

SAS[®] 9.3 ODS Graphics Procedures Guide

Third Edition



The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2012. *SAS® 9.3 ODS Graphics: Procedures Guide, Third Edition*. Cary, NC: SAS Institute Inc.

SAS® 9.3 ODS Graphics: Procedures Guide, Third Edition

Copyright © 2012, SAS Institute Inc., Cary, NC, USA

All rights reserved. Produced in the United States of America.

For a hardcopy book: No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute Inc.

For a Web download or e-book: Your use of this publication shall be governed by the terms established by the vendor at the time you acquire this publication.

The scanning, uploading, and distribution of this book via the Internet or any other means without the permission of the publisher is illegal and punishable by law. Please purchase only authorized electronic editions and do not participate in or encourage electronic piracy of copyrighted materials. Your support of others' rights is appreciated.

U.S. Government Restricted Rights Notice: Use, duplication, or disclosure of this software and related documentation by the U.S. government is subject to the Agreement with SAS Institute and the restrictions set forth in FAR 52.227–19 Commercial Computer Software-Restricted Rights (June 1987).

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

1st electronic book, August 2012

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.

SAS® and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are registered trademarks or trademarks of their respective companies.

Contents

<i>What's New in SAS ODS Graphics Procedures 9.3</i>	<i>vii</i>
<i>Recommended Reading</i>	<i>xv</i>

PART 1 Introduction 1

Chapter 1 • Introduction to SAS ODS Graphics Procedures	3
About the SAS ODS Graphics Procedures	3
Components of a Graph	5
Creating Single-Cell Graphs	6
Creating Multi-Cell Graphs	7
Creating Paneled Scatter Plots	8
Rendering Graphs from GTL Templates or ODS Graphics Editor Files	9
Producing Graphs That Were Created with ODS Graphics Designer	10
About ODS Destinations and Styles	11
About the SAS Sample Library	12
References	12
Chapter 2 • Elements of a Program	13
A Typical Program	13
The PROC Step	14
SAS Statements	16
ODS Statements	17
ODS GRAPHICS Statement Options	17
Using an Annotation Data Set	18
Using an Attribute Map Data Set	18
Chapter 3 • Overview of Plots and Charts	19
Basic Plots and Charts	19
Fit and Confidence Plots	32
Distribution Plots	37
Categorization Plots and Charts	42
Chapter 4 • SAS Statements That Are Used with ODS Graphics Procedures	51
Overview of SAS Statements That Are Used with ODS Graphics Procedures	51
Dictionary	52

PART 2 SAS ODS Graphics Procedures 65

Chapter 5 • SGDESIGN Procedure	67
Overview: SGDESIGN Procedure	67
Concepts: SGDESIGN Procedure	68
Syntax: SGDESIGN Procedure	70
Examples: SGDESIGN Procedure	73
Chapter 6 • SGPANEL Procedure	77

Overview: SG PANEL Procedure	78
Concepts: SG PANEL Procedure	79
Syntax: SG PANEL Procedure	83
Examples: SG PANEL Procedure	267
Chapter 7 • SGPLOT Procedure	275
Overview: SG PLOT Procedure	276
Concepts: SG PLOT Procedure	277
Syntax: SG PLOT Procedure	279
Examples: SG PLOT Procedure	498
Chapter 8 • SGRENDER Procedure	513
Overview: SG RENDER Procedure	513
Syntax: SG RENDER Procedure	513
Examples: SG RENDER Procedure	517
Chapter 9 • SGSCATTER Procedure	523
Overview: SG SCATTER Procedure	523
Concepts: SG SCATTER Procedure	525
Syntax: SG SCATTER Procedure	527
Examples: SG SCATTER Procedure	548
 PART 3 SG Annotation 555	
Chapter 10 • Annotating ODS Graphics	557
Overview of SG Annotation	557
SG Annotation Data Sets	558
Modifying an SG Procedure to Use the SG Annotation Data Set	561
Controlling the Drawing Space	561
Chapter 11 • SG Annotation Function Dictionary	565
Dictionary	565
Examples	597
 PART 4 SG Attribute Maps 603	
Chapter 12 • Using SG Attribute Maps to Control Visual Attributes	605
Overview of SG Attribute Maps	605
SG Attribute Map Data Sets	606
Modify the Procedure to Use the SG Attribute Map Data Set	610
Example: Create a Plot That Uses a Single SG Attribute Map	611
Example: Combine Multiple SG Attribute Maps in a Graph	612
Example: Create a Panel That Uses an Attribute Map	614
 PART 5 Customizing ODS Graphics 617	
Chapter 13 • Controlling the Appearance of Your Graphs	619
Overview	619
Understanding Styles	620

Specifying Styles	624
Using Procedure Options to Control Graph Appearance	625
Output for Grouped versus Non-Grouped Data	630
Modifying Style Templates	635
Using Fill Patterns to Distinguish Grouped Bar Charts	636
Style Elements for Use with ODS Graphics	640
Chapter 14 • Managing Your Graphics with ODS	649
Introduction	649
Specifying a Destination	649
Using the ODS GRAPHICS Statement	651
 PART 6 Appendix 657	
Appendix 1 • Units of Measurement	659
Appendix 2 • Marker Symbols	661
Appendix 3 • Line Patterns	663
Appendix 4 • ODS Graphics Software	665
Appendix 5 • Comparisons with the SAS/GRAPH Procedures	667
SAS/GRAPH Output versus ODS Graphics	667
Differences between the ODS Graphics Procedures and SAS/GRAPH Procedures	668
 Glossary	671
Index	673

What's New in SAS ODS Graphics Procedures 9.3

Overview

The *SAS ODS Graphics: Procedures Guide* contains new information explaining how styles are applied to graphs that are grouped and non-grouped.

The second edition added a new introductory section. The new introduction includes a detailed description of a typical program and an example of each supported plot type.

In addition, the procedures have the following changes and enhancements for SAS 9.3:

- inclusion with Base SAS and name change
- changes to the default ODS output
- new plot statements are available for the SGPLOT and SGPANEL procedures.
- new options and enhancements are available for the PROC SGPLOT, PROC SGPANEL, and PROC SGSCATTER statements.
- new options and enhancements are available for the existing plot statements in the SGPLOT and SGPANEL procedures.
- new options and enhancements are available for the axis statements in the SGPLOT and SGPANEL procedures.
- new options and enhancements are available for the SGRENDER procedure.
- enhancements are available for the SGDESIGN procedure.
- a new attribute map feature provides a mechanism for controlling the visual attributes that are applied to specific group data values in your graphs.
- a new annotation feature provides a mechanism for adding shapes, images, and annotations to graph output.

