not an independent axis. Rather, it mirrors the primary Y axis, making it easier to read Y-axis values when viewing the bar chart that is in the right cell.

```
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

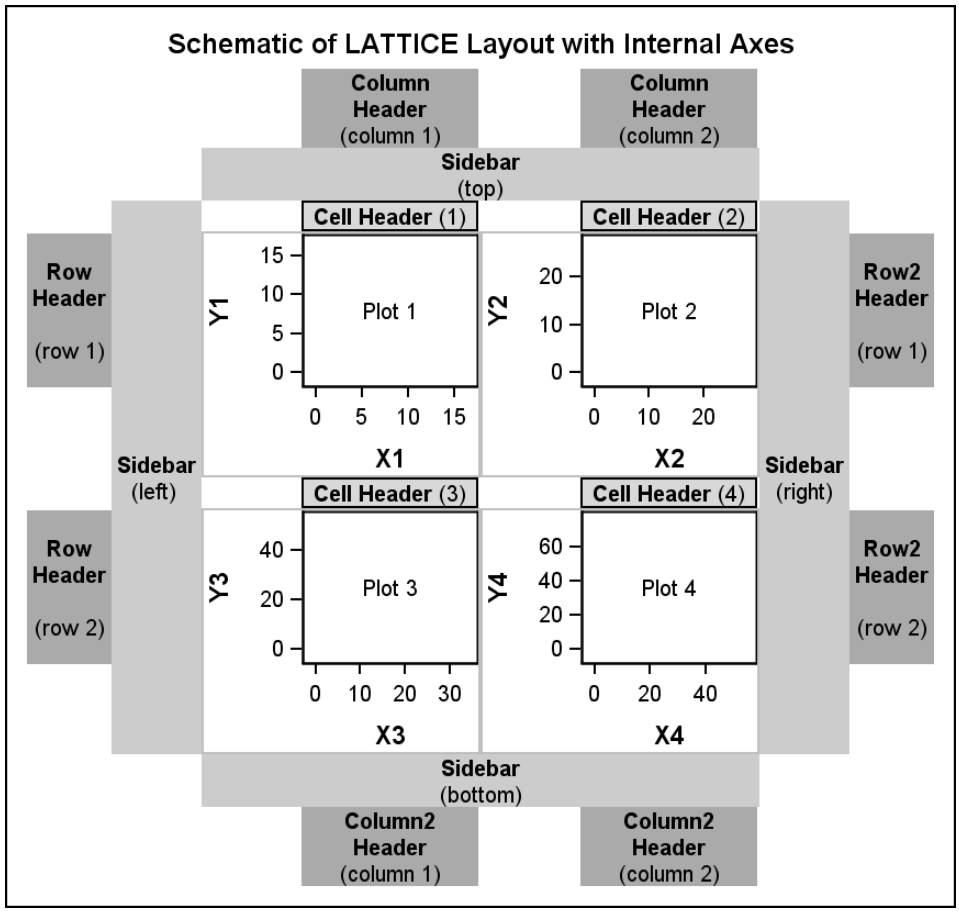
The LAYOUT LATTICE statement creates a grid of graphs that are aligned across columns and rows. For plot statements that are specified in the layout block or nested in a LAYOUT OVERLAY statement, the LATTICE layout automatically aligns the plot areas and tick display areas in the plots.

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 in 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 568.) 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. If one or more plots within the template use the XAXIS= option to produce independent X2 (top) axes, the X2 data scales can be unified, based on the values set for the COLUMN2DATARANGE= option. If one or more plots within the template use the YAXIS= option to produce independent Y2 (right) axes, the Y2 data scales can be unified, based on the values set for the ROW2DATARANGE= 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.

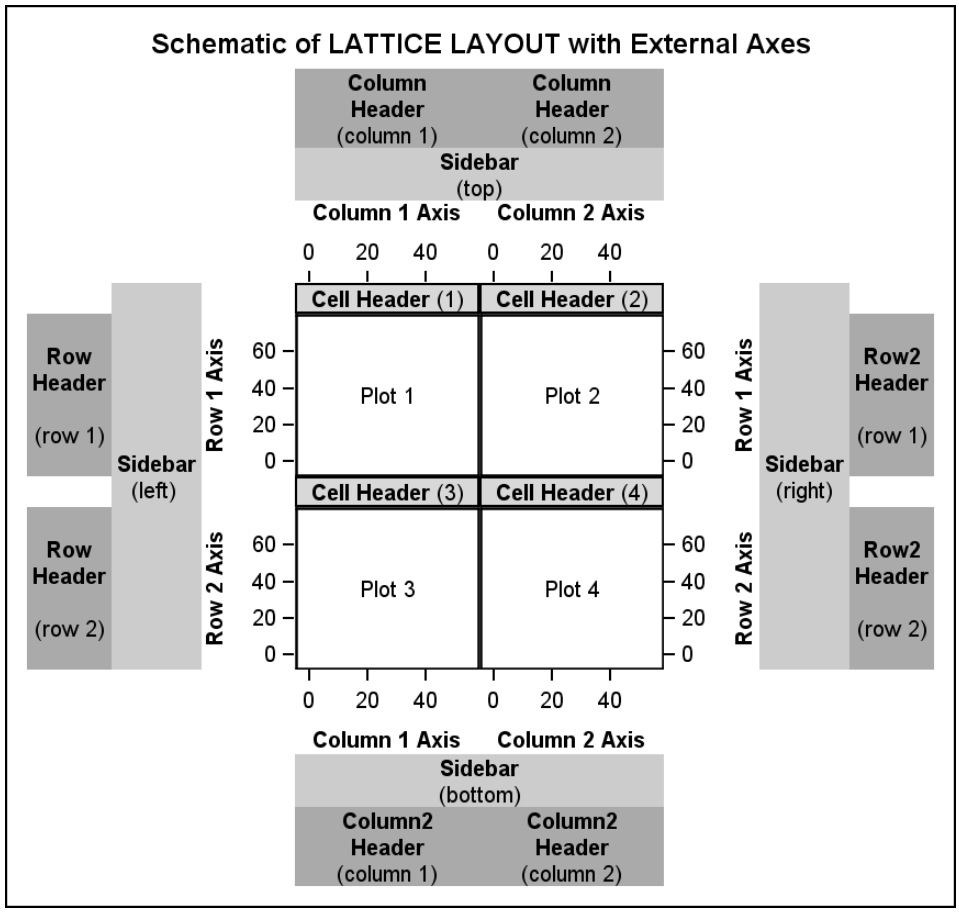
When the data-range scales are unified, you can simplify the layout by displaying only the external axes that apply to all of the graphs across the corresponding columns or rows. See "Axis Statements" on page 89 for more details.

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.

Note: The figure shows secondary X (top) and secondary Y (right) axes. The layout also enables you to generate independent X2 (top) and independent Y2 (right) axes. For details, see "Axis Statements" on page 89



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;

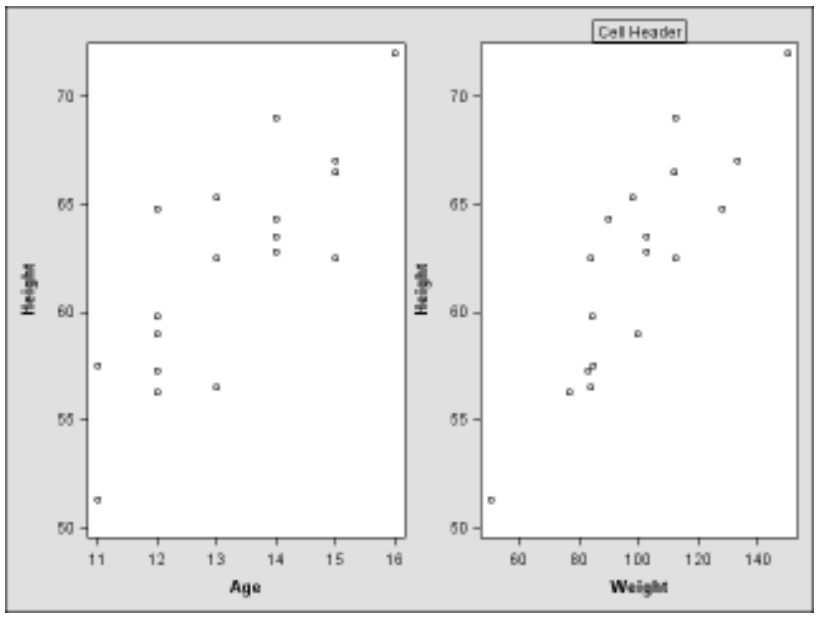
<CELLHEADER; *GTL-statement(s);*ENDCELLHEADER;>

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 that 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:



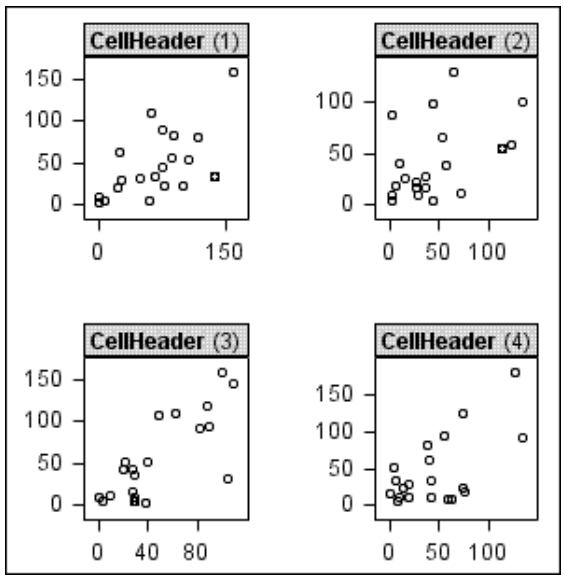

```
begingraph;
  layout lattice /
    columngutter=5 columns=2;

 /* 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;
```

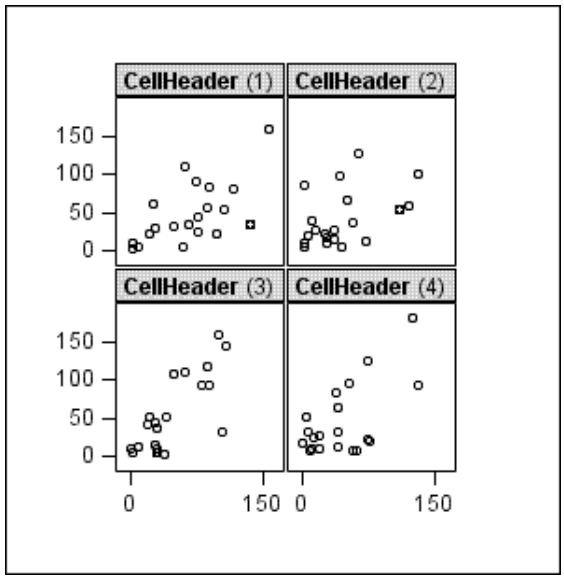
Axis Statements

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. If the axis ranges are not unified for the affected columns or rows, the axis statements in the layout are ignored.

To unify data ranges in the layout grid, the following options are available:

Axis	Option	Axis	Option
X	COLUMNDATARANGE=	X2	COLUMN2DATARANGE=
Y	ROWDATARANGE=	Y2	ROW2DATARANGE=

Specifying Axis Features. For columns, axis features for the external X axes (bottom) are specified within a COLUMNAXES block, nesting one COLUMNAXIS statement for each column that contains an X axis that you need to manage. The COLUMNAXIS statement provides a DISPLAYSECONDARY= option, which enables you to display a secondary X (top) axis that mirrors the primary X axis but can have different display features. In that case, the axis features that you specify in the COLUMNAXIS statement apply to both the primary and secondary X axes.

If one or more plots within the template use the XAXIS= option to produce independent X2 (top) axes, axis features for the external X2 axes (top) are specified within a COLUMN2AXES block, nesting one COLUMNAXIS statement for each column that contains an X2 axis that you need to manage. Within the COLUMN2AXES block, the COLUMNAXIS statement's DISPLAYSECONDARY= option enables you to display a secondary X2 (bottom) axis that mirrors the primary X2 axis but can have different display features. Here again, the axis features that you specify in the COLUMNAXIS statement apply to both the primary and secondary X2 axes.

Note: If you specify independent X and X2 scales at the same time, the DISPLAYSECONDARY= option is ignored in the COLUMNAXIS statement. This is true whether the COLUMNAXIS statement is specified in a COLUMNAXES or COLUMN2AXES block.

For both the COLUMNAXES and COLUMN2AXES blocks, if a lattice cell within the column contains a LAYOUT OVERLAY with the XAXISOPTS= or X2AXISOPTS=

option specified, these OVERLAY options are ignored. In such cases, the desired axis features should be specified in the COLUMNAXIS statement.

For rows, axis features for the external Y axes (left) are specified within a ROWAXES block, nesting one ROWAXIS statement for each row that contains a Y axis that you need to manage. The ROWAXIS statement provides a DISPLAYSECONDARY= option, which enables you to display a secondary Y (right) axis that mirrors the primary Y axis but can have different display features. In that case, the axis features that you specify in the ROWAXIS statement apply to both the primary and secondary Y axes.

If one or more plots within the template use the YAXIS= option to produce independent Y2 (right) axes, axis features for the external Y2 axes (right) are specified within a ROW2AXES block, nesting one ROWAXIS statement for each row that contains a Y2 axis that you need to manage. Within the ROW2AXES block, the ROWAXIS statement's DISPLAYSECONDARY= option enables you to display a secondary Y2 (left) axis that mirrors the primary Y2 axis but can have different display features. Here again, the axis features that you specify in the ROWAXIS statement apply to both the primary and secondary Y2 axes.

Note: If you specify independent Y and Y2 scales at the same time, the DISPLAYSECONDARY= option is ignored in the ROWAXIS statement. This is true whether the ROWAXIS statement is specified in a ROWAXES or ROW2AXES block.

For both the ROWAXES and ROW2AXES blocks, if a lattice cell within the row contains a LAYOUT OVERLAY with the YAXISOPTS= or Y2AXISOPTS= option specified, these OVERLAY options are ignored. In such cases, the desired axis features should be specified in the ROWAXIS statement.

Syntax and Restrictions for Axis Statements. The axis-statement blocks have the following general syntax:

COLUMNAXES; COLUMNAXIS / *axis-option(s)*; <…COLUMNAXIS-*n*;> ENDCOLUMNAXES;	**COLUMN2AXES**; COLUMNAXIS / *axis-option(s)*; <…COLUMNAXIS-*n*;> ENDCOLUMN2AXES;
ROWAXES; ROWAXIS / *axis-option(s)*; <…ROWAXIS-*n*;> ENDROWAXES;	**ROW2AXES**; ROWAXIS / *axis-option(s)*; <…ROWAXIS-*n*;> ENDROW2AXES;

In the LATTICE layout block, the following restrictions apply:

- If the LAYOUT LATTICE statement sets the row or column data range to DATA, the corresponding axes block is ignored. The data range must be set to UNION or UNIONALL to externalize the axes.
- Only one COLUMNAXES block can be used to manage X axes, and only one COLUMN2AXES block can be used to manage X2 axes. If more than one of either block is specified, only the last one of that block type is used.
- Within a COLUMNAXES or COLUMN2AXES block, one COLUMNAXIS statement should be specified for each column that contains axes that you need to manage. Both axes blocks can contain a COLUMNAXIS statement for the same column. For example, to manage the axes in the first column of the layout, the COLUMNAXES block can contain a COLUMNAXIS statement that manages the

column's X axes. The COLUMN2AXES block can contain a COLUMNAXIS statement that manages the column's X2 axes.

- Only one ROWAXES block can be used to manage Y axes, and only one ROWAXES block can be used to manage Y2 axes. If more than one of either block is specified, only the last one of that block type is used.
- Within a ROWAXES or ROW2AXES block, one ROWAXIS statement should be specified for each row that contains axes that you need to manage. Both axes blocks can contain a ROWAXIS statement for the same row. For example, to manage the axes in the first row of the layout, the ROWAXES block can contain a ROWAXIS statement that manages the row's Y axes. The ROW2AXES block can contain a ROWAXIS statement that manages the column's Y2 axes.
- 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 605.

The following example shows a LAYOUT LATTICE block that uses a ROWAXES block to set external axes and display grid lines 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. A ROW2AXES block is not needed because neither SCATTERPLOT statement in the template maps data to the Y2 axis.

For more information and examples that demonstrate how data are mapped to the axes, see "Plot Data Are Mapped to a Designated Axis" on page 558 .

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 83.) 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.

SPACEFILL=*boolean*
specifies whether to fill all the area of the sidebar with its contents.

Default: TRUE

Tip: To prevent a layout block within the sidebar from expanding to the sidebar boundaries, set this option to FALSE.

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 or region layout.

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.
COLUMNDATARANGE	Specifies how the X-axis data ranges of graphs within the layout columns are scaled.
COLUMN2DATARANGE	Specifies how the X2-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 or region 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.
ROW2DATARANGE	Specifies how the Y2-axis data ranges of graphs within the layout are scaled.
ROWGUTTER	Specifies amount of empty space between the rows.
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.
SHRINKFONTS	Specifies whether fonts in the layout are scaled, depending on the nesting levels of the layouts that are used.

Statement Option	Description
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 or region layout.

AUTOALIGN= NONE | AUTO | *(location-list)*
specifies whether this layout is automatically aligned within its parent when nested within an overlay-type or region layout. For more information about how child positions are determined in an overlay-type layout, see the LAYOUT OVERLAY's "Statement Summary" on page 105 .

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 or region layout. This option is ignored if this statement is the outermost layout or if the parent layout is not an overlay-type or region 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 849 for the syntax on using a *style-element* and "Line Options" on page 850 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 565.

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 565.

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 that contain X-axes to manage.

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 X-axis features. For more information, see "Axis Statements" on page 89 .

COLUMN2DATARANGE= DATA | UNION | UNIONALL
specifies how the X2-axis data ranges of graphs within the layout columns are scaled.

Default: DATA

DATA
scales the X2-axis data ranges separately for each cell in the layout.

UNION
scales the X2-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 565.

UNIONALL
scales the X2-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 565.

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 COLUMN2AXES block with as many COLUMNAXIS statements as there are columns that contain X2-axes to manage.

If column axes are externalized, and if a lattice cell contains a LAYOUT OVERLAY with the X2AXISOPTS= option specified, the X2AXISOPTS option is ignored. In such cases, the COLUMNAXIS statement should be used to specify desired X2-axis features. For more information, see "Axis Statements" on page 89 .

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 | *number*
specifies this layout's horizontal alignment within its parent when nested within an overlay-type or region layout. For more information about how child positions are determined in an overlay-type or region layout, see the LAYOUT OVERLAY's "Statement Summary" on page 105 .

Default: CENTER

Range: A *number* specification can range from 0 to 1. The *number* represents a fraction of the parent container's width, where 0 is all the way to the left and 1 is all the way to the right.

Restriction: This option is available only when this statement is nested within an overlay-type or region layout. It is ignored if this statement is the outermost layout or if the parent layout is not an overlay-type or region 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 565.

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 565.

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 that contain Y-axes to manage.

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 Y-axis features. For more information, see "Axis Statements" on page 89 .

ROW2DATARANGE= DATA | UNION | UNIONALL
specifies how the Y2-axis data ranges of graphs within the layout rows are scaled.

Default: DATA

DATA
scales the Y2-axis data ranges separately for each cell in the layout.

UNION
scales the Y2-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 565.

UNIONALL
scales the Y2-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 565.

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 ROW2AXES block with as many ROWAXIS statements as there are rows that contain Y2-axes to manage.

If row axes are externalized, and if a lattice cell contains a LAYOUT OVERLAY with the Y2AXISOPTS= option specified, the Y2AXISOPTS option is ignored. In such cases, the ROWAXIS statement should be used to specify desired Y2-axis features. For more information, see "Axis Statements" on page 89 .

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.

SHRINKFONTS=*boolean*
specifies whether fonts in the layout are scaled, depending on the nesting levels of the layouts that are used.

Default: FALSE. Fonts maintain their size regardless of the specifications in the nested layouts.

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 | *number*

specifies this layout's vertical alignment within its parent when nested within an overlay-type or region layout. For more information about how child positions are determined in an overlay-type or region layout, see the LAYOUT OVERLAY's "Statement Summary" on page 105 .

Default: CENTER

Range: A *number* specification can range from 0 to 1. The *number* represents a fraction of the parent container's height, where 0 is on the bottom and 1 is on the top.

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

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

Chapter 9

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)*>;

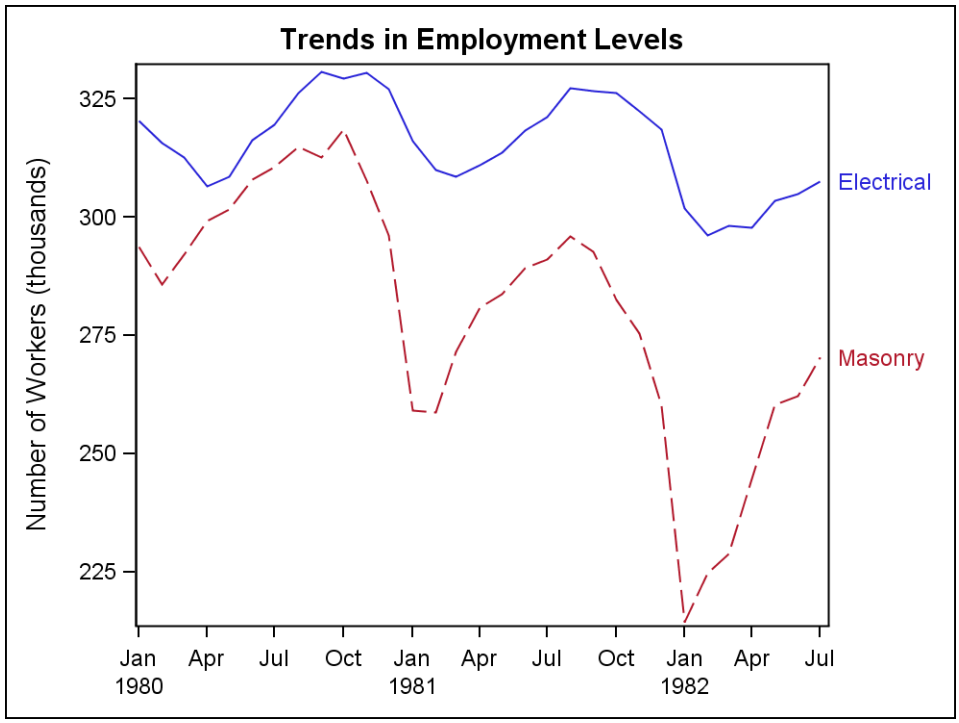
GTL-statements;

ENDLAYOUT;

Example Program and Statement Details

Example Graph

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



Example Program

```
data workers;
  format Date monyy5.;
  input Date monyy5. Electric Masonry;
datalines;
JAN80    320.3    293.8
FEB80    315.7    285.8
MAR80    312.6    292
APR80    306.5    299.3
MAY80    308.6    301.7
JUN80    316.3    307.9
JUL80    319.5    310.7
AUG80    326.4    314.9
SEP80    330.8    312.7
OCT80    329.3    318.5
NOV80    330.6    307.7
DEC80    327.2    296.2
JAN81    316.2    259.2
FEB81    310.1    258.8
MAR81    308.5    271.5
APR81    311.1    281
MAY81    313.6    283.7
JUN81    318.3    289.3
JUL81    321.3    291.1
AUG81    327.4    295.9
SEP81    326.7    292.7
OCT81    326.4    282.6
NOV81    322.5    275.5
DEC81    318.6    260.2
JAN82    301.9    214.3
```

```
FEB82    296.1    224.8
MAR82    298.3    228.7
APR82    297.7    244.7
MAY82    303.5    260.4
JUN82    305     262.2
JUL82    307.6    270.4
;
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=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 two-dimensional 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 (for example, CONTINUOUSLEGEND or DISCRETELEGEND).

The following general logic applies to rendering the composite:

Note: The details for positioning insets also apply to insets that are specified within a LAYOUT REGION block.

- 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 some insets, you can 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. This positioning achieves the best results for text-based insets whose size can be easily fit within an open area of the graph wall. A large text-based inset might not fit well, and

an inset that contains a plot might be dropped from the display without warning when the template is executed.

- 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 562.

See Also: LAYOUT OVERLAYEQUATED and LAYOUT OVERLAY3D.

INNERMARGIN Statement

An "inner margin" is a nested region at the top or bottom of the OVERLAY container. One or more inner margin plots can be specified, and each is specified within an INNERMARGIN block. Within the INNERMARGIN block, only a block plot can be specified. The following general syntax is used for the INNERMARGIN statement:

INNERMARGIN </ ALIGN= BOTTOM | TOP>;

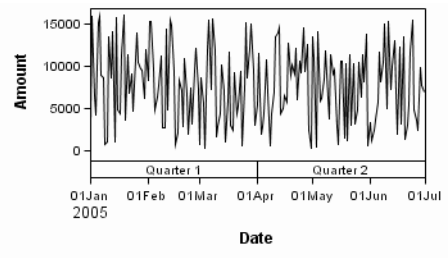
block-plot-statement(s);

ENDINNERMARGIN;

- ALIGN=BOTTOM is the default alignment.
- Two or more INNERMARGIN blocks having the same alignment are stacked.
- Multiple BLOCKPLOT statements within an INNERMARGIN are stacked.
- Offsets on each end of the Y axis are increased to make space for the inner margin plots.

The following example places a block plot in the inner margin region. The , INNERMARGIN statement uses the default alignment, so the inner margin is positioned at the bottom of the layout container, beneath the series plot:

```
layout overlay;
  innermargin;
    blockplot x=date block=quarter /
      display=(outline values)
      valuehalign=center;
  endinnermargin;
  seriesplot x=date y=amount;
endlayout;
```



Options

Statement Option	Description
ASPECTRATIO	Specifies the aspect ratio of the plot's wall area.
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.
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.

ASPECTRATIO= AUTO | *positive-number*
specifies the aspect ratio of the plot's wall area. The ratio is expressed as a positive decimal fraction representing *wall-height* divided by *wall-width*. For example, 0.75 is a 3/4 aspect ratio and 1.0 is a square aspect ratio.

Default: AUTO. The wall area is sized to the maximum area that can fill the available space inside the OVERLAY layout.

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 849 for the syntax on using a *style-element* and "Line Options" on page 850 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 143 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 571.

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 571.

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 571.

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 571.

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

Chapter 10

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)*>;

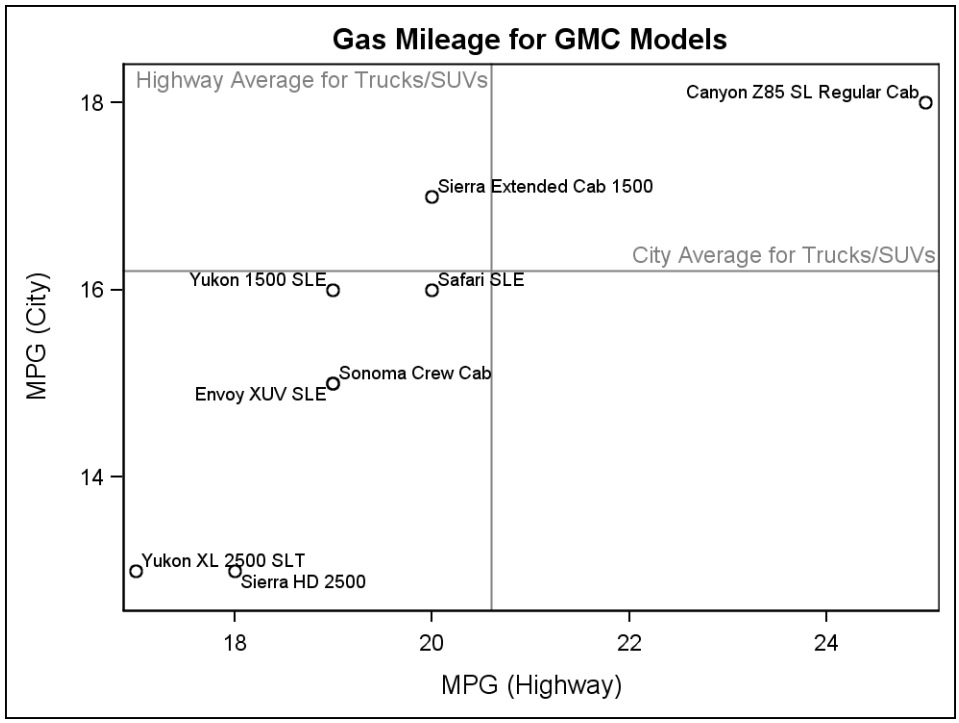
GTL-statements;

ENDLAYOUT;

Example Program and Statement Details

Example Graph

The following graph was generated by the "Example Program" on page 112:



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,

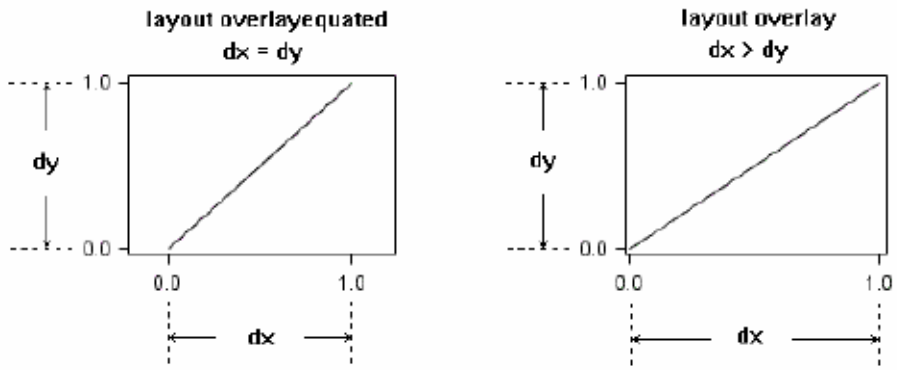
provided all plots can share the same type of axes. (Additional restrictions are discussed in a moment.) You can also specify one or more insets.

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 103 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 562).

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 849 for the syntax on using a *style-element* and "Line Options" on page 850 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 628.

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

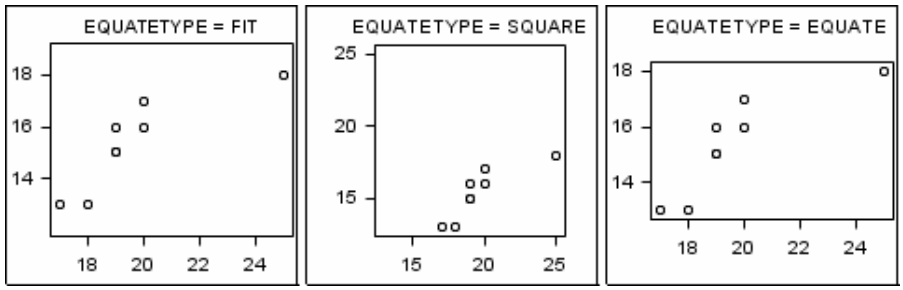
CYCLEATTRS=*boolean*
specifies whether the default visual attributes of markers, lines, and fills in nested plot statements automatically change from plot to plot. See "Cycling through Group Attributes in Overlaid Plots" on page 143 for more information.

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 631.

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 631.

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

Chapter 11

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)*>;

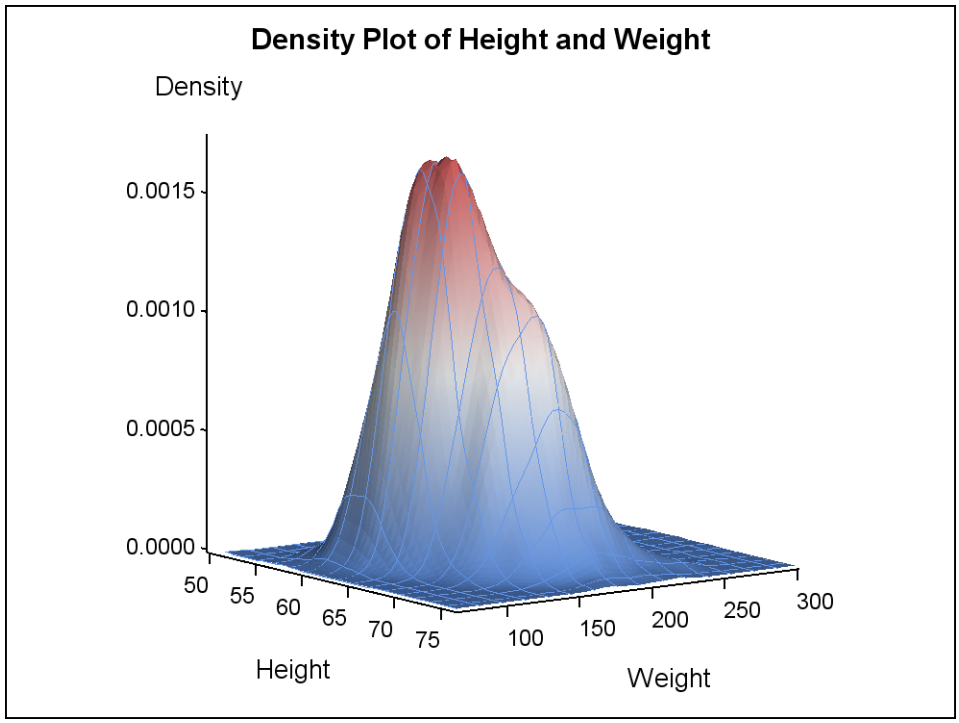
GTL-statements;

ENDLAYOUT;

Example Program and Statement Details

Example Graph

The following graph was generated by the "Example Program" on page 120:



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 103 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 562).

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 849 for the syntax on using a *style-element* and "Line Options" on page 850 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 143 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 593.

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 593.

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 593.

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 12

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 51.

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:

- You can specify only one LAYOUT PROTOTYPE block in a LAYOUT DATALATTICE or LAYOUT DATAPANEL block. If you specify more than one, only the last prototype block specified is honored. The remaining prototype blocks are ignored.
- Only non-computed two-dimensional plots can be included in the LAYOUT PROTOTYPE block. Thus, plots such as BOXPLOT, HISTOGRAM, DENSITYPLOT, ELLIPSE, LOESSPLOT, MODELBAND, and REGRESSIONPLOT cannot be used. In addition, a SCATTERPLOTMATRIX plot, a three-dimensional plot, or a region plot such as PIECHART or MOSAICPLOTPARM cannot be used in the LAYOUT PROTOTYPE block.
- A plot statement cannot be used if it contains a column defined with an EVAL expression.
- You can add one or more two-dimensional plots and one-dimensional 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.

Options

Statement Option	Description
ASPECTRATIO	Specifies the aspect ratio of the prototype cell.
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.

ASPECTRATIO= AUTO | *positive-number*
specifies the aspect ratio of the prototype cell. The ratio is expressed as a positive decimal fraction representing *wall-height* divided by *wall-width*. For example, 0.75 is a 3/4 aspect ratio and 1.0 is a square aspect ratio.

Default: AUTO. The prototype cell is sized to the maximum area that can fill the available space inside the layout cell.

Note: If AUTO is not used for the aspect ratio, the entire DATALATTICE or DATAPANEL grid is affected and changes shape.

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 143 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.

Chapter 13

LAYOUT REGION Statement

Description and Syntax

Creates the drawing area for a graph that does not use axes.

LAYOUT REGION *</option(s)>*

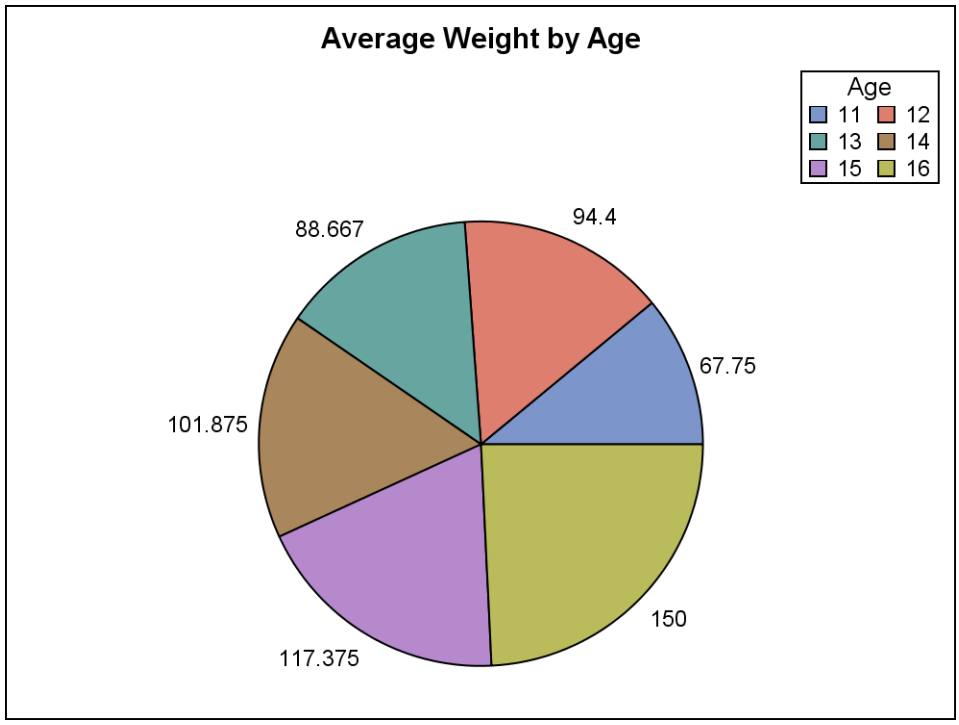
GTL-statements;

ENDLAYOUT;

Example Program and Statement Details

Example Graph

The following graph was generated by the "Example Program" on page 130:



Example Program

```
proc template;
  define statgraph layoutregion;
    begingraph;
      entrytitle "Average Weight by Age";
      layout region;
        piechart category=age response=weight /
          stat=mean name="p"
          datalabelcontent=(response) datalabellocation=outside;
        discretelegend "p" / title="Age" across=2
          border=true halign=right valign=top;
      endlayout;
    endgraph;
  end;

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

Statement Summary

The REGION layout provides a container for graphs that do not use axes. Within the LAYOUT REGION block, you can specify a single plot statement of a type that never uses axes, such as a PIECHART. If multiple plot statements are specified, only the first one is honored. 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). For example, you could specify a PIECHART statement with a DISCRETELEGEND statement and an ENTRY statement. You can also nest one or more layout blocks within the REGION layout. For example, you could nest a LAYOUT GRIDDED statement that creates a small table of text.

To control the horizontal and vertical positioning of some insets, you can use the inset statement's HALIGN= or VALIGN= options. Each nested inset determines its own relative position in the parent REGION. This positioning achieves the best results for text-based insets whose size can be easily fit within an open area of the graph wall. A large text-based inset might not fit well, and an inset that contains a plot might be dropped from the display without warning when the template is executed. For more information about how child positions are determined in an overlay-type or region layout, see the LAYOUT OVERLAY's "Example Program and Statement Details" on page 103.

When nested within another layout type, such as a GRIDDED or LATTICE layout, the REGION layout defines the graphical display for one cell of the parent layout. A separate REGION layout is specified for each cell.

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.
OPAQUE	Specifies whether the layout background is opaque or transparent.
PAD	Specifies the amount of extra space that is added inside the layout border.

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 849 for the syntax on using a *style-element* and "Line Options" on page 850 for available *line-options*.

Default: The GraphBorderLines style element.

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

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.

Part 4

Plot Statements

Chapter 14

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 that 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 Template Language: User's Guide*. For a more complete discussion of the ODS GRAPHICS statement, see *SAS Output Delivery System: User's 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. *SAS Graph Template Language: User's 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.

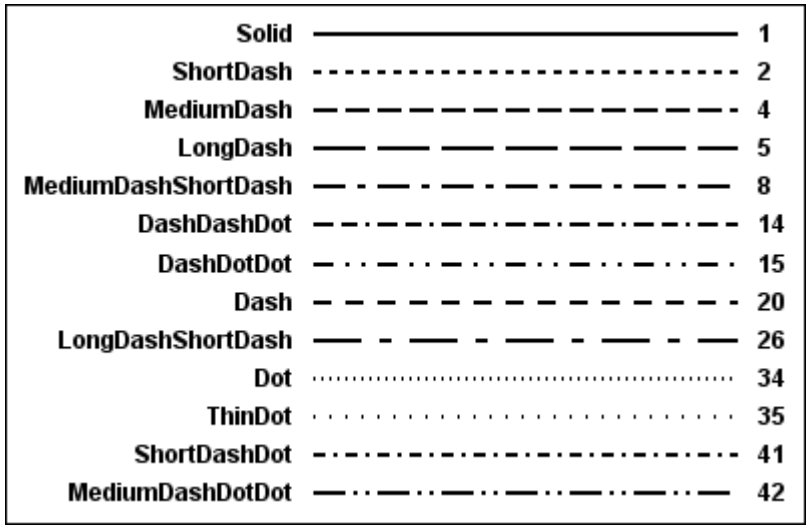
Display Attributes for Non-Grouped Data

"Display Attributes" on page 849 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 14.1 on page 140 shows the most common line patterns available for the PATTERN= suboption.

Display 14.1 *"Common Line Patterns"*



- 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 line-style numbers for the line patterns

"Available Line Patterns" on page 853 provides the complete list of line patterns that can be used with the GTL.

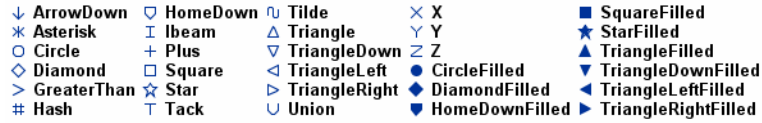
In the following template definition, the LINEPARM statement's LINEATTRS= option overrides the default line pattern by specifying PATTERN=DASH:

```
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 14.2 on page 140 shows the marker symbols available for the SYMBOL= suboption.

Display 14.2 *"Marker Symbols"*



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

Note: The MarkerSize and LineThickness style attributes are not honored in the case of grouped data.

Display 14.1 on page 140 shows the common line patterns available, and Display 14.2 on page 140 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, or marker symbols are used to represent group values, then the style determines the sequences of the line patterns, colors, or marker symbols that are used for the group values. (As discussed in "Cycling through Group Attributes in Overlaid Plots" on page 143, other plot settings might also influence the sequence.) If the number of group values exceeds the number of style elements, the following occurs for the subsequent group values:

- The sequence of line patterns and plot markers that are specified by the style is repeated.
- New fill colors are generated by repeating the GraphData1–GraphDataN colors and varying the shade of each original color on each cycle. The shade variations alternate between one shade lighter and one shade darker on each cycle.

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 pattern 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 some 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. In this example, the style is derived from the DEFAULT style, which is available for the 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, 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. To apply the style to a graph, the STYLE= option is used in the ODS HTML statement to specify the style name.

Here is the code for this example.

```
/* Sort the SASHELP.CLASS data by sex and age. */
proc sort data=sashelp.class(keep=height weight sex age)
     out=class;
  by sex age;
run;

/* Generate slope and intercept data for plot reference lines. */
proc robustreg data=class method=m
      outest=stats(rename=(weight=slope));
   by sex;
   model height=weight;
run;

data class;
   merge class stats(keep=intercept slope sex);
run;

proc template;
   /* Create custom style STYLES.MYDEFAULT from the STYLES.DEFAULT style. */
   define style Styles.MyDefault;
      parent=Styles.Default;
      style GraphData1 from GraphData1 /
         LineStyle=6;
      style GraphData2 from GraphData2 /
         LineStyle=4;
   end;

   /* Create the plot template. */
   define statgraph testPattern;
      begingraph;
         layout overlay;
            scatterplot y=height x=weight / group=sex;
            lineparm x=0 y=intercept slope=slope / group=sex name="lines";
            discretelegend "lines";
         endlayout;
      endgraph;
    end;
run;
```

```
/* Generate the plot. */
ods _all_ close;
ods html style=MyDefault; /* Apply style MyDefault to the graph. */

proc sgrender data=class template=testPattern;
run;
```

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.

Here is the code for this example.

Note: The data that was generated in the previous example is used again in this example.

```
proc template;
   /* Create custom style STYLES.MYDEFAULT from the STYLES.DEFAULT style. */
   define style Styles.MyDefault;
      parent=Styles.Default;
      style GraphData1 from GraphData1 /
         MarkerSymbol="DIAMOND";
      style GraphData2 from GraphData2 /
         MarkerSymbol="CROSS";
      style GraphData3 from GraphData3 /
         MarkerSymbol="CIRCLE";
   end;

   /* Create the plot template. */
   define statgraph testSymbols;
      begingraph;
         layout Overlay;
            scatterPlot y=height x=weight / group=age name="symbols";
            discretelegend "symbols" / title="Age";
         endlayout;
      endgraph;
   end;
run;

/* Generate the plot. */
ods html close;
ods html style=MyDefault; /* Apply style MyDefault to the graph. */

proc sgrender data=class template=testSymbols;
run;
```

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 144), 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 14.1 *Study 1*

Drug1	Index1
A	1
B	2
B	2
C	3

Table 14.2 *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:

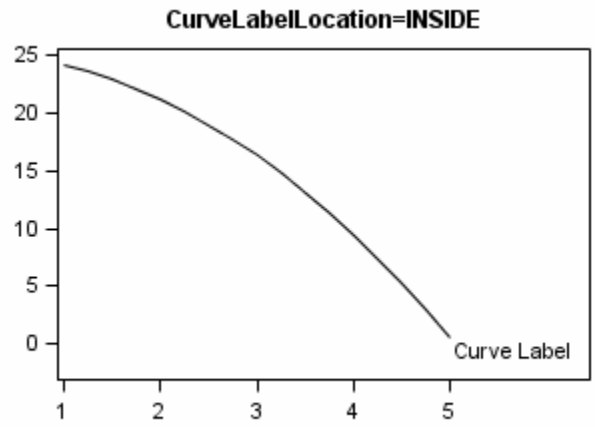
CURVELABEL
Specifies a label for the curve line.

CURVELABELLOCATION
Specifies the location of the curve line label relative to the plot area.

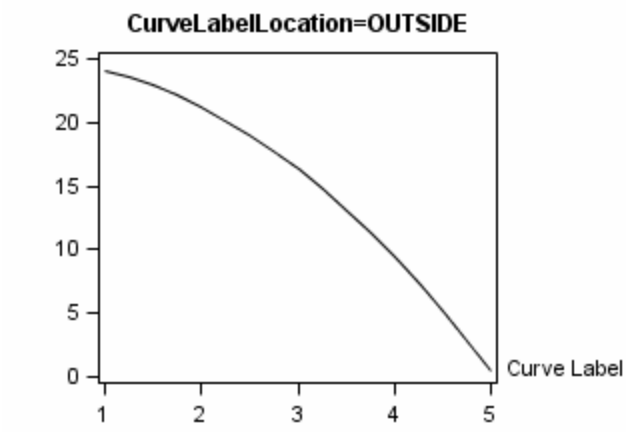
CURVELABELPOSITION
Specifies the position of the label relative to the curve line.

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 568) 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).

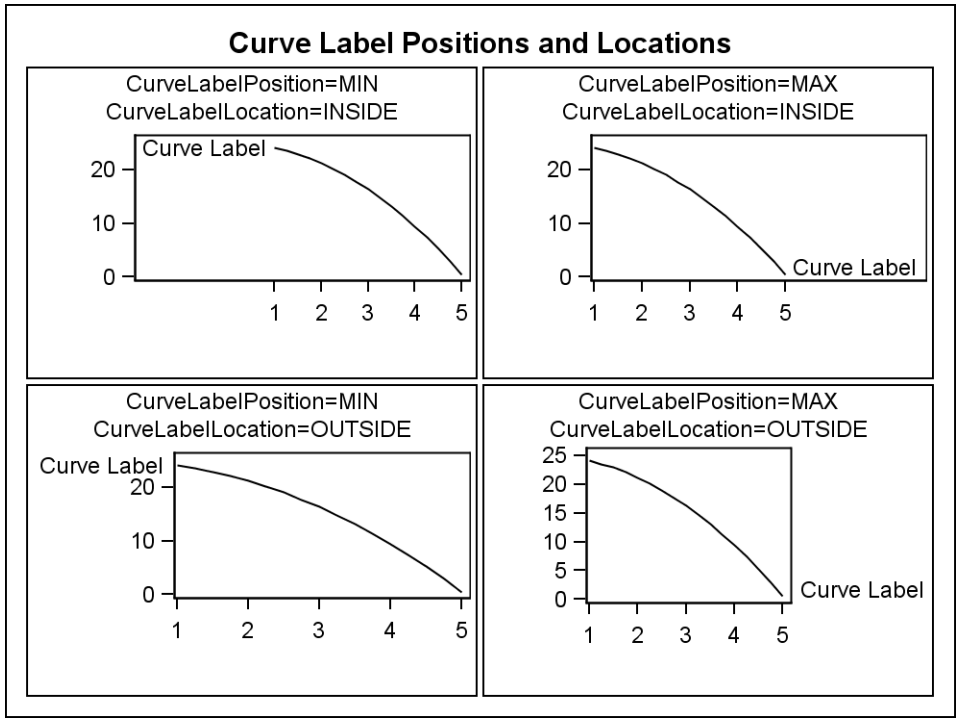
START
: Only used when CURVELABELLOCATION=INSIDE. Forces the curve label to appear near the beginning of the curve. Particularly useful when the curve line has a spiral shape.

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 568) 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 15

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)*>;

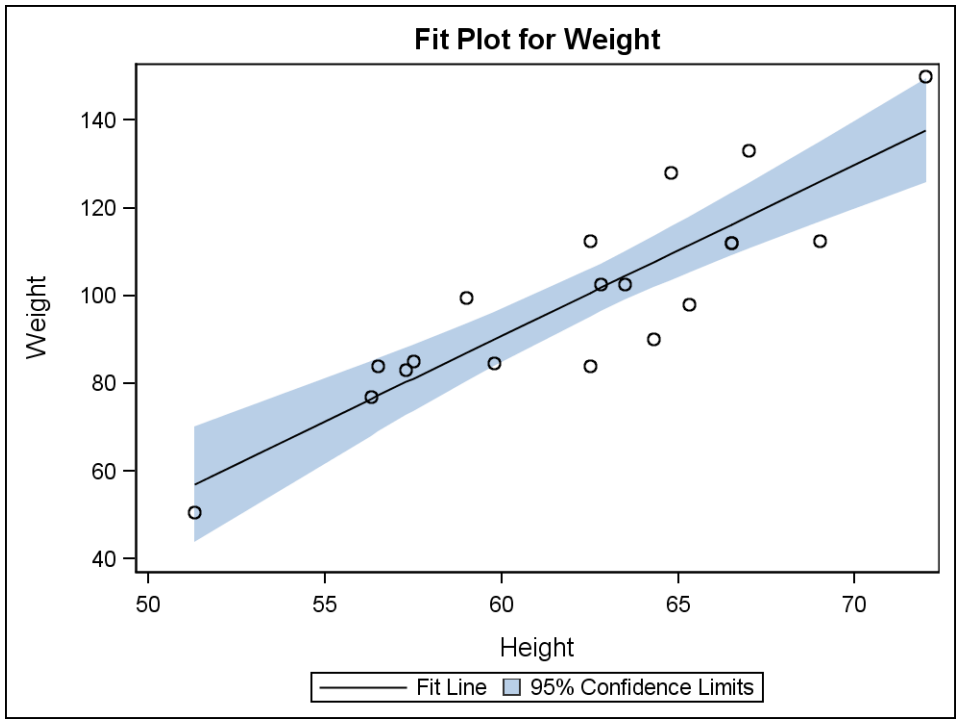
BANDPLOT Y = *numeric-column* | *expression*
LIMITLOWER = *number* | *numeric-column* | *expression*
LIMITUPPER = *number* | *numeric-column* | *expression* </option(s)>;

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 150:



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.

Note: The BANDPLOT statement is optimized to work as a Confidence or Prediction band. If the band is self intersecting (not sorted for X or for Y), the resulting band is unpredictable. With unsorted data, the band that is generated for an output Raster Image might not match the band that is generated for an output Vector Graphic.

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 or Y values 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 or Y values 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 or X values 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 or X values 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.

Statement 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 labels relative to the curve line.
DATATRANSPARENCY	Specifies the degree of the transparency of the band fill and band outline.
DISCRETEOFFSET	Specifies an amount to offset all bands from X values for a discrete X axis, or from Y values for a discrete Y axis.
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.
INCLUDEMISSINGGROUP	Specifies whether missing values of the group variable are included in the plot.
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 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 user-defined roles that can be used to display information in the tooltips.

Statement Option	Description
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.
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 this value to ensure 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 849 for the syntax on using a *style-element* and "Text Options" on page 852 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 labels are displayed

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 labels are displayed

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 146.

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 ignored if CURVELABELLOCATION=INSIDE is specified. The START and END settings are ignored 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 146.

DATATRANSPARENCY=*number*
specifies the degree of the transparency of the band fill and band outline.

Default: 0

Range: 0 (opaque) to 1 (entirely transparent)

Tip: The FILLATTRS= option can be used to set transparency for just the band area. You can combine this option with FILLATTRS= to set one transparency for the band outline but a different transparency for the band fill. Example:

```
datatransparency=0.2 fillattrs=(transparency=0.6)
```

DISCRETEOFFSET=*number*
specifies an amount to offset all bands 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.

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.

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 849 for the syntax on using a *style-element* and "Fill Options" on page 850 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.

Tip: The DATATRANSPARENCY= option sets the transparency for both the band fill and band outline. You can combine this option with DATATRANSPARENCY= to set one transparency for the band outline but a different transparency for the band fill. Example:

```
datatransparency=0.2 fillattrs=(transparency=0.6)
```

GROUP=*column* | *discrete-attr-var* | *expression*
creates a separate band plot for each unique group value of the specified column.

discrete-attr-var
specifies a discrete attribute variable that is defined in a DISCRETEATTRVAR statement.

Restriction: A discrete attribute variable specification must be a direct reference to the attribute variable. It cannot be set by a dynamic variable.

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.

Interaction: The INCLUDEMISSINGGROUP option controls whether missing group values are considered a distinct group value.

Tip: 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.

INCLUDEMISSINGGROUP=*boolean*
specifies whether missing values of the group variable are included in the plot.

Default: TRUE

Interaction: For this option to take effect, the GROUP= option must also be specified.

Tip: Unless a discrete attribute map is in effect or the INDEX= option is used, the attributes of the missing group value are determined by the GraphMissing style element except when the MISSING= system option is used to specify a non-default missing character or when a user-defined format is applied to the missing group value. In those cases, the attributes of the missing group value are determined by a GraphData1–GraphDataN style element.

INDEX=*numeric-column* | *expression*
specifies indices for mapping line attributes (color and pattern) to one of the GraphData1 - GranphDataN style elements.

Default: no default

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.

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: 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 144.

JUSTIFY=LEFT | CENTER | RIGHT
specifies the location of the data point relative to the step when TYPE=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 a label for the legend item that is associated with this plot.

Default: The *string* specified on the NAME= option.

Restriction: This option applies only to an associated DISCRETELEGEND statement.

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.

Tip: If this option is not specified, then the interpolation is set by the TYPE= option.

When this option is used, the band plot forms prediction or confidence limits for the plot that supplies the fitted model.

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 849 for the syntax on using a *style-element* and "Line Options" on page 850 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 that can be used to display information in the tooltips.

Default: no user-defined roles

(*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 557.

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 557.

Chapter 16

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)>;

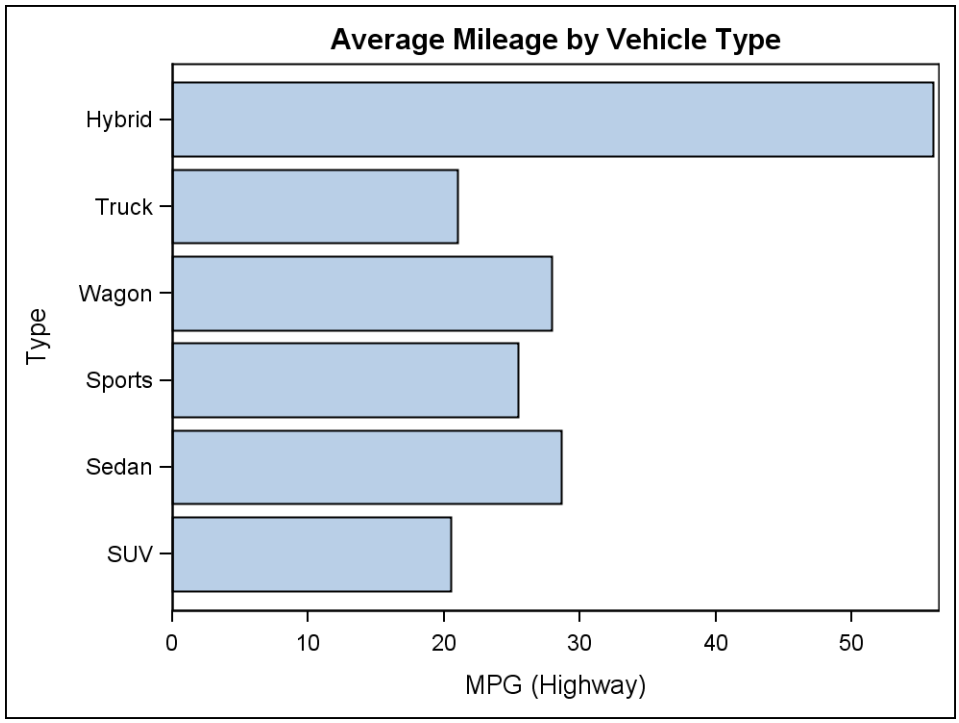
The following syntax creates a bar chart with bars representing summarized values of Y categorized by X:

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 162:



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;

proc sgrender data=sashelp.cars template=barchart;
run;
```

Statement Summary

The BARCHART statement takes raw, unsummarized data as input and calculates appropriate summarization statistics (sum, mean, and so on).

By default, if the category column is character, the bars in the chart appear in the order in which the category values are present in the input data. If the category column is numeric, the values are presented in ascending order. For non-grouped data, duplicated category values are summarized into a unique value. All values are treated as discrete. For grouped data, the category values are summarized as needed. (See the GROUP= option.)

The BARCHART statement creates a bar chart with bars that represent summarized response values that are categorized by the unique category values. To create a chart that represents response values along an interval axis, use the HISTOGRAM statement.

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.
BASELINEINTERCEPT	Specifies the response axis intercept for the baseline.
CLUSTERWIDTH	Specifies the width of the group clusters as a fraction of the midpoint spacing.
CONNECTATTRS	Specifies the properties of the line connecting bars.
DATASKIN	Enhances the visual appearance of the filled bars.
DATATRANSPARENCY	Specifies the degree of the transparency of the bar fill, bar outline, 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.
FILLPATTERNATTRS	Specifies the appearance of the pattern-filled bar area

Statement Option	Description
GROUP	Creates a separate segment or bar for each unique group value of the specified column.
GROUPDISPLAY	Specifies how to display grouped bars.
GROUPORDER	Specifies the order of the grouped bars for each category value.
INCLUDEMISSINGGROUP	Specifies whether missing values of the group variable are included in the plot.
INDEX	Specifies indices for mapping line attributes (color and pattern) to one of the GraphData1–GranphDataN style elements.
LEGENDLABEL	Specifies the label for 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.
STAT	Specifies the statistic to be computed for the Y-axis.
TARGET	Specifies the target value for each bar.
TIP	Specifies the information to display when the cursor is positioned over a bar.
TIPFORMAT	Specifies display formats for the information that is defined by the tooltip roles.
TIPLABEL	Specifies display labels for the information that is defined by the tooltip 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 849 for the syntax on using a *style-element* and "Text Options" on page 852 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. 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: 0.1 (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.

BASELINEINTERCEPT=*number*
specifies the response axis intercept for the baseline.

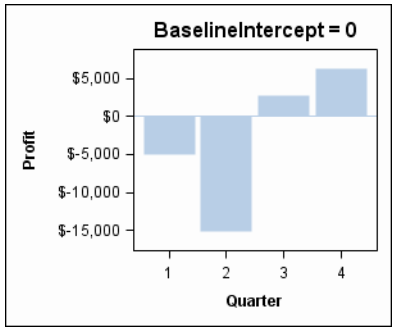
Default: 0

Interaction: Label positions are adjusted to prevent the labels from overlapping.

Interaction: If GROUPDISPLAY=STACKED is specified, this option is ignored and the baseline is not displayed.

Tip: The baseline does not add a tick or a tick value to the axis. To label the baseline, you can use a REFERENCELINE statement to overlay a line with the same X or Y value and use the REFERENCELINE statement's CURVELABEL= option.

Details: When this option is used, the axis range is adjusted to include the baseline, and the baseline is placed at the specified value on the response axis.



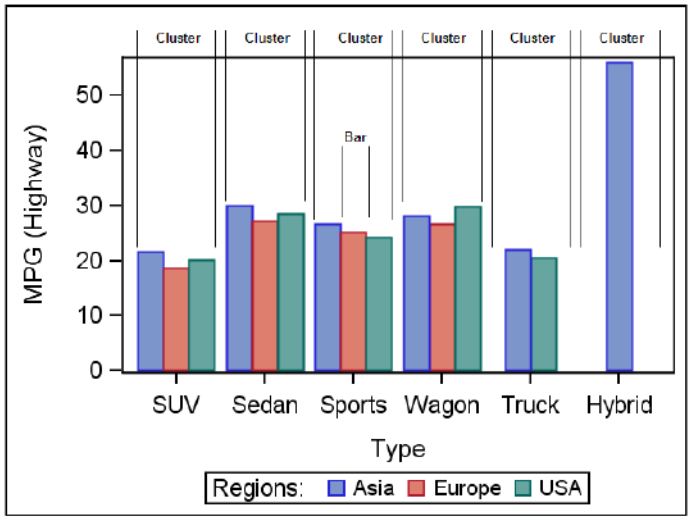
CLUSTERWIDTH= *number*
specifies the width of the group clusters as a fraction of the midpoint spacing on a discrete axis or a fraction of the minimum interval between adjacent data values on an interval axis.

Default: 0.85

Range: 0.1 (narrowest) to 1 (widest)

Interaction: For this option to take effect, the GROUP= option must also be specified, and the GROUPDISPLAY= option must be set to CLUSTER.

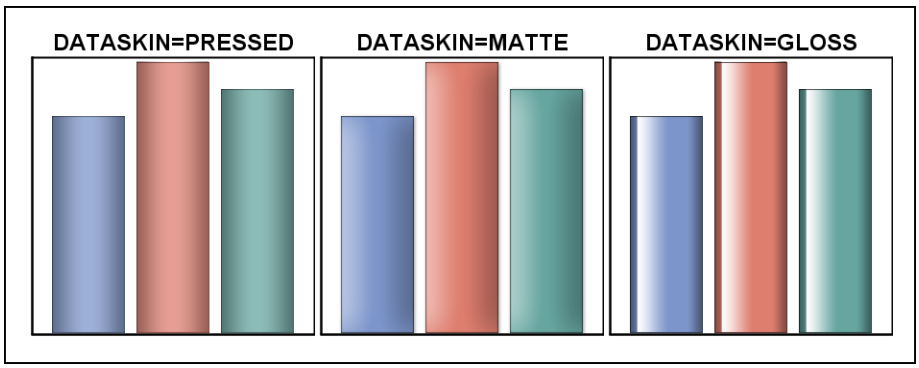
Note: When GROUPDISPLAY=CLUSTER, the default BARWIDTH is 1.0.



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 849 for the syntax on using a *style-element* and "Line Options" on page 850 for available *line-options*.

Default: The GraphConnectLine style element.

DATASKIN=NONE | PRESSED | SHEEN | CRISP | GLOSS | MATTE
enhances the visual appearance of the filled bars.



Default: NONE

Requirement: For this option to have any effect, the fill must be enabled by the ODS style or the DISPLAY= option.

Interaction: The data skin appearance is based on the FILLATTRS= color.

Interaction: When a data skin is applied, all bar outlines are set by the skin, and the OUTLINEATTRS= option is ignored.

DATATRANSPARENCY=*number*
specifies the degree of the transparency of the bar fill, bar outline, and connect line, if displayed.

Default: 0

Range: 0 (opaque) to 1 (entirely transparent)

Tip: The FILLATTRS = option can be used to set transparency for just the filled bar area. You can combine this option with FILLATTRS= to set one transparency for the bar outlines and connect lines but a different transparency for the bar fills. Example:

```
datatransparency=0.2 fillattrs=(transparency=0.6)
```

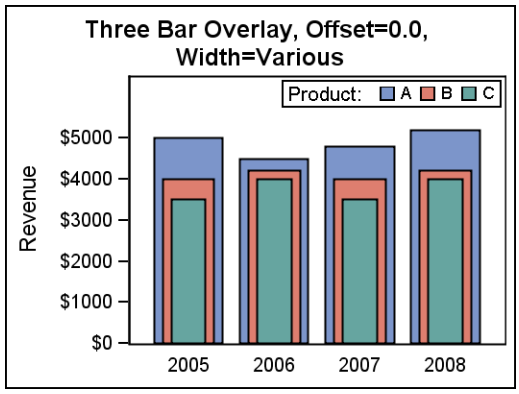
DISCRETEOFFSET=*number*
specifies an amount to offset all bars from the category midpoints.

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.)

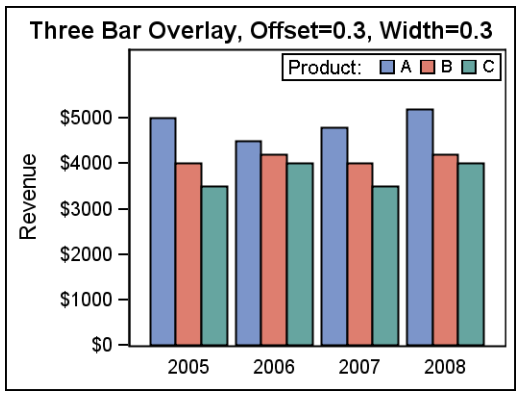
Details: This feature is useful for graphing multiple response variables side by side on a common axis. By default within an overlay-type layout, if multiple BARCHART statements are used with different response variables, the bars for matching category values are centered on the midpoints and the bars 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;
```



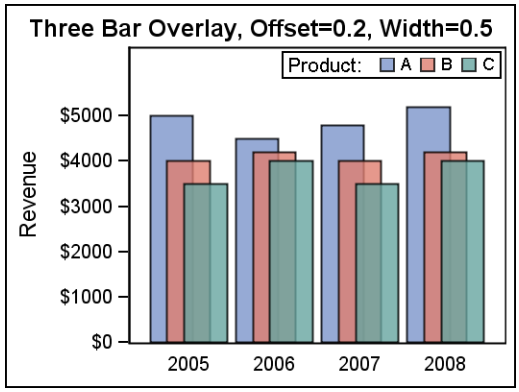
To place the different response values side by side, you can assign a different offset to each BARCHART statement. If desired, the BARWIDTH= option can be used in conjunction with DISCRETEOFFSET= to create narrower bars that require less width within the plot area:

```
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

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

OUTLINE	Displays outlined bars.
FILL	Displays bars with a solid fill.
CONNECT	Displays line segments connecting adjacent midpoints at the end of each bar
FILLPATTERN	Displays bars with a patterned fill. This setting is used primarily for grouped bar charts that must be rendered in monochrome for use in a journal article. The fill patterns make it easier to distinguish among groups when color is not available.

Restriction: Connect lines are not drawn for grouped data.

Use the OUTLINEATTRS=, FILLATTRS=, and FILLPATTERNATTRS= options to control the appearance of the bars. Use CONNECTATTRS= to control the appearance of the connect lines.

Tip: Both FILL and FILLPATTERN can be specified to combine solid fills and pattern fills in the bars.

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 849 for the syntax on using a *style-element* and "Fill Options" on page 850 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.

Tip: The DATATRANSPARENCY= option sets the transparency for the bar fills, bar outlines, and connect lines. You can combine this option with DATATRANSPARENCY= to set one transparency for the bar outlines and connect lines but a different transparency for the bar fills. Example:

```
datatransparency=0.2 fillattrs=(transparency=0.6)
```

FILLPATTERNATTRS=*style-element* | *style-element* (*fill-options*) | (*fill-pattern-options*)
specifies the appearance of the pattern-filled bar area.

Default: no default

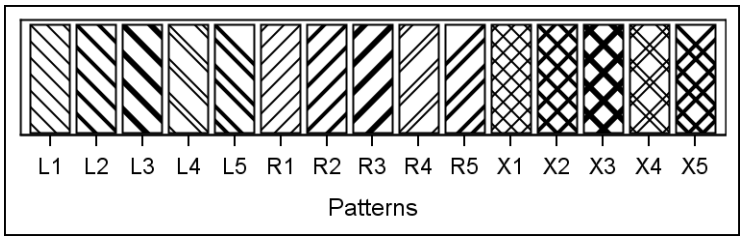
To specify a *style-element*, only one of the elements GraphData1–GraphDataN can be used, subject to the Restriction listed in a moment.

The following *fill-pattern-options* are available:

COLOR = *color* | *style-reference*
specifies a color to use for the bar-fill-pattern lines. With grouped data, this setting has the effect of holding the pattern line color constant across all group values.

PATTERN = *line-pattern*
specifies a line pattern to use for the bar fill.

To specify a *line-pattern*, combine a line-direction prefix (R for right, L for left, and X for cross hatch) with a line-identification number:



With grouped data, this PATTERN= setting has the effect of holding the fill pattern constant across all group values.

Restriction: The only SAS-delivered styles that support fill patterns are JOURNAL2, JOURNAL3, and MONOCHROMEPRINTER. If any other SAS-delivered style is in effect and this option uses *style-element* in its specification, this option is ignored.

Interaction: For this option to take effect, the DISPLAY= option must include FILLPATTERN among the display options.

GROUP=*column* | *discrete-attr-var* | *expression*
creates a separate bar segment or bar for each unique group value of the specified column.

discrete-attr-var
specifies a discrete attribute variable that is defined in a DISCRETEATTRVAR statement.

Restriction: A discrete attribute variable specification must be a direct reference to the attribute variable. It cannot be set by a dynamic variable.

For example, the *sashelp.cars* data used in the "Example Program" on page 162 contains a column named ORIGIN, which identifies the region that produces each car. This column could be used in the BARCHART statement to group the bars in the display (see the GROUPDISPLAY= option to see the output for the grouped bars):

```
layout overlay;
  barchart x=type y=mpg_highway /
    stat=mean group=origin name="b";
```

```
  discretelegend "b" / title="Regions:";
endlayout;
```

Default:

- If bar fills or fill patterns are enabled by the ODS style or the DISPLAY= option, each distinct group value is represented in the graph by a different fill color or fill pattern. The fill colors are defined by the GraphData1:Color–GraphDataN:Color style references. The fill patterns are defined by the GraphData1:FillPattern–GraphDataN:FillPattern 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 outline. The outline colors are defined by the GraphData1:ContrastColor–GraphDataN:ContrastColor style references.

Interaction: The bar display depends on the setting for the GROUPDISPLAY= option.

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.

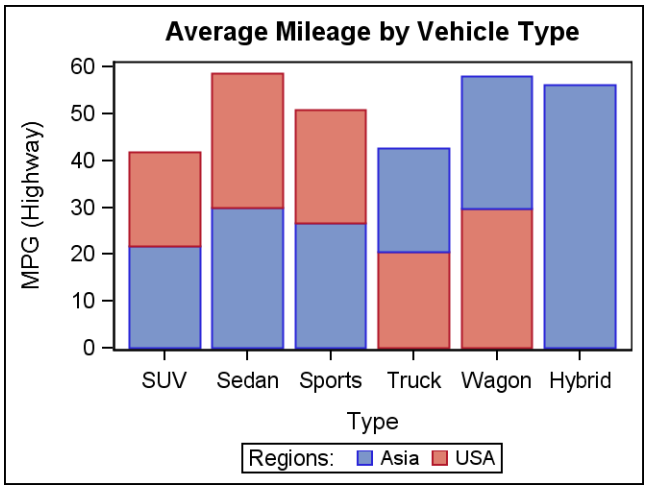
Interaction: The INCLUDEMISSINGGROUP option controls whether missing group values are considered a distinct group value.

Tip: 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.

GROUPDISPLAY=STACK | CLUSTER
specifies how to display grouped bars.

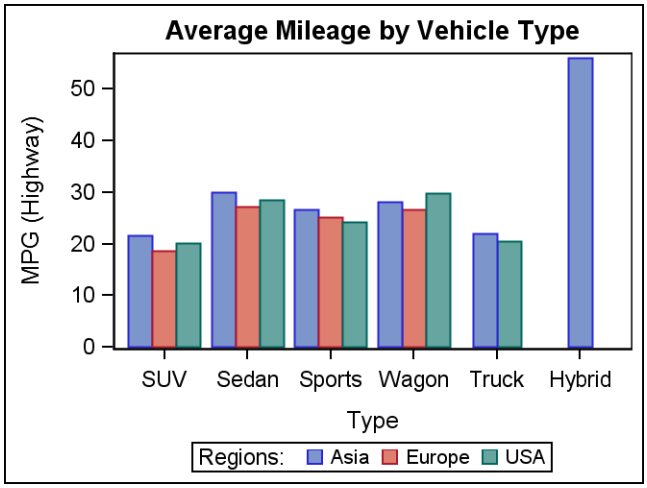
Default: STACK

STACK
Displays group values as stacked segments within the category bar.



CLUSTER
Displays group values as separate adjacent bars that replace the single category bar. Each set of group values is centered at the category midpoint on the axis.

This example illustrates the clusters and also how groups are displayed when they have an unequal number of unique values.



GROUPORDER=DATA | ASCENDING | DESCENDING
specifies the order of the grouped bars for each category value. It also sets the default order of the groups in 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 of the group column.

DESCENDING
shows each group of bars in descending order of the group column.

Interaction: This option is ignored unless GROUP= is specified.

Note: Attributes such as color, symbol, and pattern are assigned to each group in the DATA order by default regardless of the GROUPORDER= option setting.

Note: The ASCENDING and DESCENDING settings linguistically sort the group values within each category (or X value) for display position purposes only. The data order of the observations and the visual attributes that are assigned to the group values remain unchanged.

INCLUDEMISSINGGROUP=*boolean*
specifies whether missing values of the group variable are included in the plot.

Default: TRUE

Interaction: For this option to take effect, the GROUP= option must also be specified.

Tip: Unless a discrete attribute map is in effect or the INDEX= option is used, the attributes of the missing group value are determined by the GraphMissing style element except when the MISSING= system option is used to specify a non-default missing character or when a user-defined format is applied to the missing group value. In those cases, the attributes of the missing group value are determined by a GraphData1–GraphDataN style element.

INDEX=*numeric-column* | *expression*
specifies indices for mapping line attributes (color and pattern) to one of the GraphData1 - GranphDataN style elements.

Default: no default

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.

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: 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 144

LEGENDLABEL= "*string*"
specifies a label for the legend item that is associated with this plot.

Default: The Y-variable label. If a label is not defined, the Y-variable name.

Restriction: This option applies only to an associated DISCRETELEGEND statement.

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

Tip: When this option is set to VERTICAL, the category variable appears on the X (or X2) axis and the response variable appears on the Y (or Y2) axis. To set the axis properties for this chart, you should use the appropriate axis options of the layout container.

When this option is set to HORIZONTAL, the category variable appears on the Y (or Y2) axis and the response variable appears on the X (or X2) axis. To set the axis properties for this chart, you should use the appropriate axis options of the layout container.

Tip: If you change the orientation of the bar chart, you should adjust the layout container's axis options appropriately.

OUTLINEATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)
specifies the appearance of the bar outlines. See "General Syntax for Attribute Options" on page 849 for the syntax on using a *style-element* and "Line Options" on page 850 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.

Interaction: If the DATASKIN= option applies a data skin, this option is ignored.

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 562.

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:

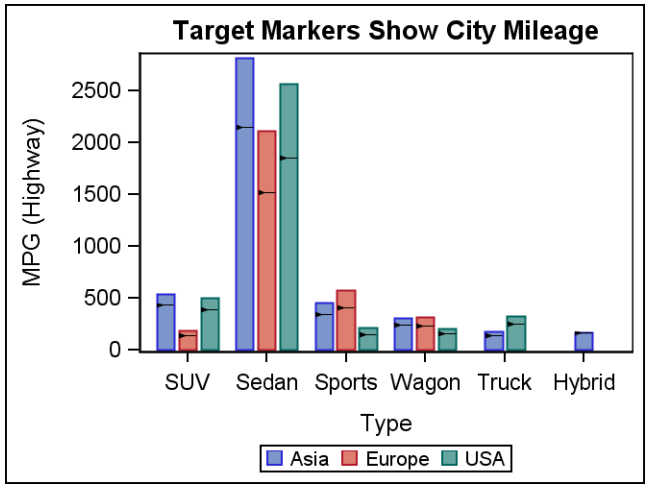
SUM
Sum

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.

TARGET=*numeric-column* | *expression*
specifies the target value for each bar. The visual representation is a triangle with a line at the target value.

```
layout overlay;
  barchart x=type y=mpg_highway / barwidth=.8
    target=mpg_city group=origin groupdisplay=cluster
    name='bar';
  discretelegend 'bar';
endlayout;
```



Default: No targets are displayed.

Interaction: For this option to take effect, the Y= argument must also be used.

Interaction: If the GROUP= option is used and GROUPDISPLAY= STACK, this option is ignored.

Tip: The statistic indicated by the STAT= option applies to the TARGET=column. If a constant value is desired for each target, specify it only once for repeated category (X) values (or category and GROUP combinations), and leave other target values missing.

Tip: The color of the target is that of the bar outline.

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 the information that is defined by the tooltip roles.

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 the information that is defined by the tooltip roles.

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 557.

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 557.

Chapter 17

BARCHARTPARM Statement

Description and Syntax

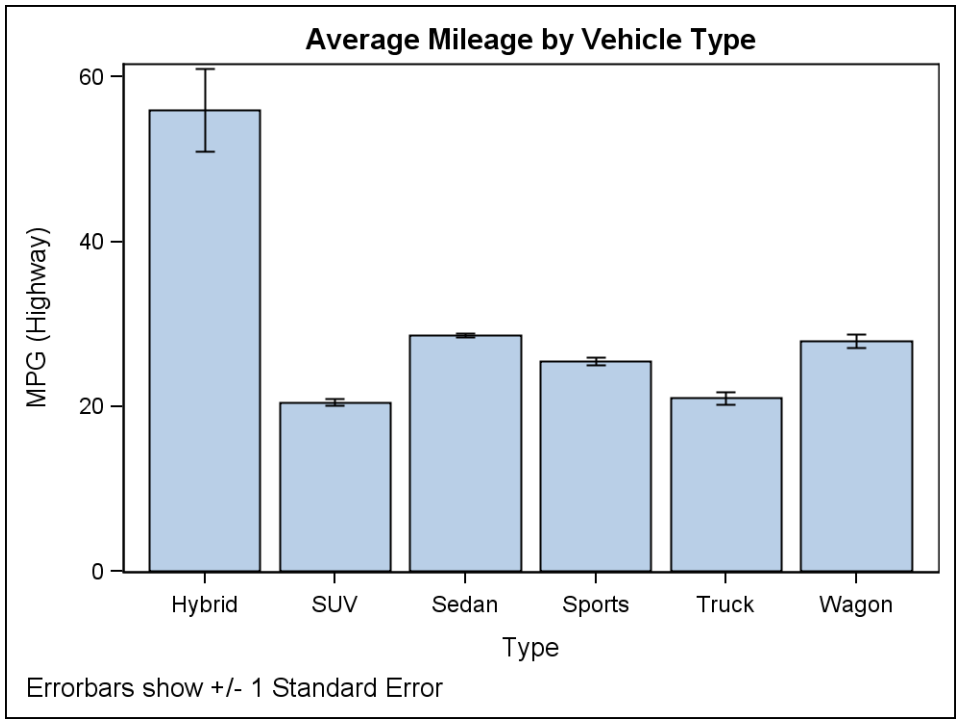
Creates a bar chart specified by pre-summarized data.

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 180:



Example Program

```
proc template;
  define statgraph barchartparm;
    begingraph;
      entrytitle "Average Mileage by Vehicle Type";
      entryfootnote halign=left
        "Error bars 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 category column is character, the bars in the chart appear in the order in which the category values are present in the input data. If the category column is numeric, 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.
BASELINEINTERCEPT	Specifies the response axis intercept for the baseline.
CLUSTERWIDTH	Specifies the width of the group clusters as a fraction of the midpoint spacing.
CONNECTATTRS	Specifies the properties of the line connecting bars.
DATALABEL	Specifies the label to appear at the end of each bar.
DATALABELATTRS	Specifies the color and font attributes of the labels that are specified in the DATALABEL= option.
DATASKIN	Enhances the visual appearance of the filled 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.

Statement Option	Description
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.
FILLPATTERNATTRS	Specifies the appearance of the pattern-filled bar area.
GROUP	Creates a separate segment within each bar for each unique group value of the specified column.
GROUPDISPLAY	Specifies how to display grouped bars.
GROUPORDER	Specifies the order of the grouped bars for each category value.
INCLUDEMISSINGGROUP	Specifies whether missing values of the group variable are included in the plot.
INDEX	Specifies indices for mapping line attributes (color and pattern) to one of the GraphData1–GranphDataN style elements.
LEGENDLABEL	Specifies the label for 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 user-defined roles that can be used to display information in the tooltips.
TARGET	Specifies the target value for each bar.
TIP	Specifies the information to display when the cursor is positioned over a bar.
TIPFORMAT	Specifies display formats for information defined by roles.

Statement Option	Description
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: 0.85. By default, the bar width automatically adjusts based on the number of bars to be displayed and the wall width.

Range: 0 .1 (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.

BASELINEINTERCEPT=*number*
specifies the response axis intercept for the baseline.

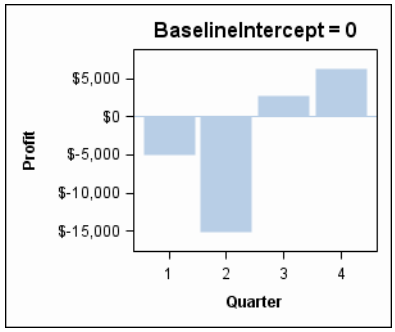
Default: 0

Interaction: Label positions are adjusted to prevent the labels from overlapping.

Interaction: If GROUPDISPLAY=STACKED is specified, this option is ignored and the baseline is not displayed.

Tip: The baseline does not add a tick or a tick value to the axis. To label the baseline, you can use a REFERENCELINE statement to overlay a line with the same X or Y value and use the REFERENCELINE statement's CURVELABEL= option.

Details: When this option is used, the axis range is adjusted to include the baseline, and the baseline is placed at the specified value on the response axis.



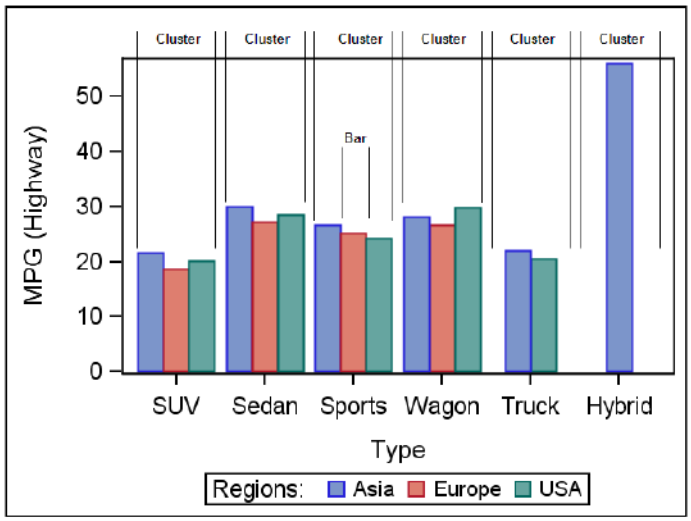
CLUSTERWIDTH= *number*
specifies the width of the group clusters as a fraction of the midpoint spacing on a discrete axis or a fraction of the minimum interval between adjacent data values on an interval axis.

Default: 0.85

Range: 0.1 (narrowest) to 1 (widest)

Interaction: For this option to take effect, the GROUP= option must also be specified, and the GROUPDISPLAY= option must be set to CLUSTER.

Note: When GROUPDISPLAY=CLUSTER, the default BARWIDTH is 1.0.



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 849 for the syntax on using a *style-element* and "Line Options" on page 850 for available *line-options*.

Default: The GraphConnectLine style element.

DATALABEL= *column* | *expression*
specifies the label to appear at the end of each bar.

Default: no bar labels are displayed

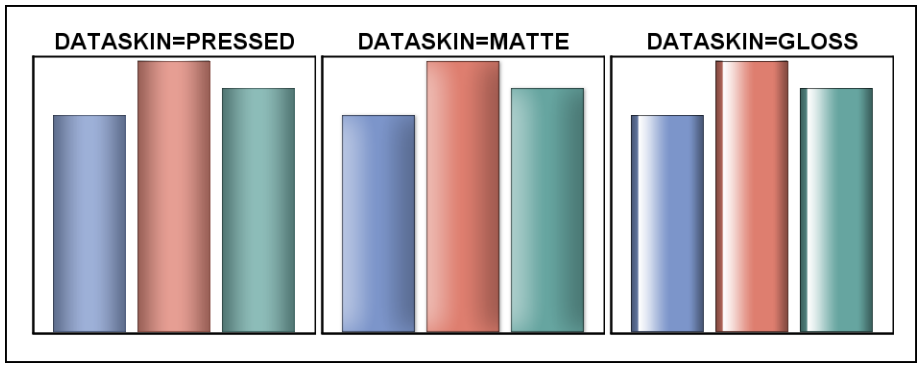
Interaction: If the GROUP= option is used and there are multiple input observations per bar for the GROUP= column, the value for the DATALABEL= column should be the same for each observation that is on the same bar.

DATALABELATTRS=*style-element* | *style-element* (*text-options*) | (*text-options*)
specifies the color and font attributes of the labels that are specified in the DATALABEL= option. See "General Syntax for Attribute Options" on page 849 for the syntax on using a *style-element* and "Text Options" on page 852 for available *text-options*.

Default: The GraphDataText style element.

Interaction: For this option to take effect, the DATALABEL= option must also be used.

DATASKIN=NONE | PRESSED | SHEEN | CRISP | GLOSS | MATTE
enhances the visual appearance of the filled bars.



Default: NONE

Requirement: For this option to have any effect, the fill must be enabled by the ODS style or the DISPLAY= option.

Interaction: The data skin appearance is based on the FILLATTRS= color.

Interaction: When a data skin is applied, all bar outlines are set by the skin, and the OUTLINEATTRS= option is ignored.

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)

Tip: The FILLATTRS= option can be used to set transparency for just the bar fills. You can combine this option with FILLATTRS= to set one transparency for the bar outlines, error bars, and connect lines but a different transparency for the bar fills. Example:

```
datatransparency=0.2 fillattrs=(transparency=0.6)
```

DISCRETEOFFSET=*number*
specifies an amount to offset all bars from the category midpoints.

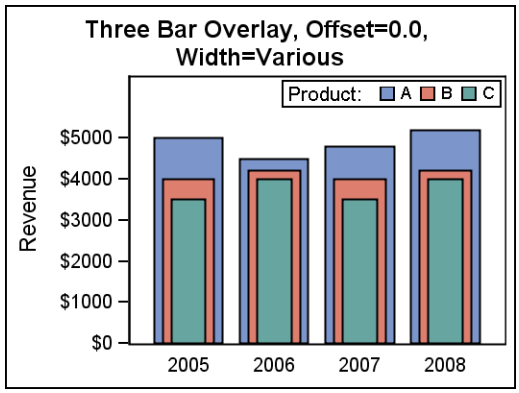
Default: 0 (no offset, all bars are centered on the category midpoints)

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:

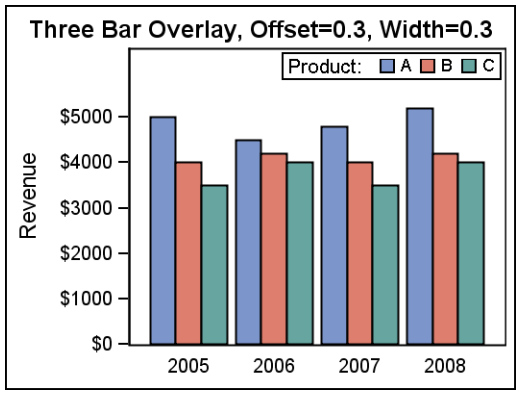
```
layout overlay / cycleattrs=true
    xaxisopts=(display=(tickvalues))
    yaxisopts=(label="Revenue" offsetmax=0.2);

  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:

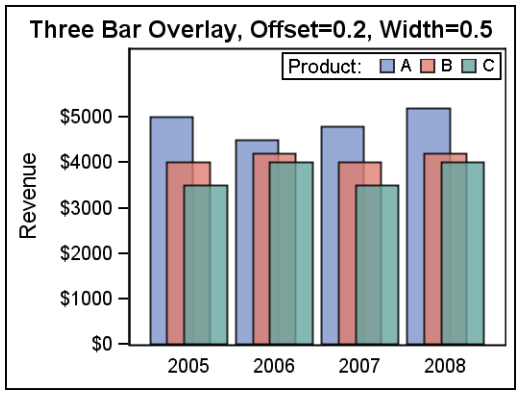
```
layout overlay / cycleattrs=true
    xaxisopts=(display=(tickvalues))
    yaxisopts=(label="Revenue" offsetmax=0.2);

  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 bars with a solid fill.
CONNECT	Displays line segments connecting adjacent midpoints at the end of each bar.
FILLPATTERN	Displays bars with a patterned fill. This setting is used primarily for grouped bar charts that must be rendered in monochrome for use in a journal article. The fill patterns make it easier to distinguish among groups when color is not available.