ODS Graphics Procedures Are Included with Base SAS

The ODS Graphics procedures, formerly called SAS/GRAPH Statistical Graphics procedures, are now available with Base SAS software. SAS/GRAPH software is not required in order to use the procedures.

Note: The ODS Graphics Designer, ODS Graphics Editor, and Graph Template Language have also moved to Base SAS.

Changes to the Default ODS Output

In Windows and UNIX operating environments, when the ODS Graphics procedures are executed in the SAS Windowing environment, the default behavior has changed as follows:

- HTML is the default ODS destination. If you close this destination and do not open another destination, then no destinations are open.
- HTMLBlue is the default style for the HTML destination. You can change this default style in the SAS Preferences.
- Graphs are no longer saved in the SAS current directory by default. They are saved in the directory that corresponds to your SAS Work library. You can specify a different directory in the SAS Preferences.

These changes do not apply when the procedures are run in batch mode. In addition, the z/OS operating environment continues to use the ODS LISTING destination as the default destination.

To create LISTING output, do one of the following:

- Specify LISTING in the Results tab in the SAS Preferences.
- Add the ODS LISTING statement to your SAS program.

New Plot Statements for the SGPLOT and SGPANEL Procedures

BUBBLE Statement

A new BUBBLE statement creates a bubble plot in which two variables determine the location of the bubble centers and a third variable controls the size of the bubble.

HBARPARM and VBARPARM Statements

New HBARPARM and VBARPARM statements create a horizontal or vertical bar chart based on a pre-summarized response value for each unique value of the category variable. You can also assign variables to the upper and lower limits.

HIGHLOW Statement

A new HIGHLOW statement creates a display of floating vertical or horizontal lines or bars that represent high and low values. The statement also gives you the option to display open and close values as tick marks and to specify a variety of plot attributes.

LINEPARM Statement

A new LINEPARM statement creates a straight line specified by a point and a slope. You can generate a single line by specifying a constant for each required argument. You can generate multiple lines by specifying a numeric variable for any or all required arguments.

WATERFALL Statement (SGPLOT Only, Preproduction)

A new WATERFALL statement creates a waterfall chart computed from input data. In the chart, bars represent an initial value of Y and a series of intermediate values identified by X leading to a final value of Y.

Updates to the PROC SGPLOT, PROC SG PANEL, and PROC SGSCATTER Statements

All three procedure statements include the following new options:

- The DATTRMAP= option specifies an SG attribute map data set.
- The SGANNO= option specifies an SG annotation data set.
- The PAD= option reserves space around the border of an annotated graph.

The UNIFORM= option in the SGPLOT procedure enables you to control axis scaling and legend marker attributes for the row and column axes independently.

Updates to Plot Statements in the SGPLOT and SG PANEL Procedures

General Updates

The following options and enhancements have been added to multiple plot statements:

- The ATTRID= option specifies the value of the ID variable in an attribute map data set. (This option is also used with the SGSCATTER procedure.)
- The CATEGORYORDER= option specifies the order in which the response values are arranged. This option affects bar charts, line plots, and dot plots.
- The CLIATTRS= and CLMATTRS= options now enable you to specify line attributes and fill attributes for confidence limits.
- The CURVELABELATTRS= and DATALABELATTRS= options specify options for setting text attributes for plot curves and labels.
- The DISCRETEOFFSET= option specifies an amount to offset graph elements from the category midpoints or from the discrete axis tick marks. This option affects bar charts, box plots.

- The following are new options for grouped data (using the GROUP= option):
 - The CLUSTERWIDTH= option specifies the cluster width as a ratio of the midpoint spacing when a group is in effect. This option affects any plot that can have a discrete axis.
 - The GROUPDISPLAY= option specifies how to display grouped graphics elements. This option affects any plot that can have a discrete axis. (The option is not available for the HBARPARM and VBARPARM statements.)
 - The GROUPORDER= option specifies the ordering of graph elements within a group. This option affects any plot that can have a discrete axis.

BAND Statement

The following options and enhancements are specific to the BAND statement:

- The CURVELABELLOWER= and CURVELABELUPPER= options specify labels for the plot's upper and lower limits.
- The TYPE= option specifies whether the data points for the band boundaries are connected as a series plot or as a step plot.

HBAR and VBAR Statements

The following options and enhancements are specific to the HBAR and VBAR statements:

- The DATALABEL= option now enables you to specify a variable that contains values for the data labels.
- The DATASKIN= option specifies a special effect to be used on all filled bars.
- Some SAS styles display fill patterns for grouped bars.

Note: These options are also available with the new HBARPARM and VBARPARM statements. The DATALABEL and DATASKIN options are available with the new WATERFALL statement.

The VBAR and VBARPARM statements in the SGPLOT procedure have a DATALABELPOS= option, which specifies the location of the data label.

HBOX and VBOX Statements

The following options and enhancements are specific to the HBOX and VBOX statements:

- The CAPSHAPE= option specifies the shape of the whisker cap lines.
- The CONNECT= option specifies that a connect line joins a statistic from box to box.
- Boxes can be grouped. In addition to the GROUP= option, the GROUPDISPLAY= and GROUPORDER= options are available.
- The NOTCHES option shows the notches.
- The NOMEAN option hides the mean symbol.
- The NOMEDIAN option hides the median line.
- The NOOUTLIERS option hides the outliers.

- You can specify appearance attributes for these elements:
 - connect lines
 - data labels
 - box fills and lines
 - mean markers, median lines, outlier markers, and whisker and cap lines

HISTOGRAM Statement

The HISTOGRAM statement provides greater control over bins with the following options:

- BINSTART= specifies the X coordinate of the first bin.
- BINWIDTH= specifies the bin width.
- NBINS= specifies the number of bins.

INSET and KEYLEGEND Statements

The INSET and KEYLEGEND statements enable you to change text attributes with the following options:

- The TITLEATTRS= and TEXTATTRS= options in the INSET statement. The INSET statement applies to the SGPLOT procedure only.
- The TITLEATTRS= and VALUEATTRS= options in the KEYLEGEND statement

VLINE Statement

The VLINE statement in the SGPLOT procedure has a DATALABELPOS= option, which specifies the location of the data label.

Axis Updates for the SGPANEL and SGPLOT Procedures

Axis Updates for the SGPLOT Procedure

The XAXIS, X2AXIS, YAXIS, and Y2AXIS statements support several enhancements and new options:

- New LABELATTRS and VALUEATTRS options specify textual attributes for axis labels and axis tick value labels, respectively.
- A new REVERSE option specifies that the tick values are displayed in reverse (descending) order.
- New THRESHOLDMAX and THRESHOLDMIN options specify a threshold for displaying one more tick mark at the high end and the low end of the axis, respectively.

Axis Updates for the SGPANEL Procedure

The COLAXIS and ROWAXIS statements support several enhancements and new options:

- The same updates are supported as listed in “Axis Updates for the SGPLOT Procedure”.
- The REFTICKS option enables you to specify whether labels and values are added to the tick marks. (This option adds tick marks to the side of the panel that is opposite from the specified axis.)

Updates to the SGRENDER Procedure

You can use the SGRENDER procedure to render a graph from a SAS ODS Graphics Editor (SGE) file.

Updates to the SGDESIGN Procedure

The SGDESIGN procedure is supported on z/OS systems. However, the following limitations apply:

- The procedure does not render SGD files that were generated with the previous release of the ODS Graphics Designer. You must open the SGD file in the 9.3 version of the ODS Graphics Designer (on a Windows or UNIX system). Then save the file in the 9.3 format.
- SGD files must be transferred to the HFS file system of UNIX System Services in order to be rendered.

New Attribute Mapping Feature

A new attribute map feature provides a mechanism for controlling the visual attributes that are applied to specific group data values in your graphs. This feature uses SG attribute map data sets to associate data values with visual attributes. The data set uses reserved variable names for the attribute map identifier, the group value, and the attributes.

You can use attribute maps in the SGPLOT, SGPANEL, and SGSCATTER procedures. The procedure statement references the name of the SG attribute map data set, and plot statements specify the group and the attribute map identifier.

New Annotation Feature

A new annotation feature provides a mechanism for adding shapes, images, and annotations to graph output. For example, you can add text labels, lines, circles, rectangles, polygons, and images. This feature uses SG attribute data sets, which contain the commands for creating the annotation elements. The data set uses reserved variable names for the draw function and the attributes that control how the function is performed.

You can use annotation in the SGPLOT, SGPANEL, and SGSCATTER procedures. The procedure statement references the name of the SG annotation data set.

Recommended Reading

- *SAS ODS Graphics: Getting Started with Business and Statistical Graphics*
- *Statistical Graphics Procedures by Example: Effective Graphs Using SAS*
- *Statistical Graphics in SAS: An Introduction to the Graph Template Language and the Statistical Graphics Procedures*
- *SAS Graph Template Language: User's Guide*
- *SAS Graph Template Language: Reference*
- *SAS ODS Graphics Designer: User's Guide*
- *SAS Output Delivery System: User's Guide*
- *Output Delivery System: The Basics and Beyond*
- *Quick Results with the Output Delivery System*

For a complete list of SAS publications, go to support.sas.com/bookstore. If you have questions about which titles you need, please contact a SAS Publishing Sales Representative:

SAS Publishing Sales
SAS Campus Drive
Cary, NC 27513-2414
Phone: 1-800-727-3228
Fax: 1-919-677-8166
E-mail: sasbook@sas.com
Web address: support.sas.com/bookstore

Part 1

Introduction

<i>Chapter 1</i>	
Introduction to SAS ODS Graphics Procedures	<i>3</i>
<i>Chapter 2</i>	
Elements of a Program	<i>13</i>
<i>Chapter 3</i>	
Overview of Plots and Charts	<i>19</i>
<i>Chapter 4</i>	
SAS Statements That Are Used with ODS Graphics Procedures ...	<i>51</i>

Chapter 1

Introduction to SAS ODS Graphics Procedures

About the SAS ODS Graphics Procedures	3
Components of a Graph	5
Creating Single-Cell Graphs	6
Creating Multi-Cell Graphs	7
Creating Paneled Scatter Plots	8
Rendering Graphs from GTL Templates or ODS Graphics Editor Files	9
Producing Graphs That Were Created with ODS Graphics Designer	10
About ODS Destinations and Styles	11
About ODS Destinations	11
About ODS Styles	12
About the SAS Sample Library	12
References	12

About the SAS ODS Graphics Procedures

The ODS Graphics procedures, sometimes called ODS Statistical Graphics procedures, use ODS Graphics functionality to produce plots for exploratory data analysis and for customized statistical displays. The procedures provide a simple, high-level syntax that enables you to produce sophisticated graphs by using a wide array of plot types and layouts. You can create scatter plots, histograms, bar charts, box plots, classification panels, scatter plot matrices, and many other types of statistical and business graphs. Your graphs can have titles, footnotes, legends, and other graphics elements.

The procedures support statistical analysis and can create simple or complex graphical views of your data. Though the procedures were initially designed to facilitate the production of standard statistical graphs, they are also well suited for the production of non-statistical or business graphs.

The ODS Graphics procedures create graphs that are based on the Graph Template Language (GTL). However, you do not need to know the details of templates and the GTL in order to use the ODS Graphics procedures. With very little coding effort, you can use the procedures to create the most commonly used graphs that are supported by the GTL.

There are five ODS Graphics procedures, each with a specific purpose:

SGPLOT

creates single-cell plots with a variety of plot and chart types and overlays.

SGPANEL

creates classification panels for one or more classification variables. Each graph cell in the panel can contain either a simple plot or multiple, overlaid plots.

Note: The SGPLOT and SGPANEL procedures largely support the same types of plots and charts. For this reason, the two procedures have an almost identical syntax. The main distinction between the two procedures is that the SGPANEL procedure produces a matrix of graphs, one for each level of a classification variable.

SGSCATTER

creates scatter plots and scatter plot matrices with optional fits and ellipses.

SGRENDER

produces graphs from graph templates that are written in the GTL. You can also render a graph from a SAS ODS Graphics Editor (SGE) file.

SGDESIGN

creates graphical output based on a graph file that has been created by using the ODS Graphics Designer application.