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= , FILLATTRS= , and FILLPATTERNATTRS= options to control the appearance of the bars. Use CONNECTATTRS= to control the appearance of the connect lines.

Tip: Both FILL and FILLPATTERN can be specified to combine solid fills and pattern fills in the bars.

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 849 for the syntax on using a *style-element* and "Line Options" on page 850 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 and GROUPDISPLAY= STACK, 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 and GROUPDISPLAY= STACK, 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 849 for the syntax on using a *style-element* and "Fill Options" on page 850 for available *fill-options*.

Default:

- For non-grouped data, the GraphDataDefault:Color style reference.
- For grouped data, the GraphData1:Color–GraphDataN:Color style references.

Tip: The DATATRANSPARENCY= option sets the transparency for bar fills, bar outlines, error bars, and connect lines. You can combine this option with DATATRANSPARENCY= to set one transparency for the bar outlines, error bars, and connect lines but a different transparency for the bar fills. Example:

```
datatransparency=0.2 fillattrs=(transparency=0.6)
```

FILLPATTERNATTRS=*style-element* | *style-element* (*fill-options*) | (*fill-pattern-options*)
specifies the appearance of the pattern-filled bar area.

Default: no default

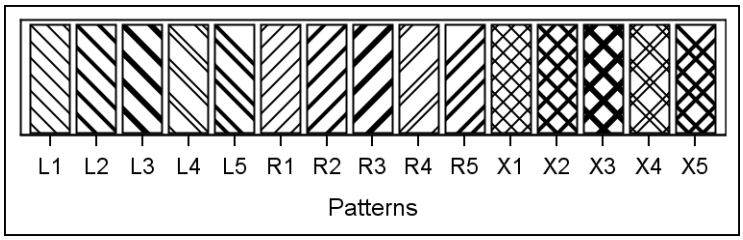
To specify a *style-element*, only one of the elements GraphData1–GraphDataN can be used, subject to the Restriction listed in a moment.

The following *fill-pattern-options* are available:

COLOR = *color* | *style-reference*
specifies a color to use for the bar-fill-pattern lines. With grouped data, this setting has the effect of holding the pattern line color constant across all group values.

PATTERN = *line-pattern*
specifies a line pattern to use for the bar fill.

To specify a *line-pattern*, combine a line-direction prefix (R for right, L for left, and X for cross hatch) with a line-identification number:



With grouped data, this PATTERN= setting has the effect of holding the fill pattern constant across all group values.

Restriction: The only SAS-delivered styles that support fill patterns are JOURNAL2, JOURNAL3, and MONOCHROMEPRINTER. If any other SAS-delivered style is in effect and this option uses a *style-element* in its specification, this option is ignored.

Interaction: For this option to take effect, the DISPLAY= option must include FILLPATTERN among the display options.

GROUP=*column* | *discrete-attr-var* | *expression*
creates a separate bar segment or bar for each unique group value of the specified column.

discrete-attr-var
specifies a discrete attribute variable that is defined in a DISCRETEATTRVAR statement.

Restriction: A discrete attribute variable specification must be a direct reference to the attribute variable. It cannot be set by a dynamic variable.

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 or fill pattern. The fill colors are defined by the GraphData1:Color–GraphDataN:Color style references. The fill patterns are defined by the GraphData1:FillPattern–GraphDataN:FillPattern 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 outline. The outline

colors are defined by the GraphData1:ContrastColor–GraphDataN:ContrastColor style references.

Interaction: The bar display depends on the setting for the GROUPDISPLAY= option. For example, for a vertical bar chart with GROUPDISPLAY=STACK, the bar segments are stacked to form the bar. The height of each segment represents the corresponding group value's proportional contribution to the response value.

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 group values. The INDEX= option can be used to alter the default sequence of colors, fill patterns, and line patterns.

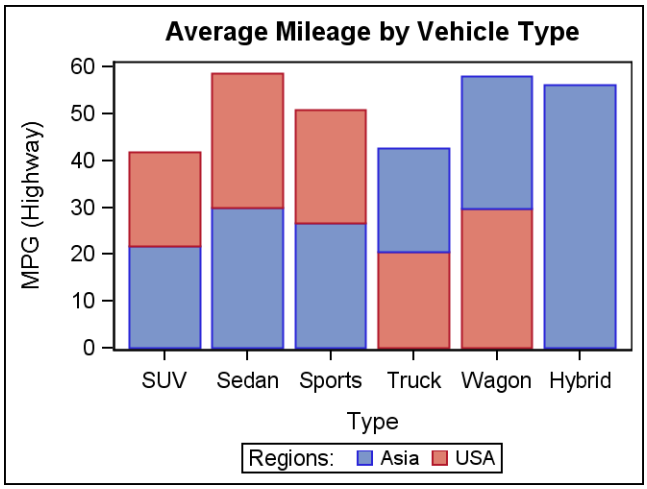
Interaction: The INCLUDEMISSINGGROUP option controls whether missing group values are considered a distinct group value.

Tip: 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 in the OUTLINEATTRS= option could be used to assign the same line pattern to all bar outlines and connect lines.

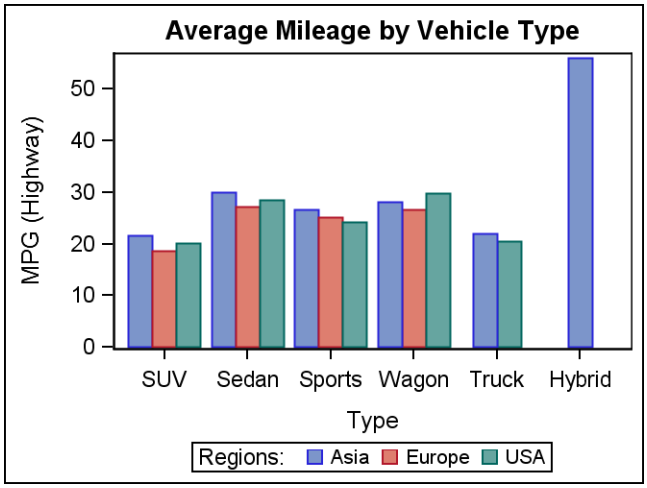
GROUPDISPLAY=STACK | CLUSTER
specifies how to display grouped bars.

Default: STACK

STACK
Displays group values as stacked segments within the category bar.



CLUSTER
Displays group values as separate adjacent bars that replace the single category bar. Each set of group values is centered at the category midpoint on the axis. This example illustrates the clusters and also how groups are displayed when they have an unequal number of unique values.



GROUPORDER=DATA | ASCENDING | DESCENDING
specifies the order of the grouped bars for each category value. It also sets the default order of the groups in the legend.

Default: DATA

DATA
shows each group of bars for each category 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.

Note: Attributes such as color, symbol, and pattern are assigned to each group in the DATA order by default regardless of the GROUPORDER= option setting.

Note: The ASCENDING and DESCENDING settings linguistically sort the group values within each category (or X value) for display position purposes only. The data order of the observations and the visual attributes that are assigned to the group values remain unchanged.

INCLUDEMISSINGGROUP=*boolean*
specifies whether missing values of the group variable are included in the plot.

Default: TRUE

Interaction: For this option to take effect, the GROUP= option must also be specified.

Tip: Unless a discrete attribute map is in effect or the INDEX= option is used, the attributes of the missing group value are determined by the GraphMissing style element except when the MISSING= system option is used to specify a non-default missing character or when a user-defined format is applied to the missing group value. In those cases, the attributes of the missing group value are determined by a GraphData1–GraphDataN style element.

INDEX=*numeric-column* | *expression*
specifies indices for mapping line attributes (color and pattern) to one of the GraphData1–GranphDataN style elements.

Default: no default

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.

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: 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 144.

LEGENDLABEL= *"string"*
specifies a label for the legend item that is associated with this plot.

Default: The Y-variable label. If a label is not defined, the Y-variable name.

Restriction: This option applies only to an associated DISCRETELEGEND statement.

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

Tip: When this option is set to VERTICAL, the category variable appears on the X (or X2) axis and the response variable appears on the Y (or Y2) axis. To set the axis properties for this chart, you should use the appropriate axis options of the layout container.

When this option is set to HORIZONTAL, the category variable appears on the Y (or Y2) axis and the response variable appears on the X (or X2) axis. To set the axis properties for this chart, you should use the appropriate axis options of the layout container.

Tip: If you change the orientation of the bar chart, you should adjust the layout container's axis options appropriately.

OUTLINEATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)
specifies the appearance of the bar outlines. See "General Syntax for Attribute Options" on page 849 for the syntax on using a *style-element* and "Line Options" on page 850 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.

Interaction: If the DATASKIN= option applies a data skin, this option is ignored.

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 562.

ROLENAME=(*role-name-list*)
specifies user-defined roles that can be used to display information in the tooltips.

Default: no user-defined roles

(*role-name-list*)
a blank-separated list of *role-name* = *column* pairs.

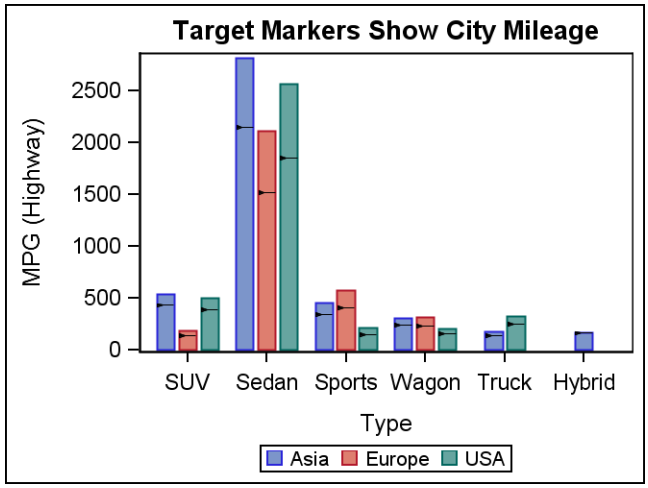
For example, ROLENAME= (TIP1=PCT) assigns the column PCT to the user-defined role TIP1.

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.

TARGET=*numeric-column* | *expression*
specifies the target value for each bar. The visual representation is a triangle with a line at the target value.

```
layout overlay;
  barchartparmm x=type y=mpg_highway / barwidth=.8
    target=mpg_city group=origin groupdisplay=cluster
    name='bar';
  discretelegend 'bar';
endlayout;
```



Default: No targets are displayed.

Interaction: For this option to take effect, the Y= argument must also be used.

Interaction: If the GROUP= option is used and GROUPDISPLAY= STACK, this option is ignored.

Tip: The target color is that of the bar outline.

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. The PCT column is not assigned to any pre-defined bar chart role, so it 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 557.

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 557.

Chapter 18

BIHISTOGRAM3DPARM Statement

Description and Syntax

Creates a three-dimensional 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*

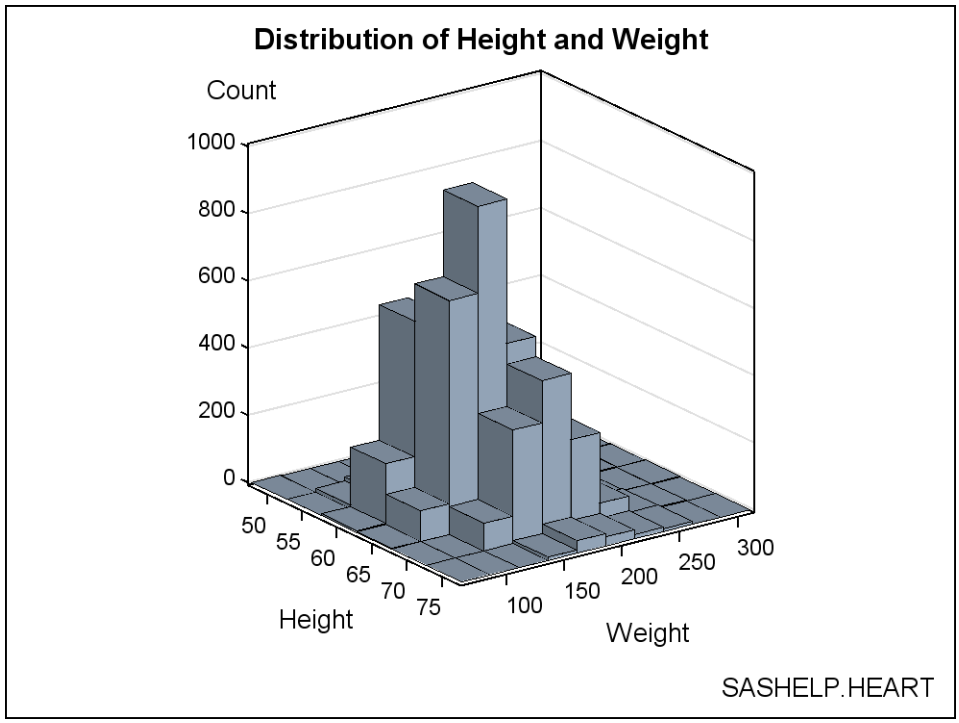
Y = *numeric-column* | *expression*

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 198:



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

Because the BIHISTOGRAM3DPARM statement does not perform a binning computation on the input columns, 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 the KDE procedure (SAS/STAT), which enables you to set the number of bins for X and Y. Alternatively, 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=*nonnegative-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 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 849 for the syntax on using a *style-element* and "Fill Options" on page 850 for available *fill-options*.

Default: The GraphDataDefault style element.

LEGENDLABEL= *"string"*
specifies a label for the legend item that is associated with this plot.

Default: The *string* specified on the NAME= option.

Restriction: This option applies only to an associated DISCRETELEGEND statement.

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 849 for the syntax on using a *style-element* and "Line Options" on page 850 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 562.

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 19

BLOCKPLOT Statement

Description and Syntax

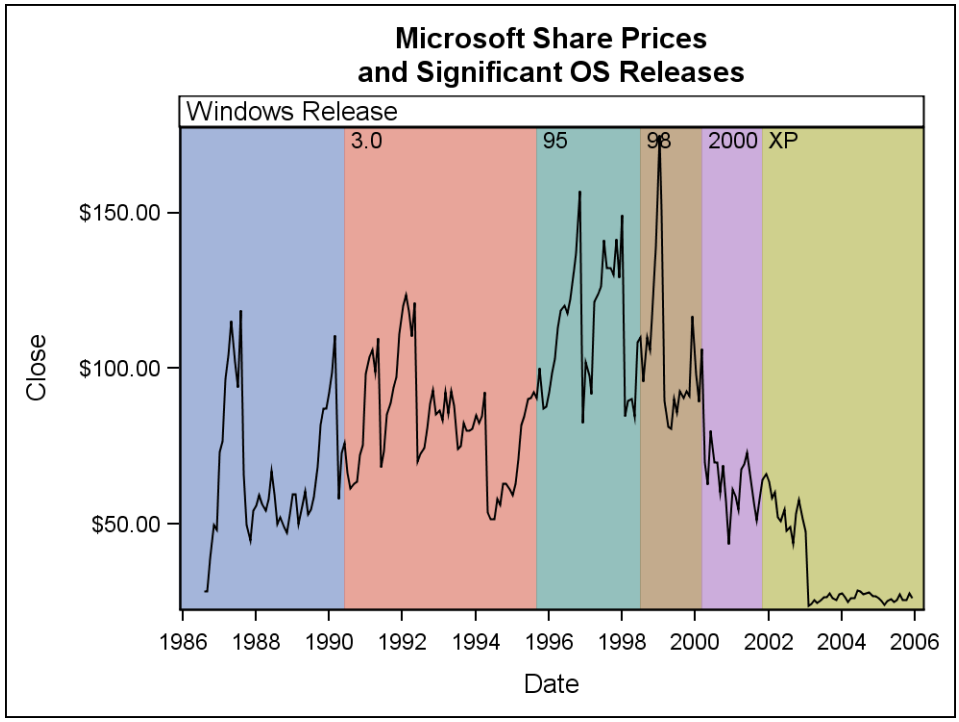
Creates one or more strips of rectangular blocks containing text values. The width of each block corresponds to specified numeric intervals.

BLOCKPLOT X = *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
;
proc sort data=sashelp.stocks(keep=date stock close)
  out=MSstock;
  where stock="Microsoft";
  by date;

data events;
  merge MSstock MSevents;
  by date;
run;
```

In the combined input data set, notice that the RELEASE value is missing for each observation until the first defined "event" occurs on **01JUN90**. The RELEASE value is missing again for subsequent observations and will not have another value until the next event occurs on **01SEP95**, when the MSEVENTS data supplies the value **95**.

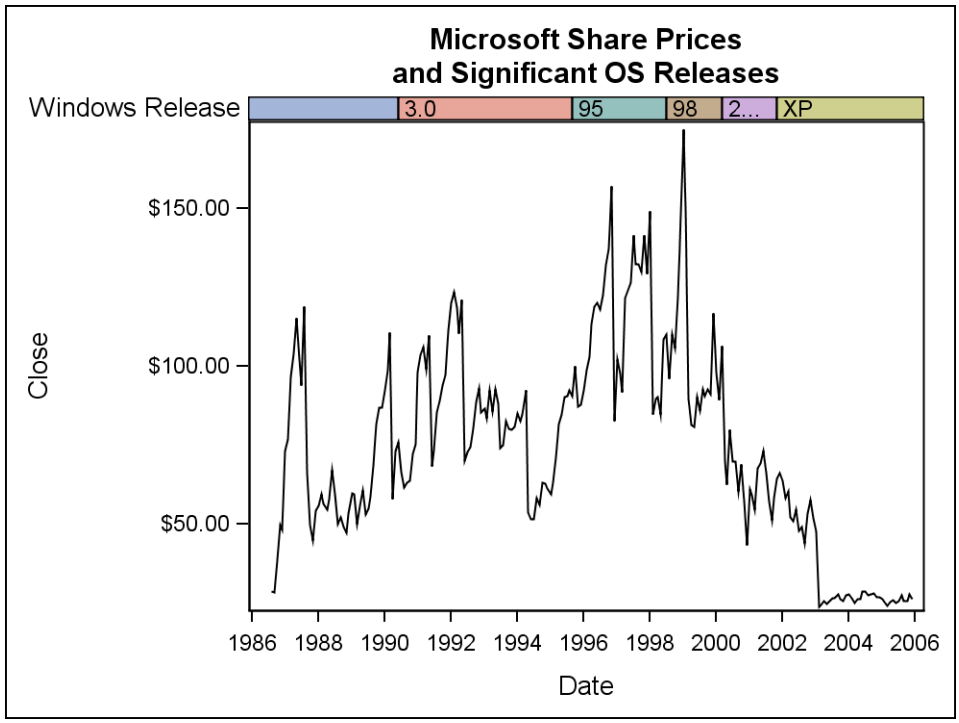
```
Obs    Date      Close   Release
...
45    02APR90   $58.00
46    01MAY90   $73.00
47    01JUN90   $76.00     3.0
48    02JUL90   $66.50
49    01AUG90   $61.50
50    04SEP90   $63.00
...
```

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. The BLOCKPLOT statement sets EXTENDBLOCKONMISSING= TRUE so that missing values in the data revert to the previous nonmissing value in the block plot. Thus, in the block plot, values are missing until **01JUN90**, when the value changes from missing to **3.0**. The block plot retains that **3.0** value for subsequent observations until the next nonmissing value replaces it (in this case, the value**95** on **01SEP95**).

```
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)
           extendblockonmissing=true ;
        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)
          extendblockonmissing=true;
        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.
EXTENDBLOCKONMISSING	Specifies whether a missing value in the BLOCK column starts a new block or reverts to the previous nonmissing value.
FILLATTRS	Specifies the appearance of the block fills.
FILLTYPE	Specifies how the blocks are filled.
INCLUDEMISSINGCLASS	Specifies whether missing values of the class variable are included in the plot.
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 the value text within the blocks.

Statement Option	Description
VALUEVALIGN	Specifies the vertical alignment of the value text within the 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 849 for the syntax on using a *style-element* and "Fill Options" on page 850 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.

Tip: The DATATRANSPARENCY= option sets the transparency for the block fills and the outlines. You can combine this option with DATATRANSPARENCY= to set one transparency for the outlines but a different transparency for the alternate block fills. Example:

```
datatransparency=0.2 altfillattrs=(transparency=0.6)
```

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)

Tip: The ALTFILLATTRS= option can be used to set transparency for just the alternate block fills. The FILLATTRS= option can be used to specify transparency for the block fills. You can combine this option with ALTFILLATTRS= and with FILLATTRS= to set one transparency for the outlines but a different transparency for the block fills. Example:

```
datatransparency=0.2
altfillattrs=(transparency=0.6) fillattrs=(transparency=0.6)
```

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 the external block label or labels

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.

EXTENDBLOCKONMISSING=*boolean*
specifies whether a missing value in the BLOCK column starts a new block or reverts to the previous nonmissing value.

Default: FALSE

Tip: When EXTENDBLOCKONMISSING = TRUE, you can set up the input data for the BLOCK= column with nonmissing values where you expect the blocks to

change and leave the remaining block values missing. For an example, see "BlockPlot Overlaid with SeriesPlot" on page 204.

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 849 for the syntax on using a *style-element* and "Fill Options" on page 850 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.

Tip: The DATATRANSPARENCY= option sets the transparency for the block fills and the outlines. You can combine this option with DATATRANSPARENCY= to set one transparency for the outlines but a different transparency for the block fills. Example:

```
datatransparency=0.2 fillattrs=(transparency=0.6)
```

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.

INCLUDEMISSINGCLASS=*boolean*
specifies whether missing values of the class variable are included in the plot.

Default: TRUE

Interaction: The CLASS=class option must be specified for this option to have any effect.

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 849 for the syntax on using a *style-element* and "Text Options" on page 852 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 849 for the syntax on using a *style-element* and "Line Options" on page 850 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 849 for the syntax on using a *style-element* and "Text Options" on page 852 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 value 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. In addition, 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 value text within the blocks.

Default: CENTER

Interaction: For this option to have any effect, the DISPLAY= option must include VALUE.

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 557

Chapter 20

BOXPLOT Statement

Description and Syntax

Creates box-and-whisker plot(s) computed from input data.

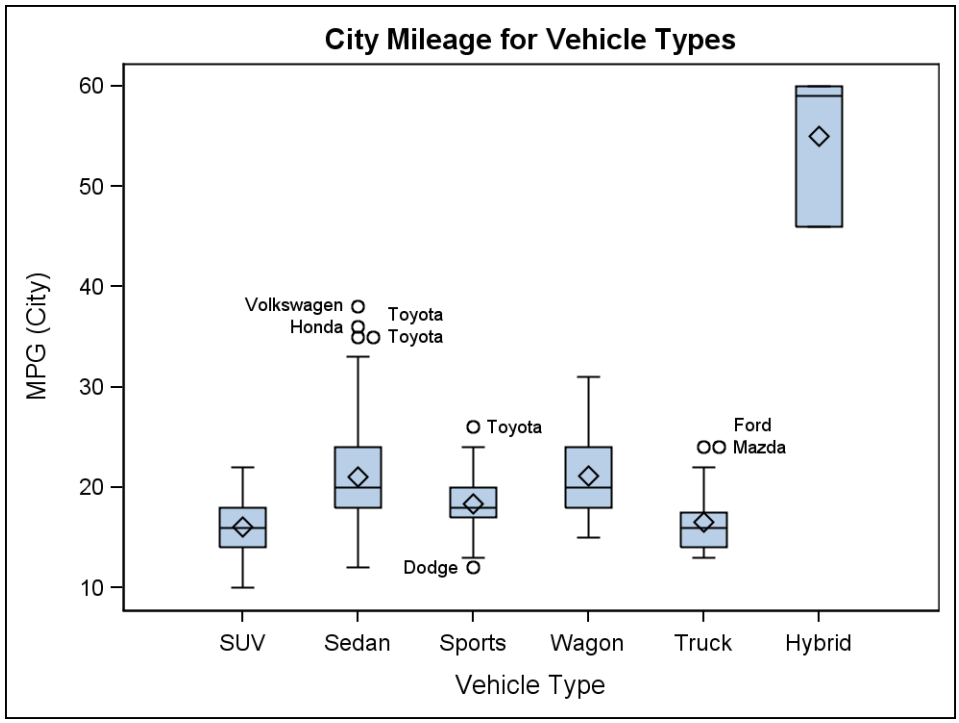
BOXPLOT Y = *numeric-column* | *expression* </*option(s)*>;

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 214:



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

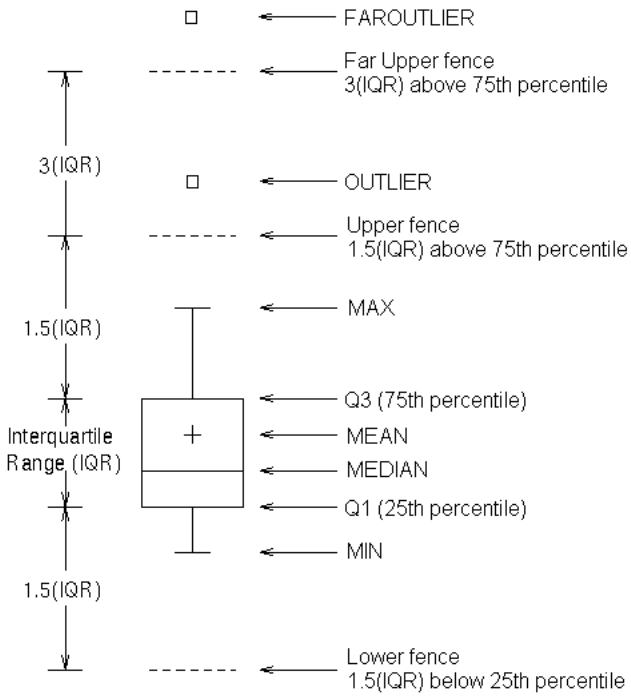
The BOXPLOT statement displays a single box if given just a Y argument. It displays multiple boxes if given both Y and X arguments and X has more than one unique value.

By default for numeric or character columns, the category (X) axis is TYPE=DISCRETE. You can override the default and set the TYPE= to LINEAR or TIME in the parent layout, provided that the X column is numeric. The axis for the analysis (Y) column is always LINEAR. When the X axis is LINEAR, you must use the INTERVALBOXWIDTH= option to specify the box width.

When ORIENT= VERTICAL, the X (or X2) axis is used for the X column and the Y (or Y2) axis is used for the Y column. When ORIENT=HORIZONTAL, the X (or X2) axis is used for the Y column and the Y (or Y2) axis is used for the X column.

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 ($\pm 1.5 IQR$). 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 the 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.
CAPSHAPE	Specifies the shape at the ends of the whiskers.
CLUSTERWIDTH	Specifies the width of the group clusters as a fraction of the midpoint spacing.
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.
DISPLAYSTATS	Specifies the statistics to be displayed for each box.
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.
GROUP	Creates a box plot for each unique group value of the specified column.
GROUPDISPLAY	Specifies how to display the boxes that represent group values for the coordinate pairs.

Statement Option	Description
GROUPORDER	Specifies the ordering of the groups within a category.
INCLUDEMISSINGGROUP	Specifies whether missing values of the group variable are included in the plot.
INDEX	Specifies indices for mapping line attributes (color and pattern) to one of the GraphData1 - GranphDataN style elements.
INTERVALBOXWIDTH	Specifies the box width when an interval category (X) column is specified.
LABELFAR	Specifies whether all outliers or only far outliers are labeled.
LEGENDLABEL	Specifies the label for 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.
OUTLIERTIP	Specifies the information to display when the cursor is positioned over an outlier.
OUTLINEATTRS	Specifies the properties of the box outlines.
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.
TIP	Specifies the information to display when the cursor is positioned over a box or whisker in the box plot.
TIPFORMAT	Specifies display formats for the information that is defined by the tooltip roles.

Statement Option	Description
TIPLABEL	Specifies display labels for the information that is defined by the tooltip roles.
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: 0.4 for non-grouped data, 0.6 for grouped data

Range: 0 (narrowest) to 1 (widest)

Interaction: For grouped box plots with a discrete X (category) axis, the box width is a percentage of the CLUSTERWIDTH .

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.

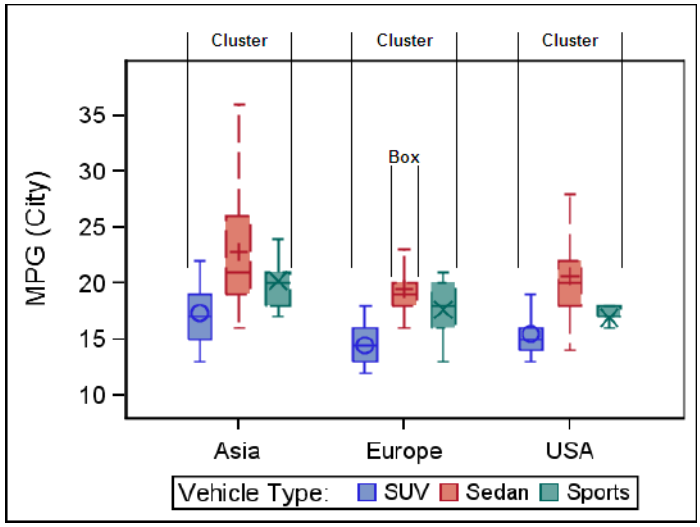
CLUSTERWIDTH= *number*
specifies the width of the group clusters as a fraction of the midpoint spacing.

Default: 0.7

Range: 0.1 (narrowest) to 1 (widest)

Interaction: For this option to take effect, the GROUP= option must also be specified, and the GROUPDISPLAY= option must be set to CLUSTER.

Note: When the X axis is an interval axis, the cluster width is a fraction of the smallest data interval.



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 *display-option* 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 849 for the syntax on using a *style-element* and "Line Options" on page 850 for available *line-options*.

Default: The GraphConnectLine style element.

Interaction: If there is only one box, this option is ignored.

Interaction: If the DISPLAY= option does not include CONNECT, or if the GROUP= option is used, this option is ignored.

DATALABEL=*column*
specifies the labels of the outliers. Either a numeric or a character column can be used.

Default: no data labels are displayed

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 849 for the syntax on using a *style-element* and "Text Options" on page 852 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).

Tip: The FILLATTRS= option can be used to set transparency for just the interior fill area of the boxes. You can combine this option with FILLATTRS= to set one transparency for the box outlines and the whiskers, mean, median, caps, and connect lines, but a different transparency for the box fills. Example:

```
datatransparency=0.2 fillattrs=(transparency=0.6)
```

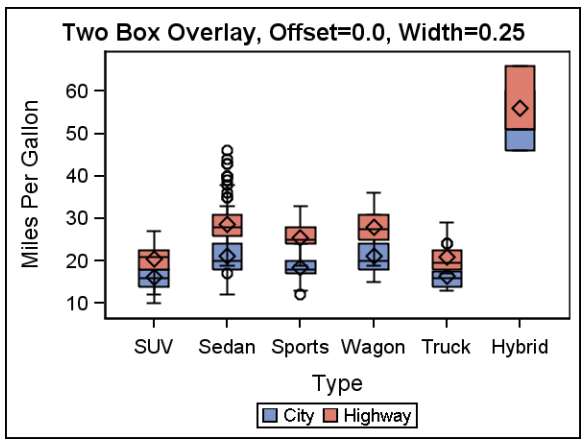
DISCRETEOFFSET=*number*
specifies an amount to offset all boxes from the discrete X ticks.

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.)

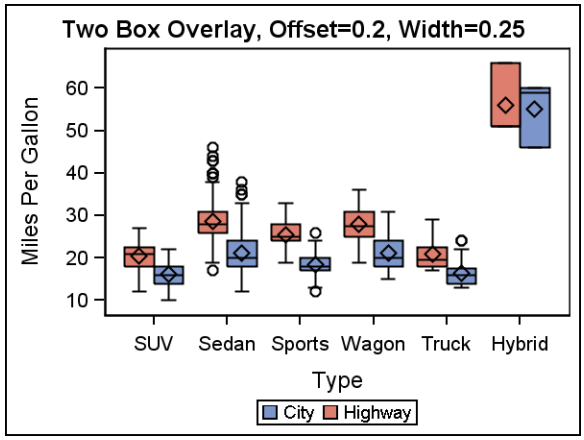
Details: This feature is useful for graphing multiple response variables side by side on a common axis. By default within an overlay-type layout, if multiple BOXPLOT statements are used with different analysis variables, the boxes for matching X values are centered on the ticks. Depending on the data, 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

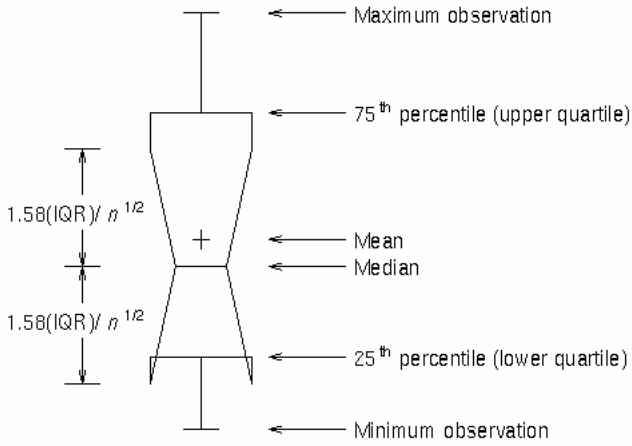
To control the appearance of these features, use the CONNECTATTRS= , FILLATTRS= , MEANATTRS= , MEDIANATTRS= , OUTLIERATTRS= , and WHISKERATTRS= options. The WHISKERATTRS= option controls affects both CAPS and WHISKERS.

Tip: Regardless of which *display-options* are being displayed, this option does not affect the axis range.

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.



DISPLAYSTATS=NONE | STANDARD | ALL | *(statistics-list)*
specifies the statistics to be displayed for each box.

Default: NONE

NONE
does not display any statistics

STANDARD
displays N, MEAN, and STD

ALL
displays all available statistics (see the *statistics-list*)

(statistics-list)
specifies a list of features to be displayed. The list must be enclosed in parentheses and can include any of the following:

MAX	maximum data value below the box upper fence
MEAN	mean data value for box
MEDIAN	median data value for box
MIN	minimum data value above the box upper fence
N	number of observations for box
Q1	lower quartile (25th percentile) for box
Q3	upper quartile (75th percentile) for box
STD	standard deviation of the data for box
RANGE	range of the data (MAX–MIN)
IQR	interquartile range (Q3–Q1)

Restriction: This option is ignored if ORIENT= HORIZONTAL

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 213 .)

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 849 for the syntax on using a *style-element* and "Fill Options" on page 850 for available *fill-options*.

Default:

- For non-grouped data, the GraphDataDefault:Color style reference.
- For grouped data, the Color attribute of 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.

Tip: The DATATRANSPARENCY option sets the transparency for the box outlines, box fill, whiskers, mean, median, caps, connect lines, and outliers. You can combine this option with DATATRANSPARENCY= to set one transparency for the box outlines and the whiskers, mean, median, caps, and connect lines, but a different transparency for the box fills. Example:

```
datatransparency=0.2 fillattrs=(transparency=0.6)
```

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.

GROUP=*column* | *discrete-attr-var* | *expression*
creates a box plot for each unique group value of the specified column.

discrete-attr-var
specifies a discrete attribute variable that is defined in a DISCRETEATTRVAR statement.

Restriction: A discrete attribute variable specification must be a direct reference to the attribute variable. It cannot be set by a dynamic variable.

For example, the SASHELP.CARS data that is used in the "Example Program" on page 214 contains a column named ORIGIN, which identifies the region that produces each car. This column could be used in the BOXPLOT statement to group the box plots in the display (see the GROUPDISPLAY= option to see the output for the grouped boxes):

```
layout overlay / xaxisopts=(display=(line ticks tickvalues));
  boxplot y=mpg_city x=cylinders / name="b"
    datalabel=make spread=true
    display=(caps fill mean median)
    group=origin ;
  discretelegend "b" / title="Vehicle Type: ";
endlayout;
```

Default:

- Each distinct group value is represented in the graph by a different box outline color. The outline colors are defined by the GraphData1:ContrastColor - GraphDataN:ContrastColor style references.
- If the MEDIAN, CAPS, or NOTCHES are enabled by the DISPLAY= option, each of these features uses the same color as the box outline. Line styles do not change by group value.
- If the MEAN or OUTLIERS are enabled by the DISPLAY= option, each distinct group value is represented by a different marker. The markers are defined by the MarkerSymbol and ContrastColor attributes of the GraphData1 - GraphDataN style elements. A marker is used for both MEAN and OUTLIERS, if displayed.
- If box fills are enabled by the ODS style or by the DISPLAY= option, each distinct group value is represented in the graph by a different fill color. The fill colors are defined by the GraphData1:Color - GraphDataN:Color style references.

Interaction: The box plot display depends on the setting for the GROUPDISPLAY= option.

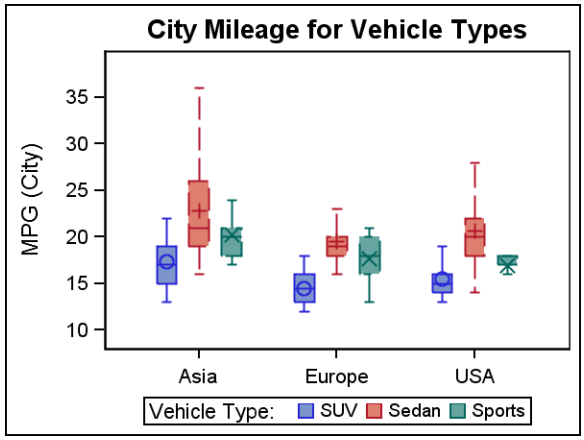
Interaction: Connect lines are not drawn for grouped data.

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 group values. The INDEX= option can be used to alter the default sequence of colors and markers.

Tip: 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 box outlines, but the PATTERN= setting in the OUTLIERATTRS= option could be used to assign the same line pattern to all box outlines and connect lines.

Interaction: The INCLUDEMISSINGGROUP option controls whether missing group values are considered a distinct group value.

GROUPDISPLAY=OVERLAY | CLUSTER
specifies how to display the boxes that represent group values for the coordinate pairs. The following example shows a box plot with GROUPDISPLAY=CLUSTER:



Default: OVERLAY

OVERLAY
draws boxes for a given group value at the exact coordinate. Depending on the data, boxes at a given coordinate might overlap.

CLUSTER
draws boxes for a given group value adjacent to each other.

Interaction: This option is ignored unless GROUP= is specified.

Tip: Use the CLUSTERWIDTH= option to control the width of the clusters when CLUSTER is in effect.

GROUPORDER=DATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category. It also sets the default order of the groups in the legend.

Default: DATA

DATA
shows each group within a category in data order of the group column.

ASCENDING
shows each group within a category in ascending order of the group column.

DESCENDING
shows each group within a category in descending order of the group column.

Interaction: This option is ignored unless GROUP= is specified.

Note: Attributes such as color, symbol, and pattern are assigned to each group in the DATA order by default regardless of the GROUPORDER= option setting.

Note: The ASCENDING and DESCENDING settings linguistically sort the group values within each category (or X value) for display position purposes only. The data order of the observations and the visual attributes that are assigned to the group values remain unchanged.

INCLUDEMISSINGGROUP=*boolean*
specifies whether missing values of the group variable are included in the plot.

Default: TRUE

Interaction: For this option to take effect, the GROUP= option must also be specified.

Tip: Unless a discrete attribute map is in effect or the INDEX= option is used, the attributes of the missing group value are determined by the GraphMissing style element except when the MISSING= system option is used to specify a non-default

missing character or when a user-defined format is applied to the missing group value. In those cases, the attributes of the missing group value are determined by a GraphData1–GraphDataN style element.

INDEX=*numeric-column* | *expression*
specifies indices for mapping line attributes (color and pattern) to one of the GraphData1 - GranphDataN style elements.

Default: no default

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.

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: 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 144.

INTERVALBOXWIDTH= AUTO | *dimension*
specifies the box width when an interval category (X) column is specified.

Default: AUTO

AUTO	uses 85% of the smallest interval between any two boxes for the given plot.
dimension	sets the box width to the specified value.

Restriction: The axis type for the category axis (X by default) must be LINEAR, and the X column must be numeric.

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 213.

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 the legend item that is associated with this plot.

Default: The *string* specified on the NAME= option.

Restriction: This option applies only to an associated DISCRETELEGEND statement.

Interaction: If the GROUP= option is specified, this option is ignored.

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 849 for the syntax on using a *style-element* and "Marker Options" on page 851 for available *marker-options*.

Default:

- For non-grouped data, GraphBoxMean style element.
- For grouped data, the MarkerSymbol, Markersize, and ContrastColor attributes of the GraphData1 – GraphDataN style elements.

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 849 for the syntax on using a *style-element* and "Line Options" on page 850 for available *line-options*.

Default:

- For non-grouped data, the GraphBoxMedian style element.
- For grouped data, the LineStyle and LineThickness attributes of the GraphBoxMedian style element, and the ContrastColor attribute of the GraphData1 – GraphDataN style elements.

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 849 for the syntax on using a *style-element* and "Marker Options" on page 851 for available *marker-options*.

Default:

- For non-grouped data, GraphOutlier style element.
- For grouped data, the MarkerSymbol, Markersize, and ContrastColor attributes of the GraphData1 – GraphDataN style elements.

Interaction: This option is ignored if EXTREME= TRUE or the DISPLAY= option does not display the outliers.

OUTLIERTIP=(*role-list*)
specifies the information to display when the cursor is positioned over an outlier. If this option is used, it replaces all of the information that is displayed by default.

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 BOXPLOT roles. BOXPLOT roles for OUTLIERTIP include X, Y, STAT, and DATALABEL.

Note: In the tooltip, the STAT role displays the text "outlier" or "far outlier" as applicable.

The following example displays tooltips only for the column that is assigned to the X role:

```
OUTLIERTIP=(X)
```

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 OUTLIERTIP variables can be controlled with the TIPLABEL= and TIPFORMAT= options.

See Also: The TIP= option for specifying the information to display when the cursor is positioned over a box or whisker in the box plot.

OUTLINEATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)
specifies the appearance of the box outline. See "General Syntax for Attribute Options" on page 849 for the syntax on using a *style-element* and "Line Options" on page 850 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 tth 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	$y = x_j$ if $g < \frac{1}{2}$ $y = x_j$ if $g = \frac{1}{2}$ and j is even $y = x_{j+1}$ if $g = \frac{1}{2}$ and j is odd $y = x_{j+1}$ if $g > \frac{1}{2}$
3	Empirical distribution function	$y = x_j$ if $g = 0$ $y = x_{j+1}$ 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	$y = \frac{1}{2}(x_j + x_{j+1})$ if $g = 0$ $y = x_{j+1}$ 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 562.

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.

TIP=(*role-list*)
specifies the information to display when the cursor is positioned over a box or whisker in the box plot. If this option is used, it replaces all of the information that is displayed by default.

Default: The columns assigned to these roles are automatically included in the tooltip information: X, N, STD, MIN, MAX, Q1, Q3, MEAN, and MEDIAN.

(*role-list*)
an ordered, blank-separated list of unique BOXPLOT roles. BOXPLOT roles for TIP include X, N, STD, MIN, MAX, MEAN, MEDIAN, Q1, Q3 and BOXWIDTH.

The following example displays tooltips only for the columns that are assigned to the roles X and Y

```
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.

Tip: Statistics such as N, MIN, MAX are special roles. They are not column-based like the X role.

See Also: The OUTLIERTIP= option for specifying the information to display when the cursor is positioned over an outlier.

TIPFORMAT=(*role-format-list*)
specifies display formats for the information that is defined by the tooltip roles.

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=(X MEAN MIN MAX) TIPFORMAT=(X=4. MEAN=4.)
```

Requirement: This option provides a way to control the formats of columns that appear in tooltips. Only the roles that appear in the OUTLIERTIP= or TIP= options are used.

Interaction: For this option to have any effect, a column must be assigned to each of the specified roles.

TIPLABEL=(*role-label-list*)
specifies display labels for the information that is defined by the tooltip roles.

Default: The column label or column name of the variable that is assigned to the role.

(*role-label-list*)
a list of *role-name* = *"string"* pairs separated by blanks.

```
TIP=(X MEAN MIN MAX) OUTLIERTIP=(Y)
TIPLABEL=(X="Box")
```

Requirement: This option provides a way to control the labels of columns that appear in tooltips. Only the roles that appear in the OUTLIERTIP= or TIP= options are used.

Interaction: For this option to have any effect, a column must be assigned to each of the specified roles.

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 849 for the syntax on using a *style-element* and "Line Options" on page 850 for available *line-options*.

Default:

- For non-grouped data, the GraphBoxWhisker style element.
- For grouped data, the LineStyle and LineThickness attributes of the GraphBoxWhisker style element, and the ContrastColor attribute of the GraphData1 – GraphDataN style elements.

Restriction: The caps are always drawn with a solid line.

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 557.

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 557.

Chapter 21

BOXPLOTPARM Statement

Description and Syntax

Creates side-by-side box plots specified by parameters.

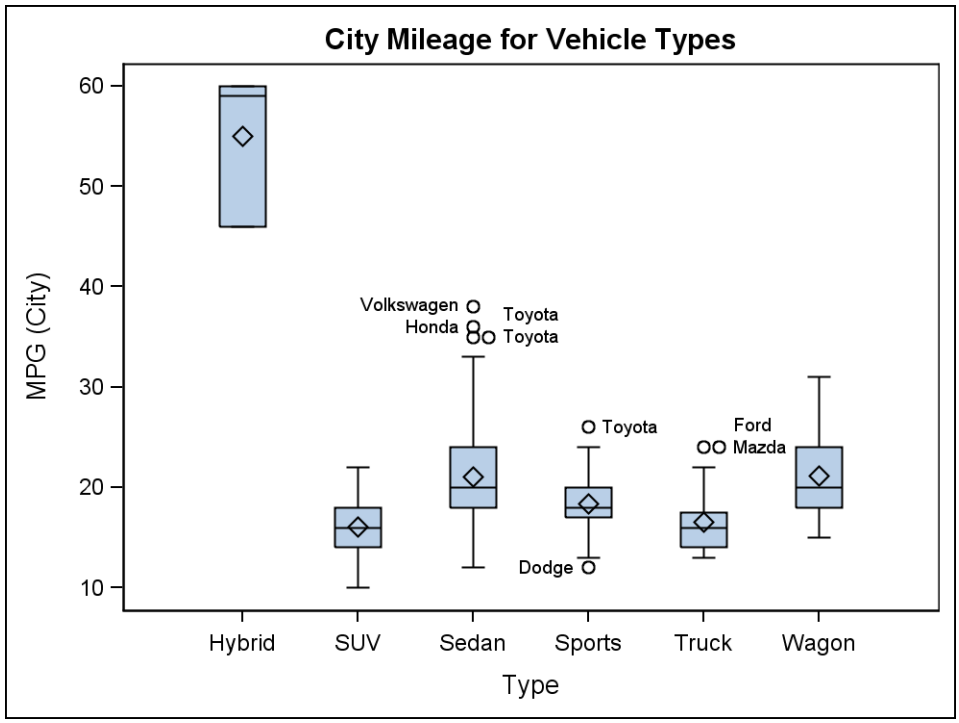
BOXPLOTPARM Y = *numeric-column* | *expression*
STAT = *string-column* </option(s)>;

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 234:



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 4, "Computing Input Data for BOXPLOTPARM," on page 859 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 compute various statistics for the box plot, such as the mean, quartiles, location of fences, outlier definition, and so on. See Appendix 4, "Computing Input Data for BOXPLOTPARM," on page 859 for examples of such computations using PROC SUMMARY and multiple DATA steps.

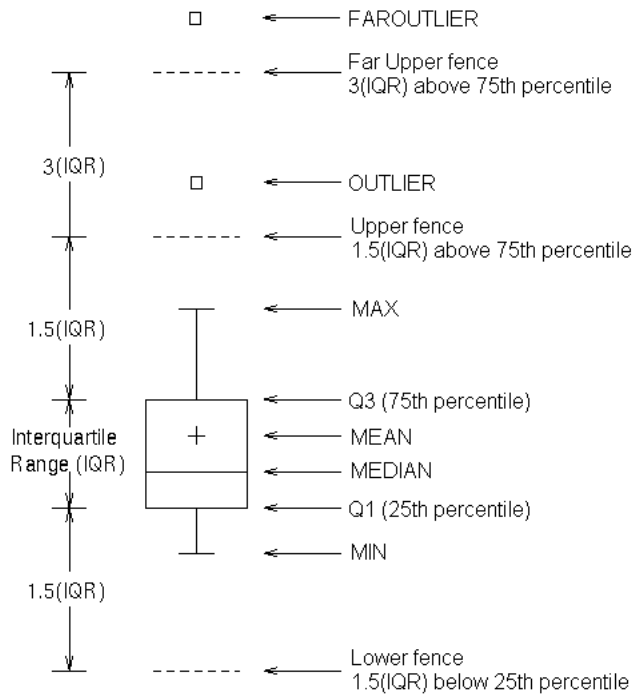
The BOXPLOTPARM statement displays a single box if given just Y and a STAT argument. It displays multiple boxes if given both Y and X and a STAT argument and X has more than one unique value.

By default for numeric or character columns, the category (X) axis is TYPE=DISCRETE. You can override the default and specify TYPE=LINEAR in the parent layout, provided that the X column is numeric. The axis for the analysis (Y) column is always LINEAR. When the X axis is LINEAR, you can use the INTERVALBOXWIDTH= option to specify the box width.

When ORIENT= VERTICAL, the X (or X2) axis is used for the X column and the Y (or Y2) axis is used for the Y column. When ORIENT=HORIZONTAL, the X (or X2) axis is used for the Y column and the Y (or Y2) axis is used for the X column.

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 the EXTREME= option.)

Input Data Requirements for the BOXPLOTPARM Statement

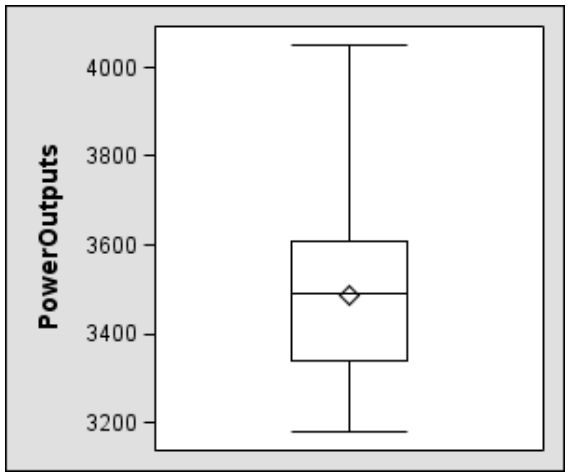
At a minimum, valid data for the BOXPLOTPARM statement must provide a numeric column (Y=) that contains calculated statistics for an analysis, and a string column (STAT=) that identifies each statistic. The Y column must contain nonmissing values for the Q1 (25th percentile) and Q3 (75th percentile) 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;
```



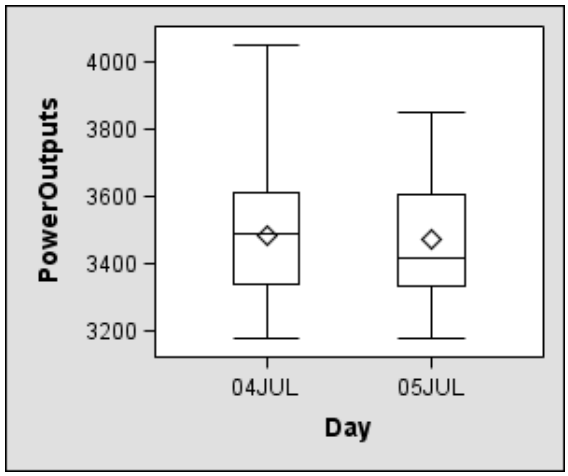
If the data contain statistics for multiple days of the process, a third column in the data must be present to identify the days that the statistics were recorded. For example, the following data show the statistics that are calculated for two days of this process:

Day	PowerOutputs	Statistic
04JUL	3180.00	MIN
04JUL	3340.00	Q1
04JUL	3487.40	MEAN
04JUL	3490.00	MEDIAN
04JUL	3610.00	Q3

Day	PowerOutputs	Statistic
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 4, “Computing Input Data for BOXPLOTPARM,” on page 859 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 (the **Requirement** paragraph that follows this list of STAT= values shows the data restrictions that apply to the STAT= specifications):

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.

BOXWIDTH
width of a box as a ratio of the maximum possible width. The range of values is 0 (narrowest) to 1 (widest). The default is 0.4. If the Y value corresponding to BOXWIDTH is in range, it overrides the setting that is specified in the BOXWIDTH= option.

Requirement: Nonmissing Y values for STAT observations of Q1 and Q3 are required for a box to be drawn. Other STAT values can be omitted or have missing Y values. The STAT values, if present, must conform to the following rules for a box 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.
CLUSTERWIDTH	Specifies the width of the group clusters as a fraction of the midpoint spacing.
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.
DISPLAYSTATS	Specifies the statistics to be displayed for each box.
EXTREME	Specifies whether the whiskers can extend beyond the fences.
FILLATTRS	Specifies the appearance of the interior fill area of the boxes.
GROUP	Creates a box plot for each unique group value of the specified column.
GROUPDISPLAY	Specifies how to display the boxes that represent group values for the coordinate pairs.

Statement Option	Description
GROUPORDER	Specifies the ordering of the grouped boxes for each category value.
INCLUDEMISSINGGROUP	Specifies whether missing values of the group variable are included in the plot.
INDEX	Specifies indices for mapping line attributes (color and pattern) to one of the GraphData1–GranphDataN style elements.
INTERVALBOXWIDTH	Specifies the box width when an interval category (X) column is specified.
LABELFAR	Specifies whether all outliers or only far outliers are labeled.
LEGENDLABEL	Specifies the label for 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.
OUTLIERTIP	Specifies the information to display when the cursor is positioned over an outlier.
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.
ROLENAME	Specifies user-defined roles that can be used to display information in the tooltips.
SPREAD	Specifies whether outliers with the same value are spread out to avoid overlap.
TIP	Specifies the information to display when the cursor is positioned over a box or whisker in the box plot.
TIPFORMAT	Specifies display formats for the information that is defined by the tooltip roles.

Statement Option	Description
TIPLABEL	Specifies display labels for the information that is defined by the tooltip roles.
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:0.4 for non-grouped data, 0.6 for grouped data

Range: 0 (narrowest) to 1 (widest)

Interaction: This setting is overridden by the Y value whenever STAT= BOXWIDTH and the corresponding Y value is in range.

Interaction: For grouped box plots with a discrete X (category) axis, the box width is a percentage of the CLUSTERWIDTH .

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.

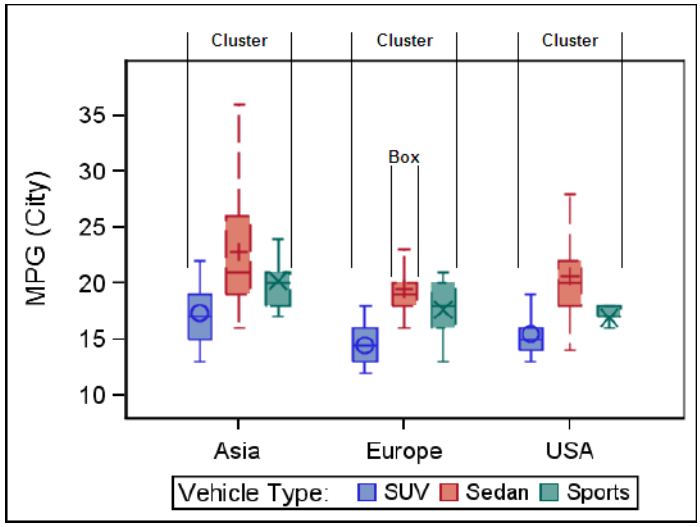
Interaction: The DISPLAY= option must include CAPS for cap lines to be shown.

CLUSTERWIDTH= *number*
specifies the width of the group clusters as a fraction of the midpoint spacing on a discrete axis or a fraction of the minimum interval between adjacent data values on an interval axis.

Default: 0.7

Range: 0.1 (narrowest) to 1 (widest)

Interaction: For this option to take effect, the GROUP= option must also be specified, and the GROUPDISPLAY= option must be set to CLUSTER.



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 *display-option* 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 849 for the syntax on using a *style-element* and "Line Options" on page 850 for available *line-options*.

Default: The GraphConnectLine style element.

Interaction: If there is only one box, this option is ignored.

Interaction: If the DISPLAY= option does not include CONNECT, or if the GROUP= option is used, this option is ignored.

DATALABEL=*column*
specifies the labels of the values that are identified as outlier or far outlier by the STAT= column. Either a numeric or a character column can be used.

Default: no data labels are displayed

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 849 for the syntax on using a *style-element* and "Text Options" on page 852 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.

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).

Tip: The FILLATTRS= option can be used to set transparency for just the interior fill area of the boxes. You can combine this option with FILLATTRS= to set one transparency for the box outlines and the whiskers, mean, median, caps, and connect lines, but a different transparency for the box fills. Example:

```
datatransparency=0.2 fillattrs=(transparency=0.6)
```

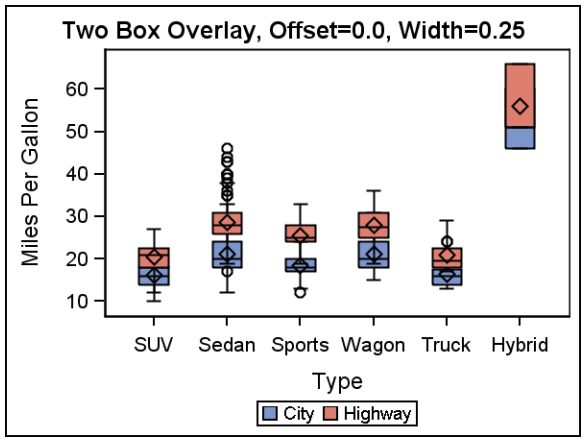
DISCRETEOFFSET=*number*
specifies an amount to offset all boxes from the discrete X ticks.

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.)

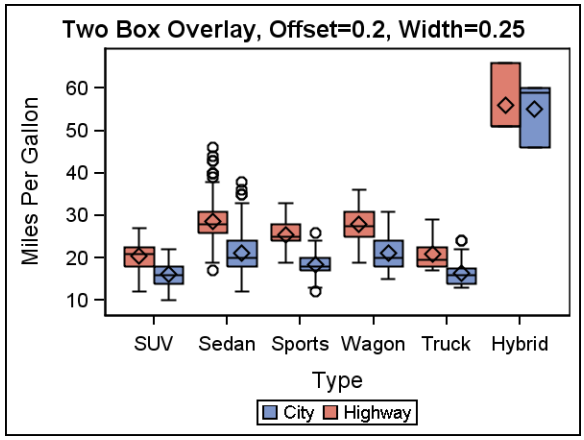
Details: This feature is useful for graphing multiple response variables side by side on a common axis. By default within an overlay-type layout, if multiple BOXPLOTPARM statements are used with different analysis variables, the boxes for matching X values are centered on the ticks. Depending on the 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

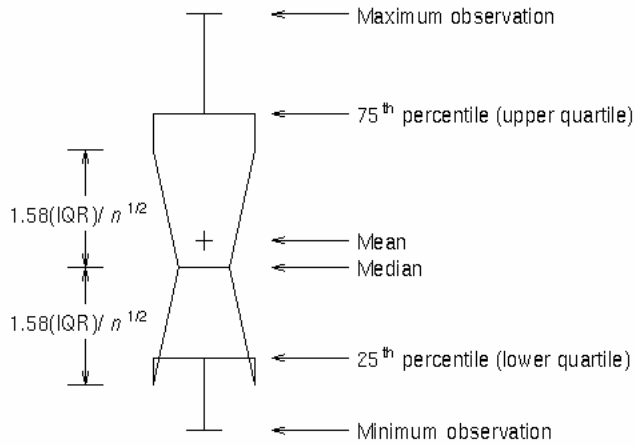
WHISKERATTRS= options. The WHISKERATTRS= option controls affects both CAPS and WHISKERS.

Tip: Regardless of which *display-options* are being displayed, this option does not affect the axis range.

Details: The endpoints of the notches are at the following computed locations.

$$median + \ and - \ 1.58(IQR/\sqrt{N})$$

In the equation, the IQR is the interquartile range and N is the sample size.



DISPLAYSTATS=NONE | STANDARD | ALL | *(statistics-list)*
specifies the statistics to be displayed for each box.

Default: NONE

NONE
does not display any statistics

STANDARD
displays N, MEAN, and STD

ALL
displays all available statistics (see the *statistics-list*)

(statistics-list)
specifies a list of features to be displayed. The list must be enclosed in parentheses and can include any of the following:

MAX	Maximum data value below the box upper fence
MEAN	Mean data value for box
MEDIAN	Median data value for box
MIN	Minimum data value above the box upper fence
N	Number of observations for box
Q1	Lower quartile (25th percentile) for box
Q3	Upper quartile (75th percentile) for box
STD	Standard deviation of the data for box
RANGE	Range of the data (MAX–MIN)
IQR	Interquartile range (Q3–Q1)

Restriction: This option is ignored if ORIENT= HORIZONTAL

Restriction: Only those statistics included in the STAT= column can be displayed. RANGE requires both MAX and MIN to be included. IQR requires both Q1 and Q3 to be included.

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 233 .)

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 849 for the syntax on using a *style-element* and "Fill Options" on page 850 for available *fill-options*.

Default:

- For non-grouped data, the Color attribute of GraphDataDefault style element.
- 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.

Tip: The DATATRANSPARENCY= option sets the transparency for the box outlines, box fill, whiskers, mean, median, caps, connect lines, and outliers. You can combine this option with DATATRANSPARENCY= to set one transparency for the box outlines and the whiskers, mean, median, caps, and connect lines, but a different transparency for the box fills. Example:

```
datatransparency=0.2 fillattrs=(transparency=0.6)
```

GROUP=*column* | *discrete-attr-var* | *expression*
creates a box plot for each unique group value of the specified column.

discrete-attr-var
specifies a discrete attribute variable that is defined in a DISCRETEATTRVAR statement.

Restriction: A discrete attribute variable specification must be a direct reference to the attribute variable. It cannot be set by a dynamic variable.

Default:

- Each distinct group value is represented in the graph by a different box outline color. The outline colors are defined by the GraphData1:ContrastColor–GraphDataN:ContrastColor style references.
- If the MEDIAN, CAPS, or NOTCHES are enabled by the DISPLAY= option, each of these features uses the same color as the box outline. Line styles do not change by group value.
- If the MEAN or OUTLIERS are enabled by the DISPLAY= option, each distinct group value is represented by a different marker. The markers are defined by the MarkerSymbol and ContrastColor attributes of the GraphData1–GraphDataN style elements. A marker is used for both MEAN and OUTLIERS, if displayed.
- If box fills are enabled by the ODS style or by the DISPLAY= option, each distinct group value is represented in the graph by a different fill color. The fill colors are defined by the GraphData1:Color–GraphDataN:Color style references.

Interaction: The box plot display depends on the setting for the GROUPDISPLAY= option.

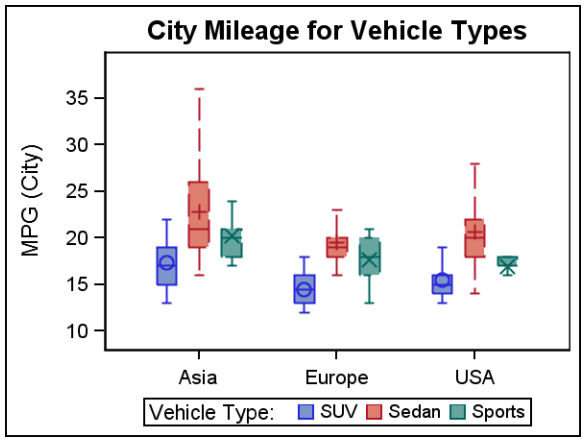
Interaction: Connect lines are not drawn for grouped data.

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 group values. The INDEX= option can be used to alter the default sequence of colors and markers.

Interaction: The INCLUDEMISSINGGROUP option controls whether missing group values are considered a distinct group value.

Tip: 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 box outlines, but the PATTERN= setting on the OUTLIERATTRS= option could be used to assign the same line pattern to all box outlines and connect lines.

GROUPDISPLAY=OVERLAY | CLUSTER
specifies how to display the boxes that represent group values for the coordinate pairs. The following example shows a box plot with GROUPDISPLAY=CLUSTER:



Default: OVERLAY

OVERLAY
draws boxes for a given group value at the exact coordinate. Depending on the data, boxes at a given coordinate might overlap.

CLUSTER
draws boxes for a given group value adjacent to each other. This option is available only when the category (X) column is discrete.

Interaction: This option is ignored unless GROUP= is specified.

Tip: Use the CLUSTERWIDTH= option to control the width of the clusters when CLUSTER is in effect.

GROUPORDER=DATA | ASCENDING | DESCENDING
specifies the ordering of the grouped boxes for each category value. It also sets the default order of the groups in the legend.

Default: DATA

DATA
shows each group within a category in data order of the group column.

ASCENDING
shows each group within a category in ascending order of the group column.

DESCENDING
shows each group within a category in descending order of the group column.

Interaction: This option is ignored unless GROUP= is specified.

Note: Attributes such as color, symbol, and pattern are assigned to each group in the DATA order by default regardless of the GROUPORDER= option setting.

Note: The ASCENDING and DESCENDING settings linguistically sort the group values within each category (or X value) for display position purposes only. The data order of the observations and the visual attributes that are assigned to the group values remain unchanged.

INCLUDEMISSINGGROUP=*boolean*
specifies whether missing values of the group variable are included in the plot.

Default: TRUE

Interaction: For this option to take effect, the GROUP= option must also be specified.

Tip: Unless a discrete attribute map is in effect or the INDEX= option is used, the attributes of the missing group value are determined by the GraphMissing style element except when the MISSING= system option is used to specify a non-default missing character or when a user-defined format is applied to the missing group value. In those cases, the attributes of the missing group value are determined by a GraphData1–GraphDataN style element.

INDEX=*numeric-column* | *expression*
specifies indices for mapping line attributes (color and pattern) to one of the GraphData1–GranphDataN style elements.

Default: no default

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.

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: 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 144.

INTERVALBOXWIDTH= AUTO | *dimension*
specifies the box width when an interval category (X) column is specified.

Default: AUTO

AUTO	Uses 85% of the smallest interval between any two boxes for the given plot.
dimension	Sets the box width to the specified value.

Restriction: The axis type for the category axis (X by default) must be LINEAR, and the X column must be numeric.

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 233.

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 the legend item that is associated with this plot.

Default: The *string* specified on the NAME= option.

Restriction: This option applies only to an associated DISCRETELEGEND statement.

Interaction: If the GROUP= option is specified, this option is ignored.

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 849 for the syntax on using a *style-element* and "Marker Options" on page 851 for available *marker-options*.

Default:

- For non-grouped data, GraphBoxMean style element.
- For grouped data, the MarkerSymbol, Markersize, and ContrastColor attributes of the GraphData1 – GraphDataN style elements.

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 849 for the syntax on using a *style-element* and "Line Options" on page 850 for available *line-options*.

Default:

- For non-grouped data, the GraphBoxMedian style element.
- For grouped data, the LineStyle and LineThickness attributes of the GraphBoxMedian style element, and the ContrastColor attribute of the GraphData1 – GraphDataN style elements.

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 849 for the syntax on using a *style-element* and "Marker Options" on page 851 for available *marker-options*.

Default:

- For non-grouped data, GraphOutlier style element.
- For grouped data, the MarkerSymbol, Markersize, and ContrastColor attributes of the GraphData1 – GraphDataN style elements.

Interaction: This option is ignored if EXTREME= TRUE or the DISPLAY= option does not display the outliers.

OUTLIERTIP=(*role-list*)
specifies the information to display when the cursor is positioned over an outlier. If this option is used, it replaces all of the information that is displayed by default. Roles for columns that do not contribute to the box plot can be specified along with roles that do contribute.

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 BOXPLOTPARM roles. BOXPLOTPARM roles for OUTLIERTIP include X , Y , STAT , and DATALABEL .

Note: In the tooltip, the STAT role displays the text "outlier" or "far outlier" as applicable.

The following example displays tooltips for the columns that are assigned to the X and Y roles, and also the data column OBS, which is not assigned to any pre-defined BOXPLOTPARM role. The OBS column must first be assigned a role:

```
ROLENAME=(TIP1= OBS) OUTLIERTIP=(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 OUTLIERTIP variables can be controlled with the TIPLABEL= and TIPFORMAT= options.

See Also: The ROLENAME= option for specifying user-defined roles. The TIP option for specifying the information to display when the cursor is positioned over a box or whisker in the box plot.

OUTLINEATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)
specifies the appearance of the box outline. See "General Syntax for Attribute Options" on page 849 for the syntax on using a *style-element* and "Line Options" on page 850 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 562.

ROLENAME=(*role-name-list*)
specifies user-defined roles that can be used to display information in the tooltips.

Default: no user-defined roles

(*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, STAT, DATALABEL , N, STD, MIN, MAX, MEAN, MEDIAN, Q1, Q3, and BOXWIDTH.

This option provides a way to add to the data columns that appear in tooltips that are specified by the OUTLIERTIP= and TIP= options.

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.

TIP=(*role-list*)
specifies the information to display when the cursor is positioned over a box or whisker in the box plot. If this option is used, it replaces all of the information that is displayed by default. Roles for columns that do not contribute to the box plot can be specified along with roles that do contribute.

Default: The columns assigned to these roles are automatically included in the tooltip information: X, N, MIN, MAX, Q1, Q3, MEAN, MEDIAN, and STD.

(*role-list*)
an ordered, blank-separated list of unique BOXPLOTPARM roles. BOXPLOTPARM roles for TIP include X, N, STD, MIN, MAX, MEAN, MEDIAN, Q1, Q3 and BOXWIDTH.

User-defined roles are defined with the ROLENAME= option.

The following example displays tooltips for the columns assigned to the roles X and Y, and also to the data column OBS, which is not assigned to any pre-defined BOXPLOTPARM 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.

Requirement: If a user-defined role is specified in the ROLENAME= option, then the data values for that role must be identical for each distinct X (category) value.

Interaction: The labels and formats for the TIP variables can be controlled with the TIPLABEL= and TIPFORMAT= options.

Tip: Statistics such as N, MIN, MAX are special roles. They are not column-based like the X role.

See Also: The ROLENAME= option for specifying user-defined roles. The OUTLIERTIP= option for specifying the information to display when the cursor is positioned over an outlier.

TIPFORMAT=(*role-format-list*)
specifies display formats for the information that is defined by the tooltip roles.

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 MEAN MIN MAX) TIPFORMAT=(X=4. 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 OUTLIERTIP or TIP options are used.

Interaction: For this option to have any effect, a column must be assigned to each of the specified roles.

TIPLABEL=(*role-label-list*)
specifies display labels for the information that is defined by the tooltip roles.

Default: The column label or column name of the variable that is assigned to the role.

(*role-label-list*)
a list of *role-name* = *"string"* pairs separated by blanks.

```
ROLENAME=(TIP1=OBS)
TIP=(TIP1 X MEAN MIN MAX) OUTLIERTIP=(TIP1)
TIPLABEL=(X="Box" TIP1="Observeration")
```

Requirement: This option provides a way to control the labels of columns that appear in tooltips. Only the roles that appear in the OUTLIERTIP= or TIP= options are used.

Interaction: For this option to have any effect, a column must be assigned to each of the specified roles.

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 849 for the syntax on using a *style-element* and "Line Options" on page 850 for available *line-options*.

Default:

- For non-grouped data, the GraphBoxWhisker style element.
- For grouped data, the LineStyle and LineThickness attributes of the GraphBoxWhisker style element, and the ContrastColor attribute of the GraphData1 – GraphDataN style elements.

Restriction: The caps are always drawn with a solid line.

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 557.

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 557.

Chapter 22

BUBBLEPLOT Statement

Description and Syntax

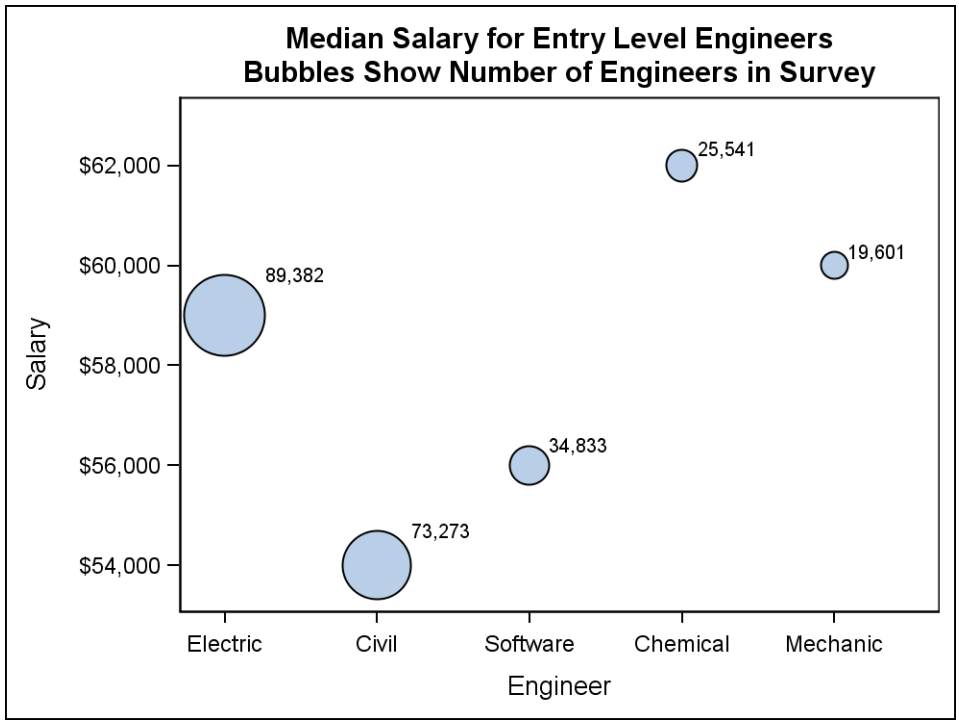
Creates a bubble plot of the input data. The locations of the bubble centers correspond to the values of X and Y variables in the data, and the bubble radii correspond to the values of a SIZE variable.

BUBBLEPLOT X= *column* | *expression*
Y= *column* | *expression*
SIZE= *numeric-column* | *expression* </ *option(s)*>;

Example Program and Statement Details

Example Graph

The following graph was generated by the "Example Program" on page 256 :



Example Program

```
data bubbleintro;
input Engineer $ Salary number;
format Salary dollar7.0 number comma6.0;
datalines;
Electric 59000 89382
Civil 54000 73273
Software 56000 34833
Chemical 62000 25541
Mechanical 60000 19601
;
proc template;
  define statgraph engineer;
    begingraph;
      entrytitle 'Median Salary for Entry Level Engineers';
      entrytitle 'Bubbles Show Number of Engineers in Survey';
      layout overlay;
        bubbleplot x=engineer y=salary
          size=number / datalabel=number;
      endlayout;
    endgraph;
  end;

proc sgrender data=bubbleintro template=engineer;
run;
```

Statement Summary

The BUBBLEPLOT statement displays one bubble for each row in the data, provided that row contains nonmissing values for X, Y, and SIZE. By default, the bubbles are

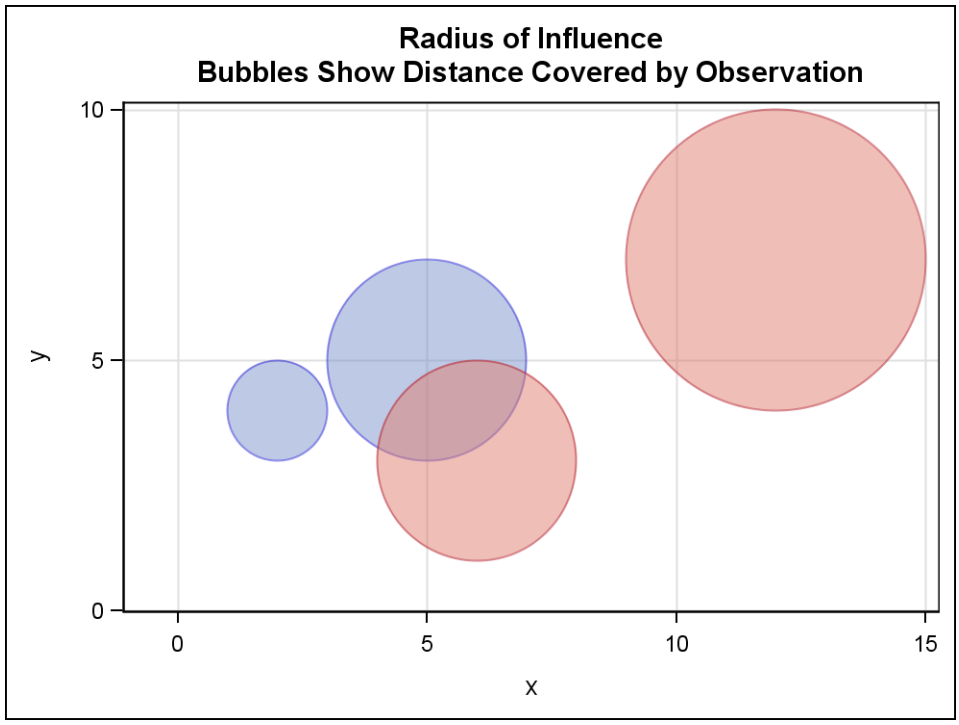
displayed as filled, outlined circles. Regardless of the data order, the bubbles are always drawn from the largest size to the smallest size.

By default, the minimum and maximum values of the SIZE= column establish a range over which the bubble radii increase in linear proportion. The actual drawing size of the smallest and largest bubble is set automatically. You can adjust the smallest and largest bubble sizes with the BUBBLERADIUSMIN= and BUBBLERADIUSMAX= options. In these cases where the bubble sizes are proportional to each other, the default setting RELATIVESCALE=TRUE is appropriate.

If the SIZE= values are in the same units as the X and Y values, and both X and Y are numeric, you can generate a plot where the bubble-radius units match the axis-scale units. To do so, specify the BUBBLEPLOT statement within a LAYOUT OVERLAYEQUATED block, and in the BUBBLEPLOT statement, set RELATIVESCALE=FALSE.

Note: Within a LAYOUT OVERLAY, the unit-interval of the X and Y axes are not necessarily the same and the bubbles might be distorted into ellipses when RELATIVESCALE=FALSE. The OVERLAYEQUATED container ensures that the bubbles are displayed as circles, assuming that both the X= and Y= arguments specify numeric columns.

```
data influence;
input x y radius category;
datalines;
2  4 1 1
5  5 2 1
6  3 2 2
12 7 3 2
;
proc template;
  define statgraph equatedbubbles;
    begingraph;
      entrytitle 'Radius of Influence';
      entrytitle 'Bubbles Show Distance Covered by Observation';
      layout overlayequated /
          xaxisopts = (griddisplay=on)
          yaxisopts = (griddisplay=on);
        bubbleplot x=x y=y size=radius /
          group=category datatransparency=0.5
          relativescale=false ;
      endlayout;
    endgraph;
  end;

proc sgrender data=influence template=equatedbubbles;
run;
```



Required Arguments

X=*column* | *expression*
specifies the column for the X values of the bubble centers.

Y=*column* | *expression*
specifies the column for the Y values of the bubble centers.

SIZE=*numeric-column* | *expression*
specifies the bubble SIZE values.

Options

Statement Option	Description
BUBBLERADIUSMAX	Specifies the drawing size of largest bubble.
BUBBLERADIUSMIN	Specifies the drawing size of smallest bubble.
COLORMODEL	Specifies a style element to be used with the COLORRESPONSE= option.
COLORRESPONSE	Specifies a column that is used to map bubble colors to a continuous gradient.
DATALABEL	Specifies a column for bubble labels.
DATALABELATTRS	Specifies the color and font attributes of the bubble labels.

Statement Option	Description
DATALABELPOSITION	Specifies the position of the bubble labels relative to the bubble.
DATASKIN	Enhances the visual appearance of the filled bubbles.
DATATRANSPARENCY	Specifies the degree of the transparency of the bubble fills and bubble outlines.
DISPLAY	Specifies whether to display outlined bubbles, filled bubbles, or outlined and filled bubbles.
FILLATTRS	Specifies the appearance of the filled bubble areas.
GROUP	Creates a separate bubble color for each unique grouping that is specified.
INCLUDEMISSINGGROUP	Specifies whether missing values of the group variable are included in the plot.
INDEX	Specifies indices for mapping bubble color and line attributes to one of the GraphData1–GranphDataN style elements.
LEGENDLABEL	Specifies a label for a legend.
NAME	Assigns a name to a plot statement for reference in other template statements.
OUTLINEATTRS	Specifies the appearance of the bubble outlines.
PRIMARY	Specifies that the data columns for this plot and the plot type be used for determining default axis features
RELATIVESCALE	Specifies whether the SIZE= column values are interpreted as relative values.
RELATIVESCALETYPE	Specifies the type of scaling that is to be applied to the SIZE= column values.
REVERSECOLORMODEL	Specifies whether to reverse the gradient (color ramp) that is defined by the COLORMODEL= option.
ROLENAME	Specifies user-defined roles that can be used to display information in the tooltips.
SIZETHRESHOLDMAX	Specifies a SIZE= column value threshold at which bubble size is clamped to the BUBBLERADIUSMAX= option value.

Statement Option	Description
TIP	Specifies the information to display when the cursor is positioned over the bubbles.
TIPFORMAT	Specifies display formats for information defined by the tooltip roles.
TIPLABEL	Specifies display labels for information defined by the tooltip roles.
URL	Specifies an HTML page to display when a bubble 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 to the secondary Y2 (right) axis.

BUBBLERADIUSMAX= *dimension*
specifies the drawing size of the largest bubble.

Default: Three times as large as the size set by GRAPHDATADEFULT:markerSize

Restriction: The BUBBLERADIUSMAX= value must be greater than the BUBBLERADIUSMIN= value.

Interaction: This option is ignored when RELATIVESCALE= FALSE.

Tip: A maximum size that is specified as a percent is interpreted as a percent of the graph's height. The height can be adjusted with the DESIGNHEIGHT= option in the BEGINGRAPH statement or the HEIGHT= option in the ODS GRAPHICS statement.

BUBBLERADIUSMIN= dimension
specifies the drawing size of the smallest bubble.

Default: GRAPHDATADEFULT:markerSize

Interaction: This option is ignored when RELATIVESCALE= FALSE.

Restriction: The BUBBLERADIUSMIN= value must be less than the BUBBLERADIUSMAX= value.

Tip: A maximum size that is specified as a percent is interpreted as a percent of the graph's height. The height can be adjusted with the DESIGNHEIGHT= option in the BEGINGRAPH statement or the HEIGHT= option in the ODS GRAPHICS statement.

COLORMODEL= *style-element*
specifies a style element to be used with the COLORRESPONSE= option.

Default: The ThreeColorRamp style element for filled bubbles and ThreeColorAltRamp for unfilled bubbles.

style-element
name of a style element. The style element should contain these style attributes:

STARTCOLOR	color for the smallest data value of the RESPONSECOLOR= variable
NEUTRALCOLOR	color for the midpoint of the range of the REPONSECOLOR variable
ENDCOLOR	color for the highest data value of the RESPONSECOLOR variable

Interaction: For this option to take effect, the COLORRESPONSE= option must also be specified.

Tip: To reverse the start and end colors of the ramp that is assigned to the color model, use the REVERSECOLORMODEL= option.

COLORRESPONSE= *numeric-column* | *range-attr-var* | *expression*
specifies a column that is used to map bubble colors to a continuous gradient. The gradient represents the range of unique response values.

range-attr-var
specifies a range attribute variable that is defined in a RANGEATTRVAR statement.

Restriction: A range attribute variable specification must be a direct reference to the attribute variable. It cannot be set by a dynamic variable.

Default:

- If a *numeric-column* or an *expression* is specified, the color ramp that is specified by the COLORMODEL= option.
- If a variable that is associated with a RANGEATTRVAR statement is specified, the color ramp is defined by the associated RANGEATTRMAP statement.

Interaction: If DISPLAY= (FILL OUTLINE), the bubble fill colors are assigned according to the gradient, but the bubble outlines are fixed according to the OUTLINEATTRS= settings.

If DISPLAY=(OUTLINE), the bubble outlines vary according to the gradient.

Interaction: The mapped color used for the bubbles is also used for the data labels.

Interaction: The GROUP= option is ignored if it is specified when this option is used.

DATALABEL= *column* | *expression*
specifies a column for bubble labels. The label positions are adjusted to prevent them from overlapping.

Default: no data labels are displayed

DATALABELATTRS=*style-element* | *style-element* (*text-options*) | (*text-options*)
specifies the color and font attributes of the bubble labels. See "General Syntax for Attribute Options" on page 849 for the syntax on using a *style-element* and "Text Options" on page 852 for available *text-options*.

Default:

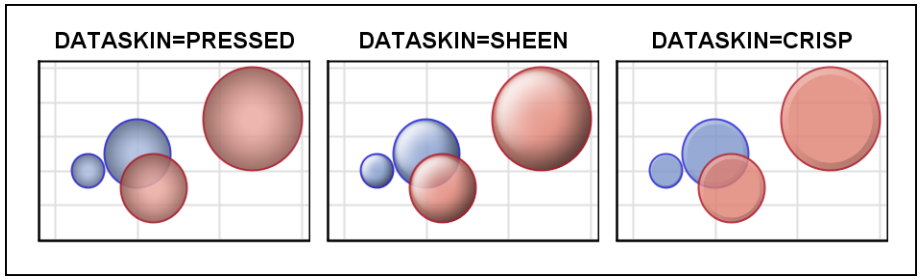
- For non-grouped data, the GraphDataText style element.
- For grouped data, the text color is derived from the GraphData1–GraphDataN style elements. The data label color changes to match the group color derived from the ContrastColor attribute of the style element that is in effect.

Interaction: The default attributes are overridden if the COLORRESPONSE= option is used.

DATALABELPOSITION = TOPRIGHT | TOP | TOPLEFT | LEFT | CENTER | RIGHT | BOTTOMLEFT | BOTTOM | BOTTOMRIGHT
specifies the location of the bubble labels relative to the bubble.

Default: TOPRIGHT

DATASKIN= NONE | PRESSED | SHEEN | CRISP | GLOSS | MATTE
enhances the visual appearance of the filled bubbles.



Default: NONE

Requirement: For this option to have any effect, DISPLAY= FILL must be in effect. Otherwise, this option is ignored.

Interaction: The appearance of the data skin is based on the FILLATTRS= color.

Interaction: This option is ignored if the RELATIVESCALE= option is set to FALSE.

Interaction: When a data skin is applied, all bubble outlines are set by the skin, and the OUTLINEATTRS= option is ignored.

DATATRANSPARENCY=*number*
specifies the degree of the transparency of the bubble fills and bubble outlines.

Default: 0

Range: 0 (opaque) to 1 (entirely transparent)

Tip: The FILLATTRS= option can be used to set transparency for just the filled bubble areas. You can combine this option with FILLATTRS= to set one transparency for the bubble outlines but a different transparency for the bubble fills. Example:

```
datatransparency=0.2 fillattrs=(transparency=0.6)
```

DISPLAY=STANDARD | ALL | (*display-options*)
specifies whether to display outlined bubbles, filled bubbles, or outlined and filled bubbles.

Default: STANDARD

STANDARD
displays outlined, filled bubbles.

ALL
displays outlined, filled bubbles.

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

OUTLINE	displays outlined bubbles.
FILL	displays filled bubbles.

Tip: Use the DATASKIN= , OUTLINEATTRS= , and FILLATTRS= options to control the appearance of the bubbles.

FILLATTRS=*style-element* | *style-element* (*fill-options*) | (*fill-options*)
specifies the appearance of the filled bubble areas. See "General Syntax for Attribute Options" on page 849 for the syntax on using a *style-element* and "Fill Options" on page 850 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.

Interaction: When COLORRESPONSE= is in effect and the DISPLAY= option enables FILL display, the FILLATTRS= suboption COLOR= is ignored, and the bubble fill colors vary according to the gradient.

Tip: The DATATRANSPARENCY= option sets the transparency for the bubble fills and the bubble outlines. You can combine this option with DATATRANSPARENCY= to set one transparency for the outlines but a different transparency for the fills. Example:

```
datatransparency=0.2 fillattrs=(transparency=0.6)
```

GROUP=*column* | *discrete-attr-var* | *expression*
creates a separate bubble color for each unique grouping that is specified.

discrete-attr-var
specifies a discrete attribute variable that is defined in a DISCRETEATTRVAR statement.

Restriction: A discrete attribute variable specification must be a direct reference to the attribute variable. It cannot be set by a dynamic variable.

Default: No default. All bubbles have only one fill and one outline color as specified by the FILLATTRS= and OUTLINEATTRS options.

Interaction: When this option is used, the unique column values are found and then the bubble colors are taken from the following sources:

- If a *column* or an *expression* is specified, the bubble attributes are derived from the GraphData1–GraphDataN style elements. If the bubbles are filled, then the COLOR attribute is used for bubble fill and the CONTRASTCOLOR attribute is used for the bubble outlines. If the bubbles are not filled, then the CONTRASTCOLOR and PATTERN are used for the bubble outlines.
- If a variable that is associated with a DISCRETEATTRVAR statement is specified, the color mapping for the bubbles is defined by the associated DISCRETEATTRMAP statement.

Interaction: The mapped color that is used for outlines is also used as the color of the data labels.

Interaction: This option is ignored if the COLORRESPONSE= option is also used.