An ODS destination must be open to create output from these procedures. By default, the ODS HTML destination is open. You can use the ODS destination options and the ODS GRAPHICS statement options to control many aspects of your graph output. For more information, see [Chapter 14, “Managing Your Graphics with ODS,”](#) on page 649.

The procedures have two facilities that enable you to modify graph output:

- The SG annotation feature enables you to add shapes, images, and other annotations to graph output.
- SG attribute maps enable you to control the visual attributes that are applied to specific data values in your graphs. For example, if you create a graph that plots items sold in different countries, you can specify the display attributes for the sales data of each country by name. Attribute maps enable you to ensure that particular visual attributes are applied based on the value of the data rather than the position of the data in the data set.

The ODS Graphics procedures enable you to create complex statistical graphics that use the principles of effective graphics¹ to accurately communicate the results of your analysis to your consumers. The minimal coding required enables you to focus on your statistical analysis instead of the visual appearance of your graphs.

See Also

- [“Overview of ODS Graphics Software”](#) on page 665
- *SAS Output Delivery System: User's Guide*
- *SAS Graph Template Language: User's Guide*

¹ For more information about the principles of effective graphics, see Cleveland (1993) and Robbins (2005).

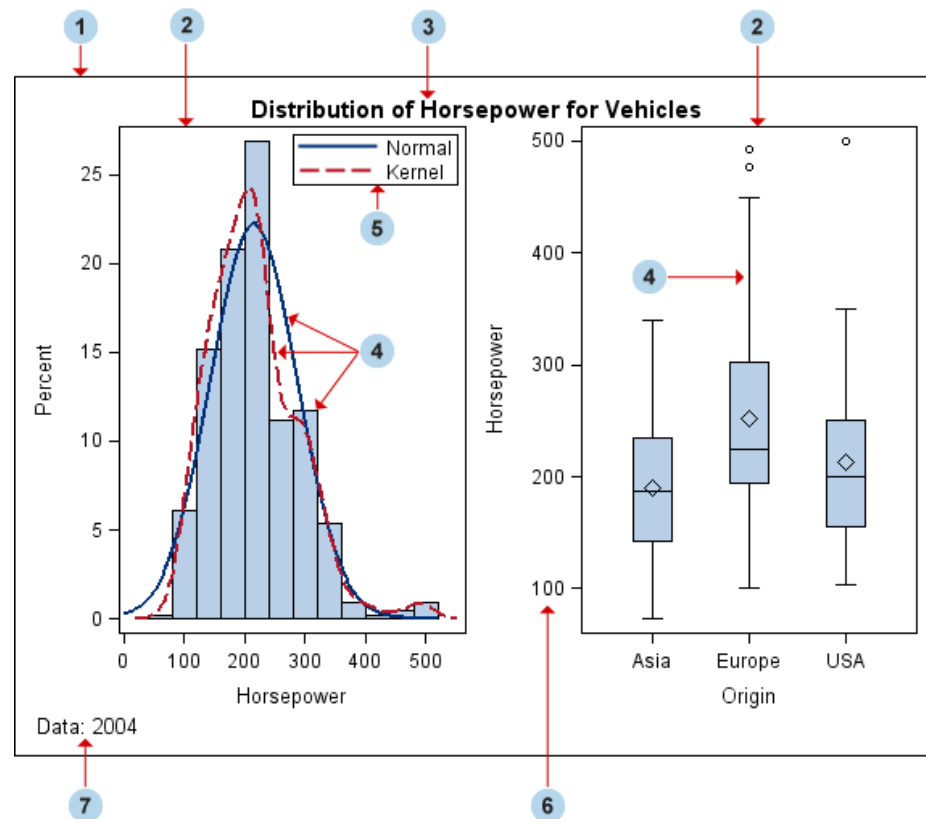
Components of a Graph

In general, a graph is made up of the following parts:

- titles and footnotes
- one or more cells that contain a composite of one or more plots
- legends, which can reside inside or outside a cell

The following figure shows the different parts of a graph:

Figure 1.1 Components of a Graph



1 Graph

a visual representation of data. The graph can contain titles, footnotes, legends, and one or more cells that have one or more plots.

2 Cell

a distinct rectangular subregion of a graph that can contain plots, text, and legends.

3 Title

descriptive text that is displayed above any cell or plot areas in the graph.

4 Plot

a visual representation of data such as a scatter plot, a series line, a bar chart, or a histogram. Multiple plots can be overlaid in a cell to create a graph.

5 Legend

refers collectively to the legend border, one or more legend entries (where each entry has a symbol and a corresponding label) and an optional legend title.

6 Axis

refers collectively to the axis line, the major and minor tick marks, the major tick mark values, and the axis label. Each cell has a set of axes that are shared by all the plots in the cell. In multi-cell graphs, the columns and rows of cells can share common axes if the cells have the same data type.

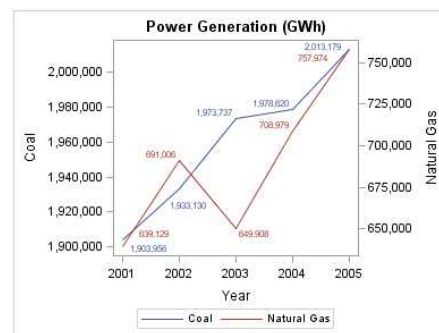
7 Footnote

descriptive text that is displayed below any cell or plot areas in the graph.

Creating Single-Cell Graphs

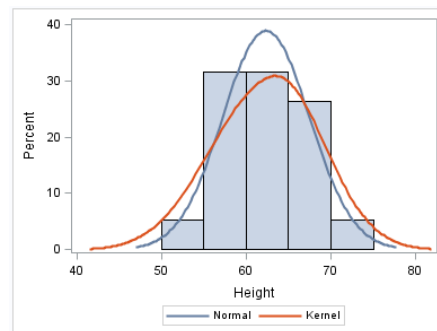
The SGPLOT procedure creates single-cell graphs with a wide range of plot types including density, dot, needle, series, bar, histograms, box, and others. The procedure can compute and display loess fits, polynomial fits, penalized B-spline fits, and ellipses. You can also add text, legends, and reference lines. Options are available for specifying colors, marker symbols, and other attributes of plot features. You can customize the axes by using axis statements such as XAXIS and YAXIS.

Plot statements can be combined to create more informative plots and charts. The following example shows two series plots that are overlaid in a single graph. Each plot is assigned to a different vertical axis. Data labels have been added for easy reference.



```
title "Power Generation (GWh)";
proc sgplot data=sashelp.electric(where=
  (year >= 2001 and customer="Residential"));
  xaxis type=discrete;
  series x=year y=coal / datalabel;
  series x=year y=naturalgas /
    datalabel y2axis;
run;
title;
```

The following example creates a graph with a histogram, a normal density curve, and a kernel density curve.



```
proc sgplot data=sashelp.class;
  histogram height;
  density height;
  density height / type=kernel;
run;
```

For more information about the SGPLOT procedure and the procedure syntax, see [Chapter 7, “SGPLOT Procedure,” on page 276](#).

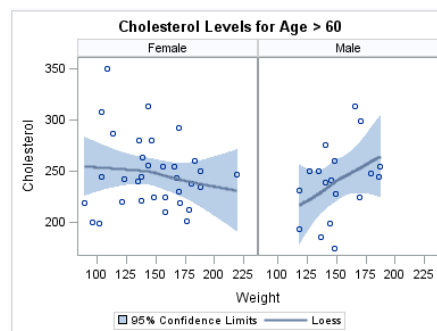
Creating Multi-Cell Graphs

The SGPPANEL procedure creates a panel for the values of one or more classification variables. Each graph cell in the panel can contain either a single plot or multiple overlaid plots.

The SGPPANEL procedure supports most of the plots and overlays that the SGPLOT procedure supports. For this reason, the two procedures have an almost identical syntax. As with the SGPLOT procedure, options are available for specifying colors, marker symbols, and other attributes.

The procedure syntax supports four types of panel layouts: PANEL, LATTICE, COLUMNLATTICE, and ROWLATTICE.

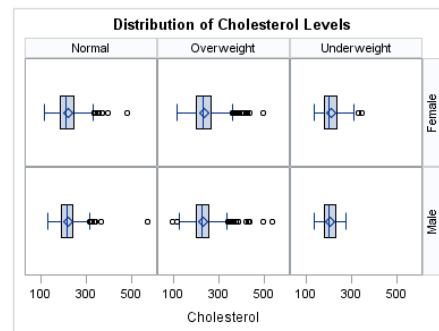
The following example creates a panel of loess curves using the default PANEL layout. In the PANEL layout, each graph cell represents a specific crossing of values for one or more classification variables. A label above each cell identifies the crossing of values that is represented in the cell. By default, cells are created only for crossings that are represented in the data set.



```
title1 "Cholesterol Levels for Age > 60";
proc sgpanel data=sashelp.heart(
  where=(AgeAtStart > 60)) ;
  panelby sex / novarname;
  loess x=weight y=cholesterol / clm;
run;
title1;
```

The following example creates a panel of box plots in a LATTICE layout. The graph cells are arranged in rows and columns by using the values of two classification variables. Labels above each column and to the right of each row identify the

classification value that is represented by that row or column. A cell is created for each crossing of classification values.



```
title1 "Distribution of Cholesterol Levels";
proc sgpanel data=sashelp.heart;
  panelby weight_status sex / layout=lattice
                                novarname;
  hbox cholesterol;
run;
title1;
```

For more information about the SG PANEL procedure and the procedure syntax, see [Chapter 6, “SG PANEL Procedure,” on page 78](#).

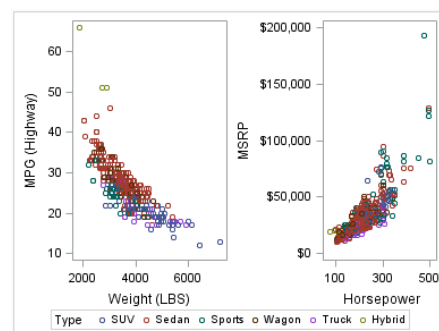
Creating Paneled Scatter Plots

The SGSCATTER procedure creates a paneled graph for multiple combinations of variables.

The procedure syntax supports the following features:

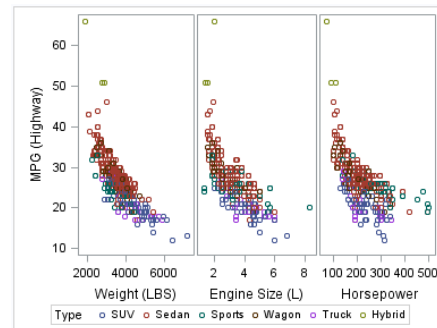
- three types of graph layouts: PLOT, COMPARE, and MATRIX
- basic scatter plots
- fit and confidence plots: loess curves, regression curves, penalized B-spline curves, and ellipses
- distribution plots: histograms and density curves (in the diagonal cells of a matrix)
- legends

The following example creates a panel using the PLOT layout. The PLOT statement creates a paneled graph of scatter plots where each cell has its own independent set of axes.



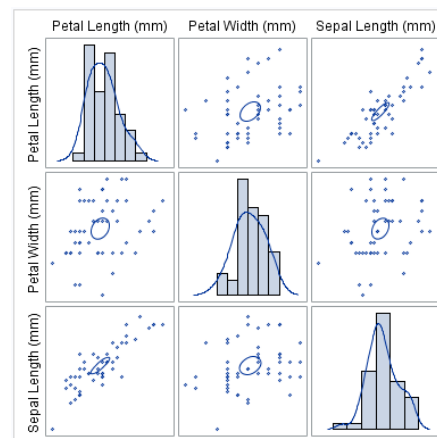
```
proc sgscatter data=sashelp.cars;
  plot mpg_highway*weight msrp*horsepower
        / group=type;
run;
```


The following example creates a panel using the COMPARE layout. The COMPARE statement creates a paneled graph that uses common axes for each row and column of cells. Cells are created for all crossing of the X and Y variables.



```
proc sgscatter data=sashelp.cars;
  compare y=mpg_highway
          x=(weight enginesize horsepower )
          / group=type;
run;
```

The following example creates a panel using the MATRIX layout. The MATRIX statement creates a matrix of scatter plots where each cell represents a different combination of variables. In the diagonal cells, you can place labels or histograms with or without density curves.



```
proc sgscatter data=sashelp.iris
              (where=(species eq "Virginica"));
  matrix petallength petalwidth sepallength
        / ellipse=(type=mean)
          diagonal=(histogram kernel);
run;
```

For more information about the SGSCATTER procedure and the procedure syntax, see [Chapter 9, “SGSCATTER Procedure,” on page 523](#).