Interaction: The INCLUDEMISSINGGROUP option controls whether missing group values are considered a distinct group value.

See Also: The FILLATTRS= option.

INCLUDEMISSINGGROUP=*boolean*
specifies whether missing values of the group variable are included in the plot.

Default: TRUE

Interaction: For this option to take effect, the GROUP= option must also be specified.

Tip: Unless a discrete attribute map is in effect or the INDEX= option is used, the attributes of the missing group value are determined by the GraphMissing style element except when the MISSING= system option is used to specify a non-default missing character or when a user-defined format is applied to the missing group value. In those cases, the attributes of the missing group value are determined by a GraphData1–GraphDataN style element.

INDEX= *numeric-column* | *expression*
specifies indices for mapping bubble color and line attributes to one of the GraphData1–GraphDataN style elements.

Default: no default

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.

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: 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 144.

LEGENDLABEL= "*string*"
specifies a label for the legend item that is associated with this plot.

Default: The *string* specified on the NAME= option.

Restriction: This option applies only to an associated DISCRETELEGEND statement.

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.

OUTLINEATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)
specifies the appearance of the bubble outlines. See "General Syntax for Attribute Options" on page 849 for the syntax on using a *style-element* and "Line Options" on page 850 for available *line-options*.

Default:

- For non-grouped data, the GraphOutlines style element.
- For grouped data, unfilled bubbles use both the CONTRASTCOLOR and PATTERN attributes of the GraphData1–GraphDataN style elements. Filled bubbles use only the CONTRASTCOLOR attribute.

  If the COLORRESPONSE= option is specified and the bubbles are filled, then the outline attributes are derived from the GraphDataDefault style element. For unfilled bubbles, the outline colors vary according to the gradient.

Interaction: For this option to have any effect, outlines must be enabled by the ODS style or the DISPLAY= option.

Interaction: If the DATASKIN= option applies a data skin, this option is ignored.

Interaction: When the COLORRESPONSE= and DISPLAY=(OUTLINE) options are in effect, the OUTLINEATTRS= suboption COLOR= is ignored, and the bubble outline colors vary according to the 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 562

RELATIVESCALE=*boolean*
specifies whether the SIZE= column values are interpreted as relative values.

Default: TRUE

Interaction: When this option is set to TRUE, the BUBBLERADIUSMAX= and BUBBLERADIUSMIN= options can be used to fix the drawing size of the smallest bubble and largest bubble. If RELATIVESCALE=FALSE, the BUBBLERADIUSMAX= and BUBBLERADIUSMIN= options are ignored.

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

Interaction: If either the X= argument or the Y= argument specifies categorical values, RELATIVESCALE=FALSE is ignored.

Details: Relative means that the size values do not translate directly into bubble radii. Rather, the bubble sizes are scaled to represent the value range of the SIZE= column.

For example, when RELATIVESCALE=TRUE, if only two bubbles are drawn with sizes of 2 and 4 they would appear the same as only two bubbles with sizes of 4000 and 8000. By contrast, when RELATIVESCALE=FALSE, the size values are interpreted in the same units as the axes.

If you set this option to FALSE, it is recommended that you also place the BUBBLEPLOT statement in a LAYOUT OVERLAYEQUATED container. This ensures that the X and Y axis units are the same. For more information, see "Statement Summary" on page 256.

RELATIVESCALETYPE=LINEAR | PROPORTIONAL
specifies the type of scaling that is to be applied to the SIZE= column values.

Note: This feature is for the second maintenance release of SAS 9.3 and later. See "What's New in the SAS 9.3 Graph Template Language" on page xi.

LINEAR
The size of the bubbles increases in linear proportion to the range of the SIZE= column values. For example, if only two bubbles are drawn with sizes of 2 and 4, they appear the same as only two bubbles with sizes of 4000 and 8000.

PROPORTIONAL
The size of each bubble is directly proportional to its corresponding SIZE= column value. For example, if only two bubbles are drawn with sizes of 50 and 100, the bubble for SIZE=50 is drawn to half the size of the bubble for SIZE=100.

Default: LINEAR

Interaction: This option is ignored when RELATIVESCALE=FALSE.

Interaction: When the SIZETHRESHOLDMAX= option is specified, for any SIZE= column value that is greater than the SIZETHRESHOLDMAX= value, the proportional scale is adjusted so that the size of the bubble for that value is clamped to the BUBBLERADIUSMAX= value.

Interaction: If all the values for the SIZE= column are negative, RELATIVESCALETYPE=PROPORTIONAL is ignored, and the default value is used.

Interaction: When RELATIVESCALETYPE=PROPORTIONAL is specified, the BUBBLERADIUSMIN= option specifies the minimum bubble size. In that case, when a SIZE= column value results in a bubble of a size that is less than the BUBBLERADIUSMIN= value, the bubble size for that value is changed to the BUBBLERADIUSMIN= value.

REVERSECOLORMODEL = *boolean*
specifies whether to reverse the gradient (color ramp) that is defined by either the ODS style that is in effect or by the COLORMODEL= option.

Default: FALSE

ROLENAME=(*role-name-list*)
specifies user-defined roles that can be used to display information in the tooltips. Columns for the tooltip display are specified in the TIP= option.

Default: no user-defined roles

(*rolename-list*)
a blank-separated list of *rolename* = *column* pairs.

For example, ROLENAME=(TIP1=POP_2009) assigns the data column POP_2009 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 , SIZE , GROUP , DATALABEL , and COLORRESPONSE .

Interaction: For this option to take effect, the TIP= option must also be used.

SIZETHRESHOLDMAX=*numeric-value*
specifies a SIZE= column value threshold at which bubble size is clamped to the BUBBLERADIUSMAX= option value. The size of the bubbles for all SIZE= column values that equal or exceed the specified threshold value is set to the BUBBLERADIUSMAX= value.

Note: This feature is for the second maintenance release of SAS 9.3 and later. See "What's New in the SAS 9.3 Graph Template Language" on page xi.

Default: The maximum SIZE= column value is mapped to the BUBBLERADIUSMAX= option value.

TIP=(*role-list*)
specifies the information to display when the cursor is positioned over the bubbles. If this option is used, it replaces all the information displayed by default. Roles for columns that do not contribute to the bubble 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 , SIZE , GROUP , DATALABEL , and COLORRESPONSE .

(*role-list*)
an ordered, blank-separated list of unique BUBBLEPLOT and user-defined roles. BUBBLEPLOT roles include X, Y, SIZE, GROUP, DATALABEL, and COLORRESPONSE.

User-defined roles are defined with the ROLENAME= option.

The following example displays tooltips for the columns assigned to the roles X, Y, and SIZE, as well as the column POP_2009. The POP_2009 column is not assigned to any pre-defined BUBBLEPLOT role, so it must first be assigned a role:

```
ROLENAME=(TIP1=POP_2009)
TIP= (TIP1 X Y SIZE)
```

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 information defined by the tooltip roles. Only the roles that appear in the TIP= option are used.

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=POP_2009)
TIP=(TIP1 X Y SIZE)
TIPFORMAT=(TIP1=comma12.)
```

Requirement: 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 information defined by the tooltip roles. Only the roles that appear in the TIP= option are used.

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=POP_2009)
TIP=(TIP1 X Y SIZE)
TIPLABEL=(TIP1="Population in 2009")
```

Requirement: 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 bubble 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* is a valid reference.

Requirement: To generate a plot with selectable bubbles, 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 bubbles 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: 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 557.

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 557.

Chapter 23

CONTOURPLOTPARM Statement

Description and Syntax

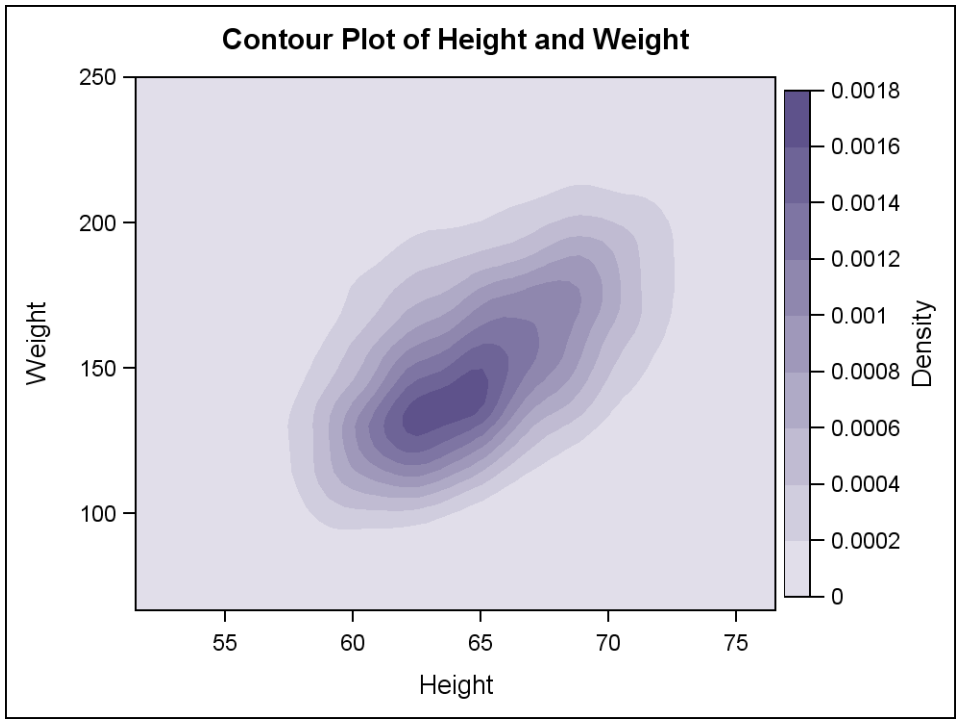
Creates a contour plot representing a response variable evaluated over a grid of X and Y values.

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 270:



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 for a 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= 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.

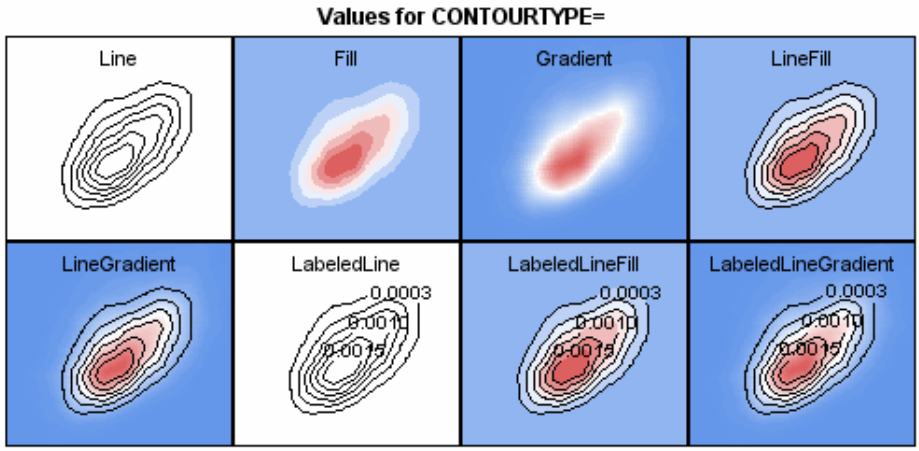
style-element
Name of a style element. The style element can contain these style attributes:

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 "Mesoscale Objective Map Analysis Using Weighted Time-Series Observations."[1]

LEGENDLABEL= *"string"*
specifies a label for the legend item that is associated with this plot.

Default: The Z-variable label. If a label is not defined, the default is 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 849 for the syntax on using a *style-element* and "Line Options" on page 850 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 849 for the syntax on using a *style-element* and "Text Options" on page 852 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

---

[1] Barnes, Stanley L March 1973. "Mesoscale Objective Map Analysis Using Weighted Time-Series Observations." Technical Memorandum (NOAA TM ERLNSSL-62), United States National Oceanic and Atmospheric Administration, Environmental Research Labs, Norman, OK..

Interaction: This option is honored only if the CONTOURTYPE= option specifies 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 the NLEVELS= option 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 562 .

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 557.

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 557.

Chapter 24

DENDROGRAM Statement

Description and Syntax

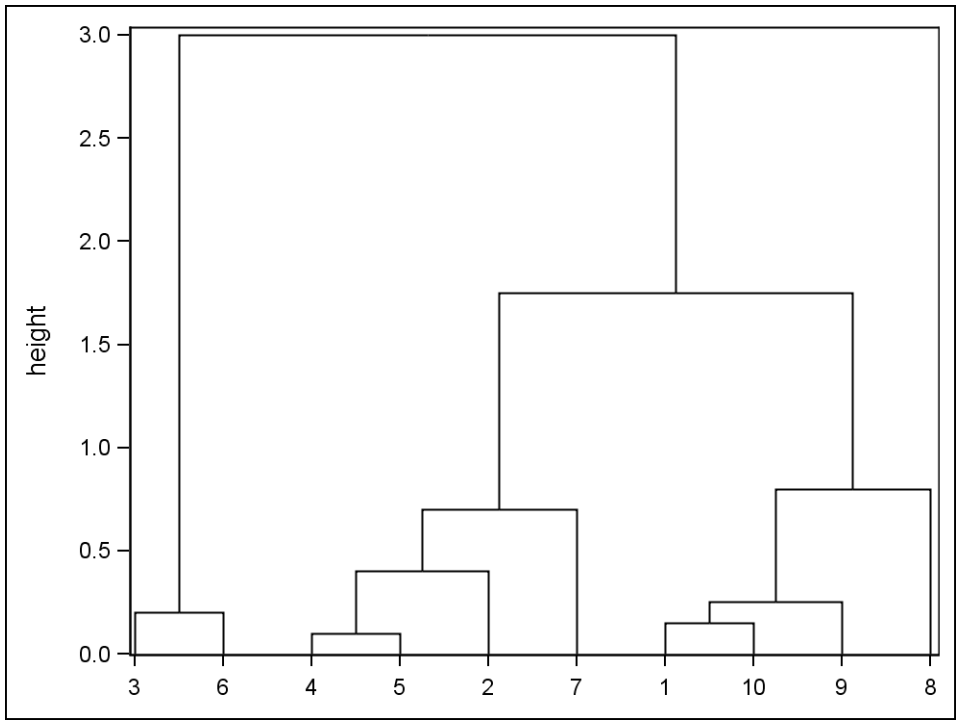
Creates a tree diagram that is typically used to display the results of a hierarchical clustering analysis.

DENDROGRAM NODEID = *column* | *expression*
PARENTID = *column* | *expression*
CLUSTERHEIGHT = *numeric-column* | *expression* </option(s)>;

Example Program and Statement Details

Example Graph

The following graph was generated by the "Example Program" on page 278:



Example Program

```
data clustree;
  input id $ parent $7-12 height nClus;
  label id="Cluster ID" parent="Parent ID";
  datalines;
clus1       3 1
clus2 clus1 0.2 7
clus3 clus1 1.75 2
clus4 clus3 0.7 4
clus5 clus3 0.8 3
clus6 clus4 0.4 5
clus7 clus6 0.1 9
clus8 clus5 0.25 6
clus9 clus8 0.15 8
1     clus9 0 10
2     clus6 0 10
3     clus2 0 10
4     clus7 0 10
5     clus7 0 10
6     clus2 0 10
7     clus4 0 10
8     clus5 0 10
9     clus8 0 10
10    clus9 0 10
run;

proc template;
  define statgraph dendrogram;
  begingraph;
    layout overlay;
```

```
      dendrogram nodeID=id parentID=parent clusterheight=height;
    endlayout;
  endgraph;
end;
run;

proc sgrender data=clustree template=dendrogram;
run;
```

Statement Summary

A dendrogram is a tree diagram that is typically used to show the cluster arrangements in hierarchical data. The DENDROGRAM statement supports clusters with only a single root. If multiple roots are found in the data, a warning is issued to the SAS log and the dendrogram is not drawn.

In the Graph Template Language, a DENDROGRAM plot typically appears by itself in a LAYOUT OVERLAY container. You can overlay REFERENCELINE or BANDPLOT statements on a DENDROGRAM, but overlaying other plot types might produce unexpected results.

Using the DENDROGRAM statement in layouts where the axis ranges are merged across cells might produce unexpected results.

Required Arguments

NODEID =*column* | *expression*
specifies a column for the ID values of the nodes. Each node ID value must be unique. If duplicate NODEID values are found, the dendrogram is not rendered . The *column* can be numeric or character, but it must be of the same type and have the same formatted length as the PARENTID column.

The maximum number of nodes that are supported by the dendrogram is determined by the DISCRETEMAX= option in the ODS GRAPHICS statement. The default value is DISCRETEMAX=1000. If the graph data contains more than 1000 discrete values, the dendrogram is not drawn and a warning is issued to the SAS log. In that case, you can use the DISCRETEMAX= option to increase the maximum number of discrete values that are allowed.

PARENTID =*column* | *expression*
specifies a column for the parent ID values of the nodes. The *column* can be numeric or character, but it must be of the same type and have the same formatted length as the NODEID column.

CLUSTERHEIGHT=*numeric-column* | *expression*
specifies the column for the height values for each node.

Options

Statement Option	Description
CLUSTERS	Specifies a column that contains the resultant number of clusters at each node.
CUT	Specifies whether the tree is to be cut.

Statement Option	Description
CUTOPTS	Specifies pruning options for cutting the dendrogram.
DATATRANSPARENCY	Specifies the degree of the transparency of the dendrogram lines.
LEGENDLABEL	Specifies a label for a legend.
LINEATTRS	Specifies the properties of the dendrogram lines.
NAME	Assigns a name to a plot statement for reference in other template statements.
ORIENT	Specifies the orientation of the dendrogram leaf axis.
PRIMARY	Specifies that the data columns for this plot be used for determining default axis features.
ROLENAME	Specifies user-defined roles that can be used to display information in the tooltips.
TIP	Specifies the information to display when the cursor is positioned over a dendrogram line.
TIPFORMAT	Specifies display formats for information defined by the tooltip roles.
TIPLABEL	Specifies display labels for information defined by the tooltip roles.
TREETYPE	Specifies the type of tree structure to draw.
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.

CLUSTERS=*numeric-column* | *expression*
specifies a numeric column containing the resultant number of clusters at each node.

Default: no default

Interaction: For this option to take effect, the *pruning-options* in the CUTOPTS= option must set TYPE=NCLUSTERS and specify a number for the NCLUSTERS= setting.

CUT=*boolean*
specifies whether the tree is to be cut.

Default: FALSE

Tip: To set the properties of the CUT, use the CUTOPTS= option.

CUTOPTS=(*pruning-options*)
specifies pruning options for cutting the dendrogram.

The following *pruning-options* must be specified as a list of *option* = *value* pairs separated by blanks. The list must be enclosed in parentheses.

CUTHEIGHT = *number*
specifies the height at which the tree is to be pruned.

Default: The tree is not pruned.

Interaction: For this setting to take effect, *pruning-option* TYPE=CUTHEIGHT must also be set. In addition, the CUT= option must be set to TRUE.

NCLUSTERS = *number*
specifies the number of clusters to use for pruning the tree.

Default: The tree is not pruned.

Interaction: For this setting to take effect, *pruning-option* TYPE=NCLUSTERS must also be set. In addition, the CLUSTERS= option must be used, and the CUT= option must be set to TRUE.

OUTLINEATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)
specifies the attributes of the cut lines. See "General Syntax for Attribute Options" on page 849 for the syntax on using a *style-element* and "Line Options" on page 850 for available *line-options*.

Default: The GraphDataDefault style element.

TYPE = CUTHEIGHT | NCLUSTERS
specifies which rule to use to prune the tree.

Default: CUTHEIGHT

DATATRANSPARENCY=*number*
specifies the degree of the transparency of the dendrogram lines.

Default: 0

Range: 0 (opaque) to 1 (entirely transparent)

LEGENDLABEL= *"string"*
specifies a label for the legend item that is associated with this plot.

Default: The name that is assigned to the dendrogram on the NAME= option.

Restriction: This option applies only to an associated DISCRETELEGEND statement.

LINEATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)
specifies the attributes of the dendrogram lines. See "General Syntax for Attribute Options" on page 849 for the syntax on using a *style-element* and "Line Options" on page 850 for available *line-options*.

Default: The GraphDataDefault style element.

NAME=*"string"*
Interaction:

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 dendrogram leaf axis.

Default: VERTICAL

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 562.

ROLENAME=(*role-name-list*)
specifies user-defined roles that can be used to display information in the tooltips.

Default: no user-defined roles

(*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 NODEID , PARENTID , and CLUSTERHEIGHT .

This option provides a way to add to the data columns that appear in tooltips, which are specified by the TIP= option.

TIP=(*role-list*)
specifies the information to display when the cursor is positioned over a dendrogram line. If this option is used, it replaces all the information displayed by default. Roles for columns that do not contribute to the dendrogram plot can be specified along with roles that do.

Default: The columns assigned to the following roles are automatically included in the tooltip information: NODEID , PARENTID , and CLUSTERHEIGHT .

(*role-list*)
an ordered, blank-separated list of unique DENDROGRAM and user-defined roles. DENDROGRAM roles include NODEID , PARENTID , and CLUSTERHEIGHT .

User-defined roles are defined with the ROLENAME= option.

The following example displays tooltips for the columns assigned to the roles NODEID and PARENTID, as well as the column PCT, which is not assigned to any predefined role. The PCT column must first be assigned a role.

```
ROLENAME=(TIP1=PCT)
TIP= (TIP1 NODEID PARENTID)
```

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 information defined by the tooltip roles.

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=PCT)
TIP=(TIP1 NODEID PARENTID)
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 information defined by the tooltip roles.

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 NODEID PARENTID)
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.

TREETYPE=RECTANGULAR | TRIANGULAR
specifies the type of tree structure to draw.

Default: RECTANGULAR

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 557.

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 557.

Chapter 25

DENSITYPLOT Statement

Description and Syntax

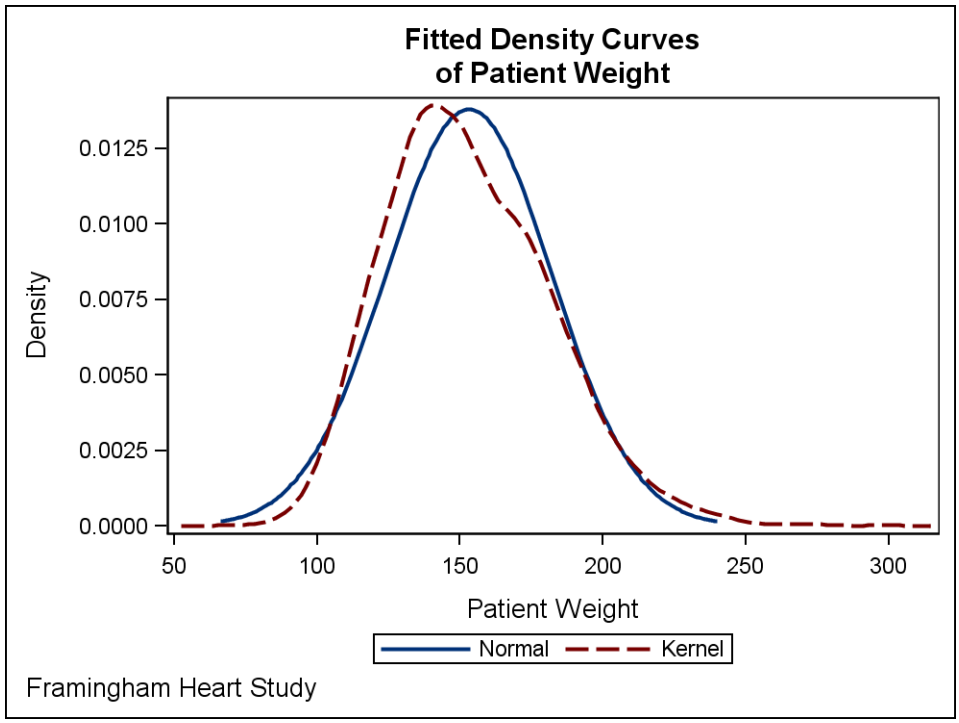
Creates a univariate probability density curve computed from input data.

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 286:

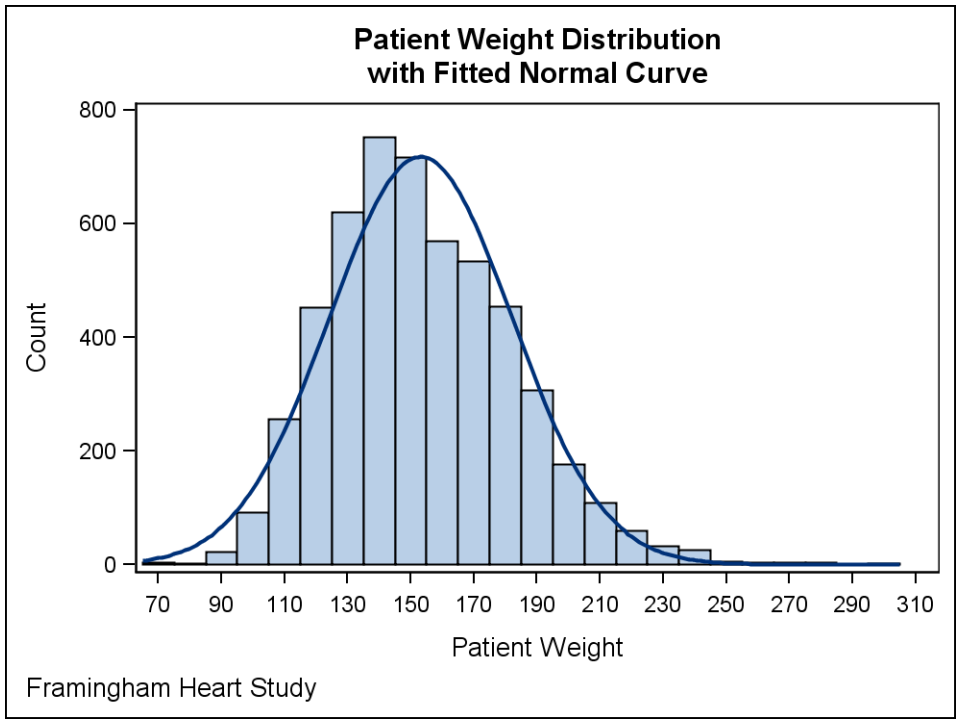


Example Program

When used as a stand-alone plot or overlaid with other density plots, 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 density plots 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, the NORMAL() option 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 ith 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 documentation for Base SAS.

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 as follows:

$$\begin{array}{lll} \text{Normal} & K_0(t) = \frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2}t^2) & \text{for } -\infty < t < \infty \\ \text{Quadratic} & K_0(t) = \frac{3}{4}(1-t^2) & \text{for } |t| \leq 1 \\ \text{Triangular} & K_0(t) = 1 - |t| & \text{for } |t| \leq 1 \end{array}$$

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 documentation for Base SAS.

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 for a 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 curve label is displayed

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 849 for the syntax on using a *style-element* and "Text Options" on page 852 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 146.

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 ignored if CURVELABELLOCATION=INSIDE is specified. The START and END settings are ignored 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 146.

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* 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 for the legend item that is associated with this plot.

Default: The *string* specified on the NAME= option.

Restriction: This option applies only to an associated DISCRETELEGEND statement.

LINEATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)
specifies the attributes of the density curve. See "General Syntax for Attribute Options" on page 849 for the syntax on using a *style-element* and "Line Options" on page 850 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 562.

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. Use *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. Use *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 557.

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 557.

Chapter 26

DROPLINE Statement

Description and Syntax

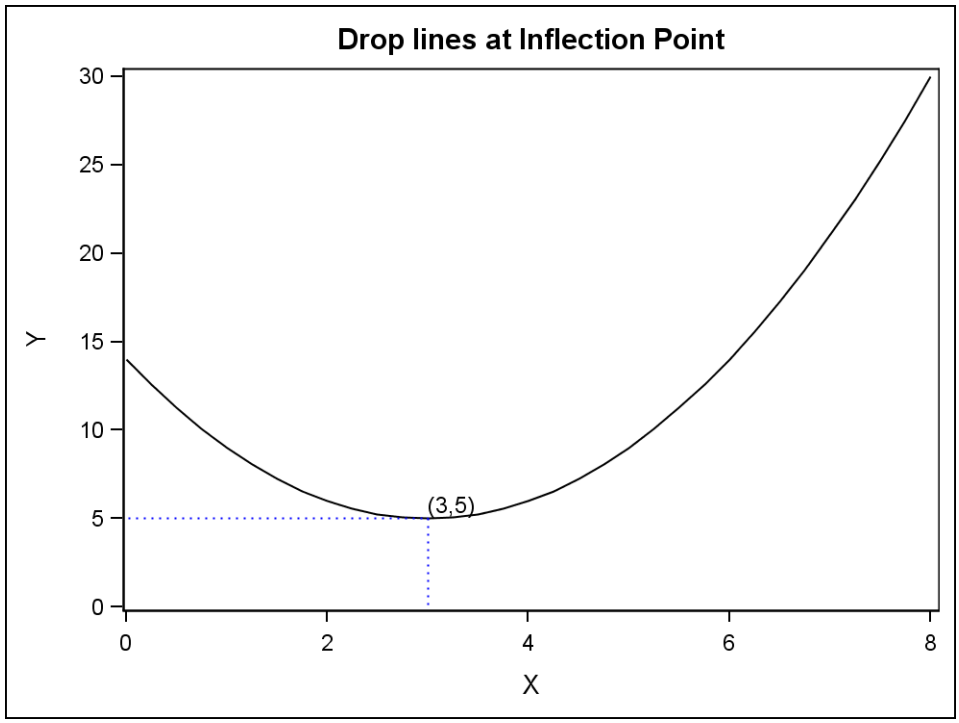
Creates a horizontal or vertical drop line from a point to an axis.

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 296:



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 used only 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).

Statement Option	Description
DISCRETEOFFSET	Specifies an amount to offset all drop lines from discrete X values, or discrete Y values, or both.
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 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 discrete X values, or Y values, or both.

Default: 0 (no offset, all drop lines are centered on discrete X values, or discrete Y values, or both)

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"* | *string-column*
specifies a label for the point(s) indicated by the X= and Y= arguments.

Default: no label is specified

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 849 for the syntax on using a *style-element* and "Text Options" on page 852 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 the legend item that is associated with this plot.

Default: The *string* specified on the NAME= option.

Restriction: This option applies only to an associated DISCRETELEGEND statement.

LINEATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)
specifies the attributes of the drop line. See "General Syntax for Attribute Options" on page 849 for the syntax on using a *style-element* and "Line Options" on page 850 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 557.

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 557.

Chapter 27

ELLIPSE Statement

Description and Syntax

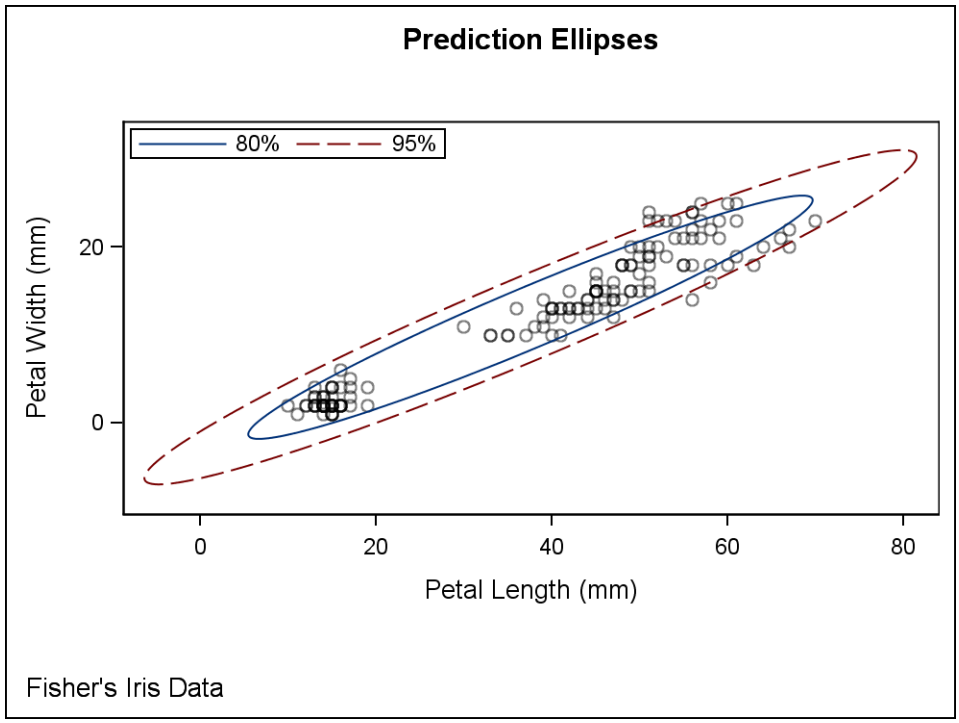
Creates a confidence ellipse computed from input data.

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 302:



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 label for a legend.
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)

Tip: The FILLATTRS= option can be used to set transparency for just the ellipse fill. You can combine this option with FILLATTRS= to set one transparency for the ellipse outline but a different transparency for the ellipse fill. Example:

```
datatransparency=0.2 fillattrs=(transparency=0.6)
```

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 849 for the syntax on using a *style-element* and "Fill Options" on page 850 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.

Tip: The DATATRANSPARENCY= option sets the transparency for the ellipse fill and ellipse outline. You can combine this option with DATATRANSPARENCY= to set one transparency for the outline but a different transparency for the fill. Example:

```
datatransparency=0.2 fillattrs=(transparency=0.6)
```

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 a label for the legend item that is associated with this plot.

Default: The *string* specified on the NAME= option.

Restriction: This option applies only to an associated DISCRETELEGEND statement.

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 849 for the syntax on using a *style-element* and "Line Options" on page 850 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 303.

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 557.

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 557.

Chapter 28

ELLIPSEPARM Statement

Description and Syntax

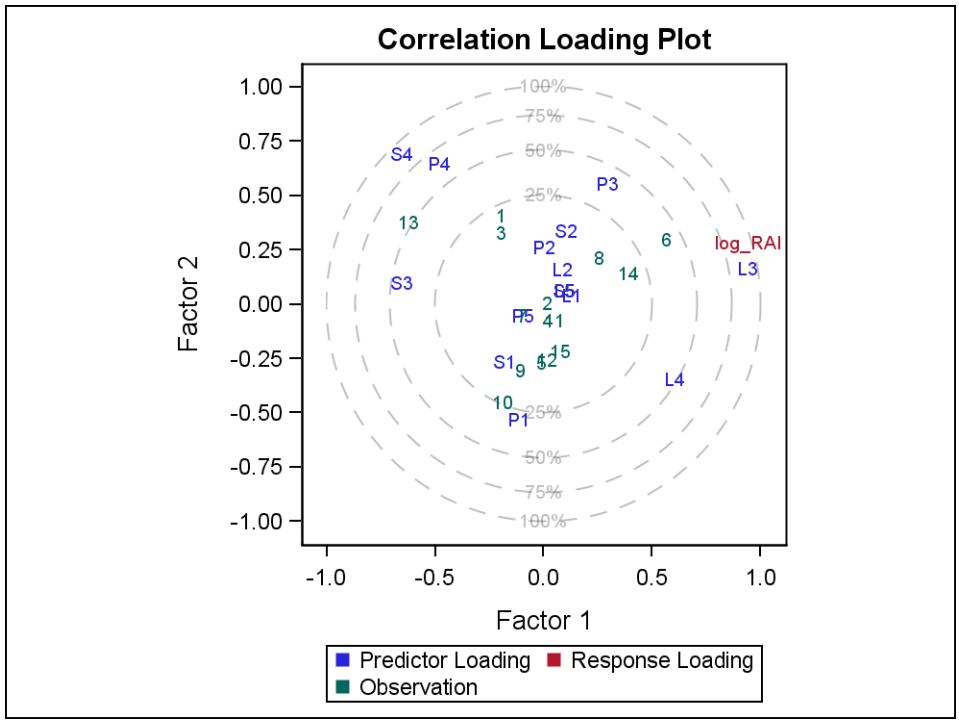
Creates an ellipse specified by slope, axis, and origin parameters.

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 310:



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 SAS/STAT user's guide.

```
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 301 for information about computed ellipses.

An ELLIPSEPARM statement can be used only within a two-dimensional 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.
INCLUDEMISSINGGROUP	Specifies whether missing values of the group variable are included in the plot.
INDEX	Specifies indices for mapping line attributes (color and pattern) to one of the GraphData1–GraphDataN style elements.
LEGENDLABEL	Specifies a label for a legend.

Statement Option	Description
NAME	Assigns a name to a plot statement so that it can be referenced in other template statements.
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)

Tip: The FILLATTRS= option can be used to set transparency for just the ellipse fill. You can combine this option with FILLATTRS= to set one transparency for the ellipse outline but a different transparency for the ellipse fill. Example:

```
datatransparency=0.2 fillattrs=(transparency=0.6)
```

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 849 for the syntax on using a *style-element* and "Fill Options" on page 850 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.

Tip: The DATATRANSPARENCY option sets the transparency for the ellipse fill and ellipse outline. You can combine this option with DATATRANSPARENCY= to set one transparency for the outline but a different transparency for the fill. Example:

```
datatransparency=0.2 fillattrs=(transparency=0.6)
```

GROUP=*column* | *discrete-attr-var* | *expression*
creates a separate ellipse for each unique group value of the specified column.

discrete-attr-var
specifies a discrete attribute variable that is defined in a DISCRETEATTRVAR statement.

Restriction: A discrete attribute variable specification must be a direct reference to the attribute variable. It cannot be set by a dynamic variable.

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.

Restriction: This option can be used only when a numeric column is specified for the SLOPE= , SEMIMAJOR= , and SEMIMINOR= arguments.

Interaction: The INCLUDEMISSINGGROUP option controls whether missing group values are considered a distinct group value.

Tip: 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.

INCLUDEMISSINGGROUP=*boolean*
specifies whether missing values of the group variable are included in the plot.

Default: TRUE

Interaction: For this option to have any effect, the GROUP= option must also be specified.

Tip: Unless a discrete attribute map is in effect or the INDEX= option is used, the attributes of the missing group value are determined by the GraphMissing style element except when the MISSING= system option is used to specify a non-default missing character or when a user-defined format is applied to the missing group value. In those cases, the attributes of the missing group value are determined by a GraphData1–GraphDataN style element.

INDEX=*numeric-column* | *expression*
specifies indices for mapping line attributes (color and pattern) and fill attributes to one of the GraphData1–GraphDataN style elements.

Default: no default

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.

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 144.

LEGENDLABEL= *"string"*
specifies a label for the legend item that is associated with this plot.

Default: The *string* specified on the NAME= option.

Restriction: This option applies only to an associated DISCRETELEGEND statement.

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 849 for the syntax on using a *style-element* and "Line Options" on page 850 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 557.

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 557.

Chapter 29

FRINGEPLOT Statement

Description and Syntax

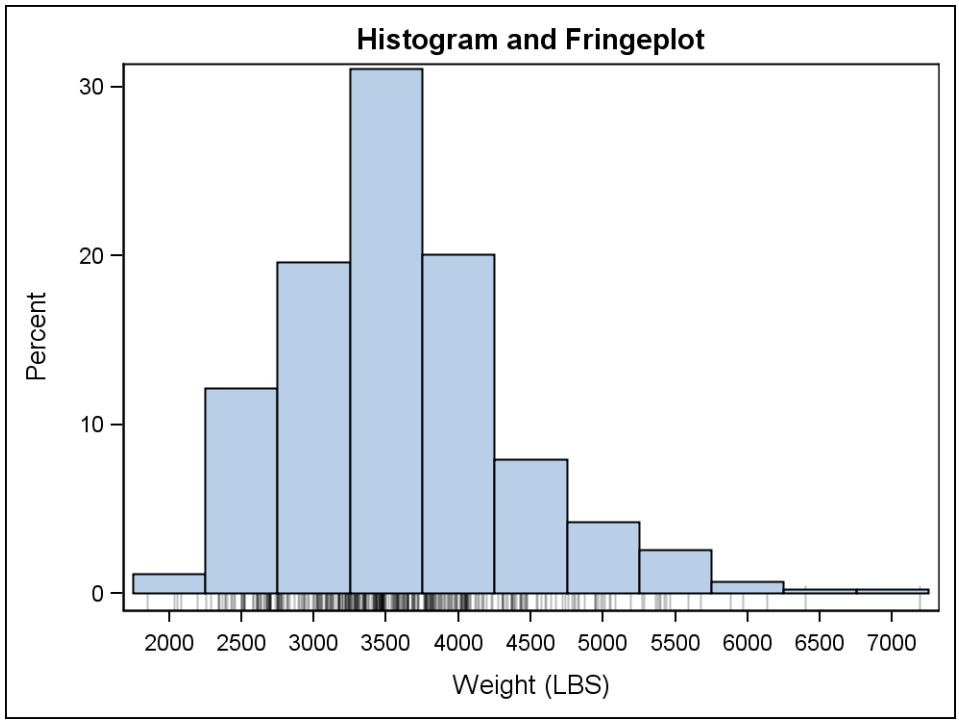
Creates a fringe plot on the X axis of an X-Y plot.

FRINGEPLOT *numeric-column | expression </option(s)>*;

Example Program and Statement Details

Example Graph

The following graph was generated by the "Example Program" on page 320:



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 user-defined roles that can be used to display information in the tooltips.
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 the legend item that is associated with this plot.

Default: The X-variable label. If a label is not defined, the X-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 fringe lines for the data points. See "General Syntax for Attribute Options" on page 849 for the syntax on using a *style-element* and "Line Options" on page 850 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 that can be used to display information in the tooltips.

Default: no user-defined roles

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 role X.

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.

role-list
an ordered, blank-separated list of unique FRINGEPLOT and user-defined roles. The FRINGEPLOT role is X.

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 557.

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 30

HEATMAPPARM Statement

Description and Syntax

Creates a two-dimensional plot that represents the values of three variables. Generating an X, Y grid of rectangles from the values of two independent variables, it colors the rectangles to represent the values of a third variable, which can be a response variable or a group variable.

The following syntax is used when the colors are determined by a grouping variable:

HEATMAPPARM X = *column* | *expression*

Y = *column* | *expression*

COLORGROUP = *column* | *discrete-attr-var* | *expression* < /*option(s)*>;

The following syntax is used when the colors are determined by a response variable that has continuous, numeric values:

HEATMAPPARM X = *column* | *expression*

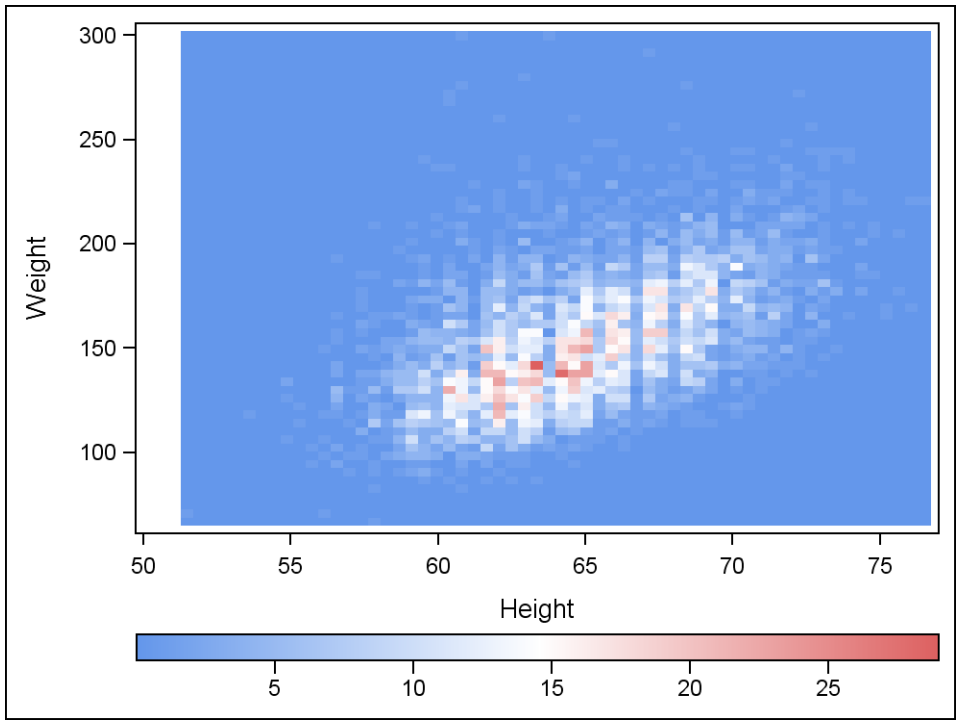
Y = *column* | *expression*

COLORRESPONSE = *numeric-column* | *range-attr-var* | *expression* < /*option(s)*>;

Example Program and Statement Details

Example Graph

The following graph was generated by the "Example Program" on page 326:



Example Program

```
proc template;
  define statgraph heatmapparm;
    begingraph;
      layout overlay;
        heatmapparm x=height y=weight colorresponse=count /
          name="heatmapparm" xbinaxis=false ybinaxis=false;
        continuouslegend "heatmapparm" / location=outside valign=bottom;
      endlayout;
    endgraph;
  end;
run;

proc sgrender data=sashelp.gridded template=heatmapparm;
run;
```

Statement Summary

A heat map is useful for visualizing the magnitude of a response variable in relation to two independent variables. For example, in molecular biology, heat maps can be used to track the expression of genes across multiple sample studies. In the HEATMAPPARM statement, you specify the independent variables in the X and Y arguments. For a response variable that has discrete values, use the COLORGROUP argument, or for a response variable that has interval values, use the COLORRESPONSE argument. For interval response variables, you can use the COLORMODEL= option to define the color ramp that is used to fill the rectangles.

Arguments

X=*column* | *expression*
specifies the column for the X values.

Y=*column* | *expression*
specifies the column for the Y values.

COLORGROUP =*column* | *discrete-attr-var* | *expression*
specifies a column that is used to discretely color the regions in the heat map. Required when the response variable is of type discrete.

discrete-attr-var
specifies a discrete attribute variable that is defined in a DISCRETEATTRVAR statement.

Restriction: A discrete attribute variable specification must be a direct reference to the attribute variable. It cannot be set by a dynamic variable.

COLORRESPONSE =*numeric-column* | *range-attr-var* | *expression*
specifies a numeric column that is used to color the regions of the heat map. The colors for each region are computed by mapping the values of this column to a color ramp that is defined by the COLORMODEL= option. This argument is required when the response variable is of type interval.

range-attr-var
specifies a range attribute variable that is defined in a RANGEATTRVAR statement.

Restriction: A range attribute variable specification must be a direct reference to the attribute variable. It cannot be set by a dynamic variable.

Options

Statement Option	Description
COLORMODEL	Specifies a style element to be used with the COLORRESPONSE= argument.
DATATRANSPARENCY	Specifies the degree of the transparency of the filled rectangles.
DISPLAY	Specifies whether to display outlined colored rectangles or just colored rectangles.
FILLATTRS	Specifies the transparency of the area fill in the rectangles.
NAME	Assigns a name to a plot statement for reference in other template statements.
OUTLINEATTRS	Specifies the appearance of the outlines of the filled rectangles.

Statement Option	Description
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 that is specified by the COLORMODEL= option.
ROLENAME	Specifies user-defined roles that can be used to display information in the tooltips.
TIP	Specifies the information to display when the cursor is positioned over a filled rectangle.
TIPFORMAT	Specifies display formats for information defined by the tooltip roles.
TIPLABEL	Specifies display labels for information defined by the tooltip roles.
XAXIS	Specifies whether data are mapped to the primary X (bottom) axis or the secondary X2 (top) axis.
XBINAXIS	Specifies whether to use bins as the basis for X-axis tick marks.
XBOUNDARIES	Specifies the boundaries of the X-value bins.
XENDLABELS	Specifies whether the axis ticks and value labels are drawn at the endpoints of the bins or midpoints of the bins.
XGAP	Specifies the amount of horizontal space on either side of each filled rectangle.
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.
YBINAXIS	Specifies whether to use bins as the basis for Y-axis tick marks.
YBOUNDARIES	Specifies the boundaries of the Y-value bins.
YENDLABELS	Specifies whether the axis ticks and value labels are drawn at the endpoints of the bins or midpoints of the bins.
YGAP	Specifies the amount of vertical space on either side of each filled rectangle.

Statement Option	Description
YVALUES	Specifies whether the Y values represent lower endpoints, midpoints, or upper endpoints of the bins.

COLORMODEL=*style-element*
specifies a style element to be used with the COLORRESPONSE= argument.

Default: The ThreeColorRamp style element. The STARTCOLOR attribute represents low response values and the ENDCOLOR attribute represents high response values.

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 COLORRESPONSE variable.
NEUTRALCOLOR	Specifies a color for the midpoint of the range of the COLORRESPONSE variable.
ENDCOLOR	Specifies a color for the highest data value of the COLORRESPONSE variable.

Interaction: The NEUTRALCOLOR attribute is not used for two-color ramps.

DATATRANSPARENCY=*number*
specifies the degree of the transparency of the filled rectangles.

Default: 0

Range: 0 (opaque) to 1 (entirely transparent)

DISPLAY=STANDARD | ALL | (*display-options*)
specifies whether to display outlined, colored rectangles or just colored rectangles.

Default: STANDARD

STANDARD
displays colored rectangles

ALL
displays outlined, colored rectangles

(*display-options*)
a list of *display-options*, enclosed in parentheses. Currently, only the following *display-option* is supported:

OUTLINE	displays outlines around the filled rectangles (same as keyword ALL)

FILLATTRS=(TRANSPARENCY=*number*)
specifies the transparency of the area fill in the rectangles.

Default: The DATATRANSPARENCY= option value

Range: 0 (opaque) to 1 (entirely transparent)

Restriction: Only the TRANSPARENCY= sub option is honored. If a style element or any other fill suboption is specified, it is ignored by the HEATMAPPARM statement.

Note: The fill colors are determined by the COLORRESPONSE= or COLORGROUP= column.

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.

Tip: The specified name is used primarily in legend statements to indicate the colors associated with the response values.

OUTLINEATTRS=*style-element* | (*line-options*)
specifies the appearance of the outlines of the filled rectangles. See "General Syntax for Attribute Options" on page 849 for the syntax on using a *style-element* and "Line Options" on page 850 for available *line-options*.

Default: The ContrastColor and LineThickness attributes of the GraphOutlines style element.

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 562.

REVERSECOLORMODEL=*boolean*
specifies whether to reverse the gradient that is specified by the COLORMODEL= option.

Default: FALSE

ROLENAME= (*role-name-list*)
specifies user-defined roles that can be used to display information in the tooltips. Columns for the tooltip display are specified in the TIP= option.

Default: no user-defined roles

(*rolename-list*)
a blank-separated list of *rolename* = *column* pairs.

For example, ROLENAME=(TIP1=PCT) assigns the data 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, COLORGROUP, and COLORRESPONSE.

Interaction: For this option to take effect, the TIP= option must be used to add the roles in the role list to the tooltips.

TIP=(*role-list*)
specifies the information to display when the cursor is positioned over a filled rectangle. If this option is used, it replaces all the information displayed by default.

Roles for columns that do not contribute to the heat map 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, and COLORGROUP or COLORRESPONSE.

(*role-list*)
an ordered, blank-separated list of unique HEATMAPPARM and user-defined roles. HEATMAPPARM roles include X and Y.

The following example displays tooltips for the columns X and PCT. The PCT column is not assigned to any pre-defined HEATMAPPARM role, so it must first be assigned a role:

```
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 information defined by the tooltip roles.

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)
TIPFORMAT=(TIP1=PERCENT7.2)
```

Requirement: Only the roles that appear in the TIP= option are used.

Requirement: Columns must be assigned to the roles for this option to have any effect. See ROLENAME=.

Tip: This option provides a way to control the formats of columns that appear in tooltips.

TIPLABEL=(*role-label-list*)
specifies display labels for information defined by the tooltip roles.

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)
TIPLABEL=(TIP1="Percent")
```

Requirement: Only the roles that appear in the TIP= option are used.

Requirement: Columns must be assigned to the roles for this option to have any effect. See ROLENAME=.

Tip: This option provides a way to control the formats 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 more information, see "How Axis Features Are Determined" on page 557.

XBINAXIS= *boolean*
specifies whether to use bins as the basis for X-axis tick marks. If this option is set to FALSE, a standard axis is used, ignoring bin boundaries and midpoints.

Default: TRUE

Interaction: If this option is set to TRUE, then the XENDLABELS= option determines how the axis ticks and value labels are displayed. If this option is set to FALSE, the XENDLABELS= option is ignored.

Interaction: When this option is set tot TRUE, the axis ticks are in predetermined locations and are not changed if the parent layout specifies axis suboptions like TICKVALUELIST= and TICKVALUESEQUENCE=.

XBOUNDARIES= (*numeric-list*)
specifies the boundaries of the X-value bins.

Default: no default

Requirement: The *numeric-list* must be enclosed in parentheses and each item separated by a blank.

Interaction: This option is ignored if the X values are not numeric.

Tip: This option can be used to specify unequal bins.

Discussion: The keywords MIN and MAX can be used as one of the values in the list of boundaries. Keywords MIN and MAX indicate the minimum and maximum data values for the X variable. Example:

```
xboundaries=(MIN 20 200 250 MAX)
```

XENDLABELS= *boolean*
specifies whether the axis ticks and value labels are drawn at the endpoints of the bins or midpoints of the bins.

Default: FALSE. The axis ticks and values labels are drawn at the bin midpoints.

Interaction: If this option is set to FALSE, then the axis ticks and value labels are drawn at the bin midpoints, regardless of whether the XVALUES= option identifies the X data as endpoint values or midpoint values.

Interaction: This option is ignored in any of the following conditions:

- if the X values are not numeric
- If XBINAXIS=FALSE

XGAP= *number*
specifies the amount of horizontal space on either side of each filled rectangle.

Default: 0

XVALUES=MIDPOINTS | LEFTPOINTS | RIGHTPOINTS
specifies whether the X values represent lower endpoints, midpoints, or upper endpoints of the bins.

Default: MIDPOINTS

See Also: The XENDLABELS= 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: The overall plot specification and the layout type determine the axis display. For more information, see "How Axis Features Are Determined" on page 557.

YBINAXIS= *boolean*
specifies whether to use bins as the basis for Y-axis tick marks. If this option is set to FALSE, a standard axis is used, ignoring bin boundaries and midpoints.

Default: TRUE

Interaction: If this option is set to TRUE, then the YENDLABELS= option determines how the axis ticks and value labels are displayed. If this option is set to FALSE, the YENDLABELS= option is ignored.

Interaction: When this option is set to TRUE, the axis ticks are in predetermined locations and are not changed if the parent layout specifies axis suboptions like TICKVALUELIST= and TICKVALUESEQUENCE=.

YBOUNDARIES= (*numeric-list*)
specifies the boundaries of the Y-value bins.

Default: no default

Requirement: The *numeric-list* must be enclosed in parentheses and each item separated by a blank.

Interaction: This option is ignored if the Y values are not numeric.

Tip: This option can be used to specify unequal bins.

Discussion: The keywords MIN and MAX can be used as one of the values in the list of boundaries. Keywords MIN and MAX indicate the minimum and maximum data values for the Y variable. Example:

```
yboundaries=(MIN 20 200 250 MAX)
```

YENDLABELS= *boolean*
specifies whether the axis ticks and value labels are drawn at the endpoints of the bins or midpoints of the bins.

Default: FALSE. The axis ticks and values labels are drawn at the bin midpoints.

Interaction: If this option is set to FALSE, then the axis ticks and value labels are drawn at the bin midpoints, regardless of whether the YVALUES= option identifies the Y data as endpoint values or midpoint values.

Interaction: This option is ignored in any of the following conditions:

- if the Y values are not numeric
- If YBINAXIS=FALSE

YGAP= *number*
specifies the amount of vertical space on either side of each filled rectangle.

Default: 0

YVALUES=MIDPOINTS | LEFTPOINTS | RIGHTPOINTS
specifies whether the Y values represent lower endpoints, midpoints, or upper endpoints of the bins.

Default: MIDPOINTS

See Also: The YENDLABELS= option.

Chapter 31

HIGHLOWPLOT Statement

Description and Syntax

Creates a display of floating vertical or horizontal lines or bars that connect the minimum and maximum response values for each value of a categorical variable.

The following syntax creates a vertical high-low chart, which would typically be used in the financial industry to plot stock values over time:

HIGHLOWPLOT X = *column* | *expression*
LOW = *numeric-column* | *expression*
HIGH = *numeric-column* | *expression* </*option(s)*>;

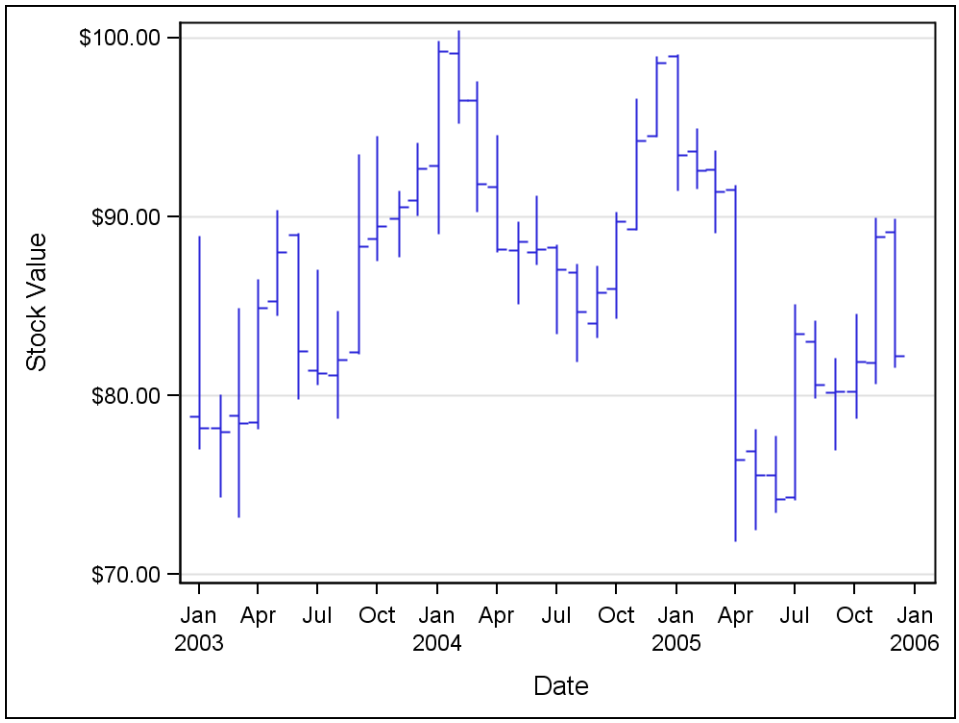
The following syntax creates a horizontal high-low chart, which would typically be used in the Health and Life Sciences industry to display over time the duration of adverse events or of adverse reactions to medication:

HIGHLOWPLOT Y = *column* | *expression*
LOW = *numeric-column* | *expression*
HIGH = *numeric-column* | *expression* </*option(s)*>;

Example Program and Statement Details

Example Graph 1

The following vertical high-low chart was generated by "Example Program 1" on page 336:

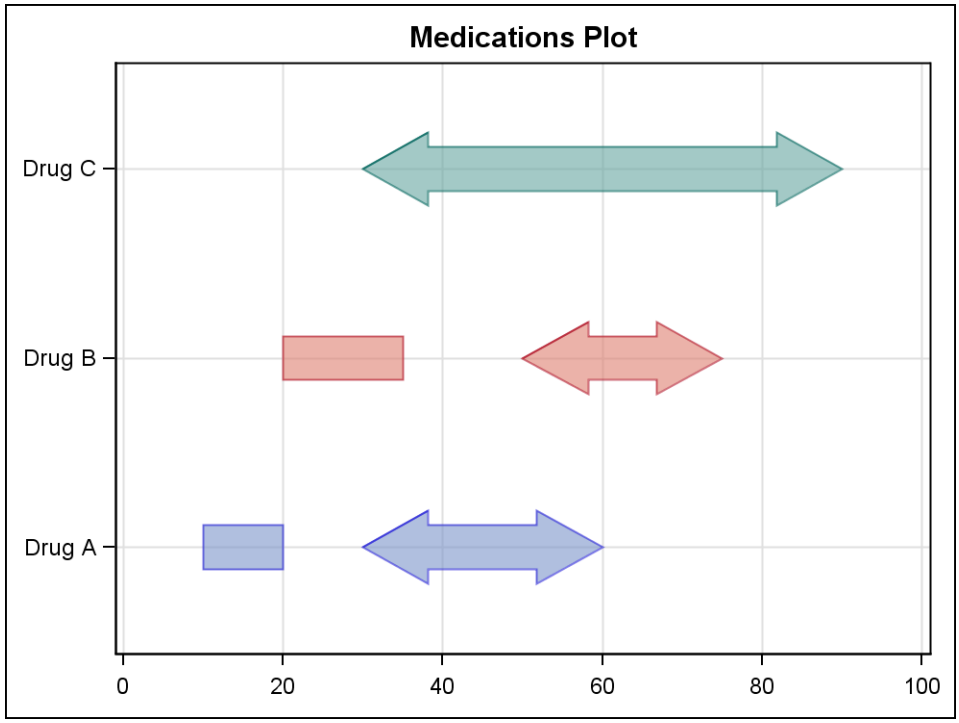


Example Program 1

```
proc template;
define statgraph highlow;
  begingraph;
    layout overlay /
        yaxisopts=(griddisplay=on label="Stock Value");
      highlowplot x=date high=high low=low /
        open=open close=close;
    endlayout;
  endgraph;
end;

proc sgrender data=sashelp.stocks template=highlow;
  where date >= '01JAN03'd and stock="IBM";
run;
```

Example Graph 2

The following horizontal high-low chart was generated by "Example Program 2" on page 337 :



Example Program 2

```
data highlowbar;
  length cap $ 12;
  input drug $ 1-10 low high cap $;
  datalines;
Drug A     10   20  NONE
Drug A     30   60  FILLEDARROW
Drug B     20   35  NONE
Drug B     50   75  FILLEDARROW
Drug C     30   90  FILLEDARROW
;
proc template;
define statgraph highlowbar;
  begingraph;
    entrytitle 'Medications Plot';
    layout overlay /
      xaxisopts=(
        griddisplay=on
        linearopts=(viewmin=0 viewmax=100)
        display=(line ticks tickvalues))
      yaxisopts=(
        griddisplay=on
        display=(line ticks tickvalues));
      highlowplot y=drug high=high low=low /
```

```
            group=drug outlineattrs=(pattern=solid)
            datatransparency=0.4
            type=bar barwidth=0.4
            highcap=cap lowcap=cap;
        endlayout;
      endgraph;
    end;

    proc sgrender data=highlowbar template=highlowbar;
    run;
```

Statement Summary

A high-low chart specifies that floating vertical or horizontal lines or bars connect the minimum and maximum response values for each value of a categorical variable. The data should have at least two response values for every category value. Otherwise, the single value is displayed without the connecting line or bar. In the statement syntax, the X and Y arguments can specify a column with character or numeric values. The LOW and HIGH arguments must specify a numeric column.

The HIGHLOWPLOT statement can be used to create a graph showing the high and low response values for observations along a time axis. The independent variable can be interval or discrete. The HIGHLOWPLOT has two common uses:

- One typical use of a high-low graph is in the financial industry to plot stock values by day. The OPEN= and CLOSE= options are typically used in the graph, and the TYPE= option is set to LINE (the default), as illustrated in “Example Graph 1” on page 336 .
- Another typical use of a high-low graph is in the Health and Life Sciences industry to display over time the duration of adverse events or of adverse reactions to medication. The HIGHCAP= and LOWCAP= options are typically used in the graph, and the TYPE= option is set to BAR, as illustrated in “Example Graph 2” on page 337 .

For both vertical and horizontal high-low charts, the data values can be grouped using the GROUP= option. For grouped data, the data values are not summarized. Each observation is plotted independently, and the line or bar segment for each category value can use different display characteristics.

Arguments

Either the X= or Y= argument must be specified, but you cannot specify both on the same HIGHLOWPLOT statement. The HIGH= and LOW= arguments are required.

X=*column* | *expression*
specifies a column or expression representing the values in a vertical high-low chart. The values can be character or numeric.

Y=*column* | *expression*
specifies a column or expression representing the values in a horizontal high-low chart. The values can be character or numeric.

LOW=*numeric-column* | *expression*
specifies a column or expression representing the values of the lower end of the floating bar or line.

HIGH=*numeric-column* | *expression*
specifies a column or expression representing the values of the higher end of the floating bar or line.

Options

Statement Option	Description
BARWIDTH	Specifies the width of a bar as a ratio of the maximum possible width.
CLOSE	Specifies a column or expressions whose values are used to display a closing-value indicator on the bar or line.
CLUSTERWIDTH	Specifies the width of the group clusters as a fraction of the midpoint spacing.
DATATRANSPARENCY	Specifies the degree of the transparency of the colored regions and the outlines around them.
DISCRETEOFFSET	Specifies an amount to offset all bars or lines from the category midpoints when graphing multiple response variables side by side on a common axis.
DISPLAY	Specifies whether to display outlined colored regions or just colored regions.
ENDCAPDISPLAYPOLICY	Specifies the policy for displaying end caps, when present.
FILLATTRS	Specifies the appearance of the interior fill area of the bars.
GROUP	Creates a distinct set of floating bars or lines for each unique group value of the specified column.
GROUPDISPLAY	Specifies whether grouped bars or lines are overlaid or clustered around the category midpoints.
GROUPORDER	Specifies the sort order for grouped bars or lines.
HIGHCAP	Specifies the type of cap to use at the high end of the bar or line.
HIGHLABEL	Specifies the label to display at the high end of the bar or line.
INCLUDEMISSINGGROUP	Specifies whether missing values of the group variable are included in the plot.

Statement Option	Description
INDEX	Specifies indices for mapping color and line patterns to group values.
INTERVALBARWIDTH	Specifies the width of the floating bars.
LABELATTRS	Specifies the color and font attributes of the labels for the bars or lines.
LINEATTRS	Specifies the attributes of the floating plot lines when the TYPE= option is set to LINE.
LOWCAP	Specifies the type of cap used at the low end of the bar or line.
LOWLABEL	Specifies the label to appear at the low end of each floating bar or line.
NAME	Assigns a name to a plot statement for reference in other template statements.
OPEN	Specifies a column or expressions whose values are used to display an opening-value indicator on the bar or line.
OUTLINEATTRS	Specifies the appearance of the outlines of the filled regions.
PRIMARY	Specifies that the data columns for this plot and the plot type be used for determining default axis features.
ROLENAME	Specifies user-defined roles that can be used to display information in the tooltips.
TIP	Specifies the information to display when the cursor is positioned over a bar or line.
TIPFORMAT	Specifies display formats for information defined by the tooltip roles.
TIPLABEL	Specifies display labels for information defined by the tooltip roles.
TYPE	Specifies whether data values should be represented by bars or lines.
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: 0.85 of the tick spacing. The bar width automatically adjusts, based on the number of bars to be displayed and the wall width.

Range: 0.1 (narrowest) to 1.0 (widest)

Restriction: This option is available only when TYPE= BAR.

Tip: This option is needed only to change the default behavior. For example, to remove any inner-bar gap, set BARWIDTH=1.

Tip: See the DISCRETEOFFSET= option for examples of using this option.

CLOSE=*numeric-column* | *expression*
specifies a column or expressions whose values are used to display a closing-value indicator. This option is typically used when TYPE=LINE, but it can be used when TYPE=BAR.

Default: no default

Discussion:

- For vertical high-low charts, the value is represented by a short horizontal line extending from the side that displays the higher X values
- For horizontal high-low charts, the value is represented by a short vertical line extending from the side that displays the higher Y values.

CLUSTERWIDTH= *number*
specifies the width of the group clusters as a fraction of the midpoint spacing on a discrete axis or a fraction of the minimum interval between adjacent data values on an interval axis.

Default: 0.85

Range: 0.1 (narrowest) to 1 (widest)

Interaction: For this option to take effect, the GROUP= option must also be specified, and the GROUPDISPLAY= option must be set to CLUSTER.

DATATRANSPARENCY=*number*
specifies the degree of the transparency of the colored regions.

Default: 0

Range: 0 (opaque) to 1 (entirely transparent)

Tip: The FILLATTRS option can be used to set transparency for just the colored regions. You can combine this option with FILLATTRS= to set one transparency for the outlines but a different transparency for the region fills . Example:

```
datatransparency=0.2 fillattrs=(transparency=0.6)
```

DISCRETEOFFSET=*number*
specifies an amount to offset all bars or lines from the category midpoints when graphing multiple response variables side by side on a common axis.

Default: 0 (no offset, all bars or lines 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 for a vertical plot and up for a horizontal plot. (If the layout's axis options set REVERSE=TRUE, then the offset direction is also reversed.)

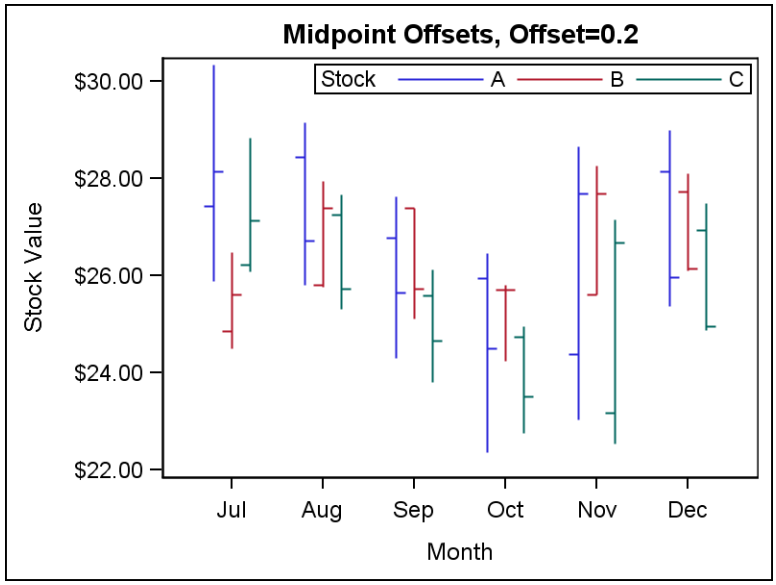
Interaction: This option is available only for a discrete X or discrete Y axis.

Discussion: 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 HIGHLOWPLOT statements are used with different response variables, the bars or lines for matching category values are centered on the midpoints and superimposed on each other. In those cases, you can make it easier to distinguish among superimposed bars or lines by assigning a different width for them in each HIGHLOWPLOT statement in the overlay.

If you prefer to avoid superimposed bars or lines, you can assign a different offset to each HIGHLOWPLOT statement. If desired, you can adjust the width of the bar or line in conjunction with DISCRETEOFFSET= to create narrower bars that require less width within the plot area.

The following example defines offsets for the lines in a stock report:

```
layout overlay / cycleattrs=true
    yaxisopts=(label="Stock Value");
  highlowplot x=month high=a_high low=a_low /
    close=a_close open=a_open
    legendlabel="A" name="ahighlow" lineattrs=(pattern=solid)
    discreteoffset=-0.2 ;
  highlowplot x=month high=b_high low=b_low /
    close=b_close open=b_open
    legendlabel="B" name="bhighlow" lineattrs=(pattern=solid)
    discreteoffset=0 ;
  highlowplot x=month high=c_high low=c_low /
    close=c_close open=c_open
    legendlabel="C" name="chighlow" lineattrs=(pattern=solid)
    discreteoffset=+0.2 ;
  discretelegend "ahighlow" "bhighlow" "chighlow" / title="Stock"
    location=inside halign=right valign=top;
endlayout;
```



DISPLAY=STANDARD | ALL | (*display-options*)
specifies whether to display outlined colored regions or just colored regions.

Default: STANDARD

STANDARD
displays outlined, colored regions

ALL
displays outlined, colored regions

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

FILL	displays filled (colored) regions
OUTLINE	displays outlined regions

Interaction: For this option to take effect, TYPE=BAR must also be specified.

ENDCAPDISPLAYPOLICY= AUTO | ALWAYS
specifies the policy for displaying end caps, when present.

Note: This feature is for the second maintenance release of SAS 9.3 and later. See "What's New in the SAS 9.3 Graph Template Language" on page xi.

AUTO
only draw the end caps for the elements with a low to high range that is large enough to accommodate the end caps. Draw just the bar or line for the rest.

ALWAYS
always draw the end caps.

Note: When the low to high range is not large enough for the end caps, the end caps might overlap. The data ranges that are smaller than the end-cap size might not be resolvable beyond the rendered size of the end caps.

Default: AUTO

Interaction: This option is honored only when the TYPE=LINE option is in effect. See TYPE= on page 349.

Interaction: This option is ignored if the LOWCAP= or HIGHCAP= option is not set or is effectively set to NONE. See LOWCAP= on page 347 and HIGHCAP= on page 345.

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 849 for the syntax on using a *style-element* and "Fill Options" on page 850 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 take effect, TYPE=BAR must also be specified.

Tip: The DATATRANSPARENCY= option sets the transparency for the colored regions and the outlines around them. You can combine this option with DATATRANSPARENCY= to set one transparency for the outlines but a different transparency for the colored regions. Example:

```
datatransparency=0.2 fillattrs=(transparency=0.6)
```

GROUP=*column* | *discrete-attr-var* | *expression*
creates a distinct set of floating bars or lines for each unique group value of the specified column.

discrete-attr-var
specifies a discrete attribute variable that is defined in a DISCRETEATTRVAR statement.

Restriction: A discrete attribute variable specification must be a direct reference to the attribute variable. It cannot be set by a dynamic variable.

Default:

- If 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. The fill colors are defined by the GraphData1:Color–GraphDataN:Color style references.
- For floating lines when TYPE=LINE. and for outlines when TYPE=BAR (if outlines are enabled by the ODS style or the DISPLAY= option), each distinct group value is represented in the graph by a different line color and line pattern. The colors and patterns are defined by the ContrastColor and LineStyle attributes of the GraphData1–GraphDataN style elements.

Interaction: If the X or Y role is discrete, then the bars or lines representing the group can be drawn in a cluster or overlaid, based on the setting for the GROUPDISPLAY= option. However, if the X or Y role is interval, then the lines representing the group are overlaid and the GROUPDISPLAY= option is ignored.

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.

Interaction: The INCLUDEMISSINGGROUP= option determines whether missing group values are considered a distinct group value.

Tip: The representations that are used to identify the groups can be overridden. For example, if each distinct group value is represented by a different line pattern, the LINEATTRS= (PATTERN=*pattern*) specification can be used to assign the same line pattern to all the plot lines. In that case, the line color denotes group values. Likewise, LINEATTRS= (COLOR= *color*) can be used to assign the same color to all lines, letting line pattern denote group values.

GROUPDISPLAY= OVERLAY | CLUSTER
specifies whether grouped bars or lines are overlaid or clustered around the category midpoints.

Default: OVERLAY

OVERLAY
bars or lines for matching category values are centered on the midpoints and superimposed on each other.

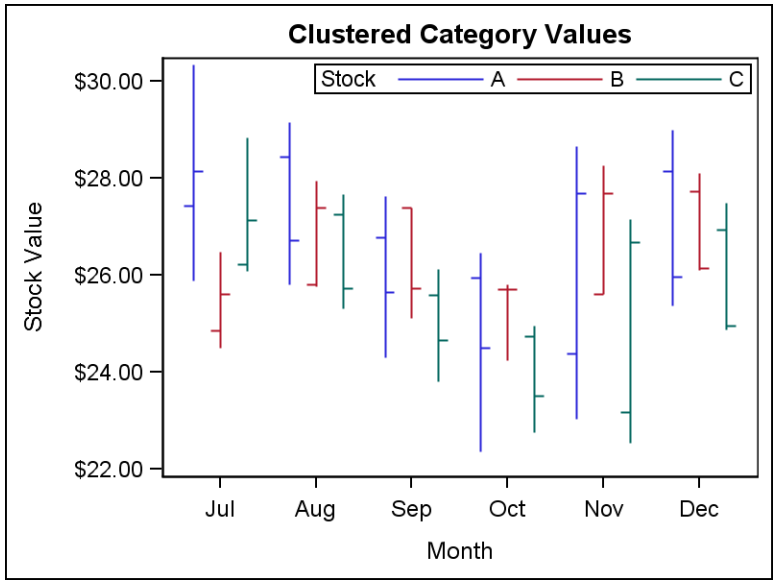
CLUSTER
bars or lines for matching category values are clustered around the midpoints. Each set of group values is centered at the midpoint for the category.

Interaction: For this option to take effect, the GROUP= option must also be specified.

The following example shows the effect of clustering the lines in a stock report when the category values are grouped into a single response variable. Note that if your category values are not grouped in the same variable but are stored in separate variables, you can get this same effect by using the DISCRETEOFFSET= option.

```
layout overlay /
    yaxisopts=(label="Stock Value");
  highlowplot x=month high=high low=low /
    close=close open=open
    legendlabel="Stock" name="cluster"
    group=stock groupdisplay=cluster
```

```
    lineattrs=(pattern=solid);
  discretelegend "cluster" / title="Stock"
    location=inside halign=right valign=top;
endlayout;
```



GROUPORDER=DATA | ASCENDING | DESCENDING
specifies the sort order for grouped bars or lines. It also sets the default order of the groups in 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: For this option to take effect, the GROUP= option must also be specified.

Note: Attributes such as color, symbol, and pattern are assigned to each group in the DATA order by default regardless of the GROUPORDER= option setting.

Note: The ASCENDING and DESCENDING settings linguistically sort the group values within each category (or X value) for display position purposes only. The data order of the observations and the visual attributes that are assigned to the group values remain unchanged.

HIGHCAP = *column* | NONE | SERIF | BARBEDARROW | FILLEDARROW | OPENARROW | CLOSEDARROW
specifies the type of cap used at the high end of the bar or line. This option is typically used when TYPE=BAR, but it can be used when TYPE=LINE.

Default: NONE

Interaction: All of the keywords can be specified for any high-low chart, but the effect of each keyword depends on the setting for the TYPE= option and also the fill state of the bars, when displayed:

- TYPE=BAR: If the bars are filled, FILLEDARROW is used for all settings other than NONE. If the bars are not filled, CLOSEDARROW is used.
- TYPE=LINE: If CLOSEDARROW is specified, it is treated as FILLEDARROW.

HIGHLABEL= *column* | *expression*
specifies the label to display at the high end of the bar or line.

INCLUDEMISSINGGROUP=*boolean*
specifies whether missing values of the group variable are included in the plot.

Default: TRUE

Interaction: For this option to take effect, the GROUP= option must also be specified.

Tip: Unless a discrete attribute map is in effect or the INDEX= option is used, the attributes of the missing group value are determined by the GraphMissing style element except when the MISSING= system option is used to specify a non-default missing character or when a user-defined format is applied to the missing group value. In those cases, the attributes of the missing group value are determined by a GraphData1–GraphDataN style element.

INDEX=*numeric-column* | *expression*
specifies indices for mapping color and line patterns to group values.

Default: no default

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.

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: 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 according to the setting of the GROUPORDER= option.

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 144.

INTERVALBARWIDTH= *dimension*
specifies the width of the floating bars.

Default: 85% of the smallest interval between any two bars for the given plot

Restriction: For this option to take effect, TYPE=BAR must be set, and the independent variable must be of type interval.

LABELATTRS=*style-element* | *style-element* (*text-options*) | (*text-options*)
specifies the color and font attributes of the labels for the bars or lines. See "General Syntax for Attribute Options" on page 849 for the syntax on using a *style-element* and "Text Options" on page 852 for available *text-options*.

Default:

- For non-grouped data, the GraphDataText style element
- For grouped data, the data label color changes to match the group color derived from the ContrastColor attribute of the GraphData1–GraphDataN style elements.

LINEATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)
specifies the attributes of the floating plot lines. See "General Syntax for Attribute Options" on page 849 for the syntax on using a *style-element* and "Line Options" on page 850 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.

Interaction: For this option to have any effect, TYPE= LINE must also be specified.

LOWCAP = *column* | NONE | SERIF | BARBEDARROW | FILLEDARROW | OPENARROW | CLOSEDARROW
specifies the type of cap used at the low end of the bar or line. This option is typically used when TYPE=BAR, but it can be used when TYPE=LINE.

Default: NONE

Interaction: All of the keywords can be specified for any high-low chart, but the effect of each keyword depends on the setting for the TYPE= option and also the fill state of the bars, when displayed:

- TYPE=BAR: If the bars are filled, FILLEDARROW is used for all settings other than NONE. If the bars are not filled, CLOSEDARROW is used.
- TYPE=LINE: If CLOSEDARROW is specified, it is treated as FILLEDARROW.

LOWLABEL=*column* | *expression*
specifies the label to appear at the low end of each floating bar or line.

Default: no default

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.

OPEN=*numeric-column* | *expression*
specifies a column or expressions whose values are used to display an opening-value indicator. This option is typically used when TYPE=LINE, but it can be used when TYPE=BAR.

Default: no default

Discussion:

- For vertical high-low charts, the value is represented by a short horizontal line extending from the side that displays the lower X values
- For horizontal high-low charts, the value is represented by a short vertical line extending from the side that displays the lower Y values.

OUTLINEATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)
specifies the appearance of the outlines of the filled regions. See "General Syntax for Attribute Options" on page 849 for the syntax on using a *style-element* and "Line Options" on page 850 for available *line-options*.

Default: The ContrastColor and LineThickness attributes of the GraphOutlines style element.

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 562.

ROLENAME= (*role-name-list*)
specifies user-defined roles that can be used to display information in the tooltips. Columns for the tooltip display are specified in the TIP= option.

Default: no user-defined roles

(*rolename-list*)
a blank-separated list of *rolename* = *column* pairs.

For example, ROLENAME=(TIP1=PCT) assigns the data 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 .

Interaction: For this option to take effect, the TIP= option must also be used.

TIP=(*role-list*)
specifies the information to display when the cursor is positioned over a bar or line. 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 and Y .

(*role-list*)
an ordered, blank-separated list of unique HIGHLOWPLOT and user-defined roles. HIGHLOWPLOT roles include X , Y , LOW , HIGH , OPEN , and CLOSE .