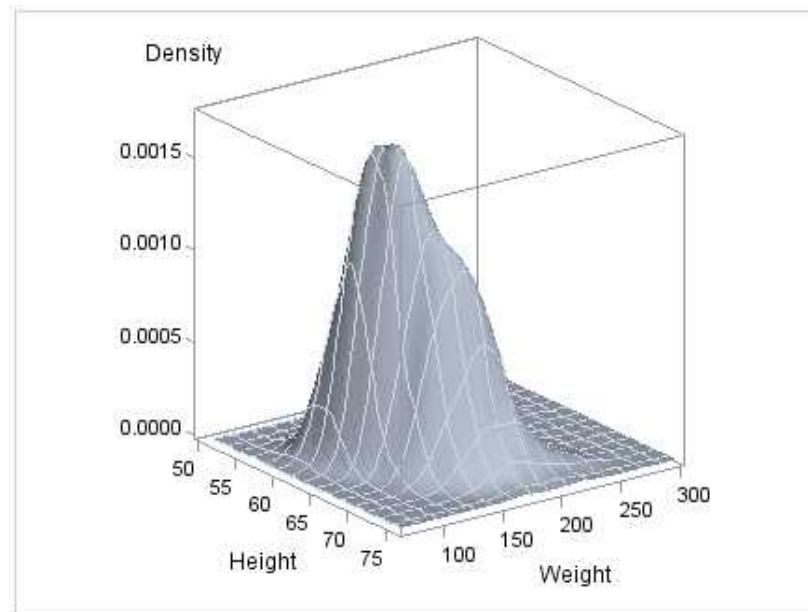
Rendering Graphs from GTL Templates or ODS Graphics Editor Files

The SGRENDER procedure creates graphical output from templates that are created using the Graph Template Language (GTL). You can use the GTL to create many different types of plots, paneled graphs, and matrices, some of which cannot be created with the ODS Graphics procedures.

The SGRENDER procedure can also produce graphical output from graphs that were edited in the SAS ODS Graphics Editor. An ODS Graphics Editor file (SGE) is created in SAS by using the SGE = ON option in the ODS destination statement. The SGRENDER procedure enables you to run one or more graphs in batch mode and render

the graphs to any ODS destination using any of the supported ODS options. For more information about the editor, see the *SAS ODS Graphics Editor: User's Guide*.

The following example shows a layout that you can create by using the GTL and the SGRENDER procedure.



```
proc template;
  define statgraph surface;
    begingraph;
      layout overlay3d;
        surfaceplotparm x=height y=weight z=density;
      endlayout;
    endgraph;
  end;
run;

proc sgrender data=sashelp.gridded template=surface;
run;
```

For more information about the SGRENDER procedure, see [Chapter 8, “SGRENDER Procedure,” on page 513](#). For more information about the GTL, see *SAS Graph Template Language: User's Guide*.

Producing Graphs That Were Created with ODS Graphics Designer

The SGDESIGN procedure creates graphical output based on a graph file (SGD) that has been created by using the SAS ODS Graphics Designer application.

Here are the main features of the SGDESIGN procedure:

- By default, the procedure uses the data set or data sets that are currently referenced by the SGD file.

- The procedure can generate any graph type that can be created in the ODS Graphics Designer.
- You can render the graph to any ODS destination by using standard ODS syntax. When it renders the graph, the procedure applies the style of the active destination rather than the style that was used in the SGD file.
- As with all the ODS Graphics procedures, you can use the ODS GRAPHICS statement options to control many aspects of your graphics.
- If the SGD file has been defined with dynamic variables, these variables can be initialized with the DYNAMIC statement of the procedure. You can use dynamic variables to generate the same graph with different data variables, a different data set, and different text elements.
- The procedure supports SAS statements such as FORMAT, LABEL, BY, and WHERE. These statements can be applied only if the DATA= option is used with the procedure.

For more information about the SGDESIGN procedure and the procedure syntax, see [Chapter 5, “SGDESIGN Procedure,” on page 67](#).

About ODS Destinations and Styles

ODS manages all output created by the procedures and enables you to control the output destination and format. ODS also enables you to control the style and other output features.

About ODS Destinations

ODS destinations determine where your graph output is sent and how the output is formatted. For example, the HTML destination creates an HTML file that points to the graph image file. The LISTING destination sends output to an image file. The output image can be displayed by opening the image file from the Results window.

For creation of ODS graphs, a valid ODS destination must be open. You can open destinations by specifying an ODS destination statement. In the SAS windowing environment on Windows and UNIX systems, the HTML destination is open by default. (The default destination for batch mode is LISTING.) If you keep the default HTML destination open and open another, the resultant output is sent to the Web as well as to the other specified destination. With the exceptions of the HTML and LISTING destinations, you must also close the destination before output is generated.

The ODS destination statement is used at the beginning and end of the program to open and close destinations.

For example, the following statements open and close an ODS LISTING destination.

```
ods listing;          /* opens the destination */
/* procedure statements and other program elements here */
ods listing close;    /* closes the destination */
```

Depending on the options available for the destination, you can specify options such as the filename or the path to an output directory. For more information, see [“Specifying a Destination” on page 649](#).

About ODS Styles

ODS styles determine the overall appearance of your output. By default, ODS applies a style to all output. A style is a template, or set of instructions, that determines the colors, fonts, line styles, fill colors, and other presentation aspects of your output. Each destination has a default style associated with it. For example, the default style for the PDF destination is Printer, and the default style for the HTML destination is HTMLBlue.

The ODS Graphics procedures automatically obtain their default appearance attributes from the current ODS style. However, you can use appearance options in your plot statements to override the default style attributes.

To change the style that is applied to your output, specify the `STYLE=` option on your ODS destination statement.

For example, suppose you want to change the overall look of your graph for the HTML destination to the Analysis style. Do this by specifying `STYLE=ANALYSIS` in the ODS HTML destination statement as follows:

```
ods html style=analysis;
```

For more information, see [Chapter 13, “Controlling the Appearance of Your Graphs,”](#) on page 619.

SAS ships predefined styles in the STYLES item store in SASHELP.TMPLMST. Some of these predefined styles are described in [“Recommended Styles”](#) on page 621. To see all available styles, see [“Viewing a Style Template”](#) on page 622.