The following example displays tooltips for the columns X and PCT. The PCT column is not assigned to any pre-defined HIGHLOWPLOT role, so it must first be assigned a role:

```
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 information defined by the tooltip roles.

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)
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.

TIPLABEL=(*role-label-list*)
specifies display labels for information defined by the tooltip roles.

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)
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.

TYPE= LINE | BAR
specifies whether data values should be represented by bars or lines.

Default: LINE

BAR
uses fill and outline attributes.

LINE
uses line attributes.

XAXIS=X | X2
specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

Default: X

Interaction: When the Y= argument is specified, the HIGH= and LOW= arguments are mapped to the axis that is specified on this option (X or X2).

Interaction: The overall plot specification and the layout type determine the axis display. For more information, see "How Axis Features Are Determined" on page 557.

YAXIS=Y | Y2
specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

Default: Y

Interaction: When the X= argument is specified, the HIGH= and LOW= arguments are mapped to the axis that is specified on this option (Y or Y2).

Interaction: The overall plot specification and the layout type determine the axis display. For more information, see "How Axis Features Are Determined" on page 557.

Chapter 32

HISTOGRAM Statement

Description and Syntax

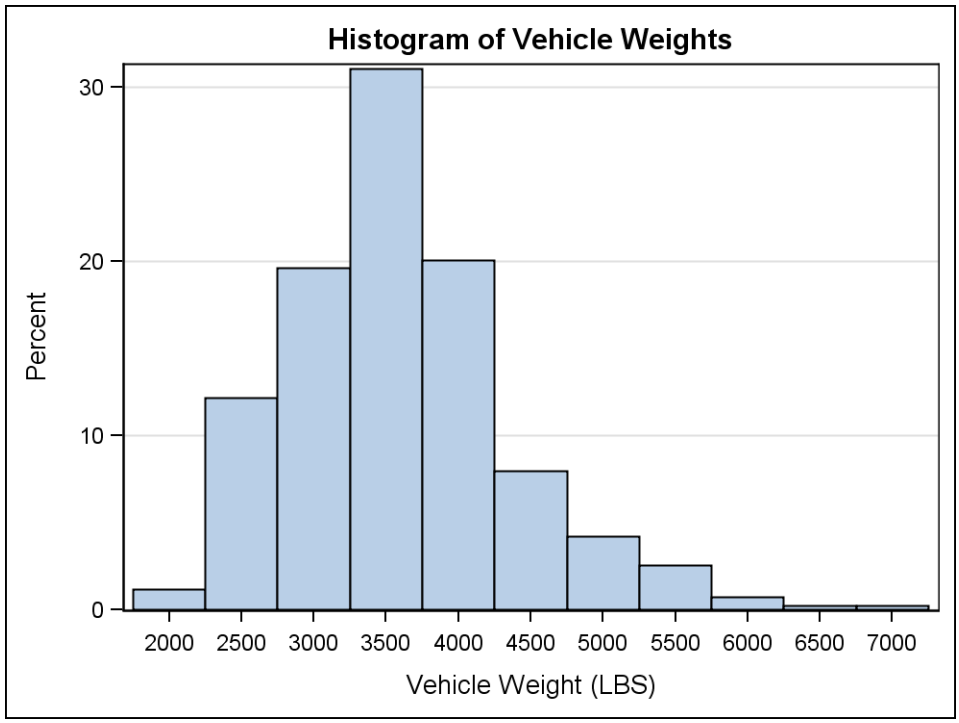
Creates a univariate histogram computed from input data.

HISTOGRAM *numeric-column | expression </option(s)>*;

Example Program and Statement Details

Example Graph

The following graph was generated by the "Example Program" on page 352:



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. By default, the X and Y axes of the histogram are linear. You can use the XAXISOPTS= and YAXISOPTS= options on the LAYOUT OVERLAY statement to change the axes properties.

Note: Log axes are not supported by the HISTOGRAM statement.

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 bar fills and bar outlines.
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 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.

Statement Option	Description
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.
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 562).

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 bar fills and bar outlines.

Default: 0

Range: 0 (opaque) to 1 (entirely transparent)

Tip: The FILLATTRS= option can be used to set transparency for just the bar fills. You can combine this option with FILLATTRS= to set one transparency for the bar outlines but a different transparency for the bar fills. Example:

```
datatransparency=0.2 fillattrs=(transparency=0.6)
```

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 at the bin midpoints.

TRUE	draws the axis ticks and value labels at the bin endpoints
FALSE	draws the axis ticks and value labels at the bin midpoints

Default: FALSE

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 849 for the syntax on using a *style-element* and "Fill Options" on page 850 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.

Tip: The DATATRANSPARENCY= option sets the transparency for bar fills and bar outlines. You can combine this option with DATATRANSPARENCY= to set one transparency for the outlines but a different transparency for the fills. Example:

```
datatransparency=0.2 fillattrs=(transparency=0.6)
```

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 the legend item that is associated with this plot.

Default: The *string* specified on the NAME= option.

Restriction: This option applies only to an associated DISCRETELEGEND statement.

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 849 for the syntax on using a *style-element* and "Line Options" on page 850 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.

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 562.

SCALE=PERCENT | COUNT | PROPORTION | DENSITY
specifies whether the Y axis displays the percentages between 0 and 100, or the frequency counts, the proportions between 0 and 1, or true density estimates.

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 557.

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 557.

Chapter 33

HISTOGRAMPARM Statement

Description and Syntax

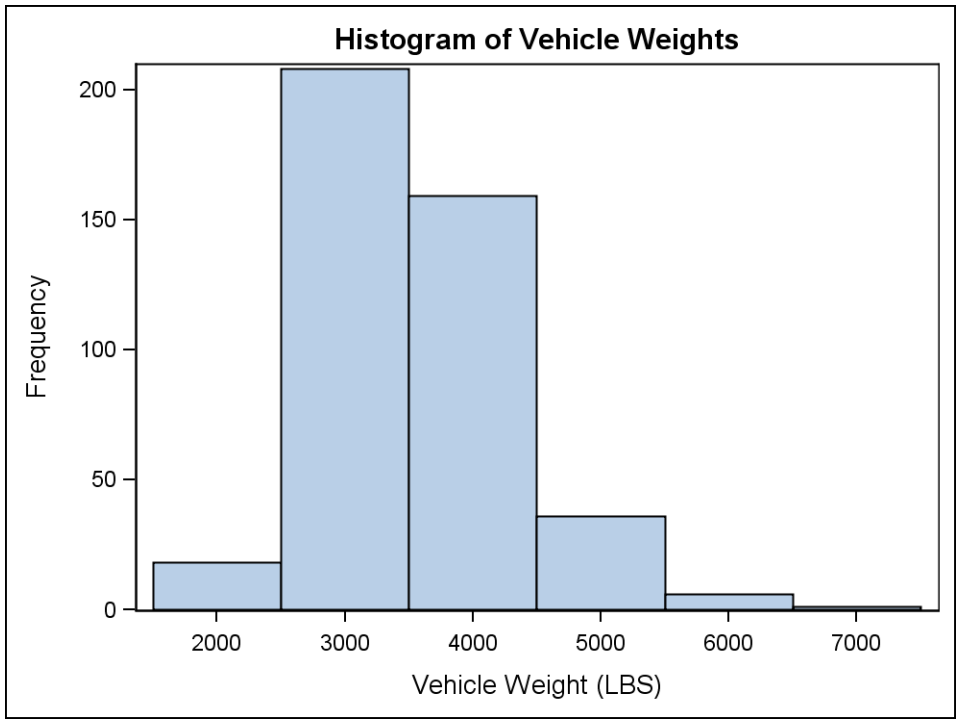
Creates a univariate histogram for specified values of bin midpoints and bin frequencies.

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 360:



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.
DATALABEL	Specifies a column for bar labels.
DATALABELATTRS	Specifies the color and font attributes of the bar labels.
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 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.

Statement Option	Description
PRIMARY	Specifies that the data columns for this plot be used for determining default axis features.
ROLENAME	Specifies user-defined roles that can be used to display information in the tooltips.
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.

Interaction: When this option is set to TRUE, some X-axis options that are set on the parent layout might not apply, such as TICKVALUELIST= and TICKVALUESEQUENCE=.

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 562).

DATALABEL= *column* | *expression*
specifies a column for the bar labels. The labels appear at the top or end of each bar, depending on the chart orientation.

Note: This feature is for the second maintenance release of SAS 9.3 and later. See "What's New in the SAS 9.3 Graph Template Language" on page xi.

Default: No data labels are displayed.

DATALABELATTRS=*style-element* | *style-element* (*text-options*) | (*text-options*)
specifies the color and font attributes of the bar labels. See "General Syntax for Attribute Options" on page 849 for the syntax on using a *style-element* and "Text Options" on page 852 for available *text-options*.

Note: This feature is for the second maintenance release of SAS 9.3 and later. See "What's New in the SAS 9.3 Graph Template Language" on page xi.

Default: The GraphDataText style element.

Interaction: The DATALABEL= option must be specified for this option to have any effect.

Interaction: When text options are specified, any font properties that are not specified (color, family, size, weight, and style) are derived from the GraphDataText style element.

DATATRANSPARENCY= *number*
specifies the degree of the transparency of the bars (outline and fill).

Default: 0

Range: 0 (opaque) to 1 (entirely transparent)

Tip: The FILLATTRS= option can be used to set transparency for just the bar fills. You can combine this option with FILLATTRS= to set one transparency for the bar outlines but a different transparency for the bar fills. Example:

```
datatransparency=0.2 fillattrs=(transparency=0.6)
```

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.

TRUE draws the axis ticks and value labels at the bin endpoints

FALSE draws the axis ticks and value labels at the bin midpoints

Default: FALSE

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 849 for the syntax on using a *style-element* and "Fill Options" on page 850 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.

Tip: The DATATRANSPARENCY= option sets the transparency for bar fills and bar outlines. You can combine this option with DATATRANSPARENCY= to set one transparency for the outlines but a different transparency for the fills. Example:

```
datatransparency=0.2 fillattrs=(transparency=0.6)
```

LEGENDLABEL= *"string"*
specifies a label for the legend item that is associated with this plot.

Default: The *string* specified on the NAME= option.

Restriction: This option applies only to an associated DISCRETELEGEND statement.

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 849 for the syntax on using a *style-element* and "Line Options" on page 850 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.

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 562.

ROLENAME=(*role-name-list*)
specifies user-defined roles that can be used to display information in the tooltips.

Default: no user-defined roles

(*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, Y, and DATALABEL.

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, Y, and DATALABEL.

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 557.

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 557.

Chapter 34

LINEPARM Statement

Description and Syntax

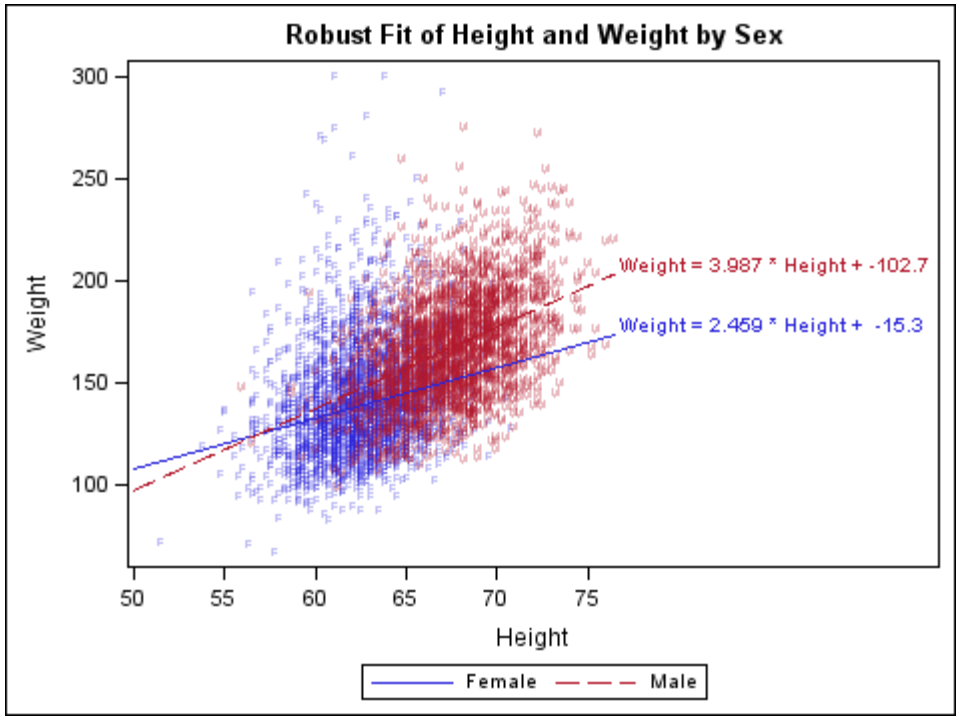
Creates a straight line specified by a point and a slope.

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 368:



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=0.4);
    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.
INCLUDEMISSINGGROUP	Specifies whether missing values of the group variable are included in the plot.
INDEX	Specifies indices for mapping line attributes (color and pattern) to one of the GraphData1 - GranphDataN style elements.
LEGENDLABEL	Specifies a label for 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 curve label is displayed

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 849 for the syntax on using a *style-element* and "Text Options" on page 852 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 146.

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 ignored 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 146.

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* | *discrete-attr-var* | *expression*
creates a separate parameterized line plot for each unique group value of the specified column.

discrete-attr-var
specifies a discrete attribute variable that is defined in a DISCRETEATTRVAR statement.

Restriction: A discrete attribute variable specification must be a direct reference to the attribute variable. It cannot be set by a dynamic variable.

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.

Interaction: The INCLUDEMISSINGGROUP option controls whether missing group values are considered a distinct group value.

Tip: 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.

INCLUDEMISSINGGROUP=*boolean*
specifies whether missing values of the group variable are included in the plot.

Default: TRUE

Interaction: For this option to take effect, the GROUP= option must also be specified.

Tip: Unless a discrete attribute map is in effect or the INDEX= option is used, the attributes of the missing group value are determined by the GraphMissing style element except when the MISSING= system option is used to specify a non-default missing character or when a user-defined format is applied to the missing group value. In those cases, the attributes of the missing group value are determined by a GraphData1–GraphDataN style element.

INDEX=*numeric-column* | *expression*
specifies indices for mapping line attributes (color and pattern) to one of the GraphData1 - GranphDataN style elements.

Default: no default

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.

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 144.

LEGENDLABEL= *"string"*
specifies a label for the legend item that is associated with this plot.

Default: The *string* specified on the NAME= option.

Restriction: This option applies only to an associated DISCRETELEGEND statement.

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 849 for the syntax on using a *style-element* and "Line Options" on page 850 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 557.

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 557.

Chapter 35

LOESSPLOT Statement

Description and Syntax

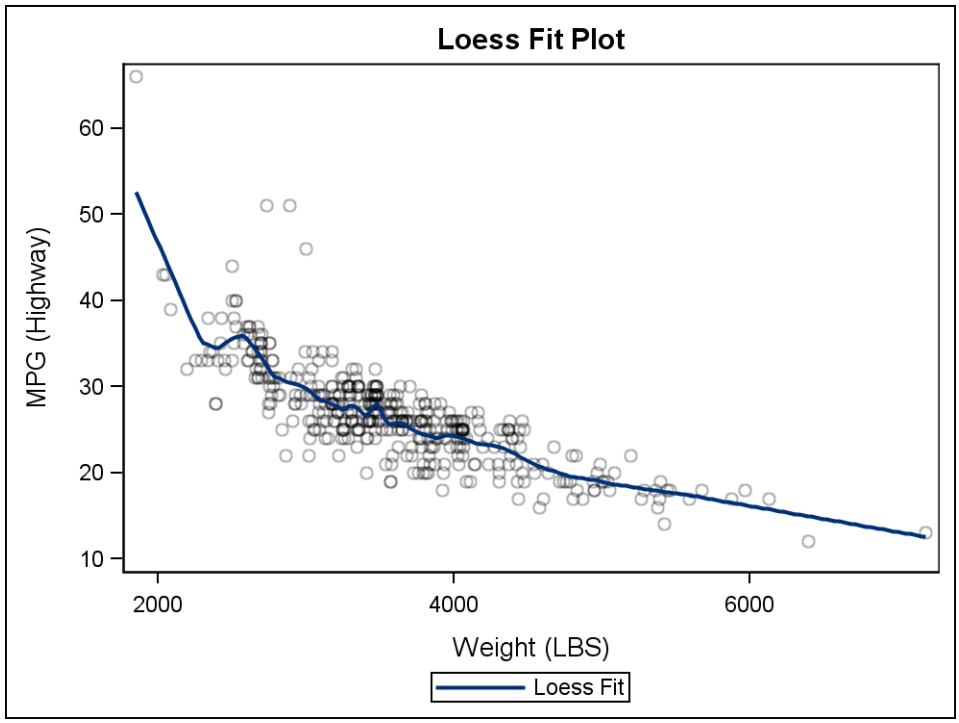
Creates a fitted loess curve computed from input data.

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 376:



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 SAS/STAT user's guide.

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 385 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.
INCLUDEMISSINGGROUP	Specifies whether missing values of the group variable are included in the plot.
INDEX	Specifies indices for mapping line attributes (color and pattern) to one of the GraphData1 - GranphDataN style elements.
LEGENDLABEL	Specifies a label for 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.

Statement Option	Description
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 loess curve.

Default: no curve label is displayed

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 849 for the syntax on using a *style-element* and "Text Options" on page 852 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 146.

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 ignored if CURVELABELLOCATION=INSIDE is specified. The START and END settings are ignored 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 146.

DATATRANSPARENCY=*number*
specifies the degree of the transparency of the loess curve.

Default: 0

Range: 0 (opaque) to 1 (entirely transparent)

GROUP=*column* | *discrete-attr-var* | *expression*
creates a distinct set of curves from just the observations that correspond to each unique group value of the specified column.

discrete-attr-var
specifies a discrete attribute variable that is defined in a DISCRETEATTRVAR statement.

Restriction: A discrete attribute variable specification must be a direct reference to the attribute variable. It cannot be set by a dynamic variable.

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.

Interaction: The INCLUDEMISSINGGROUP option controls whether missing group values are considered a distinct group value.

Tip: 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.

INCLUDEMISSINGGROUP=*boolean*
specifies whether missing values of the group variable are included in the plot.

Default: TRUE

Interaction: For this option to take effect, the GROUP= option must also be specified.

Tip: Unless a discrete attribute map is in effect or the INDEX= option is used, the attributes of the missing group value are determined by the GraphMissing style element except when the MISSING= system option is used to specify a non-default missing character or when a user-defined format is applied to the missing group value. In those cases, the attributes of the missing group value are determined by a GraphData1–GraphDataN style element.

INDEX=*numeric-column* | *expression*
specifies indices for mapping line attributes (color and pattern) to one of the GraphData1 - GranphDataN style elements.

Default no default

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.

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 144.

LEGENDLABEL= *"string"*
specifies a label for the legend item that is associated with this plot.

Default: The *string* specified on the NAME= option.

Restriction: This option applies only to an associated DISCRETELEGEND statement.

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 849 for the syntax on using a *style-element* and "Line Options" on page 850 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 562.

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 557.

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 557.

Chapter 36

MODELBAND Statement

Description and Syntax

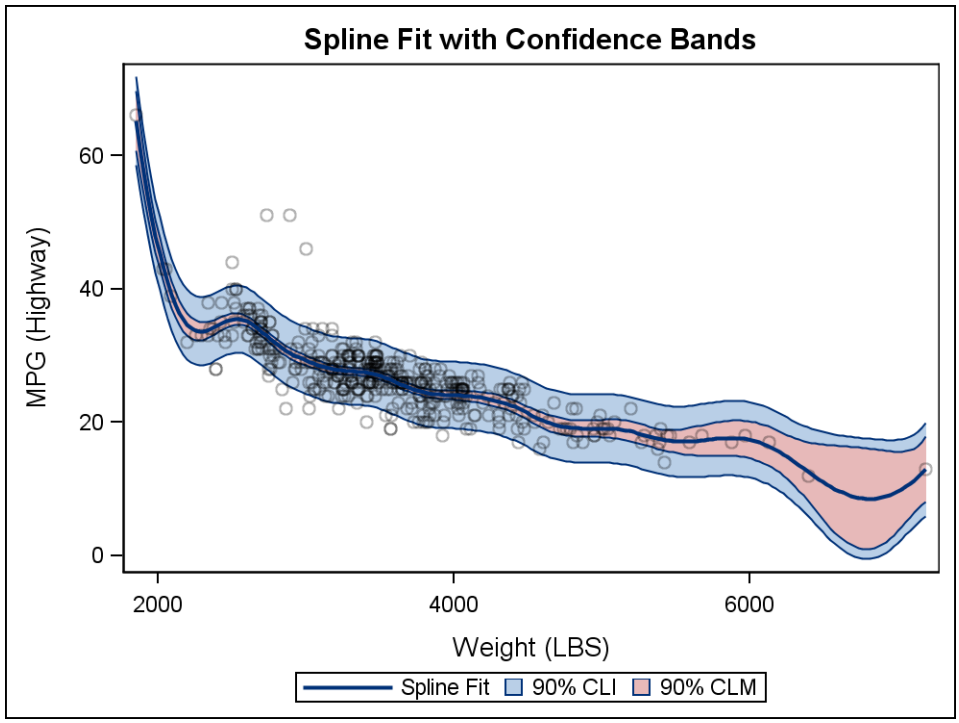
Creates a band showing confidence limits for an associated smoother plot.

MODELBAND "*confidence-name*" </*option(s)*>;

Example Program and Statement Details

Example Graph

The following graph was generated by the "Example Program" on page 386:



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 fill and the band outline.
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 filled band area.
LEGENDLABEL	Specifies a label for 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 849 for the syntax on using a *style-element* and "Text Options" on page 852 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 curve label is displayed

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 curve label is displayed

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 146.

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 ignored if CURVELABELLOCATION=INSIDE is specified. The START and END settings are ignored 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 146.

DATATRANSPARENCY=*number*
specifies the degree of the transparency of the band fill and the band outline.

Default: 0

Range: 0 (opaque) to 1 (entirely transparent)

Tip: The FILLATTRS= option can be used to set transparency for just the filled band area. You can combine this option with FILLATTRS= to set one transparency for the band outline but a different transparency for the band fill. Example:

```
datatransparency=0.2 fillattrs=(transparency=0.6)
```

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 849 for the syntax on using a *style-element* and "Fill Options" on page 850 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.

Tip: The DATATRANSPARENCY= option sets the transparency for band outline and the band fill. You can combine this option with DATATRANSPARENCY= to set one transparency for the band outline but a different transparency for the band fill. Example:

```
datatransparency=0.2 fillattrs=(transparency=0.6)
```

LEGENDLABEL= *"string"*
specifies a label for the legend item that is associated with this plot.

Default: The *string* specified on the NAME= option.

Restriction: This option applies only to an associated DISCRETELEGEND on page 681 statement.

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 849 for the syntax on using a *style-element* and "Line Options" on page 850 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 557.

The setting for this option should be the same as for the smoother statement referenced by the 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 557.

The setting for this option should be the same as for the smoother statement referenced by the confidence-name .

Chapter 37

MOSAICPLOTPARM Statement

Description and Syntax

Creates a mosaic plot from pre-summarized categorical data.

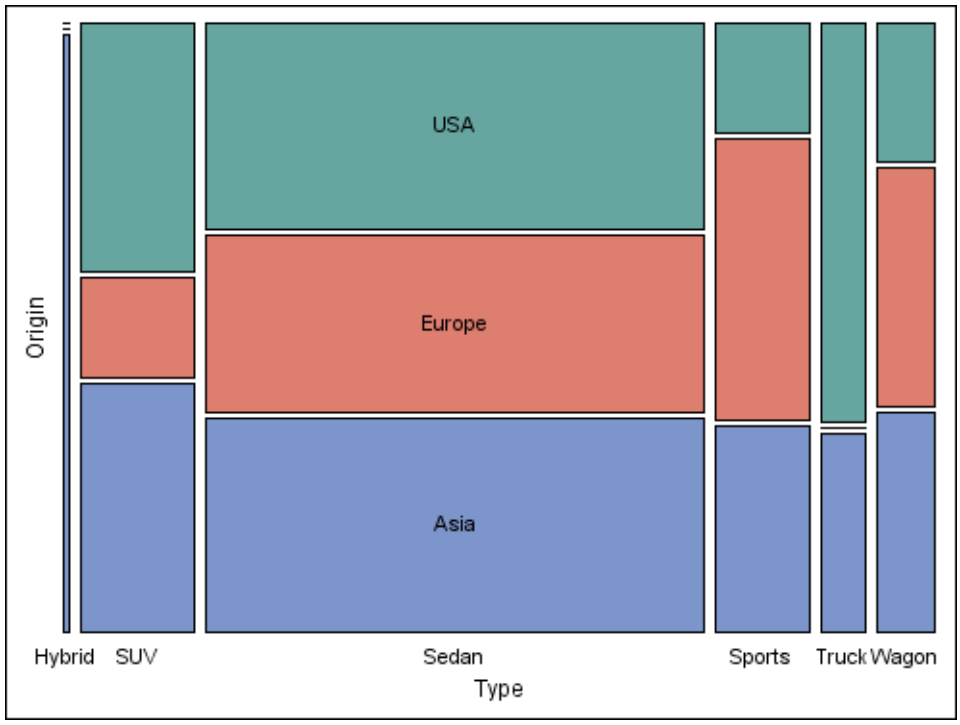
Note: This feature is for the second maintenance release of SAS 9.3 and later. See "What's New in the SAS 9.3 Graph Template Language" on page xi.

MOSAICPLOTPARM CATEGORY=(*column-list*)
COUNT=*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 394:



Example Program

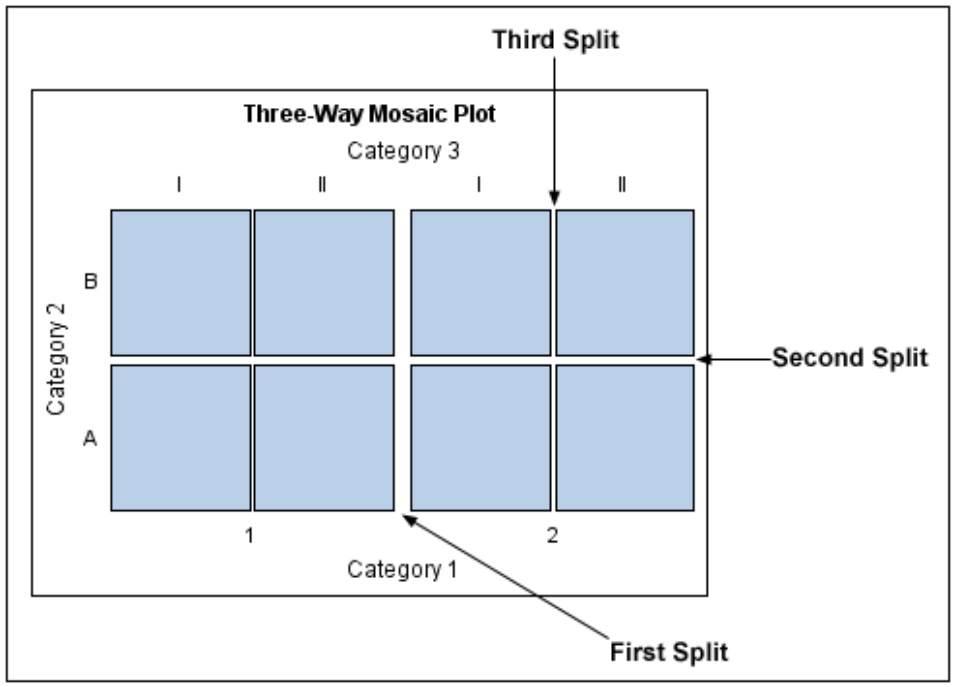
```
/* Summarize the SASHELP.CARS data for ORIGIN and TYPE */
proc summary data=sashelp.cars nway;
  class origin type;
  var mpg_highway;
  output out=mileage mean=avgMpg N=count / noinherit;
run;

/* Generate the plot */
proc template;
  define statgraph mosaicPlotParm;
    begingraph;
      layout region;
       mosaicPlotParm category=(type origin) count=count /
          colorgroup=origin;
      endlayout;
    endgraph;
  end;
run;

proc sgrender data=mileage template=mosaicPlotParm;
run;
```

Statement Summary

A mosaic plot is used to display relative frequencies for categorical variables. Each crossing of the categorical values is represented by a tile. The area of each tile is proportional to the frequency of that crossing. The graph is the result of an iterative process. The first iteration splits the plot area into tiles along the width according to the

relative frequency of the first category column values. Subsequent iterations split the tiles from the previous iteration in the direction orthogonal to the previous split using the relative frequencies of each category column's values. By default, the gap (or gutter) for each split gets progressively smaller down to a minimum gap of 3 pixels. You can use the GUTTER= option to specify a different gap for each split.

The following figure provides an example for a three-way mosaic plot, which has three categories.



Note: The plot contains equal-sized tiles for demonstration purposes.

In the example plot, the first split is along the width for CATEGORY 1. The second split is along the height for CATEGORY 2. Finally, the third split is along the width for CATEGORY 3. Notice how the gaps between the tiles get progressively smaller from the first split to the last split.

Note: The MOSAICPLOTPARM statement can be use in GRIDDED, LATTICE, and REGION layouts only.

Required Arguments

CATEGORY=(*column-list*)
specifies a list of columns of category (classification) values.

Restriction: No more than three columns can be specified.

COUNT=*non-negative-numeric-column* | *expression*
specifies the column of counts (pre-summarized) for each of the category value combinations.

Restriction: The column values cannot be negative.

Tip: You need to provide the category crossings with nonzero counts only.

Options

Statement Option	Description
COLORGROUP	Specifies the category column to use for discrete fill colors for the tiles.
COLORMODEL	Specifies the name of the color model in the style that is used by the COLORRESPONSE= option.
COLORRESPONSE	Specifies a numeric column for a continuous gradient value to tile color mapping.
DATATRANSPARENCY	Specifies the degree of the transparency of the tile fill, outlines, and (inside) values.
DISPLAY	Specifies the plot features to display.
FILLATTRS	Specifies the appearance of the tile fill.
GUTTER	Specifies the size of the gutter (gap) between the splits.
INSIDEVALUEATTRS	Specifies the text attributes of the category values when located inside a tile.
LABELATTRS	Specifies the text attributes of the category column labels.
LEGENDLABEL	Specifies a label to be used in a legend.
NAME	Assigns a name to a plot statement for reference in other template statements.
OUTLINEATTRS	Specifies the properties of tile outlines.
REVERSECOLORMODEL	Specifies whether to reverse a gradient defined by the COLORMODEL= option.
ROLENAME	Specifies user-defined roles for information contained in data columns.
SQUARED	Specifies that a square aspect ratio be used for the plot area.
TIP	Specifies the information to display when the cursor is positioned over a tile.
TIPFORMAT	Specifies display formats for information defined by roles.
TIPLABEL	Specifies display labels for information defined by roles.

Statement Option	Description
VALUEATTRS	Specifies the text attributes of the category column values when located outside a tile.
VALUELOCATION	Specifies the location of the category column values for a two-way plot.
YVALUEFITPOLICY	Specifies the fit policy for outside category values along the height of the plot.
XVALUEFITPOLICY	Specifies the fit policy for outside category values along the width of the plot.

COLORGROUP=*column* | *discrete-attr-var*
specifies the category column to use for discrete fill colors for the tiles. Each unique value of this column is mapped to the COLOR attribute of the GraphData1–GraphDataN style elements that are in effect. If a discrete attribute variable is specified, the color mapping from its associated DISCRETEATTRMAP statement is used.

discrete-attr-var
specifies a discrete attribute variable that is defined in a DISCRETEATTRVAR statement.

Restriction: A discrete attribute variable specification must be a direct reference to the attribute variable. It cannot be set by a dynamic variable.

Default: no default

Restriction: This column or the associated column in the discrete attribute variable must be one of the columns in the category column list.

Interaction: This option is ignored if the COLORRESPONSE= option is specified.

Interaction: This option overrides the FILLATTRS= option.

COLORMODEL=*style-element*
specifies a style element to be used with the COLORRESPONSE= option. The style element should contain the following style attributes:

STARTCOLOR	The color for the smallest value of the COLORRESPONSE= variable.
NEUTRALCOLOR	The color for the midpoint of the range of the COLORRESPONSE= variable.
ENDCOLOR	The color for the highest value of the COLORRESPONSE= variable.

Default: The ThreeColorRamp style element.

Interaction: The REVERSECOLORMODEL= option reverses the order of the colors that are assigned in the ramp.

Interaction: The COLORRESPONSE= option must be in effect for this option to have any effect.

COLORRESPONSE=*numeric-column* | *range-attr-var* | *expression*
specifies a numeric column that is used to map tile fill colors to a continuous gradient. The range of values of this column are linearly mapped to the color ramp

that is specified by the COLORMODEL= option. When a range attribute variable is used, the color ramp from the RANGEATTRVAR statement that is associated with the range attribute variable is used instead.

range-attr-var
specifies a range attribute variable that is defined in a RANGEATTRVAR statement.

Restriction: A reference to a range attribute variable must be a direct reference to the attribute variable. It cannot be set as a dynamic variable.

Default: no default

DATATRANSPARENCY=*number*
specifies the degree of the transparency of the tile fill, outlines, and the values that are located inside of the tiles, if displayed.

Default: 0

Range: 0 (opaque) to 1 (entirely transparent)

DISPLAY=STANDARD | ALL | (*display-options*)
specifies which graphical features to display in the plot.

STANDARD
displays tiles with fills, outlines, labels, and values.

ALL
same as STANDARD.

display-options
a list of one or more of the following options enclosed in parenthesis:

FILL	Displays filled tiles.
OUTLINE	Displays the tile outline.
LABELS	Displays the category column labels.
TICKS	Displays the category ticks.
VALUES	Displays the category values.

Default: STANDARD

Interaction: If neither FILL nor OUTLINE are present in the display-options list, then filled and outlined tiles are displayed.

Interaction: If YVALUELOCATION=INSIDE or if YVALUELOCATION=AUTO and it effectively sets it to INSIDE, then the axis ticks are not displayed even if the display of the ticks is specified for the axis. See YVALUELOCATION= on page 403.

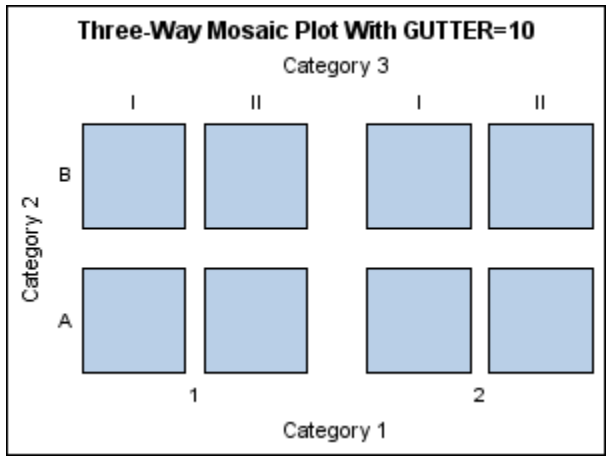
FILLATTRS=*style-element* | (*fill-options*)
specifies the appearance of the tile fill areas. See "General Syntax for Attribute Options" on page 849 for the syntax on using a *style-element* and "Fill Options" on page 850 for available *fill-options*.

Default: The GraphDataDefault style element

Note: When *style-element* is specified, only the style element's COLOR attribute is used.

GUTTER=*dimension* | (*dimension-list*)
specifies the gutter (gap) between splits as follows:

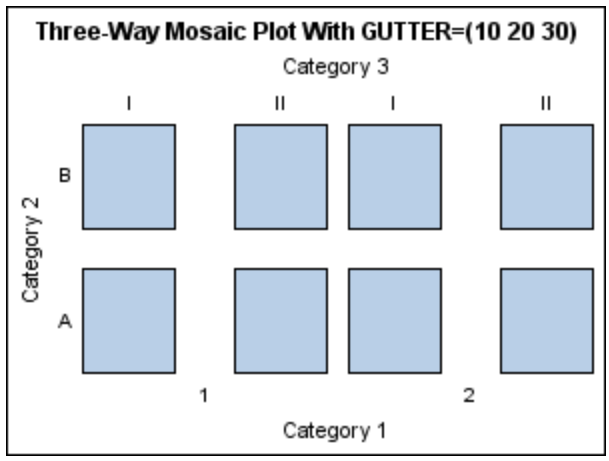
- When a single dimension is used, it applies to the gap for the last split, which has the smallest gap. From the next-to-last last split to the first split, the gutter is doubled on each split. A single dimension has the effect of setting a minimum gap for the plot. The following figure shows an example where GUTTER=10.



Note: The plot contains equal-sized tiles for demonstration purposes.

In this case, the gaps are 30 pixels for the first split, 20 pixels for the second split, and 10 pixels for the third split (minimum gap).

- When a list of dimension values is used, the values apply to each split in the order in which they are specified. The following figure shows an example where GUTTER=(10 20 30).



In this case, the gaps are 10 pixels for the first split, 20 pixels for the second split, and 30 pixels for the third split.

Default: 3px. Dimensions with no units are assumed to be in pixels.

INSIDEVALUEATTRS=*style-element* | *style-element* (*text-options*) | (*text-options*)
specifies the color and font attributes of the category values when they are located inside a tile. See "General Syntax for Attribute Options" on page 849 for the syntax on using a *style-element* and "Text Options" on page 852 for available *text-options*.

Default: The GraphValueText style element.

Interaction: If one or more text options are specified and they do not include all the font properties such as color, family, size, weight, and style, the non-specified properties are derived from the GraphValueText style element.

Note: When *style-element* is specified, only the element's COLOR, FONTFAMILY, FONTSIZE, FONTSTYLE, and FONTWEIGHT attributes are used.

Tip: You can use VALUEATTRS= option to change the text attributes for the values that are located outside the plot area.

LABELATTRS=*style-element* | *style-element* (*text-options*) | (*text-options*)
specifies the color and font attributes of the category labels. See "General Syntax for Attribute Options" on page 849 for the syntax on using a *style-element* and "Text Options" on page 852 for available *text-options*.

Default: The GraphLabelText style element.

Interaction: If one or more text options are specified and they do not include all the font properties such as color, family, size, weight, and style, the non-specified properties are derived from the GraphLabelText style element.

Note: When *style-element* is specified, only the style element's COLOR, FONTFAMILY, FONTSIZE, FONTSTYLE, and FONTWEIGHT attributes are used.

LEGENDLABEL= "*string*"
specifies a label for this plot to be used in a discrete legend.

Default: The value specified by the NAME= option or no label if the NAME= option is not set.

Interaction: If the COLORGROUP= option is in effect, this option is ignored.

NAME= "*string*"
assigns a name to a plot statement for reference in other template statements.

Restriction: The string specified by the NAME= option must be unique within the template.

Tip: The NAME= option is used mostly with the CONTINUOUSLEGEND and DISCRETELEGEND statements in order to coordinate the use of colors and line patterns between the graph and the legend.

Tip: The string is case sensitive.

OUTLINEATTRS=*style-element* | (*line-options*)
specifies the appearance of the tile outlines. See "General Syntax for Attribute Options" on page 849 for the syntax on using a *style-element* and "Line Options" on page 850 for available *line-options*.

Default: The GraphOutlines style element.

Note: When *style-element* is used, only the style element's CONTRASTCOLOR, LINESTYLE, and LINETHICKNESS are used.

REVERSECOLORMODEL=*boolean*
specifies whether to reverse the gradient (color ramp) defined by either the ODS style that is in effect or by the COLORMODEL= option.

Default: FALSE

ROLENAME= (*role-name-list*)
specifies user-defined roles for information contained in data columns. The role name list is a blank-separated list of *rolename* = *column* pairs.

Default: No user-defined roles.

Requirement: The role names that you choose must be unique and different from the predefined roles CATEGORY1 - CATEGORYn in the order in which they are

specified in the category column-list, COUNT, COLORGROUP, and COLORRESPONSE.

Interaction: This option affects the data columns that appear in tooltips. See TIP= on page 401.

The following example specifies a role for the OBS column:

```
ROLENAME=(TIP1=OBS)
```

SQUARED=*boolean*
specifies that a square aspect ratio be used for the plot area.

Default: FALSE

Restriction: This option applies to multi-way plots only.

Tip: Setting this option to TRUE makes the height of the plot the same as its width, which can make it easier to compare the proportions.

TIP=(*role-list*)
specifies the information to display when the cursor is positioned over a tile. If this option is used, it replaces all the information displayed by default. Roles for columns that do not contribute to the plot can be specified along with roles that do.

(*role-list*)
an ordered, blank-separated list of unique MOSAICPLOTPARM and user-defined roles. The MOSAICPLOTPARM roles include COUNT, COLORGROUP, and COLORRESPONSE. User-defined roles are defined with the ROLENAME= option.

The following example displays tooltips for the columns that are assigned to the roles X and Y as well as the user-defined role TIP1:

```
ROLENAME=(TIP1=OBS)
TIP=(TIP1 X Y)
```

Default: The columns assigned to the category columns and COUNT roles are automatically included in the tooltip information.

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.

(*role-format-list*)
a list of *role-name* = *format* pairs separated by blanks.

The following example specifies a format for the X and Y roles:

```
TIP=(X Y)
TIPFORMAT=(X=5.3 Y=5.3)
```

Default: The column format of the variable assigned to the role or BEST6. if no format is assigned to a numeric column.

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 ROLENAME=.

TIPLABEL= (*role-label-list*)
specifies display labels for tip columns.

(*role-label-list*)
a list of *role-name* = *"string"* pairs separated by blanks.

The following example specifies labels for the X role and the user-defined TIP1 role:

```
ROLENAME=(TIP1=PCT)
TIP=(TIP1 X Y)
TIPLABEL=(TIP1="Percent" X="Independent")
```

Default: The column label or column name of the variable assigned to the role.

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 ROLENAME=.

URL=*character-column*
specifies the URL of an HTML page to display when a tile is selected.

Default: no default

character-column
each value of the column should be a valid HTML page reference (HREF). For example: *http://www.sas.com/technologies/analytics/index.html*.

Requirement: To generate selectable tiles, you must include an ODS GRAPHICS ON statement that has the IMAGEMAP option specified, and write the graphs to the ODS HTML destination.

Tip: The URL value can be blank for some tiles, which means that no action is taken when those tiles are selected. The URL value can be the same for different tiles, which means that the same action is taken when those tiles are selected.

VALUEATTRS=*style-element* | *style-element* (*text-options*) | (*text-options*)
specifies the color and font attributes of the category values that are located outside of the tiles. See "General Syntax for Attribute Options" on page 849 for the syntax on using a *style-element* and "Text Options" on page 852 for available *text-options*.

Default: The GraphValueText style element.

Interaction: If one or more text options are specified and they do not include all the font properties such as color, family, size, weight, and style, the non-specified properties are derived from the GraphValueText style element.

Note: When *style-element* is used, only the style element's COLOR, FONTFAMILY, FONTSIZE, FONTSTYLE, and FONTWEIGHT attributes are used.

Tip: You can use INSIDEVALUEATTRS= to change text attributes for values that are located inside of tiles.

VALUELOCATION=AUTO | INSIDE | OUTSIDE
specifies the location of the category column values in a two-way plot.

AUTO
the values for the second category in the CATEGORY= list are located outside of the tiles. If the first category in the CATEGORY= list has any missing crossings with the second category, or if any of the second category values collide, the values are located inside the tiles per the default fit policy. See CATEGORY=.

INSIDE
the values are located inside of the largest tile for each category value.

OUTSIDE
the values are located outside the tiles in the plot area.

Default: AUTO

Restriction: This option applies to two-way plots only.

XVALUEFITPOLICY=ROTATE | NONE
specifies a policy for avoiding collisions among outside category values along the width of the plot.

ROTATE
rotates the values if any of the values collide.

NONE
does not attempt to fit values that collide.

Default: ROTATE

YVALUEFITPOLICY=NONE | ROTATEALWAYS
specifies a policy for avoiding collisions among outside category values along the height of the plot.

NONE
does not attempt to fit values that collide.

ROTATEALWAYS
rotate the values regardless of whether any of the values collide.

Default: NONE

Interaction: This option is effective only when VALUELOCATION=OUTSIDE.

Chapter 38

NEEDLEPLOT Statement

Description and Syntax

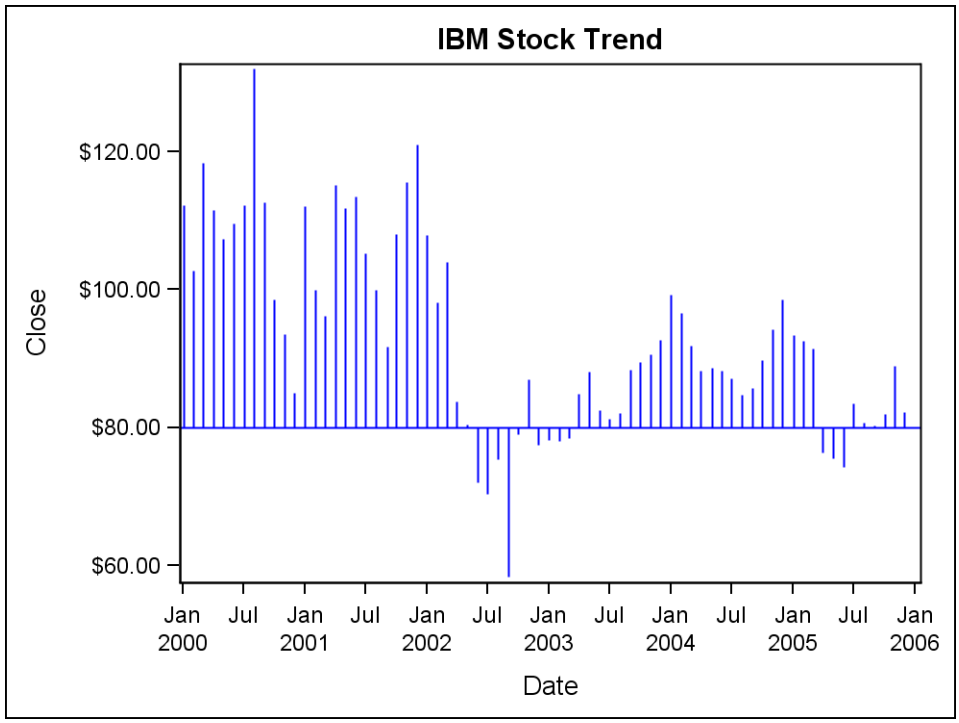
Creates a plot of observations as points connected to a baseline by vertical line segments.

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 406:



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.
CLUSTERWIDTH	Specifies the width of the group clusters as a fraction of the midpoint spacing.
DATALABEL	Specifies labels at the data points.
DATALABELATTRS	Specifies the color and font attributes of the data labels.
DATALABELPOSITION	Specifies the location of the data labels relative to the end of the needle lines and markers, if displayed.
DATATRANSPARENCY	Specifies the degree of the transparency of the needle lines and the markers, if displayed.
DISCRETEOFFSET	Specifies an amount to offset all needle lines and markers from discrete X values when graphing multiple response variables side by side on a common axis.
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.
GROUPDISPLAY	Specifies whether grouped needle lines are overlaid or clustered around the category midpoints.
GROUPORDER	Specifies the relative position of the needle lines for clustered group display.
INCLUDEMISSINGGROUP	Specifies whether missing values of the group variable are included in the plot.
INDEX	Specifies indices for mapping needle attributes (color, marker symbol, and line pattern) to one of the GraphData1–GraphDataN style elements.
LEGENDLABEL	Specifies a label for a legend.
LINEATTRS	Specifies the properties of the needle lines.

Statement Option	Description
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 user-defined roles that can be used to display information in the tooltips.
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.
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.

CLUSTERWIDTH= *number*
specifies the width of the group clusters as a fraction of the midpoint spacing on a discrete axis or a fraction of the minimum interval between adjacent data values on an interval axis.

Default: 0.85

Range: 0.1 (narrowest) to 1 (widest)

Interaction: For this option to take effect, the GROUP= option must also be specified.

DATALABEL=*column*
specifies labels at the data points.

Default: no data labels are displayed

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 849 for the syntax on using a *style-element* and "Text Options" on page 852 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.

DATALABELPOSITION = AUTO | TOPRIGHT | TOP | TOPLEFT | LEFT | CENTER | RIGHT | BOTTOMLEFT | BOTTOM | BOTTOMRIGHT
specifies the location of the data labels relative to the end of the needle lines and markers, if displayed.

Default: AUTO

DATATRANSPARENCY=*number*
specifies the degree of the transparency of the needle lines and markers, if displayed.

Default: 0

Range: 0 (opaque) to 1 (entirely transparent)

DISCRETEOFFSET=*number*
specifies an amount to offset all needle lines and markers from discrete X values when graphing multiple response variables side by side on a common axis.

Default: 0 (no offset, all needle 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 parent layout's axis options set REVERSE=TRUE, then the offset direction is also reversed.

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* | *discrete-attr-var* | *expression*
creates a distinct set of needle lines, markers, and data labels for each unique group value of the specified column.

discrete-attr-var
specifies a discrete attribute variable that is defined in a DISCRETEATTRVAR statement.

Restriction: A discrete attribute variable specification must be a direct reference to the attribute variable. It cannot be set by a dynamic variable.

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.

Interaction: The INCLUDEMISSINGGROUP= option controls whether missing group values are considered a distinct group value.

Tip: 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.

GROUPDISPLAY= OVERLAY | CLUSTER
specifies whether grouped needle lines are overlaid or clustered around the category midpoints on a discrete axis or around the intervals on an interval axis.

Default: OVERLAY

OVERLAY
Needle lines for matching category values are centered on the midpoints and superimposed on each other.

CLUSTER
Needle lines for matching category values are clustered around the midpoints. Each set of group values is centered at the midpoint for the category.

Interaction: For this option to take effect, the GROUP= option must also be specified.

GROUPORDER=DATA | ASCENDING | DESCENDING
specifies the relative position of the needle lines for clustered group display. It also sets the default order of the groups in the legend.

Default: DATA

DATA
shows each group within a category in data order of the group column.

ASCENDING
shows each group within a category in ascending order of the group column. See the Details for more information.

DESCENDING
shows each group within a category in descending order of the group column. See the Details for more information.

Interaction: This option is ignored unless GROUP= is specified.

Interaction: The SORTORDER= setting in a DISCRETELEGEND statement overrides the legend item order.

Note: Attributes such as color, symbol, and pattern are assigned to each group in the DATA order by default regardless of the GROUPORDER= option setting.

Note: The ASCENDING and DESCENDING settings linguistically sort the group values within each category (or X value) for display position purposes only. The data order of the observations and the visual attributes that are assigned to the group values remain unchanged.

Tip: The CLUSTERWIDTH= option can be used to control the distance between the group markers in a cluster

Tip: The INDEX= option can be used to alter the default sequence of visual attributes that is assigned to the needle lines.

Details: By default, the value that is set for this option determines the order in which attributes are assigned to each group. It also determines the order in which the groups are displayed in the legend.

For ASCENDING and DESCENDING sort orders, the order is based on the data type:

- For Numeric data, the order is base on unformatted values.
- For Character data the order is based on formatted values.

The ASCENDING and DESCENDING settings perform a linguistic sort on the group items and have the same effect as sorting the input data. However, the data is not changed.

INCLUDEMISSINGGROUP=*boolean*
specifies whether missing values of the group variable are included in the plot.

Default: TRUE

Interaction: For this option to take effect, the GROUP= option must also be specified.

Tip: Unless a discrete attribute map is in effect or the INDEX= option is used, the attributes of the missing group value are determined by the GraphMissing style element except when the MISSING= system option is used to specify a non-default missing character or when a user-defined format is applied to the missing group value. In those cases, the attributes of the missing group value are determined by a GraphData1–GraphDataN style element.

INDEX=*numeric-column*
specifies indices for mapping needle attributes (color, marker symbol, and line pattern) to one of the GraphData1–GraphDataN style elements.

Default: no default

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.

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 144.

LEGENDLABEL= *"string"*
specifies a label for the legend item that is associated with this plot.

Default: The Y-variable label. If a label is not defined, the Y-variable name.

Restriction: This option applies only to an associated DISCRETELEGEND statement.

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 needle lines for the data points. See "General Syntax for Attribute Options" on page 849 for the syntax on using a *style-element* and "Line Options" on page 850 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 849 for the syntax on using a *style-element* and "Marker Options" on page 851 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 562.

ROLENAME=(*role-name-list*)
specifies user-defined roles that can be used to display information in the tooltips.

Default: no user-defined roles

(*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* is a valid reference.

Requirement: To generate selectable needle lines, 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 557.

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 557.

Chapter 39

PBSPLINEPLOT Statement

Description and Syntax

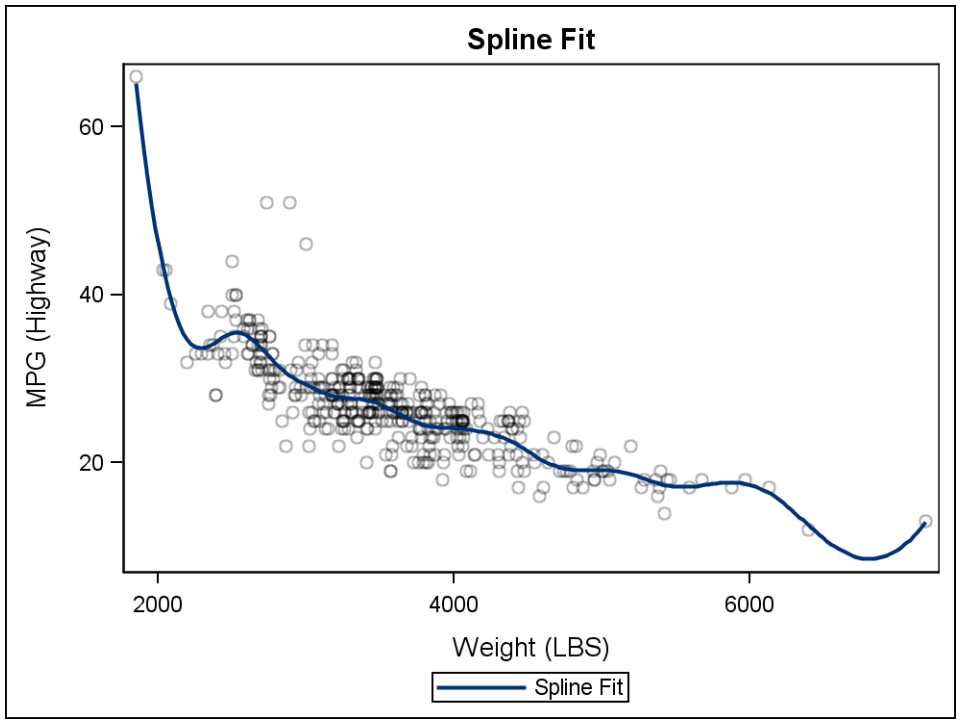
Creates a fitted penalized B-spline curve computed from input data.

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 418:



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 SAS/STAT user's guide.

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 385 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 417.

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 417.

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 in the SAS/STAT user's guide. 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
INCLUDEMISSINGGROUP	Specifies whether missing values of the group variable are included in the plot.
INDEX	Specifies indices for mapping line attributes (color and pattern) to one of the GraphData1 - GranphDataN style elements.
LEGENDLABEL	Specifies a label for 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 curve label is displayed

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 849 for the syntax on using a *style-element* and "Text Options" on page 852 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 146.

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 ignored if CURVELABELLOCATION=INSIDE is specified. The START and END settings are ignored 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 146.

DATATRANSPARENCY=*number*
specifies the degree of the transparency of the curve.

Default: 0

Range: 0 (opaque) to 1 (entirely transparent)

GROUP=*column* | *discrete-attr-var* | *expression*
creates a distinct set of curves from just the observations that correspond to each unique group value of the specified column.

discrete-attr-var
specifies a discrete attribute variable that is defined in a DISCRETEATTRVAR statement.

Restriction: A discrete attribute variable specification must be a direct reference to the attribute variable. It cannot be set by a dynamic variable.

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.

Interaction: The INCLUDEMISSINGGROUP option controls whether missing group values are considered a distinct group value.

Tip: 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.

INCLUDEMISSINGGROUP=*boolean*
specifies whether missing values of the group variable are included in the plot.

Default: TRUE

Interaction: For this option to take effect, the GROUP= option must also be specified.

Tip: Unless a discrete attribute map is in effect or the INDEX= option is used, the attributes of the missing group value are determined by the GraphMissing style element except when the MISSING= system option is used to specify a non-default missing character or when a user-defined format is applied to the missing group value. In those cases, the attributes of the missing group value are determined by a GraphData1–GraphDataN style element.

INDEX=*numeric-column* | *expression*
specifies indices for mapping line attributes (color and pattern) to one of the GraphData1 - GranphDataN style elements.

Default: no default

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.

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 144.

LEGENDLABEL= *"string"*
specifies a label for the legend item that is associated with this plot.

Default: The *string* specified on the NAME= option.

Restriction: This option applies only to an associated DISCRETELEGEND statement.

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 849 for the syntax on using a *style-element* and "Line Options" on page 850 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 562.

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 557.

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 557.

Chapter 40

PIECHART Statement

Description and Syntax

Creates a pie chart that is computed from input data.

The following syntax creates a pie chart with slices representing counts or percents of unique CATEGORY values.

PIECHART CATEGORY= *column* | *discrete-attr-var* | *expression* </ *option(s)*>;

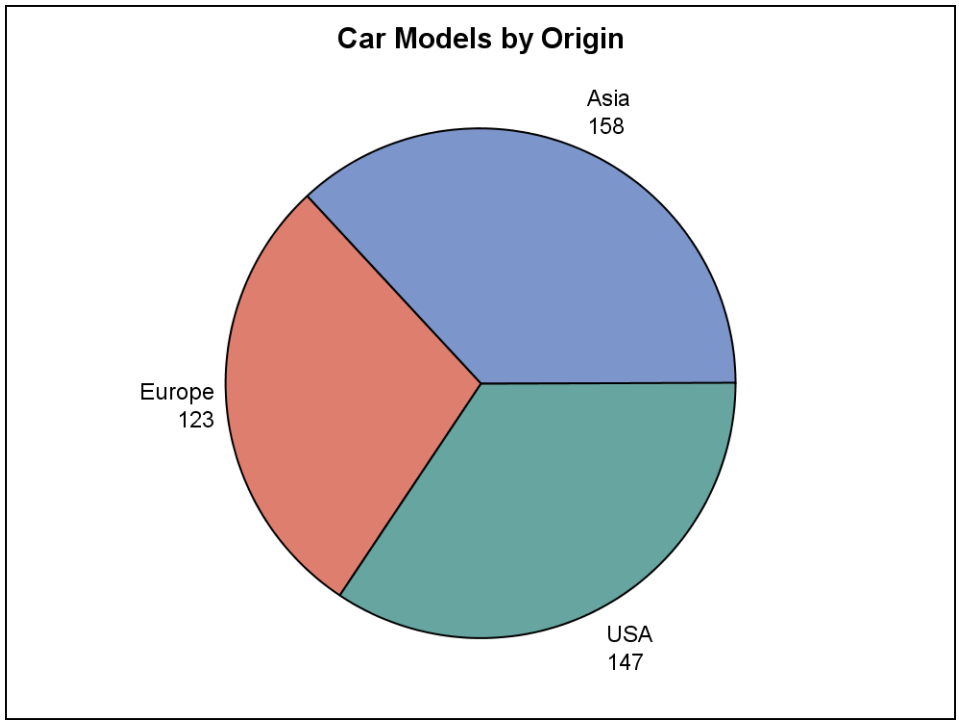
The following syntax creates a pie chart with slices representing unique CATEGORY values summarized over corresponding values of the RESPONSE column.

PIECHART CATEGORY= *column* | *discrete-attr-var* | *expression*
RESPONSE= *numeric-column* | *expression* </ *option(s)*>;

Example Program and Statement Details

Example Graph

The following graph was generated by the "Example Program" on page 428 :



Example Program

```
proc template;
define statgraph simplepie;
  begingraph;
    entrytitle "Car Models by Origin";
    layout region;
      piechart category=origin / datalabellocation=outside;
    endlayout;
  endgraph;
end;
run;
proc sgrender data=sashelp.cars
              template=simplepie;
run;
```

Statement Summary

The PIECHART statement is used in a LAYOUT REGION, LAYOUT GRIDDED, or LAYOUT LATTICE block. It cannot be used in LAYOUT OVERLAY block because a pie chart does not have axes. The input data for the PIECHART statement is raw, unsummarized input data. The PIECHART performs discrete binning for the CATEGORY column and calculates appropriate summarization statistics (sum, mean, and so on) based on the setting for the STAT= option.

By default, the fill color of each pie slice is derived from the Color attribute of the GraphData1–GraphDataN style elements. When the number of categories exceeds the number of GraphDataN elements, the GraphData1–GraphDataN colors are repeated, but the shade of each color is varied on each cycle. The shade variations alternate between one shade lighter and one shade darker on each cycle. The fill color of the “other” slice (if shown) is derived from the Color attribute of the GraphOther style element.

The default order of the pie slices depends on the data type of the CATEGORY values:

- For numeric data, the slices appear in the ascending order of the unformatted data values.
- For discrete data, the slices appear in data order.

You can use the START= and CATEGORYDIRECTION= options to control the pie slice positions and display order.

By default, the pie slices are labeled with the CATEGORY and RESPONSE values, which are displayed inside the slices. You can use the DATALABELLOCATION= and DATALABELCONTENT= options to control where the pie slices are labeled and the label content.

By default if two or more slices take up less than 4% of a pie, an “other” slice is created by consolidating those small slices. To change the default criteria, use the OTHERSLICE= and OTHERSLICEOPTS= options. The calculated “other” slice is displayed as the last slice in the pie, and as the last legend entry for the pie. If a category value is the same as the “other” slice label, two slices might be displayed with the same label ("Other" by default) and different fill attributes. In that case, both slices are represented in the pie legend.

To create a pie slice for missing CATEGORY values, specify INCLUDEMISSINGDISCRETE=TRUE in the BEGINGRAPH statement. The fill color of the missing category slice is assigned the fill color from the GraphMissing style element except when a user-defined format is applied to the category value. In that case, the missing category slice is assigned the fill color from a GraphData1–GraphDataN style element in data order instead.

Note: The PIECHART statement does not honor the MISSING= system option. Regardless of the MISSING= system option value, unless a user-defined format is applied to the value, the default missing-numeric-value character (.) is used to depict missing numeric values.

Arguments

CATEGORY=*column* | *discrete-attr-var* | *expression*
(required argument) specifies the column for the category values. Duplicated values of CATEGORY are summarized into a unique value. All values are treated as discrete.

discrete-attr-var
specifies a discrete attribute variable that is defined in a DISCRETEATTRVAR statement.

Restriction: A discrete attribute variable specification must be a direct reference to the attribute variable. It cannot be set by a dynamic variable.

RESPONSE=*numeric-column* | *expression*
(optional argument) specifies response values that are read from a numeric column or an expression.

Options

Statement Option	Description
CATEGORYDIRECTION	Specifies whether to display the pie slices in counterclockwise or clockwise sequence.
DATALABELATTRS	Specifies the color and font attributes of the data labels.
DATALABELCONTENT	Specifies the information to display in the slice labels.
DATALABELLOCATION	Specifies whether to display the slice labels within the pie slices or outside of the pie circumference.
DATASKIN	Enhances the visual appearance of the filled pie slices.
DATATRANSPARENCY	Specifies the degree of the transparency of all pie slices, outlines, and text.
DISPLAY	Specifies whether to display outlined pie slices, filled pie slices, or outlined and filled pie slices.
FILLATTRS	Specifies the appearance of the filled pie slices.
GROUP	Creates a separate concentric annulus (or stacked cylinders) for each unique group value of the specified column.
GROUPGAP	Specifies a dimension for the optional gap that can be displayed between each annulus of a grouped pie.
GROUPLABELOPTS	Specifies text attributes, location, and other options for displaying group labels.
INCLUDEMISSINGGROUP	Specifies whether missing values of the GROUP column are included in the pie.
LABELFITPOLICY	Specifies the label fitting policy to be used if a particular label does not fit within the pie slice.
NAME	Assigns a name to a plot statement for reference in other template statements.

Statement Option	Description
OTHERSLICE	Specifies whether to consolidate smaller pie slices into a single slice that represents “other” values that are in the data, or whether to display those smaller slices as separate pie slices.
OTHERSLICEOPTS	Specifies the properties of the “other” slice.
OUTLINEATTRS	Specifies the properties of the pie and slice outlines.
START	Specifies which degree between 0 and 360 serves as the starting position for the first pie slice.
STAT	Specifies the statistic to be computed.
TIP	Specifies the information to display when the cursor is positioned over a pie slice.
TIPFORMAT	Specifies display formats for information defined by the tooltip roles.
TIPLABEL	Specifies display labels for information defined by the tooltip roles.
URL	Specifies an HTML page to display when a pie slice is selected.

CATEGORYDIRECTION= COUNTERCLOCKWISE | CLOCKWISE
specifies whether to display the pie slices in counterclockwise or clockwise sequence.

Default: COUNTERCLOCKWISE

See Also: The START= option for controlling the starting angle for the first pie slice.

DATALABELATTRS=*style-element* | *style-element* (*text-options*) | (*text-options*)
specifies the color and font attributes of the slice labels. See “General Syntax for Attribute Options” on page 849 for the syntax on using a *style-element* and “Text Options” on page 852 for available *text-options*.

Default: The GraphValueText style element.

DATALABELCONTENT= ALL | STANDARD | NONE | (*content-options*)
specifies the information to display in the slice labels.

Default:

- When STAT=PCT, the default is (CATEGORY PERCENT).
- For all other values of the STAT= option, the default is STANDARD.

ALL
displays all available information

STANDARD
equivalent to specifying the two *content-options* CATEGORY and RESPONSE

NONE
does not display slice labels

(*content-options*)
a list of options, enclosed in parentheses, that must include at least one of the following:

CATEGORY—displays the CATEGORY value

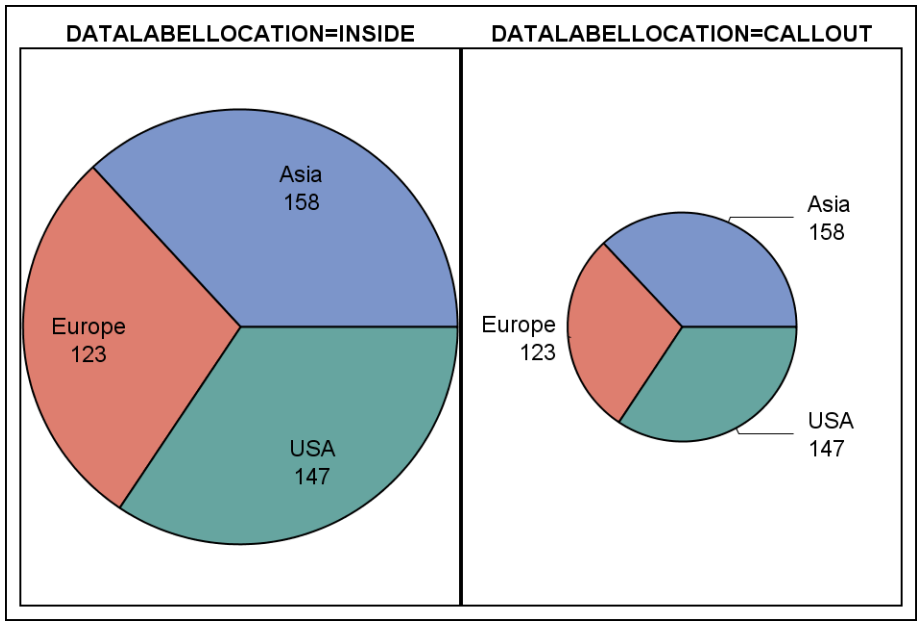
PERCENT—the display depends on the setting for the STAT= option:

- if STAT=FREQ or STAT=PCT, displays the PERCENT value
- if STAT=MEAN or STAT=SUM, displays nothing

RESPONSE—displays the statistic that is requested in the STAT= option.

Note: The position of the labels is adjusted to prevent the labels from overlapping.

DATALABELLOCATION = AUTO | INSIDE | OUTSIDE | CALLOUT
specifies whether to display the slice labels within the pie slices or outside of the pie circumference.



Default: AUTO

AUTO
automatically selects either INSIDE, OUTSIDE, or CALLOUT to optimize the label position

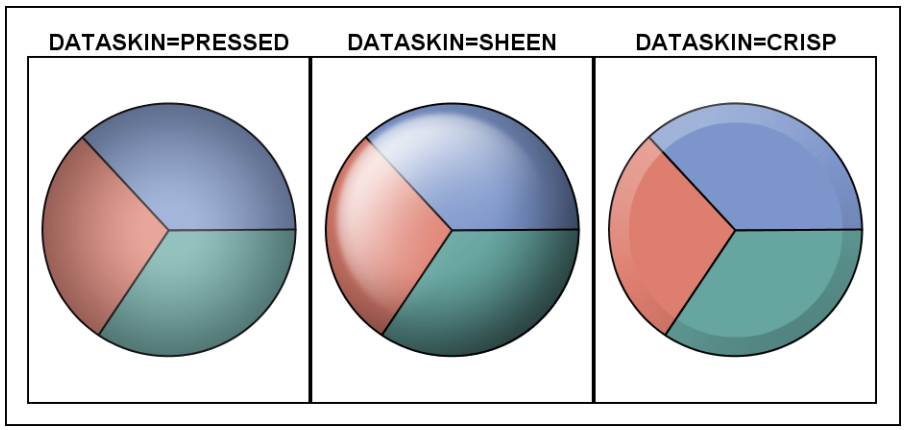
INSIDE
locates the slice labels inside the pie slices.

Note: If a particular label does not fit within the pie slice, the fit policy takes effect (set by the LABELFITPOLICY= option).

OUTSIDE
locates the slice labels outside of the pie circumference.

CALLOUT
locates the slice labels outside of the pie circumference and draws a callout from the label to its slice.

DATASKIN= NONE | PRESSED | SHEEN | CRISP | GLOSS | MATTE
enhances the visual appearance of the filled pie slices.



Default: NONE

Requirement: For this option to have any effect, DISPLAY= FILL must be in effect. Otherwise, this option is ignored.

Interaction: The appearance of the data skin is based on the FILLATTRS= color.

Interaction: When a data skin is applied, all slice outlines are set by the skin, and the OUTLINEATTRS= option is ignored.

DATATRANSPARENCY=*number*
specifies the degree of the transparency of all pie slices, outlines, and text.

Default: 0

Range: 0 (opaque) to 1 (entirely transparent)

Tip: The FILLATTRS= option can be used to set transparency for just the pie slices. The OTHERSLICEOPTS= option can be used to specify transparency for the "other" slice. You can combine this option with FILLATTRS= and with OTHERSLICEOPTS= to set one transparency for the outlines and text but a different transparency for the pie slices. Example:

```
datatransparency=0.2 fillattrs=(transparency=0.6)
```

DISPLAY=STANDARD | (*display-options*)
specifies whether to display outlined pie slices, filled pie slices, or outlined and filled pie slices.

Default: STANDARD

STANDARD
displays outlined, filled pie slices

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

OUTLINE	displays outlined pie slices. The default outline properties are set by the GraphOutline style element.
FILL	displays filled pie slices with each slice a different color. The default colors are set by the Color attribute of the GraphData1-

GraphDataN style elements. The fill color of the "other" slice (if shown) is from the color attribute of the GraphOther style element. If FILL is not specified, an opaque pie is drawn using the background color of the containing layout.

Tip: Use the OUTLINEATTRS= and FILLATTRS= options to control the appearance of the pie slices.

FILLATTRS=*style-element* | *style-element* (*fill-options*) | (*fill-options*)
specifies the appearance of the filled pie slices, excluding the "other" slice. See "General Syntax for Attribute Options" on page 849 for the syntax on using a *style-element* and "Fill Options" on page 850 for available *fill-options*.

Default: The GraphDataDefault:Color style reference.

Interaction: For this option to have any effect, the fill must be enabled by the ODS style or the DISPLAY= option.

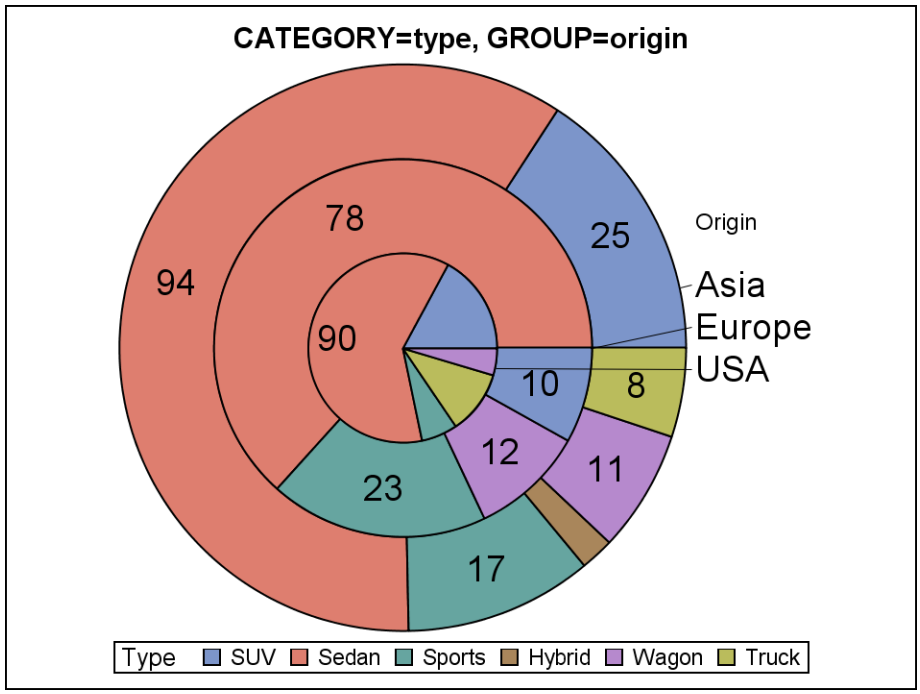
Tip: The FILLATTRS= suboption of the OTHERSLICEOPTS= option controls the visual properties of the "other" slice.

Tip: The DATATRANSPARENCY= option sets the transparency for all pie slices, outlines, and text. You can combine this option with DATATRANSPARENCY= to set one transparency for the outlines and text but a different transparency for the pie slices. Example:

```
datatransparency=0.2 fillattrs=(transparency=0.6)
```

GROUP=*column* | *expression*
creates a separate concentric annulus (or stacked cylinders) for each unique group value of the specified column. The grouped rings are displayed in data order.

Default: no default.



Interaction: When this option is used, the unique column values are found and then the slice colors are taken from the GraphData1 - GraphDataN style elements.

Interaction: Missing values in the data can affect the group order. You can use the INCLUDEMISSINGGROUP= option to manage missing group values. In addition,

you can use INCLUDEMISSINGDISCRETE=TRUE in the BEGINGRAPH statement to create pie slices for missing CATEGORY values.

Tip: This option creates only rings of pies. To create a grid of pies, specify the PIECHART statement within a LAYOUT LATTICE, LAYOUT DATALATTICE, or LAYOUT DATAPANEL statements.

See Also: The FILLATTRS= , GROUPGAP= , and GROUPLABELOPTS= options.

GROUPGAP= *dimension*
specifies a dimension for the optional gap that can be displayed between each annulus of a grouped pie.

Default: 0

Restriction: For this option to take effect, the GROUP= option must also be specified.

Interaction: If the specified *dimension* is too large for the area that is available to the pie chart, unexpected results might occur.

Note: The size of the inner pie remains the same regardless of the GROUPGAP= value.

GROUPLABELOPTS=(*grouplabel-options*)
specifies text attributes, location, and other options for displaying group labels.

Default: The group label and values are shown as a block of text to the right or left of the pie. Slice labels are moved to the inside of the pie slices. A callout is drawn from each group value to its annulus (or cylinder).

Restriction: For this option to take effect, the GROUP= option must also be specified.

The following *grouplabel-options* are available. One or more options can be specified as *name* = *value* pairs, separated by blanks.

LABEL= AUTO | NONE | "*string*"
specifies a descriptive label for the GROUP column

Default: AUTO

AUTO	The column label of the GROUP= column or the column name of the GROUP= column, if no column label exists.
NONE	No label is displayed
"*string*"	The specified string is used

LABELATTRS= *style-element* | *style-element* (*text-options*) | (*text-options*)
specifies the text properties of the group label. See "General Syntax for Attribute Options" on page 849 for the syntax on using a *style-element* and "Text Options" on page 852 for available *text-options*.

Default: The GraphLabelText style element.

LOCATION= RIGHT | LEFT
specifies whether the block of text for group labeling appears to the right or left of the pie.

Default: RIGHT

VALUEATTRS=*style-element* | *style-element* (*text-options*) | (*text-options*)
specifies the text properties of the group values. See "General Syntax for Attribute Options" on page 849 for the syntax on using a *style-element* and "Text Options" on page 852 for available *text-options*.

Default: The GraphValueText style element.

INCLUDEMISSINGGROUP=*boolean*
specifies whether missing values of the GROUP column are included in the pie. When INCLUDEMISSINGGROUP=TRUE and missing group values exist in the data, a solid ring or inner pie is added to the pie chart to represent the missing group. The missing group ring or inner pie is assigned a fill color from the GraphData1-GraphDataN style elements in data order.

Default: TRUE

Restriction: For this option to take effect, the GROUP= option must also be specified.

LABELFITPOLICY = NONE | DROP
specifies the label fitting policy to be used if a particular label does not fit within the pie slice.

Default: NONE

NONE
Draw each label regardless of whether it fits within the slice region.

DROP
Drop labels that do not fit within the slice region, but draw labels that do fit.

Tip: This option determines how labels are managed when DATALABELLOCATION= INSIDE and a particular label does not fit within the pie slice.

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.

OTHERSLICE = *boolean*
specifies whether to consolidate smaller pie slices into a single slice that represents "other" values that are in the data, or whether to display those smaller slices as separate pie slices.

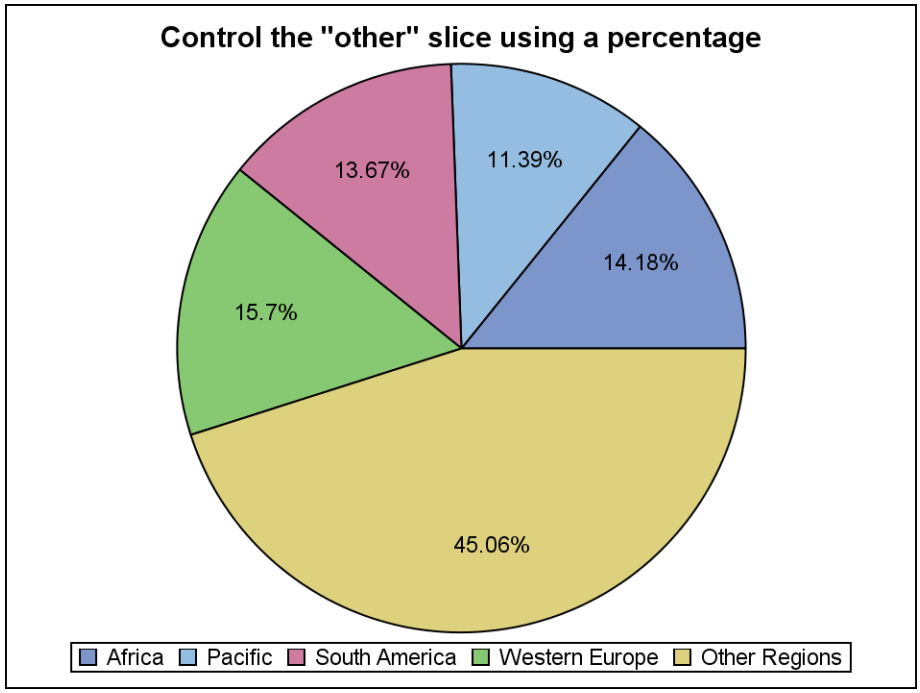
Default: TRUE

Tip: To set the properties of the "other" slice, use the OTHERSLICEOPTS= option.

Details: If this option is set to FALSE, then all unique category values appear as slices. If this option is set to TRUE, then some of the smaller slices might be combined into a single slice, referred to as the "other" slice.

OTHERSLICEOPTS=(*other-slice-options*)
specifies the properties of the "other" slice. Example:

```
piechart category=region / name="p"
  datalabelcontent=(percent) datalabellocation=inside
  otherslice=true
  othersliceopts=(type=percent percent=11 label="Other Regions") ;
```



Interaction: This option is ignored if OTHERSLICE=FALSE.

The following *other-slice-options* are available. One or more options can be specified as *name* = *value* pairs, separated by blanks.

TYPE = PERCENT | MAXSLICES
specifies which method to use to determine the size of the "other" slice.

Default: PERCENT

PERCENT	Use the percentage that is set by the PERCENT= suboption.
MAXSLICES	Use the count that is set by the MAXSLICES= suboption.

MAXSLICES= *positive-integer*
specifies the maximum number of category values to represent with pie slices. Any remaining values are consolidated into the "other" slice.

Default: 10

Interaction: For this option to have any effect, TYPE=MAXSLICES must also be specified among the suboptions for OTHERSLICEOPTS=.

Interaction: The slices are counted in the order in which they are displayed. This order is affected by the CATEGORYDIRECTION= option.

PERCENT= *percent-of-total*
collects all category values with response values less than or equal to the specified *percent-of-total* into the "other" slice.

Default: 4. Any original slice that represents 4% or less of the total is put in the "other" category.

Range: 0 to 100

Interaction: For this option to have any effect, TYPE=PERCENT must also be specified among the suboptions for OTHERSLICEOPTS=..

LABEL= "*string*"
specifies a label for the “other” slice.

Default: "OTHER"

FILLATTRS=*style-element* | *style-element* (*fill-options*) | (*fill-options*)
specifies the appearance of the “other” slice's area fill. See “General Syntax for Attribute Options” on page 849 for the syntax on using a *style-element* and “Fill Options” on page 850 for available *fill-options*.

OUTLINEATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)
specifies the properties of the pie and slice outlines. See “General Syntax for Attribute Options” on page 849 for the syntax on using a *style-element* and “Line Options” on page 850 for available *line-options*.

Default: The GraphOutlines style element.

Interaction: For this option to have any effect, outlines must be enabled by the ODS style or the DISPLAY= option.

Interaction: If the DATASKIN= option applies a data skin, this option is ignored.

START=*degrees*
specifies which degree between 0 and 360 serves as the starting position for the first pie slice.

Default: 0

Range: 0 to 360

Details: A value of 0 degrees corresponds to the three o'clock position. Degrees can be either positive or negative. Positive values move the starting position counterclockwise, and negative values move the starting position clockwise. By default, successive slices are drawn counterclockwise from the starting slice. You can change the default direction with the CATEGORYDIRECTION= option.

STAT=FREQ | PCT | SUM | MEAN
Specifies the statistic to be computed.

Default:

- FREQ for pie charts that do not specify the RESPONSE= argument
- SUM for pie charts that specify the RESPONSE= argument

Available Values:

RESPONSE= Argument Is Not Specified	RESPONSE= Argument Is Specified
FREQ—Frequency count	SUM—Sum
PCT—Percent	MEAN—Mean

Interaction: When STAT=PCT, the default label content is (CATEGORY PERCENT) instead of STANDARD. See “DATALABELCONTENT= ALL | STANDARD | NONE | (content-options)” on page 431.

TIP=(*role-list*)
specifies the information to display when the cursor is positioned over a pie slice. 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: CATEGORY and RESPONSE .

(*role-list*)
an ordered, blank-separated list of unique PIECHART roles. PIECHART roles include CATEGORY, RESPONSE, and GROUP. The RESPONSE role represents the computed statistic for the CATEGORY value, based on the statistic that is set by the STAT= option.

The following example displays tooltips for the columns assigned to the roles CATEGORY and RESPONSE:

```
TIP=(CATEGORY RESPONSE)
```

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.

Tip: 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 information defined by the tooltip roles.

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.

```
TIP=(CATEGORY RESPONSE)
TIPFORMAT=(RESPONSE=5.)
```

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 information defined by the tooltip roles. Only the roles that appear in the TIP= option are used.

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.

```
TIP=(CATEGORY RESPONSE)
TIPLABEL=(RESPONSE="Frequency")
```

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 a pie slice 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 a plot with selectable pie slices, 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 pie slices, meaning that no action is taken when the corresponding slice is selected. The URL value can be the same for any

CATEGORY and RESPONSE pairs. In that case, the same action is taken when the pie slices for those pairs are selected.

Chapter 41

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)*>;

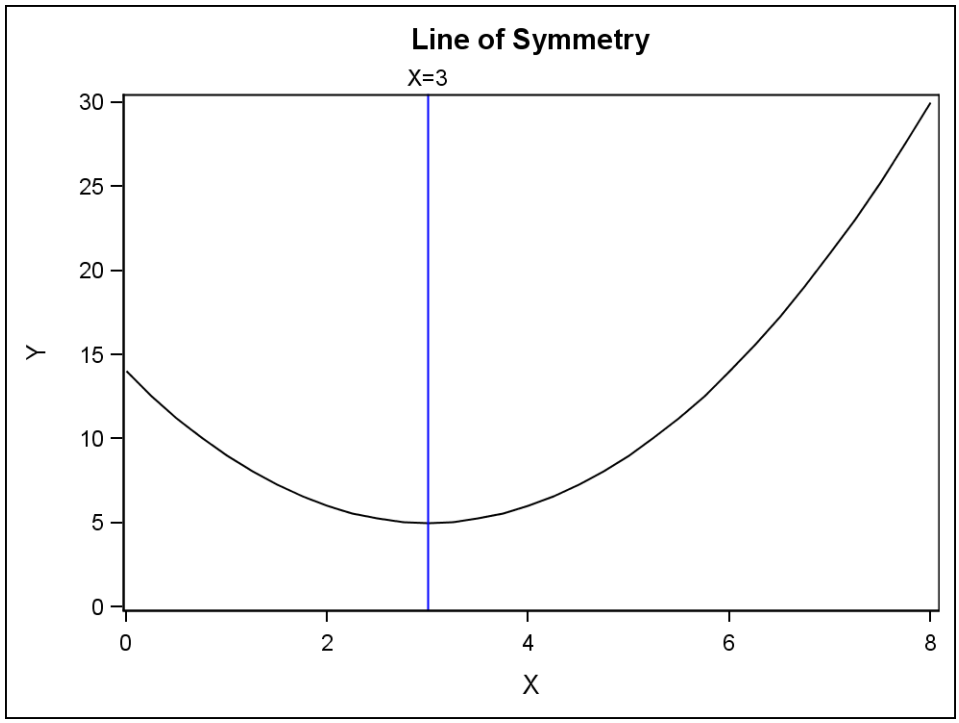
Syntax for creating a line perpendicular to the Y-axis at a Y-intercept:

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 442:



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 or lines.

Requirement: If X is 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 be the same type as 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.

Unformatted numeric values do not map to a formatted discrete axis. When the X axis is a discrete axis, the X axis value must be the formatted value that appears on the X axis. If a column is specified for the values in that case, the specified column must have the same format that is used for the X axis.

Y=*y-axis-value* | *column* | *expression*
specifies the Y intercept of the reference line or lines.

Requirement: If Y is 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 be the same type as 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.

Unformatted numeric values do not map to a formatted discrete axis. When the Y axis is a discrete axis, the Y axis value must be the formatted value that appears on the Y axis. If a column is specified for the values in that case, the specified column must have the same format that is used for the Y axis.

Options

Statement Option	Description
CLIP	Specifies whether the data for the reference line or lines are considered when determining the data ranges for the axes.
CURVELABEL	Specifies a label for the reference line or lines.
CURVELABELATTRS	Specifies the color and font attributes of the reference line label(s).

Statement Option	Description
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 or lines.
DISCRETEOFFSET	Specifies an amount to offset all reference lines from discrete X or Y values.
LEGENDLABEL	Specifies a label for a legend.
LINEATTRS	Specifies the properties of the reference line or lines.
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 or lines.

Default: no curve label is displayed

Interaction:

- If the X or Y argument specifies a value, use *"string"*.
- If the X or Y argument specifies a column, use *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 849 for the syntax on using a *style-element* and "Text Options" on page 852 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: OUTSIDE

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 146.

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 ignored 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 146.

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 discrete X or Y values.

Default: 0 (no offset, all reference lines are centered on discrete X or Y values)

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 the legend item that is associated with this plot.

Default: The *string* specified on the NAME= option.

Restriction: This option applies only to an associated DISCRETELEGEND statement.

LINEATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)
specifies the attributes of the reference line. See "General Syntax for Attribute Options" on page 849 for the syntax on using a *style-element* and "Line Options" on page 850 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 557.

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 557.

Chapter 42

REGRESSIONPLOT Statement

Description and Syntax

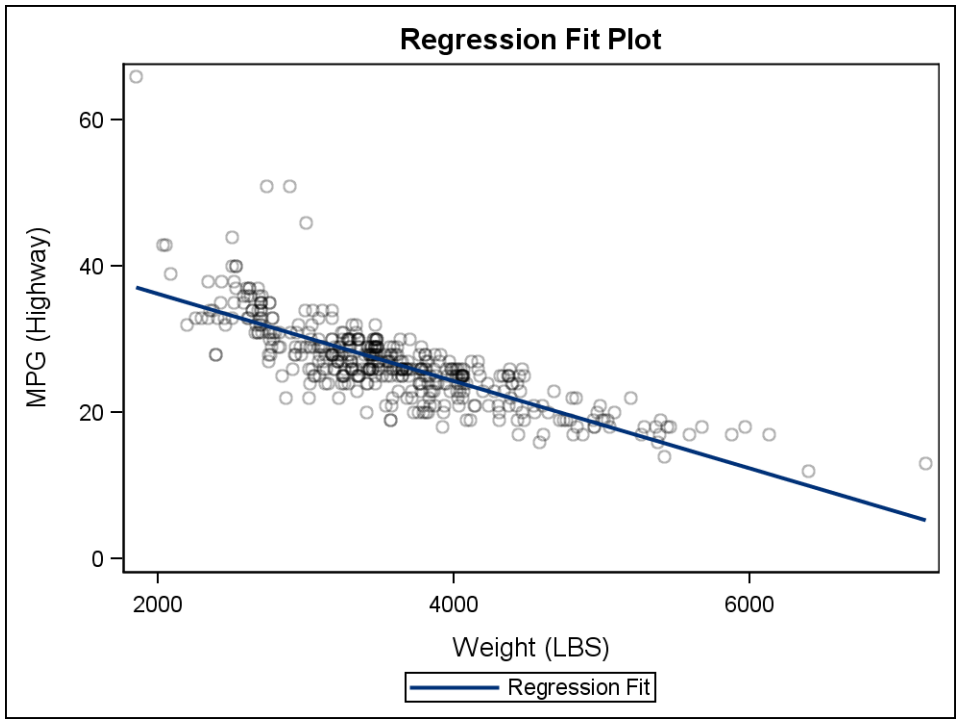
Creates a fitted regression line or curve computed from input data.

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 450:



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 SAS/STAT user's guide.

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 449.

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 449.

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.
INCLUDEMISSINGGROUP	Specifies whether missing values of the group variable are included in the plot.
INDEX	Specifies indices for mapping line attributes (color and pattern) to one of the GraphData1–GraphDataN style elements.
LEGENDLABEL	Specifies a label for a legend.
LINEATTRS	Specifies the properties of the regression line.

Statement Option	Description
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 line.

Default: no curve label is displayed

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 849 for the syntax on using a *style-element* and "Text Options" on page 852 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 146.

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 ignored if CURVELABELLOCATION=INSIDE is specified. The START and END settings are ignored 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 146.

DATATRANSPARENCY=*number*
specifies the degree of the transparency of the regression line.

Default: 0

Range: 0 (opaque) to 1 (entirely transparent)

GROUP=*column* | *discrete-attr-var* | *expression*
creates a distinct set of regression lines from just the observations that correspond to each unique group value of the specified column.

discrete-attr-var
specifies a discrete attribute variable that is defined in a DISCRETEATTRVAR statement.

Restriction: A discrete attribute variable specification must be a direct reference to the attribute variable. It cannot be set by a dynamic variable.

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.

Interaction: The INCLUDEMISSINGGROUP option controls whether missing group values are considered a distinct group value.

Tip: 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.

INCLUDEMISSINGGROUP=*boolean*
specifies whether missing values of the group variable are included in the plot.

Default: TRUE

Interaction: For this option to take effect, the GROUP= option must also be specified.

Tip: Unless a discrete attribute map is in effect or the INDEX= option is used, the attributes of the missing group value are determined by the GraphMissing style element except when the MISSING= system option is used to specify a non-default missing character or when a user-defined format is applied to the missing group value. In those cases, the attributes of the missing group value are determined by a GraphData1–GraphDataN style element.

INDEX=*numeric-column* | *expression*
specifies indices for mapping line attributes (color and pattern) to one of the GraphData1–GraphDataN style elements.

Default: no default

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.

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 144.

LEGENDLABEL= *"string"*
specifies a label for the legend item that is associated with this plot.

Default: The *string* specified on the NAME= option.

Restriction: This option applies only to an associated DISCRETELEGEND on page 681 statement.

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 849 for the syntax on using a *style-element* and "Line Options" on page 850 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 562.

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 557.

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 557.

Chapter 43

SCATTERPLOT Statement

Description and Syntax

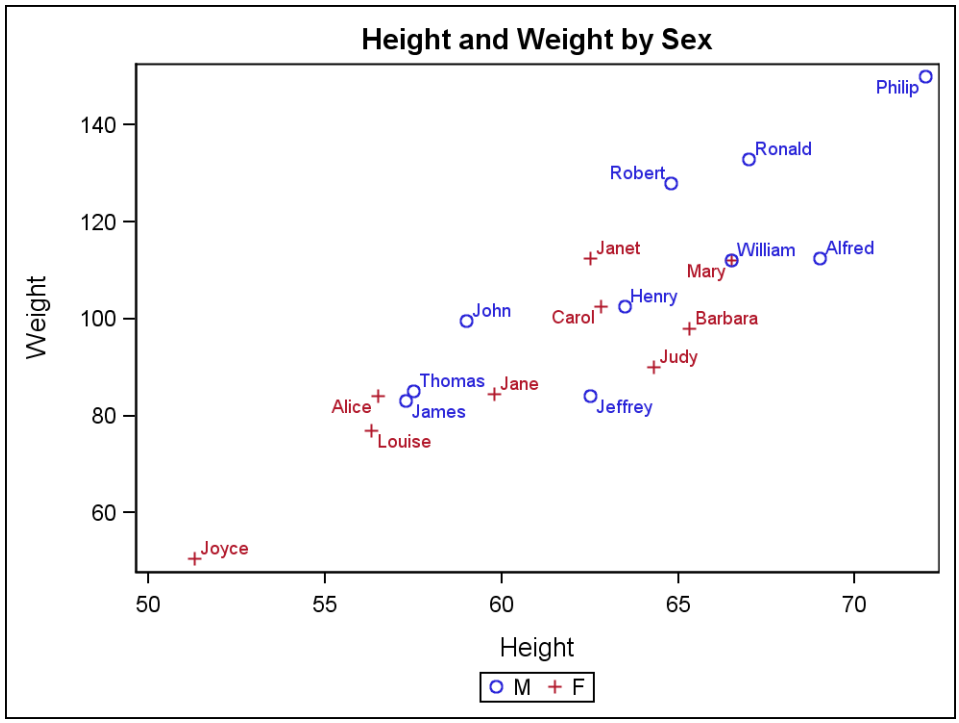
Creates a scatter plot of input data.

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 460:



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
CLUSTERWIDTH	Specifies the width of the group clusters as a fraction of the midpoint spacing.
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.
DATALABELPOSITION	Specifies the location of the data labels relative to the markers.
DATASKIN	Enhances the visual appearance of the markers.
DATATRANSPARENCY	Specifies the degree of the transparency of the markers, data labels, and error bars, when displayed.
DISCRETEMARKERSIZE	Specifies the size of a marker as a fraction of the tick spacing.
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.
GROUPDISPLAY	Specifies how marker groups are positioned for the coordinate pairs.
GROUPORDER	Specifies the relative position of the markers for a clustered group display.
INCLUDEMISSINGGROUP	Specifies whether missing values of the group variable are included in the plot.
INDEX	Specifies indices for mapping marker attributes (color and symbol) to one of the GraphData1–GraphDataN style elements.

Statement Option	Description
LEGENDLABEL	Specifies a label for a legend.
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.
MARKERSIZEMAX	Specifies a drawing size for the largest marker when the marker size represents response values.
MARKERSIZEMIN	Specifies a drawing size for the smallest marker when the marker size represents response values.
MARKERSIZERESPONSE	specifies a column that is used to map the drawing size of the 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.
REVERSECOLORMODEL	Specifies whether to reverse a gradient defined by the COLORMODEL= option.
ROLENAME	Specifies user-defined roles that can be used to display information in the tooltips.
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.
USEDISCRETESIZE	Specifies that the marker size should be based on fraction of the midpoint spacing that is set by the DISCRETEMARKERSIZE= option.
URL	Specifies an HTML page to display when a point is selected.

Statement Option	Description
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.

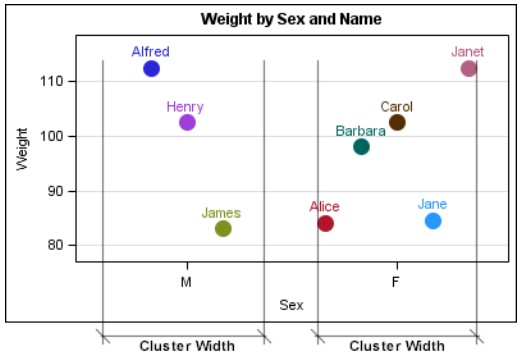
The REVERSECOLORMODEL= option can be used to reverse the start and end colors of the ramp assigned to the color model.

CLUSTERWIDTH= *number*
specifies the width of the group clusters as a fraction of the midpoint spacing on a discrete axis or a fraction of the minimum interval between adjacent data values on an interval axis.

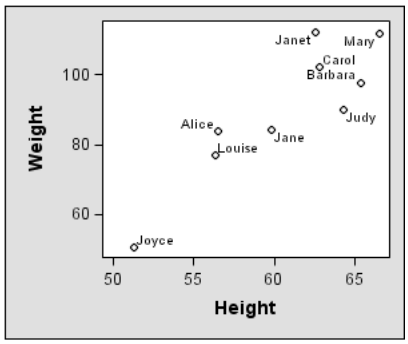
Default: 0.85

Range: 0.1 (narrowest) to 1 (widest)

Interaction: For this option to take effect, the GROUP= option must also be specified, and the GROUPDISPLAY= option must be set to CLUSTER.



DATALABEL=*column* | *expression*
specifies a column for marker labels. The label positions are adjusted to prevent them from overlapping.



Default: no data labels are displayed

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 849 for the syntax on using a *style-element* and "Text Options" on page 852 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.

DATALABELPOSITION = AUTO | TOPRIGHT | TOP | TOPLEFT | LEFT | CENTER | RIGHT | BOTTOMLEFT | BOTTOM | BOTTOMRIGHT
specifies the location of the data labels relative to the markers.

Default: AUTO

DATASKIN= NONE | PRESSED | SHEEN | CRISP | GLOSS | MATTE
enhances the visual appearance of the markers.

Default: NONE

Interaction: When a data skin is applied, all marker outlines are set by the skin, and the outline settings are ignored from the ODS style or from MARKERATTRS= option.

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)

DISCRETEMARKERSIZE= *number*
specifies the size of a marker as a fraction of the tick spacing.

Default: 0.5

Range: 0 to 1

Requirement: For this option to take effect, at least one of the axes must be discrete.

Interaction: If both of the axes are discrete, then marker size is a fraction of the smaller tick spacing.

Interaction: For this option to take effect, the USEDISCRETESIZE= option must be set to TRUE (the default is FALSE).

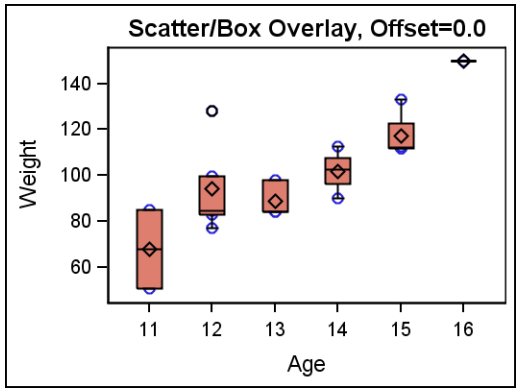
DISCRETEOFFSET=*number*
specifies an amount to offset all markers from discrete X values, or discrete Y values, or both.

Default: 0 (no offset, all markers are centered on the discrete X values, or discrete Y values, or both)

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.

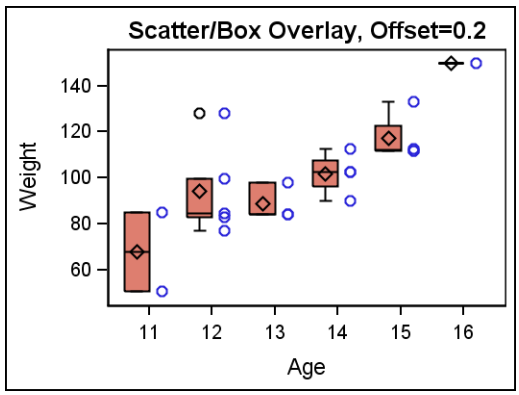
Details: This feature is useful for graphing multiple response variables side by side on a common axis. By default within an overlay-type layout, if a SCATTERPLOT is used with other plots with a discrete axis, the markers are centered on the discrete X values, or discrete Y values, or both. Depending 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 849 for the syntax on using a *style-element* and "Line Options" on page 850 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* | *discrete-attr-var* | *expression*
creates a separate marker type for each unique group value of the specified column.

discrete-attr-var
specifies a discrete attribute variable that is defined in a DISCRETEATTRVAR statement.

Restriction: A discrete attribute variable specification must be a direct reference to the attribute variable. It cannot be set by a dynamic variable.

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 INCLUDEMISSINGGROUP option controls whether missing group values are considered a distinct group value.

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 MARKERSIZERESPONSE= option overrides this option's SIZE= setting.

Tip: 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.

GROUPDISPLAY=OVERLAY | CLUSTER
specifies how marker groups are positioned for the coordinate pairs.

Default: OVERLAY

OVERLAY
draws markers for a given group value at the exact coordinate. Depending on the data, markers at a given coordinate might overlap.

CLUSTER
draws markers for a given group value adjacent to each other.

Tip: Use the CLUSTERWIDTH= option to control the width of the clusters when CLUSTER is in effect.

GROUPORDER=DATA | ASCENDING | DESCENDING
specifies the relative position of the markers for clustered group display. It also sets the default order of the groups in the legend.

Default: DATA

DATA
shows each group within a category in data order of the group column.

ASCENDING
shows each group within a category in ascending order of the group column. See the Details for more information.

DESCENDING
shows each group within a category in descending order of the group column. See the Details for more information.

Interaction: For this option to take effect, the GROUP= must also be specified.

Interaction: The SORTORDER= setting in a DISCRETELEGEND statement overrides the legend item order.

Note: Attributes such as color, symbol, and pattern are assigned to each group in the DATA order by default regardless of the GROUPORDER= option setting.

Note: The ASCENDING and DESCENDING settings linguistically sort the group values within each category (or X value) for display position purposes only. The data order of the observations and the visual attributes that are assigned to the group values remain unchanged.

Tip: The CLUSTERWIDTH= option can be used to control the distance between the group markers in a cluster

Tip: The INDEX= option can be used to alter the default sequence of visual attributes that is assigned to the group values.

Details: By default, the value that is set for this option determines the order in which attributes are assigned to each group. It also determines the order in which the groups are displayed in the legend.

For ASCENDING and DESCENDING sort orders, the order is based on the data type:

- For Numeric data, the order is based on unformatted values.
- For Character data the order is based on formatted values.

The ASCENDING and DESCENDING settings perform a linguistic sort on the group items and have the same effect as sorting the input data. However, the data is not changed.

INCLUDEMISSINGGROUP=*boolean*
specifies whether missing values of the group variable are included in the plot.

Default: TRUE

Interaction: For this option to take effect, the GROUP= option must also be specified.

Tip: Unless a discrete attribute map is in effect or the INDEX= option is used, the attributes of the missing group value are determined by the GraphMissing style element except when the MISSING= system option is used to specify a non-default missing character or when a user-defined format is applied to the missing group value. In those cases, the attributes of the missing group value are determined by a GraphData1–GraphDataN style element.

INDEX=*numeric-column* | *expression*
specifies indices for mapping marker attributes (color and symbol) to one of the GraphData1–GraphDataN style elements.

Default: no default

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.

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 144.

LEGENDLABEL= *"string"*
specifies a label for the legend item that is associated with this plot.

Default: The Y-variable label. If a label is not defined, the Y-variable name.

Restriction: This option applies only to an associated DISCRETELEGEND statement.

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 849 for the syntax on using a *style-element* and "Marker Options" on page 851 for available *marker-options*.

Default:

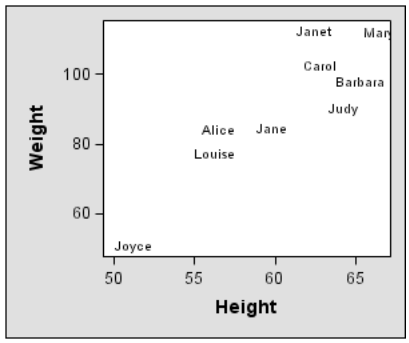
- For non-grouped data, the GraphDataDefault style element.
- For grouped data, the MarkerSymbol and ContrastColor attributes of the GraphData1–GraphDataN style elements, and the GraphDataDefault:MarkerSize style reference.

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.

Interaction: If the DATASKIN= option applies a data skin, marker outlines are determined by the data skin and the outline settings are ignored from the ODS style or the marker attributes.

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 849 for the syntax on using a *style-element* and "Text Options" on page 852 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* | *range-attr-var* | *expression*

specifies the column that is used to map marker colors to a continuous gradient.

range-attr-var

specifies a range attribute variable that is defined in a RANGEATTRVAR statement.

Restriction: A range attribute variable specification must be a direct reference to the attribute variable. It cannot be set as a dynamic variable.

Tip: The marker colors are derived from the RANGEALTCOLOR= or RANGEALTCOLORMODEL= option in the RANGEATTRMAP block RANGE statements.

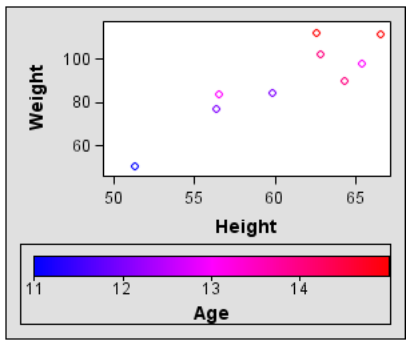
Default: ThreeColorAltRamp style element

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.

Tip: The COLORMODEL= option allows a different color range to be used.

Tip: 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.



Tip: If the MARKERCHARACTER= option is also used, the gradients that would be applied to the markers are applied to the text strings.

MARKERSIZEMAX= *dimension*
specifies a drawing size for the largest marker when the marker size represents response values.

Default: Three times the size that is indicated by the MARKERSIZEMIN= option.

Restriction: The MARKERSIZEMAX= value must be greater than the MARKERSIZEMIN= value.

Interaction: For this option to take effect, the MARKERSIZERESPONSE= option must also be used.

Tip: If you specify this size as a percent, the specified value is interpreted as a percent of the graph's height. You can control the height using the DESIGNHEIGHT= option of the BEGINGRAPH statement, or the HEIGHT= option of the ODS GRAPHICS statement. For a standard 640px by 480px output size, a percentage value of 4.5% sets a maximum size of about 21px, which is approximately the same marker size that would result from this option's typical default setting.

MARKERSIZEMIN= *dimension*
specifies a drawing size for the smallest marker when the marker size represents response values.

Default: The GraphDataDefault:MarkerSize setting, which is typically 7px.

Restriction: The MARKERSIZEMIN= value must be less than the MARKERSIZEMAX= value.

Interaction: For this option to take effect, the MARKERSIZERESPONSE= option must also be used.

Tip: If you specify this size as a percent, the specified value is interpreted as a percent of the graph's height. You can control the height using the DESIGNHEIGHT= option of the BEGINGRAPH statement, or the HEIGHT= option of the ODS GRAPHICS statement. For a standard 640px by 480px output size, a percentage value of 1.5% sets a minimum size of about 7px, which is approximately the same marker size that would result from this option's typical default setting.

MARKERSIZERESPONSE= *numeric-column* | *expression*
specifies a column that is used to map the drawing size of the markers. By default, the minimum and maximum values of this column establish a range over which the marker sizes vary in linear proportion. The actual drawing size of the smallest and largest marker is set automatically.

Default: The GraphDataDefault:MarkerSize setting, which is typically 7px.

Interaction: This option overrides the SIZE= setting in the MARKERATTRS= option.

Tip: You can adjust the smallest and largest marker size with the MARKERSIZEMIN= and MARKERSIZEMAX= options.

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 562.

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 that can be used to display information in the tooltips.

Default: no user-defined roles

(*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 460 , 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.

USEDISCRETESIZE = *boolean*
specifies that the marker size should be based on fraction of the midpoint spacing that is set by the DISCRETEMARKERSIZE= 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: 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 557.

XERRORLOWER=*numeric-column* | *expression*
specifies values for the lower endpoints on the X error bars. The error bars are drawn from the markers to the endpoints.

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. The error bars are drawn from the markers to the endpoints.

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 557.

YERRORLOWER=*numeric-column* | *expression*
specifies values for the lower endpoints on the Y error bars. The error bars are drawn from the markers to the endpoints.

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. The error bars are drawn from the markers to the endpoints.

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 44

SCATTERPLOTMATRIX Statement

Description and Syntax

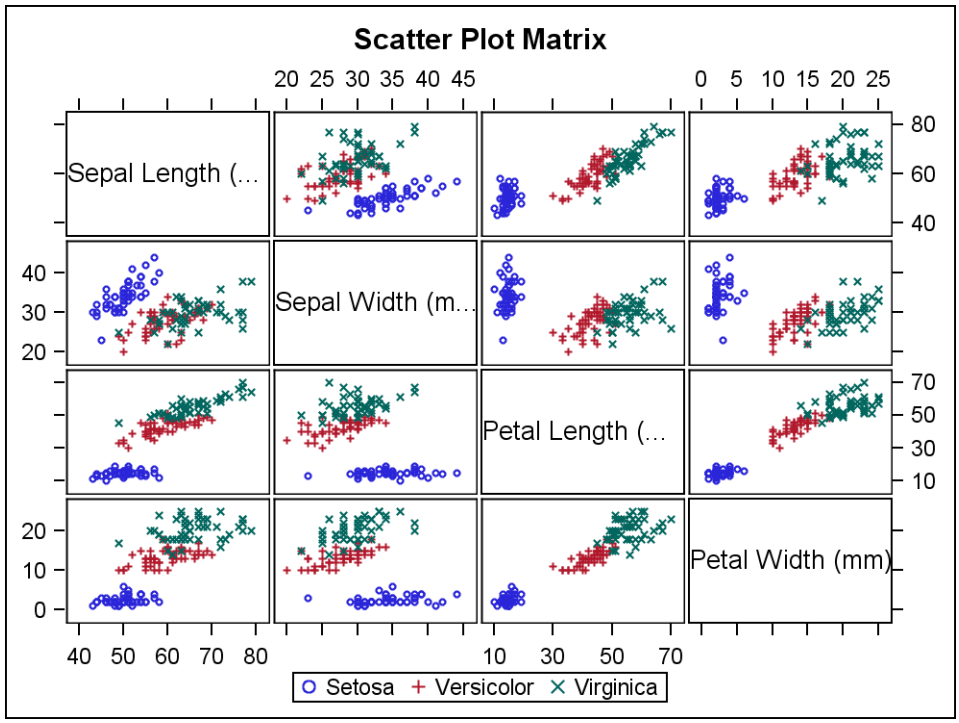
Creates a matrix of all pairwise scatter plots of the specified variables.

SCATTERPLOTMATRIX *numeric-column-list </option(s)>*

Example Program and Statement Details

Example Graph

The following graph was generated by the "Example Program" on page 478:



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.
DATALABELPOSITION	Specifies the location of the data labels relative to the markers.
DATATRANSPARENCY	Specifies the degree of the transparency of the markers.

Statement Option	Description
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.
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.
INCLUDEMISSINGGROUP	Specifies whether missing values of the group variable are included in the plot.
INDEX	Specifies indices for mapping marker attributes (color and symbol) to one of the GraphData1–GraphDataN 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 user-defined roles that can be used to display information in the tooltips.

Statement Option	Description
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.
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.

For statistical and computational details of these options, see PROC CORR in the documentation for Base SAS.

DATALABEL=*column*
specifies a column for marker labels. The label positions are adjusted to prevent the labels from overlapping.

Default: no data labels are displayed

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.

DATALABELPOSITION = AUTO | TOPRIGHT | TOP | TOPLEFT | LEFT | CENTER | RIGHT | BOTTOMLEFT | BOTTOM | BOTTOMRIGHT
specifies the location of the data labels relative to the markers.

Default: AUTO

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 849 for the syntax on using a *style-element* and "Text Options" on page 852 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: LABEL. 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 301.

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* | *discrete-attr-var* | *expression*
creates a distinct set of scatter markers, error bars, and data labels for each unique group value of the specified column.

discrete-attr-var
specifies a discrete attribute variable that is defined in a DISCRETEATTRVAR statement.

Restriction: A discrete attribute variable specification must be a direct reference to the attribute variable. It cannot be set by a dynamic variable.

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 INCLUDEMISSINGGROUP= option controls whether missing group values are considered a distinct group value.

Tip: 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.

INCLUDEMISSINGGROUP=*boolean*
specifies whether missing values of the group variable are included in the plot.

Default: TRUE

Interaction: For this option to take effect, the GROUP= option must also be specified.

Tip: Unless a discrete attribute map is in effect or the INDEX= option is used, the attributes of the missing group value are determined by the GraphMissing style element except when the MISSING= system option is used to specify a non-default missing character or when a user-defined format is applied to the missing group value. In those cases, the attributes of the missing group value are determined by a GraphData1–GraphDataN style element.

INDEX=*numeric-column* | *expression*
specifies indices for mapping marker attributes (color and symbol) to one of the GraphData1–GraphDataN style elements.

Default: no default

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.

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 144.

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)

For statistical and computational details of these options, see PROC CORR in the documentation for Base SAS.

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 849 for the syntax on using a *style-element* and "Text Options" on page 852 for available *text-options*.

Default: The GraphDataText style element.

TITLE= "*string*"
specifies a title for the inset. The title is added at the top of the inset and spans the full inset width.

Default: no default, and space is not reserved for the title when it is not set

Tip: Text properties for the title string can be set with TITLEATTRS=.

TITLEATTRS=*style-element* | *style-element* (*text-options*) | (*text-options*)
specifies the text properties of the inset's title string. See "General Syntax for Attribute Options" on page 849 for the syntax on using a *style-element* and "Text Options" on page 852 for available *text-options*.

Default: The GraphValueText 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 849 for the syntax on using a *style-element* and "Marker Options" on page 851 for available *marker-options*.

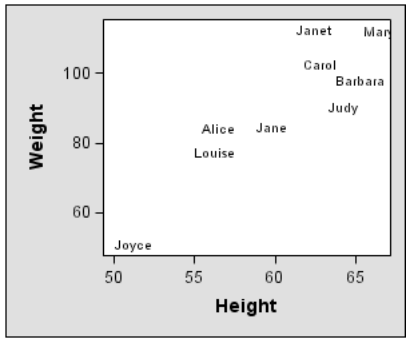
Default:

- 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 849 for the syntax on using a *style-element* and "Text Options" on page 852 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* | *range-attr-var* | *expression*
specifies the column that is used to map marker colors to a continuous gradient.

range-attr-var
specifies a range attribute variable that is defined in a RANGEATTRVAR statement.

Restriction: A range attribute variable specification must be a direct reference to the attribute variable. It cannot be set as a dynamic variable.

Tip: The marker colors are derived from the RANGEALTCOLOR= or RANGEALTCOLORMODEL= option in the RANGEATTRMAP block RANGE statements.

Default: ThreeColorAltRamp style element

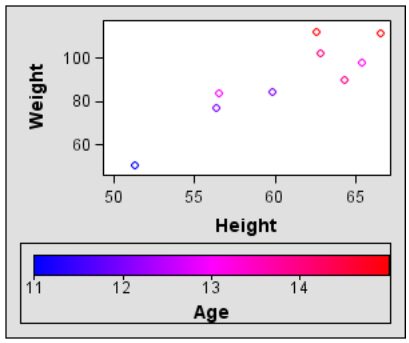
Restriction: To display a legend with this option in effect, you must use a CONTINUOUSLEGEND statement, not a DISCRETELEGEND statement.

Interaction: This option overrides the COLOR= setting of the MARKERATTRS= or MARKERCHARACTERATTRS= option.

Interaction: The DATALABELATTRS= option overrides the gradient colors specified by this option for the data labels.

Tip: 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.

Tip: The COLORMODEL= option allows a different color range to be used.



Tip: If the MARKERCHARACTER= option is also used, the gradients that would be applied to the markers are applied to the text strings.

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 that can be used to display information in the tooltips.

Default: no user-defined roles

(*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 45

SERIESPLOT Statement

Description and Syntax

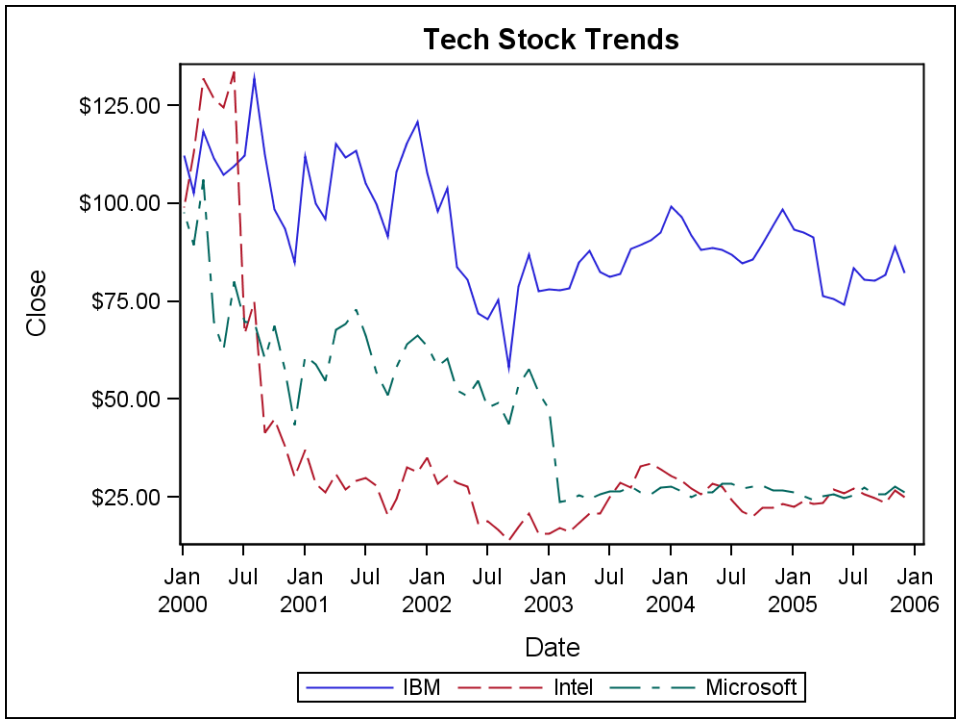
Displays a series of line segments that connect observations of input data.

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 494:



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;

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.
CLUSTERWIDTH	Specifies the width of the group clusters as a fraction of the midpoint spacing.
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.
DATALABELPOSITION	Specifies the location of the data labels relative to the vertices of the series and the markers, when displayed.
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.
GROUPDISPLAY	Specifies how marker groups are positioned for the coordinate pairs.
GROUPORDER	Specifies the relative position of the series lines and markers for a clustered group display.

Statement Option	Description
INCLUDEMISSINGGROUP	Specifies whether missing values of the group variable are included in the plot.
INDEX	Specifies indices for mapping series-line attributes (color, marker symbol, and line pattern) to one of the GraphData1–GraphDataN style elements.
LEGENDLABEL	Specifies a label for 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.
PRIMARY	Specifies that the data columns for this plot be used for determining default axis features.
ROLENAME	Specifies user-defined roles that can be used to display information in the tooltips.
SMOOTHCONNECT	Specifies that a smoothed line passes through all vertices.
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.

CLUSTERWIDTH= *number*
specifies the width of the group clusters as a fraction of the midpoint spacing on a discrete axis or a fraction of the minimum interval between adjacent data values on an interval axis.

Default: 0.85

Range: 0.1 (narrowest) to 1 (widest)

Interaction: For this option to take effect, the GROUP= option must also be specified, and GROUPDISPLAY =CLUSTER must be set..

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 curve label is displayed

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 849 for the syntax on using a *style-element* and "Text Options" on page 852 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 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 146.

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 ignored if CURVELABELLOCATION=INSIDE is specified. The START and END settings are ignored 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 146.

DATALABEL=*column* | *expression*
specifies a column that supplies values for the data point labels.

Default: no data labels are displayed

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 849 for the syntax on using a *style-element* and "Text Options" on page 852 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.

DATALABELPOSITION = AUTO | TOPRIGHT | TOP | TOPLEFT | LEFT | CENTER | RIGHT | BOTTOMLEFT | BOTTOM | BOTTOMRIGHT
specifies the location of the data labels relative to the vertices of the series and the markers, when displayed.

Default: AUTO

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 discrete X values, or discrete Y values, or both. This option is useful when graphing multiple response variables side by side on a common axis

Default: 0 (no offset, all series lines and markers are centered on the discrete X values, or discrete Y values, or both)

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* | *discrete-attr-var* | *expression*
creates a separate series plot for each unique group value of the specified column.

discrete-attr-var
specifies a discrete attribute variable that is defined in a DISCRETEATTRVAR statement.

Restriction: A discrete attribute variable specification must be a direct reference to the attribute variable. It cannot be set by a dynamic variable.

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.

Interaction: The INCLUDEMISSINGGROUP= option controls whether missing group values are considered a distinct group value.

Tip: 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.

GROUPDISPLAY= OVERLAY | CLUSTER
specifies how marker groups are positioned for the coordinate pairs.

Default: OVERLAY

OVERLAY
draws markers for a given group value at the exact coordinate. Depending on the data, markers at a given coordinate might overlap.

CLUSTER
draws markers for a given group value adjacent to each other.

Interaction: For this option to take effect, the GROUP= option must also be specified.

Tip: Use the CLUSTERWIDTH= option to control the width of the clusters when CLUSTER is in effect.

GROUPORDER=DATA | ASCENDING | DESCENDING
specifies the relative position of the series lines and markers for clustered group display. It also sets the default order of the groups in the legend.

Default: DATA

DATA
shows each group within a category in data order of the group column.

ASCENDING
shows each group within a category in ascending order of the group column. See the Details for more information.

DESCENDING
shows each group within a category in descending order of the group column. See the Details for more information.

Interaction: For this option to take effect, the GROUP= must also be specified.

Interaction: The SORTORDER= setting in a DISCRETELEGEND statement overrides the legend item order.

Note: Attributes such as color, symbol, and pattern are assigned to each group in the DATA order by default regardless of the GROUPORDER= option setting.

Note: The ASCENDING and DESCENDING settings linguistically sort the group values within each category (or X value) for display position purposes only. The data order of the observations and the visual attributes that are assigned to the group values remain unchanged.

Tip: The CLUSTERWIDTH option can be used to control the distance between the group markers in a cluster.

Tip: The INDEX= option can be used to alter the default sequence of visual attributes that is assigned to the group values.

Details: By default, the value that is set for this option determines the order in which attributes are assigned to each group. It also determines the order in which the groups are displayed in the legend.

For ASCENDING and DESCENDING sort orders, the order is based on the data type:

- For Numeric data, the order is based on unformatted values.
- For Character data the order is based on formatted values.

The ASCENDING and DESCENDING settings perform a linguistic sort on the group items and have the same effect as sorting the input data. However, the data is not changed.

INCLUDEMISSINGGROUP=*boolean*
specifies whether missing values of the group variable are included in the plot.

Default: TRUE

Interaction: For this option to take effect, the GROUP= option must also be specified.

Tip: Unless a discrete attribute map is in effect or the INDEX= option is used, the attributes of the missing group value are determined by the GraphMissing style element except when the MISSING= system option is used to specify a non-default missing character or when a user-defined format is applied to the missing group value. In those cases, the attributes of the missing group value are determined by a GraphData1–GraphDataN style element.

INDEX=*numeric-column* | *expression*
specifies indices for mapping series-line attributes (color, marker symbol, and line pattern) to one of the GraphData1–GraphDataN style elements.

Default: no default

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.

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 144.

LEGENDLABEL= *"string"*
specifies a label for the legend item that is associated with this plot.

Default: The Y-variable label. If a label is not defined, the Y-variable name.

Restriction: This option applies only to an associated DISCRETELEGEND statement.

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 849 for the syntax on using a *style-element* and "Line Options" on page 850 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 849 for the syntax on using a *style-element* and "Marker Options" on page 851 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 562.

ROLENAME=(*role-name-list*)
specifies user-defined roles that can be used to display information in the tooltips.

Default: no user-defined roles

(*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.

SMOOTHCONNECT=*boolean*
specifies that a smoothed line passes through all vertices.

Default: FALSE

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* is a valid reference.

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 557.

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 557.

Chapter 46

STEPPLOT Statement

Description and Syntax

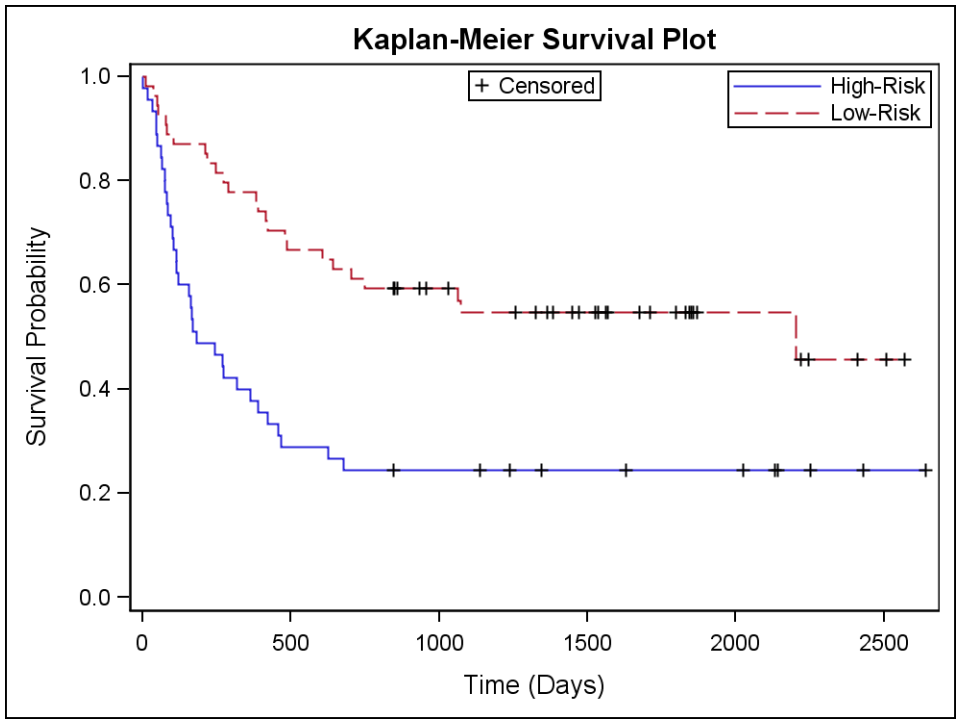
Displays a series of horizontal and vertical line segments that connect observations of input data.

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 508:



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.
CLUSTERWIDTH	Specifies the width of the group clusters as a fraction of the midpoint spacing.
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.
DATALABELPOSITION	Specifies the location of the data labels relative to the data points and markers, when displayed.
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 when graphing multiple response variables side by side on a common axis.

Statement Option	Description
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.
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.
GROUPDISPLAY	Specifies whether grouped step lines and markers are overlaid or clustered around the category midpoints.
GROUPORDER	Specifies the relative position of the step lines and markers for a group display.
INCLUDEMISSINGGROUP	Specifies whether missing values of the group variable are included in the plot.
INDEX	Specifies indices for mapping step-line attributes (color, marker symbol, and line pattern) to one of the GraphData1–GraphDataN 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 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 user-defined roles that can be used to display information in the tooltips.
TIP	Specifies the information to display when the cursor is positioned over the step line.

Statement Option	Description
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.

CLUSTERWIDTH= *number*
specifies the width of the group clusters as a fraction of the midpoint spacing on a discrete axis or a fraction of the minimum interval between adjacent data values on an interval axis.

Default: 0.85

Range: 0.1 (narrowest) to 1 (widest)

Interaction: For this option to take effect, the GROUP= option must also be specified, and the GROUPDISPLAY= option must be set to CLUSTER.

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 curve label is displayed

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 849 for the syntax on using a *style-element* and "Text Options" on page 852 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 146.

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 ignored if CURVELABELLOCATION= INSIDE is specified. The START and END settings are ignored 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 step label appears. For more information, see "Location and Position of Curve Labels" on page 146.

DATALABEL=*column* | *expression*
specifies a column that supplies values for the data point labels.

Default: no data labels are displayed

The label positions are adjusted to prevent the labels from overlapping.

DATALABELPOSITION = AUTO | TOPRIGHT | TOP | TOPLEFT | LEFT | CENTER | RIGHT | BOTTOMLEFT | BOTTOM | BOTTOMRIGHT
specifies the location of the data labels relative to the data points and markers, when displayed.

Default: AUTO

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 849 for the syntax on using a *style-element* and "Text Options" on page 852 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 discrete X values when graphing multiple response variables side by side on a common axis.

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 849 for the syntax on using a *style-element* and "Line Options" on page 850 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* | *discrete-attr-var* | *expression*
creates a distinct set of lines, markers, and data labels for each unique group value of the specified column.

discrete-attr-var
specifies a discrete attribute variable that is defined in a DISCRETEATTRVAR statement.

Restriction: A discrete attribute variable specification must be a direct reference to the attribute variable. It cannot be set by a dynamic variable.

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.

Interaction: The INCLUDEMISSINGGROUP= option controls whether missing group values are considered a distinct group value.

Tip: 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.

GROUPDISPLAY= OVERLAY | CLUSTER
specifies whether grouped step lines and markers are overlaid or clustered around the category midpoints.

Default: OVERLAY

OVERLAY
step lines and markers for matching category values are centered on the midpoints and superimposed on each other.

CLUSTER
step lines and markers for matching category values are clustered around the midpoints on a discrete axis or around the intervals on an interval axis. Each set of group values is centered at the midpoint for the category.

Interaction: For this option to take effect, the GROUP= option must also be specified.

GROUPORDER=DATA | ASCENDING | DESCENDING
specifies the relative position of the step lines and markers for group display. It also sets the default order of the groups in the legend.

Default: DATA

DATA
shows each group within a category in data order of the group column.

ASCENDING
shows each group within a category in ascending order of the group column. See the Details for more information.

DESCENDING
shows each group within a category in descending order of the group column. See the Details for more information.

Interaction: For this option to take effect, the GROUP= must also be specified.

Interaction: The SORTORDER= setting in a DISCRETELEGEND statement overrides the legend item order.

Note: Attributes such as color, symbol, and pattern are assigned to each group in the DATA order by default regardless of the GROUPORDER= option setting.

Note: The ASCENDING and DESCENDING settings linguistically sort the group values within each category (or X value) for display position purposes only. The data order of the observations and the visual attributes that are assigned to the group values remain unchanged.

Tip: The INDEX= option can be used to alter the default sequence of visual attributes that is assigned to the group values.

Details: By default, the value that is set for this option determines the order in which attributes are assigned to each group. It also determines the order in which the groups are displayed in the legend.

For ASCENDING and DESCENDING sort orders, the order is based on the data type:

- For Numeric data, the order is based on unformatted values.
- For Character data the order is based on formatted values.

The ASCENDING and DESCENDING settings perform a linguistic sort on the group items and have the same effect as sorting the input data. However, the data is not changed.

INCLUDEMISSINGGROUP=*boolean*
specifies whether missing values of the group variable are included in the plot.

Default: TRUE

Interaction: For this option to take effect, the GROUP= option must also be specified.

Tip: Unless a discrete attribute map is in effect or the INDEX= option is used, the attributes of the missing group value are determined by the GraphMissing style element except when the MISSING= system option is used to specify a non-default missing character or when a user-defined format is applied to the missing group value. In those cases, the attributes of the missing group value are determined by a GraphData1–GraphDataN style element.

INDEX=*numeric-column* | *expression*
specifies indices for mapping step-line attributes (color, marker symbol, and line pattern) to one of the GraphData1–GraphDataN style elements.

Default: no default

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.

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 144.

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 the legend item that is associated with this plot.

Default: The Y-variable label. If a label is not defined, the Y-variable name.

Restriction: This option applies only to an associated DISCRETELEGEND statement.

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 849 for the syntax on using a *style-element* and "Line Options" on page 850 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 849 for the syntax on using a *style-element* and "Marker Options" on page 851 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 562.

ROLENAME=(*rolename-list*)
specifies user-defined roles that can be used to display information in the tooltips.

Default: no user-defined roles

(*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* is a valid reference.

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 557.

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 557.

Chapter 47

SURFACEPLOTPARM Statement

Description and Syntax

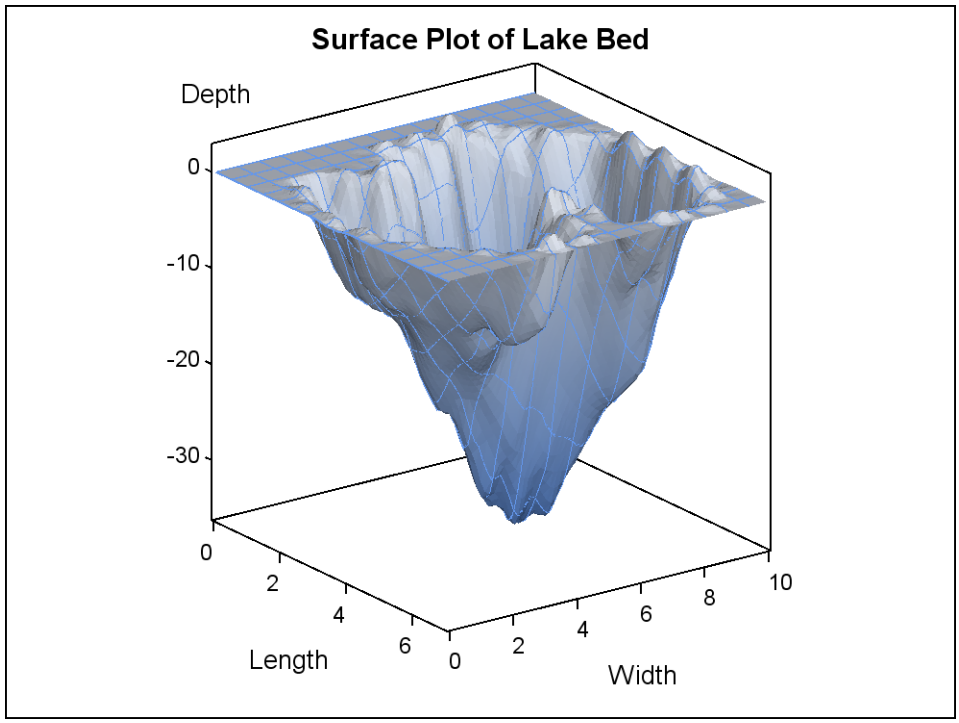
Creates a three-dimensional surface representing a response variable evaluated over a grid of X and Y values.

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 524:



Example Program

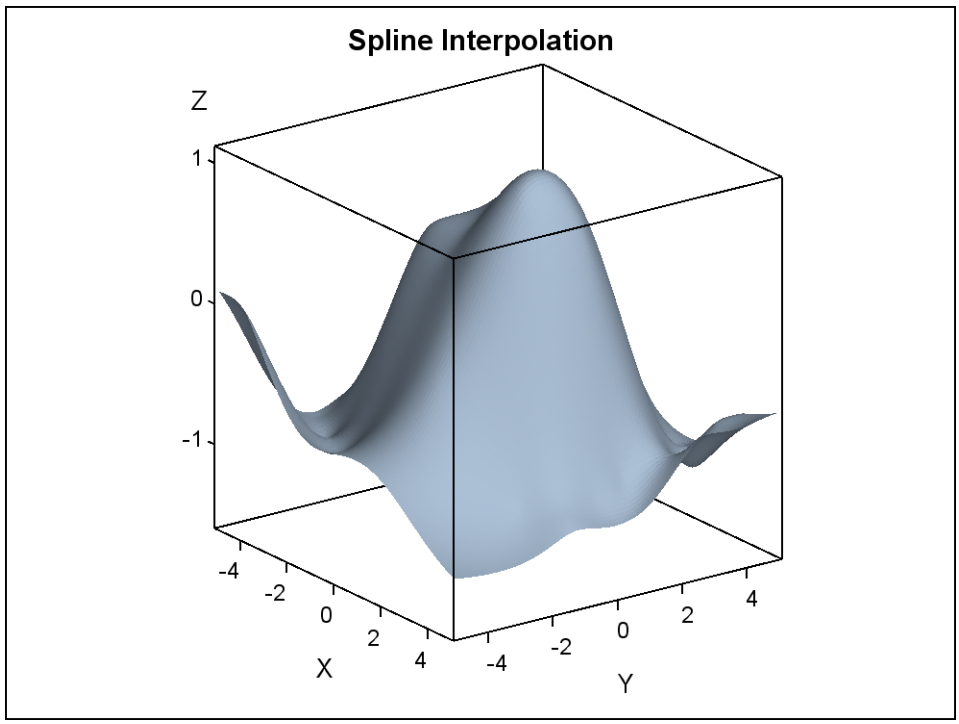
```
proc template;
  define statgraph surfaceplotparm;
    begingraph;
      entrytitle "Surface Plot of Lake Bed";
      layout overlay3d / cube=false;
        surfaceplotparm x=length y=width z=depth /
          reversecolormodel=true
          surfacecolorgradient=depth
          colormodel=twocoloraltramp;
      endlayout;
    endgraph;
  end;

/* create gridded data for surface
 * proc g3grid is a sas/graph procedure */
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.

The G3GRID procedure (requires a SAS/GRAPH license) can be used 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:

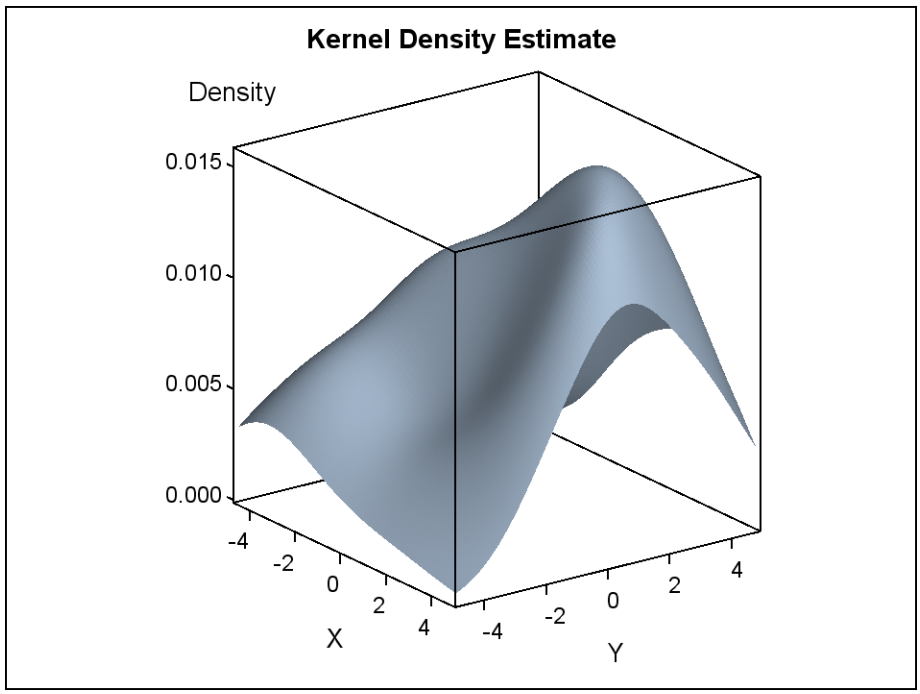



```
data nums;
  do i=1 to 30;
     X=10*ranuni(33)-5;
     Y=10*ranuni(35)-5;
     Z=sin(sqrt(x*x+y*y));
     output;
  end;
run;
proc g3grid data=nums out=gridded;
  grid y*x=z / spline
               axis1=-5 to 5 by .1
               axis2=-5 to 5 by .1;
run;
proc sort data=gridded; by y x; run;

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 the SAS/STAT user's guide.

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 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 849 for the syntax on using a *style-element* and "Fill Options" on page 850 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 the legend item that is associated with this plot.

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.

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 562.

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 48

VECTORPLOT Statement

Description and Syntax

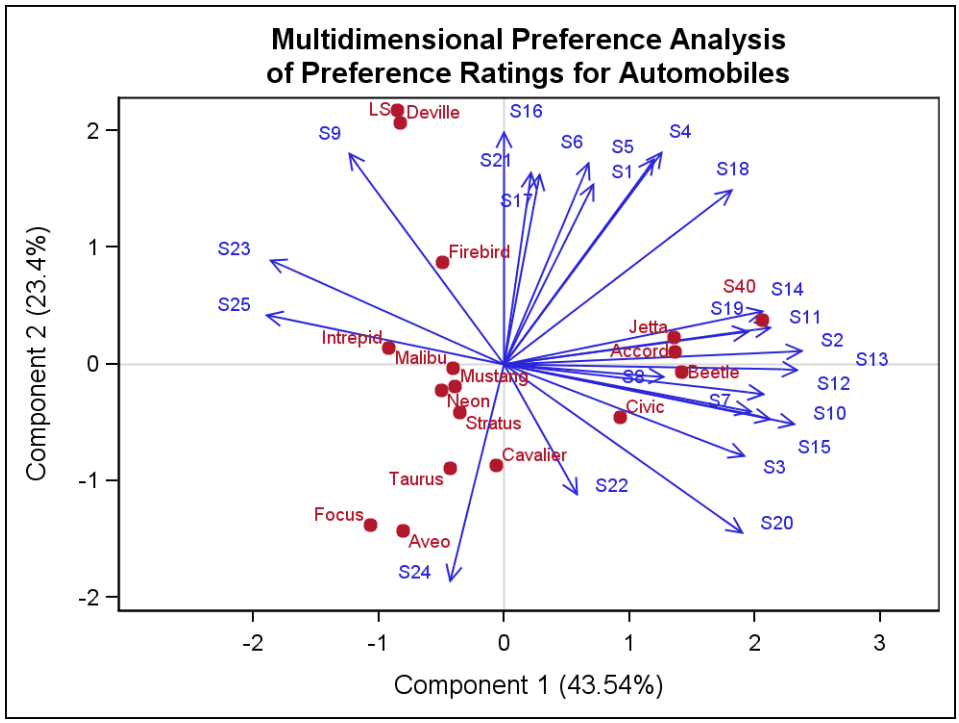
Creates a plot of vectors (directed line segments).

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 532:



Example Program

```
data CarPref;
  input Make $12. Model $13. (S1-S25) (1.);
datalines;
Cadillac    Deville      800799049124050897109380 9
Chevrolet   Aveo          005120042345104300351569 8
Chevrolet   Cavalier     405330581416164354474779 5
Chevrolet   Malibu       602740072312134554566865 8
Dodge       Intrepid     700600043410110733345870 8
Dodge       Stratus      300500563546130244467565 5
Dodge       Neon         400500361402160275447655 5
Ford        Taurus       202400671502144353064865 5
Ford        Mustang      500719770502110185065755 5
Ford        Focus        002100030303020150051407 8
Honda       Accord       595689760969995299897507 8
Honda       Civic        483670950748885256776507 5
Lincoln     LS           700899059223040996209190 9
Pontiac     Firebird     010789561320120695826590 7
Volkswagen  Jetta        485869650887779537789500 0
Volkswagen  Beetle       485850970969579548788500 0
Volvo       S40          998999890999998798991900 0
;
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;
    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.
DATALABELPOSITION	Specifies the location of the data labels relative to the end points and arrow heads
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.
INCLUDEMISSINGGROUP	Specifies whether missing values of the group variable are included in the plot.
INDEX	Specifies indices for mapping line attributes (color and pattern) to one of the GraphData1–GraphDataN style elements.
LEGENDLABEL	Specifies a label for 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 user-defined roles that can be used to display information in the tooltips.
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.

Statement Option	Description
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.

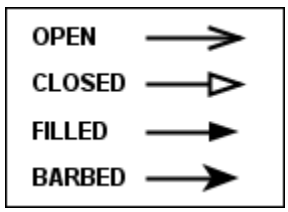
Use the ARROWHEADSHAPE= option to control arrowhead appearance.

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 data labels are displayed

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 849 for the syntax on using a *style-element* and "Text Options" on page 852 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.

DATALABELPOSITION = AUTO | TOPRIGHT | TOP | TOPLEFT | LEFT | CENTER | RIGHT | BOTTOMLEFT | BOTTOM | BOTTOMRIGHT
specifies the location of the data labels relative to the end points and arrow heads.

Default: AUTO

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* | *discrete-attr-var* | *expression*
creates a distinct set of vector lines and data label colors for each unique group value of the specified column.

discrete-attr-var
specifies a discrete attribute variable that is defined in a DISCRETEATTRVAR statement.

Restriction: A discrete attribute variable specification must be a direct reference to the attribute variable. It cannot be set by a dynamic variable.

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.

Interaction: The INCLUDEMISSINGGROUP= option controls whether missing group values are considered a distinct group value.

Tip: 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.

INCLUDEMISSINGGROUP=*boolean*
specifies whether missing values of the group variable are included in the plot.

Default: TRUE

Interaction: For this option to take effect, the GROUP= option must also be specified.

Tip: Unless a discrete attribute map is in effect or the INDEX= option is used, the attributes of the missing group value are determined by the GraphMissing style element except when the MISSING= system option is used to specify a non-default missing character or when a user-defined format is applied to the missing group value. In those cases, the attributes of the missing group value are determined by a GraphData1–GraphDataN style element.

INDEX=*numeric-column* | *expression*
specifies indices for mapping line attributes (color and pattern) to one of the GraphData1–GraphDataN style elements.

Default: no default

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.

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 144.

LEGENDLABEL= *"string"*
specifies a label for the legend item that is associated with this plot.

Default: The *string* specified on the NAME= option.

Restriction: This option applies only to an associated DISCRETELEGEND statement.

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 849 for the syntax on using a *style-element* and "Line Options" on page 850 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 562.

ROLENAME=(*role-name-list*)
specifies user-defined roles that can be used to display information in the tooltips.

Default: no user-defined roles

(*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 557.

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 557.

Chapter 49

WATERFALLCHART Statement

Description and Syntax

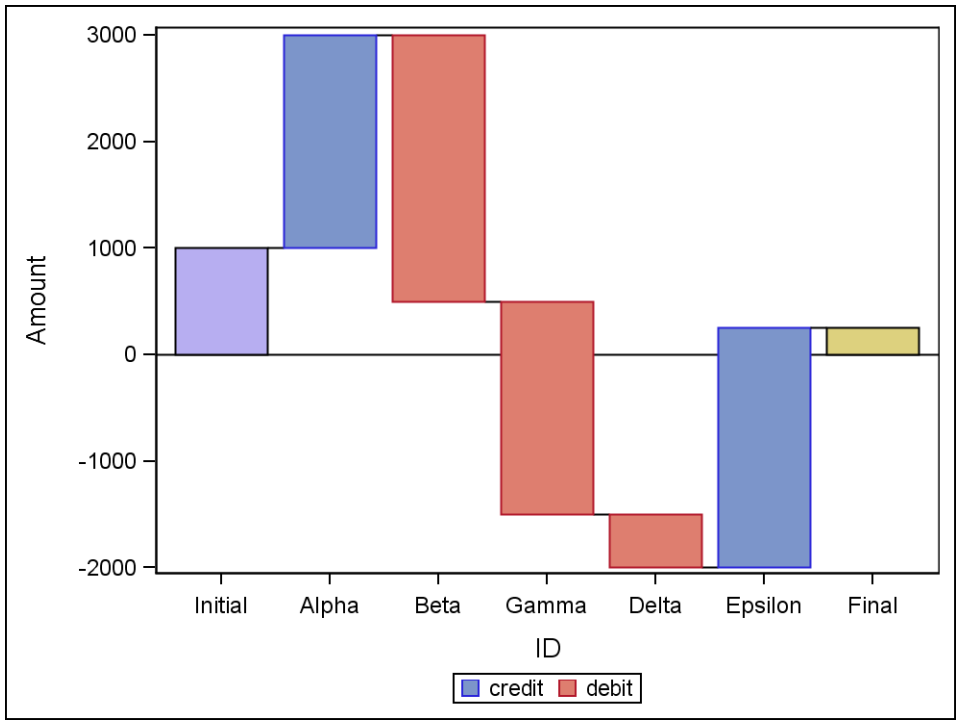
Creates a waterfall chart that is computed from input data.

WATERFALLCHART CATEGORY = *column* | *expression*
RESPONSE = *numeric-column* | *expression* </*option(s)*>;

Example Program and Statement Details

Example Graph

The following graph was generated by the "Example Program" on page 542:



Example Program

```
data transactions;
input ID $ Amount type $;
datalines;
Alpha 2000 credit
Beta -2500 debit
Gamma -2000 debit
Delta -500 debit
Epsilon 2250 credit
;
proc template;
  define statgraph waterfallchart;
    begingraph;
      layout overlay;
        waterfallchart category=id response=amount /
          colorgroup=type
          initialbarvalue=1000
          name="waterfall";
        discretelegend "waterfall";
      endlayout;
    endgraph;
  end;
run;

proc sgrender data=transactions template=waterfallchart;
run;
```

Statement Summary

The input data for the WATERFALLCHART statement is raw, unsummarized input data, and the statement calculates appropriate summarization statistics (sum or mean). By default, the bars in the chart appear in the order in which the CATEGORY values are present in the input data. A waterfall chart is typically used to show credit and debit transactions or successive changes to a given state.

In a waterfall chart, the bars that are calculated from the data are called “transaction” bars. The transaction bars represent the values of the RESPONSE variable across a series of intermediate values for the specified CATEGORY variable. You can manage the color of the transaction bars using the COLORGROUP , COLORMODEL , or COLORRESPONSE= option.

A waterfall chart can also display an “initial” bar and a “final” bar. The value of the initial bar determines the starting response value for the first transaction bar. To set the initial value, use the INITIALBARVALUE= option. If the initial bar is not displayed, the first transaction bar has a starting response value of 0. The value of the final bar is set automatically to the ending value of the last transaction bar.

Required Arguments

CATEGORY=*column* | *expression*
specifies the column or expression for the category values. Duplicated category values are summarized into a unique value. All values are treated as discrete.

RESPONSE=*numeric-column* | *expression*
specifies the numeric column or expression for the response values.

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 width of the bars as a ratio of the maximum possible width.
BASELINEINTERCEPT	Specifies the response-axis intercept for the baseline.
COLORGROUP	Specifies a column that is used to discretely color the transaction bars.
COLORMODEL	Specifies a style element to use to determine the colors of the transaction bars.

Statement Option	Description
COLORRESPONSE	Specifies the values to use to map the transaction-bar colors to a continuous color gradient.
CONNECT	Determines whether trend lines connect to the adjacent bar's starting or ending value.
CONNECTATTRS	Specifies the appearance of the trend lines that connect the bars.
CONNECTDECREASINGATTRS	Specifies the appearance of trend lines that denote a decreasing value between bars.
CONNECTINCREASINGATTRS	Specifies the appearance of trend lines that denote an increasing value between bars.
DATASKIN	Enhances the visual appearance of the filled bars.
DATATRANSPARENCY	Specifies the degree of the transparency of the bar fill, bar outline, and trend lines, if displayed.
DISPLAY	Specifies which bar features are displayed.
FILLATTRS	Specifies the appearance of the filled bar area.
FILLTYPE	Determines whether a solid or gradient fill is used in the bars.
FINALBARATTRS	Specifies the appearance of the "final" bar, if displayed.
FINALBARTICKVALUE	Specifies a tick value to use on the category axis when the "final" bar is displayed
INITIALBARATTRS	Specifies the appearance of the "initial" bar, if displayed.
INITIALBARTICKVALUE	Specifies a tick value to use on the category axis when the "initial" bar is displayed.
INITIALBARVALUE	specifies a value for the "initial" bar.
LEGENDLABEL	Specifies the label for a legends.
NAME	Assigns a name to a plot statement for reference in other template statements.
OUTLINEATTRS	Specifies the properties of the bar outlines.
ROLENAME	Specifies user-defined roles that can be used to display information in the tooltips.

Statement Option	Description
STAT	Specifies the statistic to be computed for the RESPONSE axis.
TIP	Specifies the information to display when the cursor is positioned over a bar.
TIPFORMAT	Specifies display formats for the information defined by the tooltip roles.
TIPLABEL	Specifies display labels for the information defined by the tooltip 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 849 for the syntax on using a *style-element* and "Text Options" on page 852 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 RESPONSE= variable or BEST6. if no format is assigned.

Interaction: For this option to take effect, BARLABEL=TRUE must be specified.

BARWIDTH=*number*
specifies the width of the bars 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: 0.1 (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.

BASELINEINTERCEPT=*number*
specifies the response-axis intercept for the baseline.

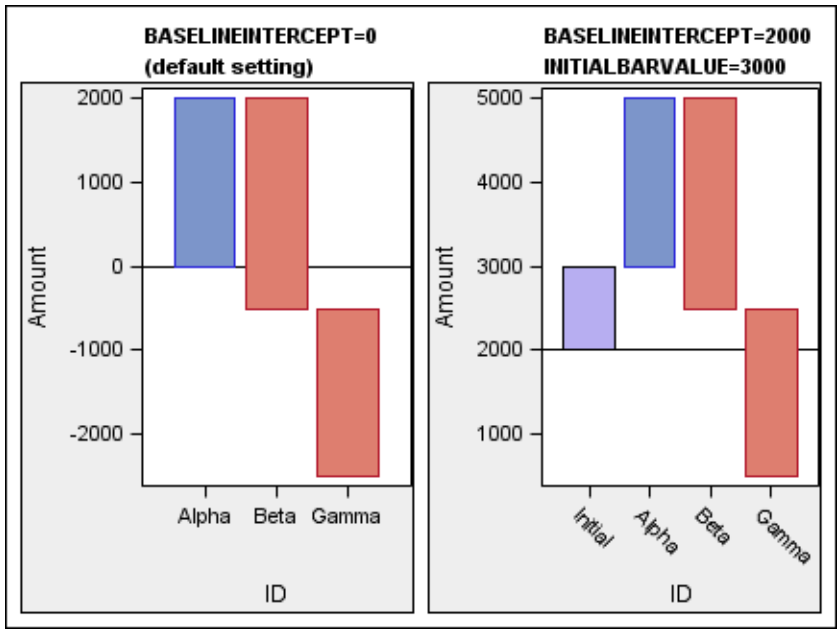
Default: 0

Interaction: The value set by this option is included in the calculation of the axis range.

Interaction: This setting affects only the chart's "initial" and "final" bars. If no initial bar value is specified, then the first transaction bar is drawn from 0, no matter what is set for the baseline value.

Tip: The baseline is always displayed in the chart, whether for a specified value or the default value. Visual attributes for the line are derived from the GraphAxisLines style-element.

Tip: Label positions are automatically adjusted to prevent the labels from overlapping.



COLORGROUP=*column* | *discrete-attr-var* | *expression*
specifies a column that is used to discretely color the transaction bars.

discrete-attr-var
specifies a discrete attribute variable that is defined in a DISCRETEATTRVAR statement.

Restriction: A discrete attribute variable specification must be a direct reference to the attribute variable. It cannot be set by a dynamic variable.

Default: no default. All the transaction bars have only one fill and one outline color, determined by the ODS style or set by the FILLATTRS= and OUTLINEATTRS= options.

Interaction: This option is ignored if the COLORRESPONSE= option is also specified.

Interaction: If a *column* or *expression* is specified, the unique column values are found and the transaction bar colors are derived from the GraphData1 - GraphDataN style elements. The COLOR attribute is used for the bar fill colors and the CONTRASTCOLOR attribute is used for the bar outline colors.

Interaction: If a variable that is associated with an attribute map is specified, then the color mapping defined by the associated DISCRETEATTRMAP statement is used for the transaction bars.

Tip: To manage the color of the "initial" bar, use the INITIALBARATTRS= option. To manage the color of the "final" bar, use the FINALBARATTRS= option.

COLORMODEL=*style-element*
specifies a style element to use to determine the colors of the transaction bars.

Default: The ThreeColorRamp 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 COLORRESPONSE= option
NEUTRALCOLOR	color for the midpoint of the range of the column that is specified on the COLORRESPONSE= option
ENDCOLOR	color for the highest data value of the column that is specified on the COLORRESPONSE= option

Interaction: For this option to take effect, the COLORRESPONSE= option must also be specified.

Tip: To manage the color of the "initial" bar, use the INITIALBARATTRS= option. To manage the color of the "final" bar, use the FINALBARATTRS= option.

COLORRESPONSE=*numeric-column* | *range-attr-var* | *expression*
specifies the values to use to map the transaction-bar colors to a continuous color gradient.

range-attr-var
specifies a range attribute variable that is defined in a RANGEATTRVAR statement.

Restriction: A range attribute variable specification must be a direct reference to the attribute variable. It cannot be set by a dynamic variable.

Default: no default.

Interaction: This option overrides the COLORGROUP= option, if it is also specified.

Interaction: The color ramp that is used is determined as follows:

- If a *numeric-column* or an *expression* is specified, then the value of the COLORMODEL= option determines the color ramp.
- If a *range-attr-var* that is associated with an attribute map is specified, then the color mapping defined by the associated RANGEATTRMAP statement determines the color ramp.

Tip: To produce "discrete" color mapping, the RANGEATTRMAP statement can define an attribute map that maps a single color to all values greater than 0, and another color to all values less than 0.

Tip: The FILLTYPE= option can be used to indicate whether the color mapping is used to produce solid or gradient fills.

CONNECT=START | END
determines whether trend lines connect to the adjacent bar's starting or ending value.

Default: START

START
trend lines are horizontal and connect to the adjacent bar's starting value. Each connecting line extends from the right corner of one bar's ending value to the left corner of the adjacent bar's starting value.

END
trend lines are diagonal and connect to the adjacent bar's ending value. Each connecting line extends from the right corner of one bar's ending value to the left corner of the adjacent bar's ending value.

Restriction: The last connect line is always drawn horizontally, extending from the right corner of the last data bar's ending value to the left corner of the "final" bar's starting value.

CONNECTATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)
specifies the appearance of the trend lines that connect the bars. See "General Syntax for Attribute Options" on page 849 for the syntax on using a *style-element* and "Line Options" on page 850 for available *line-options*.

Default: The GraphConnectLine style element.

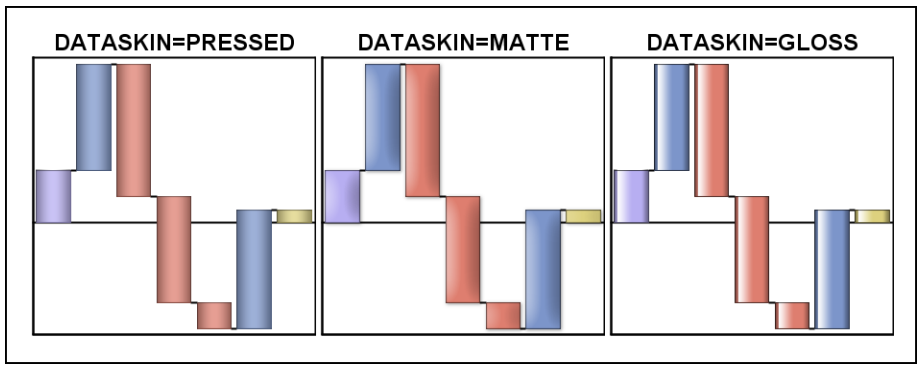
CONNECTDECREASINGATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)
specifies the appearance of trend lines that denote a decreasing value between bars. See "General Syntax for Attribute Options" on page 849 for the syntax on using a *style-element* and "Line Options" on page 850 for available *line-options*.

Default: The appearance specified by the CONNECTATTRS= option.

CONNECTINCREASINGATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)
specifies the appearance of trend lines that denote an increasing value between bars. See "General Syntax for Attribute Options" on page 849 for the syntax on using a *style-element* and "Line Options" on page 850 for available *line-options*.

Default: The appearance specified by the CONNECTATTRS= option.

DATASKIN=NONE | PRESSED | SHEEN | CRISP | GLOSS | MATTE
enhances the visual appearance of the filled bars.



Default: NONE

Requirement: For this option to have any effect, the fill must be enabled by the ODS style or the DISPLAY= option.

Interaction: The skin appearance is based on the FILLATTRS= color.

Interaction: When a data skin is applied, all bar outlines are set by the skin, and the OUTLINEATTRS= option is ignored.

DATATRANSPARENCY=*number*
specifies the degree of the transparency of the bar fill, bar outline, and trend lines, if displayed.

Default: 0

Range: 0 (opaque) to 1 (entirely transparent)

Tip: The FILLATTRS = option can be used to set transparency for just the filled bar area. The INITIALBARATTRS= and FINALBARATTRS= options can be used to specify transparency for the initial and final bars. You can combine this option with FILLATTRS=, INITIALBARATTRS=, and FINALBARATTRS= to set one transparency for the bar outlines and trend lines but a different transparency for the bar fills. Example:

```
datatransparency=0.2 fillattrs=(transparency=0.6)
```

DISPLAY=STANDARD | ALL | (*display-options*)
specifies which bar features to display.

Default: STANDARD

STANDARD
displays outlined, filled bars; connect lines; and the "final" bar.

ALL
same as STANDARD

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

FILL—displays filled bars

FINALBAR—displays the "final" bar

OUTLINE—displays outlined bars

CONNECT—Displays line segments (trend lines) connecting adjacent bar. The connection point is determined by the CONNECT= option.

Tip: To control the appearance of the bars, use the COLORMODEL= , FILLATTRS= , and OUTLINEATTRS= options.

Tip: To control the appearance of the trend lines, use the CONNECTATTRS= , CONNECTDECREASINGATTRS= , and CONNECTINCREASINGATTRS= options.

FILLATTRS=*style-element* | *style-element* (*fill-options*) | (*fill-options*)
specifies the appearance of the filled transaction bars. See "General Syntax for Attribute Options" on page 849 for the syntax on using a *style-element* and "Fill Options" on page 850 for available *fill-options*.

Default:

- If the COLORGROUP= option is not specified, the GraphDataDefault:Color style reference.
- If the COLORGROUP= option is specified, the GraphData1:Color - GraphDataN:Color style references.

Interaction: This option is ignored if either the COLORMODEL= or COLORRESPONSE= option is specified.

Tip: The DATATRANSPARENCY= option sets the transparency for the bar fill, bar outline, and trend lines. You can combine this option with

DATATRANSPARENCY= to set one transparency for the bar outlines and trend lines but a different transparency for the bar fills. Example:

```
datatransparency=0.2 fillattrs=(transparency=0.6)
```

FILLTYPE=SOLID | GRADIENT
determines whether a solid or gradient fill is used in the bars.

Default: SOLID

Tip: The colors that are used depend on whether the COLORGROUP= option or the COLORRESPONSE= option is also specified.

FINALBARATTRS=*style-element* | *style-element* (*fill-options*) | (*fill-options*)
specifies the appearance of the "final" bar, if displayed. See "General Syntax for Attribute Options" on page 849 for the syntax on using a *style-element* and "Fill Options" on page 850 for available *fill-options*.

Default: the GraphFinal style element

Interaction: This option is ignored if the DISPLAY= option does not display the "final" bar.

FINALBARTICKVALUE= "*string*"
specifies a tick value to use on the category axis when the "final" bar is displayed

Default:"Final"

Interaction: This option is ignored if the DISPLAY= option does not display the "final" bar.

INITIALBARATTRS=*style-element* | *style-element* (*fill-options*) | (*fill-options*)
specifies the appearance of the "initial" bar, if displayed. See "General Syntax for Attribute Options" on page 849 for the syntax on using a *style-element* and "Fill Options" on page 850 for available *fill-options*.

Default: the GraphInitial style element

Interaction: For this option to take effect, the INITIALBARVALUE= option must also be specified.

INITIALBARTICKVALUE= "*string*"
specifies a tick value to use on the category axis when the "initial" bar is displayed.

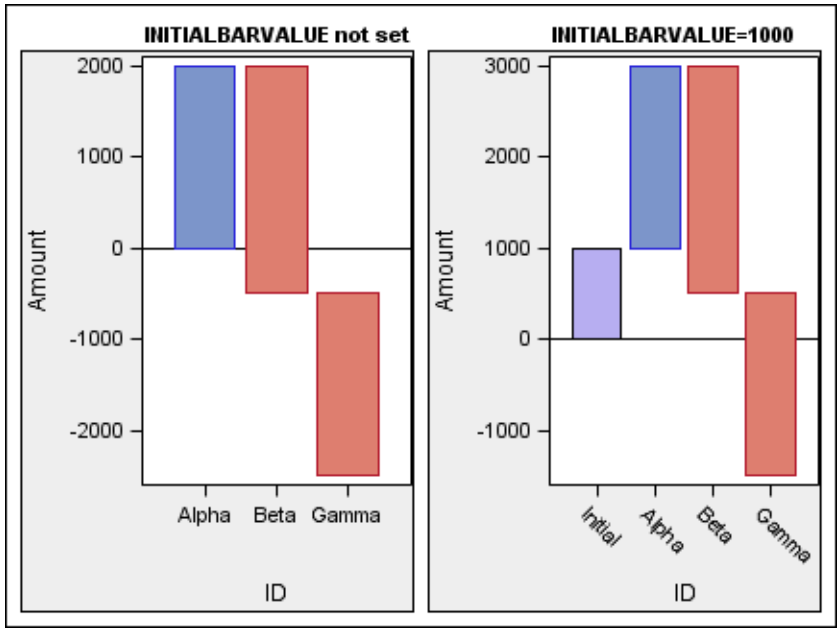
Default: "Initial"

Interaction: For this option to take effect, the INITIALBARVALUE= option must also be specified.

INITIALBARVALUE=*number*
specifies a value for the "initial" bar. The initial bar's value is used as the starting response value for the first transaction bar.

Default: no default. The first transaction bar starts at response value 0.

Interaction: If this option is not specified, the initial bar is not included in the chart and the first transaction bar is drawn from response value 0. This is true even if an intercept value is set by the BASELINEINTERCEPT= option.



LEGENDLABEL= "*string*"
specifies a label for the legend item that is associated with this plot.

Default: The *string* specified on the NAME= option.

Restriction: This option applies only to an associated DISCRETELEGEND statement.

Interaction: If the COLORGROUP= 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.

OUTLINEATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)
specifies the appearance of the bar outlines. See "General Syntax for Attribute Options" on page 849 for the syntax on using a *style-element* and "Line Options" on page 850 for available *line-options*.

Default: The ContrastColor and LineThickness attributes of the GraphOutlines style element.

Interaction: For this option to have any effect, outlines must be enabled by the ODS style or the DISPLAY= option.

Interaction: If the DATASKIN= option applies a data skin, this option is ignored.

ROLENAME=(*role-name-list*)
specifies user-defined roles that can be used to display information in the tooltips. Columns for the tooltip display are specified in the TIP= option.

Default: no user-defined roles

(*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 CATEGORY , RESPONSE , COLORGROUP , and COLORRESPONSE .

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.

STAT= SUM | MEAN
specifies the statistic to be computed for the RESPONSE axis.

Default:SUM

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 waterfall chart can be specified along with roles that do.

Default: The columns assigned to these roles are automatically included in the tooltip information: CATEGORY and RESPONSE .

(*role-list*)
an ordered, blank-separated list of unique WATERFALLCHART and user-defined roles. WATERFALLCHART roles include CATEGORY , RESPONSE , COLORGROUP , and COLORRESPONSE .

The following example displays tooltips only for the column that is assigned to the RESPONSE role:

```
TIP= (RESPONSE)
```

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 the information that is defined by the tooltip roles.

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=(RESPONSE)
TIPFORMAT=(RESPONSE=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 the information that is defined by the tooltip roles.

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=(RESPONSE)
TIPLABEL=(RESPONSE="Average Amount")
```

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 a 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 RESPONSE value. If they are not, the results might be unpredictable. The URL value can be blank for some RESPONSE values, meaning that no action is taken when the bars for those RESPONSE values are selected. The URL value can be the same for different RESPONSE values, meaning that the same action is taken when the bars for those RESPONSE values are selected.

For grouped data, the values of the column are expected to be the same for each unique RESPONSE 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 557.

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 RESPONSE= 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 557.

Part 5

Plot Axes

Chapter 50

Axis Features in Layouts

Overview

All 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 that 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 following criteria:

- 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 or columns of data that contribute to defining the axis range
- 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

Overview for Axis Mapping

Depending on the layout type and the plots that you specify within the layout, you can manage up to four axes for two-dimensional 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)

Within single-cell layouts (for example, OVERLAY layout), there can be just one each of these four axes. However, within multi-cell layouts (for example, LATTICE layouts), each cell can display the four axes. Thus, there can be multiple X, X2, Y, and Y2 axes across the columns and rows in the layout grid. In a lattice-type layout, you might have to use layout options to specify how the data ranges and axis display should be managed. This section discusses the simpler case for axis mapping, and "Axis Mapping in Lattice-type Layouts" on page 559 discusses the case for lattice-type layouts.

Note: GRIDDED layouts can be used to create a grid of cells, but the cells are independent. Thus, axes in the grid cannot be managed across columns and rows, so the simpler case applies to GRIDDED layouts.

Primary and Secondary Axes

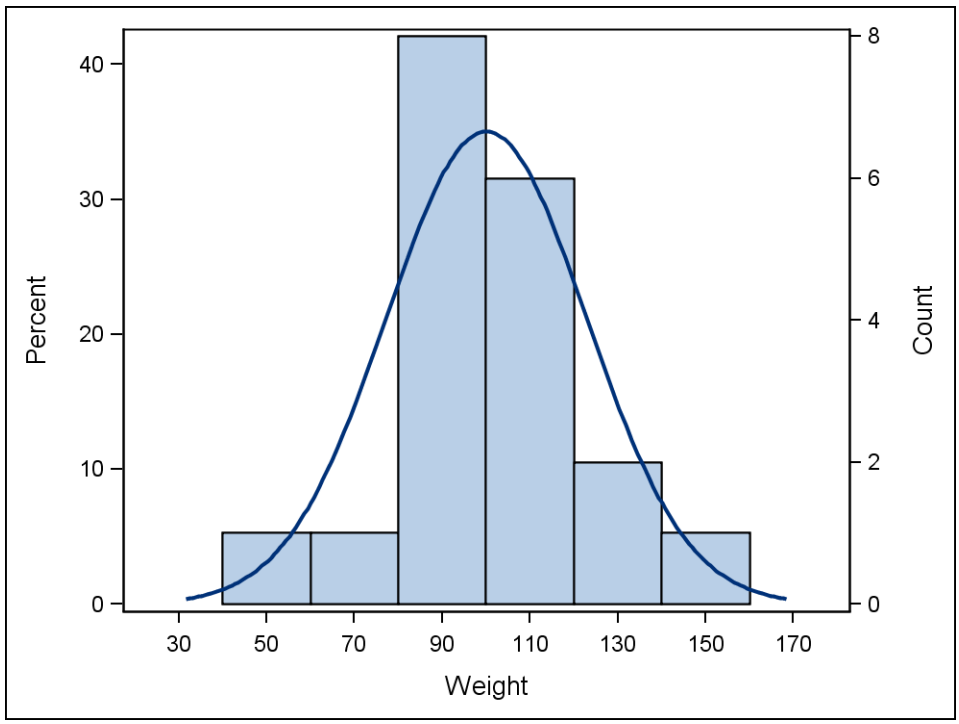
By default, plot data are mapped to the primary axes. To enable you to override the default, plot statements that support a secondary X2 axis provide an XAXIS= option that can map data to the X or X2 axis. Plot statements that support a Y axis provide a YAXIS= option that can map data to the Y or Y2 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 maps 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.

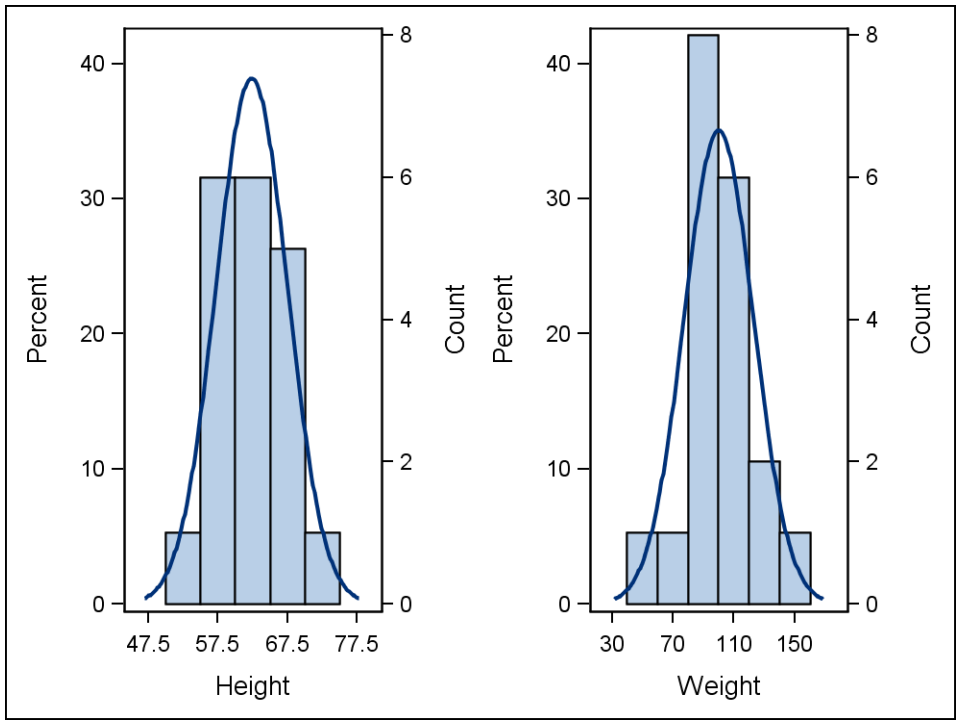
When establishing axis features for each axis, the GTL determines which plot specifications map data to the axis. The GTL also collects the data for all of the plots that must be represented and maps that data to the designated axis. "When Plots Share Data and a Common Axis" on page 562 discusses the criteria the GTL uses to determine the axis features for the axes after this mapping has been done for each axis.

Axis Mapping in Lattice-type Layouts

Lattice-type layouts (LAYOUT LATTICE, LAYOUT DATALATTICE, and LAYOUT DATAPANEL) present a grid of graphs that automatically aligns plot areas and tick display areas across grid cells. This alignment facilitates data comparisons among graphs, and for those comparisons to be meaningful, the graph axes must be coordinated across the columns and rows in the grid. All of the principles discussed in "Overview for Axis Mapping" on page 558 apply to the lattice-type layouts. In addition, because there can be multiple X, X2, Y, and Y2 axes across grid cells, you might have to use layout options to specify how the data ranges and axis display should be managed.

For example, the following template uses a LAYOUT LATTICE to generate a grid that displays a height analysis next to a weight analysis. By default in a LAYOUT LATTICE statement, the options ROWDATARANGE= and ROW2DATARANGE= are set to DATA. The DATA setting scales the Y-axis and Y2-axis data ranges separately for each cell in the layout. To ensure that the Y-axis data range is the same in both cells, the example specifies ROWDATARANGE=UNION. Similarly, to ensure that the Y2-axis data range is the same in both cells, the example specifies ROW2DATARANGE=UNION:

```
proc template;
  define statgraph y2axis;
    begingraph;
      layout lattice / columns=2 columngutter=10
          rowdatarange=union row2datarange=union ;
        layout overlay;
          histogram height / scale=count yaxis=y2;
          histogram height / scale=percent yaxis=y;
          densityplot height / normal();
        endlayout;
        layout overlay;
          histogram weight / scale=count yaxis=y2;
          histogram weight / scale=percent yaxis=y;
          densityplot weight / normal();
        endlayout;
      endlayout;
    endgraph;
  end;

proc sgrender data=sashelp.class template=y2axis;
run;
```



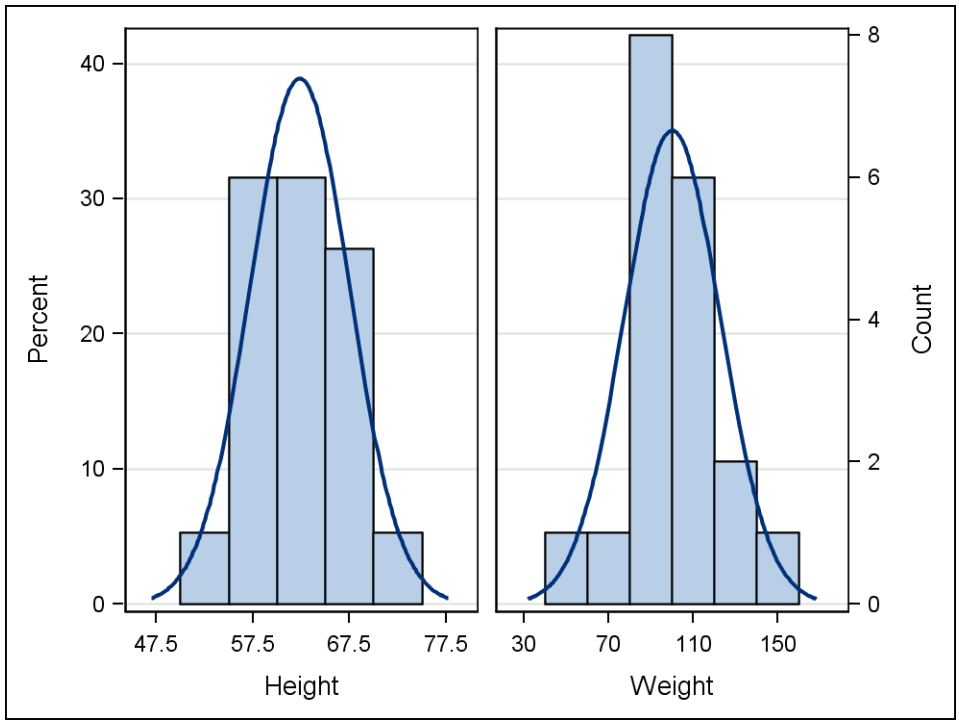
By default in a LAYOUT LATTICE statement, the options COLUMNDATARANGE= and COLUMN2DATARANGE= are also set to DATA. But in this analysis, the height is

a separate measure from the weight, so the separate scales are appropriate for the X-axes across cells. If the X-axes were displaying the same measure (for example, comparing the height of female subjects to the height of male subjects), you could specify COLUMNDATARANGE=UNIONALL. This would set the same scaling to the X-axis data ranges across the two layout columns. In this example you would not bother changing the default COLUMN2DATARANGE= setting because the X2 axis is not needed.

Note: For DATALATTICE and DATAPANEL statements, UNIONALL is the default value for the data ranges. Thus, you would not have to change the data ranges unless you wanted to set UNION to scale data ranges per row or per column in the layout.

In the example, scaling the data ranges across the row ensures proper axis scaling. However, the graph display is cluttered by the duplicate display of ticks, axis values, and axis labels on both the Y and Y2 axes. To simplify the display, you can consolidate the axes. To do so, use a ROWAXES block to display a single Y axis for both cells, and a ROW2AXES block to display a single Y2 axis for both cells. The consolidated view removes the internal axes from the grid and displays only the external axes:

```
proc template;
  define statgraph y2axis;
    begingraph;
      layout lattice / columns=2 columngutter=10
          rowdatarange=union row2datarange=union;
        rowaxes;
          rowaxis / griddisplay=on;
        endrowaxes;
        row2axes;
          rowaxis;
        endrow2axes;
        layout overlay;
          histogram height / scale=count yaxis=y2;
          histogram height / scale=percent yaxis=y;
          densityplot height / normal();
        endlayout;
        layout overlay;
          histogram weight / scale=count yaxis=y2;
          histogram weight / scale=percent yaxis=y;
          densityplot weight / normal();
        endlayout;
      endlayout;
    endgraph;
  end;

proc sgrender data=sashelp.class template=y2axis;
run;
```



When using ROWAXES or ROW2AXES blocks in a LATTICE layout, you nest within the block one ROWAXIS statement for each row in the layout grid. The ROWAXIS statements are applied sequentially to the rows, and each ROWAXIS statement specifies the axis options for the Y or Y2 axes in its corresponding row. ROWAXIS statements within the ROWAXES block apply to the Y axes, and ROWAXIS statements within the ROW2AXES block apply to the Y2 axes. This example has just a single row in the grid, so each block specifies only one ROWAXIS statement. Notice that the ROWAXIS statement in the ROW2AXES block does not use any options. Thus, it consolidates Y2 axes in the row into a single, external Y2 axis, but it does not alter the default features of that axis. For columns in the grid, the LATTICE layout provides COLUMNAXES and COLUMN2AXES blocks. These blocks use COLUMNAXIS statements to externalize X and X2 axes and specify their features.

When you use DATALATTICE and DATAPANEL layouts, the layout dynamically generates a grid that contains as many cells as can be produced from the combination of classification values. In those layouts the axes are always external, and you can use the COLUMNAXISOPTS=, COLUMN2AXISOPTS=, ROWAXISOPTS=, and ROW2AXISOPTS= options to specify the features for the axes. The settings on each option apply across the entire grid. For example, if you specify the ROWAXISOPTS= option in a DATALATTICE layout, the specified settings apply to the external Y axes in every row.

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 "Plot Data Are Mapped to a Designated Axis" on page 558, 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. Mapping to separate axes can avoid 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 layouts and plots 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 first plot that generates an axis is considered primary on a per-axis basis.
- If PRIMARY=TRUE for a plot within an overlay-type layout, that plot's data columns, data type, and plot 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.

Note: The SAS format on the primary plot's column determines the axis format, although the axis might not use that SAS format "as-is" from the column.

If a SCATTERPLOT's X= argument specifies a column that has a SAS DATETIME format, the default X-axis type is TIME. 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 563 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.

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 any or all of the options COLUMNDATARANGE=, COLUMN2DATARANGE=, ROWDATARANGE=, or ROW2DATARANGE= use the DATA setting to scale axis data ranges separately for each cell in the layout, then the layout cells are data-independent. The data-independent cells do not interact with each other for determining the axis features in the display.

Axes are shared in the layout when one of the options COLUMNDATARANGE=, COLUMN2DATARANGE=, ROWDATARANGE=, or ROW2DATARANGE= 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. When UNIONALL is in effect, those same features are used in all of the grid's layout cells. When UNION is in effect, those same features are used on a per-row or per-column basis. 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.

For an example LATTICE layout with external axes, see "Axis Mapping in Lattice-type Layouts" on page 559.

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.

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 562 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 cannot share axes because the plot types 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	Statement
Two-dimensional axis options	LAYOUT OVERLAY statement
Three-dimensional axis options	LAYOUT OVERLAY3D statement
Equated axis options	LAYOUT OVERLAYEQUATED statement
Lattice axis options	LAYOUT LATTICE statement
DataLattice and 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 plots that you are defining. For every plot in the template language, the documentation indicates what axis types it supports. Plots statements that are specified in the template are ignored if they are incompatible with the axis type.

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 bar chart 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.

Three-dimensional 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

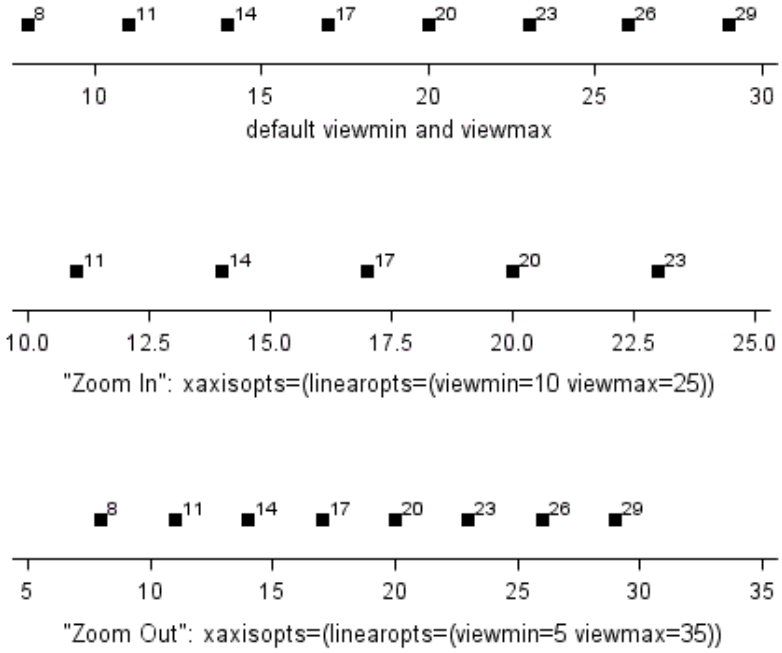
The VIEWMIN= and VIEWMAX= axis options can be used to adjust the view of an axis. You can specify minimum data values to include in the display, maximum data values, or both (the specified values might be adjusted by the threshold calculation). By

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.



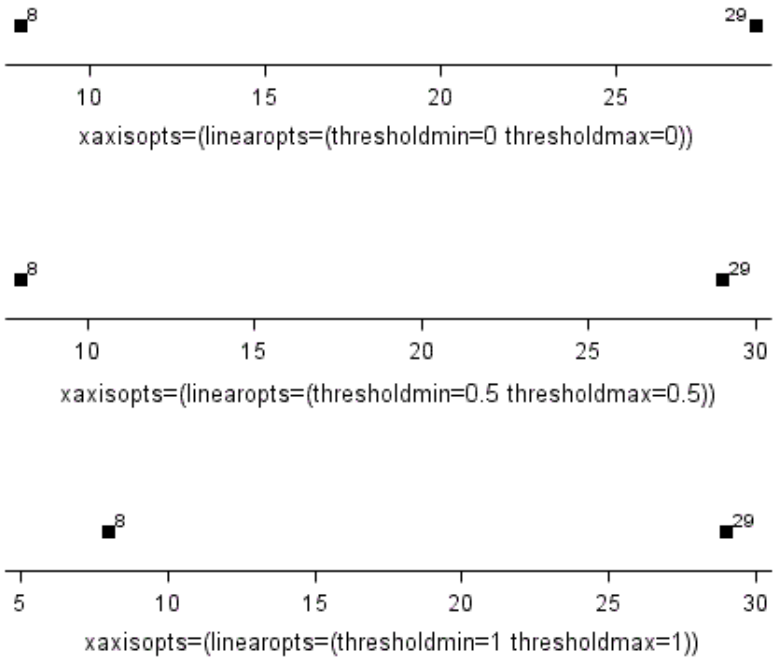
Adjusting Axis Thresholds

On a continuous, linear axis, 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 to VIEWMAX range). 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.

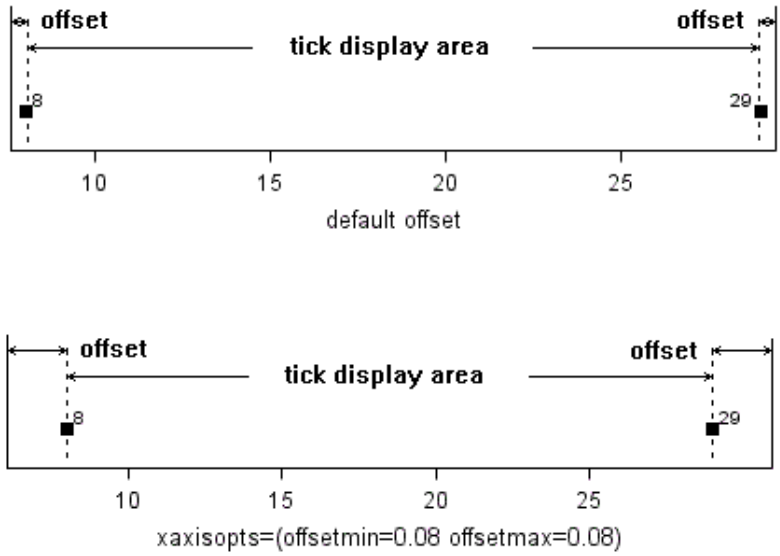


Adjusting Axis Offsets

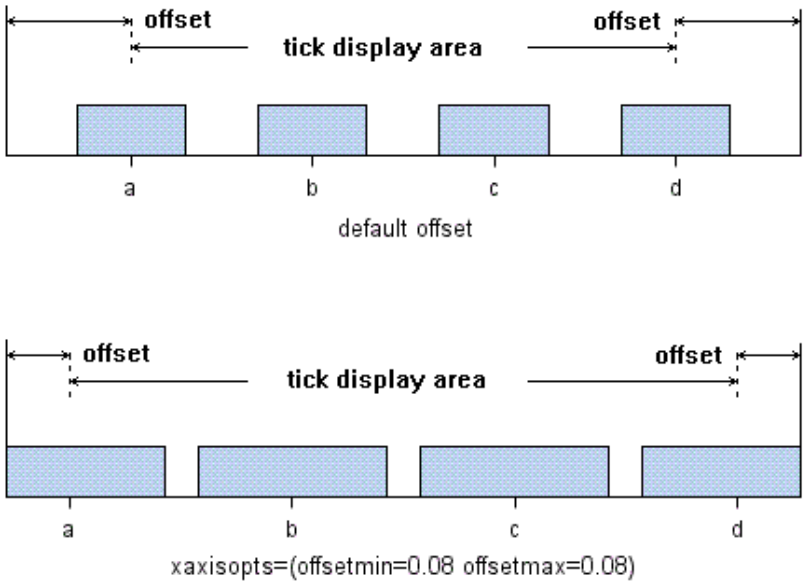
The OFFSETMIN = and OFFSETMAX = axis options can be used to reserve an area at the minimum end of an axis, the maximum end, or both ends. No tick marks are displayed in the reserved areas.

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 51

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
DISCRETEOPTS	Specifies options for a discrete 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
NAME	Assigns a name to an axis for reference in other statements.
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 axis. See "Options for Discrete Axes" on page 578.

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.

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

Interaction: A secondary axis is not an independent axis. Rather, it mirrors the primary axis. Thus, for this option to take effect, all plot statements in the layout must map data to the same primary axis. For example, a secondary X2 axis can be displayed on top in the layout, provided all plot statements set XAXIS=X to map data to the primary X axis (bottom). Similarly, a secondary Y2 axis can be displayed to the right in the layout, provided all plot statements set YAXIS=Y to map data to the primary Y axis (left). If some plot statements set XAXIS=X and others set XAXIS=X2, both the X and X2 axis are primary and a secondary X axis cannot be displayed. In that case, this option is ignored. The same applies for the Y 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 849 for the syntax on using a *style-element* and "Line Options" on page 850 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 562.

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 849 for the syntax on using a *style-element* and "Text Options" on page 852 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 580 .

LOGOPTS=(*log-axis-options*)
specifies one or more options for a log axis. See "Options for Log Axes Only" on page 586.

NAME="*string*"
assigns a name to an axis for reference in other statements. Currently, it is used only in an AXISLEGEND statement.

Default: no default

Interaction: This option is ignored unless the axis is discrete. The axis can be discrete by default, or explicitly set to discrete with a TYPE= DISCRETE setting.

Interaction: For this option to take effect, an axis legend must be enabled. To enable an axis legend, the DISCRETEOPTS= option must set the TICKVALUEFITPOLICY to either EXTRACT or EXTRACTALWAYS. In addition, an AXISLEGEND statement must be specified to generate the axis legend.

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 568.

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 568.

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

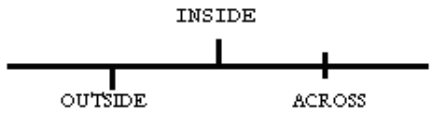
SHORTLABEL=*"string"*
specifies an alternate axis label. This label is used when the default axis label or label specified by the LABEL= option is too long for the axis length.

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 849 for the syntax on using a *style-element* and "Text Options" on page 852 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 options for a time axis. See "Options for Time Axes Only" on page 588.

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 562.

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 565.

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 Axes

DISCRETEOPTS=(*discrete-axis-options*)
specifies one or more options for a discrete 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.

Discrete Axis Options	Description
COLORBANDS	Specifies the display of alternating wall-color bands corresponding to the discrete axis bins
COLORBANDSATTRS	Specifies the appearance of the alternating wall-color bands
TICKTYPE	Specifies the position of the axis tick mark.
TICKVALUEFITPOLICY	Specifies a policy for avoiding tick value collision on an X or X2 axis

COLORBANDS= NONE | EVEN | ODD
[discrete axis]
specifies the display of alternating wall-color bands corresponding to the discrete axis bins.

Default: NONE

Interaction: Specifying this option for more than one axis in the layout might have unexpected results. The order in which color bands are drawn might not match the order in which the axis options are specified.

Tip: Borders for the color bands can be added by setting TICKTYPE= INBETWEEN in the DISCRETEOPTS= option, and by setting GRIDDISPLAY= ON.

Tip: Because alternating color bands are drawn on top of the plot wall, this option can be coordinated with the LAYOUT OVERLAY statement's WALLCOLOR= option.

COLORBANDSATTRS=*style-element* | *style-element* (*fill-options*) | (*fill-options*)
[discrete axis]
specifies the appearance of the alternating wall-color bands. For the alternating colors, one set uses the WALLCOLOR= colors that are set in the LAYOUT OVERLAY statement, and the other set uses the colors set on this option. See "General Syntax for Attribute Options" on page 849 for the syntax on using a *style-element* and "Fill Options" on page 850 for available *fill-options*.

Default: The GraphBlock style element.

TICKTYPE= MIDPOINT | INBETWEEN
[discrete axis]
specifies the position of the axis tick mark.

Default: MIDPOINT

MIDPOINT	The tick mark is placed at the midpoint value location.
INBETWEEN	The tick mark is placed half way between adjacent midpoint locations.

TICKVALUEFITPOLICY=*policy*
[discrete axis]
specifies a policy for avoiding tick value collision on an axis.

Note: A Y or Y2 axis supports only the values NONE | THIN | EXTRACT | EXTRACTALWAYS

Default:

- ROTATE for an X or X2 axis
- NONE for a Y or Y2 axis

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. This value is not available for a Y or Y2 axis.
NONE	Do not attempt to fit ticks that collide.
ROTATETHIN	Attempt ROTATE policy and then THIN policy. This value is not available for a Y or Y2 axis.

STAGGER	Tick values alternate between two rows. This value is not available for a Y or Y2 axis.
STAGGERROTATE	Attempt STAGGER policy and then ROTATE policy. This value is not available for a Y or Y2 axis.
STAGGERTHIN	Attempt STAGGER policy and then THIN policy. This value is not available for a Y or Y2 axis.
STAGGERTRUNCATE	Attempt STAGGER policy and then TRUNCATE policy. This value is not available for a Y or Y2 axis.
TRUNCATE	Tick values are shortened when they exceed a certain number of characters. This value is not available for a Y or Y2 axis.
TRUNCATEROTATE	Attempt TRUNCATE policy and then ROTATE policy. This value is not available for a Y or Y2 axis.
TRUNCATESTAGGER	Attempt TRUNCATE policy and then STAGGER policy. This value is not available for a Y or Y2 axis.
TRUNCATETHIN	Attempt TRUNCATE policy and then THIN policy. This value is not available for a Y or Y2 axis.
THIN	Some tick values are removed.
EXTRACT	Instead of actual tick values, display consecutive integers along the axis to represent those tick values. (See the Requirement for additional information.) In most cases, this policy is implemented if the system estimates that a collision might occur. If no collision occurs, the actual tick values are displayed on the axis in the normal manner.
EXTRACTALWAYS	Same as EXTRACT, except that the extraction is implemented regardless of whether collision occurs.

Requirement: The EXTRACT and EXTRACTALWAYS policies must be used in conjunction with an AXISLEGEND statement. The AXISLEGEND statement creates a legend that correlates the actual tick values with the consecutive integers that are displayed as axis-tick values in the graph. The AXISLEGEND statement must reference this axis, using the name that is assigned to the axis in its NAME= axis option.

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.

Linear Axis Option	Description
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*
[linear axis]
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*
[linear axis]
specifies that the axis perpendicular to the current axis be drawn at the indicated data value.

Default: The axis perpendicular to the current axis is drawn at the minimum tick value minus the OFFSETMIN= value.

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*
[linear axis]
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 567.

Default: .30

Range: 0 - 1

Interaction: This option is ignored if the TICKVALUELIST= or TICKVALUESEQUENCE= option is used.

THRESHOLDMIN= *number*
[linear axis]
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 567.

Default: .30

Range: 0 - 1

Interaction: This option is ignored if the TICKVALUELIST= or TICKVALUESEQUENCE= option is used.

TICKVALUEFITPOLICY=THIN | *policy*
[linear axis]
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. This value is not available for a Y or Y2 axis.

ROTATETHIN
Attempt ROTATE policy and then THIN policy. This value is not available for a Y or Y2 axis.

STAGGER
Tick values alternate between two rows. This value is not available for a Y or Y2 axis.

STAGGERROTATE
Attempt STAGGER policy and then ROTATE policy. This value is not available for a Y or Y2 axis.

STAGGERTHIN
Attempt STAGGER policy and then THIN policy. This value is not available for a Y or Y2 axis.

TICKVALUEFORMAT=(*format-options*) | DATA | *format*
[linear axis]
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 a scale factor from the tick values and use it to reduce the tick value width. For large tick values, the scale factor is set to ensure that the absolute value of the largest value is greater than 1. The scale can be millions, billions, or trillions for values of 999 trillion or less, or a multiple of 10 (denoted as 10^*n*) for values over 999 trillion. For small fractional tick values, the scale factor is set to ensure that the absolute value of the smallest value is greater than 1. The scale can be millionth, billionth, or trillionth for values of 1 trillionth or more, or a multiple of 1/10 (10^–*n*) for values less than 1 trillionth. The scale used is appended to the axis label. For example: Total Sales (millions).

Restriction: The scale that is extracted by the EXTRACTSCALE= option is derived from the English locale.

Note: When EXTRACTSCALE=TRUE and a scale is extracted, the tick values are formatted to provide the best fit on the axis. In that case, the tick value format might differ from the data format even when a named format is applied to the data values.

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: GTL currently honors most but not every SAS format. For details, see "SAS Formats Not Supported" on page 855.

Interaction: If the axis label is not displayed, the EXTRACTSCALE=TRUE option is ignored.

TICKVALUELIST=(*numeric-list*)
[linear axis]
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*
[linear axis]
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.

Interaction: This option is ignored if the DISPLAY= option does not display tick marks.

TICKVALUESEQUENCE=(*sequence-options*)
[linear axis]
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*
[linear axis]
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 566.

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*
[linear axis]
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 566.

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 and each pair is space separated.

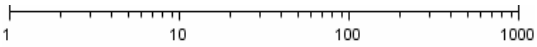
Interaction: This option is ignored if the axis type is not LOG.

Log Axis Option	Description
BASE	Specifies the base of the logarithmic scale for the axis values.
MINORTICKS	Specifies whether minor ticks are displayed.
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.
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
[log axis]
specifies the base of the logarithmic scale for the axis values.

Default: 10

MINORTICKS=*boolean*
[log axis]
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.

THRESHOLDMAX= *number*
[log axis]
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 567.

Default: .30

Range: 0 - 1

THRESHOLDMIN= *number*
[log axis]
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 567.

Default: .30

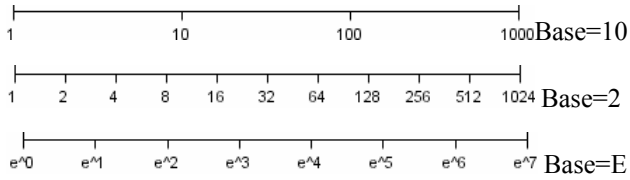
Range: 0 - 1

TICKINTERVALSTYLE=AUTO | LOGEXPAND | LOGEXPONENT | LINEAR
[log axis]
specifies how to scale and format the values for major tick marks.

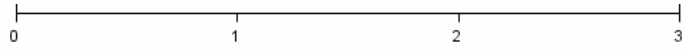
Default: AUTO

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 855.

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*
[log axis]
specifies the maximum data value to include in the display. For more information, see "Adjusting the Axis View" on page 566.

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*
[log axis]
specifies the minimum data value to include in the display. For more information, see "Adjusting the Axis View" on page 566.

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 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.
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.

Time Axis Option	Description
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*
[time axis]
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.

INTERVAL	Unit	Tick interval	Default tick value format
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.

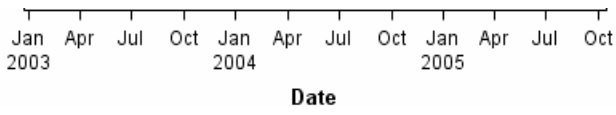
MINORTICKS=*boolean*
[time axis]
specifies whether minor ticks are displayed.

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*
[time axis]
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 | *policy*
[time axis]
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
[time axis]
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 855.

Interaction: If this option is specified, the SPLITTICKVALUE= option is ignored.

TICKVALUELIST=(*time-constant-list* | *date-constant-list* | *datetime-constant-list* | *numeric-list*)
[time axis]
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).

Interaction: If this option is specified, the SPLITTICKVALUE= and INTERVAL= options are ignored.

TICKVALUEPRIORITY=*boolean*
[time axis]
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= and VIEWMAX= options are ignored.

Interaction: This option is ignored if the DISPLAY= option does not display tick marks.

VIEWMAX= *number*
[time axis]
specifies the maximum data value to include in the display. For more information, see "Adjusting the Axis View" on page 566.

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*
[time axis]
specifies the minimum data value to include in the display. For more information, see "Adjusting the Axis View" on page 566.

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 52

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 849 for the syntax on using a *style-element* and "Line Options" on page 850 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 562.

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 849 for the syntax on using a *style-element* and "Text Options" on page 852 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 598.

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 568.

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 568.

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 849 for the syntax on using a *style-element* and "Text Options" on page 852 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 options for a time axis. See "Options for Time Axes Only" on page 601.

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 565.

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*
[linear axis]
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*
[linear axis]
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 567.

Default: .30

Range: 0 - 1

Interaction: This option is ignored if the TICKVALUELIST= or TICKVALUESEQUENCE= option is used.

THRESHOLDMIN= *number*
[linear axis]
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 567.

Default: .30

Range: 0 - 1

Interaction: This option is ignored if the TICKVALUELIST= or TICKVALUESEQUENCE= option is used.

TICKVALUEFORMAT=(*format-options*) | DATA | *format*
[linear axis]
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 a scale factor from the tick values and use it to reduce the tick value width. For large tick values, the scale factor is set to ensure that the absolute value of the largest value is greater than 1. The scale can be millions, billions, or trillions for values of 999 trillion or less, or a multiple of 10 (denoted as 10^*n*) for values over 999 trillion. For small fractional tick values, the scale factor is set to ensure that the absolute value of the smallest value is greater than 1. The scale can be millionth, billionth, or trillionth for values of 1 trillionth or more, or a multiple of 1/10 (10^–*n*) for values less than 1 trillionth. The scale used is appended to the axis label. For example: Total Sales (millions).

Restriction: The scale that is extracted by the EXTRACTSCALE= option is derived from the English locale.

Note: When EXTRACTSCALE=TRUE and a scale is extracted, the tick values are formatted to provide the best fit on the axis. In that case, the tick value format might differ from the data format even when a named format is applied to the data values.

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: GTL currently honors most but not every SAS format. For details, see "SAS Formats Not Supported" on page 855.

Interaction: If the axis label is not displayed, the EXTRACTSCALE=TRUE option is ignored.

TICKVALUELIST=(*numeric-list*)
[linear axis]
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.

See also: TICKVALUESEQUENCE= option as an alternative for customizing tick marks.

TICKVALUESEQUENCE=(*sequence-options*)
[linear axis]
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.

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*
[time axis]
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
[time axis]
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
[time axis]
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 855.

TICKVALUELIST=(*time-constant-list* | *date-constant-list* | *datetime-constant-list* | *numeric-list*)
[time axis]
: 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).

Interaction: If this option is used, the INTERVAL= option is ignored.

Chapter 53

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:

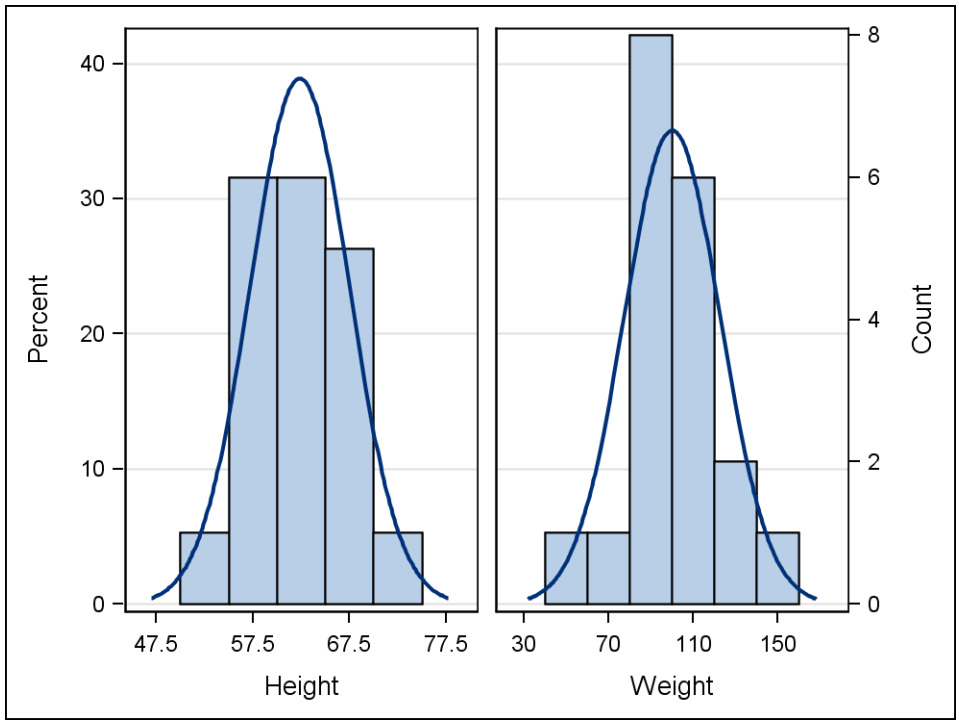
COLUMNAXIS /*external-axis-options*

ROWAXIS /*external-axis-options*

Example Program and Statement Details

Example Graph

The following graph was generated by the "Example Program" on page 606 :



Example Program

This example shows how to externalize axes in a LAYOUT LATTICE and manage the axis features on primary Y and Y2 axes.

- The first HISTOGRAM statement specifies YAXIS=Y2 to make the Y2 axis the primary axis for COUNT measures. The second HISTOGRAM statement specifies YAXIS=Y to make the Y axis the primary axis for PERCENT measures.
- In order to externalize the axes within the layout, the data ranges for the axes must be unified. In the LAYOUT LATTICE statement, the ROWDATARANGE= option unifies the data ranges for the Y axes across the row. The ROW2DATARANGE= option unifies the data ranges for the Y2 axes across the row.
- The ROWAXIS statement is used to manage axis features for the row axes. To manage the primary Y axis, a ROWAXIS statement is specified within a ROWAXES block. To manage the primary Y2 axis, another ROWAXIS statement is specified within a ROW2AXES block.
- Within the ROWAXES block, the ROWAXIS statement consolidates Y axes in the row into a single, external Y axis and also displays grid lines. Within the ROW2AXES block, the ROWAXIS statement consolidates Y2 axes in the row into a single, external Y2 axis, but it does not alter the default features of that axis.

```
proc template;
  define statgraph y2axis;
    begingraph;
      layout lattice / columns=2 columngutter=10
          rowdatarange=union row2datarange=union ;
        rowaxes;
          rowaxis / griddisplay=on;
        endrowaxes;
        row2axes;
          rowaxis;
```

```
        endrow2axes;
        layout overlay;
          histogram height / scale=count yaxis=y2 ;
          histogram height / scale=percent yaxis=y ;
          densityplot height / normal();
        endlayout;
        layout overlay;
          histogram weight / scale=count yaxis=y2 ;
          histogram weight / scale=percent yaxis=y ;
          densityplot weight / normal();
        endlayout;
      endlayout;
    endgraph;
  end;

proc sgrender data=sashelp.class template=y2axis;
run;
```

Statement Summary

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 or COLUMN2AXES block to externalize column axes for the layout. Similarly, ROWAXIS statements are used within a ROWAXES or ROW2AXES block to externalize row axes for the layout. Each axis block is used to manage the primary axis. The axis that is considered primary depends on the settings for the XAXIS= and YAXIS= options in plot statements that are specified within the layout:

Option Setting	Primary Axis	Axis Block to Use
XAXIS=X	X (bottom)	COLUMNAXES
XAXIS=X2	X2 (top)	COLUMN2AXES
YAXIS=Y	Y (left)	ROWAXES
YAXIS=Y2	Y2 (right)	ROW2AXES

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 the LAYOUT LATTICE statement's COLUMNDATARANGE=, COLUMN2DATARANGE=, ROWDATARANGE=, and ROW2DATARANGE= options.
- Within a COLUMNAXES or COLUMN2AXES block, one COLUMNAXIS statement should be specified for each column that contains axes that you need to manage. Both axes blocks can contain a COLUMNAXIS statement for the same column. For example, to manage the axes in the first column of the layout, the COLUMNAXES block can contain a COLUMNAXIS statement that manages the column's X axes. The COLUMN2AXES block can contain a COLUMNAXIS statement that manages the column's X2 axes.

- Within a ROWAXES or ROW2AXES block, one ROWAXIS statement should be specified for each row that contains axes that you need to manage. Both axes blocks can contain a ROWAXIS statement for the same row. For example, to manage the axes in the first row of the layout, the ROWAXES block can contain a ROWAXIS statement that manages the row's Y axes. The ROW2AXES block can contain a ROWAXIS statement that manages the column's Y2 axes.

In addition to managing the primary axes, you can also display "secondary" axes in the grid. A secondary axis is not an independent axis. Rather, it mirrors the primary axis, but it is displayed on the opposite side and can have different display options. For example, when the X axis (bottom) is primary, you can mirror that axis with a secondary X axis at the top of the grid. Similarly, when the Y2 axis (right) is primary, you can mirror that axis with a secondary Y2 axis on the left of the grid. A secondary axis makes it easier to interpolate values in the cells that are farthest away from the primary axis. To display a secondary axis, use the DISPLAYSECONDARY= option.

For general information about managing primary and secondary axes, see "Plot Data Are Mapped to a Designated Axis" on page 558 . For details about managing the axes within a LAYOUT LATTICE, see the discussion for a LATTICE's "Axis Statements" on page 89 .

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:

- 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 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.
NAME	Assigns a name to an axis for reference in other statements.
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 axis origin should be reversed.
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.

Statement Option	Description
TYPE	Specifies the type of axis to use.

DISCRETEOPTS=(*discrete-axis-options*)
specifies one or more options for a discrete axis. See "Options for Discrete Axes Only" on page 614.

DISPLAY=STANDARD | ALL | NONE | (*display-options*)
controls which axis features are displayed on the primary axis. For more information about the primary and secondary axes, see the "Statement Summary" on page 607 .

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. For more information about the primary and secondary axes, see the "Statement Summary" on page 607 .

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

Restriction: A secondary axis is not an independent axis. Rather, it mirrors the primary axis (though it can use different display features). Thus, for this option to take effect, all plot statements in the layout must map data to the same primary axis. For example, a secondary X2 axis can be displayed on top in the layout, provided all plot statements set XAXIS=X to map data to the primary X axis (bottom). Similarly, a secondary Y2 axis can be displayed to the right in the layout, provided all plot statements set YAXIS=Y to map data to the primary Y axis (left). If some plot statements set XAXIS=X and others set XAXIS=X2, both the X and X2 axis are primary and a secondary X axis cannot be displayed. In that case, this option is ignored. The same applies for the Y 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 849 for the syntax on using a *style-element* and "Line Options" on page 850 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 562.

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 849 for the syntax on using a *style-element* and "Text Options" on page 852 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 616.

LOGOPTS=(*log-axis-options*)
specifies one or more options for a log axis. See "Options for Log Axes Only" on page 620.

NAME="*string*"
assigns a name to an axis for reference in other statements. Currently, it is used only in an AXISLEGEND statement.

Default: no default

Interaction: This option is ignored unless the axis is discrete. The axis can be discrete by default, or explicitly set to discrete with a TYPE= DISCRETE setting.

Interaction: For this option to take effect, an axis legend must be enabled. To enable an axis legend, the DISCRETEOPTS= option must set the

TICKVALUEFITPOLICY to either EXTRACT or EXTRACTALWAYS. In addition, an AXISLEGEND statement must be specified to generate the axis legend.

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 568.

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 568.

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 the axis origin should be reversed.

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= 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 849 for the syntax on using a *style-element* and "Text Options" on page 852 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 options for a time axis. See "Options for Time Axes Only" on page 622.

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 562.

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.

Options for Discrete Axes Only

DISCRETEOPTS=(*discrete-axis-options*)
specifies one or more options for a discrete. 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.

TICKTYPE= MIDPOINT | INBETWEEN
[discrete axis]
specifies the position of the axis tick mark.

Default: MIDPOINT

MIDPOINT	The tick mark is placed at the midpoint value location.
INBETWEEN	The tick mark is placed half way between adjacent midpoint locations.

TICKVALUEFITPOLICY=ROTATE | *policy*
[discrete axis]
specifies a policy for avoiding tick value collision on an axis.

Note: A Y or Y2 axis supports only the values NONE, THIN, EXTRACT, and EXTRACTALWAYS

Default:

- ROTATE for an X or X2 axis.
- NONE for a Y or Y2 axis.

The following fit policies are available:

ROTATE	Tick values are rotated 45 degrees. This value is not available for a Y or Y2 axis.
ROTATETHIN	Attempt ROTATE policy and then THIN policy. This value is not available for a Y or Y2 axis.
STAGGER	Tick values alternate between two rows. This value is not available for a Y or Y2 axis.
STAGGERROTATE	Attempt STAGGER policy and then ROTATE policy. This value is not available for a Y or Y2 axis.
STAGGERTHIN	Attempt STAGGER policy and then THIN policy. This value is not available for a Y or Y2 axis.
STAGGERTRUNCATE	Attempt STAGGER policy and then TRUNCATE policy. This value is not available for a Y or Y2 axis.
TRUNCATE	Tick values are shortened when they exceed a certain number of characters. This value is not available for a Y or Y2 axis.
TRUNCATEROTATE	Attempt TRUNCATE policy and then ROTATE policy. This value is not available for a Y or Y2 axis.
TRUNCATESTAGGER	Attempt TRUNCATE policy and then STAGGER policy. This value is not available for a Y or Y2 axis.
TRUNCATETHIN	Attempt TRUNCATE policy and then THIN policy. This value is not available for a Y or Y2 axis.
THIN	Some tick values are removed.
EXTRACT	Instead of actual tick values, display consecutive integers along the axis to represent those tick values. (See the Requirement for additional information.) In most cases, this policy is implemented if the system estimates that a collision might occur. If no collision occurs, the actual tick values are displayed on the axis in the normal manner.
EXTRACTALWAYS	Same as EXTRACT, except that the extraction is implemented regardless of whether collision occurs.

Requirement: The EXTRACT and EXTRACTALWAYS policies must be used in conjunction with an AXISLEGEND statement. The AXISLEGEND statement creates a legend that correlates the actual tick values with the consecutive integers that are displayed as axis-tick values in the graph. The AXISLEGEND

statement must reference this axis, using the name that is assigned to the axis in its NAME= axis option.

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.
VIEWMAX	Specifies the maximum data value to include in the display.
VIEWMIN	Specifies the minimum data value to include in the display.

INTEGER=*boolean*
[linear axis]
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*
[linear axis]
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 567.

Default: .30

Range: 0 - 1

Interaction: This option is ignored if the TICKVALUELIST= or TICKVALUESEQUENCE= option is used.

THRESHOLDMIN= *number*
[linear axis]
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 567.

Default: .30

Range: 0 - 1

Interaction: This option is ignored if the TICKVALUELIST= or TICKVALUESEQUENCE= option is used.

TICKVALUEFITPOLICY=THIN | *policy*
[linear axis]
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*
[linear axis]
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 a scale factor from the tick values and use it to reduce the tick value width. For large tick values, the scale factor is set to ensure that the absolute value of the largest value is greater than 1. The scale can be millions, billions, or trillions for values of 999 trillion or less, or a multiple of 10 (denoted as 10^*n*) for values over 999 trillion. For small fractional tick values, the scale factor is set to ensure that the absolute value of the smallest value is greater than 1. The scale can be millionth, billionth, or trillionth for values of 1 trillionth or more, or a multiple of 1/10 (10^–*n*) for values less than 1 trillionth. The scale used is appended to the axis label. For example: Total Sales (millions).

Restriction: The scale that is extracted by the EXTRACTSCALE= option is derived from the English locale.

Note: When EXTRACTSCALE=TRUE and a scale is extracted, the tick values are formatted to provide the best fit on the axis. In that case, the tick value format might differ from the data format even when a named format is applied to the data values.

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: GTL currently honors most but not every SAS format. For details, see "SAS Formats Not Supported" on page 855.

Interaction: If the axis label is not displayed, the EXTRACTSCALE=TRUE option is ignored.

TICKVALUELIST=(*numeric-list*)
[linear axis]
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= option.

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*
[linear axis]
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.

Interaction: This option is ignored if the DISPLAY= or DISPLAYSECONDARY= option does not display the tick values.

TICKVALUESEQUENCE=(*sequence-options*)
[linear axis]
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*
[linear axis]
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 566.

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*
[linear axis]
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 566.

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 minimum 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 and each pair is space separated.

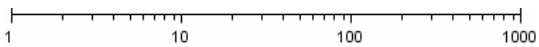
Interaction: This option is ignored if the axis type is not LOG.

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
[log axis]
specifies the base of the logarithmic scale for the axis values.

Default: 10

MINORTICKS=*boolean*
[log axis]
specifies whether minor ticks are displayed:



Default: FALSE

Restriction: Minor ticks can be displayed only when BASE=10 and TICKINTERVALSTYLE= is LOGEXPAND or LOGEXPONENT.

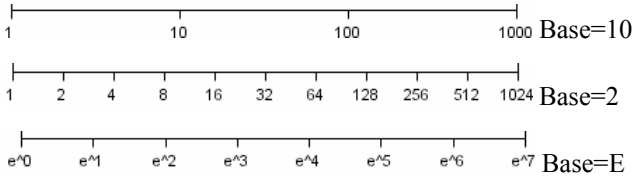
If this option is set to TRUE, the number of minor ticks is automatically determined.

TICKINTERVALSTYLE=AUTO | LOGEXPAND | LOGEXPONENT | LINEAR
[log axis]
specifies how to scale and format the values for major tick marks.

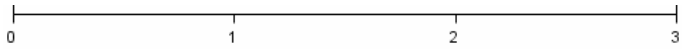
Default: AUTO

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 855.

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*
[log axis]
specifies the maximum data value to include in the display. For more information, see "Adjusting the Axis View" on page 566.

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*
[log axis]
specifies the minimum data value to include in the display. For more information, see "Adjusting the Axis View" on page 566.

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 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*
[time axis]
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.

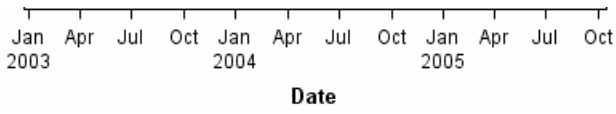
MINORTICKS=*boolean*
[time axis]
specifies whether minor tick marks are displayed.

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*
[time axis]
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 | *policy*
[time axis]
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
[time axis]
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 855.

Interaction: If this option is specified, the SPLITTICKVALUE= option is ignored.

TICKVALUELIST=(*time-constant-list* | *date-constant-list* | *datetime-constant-list* | *numeric-list*)
[time axis]
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).

Interaction: If this option is specified, the SPLITTICKVALUE= and INTERVAL= options are ignored.

TICKVALUEPRIORITY=*boolean*
[time axis]
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= and VIEWMAX= options are ignored.

Interaction: This option is ignored if the DISPLAY= or DISPLAYSECONDARY= option does not display tick marks.

VIEWMAX= *number*
[time axis]
specifies the maximum data value to include in the display. For more information, see "Adjusting the Axis View" on page 566.

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*
[time axis]
specifies the minimum data value to include in the display. For more information, see "Adjusting the Axis View" on page 566.

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 54

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 631 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.

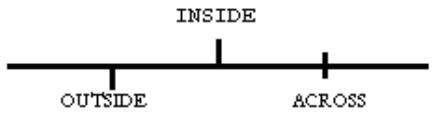
INTEGER=*boolean*
[applied to both axes]
specifies that evenly spaced integer values are used for tick marks for all axes.

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
[applied to both axes]
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*)
[applied to both axes]
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*
[applied to both axes]
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.

Interaction: This option is ignored if the DISPLAY= option does not display tick marks.

TICKVALUESEQUENCE=(*sequence-options*)
[applied to both axes]
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*
[applied to both axes]
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 566.

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*
[applied to both axes]
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 566.

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 628 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.

Equated Axis Option	Description
GRIDDISPLAY	Specifies when axis grid lines are displayed.
LABEL	Specifies the axis label.
LABELATTRS	Specifies the color and font attributes of the axis label.
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 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.
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*)
[available for an X axis or a Y axis]
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*)
[available for an X axis or a Y axis]
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*)
[available for an X axis or a Y axis]
specifies the attributes of the grid lines. See "General Syntax for Attribute Options" on page 849 for the syntax on using a *style-element* and "Line Options" on page 850 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
[available for an X axis or a Y axis]
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"*)
[available for an X axis or a Y axis]
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 562.

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*)
[available for an X axis or a Y axis]
specifies the color and font attributes of the axis label. See "General Syntax for Attribute Options" on page 849 for the syntax on using a *style-element* and "Text Options" on page 852 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*
[available for an X axis or a Y axis]
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 568.

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*
[available for an X axis or a Y axis]
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 568.

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 the tick values should appear in the reverse order.

Note: This feature is for the second maintenance release of SAS 9.3 and later. See "What's New in the SAS 9.3 Graph Template Language" on page xi.

Default: FALSE

SHORTLABEL=*"string"*
[available for an X axis or a Y axis]
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*
[available for an X axis or a Y axis]
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 567.

Default: .30

Range: 0 - 1

Interaction: This option is ignored if the TICKVALUELIST= or TICKVALUESEQUENCE= option is used.

THRESHOLDMIN= *number*
[available for an X axis or a Y axis]
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 567.

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*)
[available for an X axis or a Y axis]
specifies the color and font attributes of the axis tick values. See "General Syntax for Attribute Options" on page 849 for the syntax on using a *style-element* and "Text Options" on page 852 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*
[available for an X axis or a Y axis]
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*
[available for an X axis or a Y axis]
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 a scale factor from the tick values and use it to reduce the tick value width. For large tick values, the scale factor is set to ensure that the absolute value of the largest value is greater than 1. The scale can be millions, billions, or trillions for values of 999 trillion or less, or a multiple of 10 (denoted as 10^*n*) for values over 999 trillion. For small fractional tick values, the scale factor is set to ensure that the absolute value of the smallest value is greater than 1. The scale can be millionth, billionth, or trillionth for values of 1 trillionth or more, or a multiple of 1/10 (10^–*n*) for values less than 1 trillionth. The scale used is appended to the axis label. For example: Total Sales (millions).

Restriction: The scale that is extracted by the EXTRACTSCALE= option is derived from the English locale.

Note: When EXTRACTSCALE=TRUE and a scale is extracted, the tick values are formatted to provide the best fit on the axis. In that case, the tick value format might differ from the data format even when a named format is applied to the data values.

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: GTL currently honors most but not every SAS format. For details, see "SAS Formats Not Supported" on page 855.

Interaction: If the axis label is not displayed, the EXTRACTSCALE=TRUE option is ignored.

Chapter 55

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:

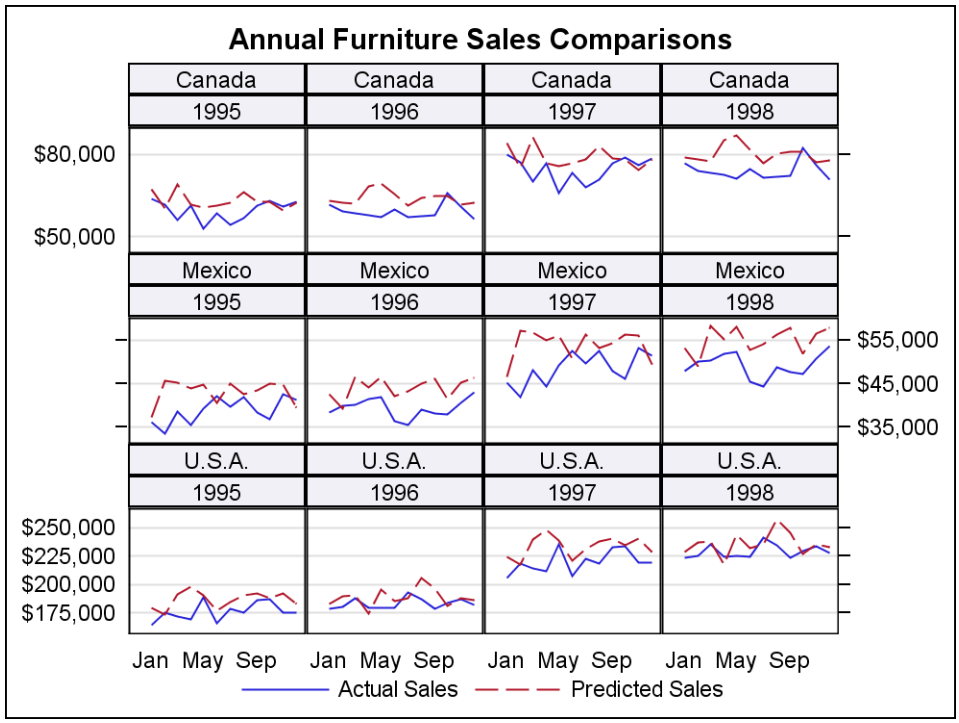
COLUMNAXISOPTS= (*axis-option(s)*)

ROWAXISOPTS= (*axis-option(s)*)

Example Program and Statement Details

Example Graph

The following graph was generated by the "Example Program" on page 640:



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 displayed on the “primary” axes by default. The axes that are considered primary depend on the settings for the XAXIS= and YAXIS= options in plot statements that are specified within the LAYOUT PROTOTYPE. For managing the primary axes, both the LAYOUT DATALATTICE and LAYOUT DATAPANEL statements have COLUMNAXISOPTS=, COLUMN2AXISOPTS=, ROWAXISOPTS=, and ROW2AXISOPTS= options that manage the axis features separately for columns and rows. The settings that are available can manage odd and even columns and rows separately, enabling you to simplify the axis display within the grid.

The following table shows which axis is primary for the XAXIS= and YAXIS= settings, and which axis option to use to manage that primary axis.

Option Setting	Primary Axis	Axis Option to Use
XAXIS=X	X (bottom)	COLUMNAXISOPTS=
XAXIS=X2	X2 (top)	COLUMN2AXISOPTS=
YAXIS=Y	Y (left)	ROWAXISOPTS=
YAXIS=Y2	Y2 (right)	ROW2AXISOPTS=

The settings that are available for the axis options 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.

You can also display “secondary” axes in the grid. A secondary axis is not an independent axis. Rather, it mirrors the primary axis, but it is displayed on the opposite side and can have different display options. For example, when the X axis (bottom) is primary, you can mirror that axis with a secondary X axis at the top of the grid. Similarly, when the Y2 axis (right) is primary, you can mirror that axis with a secondary Y2 axis on the left of the grid. A secondary axis makes it easier to interpolate values in the cells that are farthest away from the primary axis.

Secondary axes can be displayed in the graph, provided all plot statements in the LAYOUT PROTOTYPE map data to the same primary axis. For example, a secondary X axis can be displayed at the top of the layout, provided all plot statements set XAXIS=X to map data to the primary X axis (bottom). Similarly, a secondary Y2 axis can be displayed to the left in the layout, provided all plot statements set YAXIS=Y2 to map data to the primary Y2 axis (right). If all plot statements in the LAYOUT PROTOTYPE do not map data to the same primary axis, the secondary axes are not displayed.

To display secondary axes in the grid, use the DISPLAYSECONDARY= and ALTDISPLAYSECONDARY= options. As with the options for the primary axes, the DISPLAYSECONDARY= option manages the first, third, and odd occurrences of a secondary axis. The ALTDISPLAYSECONDARY= option manages the second, fourth, and even occurrences of a secondary axis.

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.

Statement Option	Description
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.
NAME	Assigns a name to an axis for reference in other statements.
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.
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. For more information about the primary and secondary axes, see "Statement Summary" on page 641 .

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.

Tip: This option can be used to obtain the alternating axis information as seen in a ScatterPlotMatrix.

See Also: GRIDDISPLAY= and GRIDATTRS= for setting axis grid lines.

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. For more information about the primary and secondary axes, see "Statement Summary" on page 641 .

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 default line attributes for the axis line and axis tick marks are defined in the GraphAxisLine style element.

Restriction: A secondary axis is not an independent axis. Rather, it mirrors the primary axis (though it can use different display features). Thus, for this option to take effect, all plot statements in the LAYOUT PROTOTYPE must map data to the same primary axis. For example, a secondary X2 axis can be displayed on top in the layout, provided all plot statements set XAXIS=X to map data to the primary X axis (bottom). Similarly, a secondary Y2 axis can be displayed to the right in the layout, provided all plot statements set YAXIS=Y to map data to the primary Y axis (left). If some plot statements set XAXIS=X and others set XAXIS=X2, both the X and X2 axis are primary and a secondary X axis cannot be displayed. In that case, this option is ignored. The same applies for the Y axes.

Tip: This option can be used to obtain the alternating axis information as seen in a ScatterPlotMatrix.

See Also: GRIDDISPLAY= and GRIDATTRS= for setting axis grid lines.

DISCRETEOPTS=(*discrete-axis-options*)
specifies one or more options for a discrete axis. See "Options for Discrete Axes" on page 650.

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. For more information about the primary and secondary axes, see "Statement Summary" on page 641 .

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 Also: 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. For more information about the primary and secondary axes, see "Statement Summary" on page 641 .

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.

Restriction: A secondary axis is not an independent axis. Rather, it mirrors the primary axis (though it can use different display features). Thus, for this option to take effect, all plot statements in the LAYOUT PROTOTYPE must map data to the same primary axis. For example, a secondary X2 axis can be displayed on top in the layout, provided all plot statements set XAXIS=X to map data to the primary X axis (bottom). Similarly, a secondary Y2 axis can be displayed to the right in the layout, provided all plot statements set YAXIS=Y to map data to the primary Y axis (left). If some plot statements set XAXIS=X and others set XAXIS=X2, both the X and X2 axis are primary and a secondary X axis cannot be displayed. In that case, this option is ignored. The same applies for the Y axes.

Tip: The default line attributes for the axis line and axis tick marks are defined in the GraphAxisLine style element.

Tip: 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 849 for the syntax on using a *style-element* and "Line Options" on page 850 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 562.

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 849 for the syntax on using a *style-element* and "Text Options" on page 852 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 652.

LOGOPTS=(*log-axis-options*)
specifies one or more options for a log axis. See "Options for Log Axes Only" on page 657.

NAME="*string*"
assigns a name to an axis for reference in other statements. Currently, it is used only in an AXISLEGEND statement.

Default: no default

Interaction: This option is ignored unless the axis is discrete. The axis can be discrete by default, or explicitly set to discrete with a TYPE= DISCRETE setting.

Interaction: For this option to take effect, an axis legend must be enabled. To enable an axis legend, the DISCRETEOPTS= option must set the TICKVALUEFITPOLICY to either EXTRACT or EXTRACTALWAYS. In addition, an AXISLEGEND statement must be specified to generate the axis legend.

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 568.

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 568.

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 849 for the syntax on using a *style-element* and "Text Options" on page 852 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 options for a time axis. See "Options for Time Axes Only" on page 659.

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 565.

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 Axes

All settings for discrete axes are set as options on the DISCRETEOPTS= option.

DISCRETEOPTS=(*discrete-axis-options*)
specifies one or more options for a discrete 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.

TICKTYPE= MIDPOINT | INBETWEEN
[discrete axis]
specifies the position of the axis tick mark.

Default: MIDPOINT

MIDPOINT	The tick mark is placed at the midpoint value location.
INBETWEEN	The tick mark is placed half way between adjacent midpoint locations.

TICKVALUEFITPOLICY=*policy*
[discrete axis]
specifies a policy for avoiding tick value collision.

Note: A Y or Y2 axis supports only the values NONE | THIN | EXTRACT | EXTRACTALWAYS

Default:

- ROTATE for an X or X2 axis
- NONE for a Y or Y2 axis

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:

NONE	Do not attempt to fit ticks that collide. This value is not available for an X or X2 axis.

ROTATE	Tick values are rotated 45 degrees. This value is not available for a Y or Y2 axis.
THIN	Some tick values are removed.
ROTATETHIN	Attempt ROTATE policy and then THIN policy. This value is not available for a Y or Y2 axis.
STAGGER	Tick values alternate between two rows. This value is not available for a Y or Y2 axis.
STAGGERROTATE	Attempt STAGGER policy and then ROTATE policy. This value is not available for a Y or Y2 axis.
STAGGERTHIN	Attempt STAGGER policy and then THIN policy. This value is not available for a Y or Y2 axis.
STAGGERTRUNCATE	Attempt STAGGER policy and then TRUNCATE policy. This value is not available for a Y or Y2 axis.
TRUNCATE	Tick values are shortened when they exceed a certain number of characters. This value is not available for a Y or Y2 axis.
TRUNCATEROTATE	Attempt TRUNCATE policy and then ROTATE policy. This value is not available for a Y or Y2 axis.
TRUNCATESTAGGER	Attempt TRUNCATE policy and then STAGGER policy. This value is not available for a Y or Y2 axis.
TRUNCATETHIN	Attempt TRUNCATE policy and then THIN policy. This value is not available for a Y or Y2 axis.
EXTRACT	Instead of actual tick values, display consecutive integers along the axis to represent those tick values. (See the Requirement for additional information.) In most cases, this policy is implemented if the system estimates that a collision might occur. If no collision occurs, the actual tick values are displayed on the axis in the normal manner.
EXTRACTALWAYS	Same as EXTRACT, except that the extraction is implemented regardless of whether collision occurs.

Requirement: The EXTRACT and EXTRACTALWAYS policies must be used in conjunction with an AXISLEGEND statement. The AXISLEGEND statement creates a legend that correlates the actual tick values with the consecutive integers that are displayed as axis-tick values in the graph. The AXISLEGEND statement must reference this axis, using the name that is assigned to the axis in its NAME= axis option.

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.
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*
[linear axis]
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*
[linear axis]
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 567.

Default: .30

Range: 0 - 1

Interaction: This option is ignored if the TICKVALUELIST= or TICKVALUESEQUENCE= option is used.

THRESHOLDMIN= *number*
[linear axis]
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 567.

Default: .30

Range: 0 - 1

Interaction: This option is ignored if the TICKVALUELIST= or TICKVALUESEQUENCE= option is used.

TICKVALUEFITPOLICY=THIN | *policy*
[linear axis]
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*
[linear axis]
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 a scale factor from the tick values and use it to reduce the tick value width. For large tick values, the scale factor is set to ensure that the absolute value of the largest value is greater than 1. The scale can be millions, billions, or trillions for values of 999 trillion or less, or a multiple of 10 (denoted as 10^*n*) for values over 999 trillion. For small fractional tick values, the scale factor is set to ensure that the absolute value of the smallest value is greater than 1. The scale can be millionth, billionth, or trillionth for values of 1 trillionth or more, or a multiple of 1/10 (10^–*n*) for values less than 1 trillionth. The scale used is appended to the axis label. For example: Total Sales (millions).

Restriction: The scale that is extracted by the EXTRACTSCALE= option is derived from the English locale.

Note: When EXTRACTSCALE=TRUE and a scale is extracted, the tick values are formatted to provide the best fit on the axis. In that case, the tick value format might differ from the data format even when a named format is applied to the data values.

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: GTL currently honors most but not every SAS format. For details, see "SAS Formats Not Supported" on page 855.

Interaction: If the axis label is not displayed, the EXTRACTSCALE=TRUE option is ignored.

TICKVALUELIST=(*numeric-list*)
[linear axis]
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*
[linear axis]
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.

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*)
[linear axis]
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*
[linear axis]
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 566.

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*
[linear axis]
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 566.

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.

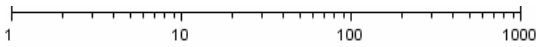
Interaction: This option is ignored if the axis type is not LOG.

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
[log axis]
specifies the base of the logarithmic scale for the axis values.

Default: 10

MINORTICKS=*boolean*
[log axis]
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
[log axis]
specifies how to scale and format the values for major tick marks.

Default: AUTO

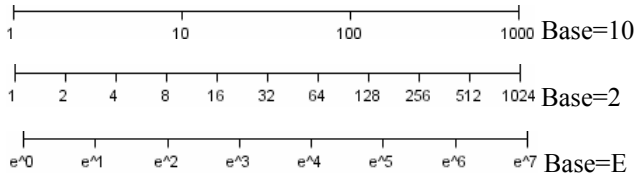
Restriction: For LOGEXPAND and LOGEXPONENT, formats on data columns contributing to the axis are ignored. For LINEAR, ticks values are automatically formatted when the column format is not assigned or one of *w.d*, *Ew.*, or *BESTw*. Other formats (SAS defined or user-defined) are used if specified.

Restriction: GTL currently honors most but not every SAS format. For details, see "SAS Formats Not Supported" on page 855.

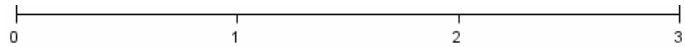
If you use TICKINTERVALSTYLE=LOGEXPONENT, you might want to include information in the axis label about which base is used.

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 855.

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*
[log axis]
specifies the maximum data value to include in the display. For more information, see "Adjusting the Axis View" on page 566.

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*
[log axis]
specifies the minimum data value to include in the display. For more information, see "Adjusting the Axis View" on page 566.

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.
VIEWMAX	Specifies the maximum data value to include in the display.
VIEWMIN	Specifies the minimum data value to include in the display.

INTERVAL=*interval*
[time axis]
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.
SEMIYEAR	DATE or DATETIME	6 months	MONYY7.
YEAR	DATE or DATETIME	year	YEAR4.

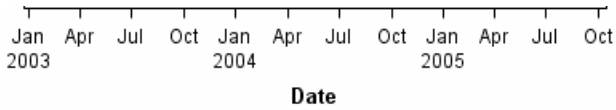
MINORTICKS=Boolean
[time axis]
specifies whether minor ticks are displayed.

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
[time axis]
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 | *policy*
[time axis]
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
[time axis]
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 855.

Interaction: If this option is specified, the SPLITTICKVALUE= option is ignored.

TICKVALUELIST=(*time-constant-list* | *date-constant-list* | *datetime-constant-list* | *numeric-list*)
[time axis]
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).

Interaction: If this option is specified, the SPLITTICKVALUE= and INTERVAL= options are ignored.

VIEWMAX= *number*
[time axis]
specifies the maximum data value to include in the display. For more information, see "Adjusting the Axis View" on page 566.

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*
[time axis]
specifies the minimum data value to include in the display. For more information, see "Adjusting the Axis View" on page 566.

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 56

AXISLEGEND Statement

Description and Syntax

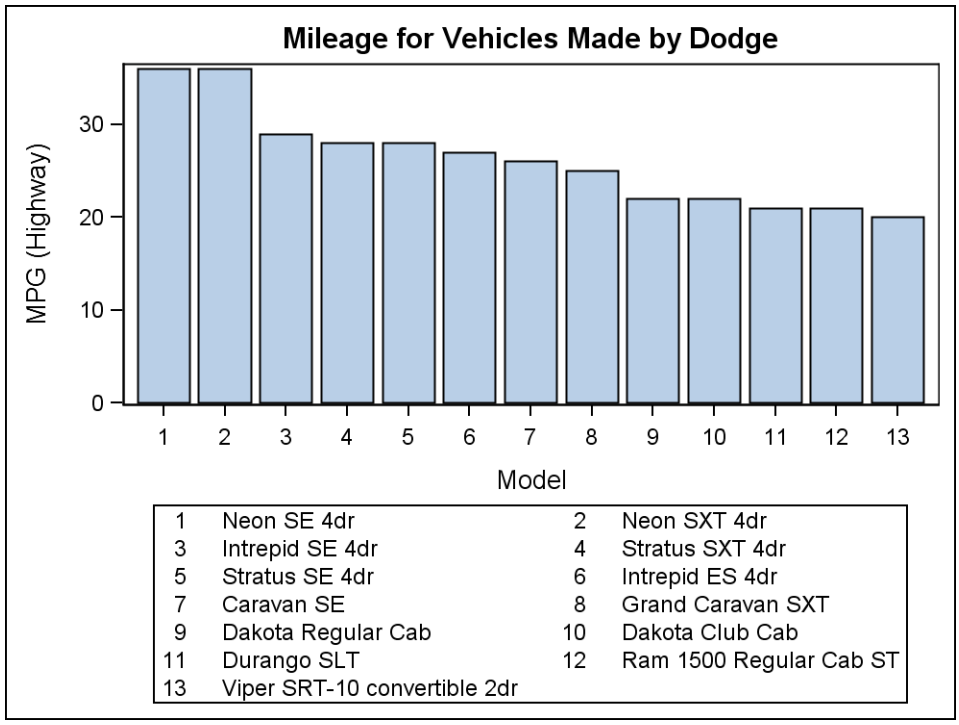
Generates consecutive integers for display as axis-tick values in the graph, and creates a legend that correlates those integers with the actual tick values that they represent.

AXISLEGEND *"axis-name" </option(s)>*;

Example Program and Statement Details

Example Graph

The following graph was generated by the “Example Program” on page 666 . The LAYOUT OVERLAY statement’s XAXISOPTS= option uses NAME= to assign a name to the X axis so that it can be referenced in an AXISLEGEND statement. The LAYOUT OVERLAY statement’s DISCRETEOPTS= option specifies TICKVALUEFITPOLICY=EXTRACT, which implements the axis legend if a collision occurs in the tick-value display for the X axis. The AXISLEGEND statement references the name that was assigned to the X axis.



Example Program

```
proc template;
 define statgraph axislegend ;
  begingraph;
   entrytitle "Mileage for Vehicles Made by Dodge";
   layout overlay / xaxisopts=(name="xaxis"
       discreteopts=(tickvaluefitpolicy=extract)) ;
     barchart x=model y=mpg_highway / stat=mean ;
     axislegend "xaxis";
   endlayout;
  endgraph;
 end;

proc sort data=sashelp.cars out=dodge; by descending mpg_highway;
  where make="Dodge";

proc sgrender data=dodge template=axislegend;
run;
```

Statement Summary

An AXISLEGEND statement is used to consolidate the axis display so that a large number of tick values can be clearly displayed without collision. In an axis legend, the legend symbol is a positive integer that is generated to represent a specific axis tick position, and the legend value displays the axis tick value. In the “Example Graph” on page 665 , the first legend symbol is the integer **1** and the first legend value is **Neon SE 4dr**.

To implement an axis legend, all of the following requirements must be met:

- The axis must be a discrete axis. The axis can be discrete by default, or explicitly set to discrete by setting TYPE=DISCRETE among the parent layout's axis-option settings.
- The parent layout's axis options must specify NAME= to assign a name to the axis.
- The parent layout's axis options must use the DISCRETEOPTS= option to specify either TICKVALUEFITPOLICY=EXTRACT or TICKVALUEFITPOLICY=EXTRACTALWAYS. EXTRACT implements an axis legend if the system estimates that a collision might occur. EXTRACTALWAYS implements an axis legend regardless of whether a collision occurs.
- The AXISLEGEND statement must reference the axis name that was specified in the parent layout's NAME= axis option.

When all of these requirements are satisfied, the tick values of the named discrete axis can be replaced with consecutive positive integers. The axis legend displays those integer values and the corresponding tick values that the integers represent.

Within an overlay-type layout, when an axis legend is placed inside the plot area with LOCATION= INSIDE,

- The axis legend 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 an axis 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 an axis legend is placed within nested layouts, it might be necessary to do one of the following to obtain the desired legend organization:

- use the ACROSS= option and also set ORDER= ROWMAJOR
- use the DOWN= option and also set ORDER= COLUMNMAJOR

A legend might be dropped if the total legend area in the graph exceeds the percentage that is set by the MAXLEGENDAREA= option in an ODS GRAPHICS statement that is in effect for the output destination. A legend might also be dropped if DISPLAYCLIPPED= FALSE and the full legend cannot be displayed.

Arguments

"*axis-name*"
references an axis's name. The name controls which axis determines the legend entries.

Requirement: The *axis-name* must be enclosed in quotation marks.

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.
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 cannot be rendered entirely.
DOWN	Specifies the number of legend entries that are placed vertically before the next column begins.
HALIGN	Specifies the horizontal alignment of the legend within its parent layout when nested within an overlay-type or region layout.
LOCATION	Specifies whether the legend appears inside or outside the plot area when the legend is specified within an overlay-type or region layout.
OPAQUE	Specifies whether the legend background is opaque.
ORDER	Specifies whether legend entries are organized into rows or into columns.
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 or region 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 works only when the legend is nested in the template's outermost layout.

Restriction: This option is ignored if ORDER= COLUMNMAJOR

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 "Statement Summary" on page 105 .

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 849 for the syntax on using a *style-element* and "Line Options" on page 850 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 cannot be rendered entirely.

Default: FALSE

Discussion: Based on the legend contents and placement, or when the graph size is reduced, parts of the legend (title, legend symbol, or legend value) might be clipped (truncated). When clipping occurs and this option is set to FALSE, the entire legend is removed from the graph. The space for that legend is then reclaimed by the remainder of the graph. When this option is set to TRUE, the legend always appears, even if some parts of it 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 works only when the legend is nested in the template's outermost layout.

Restriction: This option is ignored if ORDER= ROWMAJOR

HALIGN=CENTER | LEFT | RIGHT | *number*
specifies the horizontal alignment of the legend within its parent layout when nested within an overlay-type or region layout. For more information about how child positions are determined in an overlay-type or region layout, see the LAYOUT OVERLAY's "Statement Summary" on page 105 .

Default: CENTER

Range: A *number* specification can range from 0 to 1. The *value* represents a fraction of the parent container's width, where 0 is all the way to the left and 1 is all the way to the right.

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

Interaction: If LOCATION= OUTSIDE, the HALIGN= and VALIGN= options cannot both be set to CENTER.

Interaction: For a *number* setting to take effect, LOCATION=INSIDE must be set. A *number* setting is invalid on this option when LOCATION=OUTSIDE.

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 when the legend is specified within an overlay-type or region layout.

Default: OUTSIDE

Restriction: This option has effect only when the legend statement appears within an overlay-type or region layout and at least one stand-alone plot statement is referenced by the parent layout. For more information about how child positions are determined in an overlay-type or region layout, see the LAYOUT OVERLAY's "Example Program and Statement Details" on page 103.

Interaction: The actual position is determined by the settings for the LOCATION=, AUTOALIGN= , HALIGN= , and VALIGN= options.

Interaction: If this option is set to OUTSIDE, the HALIGN= and VALIGN= options must specify a keyword (LEFT, RIGHT, or CENTER). The *number* setting for the alignment is invalid when the legend is positioned outside of the plot area.

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 organized into rows or into 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 849 for the syntax on using a *style-element* and "Text Options" on page 852 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 | *number*
specifies the vertical alignment of the legend within its parent layout when nested within an overlay-type or region layout. For more information about how child positions are determined in an overlay-type or region layout, see the LAYOUT OVERLAY's "Statement Summary" on page 105 .

Default: CENTER

Range: A *number* specification can range from 0 to 1. The *value* represents a fraction of the parent container's height, where 0 is on the bottom and 1 is on the top.

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

Interaction: If LOCATION= OUTSIDE, the VALIGN= and HALIGN= options cannot both be set to CENTER.

Interaction: For a *number* setting to take effect, LOCATION=INSIDE must be set. A *number* setting is invalid on this option when LOCATION=OUTSIDE.

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 849 for the syntax on using a *style-element* and "Text Options" on page 852 for available *text-options*.

Default: The GraphValueText style element.

Chapter 57

CONTINUOUSLEGEND Statement

Description and Syntax

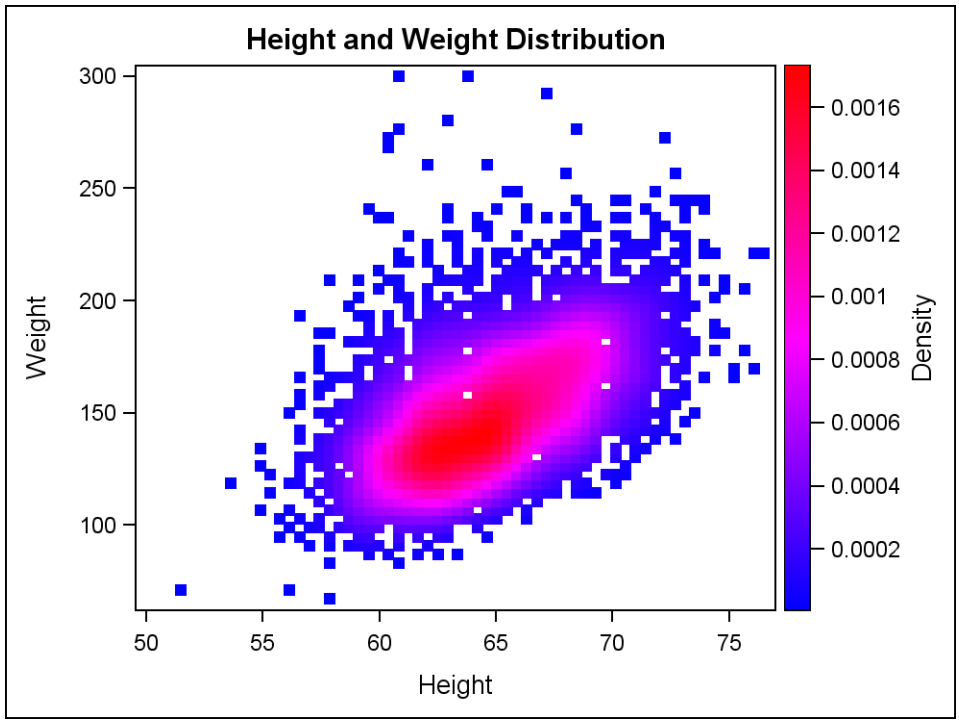
Creates a legend with a color ramp corresponding to a range of values.

CONTINUOUSLEGEND "*graph-name*" </*option(s)*>;

Example Program and Statement Details

Example Graph

The following graph was generated by the "Example Program" on page 674:



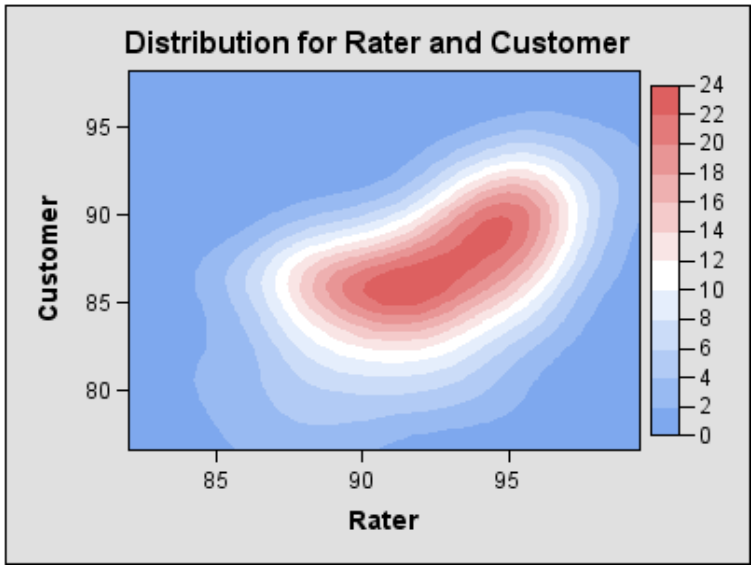
Example Program

```
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;
```

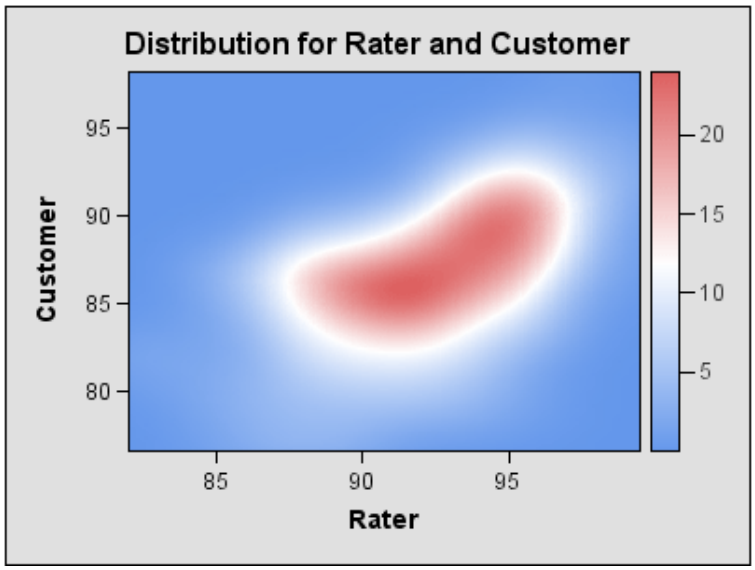
Statement Summary

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 or region 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 or region 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 or region 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 or region layout. For more information about how child positions are determined in an overlay-type layout, see the LAYOUT OVERLAY's "Statement Summary" on page 105 .

Default: NONE

NONE
does not automatically align the legend within its parent layout. The legend's position is set by the HALIGN= and VALIGN= options.

AUTO
centers the legend in the area that is farthest from any surrounding data point markers within the parent layout. Available only if the parent layout contains a scatter plot; ignored otherwise.

(*location-list*)
within the parent layout, restricts the legend's possible locations to those locations in the specified *location-list*, and uses 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 849 for the syntax on using a *style-element* and "Line Options" on page 850 for available *line-options*.

Default: The GraphBorderLines style element.

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

Tip: The color of the frame around the color ramp and the color ramp tick lines is controlled by the GraphAxisLines:contrastcolor style element.

HALIGN= RIGHT | CENTER | LEFT | *number*
specifies the horizontal alignment of the legend within its parent layout when nested within an overlay-type or region layout. For more information about how child positions are determined in an overlay-type or region layout, see the LAYOUT OVERLAY's "Statement Summary" on page 105 .

Default: RIGHT if LOCATION=OUTSIDE. CENTER if LOCATION=INSIDE

Range: A *number* specification can range from 0 to 1. The *number* represents a fraction of the parent container's width, where 0 is all the way to the left and 1 is all the way to the right.

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

Interaction: If LOCATION= OUTSIDE, the HALIGN= and VALIGN= options cannot both be set to CENTER.

Interaction: For a *number* setting to take effect, LOCATION=INSIDE must be set. A *number* setting is invalid on this option when LOCATION=OUTSIDE.

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 103.

Interaction: The actual position is determined by the settings for the LOCATION=, AUTOALIGN= , HALIGN= , and VALIGN= options.

Interaction: If this option is set to OUTSIDE, the HALIGN= and VALIGN= options must specify a keyword (LEFT, RIGHT, or CENTER). The *number* setting for the alignment is invalid when the legend is positioned outside of the plot area.

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 849 for the syntax on using a *style-element* and "Text Options" on page 852 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 | *number*
specifies the vertical alignment of the legend within its parent layout when nested within an overlay-type or region layout. For more information about how child positions are determined in an overlay-type or region layout, see the LAYOUT OVERLAY's "Statement Summary" on page 105 .

Default: CENTER

Range: A *number* specification can range from 0 to 1. The *number* represents a fraction of the parent container's height, where 0 is on the bottom and 1 is on the top.

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

Interaction: If LOCATION= OUTSIDE, the VALIGN= and HALIGN= options cannot both be set to CENTER.

Interaction: For a *number* setting to take effect, LOCATION=INSIDE must be set. A *number* setting is invalid on this option when LOCATION=OUTSIDE.

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 849 for the syntax on using a *style-element* and "Text Options" on page 852 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 58

DISCRETELEGEND| MERGEDLEGEND Statement

Description and Syntax

Creates a legend with entries that refer to plots, or group values, or both legend items.

Often the data in a plot is classified by a group variable. Or, multiple columns of data are plotted in the same graph. These groups or columns are represented in the graph by different color or line patterns or marker symbols. In these cases, a DISCRETELEGEND can be added to the graph to help decode the data. The DISCRETELEGEND statements uses the following syntax:

DISCRETELEGEND

"graph-name" | *"legend-item-name"* | *"discrete-attribute-map-name"*

<…*"graph-name-n"* | *"legend-item-name-n"* | *"discrete-attribute-map-name-n"*>

</option(s)>;

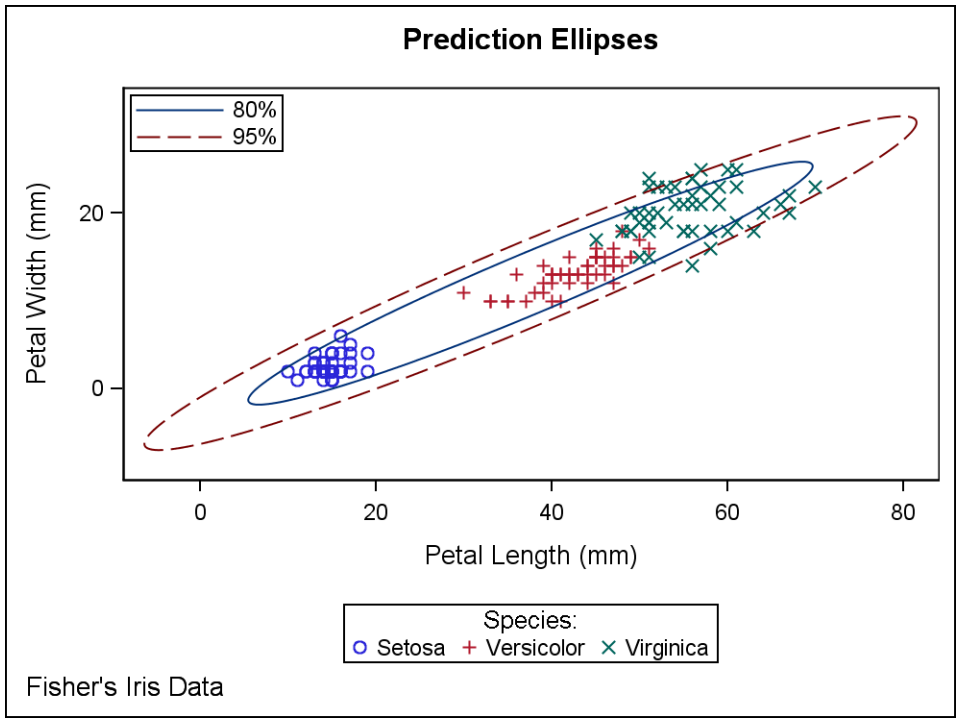
The MERGEDLEGEND statement can be used to consolidate legend entries when the graph displays grouped data for two plots. The statement must specify exactly two names that reference the source for the legend entry values:

MERGEDLEGEND
"graph-name" | "legend-item-name"
"graph-name" | "legend-item-name"
</option(s)>;

Example Program and Statement Details

Example Graph 1

The following graph using the DISCRETELEGEND statement was generated by "Example Program 1" . It displays two discrete legends, one that shows the confidence limits for two ellipses and a second that shows the values for a GROUP= variable:



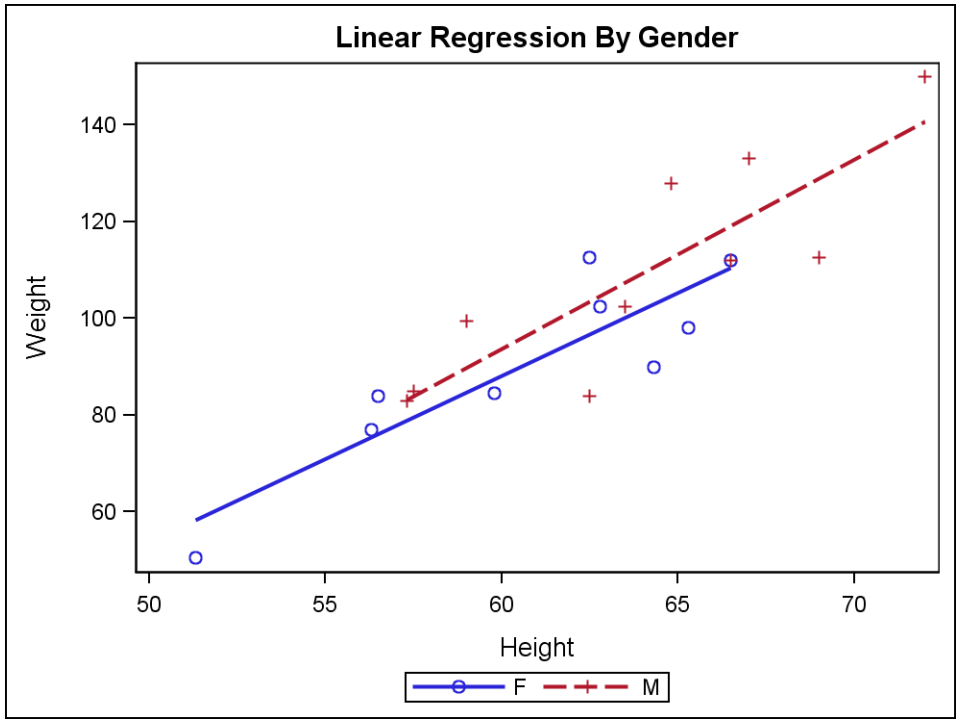
Example Program 1

```
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;

proc sgrender data=sashelp.iris template=discretelegend;
run;
```

Example Graph 2

The following graph using the MERGEDLEGEND statement was generated by "Example Program 2" . In the template definition, a grouped scatter plot is overlaid with a series plot for each group, and the two plots are referenced by a single merged legend:



Example Program 2

```
proc template;
  define statgraph mergedLegend;
    begingraph;
      Layout overlay;
        scatterplot x=age y=height / group=sex name='scatter';
        seriesplot x=age y=height / group=sex name='series' connectorder=xaxis;
        mergedlegend 'scatter' 'series' ;
      endlayout;
    endgraph;
```

```
  end;

proc sgrender data=sashelp.class template=mergedLegend;
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 filled symbol that represents a specific area in the plot. The legend value is descriptive text that is derived from group values, or assigned with the plot's LEGENDLABEL= option, or with the LEGENDITEM statement's LABEL= option. To specify a discrete legend, you can use the DISCRETELEGEND statement. To consolidate legend entries for common grouped values that are represented by two separate plots, you can use the MERGEDLEGEND statement.

For grouped plots, a discrete legend represents all of the values that are present in the data. To ensure that legend entries are displayed for group values, regardless of whether those values are present in the data, you can define an attribute map for the group values. You can then specify that attribute map directly in the DISCRETELEGEND statement. For more information and an example, see the DISCRETEATTRMAP statement's "Statement Summary" on page 807 .

Note: An attribute map cannot be referenced directly in a MERGEDLEGEND statement.

The DISCRETELEGEND statement can specify one or more names that reference the source for the legend entry values. You can use the TYPE= option to control the visual attributes for the legend display. The TYPE= option is required when you specify an attribute map directly in the DISCRETELEGEND statement.

For legend items that represent fills or colors without a specific shape, a filled symbol with a one pixel, solid outline is used to represent the data values. If the feature being represented by the legend item has an outline, the default outline color is derived from the feature's outline color. If the feature does not have an outline, the default outline color for its corresponding legend entry is derived from the GraphOutline style-element.

The MERGEDLEGEND statement can be used to consolidate lines and marker symbols from discrete legend entries when a graph displays grouped values for exactly two plots. With a MERGEDLEGEND statement, the legend values from the group variables in two plots are compared. For each common value, the corresponding legend lines and marker symbols are combined, creating only one legend entry for each matching set of group values. The MERGEDLEGEND statement can be used only for grouped plots and must specify two names that reference the source for the legend entry values.

Within an overlay-type layout, when a discrete legend is placed inside the plot area with LOCATION= INSIDE,

- The discrete legend 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. (The AUTOALIGN= option is not available in a LAYOUT OVERLAY3D statement.)

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 do one of the following to obtain the desired legend organization:

- use the ACROSS= option and also set ORDER= ROWMAJOR
- use the DOWN= option and also set ORDER=COLUMNMAJOR

A legend might be dropped if the total legend area in the graph exceeds the percentage that is set by the MAXLEGENDAREA= option in an ODS GRAPHICS ON statement that is in effect for the output destination. A legend might also be dropped if DISPLAYCLIPPED= FALSE and the full legend cannot be displayed.

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 that is present in the data.

- 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 or MERGEDLEGEND statement because it requires a continuous legend.

"*legend-item-name*"
references one or more unique values specified by the NAME= option of a LEGENDITEM statement. Each *legend-item-name* must be enclosed in quotation marks.

"*discrete-attribute-map-name*"
references one or more unique values specified by the NAME= option of a DISCRETEATTRMAP statement. This usage results in the referenced discrete attribute map contributing all its value statements as legend entries, regardless of whether they match the data. Each *discrete-attribute-map-name* must be enclosed in quotation marks. For more details and an example use, see the DISCRETEATTRMAP statement's "Statement Summary" on page 807 .

Restriction. For legend items that are contributed by a discrete attribute map, the marker-symbol size and line thickness are fixed and cannot be changed. Thus, any settings in the attribute map that specify a size for a marker symbol or a thickness for a line are ignored in the legend.

Note: This form of reference cannot be used in a MERGEDLEGEND statement.

Options

The following options are available for both the DISCRETELEGEND and MERGEDLEGEND statements, unless noted otherwise.

Statement Option	Description
ACROSS	Specifies the number of legend entries that are placed horizontally before the next row begins.

Statement Option	Description
AUTOALIGN	Specifies whether the legend is automatically aligned within its parent layout when nested within an overlay-type layout.
AUTOITEMSIZE	Specifies that all markers, lines, and filled symbols in the legend are sized in proportion to the font size used for the legend entry labels.
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 cannot be rendered entirely.
DOWN	Specifies the number of legend entries that are placed vertically before the next column begins.
EXCLUDE	Specifies a list of legend items to exclude from the display.
HALIGN	Specifies the horizontal alignment of the legend within its parent layout when nested within an overlay-type or region layout.
LOCATION	Specifies whether the legend appears inside or outside the plot area when the legend is specified within an overlay-type or region layout.
OPAQUE	Specifies whether the legend background is opaque.
ORDER	Specifies whether legend entries are organized into rows or into columns.
PAD	Specifies the amount of extra space that is added inside the legend perimeter.
SORTORDER	Specifies the sort order to use for the legend entry labels.
TITLE	Specifies the title of the legend.
TITLEATTRS	Specifies the color and font attributes of the legend title.

Statement Option	Description
TITLEBORDER	Specifies a border around the legend title that separates the title from the legend body.
TYPE	specifies which visual attributes to display for legend items in the legend.
VALIGN	Specifies the vertical alignment of the legend within its parent layout when nested within an overlay-type or region 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. A legend entry typically consists of two parts, such as a marker symbol and an associated value.

Default: The entries are placed to best fit the available area. This "best fit" approach works only when the legend is nested in the template's outermost layout.

Restriction: This option is ignored if ORDER= COLUMNMAJOR

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 "Statement Summary" on page 105 .

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= .

AUTOITEMSIZE= *boolean*
specifies that all markers, lines, and filled symbols in the legend are sized in proportion to the font size used for the legend entry labels. These proportional sizes take effect regardless of the size reported by the plot or LEGENDITEM. The line

segments are drawn as deemed appropriate by the legend, regardless of the line thickness reported by the plot.

Default: FALSE

Interaction: When set to TRUE, this setting considers the font size in effect from the VALUEATTRS= option.

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 849 for the syntax on using a *style-element* and "Line Options" on page 850 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 cannot be rendered entirely.

Default: FALSE

Discussion: Based on the legend contents and placement, or when the graph size is reduced, parts of the legend (title, legend symbol, or legend value) might be clipped (truncated). When clipping occurs and this option is set to FALSE, the entire legend is removed from the graph. The space for that legend is then reclaimed by the remainder of the graph. When this option is set to TRUE, the legend always appears, even if some parts of it have been clipped.

DOWN=*positive-integer*
specifies the number of legend entries that are placed vertically before the next column begins. A legend entry typically consists of two parts, such as a marker symbol and an associated value.

Default: The entries are placed to best fit the available area. This "best fit" approach works only when the legend is nested in the template's outermost layout.

Restriction: This option is ignored if ORDER= ROWMAJOR

EXCLUDE= (*item-names*)
specifies a list of legend entries to exclude from the display. The names must match the legend entry label that you want to exclude.

Default: No items are excluded.

Tip: For plots with groups, you can exclude specific group values.

HALIGN=CENTER | LEFT | RIGHT | *number*
specifies the horizontal alignment of the legend within its parent layout when nested within an overlay-type or region layout. For more information about how child positions are determined in an overlay-type or region layout, see the LAYOUT OVERLAY's "Statement Summary" on page 105 .

Default: CENTER

Range: A *number* specification can range from 0 to 1. The *number* represents a fraction of the parent container's width, where 0 is all the way to the left and 1 is all the way to the right.

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

Interaction: If LOCATION= OUTSIDE, the HALIGN= and VALIGN= options cannot both be set to CENTER.

Interaction: For a *number* setting to take effect, LOCATION=INSIDE must be set. A *number* setting is invalid on this option when LOCATION=OUTSIDE.

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 when the legend is specified within an overlay-type or region layout.

Default: OUTSIDE

Restriction: This option has effect only when the legend statement appears within an overlay-type or region layout and at least one stand-alone plot statement is referenced by the parent layout. For more information about how child positions are determined in an overlay-type or region layout, see the LAYOUT OVERLAY's "Example Program and Statement Details" on page 103.

Interaction: The actual position is determined by the settings for the LOCATION=, AUTOALIGN= , HALIGN= , and VALIGN= options.

Interaction: If this option is set to OUTSIDE, the HALIGN= and VALIGN= options must specify a keyword (LEFT, RIGHT, or CENTER). The *number* setting for the alignment is invalid when the legend is positioned outside of the plot area.

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 organized into rows or into 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.

SORTORDER= AUTO | ASCENDINGFORMATTED | DESCENDINGFORMATTED
specifies the sort order to use for the legend entry labels.

Default: AUTO. Groups of legend entries are presented in the order in which they are listed in the legend statement. The internal ordering of the entries is derived from the constituent plot-statement options or the attribute map.

Interaction: This option overrides the order that is set by any constituent plot statement's GROUPORDER= option.

Interaction: If this option is set to ASCENDINGFORMATTED or DESCENDINGFORMATTED, the entries from separate plots, and separate legend items, and separate attribute maps are combined and ordered as a single list.

Note: The ASCENDINGFORMATTED and DESCENDINGFORMATTED settings perform a linguistic sort on the group items and have the same effect as sorting the input data. However, the data is not changed.

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 849 for the syntax on using a *style-element* and "Text Options" on page 852 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.

TYPE= ALL | FILL | FILLCOLOR | LINE | LINECOLOR | LINEPATTERN | MARKER | MARKERCOLOR | MARKERSYMBOL
specifies which visual attributes to display for legend entries in the legend.

Default: ALL

Restrictions:

- This option cannot be used in the MERGEDLEGEND statement
- This option is required when a discrete attribute map is directly referenced by the legend. Keyword ALL, the default, cannot be used in that case, and this option must be used to change the default. If the value ALL is not changed, the discrete-attribute map's contributions to the legend entries are ignored.
- A LEGENDITEM statement can be referenced only from a discrete legend of the same or overlapping attribute type. For legends that display multiple visual attributes (use both colors and marker symbols, for example), the default visual properties are derived from the GraphDataDefault style-element.

Discussion: This option can be used as a filter. If a statement contributing to the legend does not have any visual attributes that match the TYPE specified, the legend does not display any entries from that statement.

Some keywords can be used to create specialized legends that display a single visual attribute. For example, keywords FILLCOLOR or MARKERSYMBOL result in the display of a single attribute. Other keywords (for example, FILL, LINE, or MARKER) result in legends that display a set of visual attributes. For example, keyword LINE results in the display of both line color and line pattern for legend entries that include lines in their display.

If this option is set to LINEPATTERN or MARKERSYMBOL, a filled symbol is drawn using the same text color as the color used for the legend entry labels. The symbol is sized automatically, as if the AUTOITEMSIZE= option is set to TRUE. For keywords LINECOLOR and MARKERCOLOR, the filled symbols are drawn as color swatches.

VALIGN=CENTER | TOP | BOTTOM | *number*
specifies the vertical alignment of the legend within its parent layout when nested within an overlay-type or region layout. For more information about how child positions are determined in an overlay-type or region layout, see the LAYOUT OVERLAY's "Statement Summary" on page 105 .

Default: CENTER

Range: A *number* specification can range from 0 to 1. The *number* represents a fraction of the parent container's height, where 0 is on the bottom and 1 is on the top.

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

Interaction: If LOCATION= OUTSIDE, the VALIGN= and HALIGN= options cannot both be set to CENTER.

Interaction: For a *number* setting to take effect, LOCATION=INSIDE must be set. A *number* setting is invalid on this option when LOCATION=OUTSIDE.

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 849 for the syntax on using a *style-element* and "Text Options" on page 852 for available *text-options*.

Default: The GraphValueText style element.

LEGENDITEM Statement

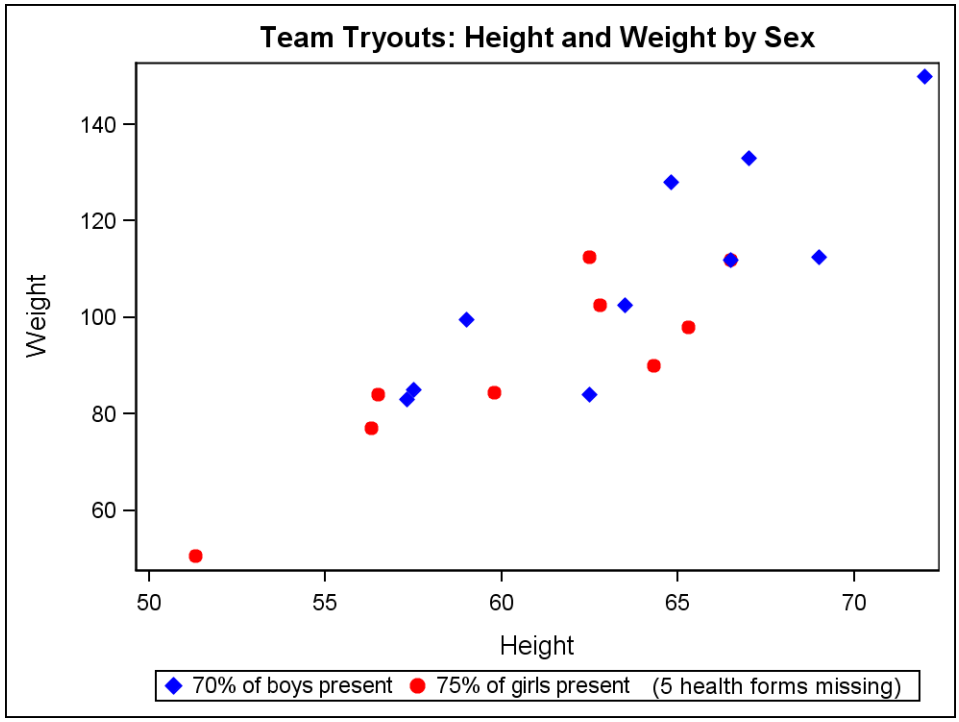
LEGENDITEM Description and Syntax

Creates the definition for a legend item that can be included in a discrete legend.