About the SAS Sample Library

Many of the examples in this guide also reside in the SAS Sample Library. These examples include the name of the sample library member in their syntax description.

How you access the code in the sample library depends on how it is installed at your site.

- In most operating environments, you can access the sample code through the SAS Help facility. Select **Help** ⇒ **SAS Help and Documentation**. On the **Contents** tab, select **Learning to Use SAS** ⇒ **Sample SAS Programs** ⇒ **Base SAS** ⇒ **Samples**.
- In other operating environments, the SAS Sample Library might have been installed in your file system. If the SAS Sample Library has been installed at your site, ask your on-site SAS support personnel where the library is located.

References

- Cleveland, W. S. 1993. *Visualizing Data*. Summitt, NJ: Hobart Press.
- Robbins, N. B. 2005. *Creating More Effective Graphs*. Hoboken, NJ: Wiley InterScience.

Chapter 2

Elements of a Program

A Typical Program	13
The PROC Step	14
About the PROC Step	15
Procedure Statements and Options	15
Plot Statements and Options	15
(Optional) Legend Statement and Options	15
(Optional) Axis Statements and Options	16
Other Required Statements	16
SAS Statements	16
ODS Statements	17
ODS GRAPHICS Statement Options	17
Using an Annotation Data Set	18
Using an Attribute Map Data Set	18

A Typical Program

Your programs must include at least one procedure (PROC step), which in turn contains a number of statements related to the procedure. The programs can also include ODS statements, ODS GRAPHICS statements, and Base SAS statements. In addition, the programs can specify an annotation data set or an attribute map data set.

The sample program below identifies the basic elements of a typical program.

```

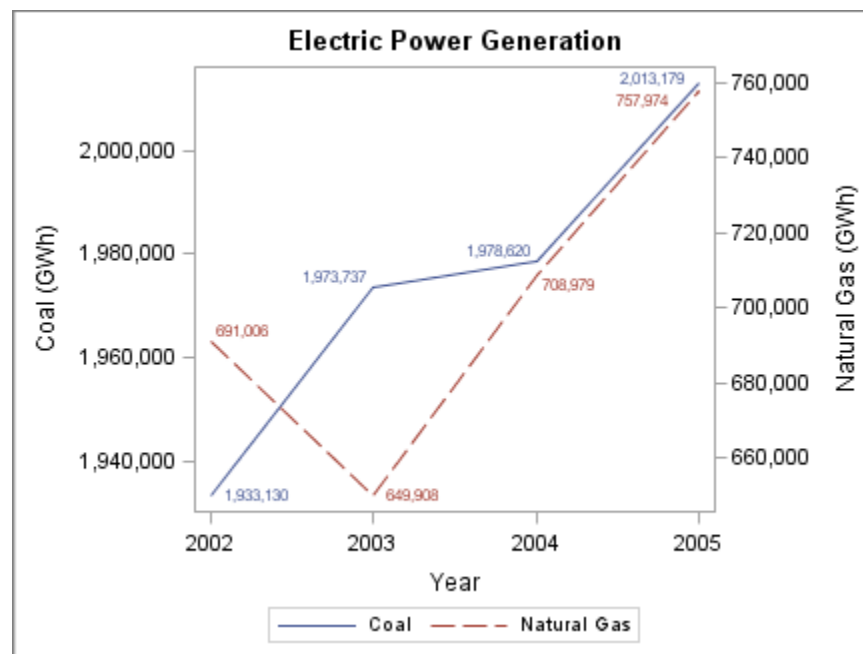
1 ods html style=statistical;
2 ods graphics on / width=4.5in;
3 title "Electric Power Generation";
4 proc sgplot data=sashelp.electric;
5   where (year >= 2002) and
      (customer="Residential");
6   series x=year y=coal / datalabel;
7   series x=year y=naturalgas /
      datalabel y2axis;
8   xaxis type=discrete;
9   yaxis label="Coal (GWh)";
10  y2axis label="Natural Gas (GWh)";
11 run;
12 title;
13 ods graphics on / reset=all;
14 ods html style=htmlblue;

```

Global SAS statements

PROC step

Here is the output for the sample program.



The following sections describe each element of the program in more detail and explain how the elements relate to one another.

The PROC Step

The descriptions in this topic refer to the example that is shown in “A Typical Program” on page 13.

About the PROC Step

A group of SAS procedure statements is called a PROC step.

The PROC step consists of the following:

- a beginning procedure (PROC) 4 statement with options
- typically, statements specifying plot types, variables, and options
- an ending RUN 9 statement

These statements can identify and analyze the data in SAS data sets. They can generate the graphics output and control the appearance of the output. Statements can define variables and perform other operations on your data. You can also specify global statements and options within the PROC step.

Procedure Statements and Options

The procedure statement 4 identifies which procedure you are invoking, such as the SGPLOT procedure, the SGPANEL procedure, and so on.

The statement also specifies which input data set is to be used. A data set is not required for all of the procedures. For example, the SGRENDER procedure defaults to the most recently created SAS data set if none is provided. The SGDESIGN procedure defaults to the data set or data sets that are currently defined in the SGD file.

The statement can include options that are related to the procedure. For example, the DESCRIPTION= option can be used with several procedures to provide a description for the output image.

Plot Statements and Options

Plot statements are used within the procedure to identify the type of plot that you want the procedure to produce. The SGPLOT and SGPANEL procedures require at least one plot statement.

Multiple plot statements can be used, as shown in the example. A SERIES statement 6 is used to create a series plot that shows power generation from coal. A second SERIES statement creates a series plot that shows power generation from natural gas.

Options are available for specifying colors, marker symbols, and other attributes of plot features. In the example, both series plots specify that data labels are displayed. The second SERIES statement uses the Y2AXIS option to plot natural gas power output along the Y axis on the right side of the plot 7.

The SGPLOT procedure also enables you to add a text inset to a plot using the INSET statement (not shown in the example). The INSET statement adds a text box within the axes of the plot. Options are available for specifying the visual attributes of the text box and the text.

(Optional) Legend Statement and Options

By default, legends are created automatically for some plots, depending on their content. The graph shown in the example has an automatically generated legend.

You can manually add a legend to the graph for the SGPLOT, SGPANEL, and SGSCATTER procedures. When you manually add a legend, options are available for specifying the legend title, its position in the graph, and other attributes.