LEGENDITEM TYPE=*type* NAME="*string*" <*/option(s)*>;

LEGENDITEM Example Graph

The following graph was generated by the "LEGENDITEM Example Program" on page 693. The example specifies three LEGENDITEM statements to define graphical properties for two marker symbols and a text string. The NAME= option on each LEGENDITEM statement assigns a name to the definition. Those names are referenced in a DISCRETELEGEND statement so that the definitions are displayed in the graph legend. To correlate the legend with the scatter plot, the example creates an attribute map that matches values M and F to the same graphical properties that are specified in the LEGENDITEM statements. That attribute map is referenced in the scatter plot.



LEGENDITEM Example Program

```
proc template;
  define statgraph scatterplot;
    begingraph;
      entrytitle "Team Tryouts: Height and Weight by Sex";
      discreteattrmap name="symbols" / ignorecase=true trimleading=true;
        value "m"    / markerattrs=(color=blue symbol=diamondfilled);
        value "f"    / markerattrs=(color=red  symbol=circlefilled);
      enddiscreteattrmap;
      discreteattrvar attrvar=groupmarkers var=sex attrmap="symbols";
      legendItem type=marker name="m_marker" /
        markerattrs=(color=blue symbol=diamondfilled)
        label="70% of boys present" ;
      legendItem type=marker name="f_marker" /
        markerattrs=(color=red  symbol=circlefilled)
        label="75% of girls present" ;
      legendItem type=text name="status" /
        text=" (5 health forms missing)" ;
      layout overlay;
        scatterplot x=height y=weight / group=groupmarkers;
        discretelegend "m_marker""f_marker" "status" /
          autoitemsize=true;
      endlayout;
    endgraph;
  end;
proc sgrender data=sashelp.class template=scatterplot;
run;
```

LEGENDITEM Summary

The LEGENDITEM statement creates a definition for a legend item that can be included in a legend. The item that you define is independent of the data and enables you to customize the legend to enhance or replace a standard legend. For example, to display annotation text within the legend area, you can define a LEGENDITEM statement with TYPE=TEXT and specify the text in the TEXT= option.

As demonstrated in the "LEGENDITEM Example Program" on page 693, multiple LEGENDITEM statements can be used to replace a plot statement's data-driven legend by defining custom legend items to display in the legend. This use enables you to communicate information that is not in the data. For this use, you would define one or more LEGENDITEM statements to specify legend attributes, and then reference those items in your legend statement. You must not reference the plot statement itself in the legend statement. Although no direct connection would exist between the plot data and the legend, you could communicate the connection by managing the visual attributes in both the plot and the legend.

The LEGENDITEM statement must be located within the BEGINGRAPH block but outside of the outermost layout block. You can use multiple LEGENDITEM statements to define multiple definitions. In that case, each definition specifies a single legend entry and each item name must be referenced in the legend statement.

Note: A single legend statement can reference multiple item names and also multiple plot names.

When specifying attribute options for a particular LEGENDITEM statement, options that do not apply to the specified TYPE= value are ignored. For example, the MARKERATTRS= option sets the attributes for a marker and is useful if you set TYPE=MARKER or TYPE=MARKERLINE. However, if TYPE= is set to a value that does not display a marker symbol, the MARKERATTRS= option is ignored.

LEGENDITEM Required Arguments

TYPE = FILL | MARKER | MARKERLINE | LINE | TEXT
specifies a type for the legend item.

FILL	specifies a fill (displayed as a filled square). The appearance can be managed with the FILLATTRS= and OUTLINEATTRS= options.
MARKER	specifies a marker. The appearance can be managed with the MARKERATTRS= option.
MARKERLINE	specifies a marker and a line. The appearance can be managed with the MARKERATTRS= and LINEATTRS= options.
LINE	specifies a line. The appearance can be managed with the LINEATTRS= option.
TEXT	specifies text that can be displayed in the legend area. The text string is defined with the TEXT= option, and the text appearance can be managed with the TEXTATTRS= option.

NAME="*string*"
assigns a name to the legend item for reference in a DISCRETELEGEND or MERGEDLEGEND statement.

Restriction: The *string* is case sensitive and must define a unique name within the template.

LEGENDITEM Options

Statement Option	Description
FILLATTRS	Specifies the color of the fill when TYPE= is set to FILL.
LABEL	Specifies a label to be used for the legend item.
LABELATTRS	Specifies the color and font attributes of the legend item's label.
LINEATTRS	Specifies the appearance of the line when TYPE= is set to LINE or MARKERLINE.
MARKERATTRS	Specifies the appearance of the marker when TYPE= is set to MARKER or MARKERLINE.

Statement Option	Description
OUTLINEATTRS	Specifies the color of the outline when TYPE= is set to FILL.
TEXT	Specifies the text to display when TYPE= is set to TEXT.
TEXTATTRS	Specifies the font and color attributes of the *string* that is specified on the TEXT= option when TYPE= is set to TEXT.

FILLATTRS=*style-element* | *style-element* (*fill-options*) | (*fill-options*)
specifies the color of the fill when TYPE= is set to FILL. See "General Syntax for Attribute Options" on page 849 for the syntax on using a *style-element* and "Fill Options" on page 850 for available *fill-options*.

Default: The GraphDataDefault style element.

Interaction: The TRANSPARENCY attribute cannot be derived from the style element, but it can be set with this option.

LABEL= "*string*"
specifies a label to be used for the legend item.

Default: no label is displayed

Tip: The font and color attributes for the label are specified by the LABELATTRS= option.

LABELATTRS
specifies the color and font attributes of the legend item's label.

Default: The GraphValueText style element.

Interaction: For this option to take effect, the LABEL= option must also be specified.

LINEATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)
specifies the appearance of the line when TYPE= is set to LINE or MARKERLINE. See "General Syntax for Attribute Options" on page 849 for the syntax on using a *style-element* and "Line Options" on page 850 for available *line-options*.

Default: The GraphDataDefault style element.

MARKERATTRS=*style-element* | *style-element* (*marker-options*) | (*marker-options*)
specifies the appearance of the marker when TYPE= is set to MARKER or MARKERLINE. See "General Syntax for Attribute Options" on page 849 for the syntax on using a *style-element* and "Marker Options" on page 851 for available *marker-options*.

Default: The GraphDataDefault style element.

Interaction: The WEIGHT attribute cannot be derived from the style element, but it can be set with this option.

OUTLINEATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)
specifies the color of the outline when TYPE= is set to FILL. See "General Syntax for Attribute Options" on page 849 for the syntax on using a *style-element* and "Line Options" on page 850 for available *line-options*.

Default: The GraphDataDefault style element.

This option enables you to set the color. It ignores settings for the line pattern and line thickness.

TEXT= "*string*"
specifies the text to display when TYPE= is set to TEXT.

Default: a blank space

The font and color attributes for the text are specified by the TEXTATTRS= option.

TEXTATTRS=*style-element* | *style-element* (*text-options*) | (*text-options*)
specifies the font and color attributes of the *string* that is specified on the TEXT= option when TYPE= is set to TEXT. See "General Syntax for Attribute Options" on page 849 for the syntax on using a *style-element* and "Text Options" on page 852 for available *text-options*.

Default: The GraphValueText style element.

Part 7

Text Statements

Chapter 59

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 702).

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*.

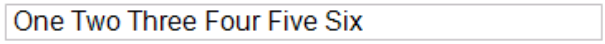
When used, a prefix option applies not only to immediately following piece of text but also to ALL subsequent text strings and text-commands. If the same prefix option appears more than once, each use overrides the last used prefix of the same name.

Managing Horizontal Alignment

For the ENTRY, ENTRYTITLE, and ENTRYFOOTNOTE statements, the default horizontal alignment is CENTER.


```
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:




```
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:

```
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:


```
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 849 for the syntax on using a *style-element* and "Text Options" on page 852 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:


```
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 704 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"};
```

Rules for Unicode and Special Character Specifications

The following rules apply to Unicode and special character specifications in ODS graphics:

- Each character can be specified by looking up its code and specifying it as a hexadecimal constant. Example: **{unicode '221e'x}**.
- Lower case Greek letters can be specified by using names instead of hexadecimal constants. Example: **{unicode alpha}**.
- Upper case Greek letters can be specified by using names followed by **_u** instead of a hexadecimal constants. Example: **{unicode alpha_u}**.
- Superscript and subscript have special abbreviations. Examples: **{sup 2}** and **{sub 2}**.
- The {SUP} and {SUB} specifications must not appear escaped and in quotes in the GTL. They must appear outside of quotes.
- Some characters overprint the character that comes before. Example: **'El nin' {tilde} 'o'**, which is equivalent to **'El nin' {unicode '0303'x} 'o' creates 'El niño'**.
- Specifications inside quotes are escaped. Example: **"(*ESC*){unicode beta}"**.
- Specifications outside quotes are not escaped. Example: **{unicode beta}**.

For more information about using text throughout the GTL (for example, using Unicode values in labels), see *SAS Graph Template Language: User's Guide*.

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 60

ENTRY Statement

Description and Syntax

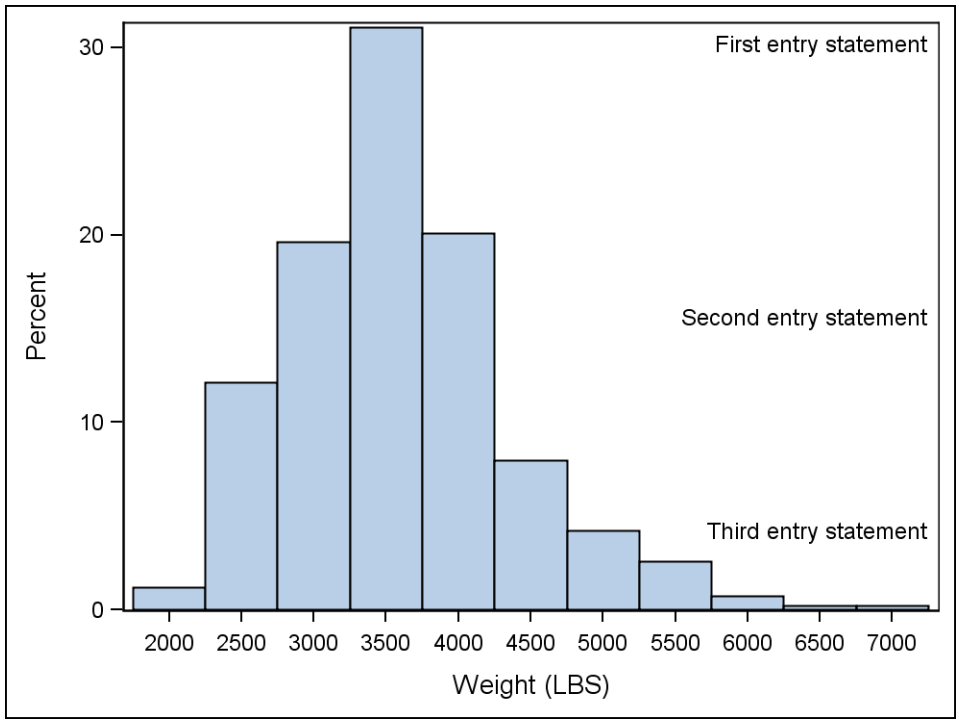
Displays a line of text in the plot area.

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 708:

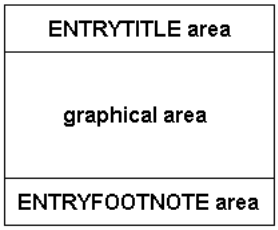


Example Program

```
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 709 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.

Restriction: The maximum length for the entry text is 512 characters.

See Chapter 59, "Managing Text Items," on page 699 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 or region 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 or region layout. For more information about how child positions are determined in an overlay-type or region layout, see the LAYOUT OVERLAY's "Statement Summary" on page 105 .

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 849 for the syntax on using a *style-element* and "Line Options" on page 850 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";

See Also: “Rules for Unicode and Special Character Specifications” on page 704 .

{ 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} ;

See Also: “Rules for Unicode and Special Character Specifications” on page 704 .

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 849 for the syntax on using a *style-element* and “Text Options” on page 852 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 see "Reserved Keywords and Unicode Values" on page 704..

dynamic
The dynamic must resolve to either *"hex-string"x* or a keyword for a UNICODE character.

Examples: The following statements show how to use the {UNICODE} text command:

```
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.

See Also: "Using UNICODE Text" on page 703 and "Rules for Unicode and Special Character Specifications" on page 704 .

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

Default: CENTER

Chapter 61

ENTRYFOOTNOTE Statement

Description and Syntax

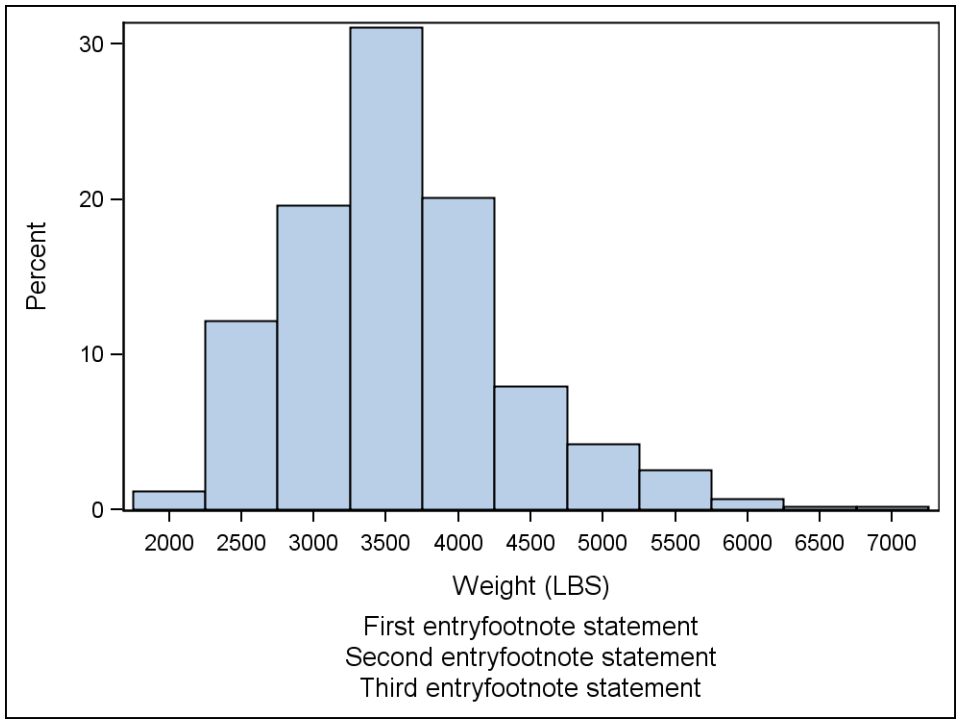
Displays a footnote.

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 716:

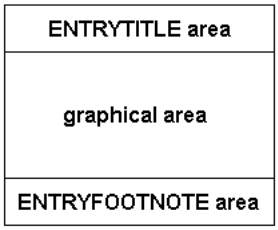


Example Program

```
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 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.

Restriction: The maximum length for the footnote text is 512 characters.

See Chapter 59, "Managing Text Items," on page 699 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 849 for the syntax on using a *style-element* and "Line Options" on page 850 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";

See Also: “Rules for Unicode and Special Character Specifications” on page 704 .

{ 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;

See Also: “Rules for Unicode and Special Character Specifications” on page 704 .

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 849 for the syntax on using a *style-element* and “Text Options” on page 852 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 704.

dynamic
The dynamic must resolve to either *"hex-string"x* or a keyword for a UNICODE character.

Examples: The following statements show how to use the {UNICODE} text command:

```
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.

See Also: "Using UNICODE Text" on page 703 and "Rules for Unicode and Special Character Specifications" on page 704 .

Chapter 62

ENTRYTITLE Statement

Description and Syntax

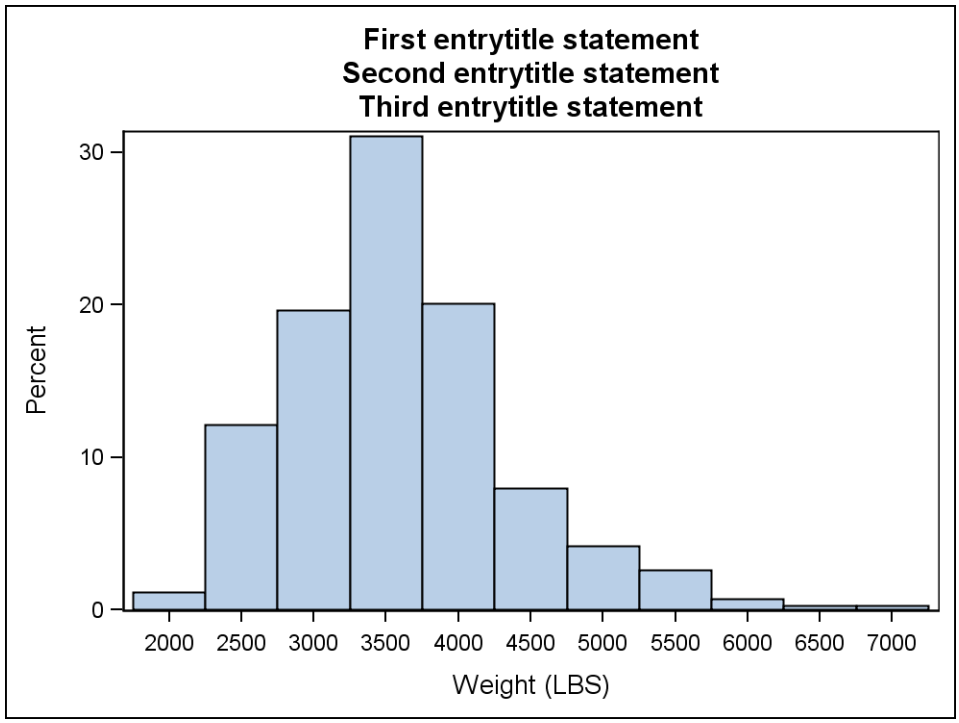
Displays a title.

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 724:

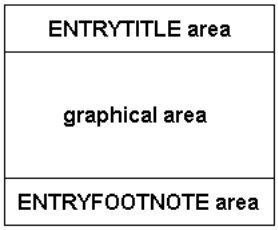


Example Program

```
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 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.

Restriction: The maximum length for the title text is 512 characters.

See Chapter 59, "Managing Text Items," on page 699 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 849 for the syntax on using a *style-element* and "Line Options" on page 850 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";

See Also: “Rules for Unicode and Special Character Specifications” on page 704 .

{ 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;

See Also: “Rules for Unicode and Special Character Specifications” on page 704 .

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 852 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 704.

dynamic
The dynamic must resolve to either *"hex-string"x* or a keyword for a UNICODE character.

Examples: The following statements show how to use the {UNICODE} text command:

```
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.

See Also: "Using UNICODE Text" on page 703 and "Rules for Unicode and Special Character Specifications" on page 704 .

Part 8

Draw Statements

Chapter 63

Key Concepts for Using Draw Statements

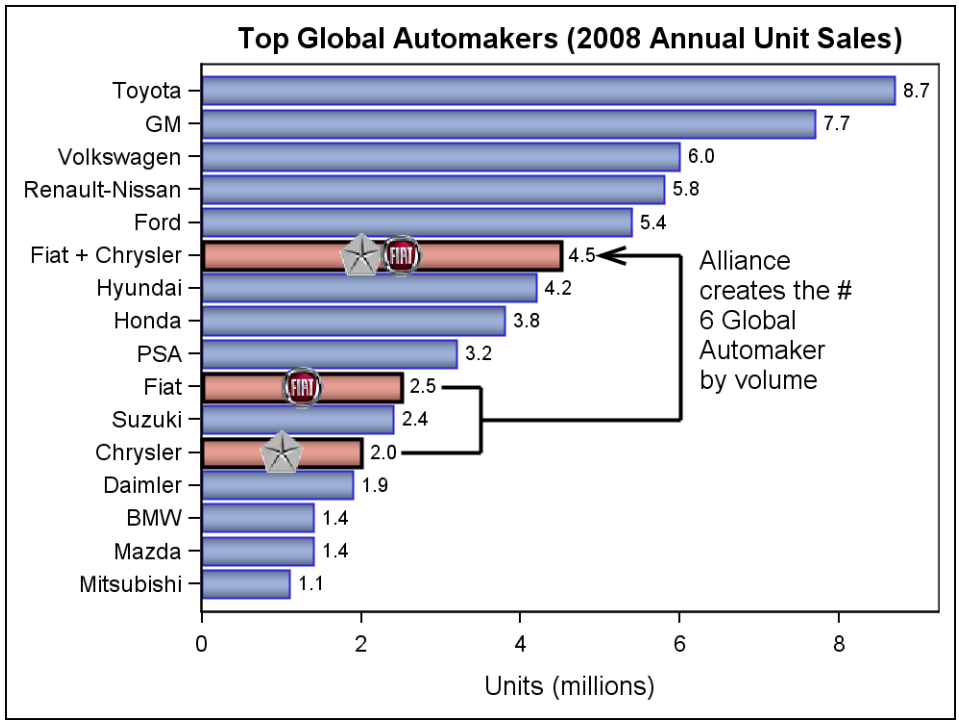
Introduction

The draw statements enable you to customize a graph by drawing visual elements anywhere within the graph. The following sections provide a comprehensive example and a brief overview to the major drawing concepts. For more details about the individual draw statements and a simple example of each, see the discussion for each individual statement. For complete usage information for the draw statements, consult the *SAS Graph Template Language: User's Guide*.

Example

The following example shows a bar chart of the top global automobile makers in 2008. To focus the graph on the 2008 merger between Chrysler and Fiat, the example uses DRAWRECTANGLE and DRAWIMAGE statements to highlight the three bars in the chart that correspond to the unit sales for the two auto makers. The example also uses a DRAWARROW statement and two BEGINPOLYLINE blocks to point to the sales figures, and it uses a DRAWTEXT statement to explain the implications of the merger.

All of the draw statements in this example use the data space (see "About the Drawing Space and Drawing Units" on page 736) to integrate the drawn elements into the graph's data area.




```
/* Create Data for Chrysler, Fiat and Chriysler+Fiat widths */
data mydata;
 length automaker $30;
 input automaker $ 1-30 million_units;
 FORMAT million_units 3.1;

 if index(automaker,'Chrysler')^=0 or index(automaker,'Fiat')^=0 then
    colorvar=1;
 else
   colorvar=2;

 if index(automaker,'Chrysler')^=0 and index(automaker,'Fiat')^=0 then
   do;
      call symput("cfwidth",  million_units);
   end;
 else if index(automaker,'Fiat')^=0 then
   do;
      call symput("fwidth",  million_units);
   end;
 else if  index(automaker,'Chrysler')^=0 then
   do;
      call symput("cwidth",  million_units);
   end;

datalines;
Toyota                          8.7
GM                              7.7
Volkswagen                      6.0
Renault-Nissan                  5.8
Ford                            5.4
Fiat + Chrysler                 4.5
Hyundai                         4.2
Honda                           3.8
```

```
PSA                          3.2
Fiat                         2.5
Suzuki                       2.4
Chrysler                     2.0
Daimler                      1.9
BMW                          1.4
   Mazda                        1.4
Mitsubishi                   1.1
;
/* Create template definition */
proc template;
 define statgraph automerger;
 begingraph /  drawspace=datavalue;
   entrytitle halign=center
     'Top Global Automakers (2008 Annual Unit Sales)';
   layout lattice / rowdatarange=data columndatarange=data
         rowgutter=10 columngutter=10;
     layout overlay / xaxisopts=(label=('Units (millions)'))
         yaxisopts=(reverse=true display=(ticks tickvalues line));
       barchart x=automaker y=million_units / group=colorvar
         name='bar(h)' barlabel=true dataskin=pressed orient=horizontal;
       drawrectangle x=eval(&cwidth/2.0) y="Chrysler"
         width=&cwidth height=0.85 / widthunit=data heightunit=data
         display=(outline) outlineattrs=(color=black) ;
       drawrectangle x=eval(&fwidth/2.0) y="Fiat"
         width=&fwidth height=0.85 /
         widthunit=data heightunit=data
         display=(outline) outlineattrs=(color=black) ;
       drawrectangle x=eval(&cfwidth/2.0) y="Fiat + Chrysler"
         width=&cfwidth height=0.85 / widthunit=data heightunit=data
         display=(outline) outlineattrs=(color=black) ;
       drawimage "C:\chrysler_logo.png" /
         x=eval(&cwidth/2.0) y="Chrysler" ;
       drawimage "C:\fiat_logo.png" /
         x=eval(&fwidth/2.0) y="Fiat" ;
       drawimage "C:\chrysler_logo.png" /
         x=eval(&cfwidth/2.0 - 0.25) y="Fiat + Chrysler" ;
       drawimage "C:\fiat_logo.png" /
         x=eval(&cfwidth/2.0 + 0.25) y="Fiat + Chrysler" ;
       beginpolyline x=eval(&cwidth + 0.5) y="Chrysler" ;
         draw x=eval(&cwidth + 1.5) y="Chrysler" ;
         draw x=eval(&cwidth + 1.5)  y="Fiat" ;
         draw x=eval(&cwidth + 0.5)  y="Fiat" ;
       endpolyline ;
       beginpolyline x=eval(&cwidth + 1.5)  y="Suzuki" ;
         draw x=eval(&cfwidth + 1.5)  y="Suzuki" ;
         draw x=eval(&cfwidth  + 1.5) y="Fiat + Chrysler" ;
       endpolyline ;
       drawarrow x1=eval(&cfwidth + 1.5) x2=eval(&cfwidth + 0.5)
         y1="Fiat + Chrysler" y2="Fiat + Chrysler" ;
       drawtext "Alliance creates the #6 Global Automaker by volume" /
         y="Honda" x=eval(&cfwidth+2.5) width=2 widthunit=data ;
     endlayout;
   endlayout;
 endgraph;
end;
```

```
proc sgrender data=mydata template=automerger;
run;
```

Types of Elements That Can Be Drawn

The GTL draw statements enable you to draw the following types of elements in a graph:

- text
- arrows and lines
- geometric shapes like ovals, rectangles, polygons, and polylines (a set of connecting lines)
- images

Using these elements individually and especially by combining them, you can describe the non-data aspects of your graph. For example, you can display a company logo in a specified location within the graph. Or you can create custom features that are difficult to create by other means—for example, you could draw a broken axis. Using the draw statements creatively, you can direct viewer attention to features of interest in the graph by drawing elements that highlight those features.

About the Drawing Space and Drawing Units

Each draw statement positions a drawn element using Cartesian coordinates that you specify as X and Y values in the statement. The coordinates that you specify are positioned relative to the *drawing space* that is in effect for the statement. The available drawing spaces are the *data area*, the *wall area*, the *layout area*, and the *graph area*. The coordinates can be specified in pixel units, percentage units, or as values that are in the units of the data (available only in the data area).

The images that accompany the following descriptions all use pixel data units. Each, however, was drawn in a different data space using this same LAYOUT OVERLAY definition block:

```
layout overlay /
    xaxisopts=(display=(line ticks tickvalues))
    yaxisopts=(display=(line ticks tickvalues));
  scatterplot x=height y=weight;
    drawrectangle x=0 y=0 width=50 height=50 /
      anchor=bottomleft display=(fill) fillattrs=(color=green)
      transparency=0.75 widthunit=pixel heightunit=pixel ;
    drawline x1=0 y1=0 x2=18 y2=18 / lineattrs=(color=red) ;
endlayout;
```

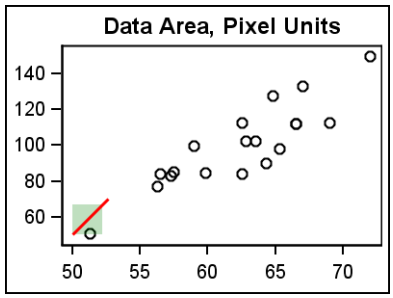
Drawing Space: Data Area
Drawing Units: VALUE, PERCENT, PIXEL

Drawing Space: Data Area

The area where data is displayed in the graph, honoring the offsets that are set for the axes. For a discussion on axis offsets, see "Adjusting Axis Offsets" on page 568.

The data area does not apply to graphs that do not have axes, such as pie charts, which must be drawn in a REGION layout.

For graphs produced within LAYOUTDATAPANEL and LAYOUTDATALATTICE layouts, drawn elements are clipped if they extend outside of the wall boundaries.



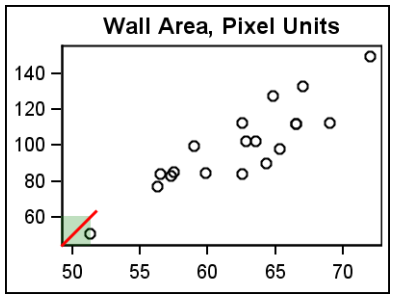
Drawing Space: Wall Area

Drawing Units: PERCENT, PIXEL

The area bounded by orthogonal axis pairs, ignoring the offsets that are set for the axes. In two-dimensional graphs, there is one wall bounded by the XY axes. including the secondary axes, if used. In three-dimensional graphs, there are three walls, bounded by the XY, YZ, and XZ axes.

The wall area does not apply to graphs that do not have axes, such as pie charts, which must be drawn in a REGION layout.

For graphs produced within LAYOUTDATAPANEL and LAYOUTDATALATTICE layouts, drawn elements that extend outside of the wall boundaries are clipped.

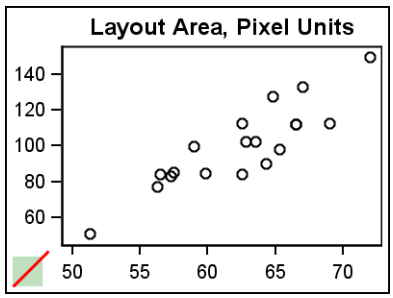


Drawing Space: Layout Area

Drawing Units: PERCENT, PIXEL

The entire area of the layout container that is the immediate parent container of the draw statement. The figure to the right shows the case where a LAYOUT OVERLAY is the draw statement's layout container.

Titles and footnotes are always displayed outside of the outermost layout, so those areas are never part of the layout drawing space.



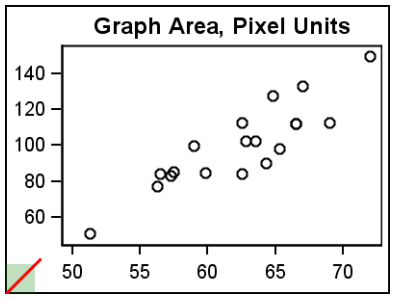
Drawing Space: Graph Area

Drawing Units: PERCENT, PIXEL

Drawing Space: Graph Area

The entire area that is available to the graph display, whether a single-cell or multi-cell graph.

Because the graph drawing space spans the entire graph, the location of the drawn element in the graph is independent of the draw statement's placement within the template definition, even if the draw statement is specified within a nested layout.



To specify the drawing space for both the X and the Y dimension, you use the DRAWSPACE= option. To specify the drawing space individually for either the X or the Y dimension, you use the options XSPACE=, X1SPACE=, X2SPACE=, YSPACE=, Y1SPACE=, or Y2SPACE=, depending on the draw statement. The value that you set on any of these options is a single composite value that specifies both the drawing space and the drawing units in the following format:

```
<DrawingSpace><Units>
```

For example, DRAWSPACE=GRAPHPIXEL specifies the GRAPH drawing space with PIXEL drawing units, indicating that the statement's X,Y coordinates are expressed in pixels. Similarly, DRAWSPACE=LAYOUTPERCENT specifies the LAYOUT drawing space with PERCENT drawing units, indicating that the statement's X,Y coordinates are expressed as percentages.

The global DRAWSPACE is LAYOUTPERCENT. The global DRAWSPACE= setting for all of the draw statements is LAYOUTPERCENT. The draw statements inherit the global setting from the DRAWSPACE= setting in the BEGINGRAPH statement.

- To change the global drawing space and drawing units for all of the draw statements within the template definition, use the DRAWSPACE= option in the BEGINGRAPH statement.
- To change the default for an individual draw statement, use that statement's DRAWSPACE= option. If needed, you can specify different settings for the different draw statements within the template definition.

For the DATA drawing space, the VALUE drawing units specify that the coordinates are expressed as values along the axis. When you specify the DATA drawing space, you can use the draw statement's XAXIS= and YAXIS= options to specify which axis scale to use for the coordinates.

- If the specified axis does not exist in the plot or is not valid for the draw statement's layout container, the draw statement is ignored.
- For a discrete axis, if the statement's specified X or Y value does not exist in the data, the draw statement is ignored.
- For a continuous axis, if the statement's specified X or Y value does not exist in the data, the value is extrapolated.

When specifying the drawing space and drawing units, you can set a common setting for all of the X and Y coordinates. Or you can specify different settings for each individual coordinate. The DRAWSPACE= setting in the BEGINGRAPH statement applies the global space and unit settings to all of the draw statements within the BEGINGRAPH/ ENDGRAPH block. The DRAWSPACE= setting in an individual draw statement

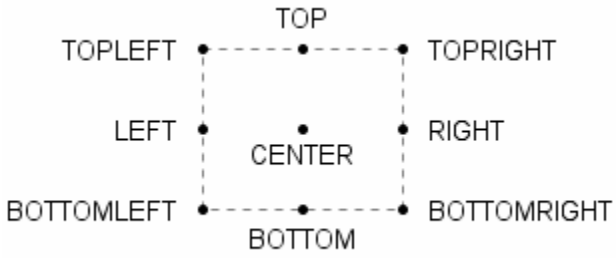
applies the space and unit setting only to the coordinate(s) for that statement. Thus, for lines and arrows, the setting applies to both the X1, Y1 coordinate and the X2, Y2 coordinate.

To specify the drawing space and drawing units separately for the X coordinate and for the Y coordinate, use the XSPACE=, YSPACE=, X1SPACE=, Y1SPACE=, X2SPACE=, and Y2SPACE= options, as applicable, in each draw statement. These options override the DRAWSPACE= option.

How the Drawn Elements Are Anchored

When you specify the X and Y coordinates in a draw statement, the element is drawn from an anchor point that is placed in the drawing area at the X,Y coordinate point.

- For lines and arrows, the anchor point is the starting point of the line or arrow, which is specified with the draw statement's X1= and Y1= values.
- For elements that have height and width, the anchor point can be one of the following points:



The default anchor position is CENTER. To change the anchor position, use the draw statement's ANCHOR= option.

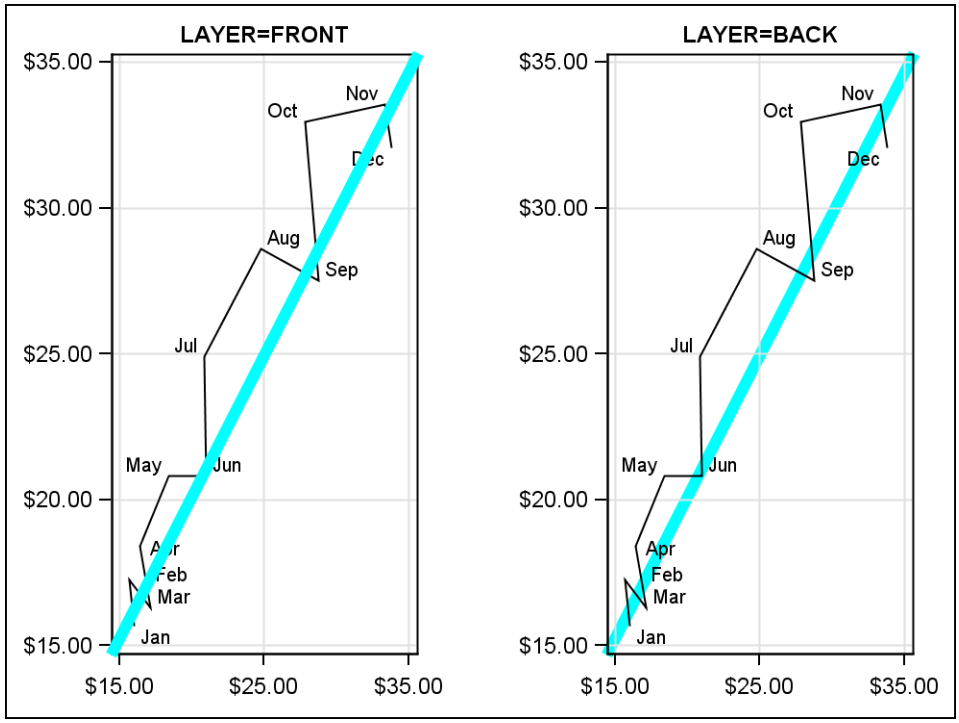
About Drawing Layers

A draw statement can draw its element in either of two "layers" in the graph: the front layer or the back layer. By default, the statement draws in the front layer, which places the element in front of all other graphics elements, including data points, data labels, axis labels, and so on. In some cases, this might cause the drawn element to block the view of other graphics elements in the graph.

To prevent a drawn element for blocking the view of other graphics elements in the graph, you can use the draw statement's TRANSPARENCY= option to add transparency to the drawn element. With an appropriate transparency setting, you should be able to see any graphics elements that are behind the drawn element.

Alternatively, you can use the draw statement's LAYER= option to draw the element in the back layer, which places the drawn element behind all other graphics elements in the graph. For example, the following figure shows two different versions of a graph that uses DRAWLINE to draw a diagonal line across the axis area. The version to the left draws the line in the front layer, which causes the line to cover some of the data labels in the graph and portions of the series line. The version to the right uses LAYER=BACK to

draw the line in the back layer. This prevents the line from covering the data labels and the portions of the series line.



The following code fragment shows the code that positions the line in the back layer:

```
layout overlay / walldisplay=(outline)
    xaxisopts=(griddisplay=on display=(line ticks tickvalues))
    yaxisopts=(griddisplay=on display=(line ticks tickvalues));
  seriesplot x=open y=close / datalabel=date;
  drawline x1=0 y1=0 x2=100 y2=100 /
    x1space=wallpercent y1space=wallpercent
    x2space=wallpercent y2space=wallpercent
    lineattrs=(color=cyan thickness=6) layer=back ;
endlayout;
```

The Back Layer is Behind the Background. Although drawing elements in the back layer prevents them from obstructing other data elements in the graph, it is not always the right solution to the problem.

If a draw statement uses LAYER=BACK, it draws the element behind all other graphics elements, such as the layout background or a discrete legend's background. To ensure that the element is visible in the graph, you might have to do one or more of the following:

- In overlay-type layouts or in a SCATTERPLOTMATRIX, you can use the parent layout's WALLDISPLAY= option to turn off the display of the plot wall. In the example code just shown, WALLDISPLAY=(OUTLINE) displays an outline around the graph wall but does not display the wall fill. Suppressing the fill ensures that the drawn line is visible behind the plot wall.
- If a layout container uses OPAQUE=TRUE so that it can set visual attributes for the background, the opaque background covers and therefore hides any drawn element that is behind the background. When assigning visual attributes to a graph background, therefore, it might be better to use TRANSPARENCY= rather than

LAYER= to prevent drawn elements from covering other graphics elements in the graph.

- If the results of a plot statement or other *GTL-statement* covers the drawn element, you can use transparency to reveal the drawn element. For example, you could use a plot statement's DATATRANSPARENCY= option to set an appropriate transparency level for the plot.

Chapter 64

BEGINPOLYGON Statement

Description and Syntax

Defines the starting point for drawing a polygon in the graph.

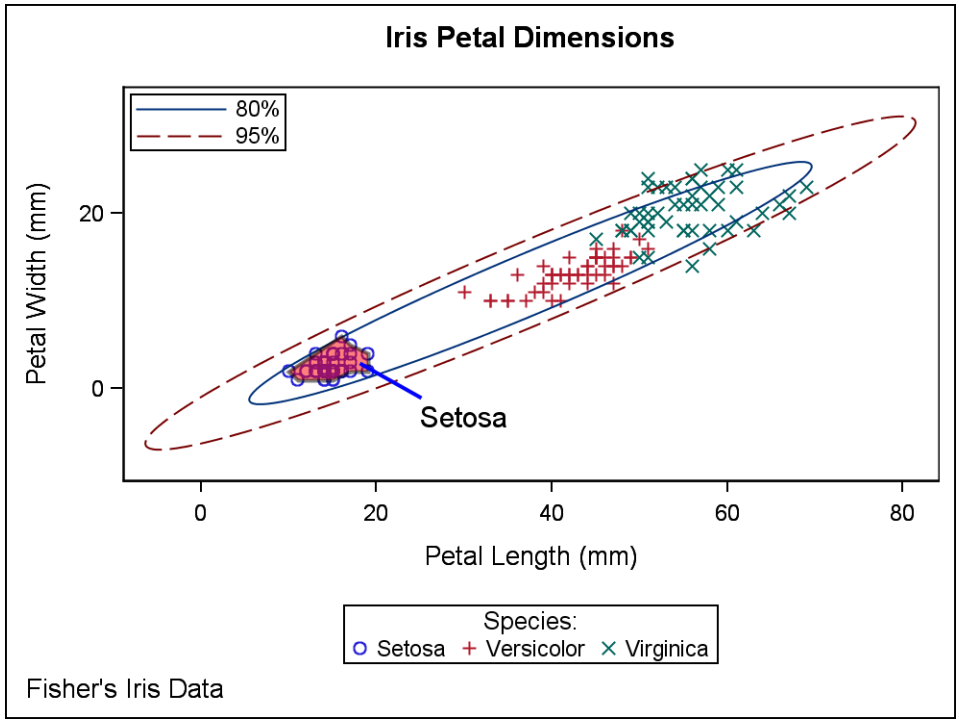
BEGINPOLYGON X=*constant | scalar-expression*
Y=*constant | scalar-expression* </*option(s)*>;
DRAW statement;
<…*DRAW statement*;>
ENDPOLYGON;

Example Program and Statement Details

Example Graph

The following graph was generated by the "Example Program" on page 744. The example uses a BEGINPOLYGON block to draw a polygon around the Setosa species of Iris in a plot that shows petal sizes for three Iris species. The DRAWSPACE= option in the BEGINGRAPH statements specifies that the polygon be drawn in the data space. The BEGINPOLYGON statement specifies the starting X,Y point. For the BEGINPOLYGON options, DISPLAY= displays both a fill and an outline for the polygon. TRANSPARENCY= adds a degree of transparency to the fill and outline, and

FILLATTRS= colors the fill red. The example also uses DRAWLINE and DRAWTEXT statements to draw a callout to the polygon.



Example Program

```
proc template;
  define statgraph discretelegend;
    begingraph / drawspace=datavalue;
      entrytitle "Iris Petal Dimensions";
      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;

        beginpolygon x=10 y=2 /display=all
            transparency=0.5 fillAttrs=(color=red) ;
          draw x=13 y=4 ;
          draw x=16 y=6 ;
          draw x=17 y=5 ;
          draw x=19 y=4 ;
          draw x=19 y=2 ;
          draw x=17 y=2 ;
          draw x=15 y=1 ;
          draw x=14 y=1 ;
          draw x=11 y=1 ;
        endpolygon ;
        drawtext textattrs=(size=11pt) "Setosa" /
          x=30 y=-1 width=12 anchor=top;
        drawline x1=18 y1=3 x2=25 y2=-1 / lineattrs=(color=blue);
```

```
        discretelegend "s" / title="Species:";
        discretelegend "p80" "p95" /across=1 autoalign=(topleft) location=inside;
      endlayout;
      entryfootnote halign=left "Fisher's Iris Data" ;
    endgraph;
  end;

proc sgrender data=sashelp.iris template=discretelegend;
run;
```

Statement Summary

A polygon is built by using the BEGINPOLYGON statement to specify the polygon's starting X,Y coordinate, and then specifying the remaining points by nesting a series of DRAW statements (see "DRAW Statement" on page 748) within the BEGINPOLYGON block. The block is closed with an ENDPOLYGON statement. To manage the location and drawing units for the polygon, you can use the XAXIS=, YAXIS=, XSPACE=, and YSPACE= options.

For general information about the types of elements that can be drawn with the draw statements, the drawing space and drawing units that they use, and how the drawn elements are anchored, see Chapter 63, "Key Concepts for Using Draw Statements," on page 733 . For detailed usage information, consult the *SAS Graph Template Language: User's Guide*.

Required Arguments

X=*constant* | *scalar-expression*
specifies the X value for the starting point in the polygon.

Interaction: The value that is set for this argument is interpreted using the XSPACE= option. When XSPACE=DATAVALUE, the value is interpreted using the XAXIS= option.

Y=*constant* | *scalar-expression*
specifies the Y value for the starting point in the polygon.

Interaction: The value that is set for this argument is interpreted using the YSPACE= option. When YSPACE=DATAVALUE, the value is interpreted using the YAXIS= option.

Options

Statement Option	Description
DISCRETEOFFSET	Specifies an amount to offset the polygon from discrete X values, or discrete Y values, or both.
DISPLAY	Specifies the features to display for the polygon.
DRAWSPACE	Specifies a default drawing space and drawing units for this BEGINPOLYGON statement.

Statement Option	Description
FILLATTRS	Specifies the properties of the polygon's filled area.
LAYER	Specifies whether the polygon appears on top of or behind the graph.
OUTLINEATTRS	Specifies the properties of the outline that is drawn around the polygon.
TRANSPARENCY	Specifies the degree of the transparency of the fill and the outline.
XAXIS	Specifies whether the X value is interpreted using the primary X axis scale or to the secondary X (X2) axis scale.
XSPACE	Specifies the drawing space and drawing units for interpreting the X value.
YAXIS	Specifies whether the Y value is interpreted using the primary Y axis scale or to the secondary Y (Y2) axis scale.
YSPACE	Specifies the drawing space and drawing units for interpreting the Y value.

DISCRETEOFFSET= *number*
specifies an amount to offset the polygon from discrete X values, or discrete Y values, or both.

Default: 0 (no offset, the polygon is centered on discrete X values, or discrete Y values, or both)

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.

Interaction: This option applies only when the options XSPACE= or YSPACE= use DATAVALUE, and when X or Y are values on a discrete axis.

DISPLAY=STANDARD | ALL | (*display-options*)
specifies the features to display for the polygon.

Default: STANDARD

STANDARD
Displays an outlined polygon.

ALL
Displays an outlined, filled polygon.

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

OUTLINE—displays an outlined polygon

FILL—displays a filled polygon

Tip: Use the OUTLINEATTRS= and FILLATTRS= options to control the appearance of the polygon.

DRAWSPACE= GRAPHPERCENT | GRAPHPIXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALUE
specifies a default drawing space and drawing units for this BEGINPOLYGON block.

Default: LAYOUTPERCENT

Interaction: This statement and all of the draw statements inherit the global DRAWPSACE= setting from the DRAWSPACE= option in the BEGINGRAPH statement. Setting this option changes the setting for only this BEGINPOLYGON statement.

Interaction: This option sets the default drawing space, but individual settings in the X or Y dimension can be overridden by the XSPACE= and YSPACE= options.

See Also: "About the Drawing Space and Drawing Units" on page 736 .

FILLATTRS=*style-element* | *style-element* (*fill-options*) | (*fill-options*)
specifies the properties of the polygon's filled area. See "General Syntax for Attribute Options" on page 849 for the syntax on using a *style-element* and "Fill Options" on page 850 for available *fill-options*.

Default: The GraphAnnoShape style element

Tip: The TRANSPARENCY= option sets the transparency for the fill and the outline. You can combine this option with TRANSPARENCY= to set one transparency for the outlines but a different transparency for the fill. Example:

```
transparency=0.2 fillattrs=(transparency=0.6)
```

LAYER=FRONT | BACK
specifies whether the polygon appears on top of (FRONT) or behind (BACK) the graph.

Default: FRONT

Interaction: If this option is set to BACK, the polygon is drawn behind background areas, such as a layout or legend background. For elements that are obstructed because they are in the back layer, you can suppress the display of filled areas in the graph. Or, you can use transparency to manage the element visibility. For more information, see "About Drawing Layers" on page 739 .

OUTLINEATTRS= *style-element* | *style-element* (*line-options*) | (*line-options*)
specifies the properties of the outline that is drawn around the polygon. See "General Syntax for Attribute Options" on page 849 for the syntax on using a *style-element* and "Line Options" on page 850 for available *line-options*.

Default: The GraphAnnoShape style element.

Interaction: For this option to have any effect, the outline must be enabled by the DISPLAY= option.

TRANSPARENCY= *number*
specifies the degree of the transparency of the fill and the outline.

Default: 0

Range: 0 (opaque) to 1 (entirely transparent)

Tip: The FILLATTRS option can be used to set transparency for just the polygon's filled area. You can combine this option with FILLATTRS= to set one transparency for the outlines but a different transparency for the fill. Example:

```
transparency=0.2 fillattrs=(transparency=0.6)
```

XAXIS= X | X2
specifies whether the X value is interpreted using the primary X axis scale or to the secondary X (X2) axis scale.

Default: X

Interaction: This option has effect only if XSPACE=DATAVALUE.

XSPACE= GRAPHPERCENT | GRAPHPIXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALUE
specifies the drawing space and drawing units for interpreting the X value.

Default: The setting that is in effect for the DRAWSPACE= option.

YAXIS= Y | Y2
specifies whether the Y value is interpreted using the primary Y axis scale or to the secondary Y (Y2) axis scale.

Default: Y

Interaction: This option has effect only if YSPACE=DATAVALUE.

YSPACE= GRAPHPERCENT | GRAPHPIXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALUE
specifies the drawing space and drawing units for interpreting the Y value.

Default: The setting that is in effect for the DRAWSPACE= option.

DRAW Statement

DRAW Statement Summary

The nested DRAW statements within a BEGINPOLYGON block specify a series of points for a polygon. Each DRAW statement draws a straight line from the previous point to the endpoint that is specified in the DRAW statement's X and Y arguments. The first DRAW statement starts its line from the X,Y point that is specified in the BEGINPOLYGON statement.

You can specify as many DRAW statements as needed to complete the polygon. If the last DRAW statement does not end at the first point in the polygon (specified in BEGINPOLYGON), then a line is automatically generated to close the polygon shape.

Each DRAW statement uses the following general syntax:

DRAW X=*constant* | *scalar-expression*
Y=*constant* | *scalar-expression*</*option(s)*>;

For a specific example, see the "Example Program" on page 744.

DRAW Statement Arguments

X=*constant* | *scalar-expression*
specifies the X value for one point in a polygon.

Interaction: The value that is set for this argument is interpreted using the XSPACE= option. When XSPACE=DATAVALUE, the value is interpreted using the BEGINPOLYGON statement's XAXIS= option.

Y=*constant* | *scalar-expression*
specifies the Y value for one point in a polygon.

Interaction: The value that is set for this argument is interpreted using the YSPACE= option. When YSPACE=DATAVALUE, the value is interpreted using the BEGINPOLYGON statement's YAXIS= option.

DRAW Statement Options

Statement Option	Description
DRAWSPACE	Specifies a default drawing space and drawing units for the drawn lines.
XSPACE	Specifies the drawing space and drawing units for interpreting the X value.
YSPACE	Specifies the drawing space and drawing units for interpreting the Y value.

Chapter 65

BEGINPOLYLINE Statement

Description and Syntax

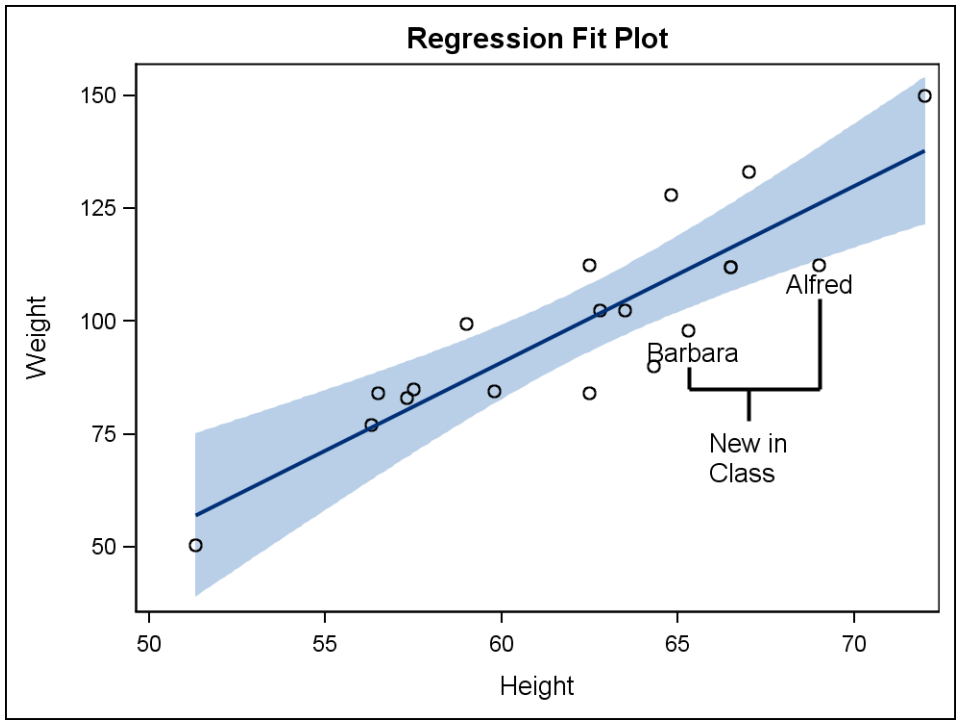
Defines the starting point for drawing a polyline in the graph.

BEGINPOLYLINE X=*constant | scalar-expression*
Y=*constant | scalar-expression* <*/option(s)*>;
DRAW statement;
<*…DRAW statement*;>
ENDPOLYLINE;

Example Program and Statement Details

Example Graph

The following graph was generated by the "Example Program" on page 752. The example uses two BEGINPOLYLINE blocks to highlight two student data points. The DRAWSPACE= option in the BEGINGRAPH statements specifies that the polyline be drawn in the data space. The BEGINPOLYLINE statements specify the starting X,Y points for two polyline, and the DRAW statements complete the lines. The example also uses two DRAWTEXT statements to label the data points of interest.



Example Program

```
proc template;
  define statgraph drawoval;
    begingraph / drawspace=datavalue;
      entrytitle "Regression Fit Plot";
      layout overlay;
        modelband "myclm";
        scatterplot x=height y=weight;
        regressionplot x=height y=weight / alpha=.01 clm="myclm";
          drawtext "Alfred" / x=69 y=112 anchor=top;
          drawtext "Barbara" / x=65.4 y=97 anchor=top width=15;
          beginpolyline x=69 y=105 ;
            draw x=69 y=85 ;
            draw x=65.3 y=85 ;
            draw x=65.3 y=90 ;
          endpolyline ;
          beginpolyline x=67 y=85 ;
            draw x=67 y=78 ;
          endpolyline ;
          drawtext "New in Class" / x=67 y=77 anchor=top width=15;
      endlayout;
    endgraph;
  end;

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

Statement Summary

A polyline is built by using the BEGINPOLYLINE statement to specify the polyline's starting X,Y coordinate, and then specifying the remaining points by nesting a series of DRAW statements (see "DRAW Statement" on page 755) within the BEGINPOLYLINE block. The block is closed with an ENDPOLYLINE statement. To manage the location and drawing units for the polyline, you can use the XAXIS=, YAXIS=, XSPACE=, and YSPACE= options.

For general information about the types of elements that can be drawn with the draw statements, the drawing space and drawing units that they use, and how the drawn elements are anchored, see Chapter 63, "Key Concepts for Using Draw Statements," on page 733 . For detailed usage information, consult the *SAS Graph Template Language: User's Guide*.

Required Arguments

X=*constant* | *scalar-expression*
specifies the X value for the starting point in the polyline.

Interaction: The value that is set for this argument is interpreted using the XSPACE= option. When XSPACE=DATAVALUE, the value is interpreted using the XAXIS= option.

Y=*constant* | *scalar-expression*
specifies the Y value for the starting point in the polyline.

Interaction: The value that is set for this argument is interpreted using the YSPACE= option. When YSPACE=DATAVALUE, the value is interpreted using the YAXIS= option.

Options

Statement Option	Description
DISCRETEOFFSET	Specifies an amount to offset the polyline from discrete X values, or discrete Y values, or both.
DRAWSPACE	Specifies a default drawing space and drawing units for this BEGINPOLYLINE statement.
LAYER	Specifies whether the polyline appears on top of or behind the graph.
LINEATTRS	Specifies the properties of the line segments for the polyline.
TRANSPARENCY	Specifies the degree of the transparency of the polyline.
XAXIS	Specifies whether the X value is interpreted using the primary X axis scale or to the secondary X (X2) axis scale.

Statement Option	Description
XSPACE	Specifies the drawing space and drawing units for interpreting the X value.
YAXIS	Specifies whether the Y value is interpreted using the primary Y axis scale or to the secondary Y (Y2) axis scale.
YSPACE	Specifies the drawing space and drawing units for interpreting the Y value.

DISCRETEOFFSET= *number*
specifies an amount to offset the polyline from discrete X values, or discrete Y values, or both.

Default: 0 (no offset, the polyline is centered on discrete X values, or discrete Y values, or both)

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.

Interaction: This option applies only when the options XSPACE= or YSPACE= use DATAVALUE, and when X or Y are values on a discrete axis.

DRAWSPACE= GRAPHPERCENT | GRAPHPIXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALUE
specifies a default drawing space and drawing units for this BEGINPOLYLINE block.

Default: LAYOUTPERCENT

Interaction: This statement and all of the draw statements inherit the global DRAWPSACE= setting from the DRAWSPACE= option in the BEGINGRAPH statement. Setting this option changes the setting for only this BEGINPOLYLINE statement.

Interaction: This option sets the default drawing space, but individual settings in the X or Y dimension can be overridden by the XSPACE= and YSPACE= options.

See Also: “About the Drawing Space and Drawing Units” on page 736 .

LAYER=FRONT | BACK
specifies whether the polyline appears on top of (FRONT) or behind (BACK) the graph.

Default: FRONT

Interaction: If this option is set to BACK, the polyline is drawn behind background areas, such as a layout or legend background.For elements that are obstructed because they are in the back layer, you can suppress the display of filled areas in the graph. Or, you can use transparency to manage the element visibility. For more information, see “About Drawing Layers” on page 739 .

LINEATTRS= *style-element* | *style-element* (*line-options*) | (*line-options*)
specifies the properties of the line segments for the polyline. See “General Syntax for Attribute Options” on page 849 for the syntax on using a *style-element* and “Line Options” on page 850 for available *line-options*.

Default: The GraphAnnoLine style element.

TRANSPARENCY= *number*
specifies the degree of the transparency of the polyline.

Default: 0

Range: 0 (opaque) to 1 (entirely transparent)

XAXIS= X | X2
specifies whether the X value is interpreted using the primary X axis scale or to the secondary X (X2) axis scale.

Default: X

Interaction: This option has effect only if XSPACE=DATAVALUE.

XSPACE= GRAPHPERCENT | GRAPHPIXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALUE
specifies the drawing space and drawing units for interpreting the X value.

Default: The setting that is in effect for the DRAWSPACE= option.

YAXIS= Y | Y2
specifies whether the Y value is interpreted using the primary Y axis scale or to the secondary Y (Y2) axis scale.

Default: Y

Interaction: This option has effect only if YSPACE=DATAVALUE.

YSPACE= GRAPHPERCENT | GRAPHPIXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALUE
specifies the drawing space and drawing units for interpreting the Y value.

Default: The setting that is in effect for the DRAWSPACE= option.

DRAW Statement

DRAW Statement Summary

The nested DRAW statements within a BEGINPOLYLINE block specify a series of points for a polyline. Each DRAW statement draws a straight line from the previous point to the endpoint that is specified in the DRAW statement's X and Y arguments. The first DRAW statement starts its line from the X,Y point that is specified in the BEGINPOLYLINE statement. You can specify as many DRAW statements as needed to complete the polyline.

Each DRAW statement uses the following general syntax:

DRAW X=*constant* | *scalar-expression*
Y=*constant* | *scalar-expression*</*option(s)*>;

For a specific example, see the "Example Program" on page 752.

DRAW Statement Arguments

X=*constant* | *scalar-expression*
specifies the X value for one point in a polyline.

Interaction: The value that is set for this argument is interpreted using the XSPACE= option. When XSPACE=DATAVALUE, the value is interpreted using the BEGINPOLYLINE statement's XAXIS= option.

Y=*constant* | *scalar-expression*
specifies the Y value for one point in a polyline.

Interaction: The value that is set for this argument is interpreted using the YSPACE= option. When YSPACE=DATAVALUE, the value is interpreted using the BEGINPOLYLINE statement's YAXIS= option.

DRAW Statement Options

Statement Option	Description
DRAWSPACE	Specifies a default drawing space and drawing units for the drawn lines.
XSPACE	Specifies the drawing space and drawing units for interpreting the X value.
YSPACE	Specifies the drawing space and drawing units for interpreting the Y value.

Chapter 66

DRAWARROW Statement

Description and Syntax

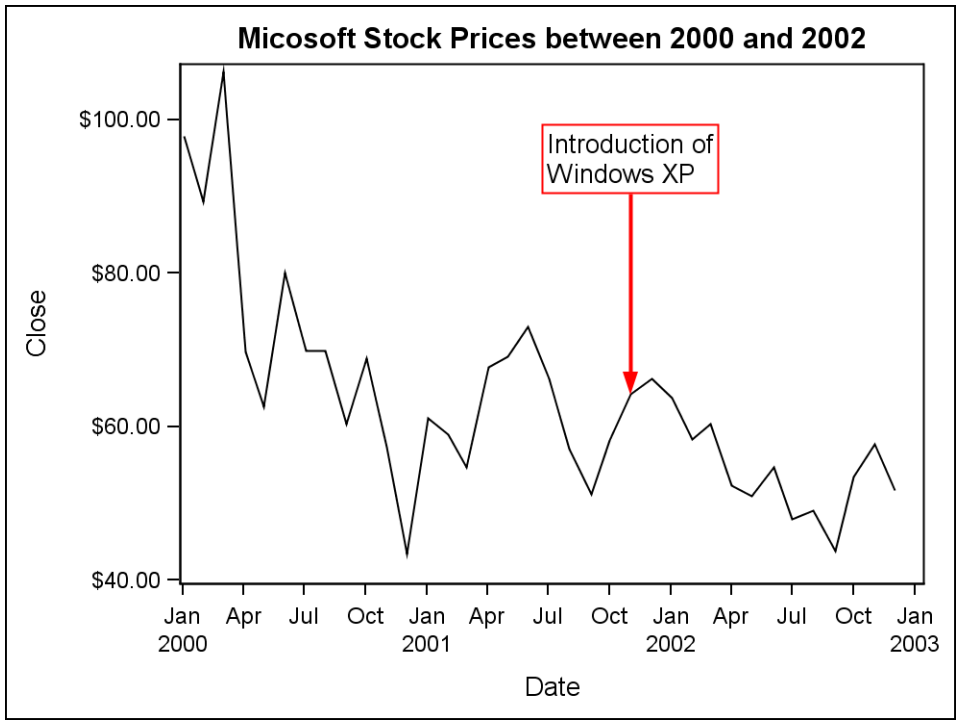
Draws an arrow (a directed line segment) from one point to another point.

DRAWARROW X1=*constant* | *scalar-expression*
Y1=*constant* | *scalar-expression*
X2=*constant* | *scalar-expression*
Y2=*constant* | *scalar-expression* </*option(s)*>;

Example Program and Statement Details

Example Graph

The following graph was generated by the "Example Program" on page 758. The example shows a common application of a DRAWARROW and DRAWTEXT statements to identify a specific part of the graph and add explanatory text.



Example Program

```
proc template;
  define statgraph arrow;
    begingraph;
      entrytitle "Micosoft Stock Prices between 2000 and 2002";
      layout overlay;
        seriesplot x=date y=close;
        drawarrow x1="01NOV2001"d  y1=75 x2="01NOV2001"d y2=64.21 /
           x1space=datavalue y1space=wallpercent
           x2space=datavalue y2space=datavalue
           arrowheadshape=filled lineattrs=(color=red) ;
        drawtext "Introduction of Windows XP" / width=25 anchor=bottom
           border=true borderattrs=(color=red)
           x="01NOV2001"d y=75 xspace=datavalue yspace=wallpercent;
      endlayout;
    endgraph;
  end;

proc sgrender data=sashelp.stocks template=arrow;
  where stock="Microsoft" and Year(date) between 2000 and 2002;
run;
```

Statement Summary

A DRAWARROW statement draws a line (arrow shaft) from a specified starting point (X1,Y1) to a specified ending point (X2,Y2). It also displays an arrowhead at either or both ends of the line. DRAWARROW is similar to a DRAWLINE statement, using many of the same options, but it has additional options for controlling the arrowhead(s).

For general information about the types of elements that can be drawn with the draw statements, the drawing space and drawing units that they use, and how the drawn elements are anchored, see Chapter 63, "Key Concepts for Using Draw Statements," on page 733 . For detailed usage information, consult the *SAS Graph Template Language: User's Guide*.

Required Argument

X1=*constant* | *scalar-expression*
specifies the X value of one arrow-shaft endpoint.

Interaction: This value that is set for this option is interpreted using the X1SPACE= option. When X1SPACE=DATAVALUE, the value is interpreted using the XAXIS= option.

Y1=*constant* | *scalar-expression*
specifies the Y value of one arrow-shaft endpoint.

Interaction: This value that is set for this option is interpreted using the Y1SPACE= option. When Y1SPACE=DATAVALUE, the value is interpreted using the YAXIS= option.

X2=*constant* | *scalar-expression*
specifies the X value of one arrow-shaft endpoint.

Interaction: This value that is set for this option is interpreted using the X2SPACE= option. When X2SPACE=DATAVALUE, the value is interpreted using the XAXIS= option.

Y2=*constant* | *scalar-expression*
specifies the Y value of one arrow-shaft endpoint.

Interaction: This value that is set for this option is interpreted using the Y2SPACE= option. When Y2SPACE=DATAVALUE, the value is interpreted using the YAXIS= option.

Options

Statement Option	Description
ARROWHEADDIRECTION	Specifies the direction of the arrowhead(s) at the end(s) of the arrow shaft.
ARROWHEADSHAPE	Specifies the shape of the arrowhead(s).
ARROWHEADSCALE	Specifies an arrowhead scaling factor based on the thickness of the arrow shaft.
DISCRETEOFFSET	Specifies an amount to offset the arrow from discrete X values, or discrete Y values, or both.
DRAWSPACE	Specifies a default drawing space and drawing units for this DRAWARROW statement.
LAYER	Specifies whether the arrow appears on top of or behind the graph.

Statement Option	Description
LINEATTRS	Specifies the appearance of the arrow shaft and arrowhead(s).
TRANSPARENCY	Specifies the degree of the transparency of the arrow shaft and arrowhead(s).
XAXIS	Specifies whether the data value for the arguments X1 and X2 are interpreted using the primary X axis scale or to the secondary X (X2) axis scale.
X1SPACE	Specifies the drawing space and drawing units for interpreting the X1 value
X2SPACE	Specifies the drawing space and drawing units for interpreting the X2 value
YAXIS	Specifies whether the data value for the arguments Y1 and Y2 are interpreted using the primary Y axis scale or to the secondary Y (Y2) axis scale.
Y1SPACE	Specifies the drawing space and drawing units for interpreting the Y1 value
Y2SPACE	Specifies the drawing space and drawing units for interpreting the Y2 value

ARROWHEADDIRECTION= OUT | IN | BOTH
specifies the direction of the arrowhead(s) at the end(s) of the arrow shaft.

OUT specifies a single arrowhead drawn at (X2,Y2) and pointing away from (X1,Y1)

IN specifies a single arrowhead drawn at (X1,Y1) and pointing away from (X2,Y2)

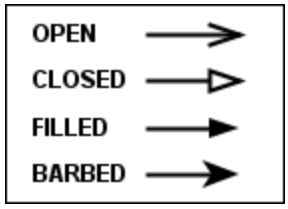
BOTH specifies two arrowheads, one at the IN position and one at the OUT position

Default: OUT

Tip: Use the ARROWHEADSHAPE= option to control the arrowhead appearance.

ARROWHEADSHAPE= OPEN | CLOSED | FILLED | BARBED
specifies the shape of the arrowhead(s).

Default: OPEN



Tip: Use the ARROWHEADDIRECTION= option to control the arrow direction.

ARROWHEADSCALE= *positive-number*
specifies an arrowhead scaling factor based on the thickness of the arrow shaft. Use a factor larger than 1.0 to make a larger arrowhead.

Default: 1

Range: 0.5 to 2

DISCRETEOFFSET= *number*
specifies an amount to offset the arrow from discrete X values, or discrete Y values, or both.

Default: 0 (no offset, all arrows are centered on discrete X values, or discrete Y values, or both)

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.

Interaction: This option applies only when the options X1SPACE=, X2SPACE=, Y1SPACE=, or Y2SPACE= use DATAVALUE, and when X1, X2, Y1, or Y2 are values on a discrete axis.

DRAWSPACE= GRAPHPERCENT | GRAPHPIXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALUE
specifies a default drawing space and drawing units for this DRAWARROW statement.

Default: LAYOUTPERCENT

Interaction: This statement and all of the draw statements inherit the global DRAWPSACE= setting from the DRAWSPACE= option in the BEGINGRAPH statement. Setting this option changes the setting for only this DRAWARROW statement.

Interaction: This option sets the default drawing space, but individual settings in the X or Y dimension can be overridden by the options X1SPACE=, Y1SPACE=, X2SPACE=, or Y2SPACE=.

See Also: "About the Drawing Space and Drawing Units" on page 736 .

LAYER= FRONT | BACK
specifies whether the arrow appears on top of (FRONT) or behind (BACK) the graph.

Default: FRONT

Interaction: If this option is set to BACK, the arrow is drawn behind background areas, such as a layout or legend background.For elements that are obstructed because they are in the back layer, you can suppress the display of filled areas in the graph. Or, you can use transparency to manage the element visibility. For more information, see "About Drawing Layers" on page 739 .

LINEATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)
specifies the appearance of the arrow shaft and arrowhead(s). See "General Syntax for Attribute Options" on page 849 for the syntax on using a *style-element* and "Line Options" on page 850 for available *line-options*.

Default: The GraphAnnoLine style element.

TRANSPARENCY= *number*
specifies the degree of the transparency of the arrow shaft and arrowhead(s).

Default: 0

Range: 0 (opaque) to 1 (entirely transparent)

XAXIS= X | X2
specifies whether the data value for the arguments X1 and X2 are interpreted using the primary X axis scale or to the secondary X (X2) axis scale.

Default: X

Interaction: This option has effect only if X1SPACE=DATAVALUE.

X1SPACE= GRAPHPERCENT | GRAPHPIXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALUE
specifies the drawing space and drawing units for interpreting the X1 value.

Default: The setting that is in effect for the DRAWSPACE= option.

Interaction: This option overrides the DRAWSPACE= setting only for the X1 value.

X2SPACE= GRAPHPERCENT | GRAPHPIXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALUE
specifies the drawing space and drawing units for interpreting the X2 value

Default: The setting that is in effect for the DRAWSPACE= option.

Interaction: This option overrides the DRAWSPACE= setting only for the X2 value.

YAXIS= Y | Y2
specifies whether the data value for the arguments Y1 and Y2 are interpreted using the primary Y axis scale or to the secondary Y (Y2) axis scale.

Default: Y

Interaction: This option has effect only if Y1SPACE=DATAVALUE.

Y1SPACE= GRAPHPERCENT | GRAPHPIXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALUE
specifies the drawing space and drawing units for interpreting the Y1 value.

Default: The setting that is in effect for the DRAWSPACE= option.

Interaction: This option overrides the DRAWSPACE= setting only for the Y1 value.

Y2SPACE= GRAPHPERCENT | GRAPHPIXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALUE
specifies the drawing space and drawing units for interpreting the Y2 value

Default: The setting that is in effect for the DRAWSPACE= option.

Interaction: This option overrides the DRAWSPACE= setting only for the Y2 value.

Chapter 67

DRAWIMAGE Statement

Description and Syntax

Draws an image in the graph.

DRAWIMAGE "*image-file-spec*" <*option(s)*>

Example Program and Statement Details

Example Graph

The following graph was generated by the “Example Program” on page 764. The example shows how to display an image in the bottom right corner of the graph wall.

Example Program

```
proc template;
  define statgraph image;
    begingraph;
      entrytitle "Regression Fit Plot";
      layout overlay;
        modelband "myclm";
        scatterplot x=height y=weight / primary=true;
        regressionplot x=height y=weight / alpha=.01 clm="myclm";
        drawimage "c:\temp\saslogo.gif" /
          anchor=bottomright x=98 y=2
          drawspace=wallpercent ;
      endlayout;
    endgraph;
  end;

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

Statement Summary

A DRAWIMAGE statement draws an image in a graph. By default, the image is drawn in the center of the graph. You can change the default position with the options ANCHOR=, X=, Y=, XSPACE=, and YSPACE=. By default, the image is drawn in the actual image size. You can change the default size with the WIDTH= and HEIGHT= options.

For general information about the types of elements that can be drawn with the draw statements, the drawing space and drawing units that they use, and how the drawn elements are anchored, see Chapter 63, “Key Concepts for Using Draw Statements,” on

page 733 . For detailed usage information, consult the *SAS Graph Template Language: User's Guide*.

Required Argument

image-file-spec
specifies the name, image type, and location of the image. The *image-file-spec* must be enclosed in quotation marks and must specify a local, physical file path (for example, `"c:\temp\saslogo.gif"`). The supported image types are GIF, JPEG, and PNG (raster or bitmap format).

Restriction: URL access to image files is not supported. The image file must exist on the file system.

Options

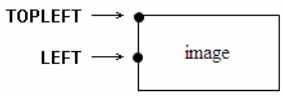
Statement Option	Description
ANCHOR	Specifies an anchor point for the image.
BORDER	Specifies whether a border is drawn around the image.
BORDERATTRS	Specifies the properties of the border line that is drawn around the image.
DISCRETEOFFSET	Specifies an amount to offset the anchor point from discrete X values, or discrete Y values, or both.
DRAWSPACE	Specifies a default drawing space and drawing units for this DRAWIMAGE statement.
HEIGHT	Specifies the height of the image's bounding box.
HEIGHTUNIT	Specifies whether the HEIGHT= setting is interpreted as a percentage value, a pixel value, or a value that is in the units of the data.
LAYER	Specifies whether the image appears on top of or behind the graph.
ROTATE	Specifies the angle of rotation for the image, measured in degrees.
SCALE	Specifies how the image is scaled within the bounding box.
SIZEUNIT	Specifies whether the default units for the size of the image's bounding box are percentage values, or pixel values, or values that are in the unit of the data.

Statement Option	Description
TRANSPARENCY	Specifies the degree of the transparency of the image and the border.
WIDTH	Specifies the width of the image's bounding box.
WIDTHUNIT	Specifies whether the WIDTH= setting is interpreted as a percentage value, a pixel value, or a value that is in the units of the data.
X	Specifies the anchor point's X coordinate.
XAXIS	Specifies whether the data value for the X= option is interpreted using the primary X axis scale or to the secondary X (X2) axis scale.
XSPACE	Specifies the drawing space and drawing units for interpreting the value that is specified in the X= option.
Y	Specifies the anchor point's Y coordinate.
YAXIS	Specifies whether the data value for the Y= option is interpreted using the primary Y axis scale or to the secondary Y (Y2) axis scale.
YSPACE	Specifies the drawing space and drawing units for interpreting the value that is specified in the Y= option.

ANCHOR=CENTER | TOPLEFT | TOP | TOPRIGHT | LEFT | RIGHT | BOTTOMLEFT | BOTTOM | BOTTOMRIGHT
specifies an anchor point for the image.

Default: CENTER

Discussion: The anchor point can be at the center of the image or at eight points on the border of the image bounding box. The following figure shows the anchor points for TOPLEFT and LEFT.



The coordinates of the anchor point are set by the X= and Y= options, and by the XSPACE= and YSPACE= options. The XAXIS= and YAXIS= option might affect positioning when the XSPACE= or YSPACE= options are set to DATAPIXEL, DATAPERCENT, or DATAVALUE.

The image has a fixed height and a fixed width, determined by the HEIGHT=, HEIGHTUNIT=, WIDTH= and WIDTHUNIT= options. The height of the text grows in a direction that is related to the anchor point. For example, if ANCHOR=TOPLEFT, the image height extends downward from the anchor point and its width extends to the right. If ANCHOR=CENTER, half the image width and half the image height extend equally left and right, as well as top to bottom from the

anchor point. If ANCHOR=BOTTOM, the image height extends upward from the anchor point and the image width is centered at the anchor point.

BORDER
specifies whether a border is drawn around the image.

Default: FALSE

Tip: Use the BORDERATTRS= option to control the appearance of the border.

BORDERATTRS= *style-element* | *style-element* (*line-options*) | (*line-options*)
specifies the properties of the border line that is drawn around the image. See "General Syntax for Attribute Options" on page 849 for the syntax on using a *style-element* and "Line Options" on page 850 for available *line-options*.

Default: The GraphBorderLines style element.

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

DISCRETEOFFSET= *number*
specifies an amount to offset the anchor point from discrete X values, or discrete Y values, or both.

Default: 0 (no offset, the anchor point is centered on discrete X values, or discrete Y values, or both)

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.

DRAWSPACE= GRAPHPERCENT | GRAPHPIXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALUE
specifies a default drawing space and drawing units for this DRAWIMAGE statement.

Default: LAYOUTPERCENT

Interaction: This statement and all of the draw statements inherit the global DRAWPSACE= setting from the DRAWSPACE= option in the BEGINGRAPH statement. Setting this option changes the setting for only this DRAWIMAGE statement.

Interaction: This option sets the default drawing space, but individual settings in the X or Y dimension can be overridden by the options XSPACE=, YSPACE=, HEIGHTUNIT =, or WIDTHUNIT=.

See Also: "About the Drawing Space and Drawing Units" on page 736 .

HEIGHT=*positive-number*
specifies the height of the image's bounding box.

Default: The height of the image.

Interaction: The interpretation of this height setting is determined by the combined settings of the HEIGHTUNIT= and YSPACE= options.

HEIGHTUNIT=PERCENT | PIXEL | DATA
specifies whether the *positive-number* that is specified in the HEIGHT= option is interpreted as a percentage value, a pixel value, or a value that is in the units of the data.

Default: PERCENT

Interaction: This setting combines with the YSPACE= setting to interpret the height that is set in the HEIGHT= option.

LAYER=FRONT | BACK
specifies whether the image appears on top of (FRONT) or behind (BACK) the graph.

Default: FRONT

Interaction: If this option is set to BACK, the image is drawn behind background areas, such as a layout or legend background.For elements that are obstructed because they are in the back layer, you can suppress the display of filled areas in the graph. Or, you can use transparency to manage the element visibility. For more information, see “About Drawing Layers” on page 739 .

ROTATE=*number*
specifies the angle of rotation for the image, measured in degrees. The angle is measured as if a horizontal line extended to the right through the image anchor point. Positive angles rotate the image counter clockwise, and negative angles rotate the image clockwise. The angle specification can exceed 360 degrees in absolute value.

Default: 0. No rotation is performed

SCALE= FIT | FITHEIGHT | FITWIDTH | TILE
specifies how the image is scaled within the bounding box.

Default: FIT

FIT	Scale the image to fit the bounding box. Aspect ratio is not maintained.
FITHEIGHT	Scale the image to fit the height of the bounding box. The width is computed from the height and the image's aspect ratio.
FITWIDTH	Scale the image to fit the width of the bounding box. The height is computed from the width and the image's aspect ratio.
TILE	Tile the image as needed to fit the bounding image. The last tile in a row or column might be clipped by the bounding box.

SIZEUNIT= PERCENT | PIXEL | DATA
specifies whether the default units for the size of the image’s bounding box are percentage values, or pixel values, or values that are in the unit of the data.

Default: PERCENT

Interaction: If the HEIGHTUNIT= or WIDTHUNIT= option is also used, it overrides this option for that dimension.

TRANSPARENCY= *number*
specifies the degree of the transparency of the image and the border.

Default: 0

Range: 0 (opaque) to 1 (entirely transparent)

WIDTH= *positive-number*
specifies the width of the image’s bounding box.

Default: The width of the image.

WIDTHUNIT=PERCENT | PIXEL | DATA
specifies whether the *positive-number* that is specified in the WIDTH= option is interpreted as a percentage value, a pixel value, or a value that is in the units of the data.

Default: PERCENT

Interaction: This setting combines with the XSPACE= setting to interpret the width that is set in the WIDTH= option.

X=*constant* | *scalar-expression*
specifies the anchor point's X coordinate.

Default: 50

Interaction: The DRAWSPACE= option determines the default interpretation of the units for this setting. You can override the default with the XSPACE= option.

Interaction: If XSPACE=DATAVALUE, this option's value is interpreted using the XAXIS= option.

XAXIS= X | X2
specifies whether the value that is specified for the X= option is interpreted using the primary X axis scale or to the secondary X (X2) axis scale.

Default: X

Interaction: This option has effect only if XSPACE=DATAVALUE.

XSPACE= GRAPHPERCENT | GRAPHPIXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALUE
specifies the drawing space and drawing units for interpreting the value that is specified in the X= option.

Default: The setting that is in effect for the DRAWSPACE= option.

Y=*constant* | *scalar-expression*
specifies the anchor point's Y coordinate.

Default: 50

Interaction: The DRAWSPACE= option determines the default interpretation of the units for this setting. You can override the default with the YSPACE= option.

Interaction: If YSPACE=DATAVALUE, this option's value is interpreted using the YAXIS= option.

YAXIS= Y | Y2
specifies whether the value that is specified for the Y= option is interpreted using the primary Y axis scale or to the secondary Y (Y2) axis scale.

Default: Y

Interaction: This option has effect only if YSPACE=DATAVALUE.

YSPACE= GRAPHPERCENT | GRAPHPIXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALUE
specifies the drawing space and drawing units for interpreting the value that is specified in the Y= option.

Default: The setting that is in effect for the DRAWSPACE= option.

Chapter 68

DRAWLINE Statement

Description and Syntax

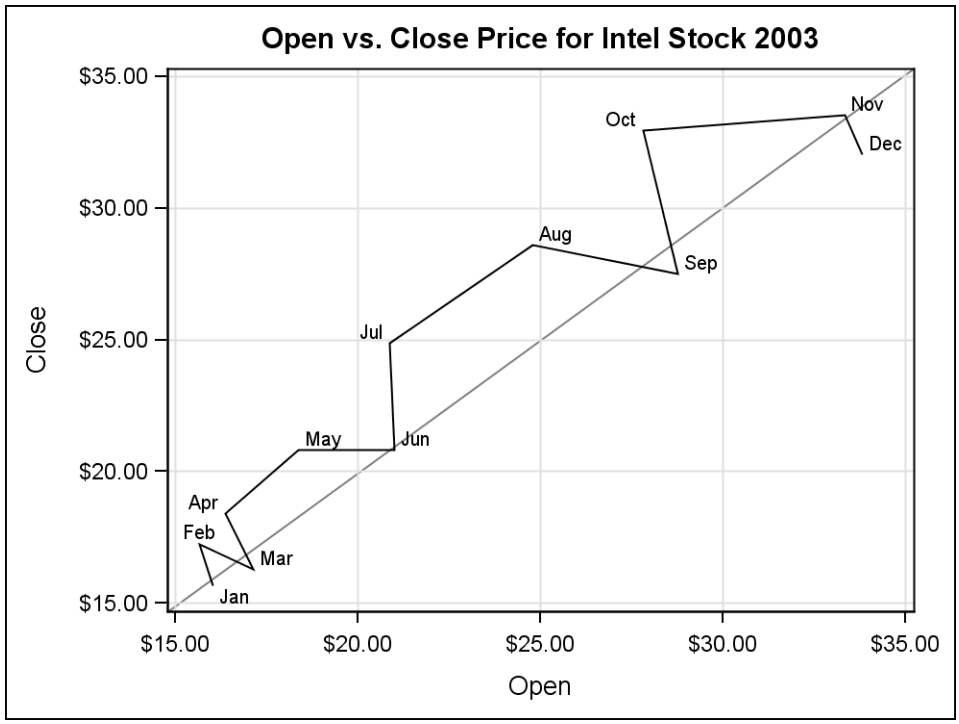
Draws a line from one point to another point.

DRAWLINE X1=*constant* | *scalar-expression*
Y1=*constant* | *scalar-expression*
X2=*constant* | *scalar-expression*
Y2=*constant* | *scalar-expression* </*option(s)*>;

Example Program and Statement Details

Example Graph

The following graph was generated by the "Example Program" on page 772. The example shows how to draw a diagonal reference line. One endpoint is point 0,0 and the other is point 100,100. Both points are specified in the WALL area with PERCENT units, making it easy to position the line without regard to the axis data ranges or the axis offsets. To draw the line behind the series line and grid lines, you can set LAYER= BACK and use the parent layout statement to turn off the display of the wall.



Example Program

```
proc template;
  define statgraph diagonal;
    begingraph;
      entrytitle "Open vs. Close Price for Intel Stock 2003";
      layout overlay / walldisplay=(outline)
          xaxisopts=(griddisplay=on
          yaxisopts=(griddisplay=on);
        seriesplot x=open y=close / datalabel=date;
        drawline x1=0 y1=0 x2=100 y2=100 /
          x1space=wallpercent y1space=wallpercent
          x2space=wallpercent y2space=wallpercent
          lineattrs=GraphReference layer=back ;
      endlayout;
    endgraph;
  end;

proc sgrender data=sashelp.stocks template=diagonal;
  where stock="Intel" and year(date)=2003;
  format date monname3.;
run;
```

Statement Summary

A DRAWLINE statement draws a line from a starting point that is specified with the X1 and Y1 arguments, to an ending point that is specified with the X2 and Y2 arguments. DRAWLINE is similar to a DRAWARROW statement, using many of the same options, but without the options for controlling the arrowhead(s).

For general information about the types of elements that can be drawn with the draw statements, the drawing space and drawing units that they use, and how the drawn elements are anchored, see Chapter 63, "Key Concepts for Using Draw Statements," on page 733 . For detailed usage information, consult the *SAS Graph Template Language: User's Guide*.

Required Argument

X1=*constant* | *scalar-expression*
specifies the X value of the starting point for the line.

Interaction: This value that is set for this option is interpreted using the X1SPACE= option. When X1SPACE=DATAVALUE, the value is interpreted using the XAXIS= option.

Y1=*constant* | *scalar-expression*
specifies the Y value of the starting point for the line.

Interaction: This value that is set for this option is interpreted using the Y1SPACE= option. When Y1SPACE=DATAVALUE, the value is interpreted using the YAXIS= option.

X2=*constant* | *scalar-expression*
specifies the X value of the endpoint for the line.

Interaction: This value that is set for this option is interpreted using the X2SPACE= option. When X2SPACE=DATAVALUE, the value is interpreted using the XAXIS= option.

Y2=*constant* | *scalar-expression*
specifies the Y value of the endpoint for the line.

Interaction: This value that is set for this option is interpreted using the Y2SPACE= option. When Y2SPACE=DATAVALUE, the value is interpreted using the YAXIS= option.

Options

Statement Option	Description
DISCRETEOFFSET	Specifies an amount to offset the line from discrete X values, or discrete Y values, or both.
DRAWSPACE	Specifies a default drawing space and drawing units for this DRAWLINE statement.
LAYER	Specifies whether the line appears on top of or behind the graph.
LINEATTRS	Specifies the appearance of the line.
TRANSPARENCY	Specifies the degree of the transparency of the line.

Statement Option	Description
XAXIS	Specifies whether the data value for the arguments X1 and X2 are interpreted using the primary X axis scale or to the secondary X (X2) axis scale.
X1SPACE	Specifies the drawing space and drawing units for interpreting the X1 value
X2SPACE	Specifies the drawing space and drawing units for interpreting the X2 value
YAXIS	Specifies whether the data value for the arguments Y1 and Y2 are interpreted using the primary Y axis scale or to the secondary Y (Y2) axis scale.
Y1SPACE	Specifies the drawing space and drawing units for interpreting the Y1 value
Y2SPACE	Specifies the drawing space and drawing units for interpreting the Y2 value

DISCRETEOFFSET= *number*
specifies an amount to offset the line from discrete X values, or discrete Y values, or both.

Default: 0 (no offset, all lines are centered on discrete X values, or discrete Y values, or both)

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.

Interaction: This option applies only when the options X1SPACE=, X2SPACE=, Y1SPACE=, or Y2SPACE= use DATAVALUE, and when X1, X2, Y1, or Y2 are values on a discrete axis.

DRAWSPACE= GRAPHPERCENT | GRAPHPIXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALUE
specifies a default drawing space and drawing units for this DRAWLINE statement.

Default: LAYOUTPERCENT

Interaction: This statement and all of the draw statements inherit the global DRAWPSACE= setting from the DRAWSPACE= option in the BEGINGRAPH statement. Setting this option changes the setting for only this DRAWLINE statement.

Interaction: This option sets the default drawing space, but individual settings in the X or Y dimension can be overridden by the options X1SPACE=, Y1SPACE=, X2SPACE=, or Y2SPACE=.

See Also: “About the Drawing Space and Drawing Units” on page 736 .

LAYER= FRONT | BACK
specifies whether the line appears on top of (FRONT) or behind (BACK) the graph.

Default: FRONT

Interaction: If this option is set to BACK, the line is drawn behind background areas, such as a layout or legend background.For elements that are obstructed because they are in the back layer, you can suppress the display of filled areas in the graph. Or, you can use transparency to manage the element visibility. For more information, see "About Drawing Layers" on page 739 .

LINEATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)
specifies the appearance of the line. See "General Syntax for Attribute Options" on page 849 for the syntax on using a *style-element* and "Line Options" on page 850 for available *line-options*.

Default: The GraphAnnoLine style element.

TRANSPARENCY= *number*
specifies the degree of the transparency of the line.

Default: 0

Range: 0 (opaque) to 1 (entirely transparent)

XAXIS= X | X2
specifies whether the data value for the arguments X1 and X2 are interpreted using the primary X axis scale or to the secondary X (X2) axis scale.

Default: X

Interaction: This option has effect only if X1SPACE=DATAVALUE.

X1SPACE= GRAPHPERCENT | GRAPHPIXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALUE
specifies the drawing space and drawing units for interpreting the X1 value.

Default: The setting that is in effect for the DRAWSPACE= option.

Interaction: This option overrides the DRAWSPACE= setting only for the X1 value.

X2SPACE= GRAPHPERCENT | GRAPHPIXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALUE
specifies the drawing space and drawing units for interpreting the X2 value

Default: The setting that is in effect for the DRAWSPACE= option.

Interaction: This option overrides the DRAWSPACE= setting only for the X2 value.

YAXIS= Y | Y2
specifies whether the data value for the arguments Y1 and Y2 are interpreted using the primary Y axis scale or to the secondary Y (Y2) axis scale.

Default: Y

Interaction: This option has effect only if Y1SPACE=DATAVALUE.

Y1SPACE= GRAPHPERCENT | GRAPHPIXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALUE
specifies the drawing space and drawing units for interpreting the Y1 value.

Default: The setting that is in effect for the DRAWSPACE= option.

Interaction: This option overrides the DRAWSPACE= setting only for the Y1 value.

Y2SPACE= GRAPHPERCENT | GRAPHPIXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALUE

specifies the drawing space and drawing units for interpreting the Y2 value

Default: The setting that is in effect for the DRAWSPACE= option.

Interaction: This option overrides the DRAWSPACE= setting only for the Y2 value.

Chapter 69

DRAWOVAL Statement

Description and Syntax

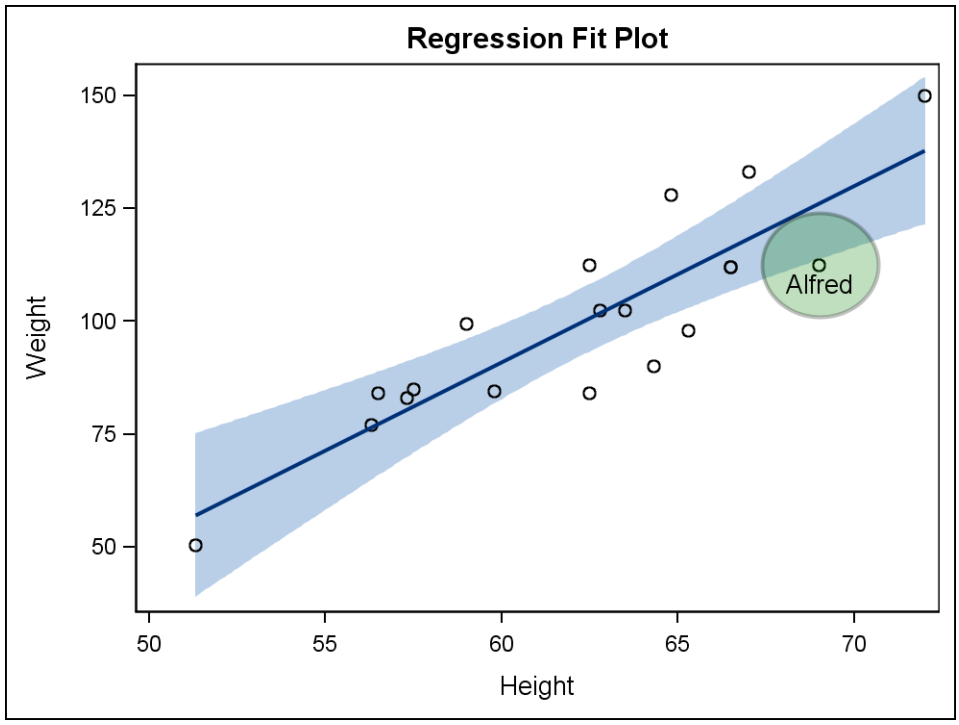
Draws an oval in the graph.

DRAWOVAL X=*constant* | *scalar-expression*
Y=*constant* | *scalar-expression*
WIDTH=*constant* | *scalar-expression*
HEIGHT=*constant* | *scalar-expression* </*option(s)*>

Example Program and Statement Details

Example Graph

The following graph was generated by the "Example Program" on page 778. The example uses DRAWOVAL to highlight a student's data point. It draws an oval around the marker symbol that represents the student's height and weight, and it displays the student's name inside of the oval. In the BEGINGRAPH statement, the setting for the DRAWSPACE= option sets the drawing space and drawing units for the DRAWOVAL and DRAWTEXT statements. In the DRAWOVAL statement, setting TRANSPARENCY=0.75 ensures that the marker for Alfred is visible behind the oval. The DRAWTEXT statement draws the text that identifies the student's name, using the ANCHOR=, X=, and Y= options to position the text within the oval.



Example Program

```
proc template;
  define statgraph drawoval;
    begingraph / drawspace=datavalue;
      entrytitle "Regression Fit Plot";
      layout overlay;
        modelband "myclm";
        scatterplot x=height y=weight;
        regressionplot x=height y=weight / alpha=.01 clm="myclm";
          drawoval x=69 y=112.5 width=15 height=20 /
            display=all fillattrs=(color=green)
            transparency=0.75 ;
          drawtext "Alfred" / x=69 y=112 anchor=top;
      endlayout;
    endgraph;
  end;

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

Statement Summary

A DRAWOVAL statement draws an oval in a graph. The oval position is determined by the X and Y anchor points, and the size is determined by the HEIGHT and WIDTH settings. You can manage the oval position with the options ANCHOR= , XSPACE= , and YSPACE= . You can manage the oval size with the HEIGHTUNIT= and WIDTHUNIT= options.

For general information about the types of elements that can be drawn with the draw statements, the drawing space and drawing units that they use, and how the drawn

elements are anchored, see Chapter 63, "Key Concepts for Using Draw Statements," on page 733 . For detailed usage information, consult the *SAS Graph Template Language: User's Guide*.

Required Argument

X=*constant* | *scalar-expression*
specifies the X value of the anchor point.

Interaction: The value that is set for this argument is interpreted using the XSPACE= option. When XSPACE=DATAVALUE, the value is interpreted using the XAXIS= option.

Y=*constant* | *scalar-expression*
specifies the Y value of the anchor point.

Interaction: The value that is set for this argument is interpreted using the YSPACE= option. When YSPACE=DATAVALUE, the value is interpreted using the YAXIS= option.

WIDTH=*constant* | *scalar-expression*
specifies the width of the oval.

Interaction: The value that is set for this argument is interpreted using the WIDTHUNIT= and XSPACE= options. When WIDTHUNIT=DATA, the value is interpreted using the XAXIS= option.

HEIGHT=*constant* | *scalar-expression*
specifies the height of the oval.

Interaction: The value that is set for this argument is interpreted using the HEIGHTUNIT= and YSPACE= options. When HEIGHTUNIT=DATA, the value is interpreted using the YAXIS= option.

Options

Statement Option	Description
ANCHOR	Specifies an anchor point for the oval.
DISCRETEOFFSET	Specifies an amount to offset the anchor point from discrete X values, or discrete Y values, or both.
DISPLAY	Specifies the features to display for the oval.
DRAWSPACE	Specifies a default drawing space and drawing units for this DRAWOVAL statement.
FILLATTRS	Specifies the appearance of the oval's filled area.
HEIGHTUNIT	Specifies whether the value that is specified for the oval's HEIGHT is interpreted as a percentage value, a pixel value, or a value that is in the units of the data.

Statement Option	Description
LAYER	Specifies whether the oval appears on top of or behind the graph.
OUTLINEATTRS	Specifies the properties of the outline that is drawn around the oval.
ROTATE	Specifies the angle of rotation for the oval, measured in degrees.
TRANSPARENCY	Specifies the degree of the transparency of the fill and the outline.
WIDTHUNIT	Specifies whether the value that is specified for the oval's WIDTH is interpreted as a percentage value, a pixel value, or a value that is in the units of the data.
XAXIS	Specifies whether the value that is specified for the oval's X anchor point is interpreted using the primary X axis scale or to the secondary X (X2) axis scale.
XSPACE	Specifies the drawing space and drawing units for interpreting the value that is specified for the oval's X anchor point.
YAXIS	Specifies whether the value that is specified for the oval's Y anchor point is interpreted using the primary Y axis scale or to the secondary Y (Y2) axis scale.
YSPACE	Specifies the drawing space and drawing units for interpreting the value that is specified for the oval's Y anchor point.

ANCHOR=CENTER | TOPLEFT | TOP | TOPRIGHT | LEFT | RIGHT | BOTTOMLEFT | BOTTOM | BOTTOMRIGHT
specifies an anchor point for the oval.

Default: CENTER

Discussion: The anchor point can be at the center of the oval or at eight points on the bounding box of the rectangle. The following figure shows the anchor points for TOPLEFT and CENTER.

The coordinates of the anchor point are set by the statement's X and Y values, and by the settings for the XSPACE= and YSPACE= options. The XAXIS= and YAXIS= option might affect positioning when the XSPACE= or YSPACE= options are set to DATAPIXEL, DATAPERCENT, or DATAVALUE.

DISCRETEOFFSET= *number*
specifies an amount to offset the anchor point from discrete X values, or discrete Y values, or both.

Default: 0 (no offset, the anchor point is centered on discrete X values, or discrete Y values, or both)

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.

DISPLAY=STANDARD | ALL | (*display-options*)
specifies the features to display for the oval.

Default: STANDARD

STANDARD
Displays an outlined oval.

ALL
Displays an outlined, filled oval.

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

OUTLINE	displays an outlined oval
FILL	displays a filled oval

Tip: Use the OUTLINEATTRS= and FILLATTRS= options to control the appearance of the oval.

DRAWSPACE= GRAPHPERCENT | GRAPHPIXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALUE
specifies a default drawing space and drawing units for this DRAWOVAL statement.

Default: LAYOUTPERCENT

Interaction: This statement and all of the draw statements inherit the global DRAWPSACE= setting from the DRAWSPACE= option in the BEGINGRAPH statement. Setting this option changes the setting for only this DRAWOVAL statement.

Interaction: This option sets the default drawing space, but individual settings in the X or Y dimension can be overridden by the options XSPACE=, YSPACE=, HEIGHTUNIT =, or WIDTHUNIT=.

See Also: "About the Drawing Space and Drawing Units" on page 736 .

FILLATTRS=*style-element* | *style-element* (*fill-options*) | (*fill-options*)
specifies the appearance of the oval's filled area. See "General Syntax for Attribute Options" on page 849 for the syntax on using a *style-element* and "Fill Options" on page 850 for available *fill-options*.

Default: The GraphAnnoShape style element

Tip: The TRANSPARENCY= option sets the transparency for the fill and the outline. You can combine this option with TRANSPARENCY= to set one transparency for the outline but a different transparency for the fill. Example:

```
transparency=0.2 fillattrs=(transparency=0.6)
```

HEIGHTUNIT=PERCENT | PIXEL | DATA
specifies whether the value that is specified for the oval's HEIGHT is interpreted as a percentage value, a pixel value, or a value that is in the units of the data.

Default: PERCENT

Interaction: This setting combines with the YSPACE= setting to interpret the height that is set in the HEIGHT= argument.

LAYER=FRONT | BACK
specifies whether the oval appears on top of (FRONT) or behind (BACK) the graph.

Default: FRONT

Interaction: If this option is set to BACK, the oval is drawn behind background areas, such as a layout or legend background.For elements that are obstructed because they are in the back layer, you can suppress the display of filled areas in the graph. Or, you can use transparency to manage the element visibility. For more information, see "About Drawing Layers" on page 739 .

OUTLINEATTRS= *style-element* | *style-element* (*line-options*) | (*line-options*)
specifies the properties of the outline that is drawn around the oval. See "General Syntax for Attribute Options" on page 849 for the syntax on using a *style-element* and "Line Options" on page 850 for available *line-options*.

Default: The GraphAnnoShape style element.

Interaction: For this option to have any effect, the outline must be enabled by the DISPLAY= option or by the ODS style that is in effect.

ROTATE=*number*
specifies the angle of rotation for the oval, measured in degrees. The angle is measured as if a horizontal line extended to the right through the oval anchor point. Positive angles rotate the oval counter clockwise, and negative angles rotate the oval clockwise. The angle specification can exceed 360 degrees in absolute value.

Default: 0. No rotation is performed

TRANSPARENCY= *number*
specifies the degree of the transparency of the fill and the outline.

Default: 0

Range: 0 (opaque) to 1 (entirely transparent)

Tip: The FILLATTRS= option can be used to set transparency for just the filled area. You can combine this option with FILLATTRS= to set one transparency for the outline but a different transparency for the fill. Example:

```
transparency=0.2 fillattrs=(transparency=0.6)
```

WIDTHUNIT=PERCENT | PIXEL | DATA
specifies whether the value that is specified for the oval's WIDTH is interpreted as a percentage value, a pixel value, or a value that is in the units of the data.

Default: PERCENT

Interaction: This setting combines with the XSPACE= setting to interpret the width that is set in the WIDTH= argument.

XAXIS= X | X2
specifies whether the value that is specified for the oval's X anchor point is interpreted using the primary X axis scale or to the secondary X (X2) axis scale.

Default: X

Interaction: This option has effect only if XSPACE=DATAVALUE.

XSPACE= GRAPHPERCENT | GRAPHPIXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALUE
specifies the drawing space and drawing units for interpreting the value that is specified for the oval's X anchor point.

Default: The setting that is in effect for the DRAWSPACE= option.

YAXIS= Y | Y2
specifies whether the value that is specified for the oval's Y anchor point is interpreted using the primary Y axis scale or to the secondary Y (Y2) axis scale.

Default: Y

Interaction: This option has effect only if YSPACE=DATAVALUE.

YSPACE= GRAPHPERCENT | GRAPHPIXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALUE
specifies the drawing space and drawing units for interpreting the value that is specified for the oval's Y anchor point.

Default: The setting that is in effect for the DRAWSPACE= option.

Chapter 70

DRAWRECTANGLE Statement

Description and Syntax

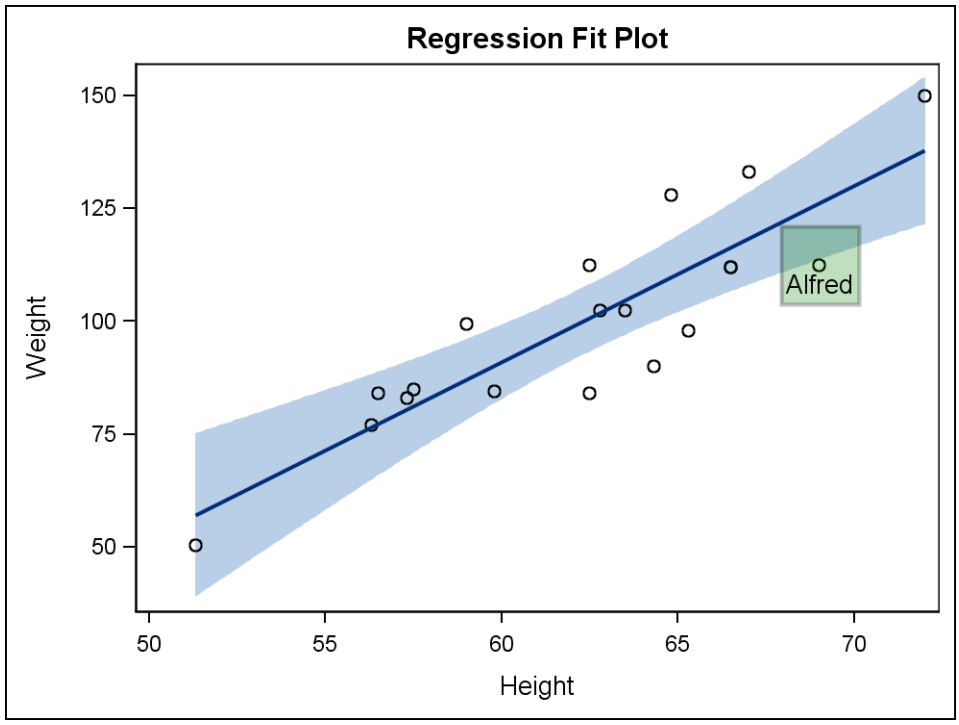
Draws a rectangle in the graph.

DRAWRECTANGLE X=*constant* | *scalar-expression*
Y=*constant* | *scalar-expression*
WIDTH=*constant* | *scalar-expression*
HEIGHT=*constant* | *scalar-expression* </*option(s)*>

Example Program and Statement Details

Example Graph

The following graph was generated by the "Example Program" on page 786. The example uses DRAWRECTANGLE to highlight a student's data point. It draws a rectangle around the marker symbol that represents the student's height and weight, and it displays the student's name inside of the rectangle. In the BEGINGRAPH statement, the setting for the DRAWSPACE= option sets the drawing space and drawing units for the DRAWRECTANGLE and DRAWTEXT statements. In the DRAWRECTANGLE statement, setting TRANSPARENCY=0.75 ensures that the marker for Alfred is visible behind the rectangle. The DRAWTEXT statement draws the text that identifies the student's name, using the ANCHOR=, X=, and Y= options to position the text within the rectangle.



Example Program

```
proc template;
  define statgraph drawrectangle;
    begingraph / drawspace=datavalue;
      entrytitle "Regression Fit Plot";
      layout overlay;
        modelband "myclm";
        scatterplot x=height y=weight;
        regressionplot x=height y=weight / alpha=.01 clm="myclm";
          drawrectangle x=69 y=112.5 width=10 height=15 /
            display=all fillattrs=(color=green)
            transparency=0.75 ;
          drawtext "Alfred" / x=69 y=112 anchor=top;
      endlayout;
    endgraph;
  end;

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

Statement Summary

A DRAWRECTANGLE statement draws a rectangle in a graph. The rectangle position is determined by the X and Y anchor points, and the size is determined by the HEIGHT and WIDTH settings. You can manage the rectangle position with the options ANCHOR= , XSPACE= , and YSPACE= . You can manage the rectangle size with the HEIGHTUNIT= and WIDTHUNIT= options.

For general information about the types of elements that can be drawn with the draw statements, the drawing space and drawing units that they use, and how the drawn

elements are anchored, see Chapter 63, “Key Concepts for Using Draw Statements,” on page 733 . For detailed usage information, consult the *SAS Graph Template Language: User's Guide*.

Required Argument

X=*constant* | *scalar-expression*
specifies the X value of the anchor point.

Interaction: The value that is set for this argument is interpreted using the XSPACE= option. When XSPACE=DATAVALUE, the value is interpreted using the XAXIS= option.

Y=*constant* | *scalar-expression*
specifies the Y value of the anchor point.

Interaction: The value that is set for this argument is interpreted using the YSPACE= option. When YSPACE=DATAVALUE, the value is interpreted using the YAXIS= option.

WIDTH=*constant* | *scalar-expression*
specifies the width of the rectangle.

Interaction: The value that is set for this argument is interpreted using the WIDTHUNIT= and XSPACE= options. When WIDTHUNIT=DATA, the value is interpreted using the XAXIS= option.

HEIGHT=*constant* | *scalar-expression*
specifies the height of the rectangle.

Interaction: The value that is set for this argument is interpreted using the HEIGHTUNIT= and YSPACE= options. When HEIGHTUNIT=DATA, the value is interpreted using the YAXIS= option.

Options

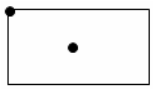
Statement Option	Description
ANCHOR	Specifies an anchor point for the rectangle.
CORNERRADIUS	Specifies the radius of the rectangle corners.
DISCRETEOFFSET	Specifies an amount to offset the anchor point from discrete X values, or discrete Y values, or both.
DISPLAY	Specifies the features to display for the rectangle.
DRAWSPACE	Specifies a default drawing space and drawing units for this DRAWRECTANGLE statement.
FILLATTRS	Specifies the appearance of the rectangle’s filled area.

Statement Option	Description
HEIGHTUNIT	Specifies whether the value that is specified for the rectangle's HEIGHT is interpreted as a percentage value, a pixel value, or a value that is in the units of the data.
LAYER	Specifies whether the rectangle appears on top of or behind the graph.
OUTLINEATTRS	Specifies the properties of the outline that is drawn around the rectangle.
TRANSPARENCY	Specifies the degree of the transparency of the fill and the outline.
WIDTHUNIT	Specifies whether the value that is specified for the rectangle's WIDTH is interpreted as a percentage value, a pixel value, or a value that is in the units of the data.
XAXIS	Specifies whether the value that is specified for the rectangle's X anchor point is interpreted using the primary X axis scale or to the secondary X (X2) axis scale.
XSPACE	Specifies the drawing space and drawing units for interpreting the value that is specified for the rectangle's X anchor point.
YAXIS	Specifies whether the value that is specified for the rectangle's Y anchor point is interpreted using the primary Y axis scale or to the secondary Y (Y2) axis scale.
YSPACE	Specifies the drawing space and drawing units for interpreting the value that is specified for the rectangle's Y anchor point.

ANCHOR=CENTER | TOPLEFT | TOP | TOPRIGHT | LEFT | RIGHT | BOTTOMLEFT | BOTTOM | BOTTOMRIGHT
specifies an anchor point for the rectangle.

Default: CENTER

Discussion: The anchor point can be at the center of the rectangle or at eight points on the rectangle border. The following figure shows the anchor points for TOPLEFT and CENTER.



The coordinates of the anchor point are set by the statement's X and Y values, and by the settings for the XSPACE= and YSPACE= options. The XAXIS= and YAXIS= option might affect positioning when the XSPACE= or YSPACE= options are set to DATAPIXEL, DATAPERCENT, or DATAVALUE.

CORNERRADIUS=*number*
specifies the radius of the rectangle corners.

Note: This feature is for the second maintenance release of SAS 9.3 and later. See "What's New in the SAS 9.3 Graph Template Language" on page xi.

Default: 0

Range: 0 (square corners) to 1 (most rounded corners)

DISCRETEOFFSET= *number*
specifies an amount to offset the anchor point from discrete X values, or discrete Y values, or both.

Default: 0 (no offset, the anchor point is centered on discrete X values, or discrete Y values, or both)

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.

DISPLAY=STANDARD | ALL | (*display-options*)
specifies the features to display for the rectangle.

Default: STANDARD

STANDARD
displays an outlined rectangle.

ALL
displays an outlined, filled rectangle.

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

OUTLINE	displays an outlined rectangle
FILL	displays a filled rectangle

Tip: Use the OUTLINEATTRS= and FILLATTRS= options to control the appearance of the rectangle.

DRAWSPACE= GRAPHPERCENT | GRAPHPIXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALUE
specifies a default drawing space and drawing units for this DRAWRECTANGLE statement.

Default: LAYOUTPERCENT

Interaction: This statement and all of the draw statements inherit the global DRAWPSACE= setting from the DRAWSPACE= option in the BEGINGRAPH statement. Setting this option changes the setting for only this DRAWRECTANGLE statement.

Interaction: This option sets the default drawing space, but individual settings in the X or Y dimension can be overridden by the options XSPACE=, YSPACE=, HEIGHTUNIT =, or WIDTHUNIT=.

See Also: "About the Drawing Space and Drawing Units" on page 736 .

FILLATTRS=*style-element* | *style-element* (*fill-options*) | (*fill-options*)
specifies the appearance of the fill. See "General Syntax for Attribute Options" on page 849 for the syntax on using a *style-element* and "Fill Options" on page 850 for available *fill-options*.

Default: The GraphAnnoShape style element

Tip: The TRANSPARENCY= option sets the transparency for the fill and the outline. You can combine this option with TRANSPARENCY= to set one transparency for the outline but a different transparency for the fill. Example:

```
transparency=0.2 fillattrs=(transparency=0.6)
```

HEIGHTUNIT=PERCENT | PIXEL | DATA
specifies whether the value that is specified for the rectangle's HEIGHT is interpreted as a percentage value, a pixel value, or a value that is in the units of the data.

Default: PERCENT

Interaction: This setting combines with the YSPACE= setting to interpret the height that is set in the HEIGHT= argument.

LAYER=FRONT | BACK
specifies whether the rectangle appears on top of (FRONT) or behind (BACK) the graph.

Default: FRONT

Interaction: If this option is set to BACK, the rectangle is drawn behind background areas, such as a layout or legend background.For elements that are obstructed because they are in the back layer, you can suppress the display of filled areas in the graph. Or, you can use transparency to manage the element visibility. For more information, see "About Drawing Layers" on page 739 .

OUTLINEATTRS= *style-element* | *style-element* (*line-options*) | (*line-options*)
specifies the properties of the outline that is drawn around the rectangle. See "General Syntax for Attribute Options" on page 849 for the syntax on using a *style-element* and "Line Options" on page 850 for available *line-options*.

Default: The GraphAnnoShape style element.

Interaction: For this option to have any effect, the outline must be enabled by the DISPLAY= option or by the ODS style that is in effect.

TRANSPARENCY= *number*
specifies the degree of the transparency of the fill and the outline.

Default: 0

Range: 0 (opaque) to 1 (entirely transparent)

Tip: The FILLATTRS= option can be used to set transparency for just the fill. You can combine this option with FILLATTRS= to set one transparency for the outline but a different transparency for the fill. Example:

```
transparency=0.2 fillattrs=(transparency=0.6)
```

WIDTHUNIT=PERCENT | PIXEL | DATA
specifies whether the value that is specified for the rectangle's WIDTH is interpreted as a percentage value, a pixel value, or a value that is in the units of the data.

Default: PERCENT

Interaction: This setting combines with the XSPACE= setting to interpret the width that is set in the WIDTH= argument.

XAXIS= X | X2
specifies whether the value that is specified for the rectangle's X anchor point is interpreted using the primary X axis scale or to the secondary X (X2) axis scale.

Default: X

Interaction: This option has effect only if XSPACE=DATAVALUE.

XSPACE= GRAPHPERCENT | GRAPHPIXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALUE
specifies the drawing space and drawing units for interpreting the value that is specified for the rectangle's X anchor point.

Default: The setting that is in effect for the DRAWSPACE= option.

YAXIS= Y | Y2
specifies whether the value that is specified for the rectangle's Y anchor point is interpreted using the primary Y axis scale or to the secondary Y (Y2) axis scale.

Default: Y

Interaction: This option has effect only if YSPACE=DATAVALUE.

YSPACE= GRAPHPERCENT | GRAPHPIXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALUE
specifies the drawing space and drawing units for interpreting the value that is specified for the rectangle's Y anchor point.

Default: The setting that is in effect for the DRAWSPACE= option.

Chapter 71

DRAWTEXT Statement

Description and Syntax

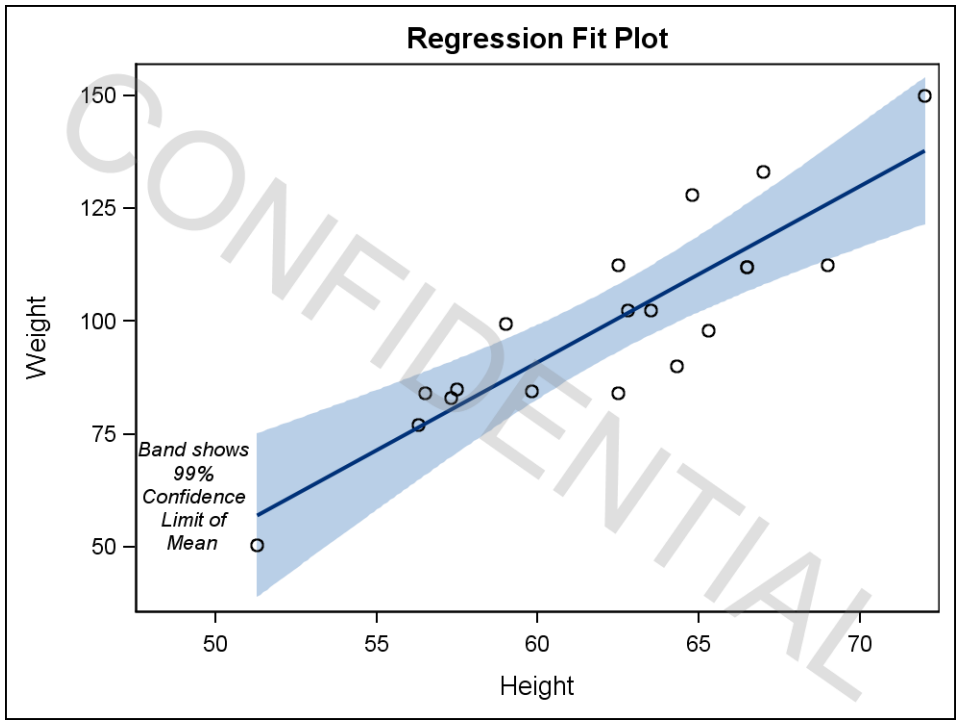
Draws and anchors in a graph a text box that contains one or more lines of formatted text.

DRAWTEXT *text-item <…<text-item>><\/option(s)>*

Example Program and Statement Details

Example Graph

The following graph was generated by the "Example Program" on page 794. The first DRAWTEXT statement shows how to draw multiple lines of text in a specific position within the graph. The second DRAWTEXT statement shows how to create a watermark, which is achieved by applying transparency to text that is rotated within the graph.



Example Program

```
proc template;
 define statgraph modelfit;
  begingraph;
    entrytitle "Regression Fit Plot";
    layout lattice;
      layout overlay / xaxisopts=(offsetmin=.1);
        drawtext textattrs=(style=italic size=8pt)
          "Band shows 99% Confidence Limit of Mean" /
          anchor=bottomleft width=15 widthunit=percent
          xspace=wallpercent yspace=wallpercent
          x=0 y=10 justify=center ;
        modelband "myclm";
        scatterplot x=height y=weight / primary=true;
        regressionplot x=height y=weight / alpha=.01 clm="myclm";
      endlayout;
    endlayout;
    drawtext textattrs=(color=gray size=52pt) "CONFIDENTIAL" /
      transparency=.75 rotate=-35
      width=110 widthunit=percent justify=center ;
  endgraph;
 end;

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

Statement Summary

A DRAWTEXT statement draws a text box that contains one or more lines of text. The text can be formatted, using the TEXTATTRS= prefix option. By default, the text box is

drawn in the center of the graph. You can change the default position with the options ANCHOR= , X= , Y= , XSPACE= , and YSPACE= .

For general information about the types of elements that can be drawn with the draw statements, the drawing space and drawing units that they use, and how the drawn elements are anchored, see Chapter 63, "Key Concepts for Using Draw Statements," on page 733 . For detailed usage information, consult the *SAS Graph Template Language: User's Guide*.

Required Argument

text-item <…<*text-item*>>
specifies one or more pieces of text for the text box. Each *text-item* has the following form:

<*prefix-option*> "*string*" | *dynamic* | *character-expression* | {*text-command*}

Each piece of text can have a prefix setting that precedes the piece of text. A piece of text is either a string literal, a dynamic, or a text command. All *text-items* are concatenated into one string, which might be wrapped, based on the settings for the WIDTH= and WIDTHUNIT= settings. Leading and trailing blanks in the concatenated string are always used.

When used, a prefix option applies to the immediately following piece of text and also to all subsequent text strings and text-commands until another prefix option is specified.

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 Chapter 59, "Managing Text Items," on page 699 for more information and several examples.

Options and Text Commands

Prefix Option	Description
TEXTATTRS	Specifies the color and font properties of all of the specified text or individual *text-items*.

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.

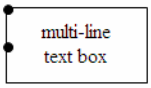
Statement Option	Description
ANCHOR	Specifies an anchor point for the text box.
BORDER	Specifies whether a border is drawn around the text box.
BORDERATTRS	Specifies the properties of the border line that is drawn around the text box.
DISCRETEOFFSET	Specifies an amount to offset the anchor point from discrete X values, or discrete Y values, or both.
DRAWSPACE	Specifies a default drawing space and drawing units for this DRAWTEXT statement.
JUSTIFY	Specifies the alignment of text that wraps within the text box.
LAYER	Specifies whether the text box appears on top of or behind the graph.
PAD	Specifies the amount of extra space that is reserved inside the text box's border
ROTATE	Specifies the angle of rotation for the text box, measured in degrees.
TRANSPARENCY	Specifies the degree of the transparency of the background, text, and border.
WIDTH	Specifies the width of the text box.
WIDTHUNIT	Specifies whether the value that is specified for the WIDTH= option is interpreted as a percentage value, a pixel value, or a value that is in the units of the data.
X	Specifies the anchor point's X coordinate.
XAXIS	Specifies whether the data value for the X= option is interpreted using the primary X axis scale or to the secondary X (X2) axis scale.
XSPACE	Specifies the drawing space and drawing units for interpreting the value that is specified in the X= option.
Y	Specifies the anchor point's Y coordinate.
YAXIS	Specifies whether the data value for the Y= option is interpreted using the primary Y axis scale or to the secondary Y (Y2) axis scale.

Statement Option	Description
YSPACE	Specifies the drawing space and drawing units for interpreting the value that is specified in the Y= option.

ANCHOR=CENTER | TOPLEFT | TOP | TOPRIGHT | LEFT | RIGHT | BOTTOMLEFT | BOTTOM | BOTTOMRIGHT
specifies an anchor point for the text box.

Default: CENTER

Discussion: The anchor point can be at the center of the text box or at eight points on the border of the text box bounding box. The following figure shows the anchor points for TOPLEFT and LEFT.



The coordinates of the anchor point are set by the X= and Y= options, and by the XSPACE= and YSPACE= options. The XAXIS= and YAXIS= option might affect positioning when the XSPACE= or YSPACE= options are set to DATAPIXEL, DATAPERCENT, or DATAVALUE.

The text box has a fixed width, determined by the WIDTH= and WIDTHUNIT= options. The height of the text box is based on the amount of text specified and the font size. The height of the text grows in a direction that is related to the anchor point. For example, if ANCHOR=TOPLEFT, the text box height extends downward from the anchor point and its width extends to the right. If ANCHOR=CENTER, half the text box width and half the text box height extend equally left and right, as well as top to bottom from the anchor point. If ANCHOR=BOTTOM, the text box height extends upward from the anchor point and the text box width is centered at the anchor point.

BORDER
specifies whether a border is drawn around the text box.

Default: FALSE

Tip: Use the BORDERATTRS= option to control the appearance of the border.

BORDERATTRS= *style-element* | *style-element* (*line-options*) | (*line-options*)
specifies the properties of the border line that is drawn around the text box. See "General Syntax for Attribute Options" on page 849 for the syntax on using a *style-element* and "Line Options" on page 850 for available *line-options*.

Default: The GraphBorderLines style element.

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

DISCRETEOFFSET= *number*
specifies an amount to offset the anchor point from discrete X values, or discrete Y values, or both.

Default: 0 (no offset, the anchor point is centered on discrete X values, or discrete Y values, or both)

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.

DRAWSPACE= GRAPHPERCENT | GRAPHPIXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALUE
specifies a default drawing space and drawing units for this DRAWTEXT statement.

Default: LAYOUTPERCENT

Interaction: This statement and all of the draw statements inherit the global DRAWPSACE= setting from the DRAWSPACE= option in the BEGINGRAPH statement. Setting this option changes the setting for only this DRAWTEXT statement.

Interaction: This option sets the default drawing space, but individual settings in the X or Y dimension can be overridden by the options XSPACE=, YSPACE=, or WIDTHUNIT=.

See Also: "About the Drawing Space and Drawing Units" on page 736 .

JUSTIFY=LEFT | CENTER | RIGHT
Specifies the alignment of text that wraps within the text box.

Default: LEFT

LEFT
forces the first character of each line to appear at the left margin (distance from the left border plus the left pad amount).

CENTER
forces each line to be centered in the text box between the left and right pad amounts.

RIGHT
forces the last character of each line to appear at the right margin (distance from the right border minus the right pad amount).

Interaction: Text is wrapped based on the width of the specified text, the font size, and the setting in the PAD= option.

LAYER=FRONT | BACK
specifies whether the text box appears on top of (FRONT) or behind (BACK) the graph.

Default: FRONT

Interaction: If this option is set to BACK, the text box is drawn behind background areas, such as a layout or legend background.For elements that are obstructed because they are in the back layer, you can suppress the display of filled areas in the graph. Or, you can use transparency to manage the element visibility. For more information, see "About Drawing Layers" on page 739 .

PAD=*dimension* | *(pad-options)*
specifies the amount of extra space that is reserved inside the text box's 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 text box 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 for the text box, measured in degrees. The angle is measured as if a horizontal line extended to the right through the text box anchor point. Positive angles rotate the text box counter clockwise, and negative angles rotate the text box clockwise. The angle specification can exceed 360 degrees in absolute value.

Default: 0. No rotation is performed

{ SUB *"string"* | *dynamic* }
text-command that specifies that the string or dynamic is to appear as subscript text.

Default: no default

Example:

```
drawtext "y = " b{sub "0"} " + b" {sub "1"} "x";
```

See Also: “Rules for Unicode and Special Character Specifications” on page 704 .

{ SUP *"string"* | *dynamic* }
text-command that specifies that the string or dynamic is to appear as superscript text.

Default: no default

Example:

```
drawtext "R" {sup "2"} " = " {format (6.4) RSQUARED} ;
```

See Also: “Rules for Unicode and Special Character Specifications” on page 704 .

TEXTATTRS=*style-element* | *style-element* (*text-options*) | (*text-options*)
prefix-option that specifies the color and font properties of the entire text string or individual *text-items*. See “General Syntax for Attribute Options” on page 849 for the syntax on using a *style-element* and “Text Options” on page 852 for available *text-options*.

Default: The GraphValueText style element.

Interaction: When multiple TEXTATTRS=( ) prefix options are used, each one cancels the last, resetting all text properties to the default set by the GraphValueText style element. Subsequent *text-items* to the right are then assigned the text properties specified in the closest TEXTATTRS=( ) setting to their left. Thus, to vary the text properties across *text-items*, you do not have to override settings from a previous TEXTATTRS=( ) setting. Each TEXTATTRS=( ) specification resets all text properties to the default so that only the new settings are applied to subsequent *text-items*.

Tip: To ensure that all text has the same text properties, use this prefix option once only and place it before the first *text-item*.

TRANSPARENCY= *number*
specifies the degree of the transparency of the background, text, and border.

Default: 0

Range: 0 (opaque) to 1 (entirely transparent)

{ 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 704..

dynamic
The dynamic must resolve to either *"hex-string"x* or a keyword for a UNICODE character.

Examples: The following statements show how to use the {UNICODE} text command:

```
drawtext {unicode alpha} "=" CONF;

drawtext {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 support only 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.

See Also: "Using UNICODE Text" on page 703 and "Rules for Unicode and Special Character Specifications" on page 704 .

WIDTH= *positive-number*
specifies the width of the text box.

Default: 10

Interaction: The WIDTHUNIT= option determines the units of *positive-number*, which is PERCENT by default.

WIDTHUNIT=PERCENT | PIXEL | DATA
specifies whether the value that is specified for the WIDTH= option is interpreted as a percentage value, a pixel value, or a value that is in the units of the data.

Default: PERCENT

Interaction: This setting combines with the XSPACE= setting to interpret the width that is set in the WIDTH= argument.

X=*constant* | *scalar-expression*
specifies the anchor point's X coordinate.

Default: 50

Interaction: The DRAWSPACE= option determines the default interpretation of the units for this setting. You can override the default with the XSPACE= option.

Interaction: If XSPACE=DATAVALUE, this option's value is interpreted using the XAXIS= option.

XAXIS= X | X2
specifies whether the value that is specified for the X= option is interpreted using the primary X axis scale or to the secondary X (X2) axis scale.

Default: X

Interaction: This option has effect only if XSPACE=DATAVALUE.

XSPACE= GRAPHPERCENT | GRAPHPIXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALUE
specifies the drawing space and drawing units for interpreting the value that is specified in the X= option.

Default: The setting that is in effect for the DRAWSPACE= option.

Y=*constant* | *scalar-expression*
specifies the anchor point's Y coordinate.

Default: 50

Interaction: The DRAWSPACE= option determines the default interpretation of the units for this setting. You can override the default with the YSPACE= option.

Interaction: If YSPACE=DATAVALUE, this option's value is interpreted using the YAXIS= option.

YAXIS= Y | Y2
specifies whether the value that is specified for the Y= option is interpreted using the primary Y axis scale or to the secondary Y (Y2) axis scale.

Default: Y

Interaction: This option has effect only if YSPACE=DATAVALUE.

YSPACE= GRAPHPERCENT | GRAPHPIXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALUE
specifies the drawing space and drawing units for interpreting the value that is specified in the Y= option.

Default: The setting that is in effect for the DRAWSPACE= option.

Part 9

Attribute Maps

Chapter 72

DISCRETEATTRMAP Statement

Description and Syntax

Creates associations between discrete data values and graphical properties. The associations can be used to control the graphical properties that correspond to values from an input data column or that are referenced directly in a discrete legend.

DISCRETEATTRMAP NAME="*string*" <*/option(s)*>;

VALUE statement;

<…*VALUE statement;*>

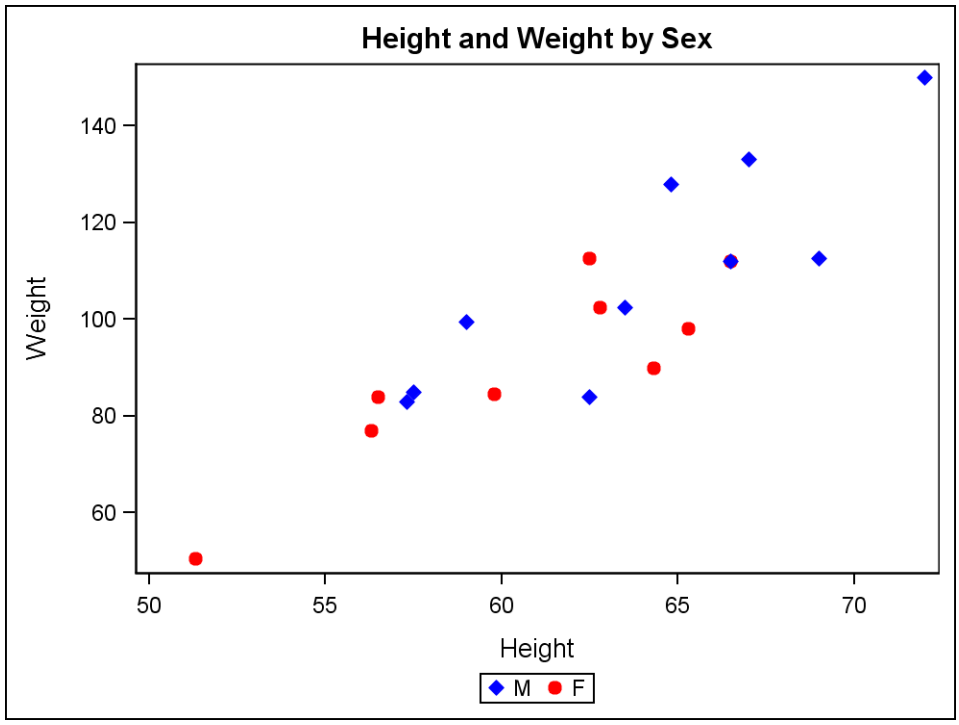
ENDDISCRETEATTRMAP;

Example Program and Statement Details

Example Graph

The following graph was generated by the "Example Program" on page 806. The example defines graphical properties to associate with classification values in an input column that is used in a scatter plot. The DISCRETEATTRMAP statement starts the

attribute map definition, assigns a name to it, and ensures that the data mapping is not case sensitive. The VALUE statements define the colors and marker symbols to associate with the values M and F. The DISCRETEATTRVAR statement associates the attribute map with the data column SEX and assigns the name GROUPMARKERS to the association. The SCATTERPLOT statement references the named association in its GROUP= option.



Example Program

```
proc template;
  define statgraph scatterplot;
    begingraph;
      entrytitle "Height and Weight by Sex";

      /* define the attribute map and assign the name "symbols" */
      discreteattrmap name="symbols" / ignorecase=true ;
        value "m" / markerattrs=(color=blue symbol=diamondfilled) ;
        value "f" / markerattrs=(color=red  symbol=circlefilled) ;
      enddiscreteattrmap ;

      /* associate the attribute map with input data column SEX and assign
       * the name GROUPMARKERS to the named association */
      discreteattrvar attrvar=groupmarkers var=sex attrmap="symbols" ;

      /* reference GROUPMARKERS in a plot statement */
      layout overlay;
        scatterplot x=height y=weight / name="scatter"
          group=groupmarkers ;
        discretelegend "scatter";
      endlayout;
    endgraph;
```

```
  end;

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

Statement Summary

The DISCRETEATTRMAP statement creates an attribute map that matches graphical properties to discrete values. The attribute map can be associated with a data input column that is used as a classification variable in a graph. Or it can be specified directly in a discrete legend.

Attribute maps can be useful for ensuring that a particular value (a company name, for example) is always represented by the same visual characteristics in your graphs, regardless of the value's order in the input data. When specified directly in a discrete legend, an attribute map can be used to display legend entries for group values that are not in the data. (This point is discussed in more detail in a moment.)

Note: If you do not want to manage the graphical properties that are associated with each unique discrete value, you can simply specify an input column in the grouping option. For example, in the "Example Program" on page 806, you can specify input column SEX directly in the SCATTERPLOT statement's GROUP= option. In that case, each discrete value is still represented by different graphical properties in the graph. The default properties are derived from options that are set in the plot statement or from the GraphData1 through GraphDataN style elements that are defined in the ODS style that is in effect. However, to ensure that specific graphical properties are used to represent a discrete value in the graph, regardless of that value's order in the data, you can use an attribute map to create that association. Any values in the data that are not accounted for in the attribute map are assigned the graphical properties that they would receive if the attribute map is not defined.

Defining and using a discrete attribute map requires you to coordinate settings on several statements:

- Use the DISCRETEATTRMAP statement to start the attribute definition and assign a name to it. The DISCRETEATTRMAP statement determines whether the data mapping is case-sensitive and whether leading blanks are trimmed from the data values during the mapping.
- Nest within the DISCRETEATTRMAP block at least one VALUE statement that specifies graphical properties to associate with a classification value. Use one VALUE statement for each classification value that you want to map.
- Use the DISCRETEATTRVAR statement to create a named association between the defined attribute map and the input column that contains the classification values. The ATTRMAP= argument identifies the attribute map and the VAR= argument identifies the input column. Use the ATTRVAR= argument to assign a name that can be used to reference the named association in plot statements within the template.
- Reference the attribute map where needed.

  In plot statements, reference the attribute map by specifying the name that is assigned in the DISCRETEATTRVAR statement's ATTRVAR= argument. Each plot statement's documentation in this reference indicates which options support an attribute map as the specified value.

  In a DISCRETELEGEND statement, you can reference the plot statement that uses the attribute map. The plot statement must have a NAME= option that assigns a name to the plot, and then the DISCRETELEGEND statement references that name.

Because the attribute map is referenced in the plot statement, the legend uses the attribute map to represent the group values that are present in the data.

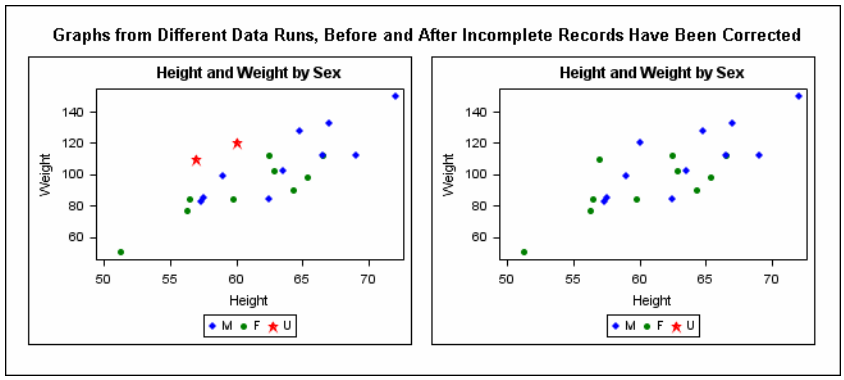
The DISCRETEATTRMAP statement must be located within the BEGINGRAPH block but outside of the outermost layout block. It cannot be nested in another DISCRETEATTRMAP statement or in a RANGEATTRMAP statement. The DISCRETEATTRMAP block must contain at least one VALUE statement. Values that are referenced in the attribute map can be character or numeric. Discrete values that are not accounted for in the attribute map are assigned the graphical properties that they would receive if the attribute map is not defined.

Displaying Legend Entries for Group Values That Are Not in the Data. To display legend entries for the values in a grouped plot, you typically use the plot statement's NAME= option to assign a name to the plot, and then reference that name in the DISCRETELEGEND statement. In this usage case, the legend displays entries for the group values that are present in the data. The legend does not display entries for group values that are not present in the data.

To represent group values in the legend, regardless of whether those values are present in the data, you can define a discrete attribute map for all of the group values that you want to represent. You can then specify the attribute map directly in the DISCRETELEGEND statement. In this usage case, the DISCRETELEGEND statement must also specify the TYPE= option to identify the type of legend entries to display.

Note: An attribute map cannot be specified directly in a MERGEDLEGEND statement.

Referencing an attribute map directly in a DISCRETELEGEND statement can be useful for flagging data in the graph. For example, assume you have weight and height values for all students in an analysis group, but some of the observations are incomplete and do not record the student's sex. In that case, you could represent the unknown values by entering the value U in the data. You could then define an attribute map that matches string value "U" to specified graphical properties. Reference the attribute map in your plot statement that the attribute map is used in the plot. In the DISCRETELEGEND statement, do not reference the plot statement. Instead, reference the attribute map directly. As a result, the legend displays the attribute-map definitions. Observations with value U for SEX are effectively flagged as incomplete observations. After the problem records have been corrected, the absence of the flag in the plot indicates that the value U is no longer present in the data, so all records are now complete.




```
begingraph;
  entrytitle "Height and Weight by Sex";

  discreteattrmap name="symbols" / ignorecase=true trimleading=true;
    value "m" / markerattrs=(color=blue symbol=diamondfilled);
    value "f" / markerattrs=(color=green  symbol=circlefilled);
    value "u" / markerattrs=(color=red symbol=starfilled) ;
```

```
  enddiscreteattrmap;

  discreteattrvar attrvar=groupmarkers var=sex attrmap="symbols";

  layout overlay;
    scatterplot x=height y=weight / name="scatter"
      group=groupmarkers ;
    discretelegend "symbols" / type=marker ;
  endlayout;
endgraph;
```

When referencing an attribute map directly in a DISCRETELEGEND statement, be aware of the following points:

- The legend entries are defined entirely by the attribute map and are independent of the data. To correlate those legend entries with the data representations in the graph, you must also specify the attribute map in the appropriate plot statement(s).
- The TYPE= option is required to specify the type of legend entries to display. If TYPE= is not specified or is set to the default TYPE=ALL, then the DISCRETELEGEND statement ignores the reference to the attribute map.

Required Argument

NAME="*string* "
assigns a name to the attribute definition. The name can be referenced in a DISCRETEATTRVAR statement, which is used to associate the attribute map with an input data column. The name can also be referenced in a DISCRETELEGEND statement to map the specified graphical properties directly to a discrete legend.

Restriction: The *string* is case sensitive, must be enclosed in quotation marks, and must define a unique name within the template.

Options

IGNORECASE= *boolean*
specifies whether case is ignored when comparing the values that are specified in the attribute map with values from an input data column.

Default: FALSE. Value comparisons are case-sensitive.

Tip: The effect of this option can be achieved by applying a function like UPCASE to the data column and using only uppercase strings in each VALUE statement.

TRIMLEADING= *boolean*
specifies whether leading blanks are trimmed (removed) from both the attribute map values and the input data values before those values are compared. Trailing blanks are always trimmed.

Default: TRUE. Leading blanks are trimmed.

VALUE Statement

VALUE Statement Summary

A VALUE statement within the DISCRETEATTRMAP block associates graphical properties to with a discrete value in the attribute map. To associate graphical properties with multiple values, specify multiple VALUE statements in the attribute map, using the following general syntax for each of the statements:

VALUE *value-spec* <*/option(s)*>;

For a specific example, see the "Example Program" on page 806.

If the discrete attribute map is referenced by a plot statement in the template, the graphical properties that are defined in the VALUE statements are used in the plot. If a discrete legend is generated for the plot, the graphical properties are represented in that legend.

If a discrete attribute map is referenced directly in a DISCRETELEGEND statement, the graphical properties that are defined in the VALUE statement are mapped directly to the legend and are independent of the values in the data. For more information and an example, see the "Statement Summary" on page 807 .

If two or more VALUE statements define attributes to associate with the same classification values, the last VALUE statement's settings are used.

By default when comparing a column's value to a string that is specified for the VALUE statement's *value-spec*,

- the column value is formatted to a string, using the format that is defined for the column or the default format if no format has been define for the column
- leading spaces are trimmed from the string that is specified in the VALUE statement
- a case-sensitive comparison is performed between the column string and the VALUE string.

To change the default behavior for the comparison, you can use the DISCRETEATTRMAP statement's TRIMLEADING= and IGNORECASE= options.

VALUE Statement Argument

value-spec
specifies one or more formatted strings or the keyword OTHER. Strings are always quoted. Multiple strings must be separated by blanks, and each of the strings must be enclosed in its own set of quotation marks. The formatted strings must be equal to the formatted values of the classification column that is used with the DISCRETEATTRVAR statement.

OTHER
Creates a category for all other column values that are not explicitly assigned with VALUE statements. This keyword is not quoted. The default attributes for these values are derived from the GraphOther style element.

Restriction: If a user-defined format is associated with the classification column, you should specify the same formatted strings that appear in the format definition.

The following examples elaborate on the *value-spec* strings:

"Hybrid"	By default, all string comparisons are case-sensitive. By default, the string *Hybrid* does not match the string *HYBRID*.
"HYBRID"	If IGNORECASE=TRUE in the DISCRETEATTRMAP statement, you can specify an upper-, lower-, or mixed-cased string for the *value-spec* string. When IGNORECASE=TRUE, the string *Hybrid* matches the string *HYBRID*.
"15JAN2011"	If a numeric column is being mapped with a VALUE statement, you must specify the formatted value of the column. This example shows how to specify the value of a numeric SAS date column that has a DATE9. format associated with it.
"."	If a numeric column has a missing value, you should use the formatted value for missing, which is "." by default. If the MISSING= system option is used to change the default string, you should match that value. For example, if OPTIONS MISSING="M" is specified in the SAS program, you should use "M" in the VALUE statement to represent missing values.
" "	If a character column has a missing value, you should use the formatted value for missing, which is " " by default.
"Truck" "SUV"	Multiple strings can be specified to indicate that each of the specified values matches to the same graphical properties. It does not mean that a single new category is formed. The list of strings is separated by blanks, and each string is enclosed in its own set of quotation marks.

VALUE Statement Options

FILLATTRS=*style-element* | *style-element* (*fill-options*) | (*fill-options*)
the fill attributes to use when an attribute map is applied to filled areas in a graph. See "General Syntax for Attribute Options" on page 849 for the syntax on using a *style-element* and "Fill Options" on page 850 for available *fill-options*.

Default:

- If the attribute map is used in a plot, unspecified attributes receive the attributes that they would have if the attribute map were not defined.
- If the attribute map is used directly in a legend, unspecified attributes derive attributes from the GraphDataDefault style element.

Restriction: This option does not support the TRANSPARENCY= setting in the specified fill attribute.

LINEATTRS=*style-element* | *style-element* (*line-options*) | (*line-options*)
specifies the line attributes to use when an attribute map is applied to lines in a graph. See "General Syntax for Attribute Options" on page 849 for the syntax on using a *style-element* and "Line Options" on page 850 for available *line-options*.

Default:

- If the attribute map is used in a plot, unspecified attributes receive the attributes that they would have if the attribute map were not defined.
- If the attribute map is used directly in a legend, unspecified attributes derive attributes from the GraphDataDefault style element.

Restriction: This option does not support the THICKNESS= setting in the specified line attribute.

MARKERATTRS=*style-element* | *style-element* (*marker-options*) | (*marker-options*)
specifies the marker attributes to use when an attribute map is applied to marker symbols in a graph. See "General Syntax for Attribute Options" on page 849 for the syntax on using a *style-element* and "Marker Options" on page 851 for available *marker-options*.

Default:

- If the attribute map is used in a plot, unspecified attributes receive the attributes that they would have if the attribute map were not defined.
- If the attribute map is used directly in a legend, unspecified attributes derive attributes from the GraphDataDefault style element.

Restriction: The SIZE=, TRANSPARENCY=, and WEIGHT= suboptions are ignored.

DISCRETEATTRVAR Statement

DISCRETEATTRVAR Syntax

Creates a named association between an attribute map of discrete values and an input data column.

DISCRETEATTRVAR ATTRVAR=*attrvar-name*
VAR=*data-column* | *expression* | *dynamic*
ATTRMAP="*attrmap-name*";

For a specific example, see the "Example Program" on page 806.

DISCRETEATTRVAR Statement Summary

The DISCRETEATTRVAR statement creates and names an association between graphical properties that are specified in a DISCRETEATTRMAP block and a classification column that is in the data. The name that is assigned to the association in the DISCRETEATTRVAR statement is the name that plot statements must reference to use the attribute map.

Defining and using a discrete attribute map requires you to coordinate settings on several statements. For more information, see the DISCRETEATTRMAP statement's "Statement Summary" on page 807 .

The DISCRETEATTRVAR statement must be located within the BEGINGRAPH block but outside of the outermost layout block. It cannot be nested in a DISCRETEATTRMAP statement.

DISCRETEATTRVAR Statement Required Arguments

ATTRVAR= *attrvar-name*
specifies a SAS name for this association between the attribute map and the input column. This name must be unique within the template and can be referenced by

other statements that can be associated with the attribute map. The *attrvar-name* should not be used in an expression. If it is, the results are unpredictable.

Note: The assigned SAS name can be the same as the name of the data input column, but it is not recommended. If an assigned *attrvar-name* matches the name of an input data column, the *attrvar-name* takes precedence.

Restriction. The name that is assigned in this argument is used to associate an attribute map with the discrete values in an input data column. Thus, it is not the name to reference when you want to display legend entries that are independent of the data. For that special use, a DISCRETELEGEND statement can reference the attribute map directly by the name that is assigned in the DISCRETEATTRMAP statement. For more information, see the DISCRETEATTRMAP statement's "Statement Summary" on page 807 .

VAR= *data-column* | *expression* | *dynamic*
specifies an input data column to be associated with an attribute map at run time. If an *expression* is used, a new column of transformed values is created and then matched with the attribute map.

Interaction. If the column is not found or the column is of the wrong type for the attribute map, the DISCRETEATTRVAR statement is ignored.

Tip. The input data column can be character or numeric, but the values must match the type of the values that are specified in the attribute map. For numeric columns, all values are treated as discrete values.

ATTRMAP="*attrmap-name* "
specifies the name of an existing discrete attribute map.

Restriction: The *attrmap-name* is case sensitive, must be enclosed in quotation marks, and must be the name that was assigned to the attribute map in the DISCRETEATTRMAP statement's NAME= argument.

Chapter 73

RANGEATTRMAP Statement

Description and Syntax

Creates an attribute map that matches colors to numeric values or numeric ranges so that the colors can be associated with the values of an input data column.

RANGEATTRMAP NAME="*string*";

RANGE statement;

<…*RANGE statement;*>

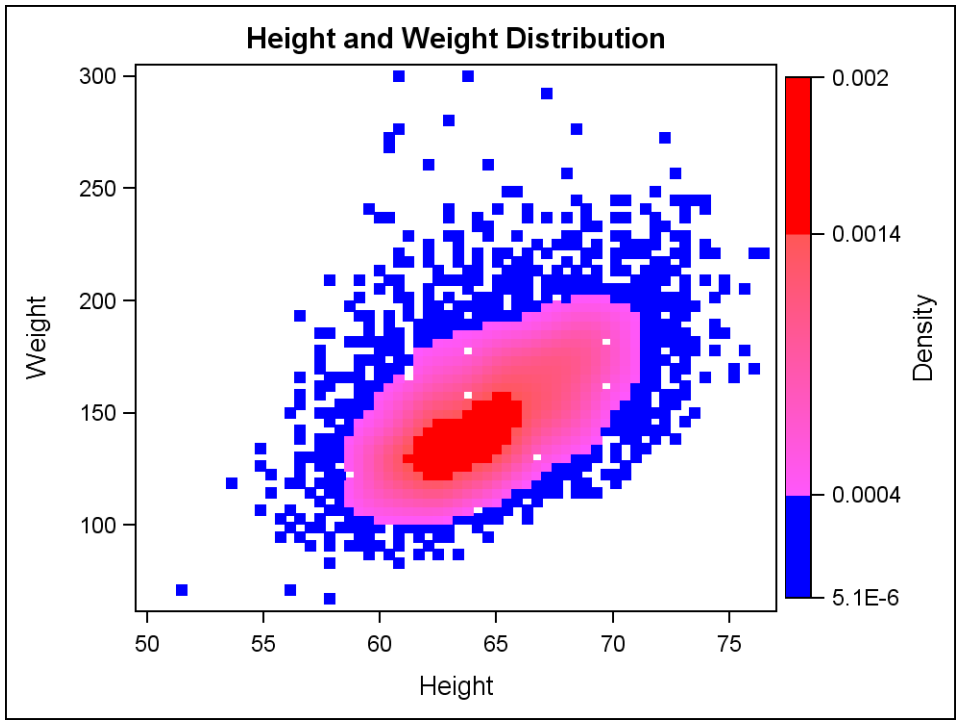
ENDRANGEATTRMAP;

Example Program and Statement Details

Example Graph

The following graph was generated by the "Example Program" on page 816. The example defines colors to associate with data ranges in an input column that is used in a scatter plot. The RANGEATTRMAP statement starts the attribute map definition and assigns a name to it. The RANGE statements define the value ranges and the colors to

associate with those ranges. The highest value specified, 0.002, does not have to be an actual value in the data. The RANGEATTRVAR statement associates the attribute map with the data column DENSITY and assigns the name RANGEVAR to the association. The SCATTERPLOT statement references the named association in its MARKERCOLORGRADIENT= option.



Example Program

```
proc template;
define statgraph attrmap;
  begingraph;
    entrytitle "Height and Weight Distribution" ;

    /* define the attribute map and assign the name "densityrange" */
    rangeattrmap name="densityrange" ;
      range MIN - 0.0004      / rangealtcolor=blue ;
      range 0.0004 < - 0.0014 / rangealtcolormodel=(lightpurple lightred) ;
      range 0.0014 < - 0.002  / rangealtcolor=red ;
    endrangeattrmap ;

    /* associate the attribute map with     input data column DENSITY and
     * assign variable name RANGEVAR to the named association */
    rangeattrvar attrvar=rangevar var=density attrmap="densityrange" ;

    /* reference the RANGEVAR attribute map in a plot statement */
    layout overlay;
      scatterplot x=height y=weight / markercolorgradient=rangevar
        markerattrs=(symbol=squarefilled size=6px) name="scatter";
      continuouslegend "scatter" / orient=vertical
        halign=right title="Density";
    endlayout;
```

```
  endgraph;
end;

proc sgrender data=sashelp.gridded(where=(count>0)) template=attrmap;
run;
```

Statement Summary

The RANGEATTRMAP statement creates an attribute map that matches colors to numeric values or value ranges. The attribute map can be associated with a data input column that uses color to represent response values in a graph. Attribute maps can be useful for controlling the application of gradient color in a graph. In addition, they enable you to map color to data values, independent of the actual data that is used in the graph. For example, for temperature data, you can set Blue for 0 and Red for 100, even if the values 0 and 100 are not in the data.

Defining and using the attribute map requires you to coordinate settings on several statements:

- Use the RANGEATTRMAP statement to start the attribute definition and assign a name to it.
- Nest within the RANGEATTRMAP block at least one RANGE statement that specifies a numeric value or numeric range and the color to associate with that value or range. Use one RANGE statement for each value range that you want to map. You can use keywords like MIN and MAX in the range specification. For complete details about the range specifications, see the RANGE statement.
- Use the RANGEATTRVAR statement to create a named association between the defined attribute map and the input column that contains the numeric values. The ATTRMAP= argument identifies the attribute map and the VAR= argument identifies the input column. Use the ATTRVAR= argument to assign a name that can be used to reference the named association in plot statements within the template.
- Reference the attribute variable in one or more plot statements by specifying the name that is assigned in the RANGEATTRVAR statement's ATTRVAR= argument. Each plot statement's documentation in this reference indicates which options support an attribute variable as the specified value.

  In a CONTINUOUSLEGEND statement, reference the plot statement that uses the attribute map. The plot statement must have a NAME= option that assigns a name to the plot, and then the CONTINUOUSLEGEND statement references that name. Because the attribute map is referenced in the plot statement, the legend uses that attribute map to represent the numeric values that are present in the data.

The RANGEATTRMAP statement must be located within the BEGINGRAPH block but outside of the outermost layout block. It cannot be nested in another RANGEATTRMAP statement or in a RANGEATTRMAP statement. The RANGEATTRMAP block must contain at least one RANGE statement. All values that are referenced in the attribute map must be numeric. Data values that are not accounted for in the attribute map receive the default color from the GraphOther, GraphOverflow, or GraphUnderflow style element, depending on where the unassigned values are relative to the specified data ranges.

Note: Unlike a DISCRETEATTRMAP statement, a RANGEATTRMAP statement cannot be directly referenced in a legend. This is because the RANGE statements can specify keywords like MIN and MAX that require a data association for interpretation.

Required Argument

NAME="*string* "
assigns a name to the attribute definition for reference in a RANGEATTRVAR statement.

Restriction: The *string* is case sensitive, must be enclosed in quotation marks, and must define a name that is unique among RANGEATTRMAP names within the template.

RANGE Statement

RANGE Statement Summary

A RANGE statement within the RANGEATTRMAP block matches color to a numeric value, or a color ramp to a numeric range. To match colors to multiple values or ranges, specify multiple RANGE statements, using the following general syntax for each of the statements.

RANGE *range-spec* </option(s)>;

For a specific example, see the "Example Program" on page 816.

When defining multiple RANGE statements, be careful not to define conflicting ranges. A RANGE statement might be syntactically correct by itself while nevertheless conflicting with settings in another RANGE statement.

RANGE Statement Argument

range-spec
specifies a range of numeric values or a keyword, such as OTHER or MISSING.

A range of numeric values is specified in the form *low-value* - *high-value*. Both *low-value* and *high-value* can be specified as an unformatted numeric value. A less-than symbol (<) can be placed after the low numeric value, before the high numeric value, or in both positions to exclude that value from the range endpoint (similar to the VALUE statement of the FORMAT procedure). If you are excluding the first value in a range, put the < after the *low-value*. If you are excluding the last value in a range, put the < before the *high-value*. You can also exclude both the low and the high value.

For example, the following range does not include 0:

```
0 < - 100
```

Likewise, the following range does not include 100:

```
0 - < 100
```

If a value at the high end of one range also appears at the low end of another range and you do not use the < exclusion notation, then the value is assigned to the first range.

If two or more RANGE statements define colors to associate with the same numeric values or ranges, the first RANGE statement's settings are used. If any RANGE

statement's range overlaps another RANGE statement's range (for example, 10 - 20 and 15 - 25), then the entire attribute map is ignored and default coloring is used.

If two ranges share a common endpoint (for example, 10 - 20 and 20 - 30) and no exclusion operator is used, the common endpoint belongs to the lower encountered range (10 - 20 in this case). The order of the specification does not matter.

To set a single numeric value, specify the same value for both *low-value* and *high-value*.

If *low-value* is not less than or equal to *high-value*, the *range-spec* is invalid and the RANGE statement is ignored in the attribute map definition.

Note: If a range is not defined for keyword OTHER, gaps within the attribute map ranges are assigned the default color that is defined by the GraphOther:ContrastColor style reference.

Rather than using a numeric value, you can specify one of the following keywords as *low-value* or *high-value*:

MIN	Indicates the minimum data value for column values.
MAX	Indicates the highest data value for the column values.
MAXABS	Indicates max(abs(MIN) , abs(MAX))
NEGMAX	Indicates -MAX
NEGMAXABS	Indicates -max(abs(MIN) , abs(MAX))

Rather than specifying a *low-value* to *high-value* range, you can use one of the following keywords for *range-spec*:

MISSING	Indicates a mapping for missing values. The visual attributes for this setting are obtained from the GraphMissing style element. If one RANGE statement specifies this value and another RANGE statement specifies keyword OTHER, then the OTHER range does not include missing values.
UNDER	Creates a range for all data values between the lowest mapped value and the lowest actual data value. The visual attributes for this setting are obtained from the GraphUnderflow style element. If one RANGE statement specifies this value and another RANGE statement specifies keyword OTHER, then the OTHER range does not include underflow values.
OVER	Creates a range for all data between the highest mapped value and the highest actual data value. The visual attributes for this setting are obtained from the GraphOverflow style element. If one RANGE statement specifies this value and another RANGE statement specifies keyword OTHER, then the OTHER range does not include overflow values.
OTHER	Creates a category for all other column values not explicitly assigned to a range. The OTHER values can be composed of several non-contiguous ranges. The visual attributes for this setting are obtained from the GraphOther style element.

RANGE Statement Options

RANGEALTCOLOR=*style-reference* | *color* | GRADIENTSTEPPER(*color1,color2, num-steps, step*)
specifies a single contrast color to represent the defined value range.

Note: This feature is for the first maintenance release of SAS 9.3 and later. See "What's New in the SAS 9.3 Graph Template Language" on page xi.

Default: The GraphDataDefault:ContrastColor style reference.

GRADIENTSTEPPER (*color1,color2, num-steps, step*)
A gradient stepper partitions a color range into equal-sized intervals and returns the color that is in the specified step position. The start and end colors for the range are specified in parameters *color1* and *color2*. The number of equal-sized intervals is specified in parameter *num-steps*, and the step position for the color to return is specified in parameter *step*. Example:

```
rangeattrmap name="incomemap";
  range min - 13000    / rangealtcolor=gradientstepper(red,green,4,1);
  range 13000 < - 25000 / rangealtcolor=gradientstepper(red,green,4,2);
  range 25000 < - 50000 / rangealtcolor=gradientstepper(red,green,4,3);
  range 50000 < - max   / rangealtcolor=gradientstepper(red,green,4,4);
endrangeattrmap;
```

Interaction: If this option is specified, the RANGEALTCOLORMODEL= option is ignored.

RANGECOLOR=*style-reference* | *color* | GRADIENTSTEPPER(*color1,color2, num-steps, step*)
specifies a single color to represent the defined value range.

Default: The GraphDataDefault:Color style reference.

GRADIENTSTEPPER (*color1,color2, num-steps, step*)
A gradient stepper partitions a color range into equal-sized intervals and returns the color that is in the specified step position. The start and end colors for the range are specified in parameters *color1* and *color2*. The number of equal-sized intervals is specified in parameter *num-steps*, and the step position for the color to return is specified in parameter *step*. Example:

```
rangeattrmap name="incomemap";
  range min - 13000    / rangecolor=gradientstepper(red,blue,4,1);
  range 13000 < - 25000 / rangecolor=gradientstepper(red,blue,4,2);
  range 25000 < - 50000 / rangecolor=gradientstepper(red,blue,4,3);
  range 50000 < - max   / rangecolor=gradientstepper(red,blue,4,4);
endrangeattrmap;
```

Interaction: If this option is specified, the RANGECOLORMODEL= option is ignored.

RANGEALTCOLORMODEL=*style-element* | (*list-of-colors*)
specifies a style element or a list of one or more specific contrast colors to represent the defined value range.

Note: This feature is for the first maintenance release of SAS 9.3 and later. See "What's New in the SAS 9.3 Graph Template Language" on page xi.

Default: no default

style-element
The name of a style element. To display the range as a gradient ramp, choose a style element such as TwoColorRamp, TwoColorAltRamp, ThreeColorRamp, or ThreeColorAltRamp. The style element should contain the following style attributes:

STARTCOLOR	specifies a color for the smallest data value.
NEUTRALCOLOR	specifies a color for the midpoint of the data range. This attribute is not needed when defining a two-color ramp.
ENDCOLOR	specifies a color for the highest data value.

To display the range or single value contrast color as a color that is defined in a style, use a *style-reference* (for example, **`GraphData1:color`**) to refer to a color attribute. The following style references correspond to the keywords that are available in this statement's *range-spec* argument:

***range-spec* Keyword**	**Corresponding Style Reference**
MISSING	GraphMissing:ContrastColor
OTHER	GraphOther:ContrastColor
UNDER	GraphUnderflow:ContrastColor
OVER	GraphOverflow:ContrastColor

(*list-of-colors*)
A list of two or more color keywords that are enclosed in parentheses and separated by a blank.

Two colors create the endpoints of a ramp. The first color is assigned to *low-value* in *range-spec*, and the second color is assigned to *high-value*. Three or more colors partition *range-spec* into *n*–1 equal-sized intervals, with each adjacent color pair defining a two-color ramp.

Interaction: This option is ignored if the RANGEALTCOLOR= option is specified.

RANGECOLORMODEL=*style-element* | (*list-of-colors*)
specifies either a style element or a list of one or more specific colors to represent the defined value range in this *range-spec* argument.

Default: no default

style-element
The name of a style element. To display the range as a gradient ramp, choose a style element such as TwoColorRamp, TwoColorAltRamp, ThreeColorRamp, or ThreeColorAltRamp. The style element should contain the following style attributes:

STARTCOLOR	specifies a color for the smallest data value.
NEUTRALCOLOR	specifies a color for the midpoint of the data range. This attribute is not needed when defining a two-color ramp.
ENDCOLOR	specifies a color for the highest data value.

To display the range or single value as a color that is defined in a style, use a *style-reference* (for example, **`GraphData1:color`**) to refer to a color attribute. The following style references correspond to the keywords that are available in this statement's *range-spec* argument:

***range-spec* Keyword**	**Corresponding Style Reference**
MISSING	GraphMissing:Color
OTHER	GraphOther:Color
UNDER	GraphUnderflow:Color
OVER	GraphOverflow:Color

(*list-of-colors*)
A list of two or more color keywords, enclosed in parentheses and separated by blanks.

Two colors create the endpoints of a ramp. The first color is assigned to *low-value* in *range-spec*, and the second color is assigned to *high-value*. Three or more colors partition *range-spec* into *n*–1 equal-sized intervals, with each adjacent color pair defining a two-color ramp.

Interaction: This option is ignored if the RANGECOLOR option is specified.

Specifying Color in a RANGE Option

For specifying a color in one of the RANGE options, the color keywords can be any of the following:

- a SAS color name (for example, BLUE)
- an RGB name (for example, CX0000FF or x0000FF)
- an HLS value (for example, H14E162D)
- a gray-scale color code (for example, GRAYBB)
- an HTML color name (for example, AZURE)
- a SAS session color (for example, DMSBLUE).

RANGEATTRVAR Statement

RANGEATTRVAR Statement Syntax

Creates a named association between a range attribute map of numeric values or value ranges and an input data column.

RANGEATTRVAR ATTRVAR=*attrvar-name*
VAR=*data-column* | *expression* | *dynamic*
ATTRMAP="*attrmap-name*";

For a specific example, see the "Example Program" on page 816.

RANGEATTRVAR Statement Summary

The RANGEATTRVAR statement creates and names an association between colors that are specified in a RANGEATTRMAP block and a numeric column that is in the data. Attribute maps can be useful for controlling the application of gradient color in a graph or specifying data values that are independent of the actual data. The name that is assigned to the association in the RANGEATTRVAR statement is the name that plot statements must reference to use the attribute map.

Defining and using a numeric-range attribute map requires you to coordinate settings on several statements. For more information, see the RANGEATTRMAP statement's "Statement Summary" on page 817 .

The RANGEATTRVAR statement must be located within the BEGINGRAPH block but outside of the outermost layout block. It cannot be nested in a RANGEATTRMAP statement.

RANGEATTRVAR Statement Required Arguments

ATTRVAR= *attrvar-name*
specifies a SAS name for the map. This name must be unique among the RANGEATTRVAR statements within the template. The assigned name can be referenced by other statements that can be associated with the range attribute map.the *attrvar-name*

VAR= *data-column* | *expression* | *dynamic*
specifies a numeric input data column to be associated with an attribute map at run time. If an *expression* is used, a new column of transformed values is created and then matched with the attribute map.

Interaction. If the column is not found or the column is of the wrong type for the attribute map, the RANGEATTRVAR statement is ignored.

ATTRMAP="*attrmap-name* "
specifies the name of an existing range attribute map.

Restriction: The *attrmap-name* is case sensitive, must be enclosed in quotation marks, and must be the name that was assigned to the attribute map in the RANGEATTRMAP statement's NAME= argument.

Part 10

Runtime Programming Features

Chapter 74

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.

The values for a dynamic variable do not have to be provided by the data source. Rather, you can provide the values in the DYNAMIC statement in PROC SGRENDER, specifying the values as a space delimited list, enclosed in quotation marks. Do not use parentheses in the specification.

In the following example, the graph template specifies a dynamic variable named TICKS, which is referenced on the XAXISOPTS= option in LAYOUT OVERLAY. The DYNAMIC statement in PROC SGRENDER provides values for TICKS:

```
proc template;
  define statgraph regress;
    dynamic TICKS ;
    begingraph;
      layout overlay /xaxisopts=(linearopts=(tickvaluelist=TICKS));
        scatterplot x=age y=weight;
      endlayout;
    endgraph;
  end;
run;

proc sgrender data=sashelp.class template=regress;
  dynamic TICKS="11 13 16" ;
run;
```

If your template uses a dynamic variable to specify a required attribute, such as a variable name, and the name is misspelled or is not provided in the SGRENDER procedure, then a warning is issued and the respective plot statement drops out of the

final graph. A graph is produced, but it might be a blank graph, or it might show the results of all statements except those that are in error.

For more information about using dynamics and macro variables in your templates, see *SAS Graph Template Language: User's Guide*.

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 75

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 76

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. **Limitation:** only one sort operation (whether an ASORT() or DSORT() function) can be used within a single template definition.
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. **Limitation:** only one sort operation (whether an ASORT() or DSORT() function) can be used within a single template definition.
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

number = EVAL(function-name(numeric-column))	
Function Name	**Description**
Q3	Third quartile
QRANGE	Interquartile range
RANGE	Range
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 77

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 that are 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 11

Appendices

Appendix 1

Syntax Conventions and Argument Value Types

Syntax Conventions

< > ==> optional

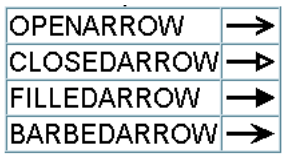
a | b ==> either a or b

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.



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. 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).

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 75, "Expressions," on page 831.

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 850.

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 851.

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 850, line-options on page 850, marker-options on page 851, or text-options on page 852.

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: User's 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 850, line-options on page 850, marker-options on page 851, or text-options on page 852.

Fill Options

When specifying the attributes for an area fill, the *fill-options* can be one or more of the following settings. 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 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).

TRANSPARENCY=*number*
specifies the degree of the transparency of the filled area. This setting enables you to set the transparency for the filled elements of some graph types. You can set just this fill transparency, or set the fill independently of the other transparent elements in the graph. For example, you can use this setting to set the transparency level for the filled bars of a bar chart, and use the bar chart's DATATRANSPARENCY= option to set a different transparency level for the bar outlines.

Default: The same as the setting of the statement's DATATRANSPARENCY= option.

Range: 0 (opaque) to 1 (entirely transparent)

Interaction: This setting overrides the statement's DATATRANSPARENCY= setting for the fills but not for the outlines.

Example: fillattrs=(transparency=0.5)

Line Options

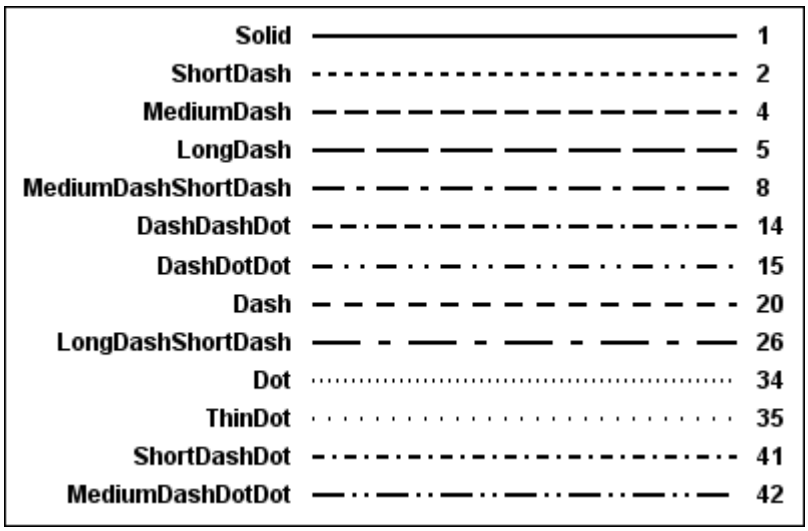
When specifying the attributes for a line, the available *line-options* can be any one or more of the following settings. The options must be enclosed in parentheses, and each option is specified as a *name = value pair*. In all cases, the *value* can be a *style-reference* in the form *style-element:style-attribute*.

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 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).

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 853 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.

Note: Anti-aliasing might alter the appearance of some line patterns that have fine detail such as line patterns 33 through 46. For example, if you specify the color black and the pattern 33 for a line, and anti-aliasing is enabled, the line might appear gray. In that case, you can use the following command to disable anti-aliasing in order to show the line detail:

```
ods graphics / antialias=off;
```

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 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).

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	X	SquareFilled
Asterisk	Ibeam	Triangle	Y	StarFilled
Circle	Plus	TriangleDown	Z	TriangleFilled
Diamond	Square	TriangleLeft	CircleFilled	TriangleDownFilled
GreaterThan	Star	TriangleRight	DiamondFilled	TriangleLeftFilled
Hash	Tack	Union	HomeDownFilled	TriangleRightFilled

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 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).

FAMILY =*style-reference* | *"string"*
specifies the font family of the text. If you use a *style-reference*, the *style-attribute* should be FONTFAMILY.

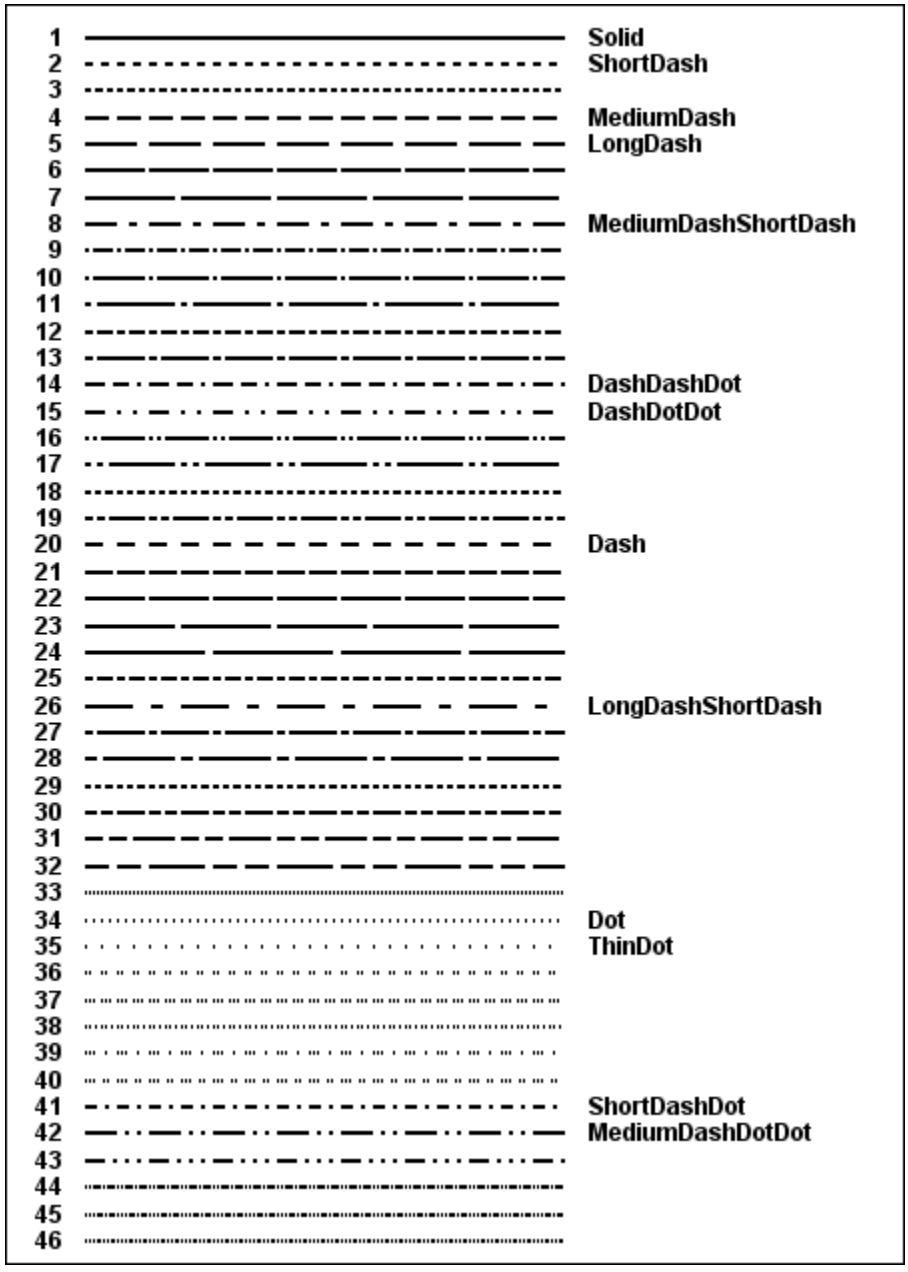
SIZE=*style-reference* | *dimension*
specifies the font size of the text. If you use a *style-reference*, the *style-attribute* should be FONTSIZE.

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.



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. Also, 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
```

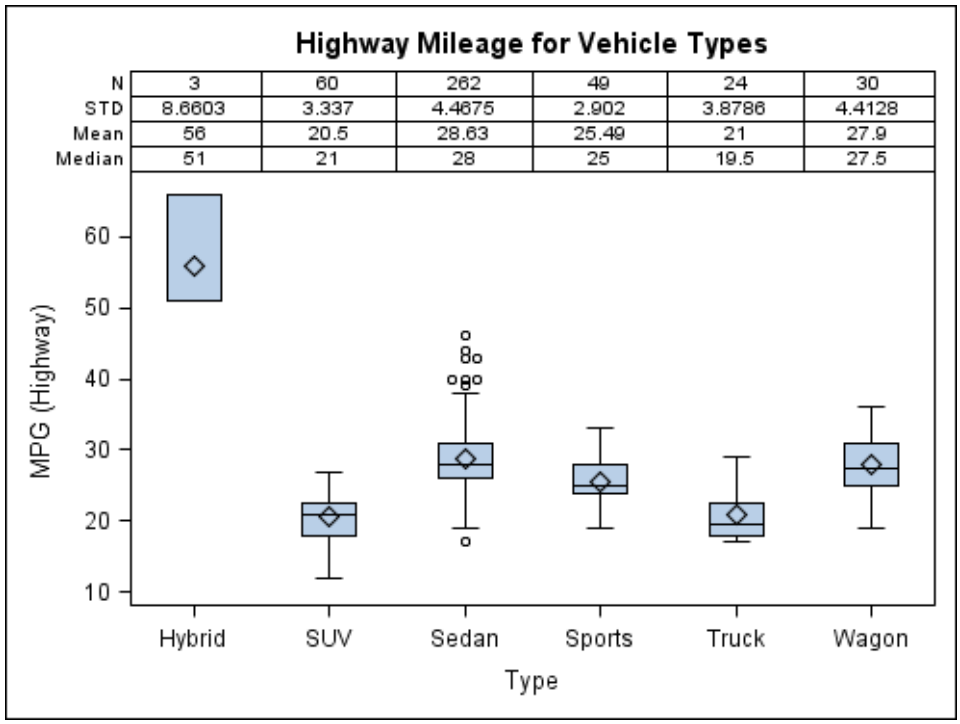
```
        in the same values produced by the BOXPLOT
        statement, however they can be modified to
        satify other statistical definitions */
      if value <= upperFence then tempmax=value;
      else tempmax=.;
      if value => lowerFence then tempmin=value;
      else tempmin=.;
  end;
  /* 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 233 of "BOXPLOTPARM Statement" on page 233 .

```
%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.




```
%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 overlay / xaxisOpts=(offsetMin=0.08 offsetMax=0.08);
        innerMargin / align=top;
          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;
        endInnerMargin;
        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 set of layout cells that are stacked vertically and share 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

C

D

G

H

I

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Z