(Optional) Axis Statements and Options

The SGPLOT and SGPANEL procedures contain statements that enable you to change the type and appearance for the axes of the graph. By default, the type of each axis is determined by the types of plots that use the axis and the data that is applied to the axis. You can change the type of axis that is used for a plot. For example, to display independent data values rather than a range of numeric values on the axis, specify the TYPE=DISCRETE option. (Not all plot types support discrete axes.)

The SGPLOT procedure supports the use of secondary axes, as shown in the example [7](#). Secondary axes are denoted as X2 and Y2 axes. The secondary axes support the same options as the primary axes.

When you use an axis statement, options are available for showing or hiding axis features, such as ticks and labels, and for specifying other attributes. The graph shown in [“A Typical Program” on page 13](#) has three axis statements [8](#). The first statement changes the X axis to be discrete. The other two statements change the labels for the Y and Y2 axes.

Other Required Statements

The SGPANEL and SGSCATTER procedures include some important statements that are not shown in the example. These statements are required with the procedure statement.

The SGPANEL procedure requires a PANELBY statement. This statement specifies one or more classification variables for the panel, the layout type, and other options for the panel. For more information, see [“PANELBY Statement” on page 87](#).

The SGSCATTER procedure requires one of these three statements:

PLOT	creates a paneled graph of scatter plots where each graph cell has its own independent set of axes.
COMPARE	creates a shared axis panel, also called an MxN matrix.
MATRIX	creates a scatter plot matrix.

For more information, see [Concepts: SGSCATTER Procedure on page 525](#).

SAS Statements

The ODS Graphics procedures support a number of SAS statements. Some of these, such as the TITLE statement, are global statements.

A global statement is a statement that you can specify anywhere in a SAS program. A global statement sets values and attributes for all the output created after that global statement is specified in the program. The specifications in a global statement are not confined to the output generated by any one procedure. However, they do apply to all the output generated thereafter in the program, unless they are overridden by a procedure option or another global statement.

As shown in “A Typical Program” on page 13, the TITLE statement is used toward the beginning and end of the program. The first statement 3 specifies the title. The second statement 10 cancels the current title.

The example program also uses a WHERE statement 5 to subset the data that is used in the graph. In the example, the WHERE statement selects observations based on their date (2002 or greater) and the type of customer (residential).

For more information, see Chapter 4, “SAS Statements That Are Used with ODS Graphics Procedures,” on page 51.

ODS Statements

The ODS Graphics procedures use ODS destination statements to control where the output goes and how it looks. Although ODS statements are not required in every program, they are necessary if you want to generate graphs for destinations other than the default HTML destination. Some other destinations include LISTING, RTF, and PDF.

You can use the STYLE= option in the ODS destination statement to change the style that is applied to your output. As shown in “A Typical Program” on page 13, the ODS destination statement is used at the beginning and end of the program to modify the default style. The beginning statement 1 specifies a different style. The end statement 12 sets the HTML style back to its default of HTMLBlue. The ODS destination statement can also be used to open a different destination.

Depending on the options available for the destination, you can specify other features such as the name of the output file or the directory where images are stored.

An ODS destination must be open to create output from the procedures. If you want to use a destination other than the default, you should always open the destination before calling the procedure. Opening a non-default destination results in output being sent both to HTML by default as well as to the additional specified destination. Conserve system resources by using the ODS destination statement at the end of the SAS program to close a destination that was opened in that program.

See Also

- “Understanding ODS Destinations” in Chapter 3 of *SAS Output Delivery System: User's Guide*
- “Working with Styles ” in Chapter 13 of *SAS Output Delivery System: User's Guide*

ODS GRAPHICS Statement Options

You can use the ODS GRAPHICS statement options to control many aspects of your graphics. The ODS GRAPHICS statement is a global statement that can be used anywhere in your program. The settings that you specify remain in effect for all graphics until you change or reset these settings with another ODS GRAPHICS statement.

As shown in “A Typical Program” on page 13, the ODS GRAPHICS statement is used at the beginning and end of the program to modify the size of the graph. The beginning

statement [2](#) specifies the size. The end statement [11](#) set all options back to their defaults.

Using an Annotation Data Set

The SG annotation feature enables you to add shapes, arrows, text, images, and other annotations to graph output.

Two main steps are required to add annotation elements to a graph:

1. Create an SG annotation data set, which contains the commands for creating the annotation elements.
2. Modify the procedure to use the SG annotation data set. You can use annotation in the SGPLOT, SGPANEL, and SGSCATTER procedures.

For more information, see [Chapter 10, “Annotating ODS Graphics,” on page 557](#).

Using an Attribute Map Data Set

The attribute map feature enables you to control the visual attributes that are applied to specific data values in your graphs. For example, if you create a graph that plots items sold in different countries, you can specify the display attributes for the sales data of each country by name.

Attribute maps apply only to group data. Attribute maps enable you to ensure that particular visual attributes are applied based on the value of the data instead of the position of the data in the data set.

Two main steps are required for attribute mapping:

1. Create an SG attribute map data set, which associates data values with particular visual attributes. Each observation defines the attributes for a group value.
2. Modify the procedure and its plot statements to use the data in the SG attribute map. You can use attribute maps in the SGPLOT, SGPANEL, and SGSCATTER procedures (not all plot statements support attribute maps).

For more information, see [Chapter 12, “Using SG Attribute Maps to Control Visual Attributes,” on page 605](#).

Chapter 3

Overview of Plots and Charts

Basic Plots and Charts	19
About Basic Plots and Charts	19
About Band Plots	20
About Bubble Plots	21
About High-Low Charts	22
About Lines	23
About Needle Plots	26
About Scatter Plots	27
About Series Plots	29
About Step Plots	30
About Text Insets	31
About Vector Plots	31
Fit and Confidence Plots	32
About Fit and Confidence Plots	32
About Ellipse Plots	33
About Loess Plots	34
About Penalized B-Spline Plots	35
About Regression Plots	36
Distribution Plots	37
About Distribution Plots	37
About Box Plots	37
About Density Plots	39
About Histograms	40
Categorization Plots and Charts	42
About Categorization Plots and Charts	42
About Bar Charts	42
About Dot Plots	46
About Line Charts	47
About Waterfall Charts (Preproduction)	49

Basic Plots and Charts

About Basic Plots and Charts

You can use the SGPLOT and SGPANEL procedures to produce basic plots and charts.

The plot and chart statements include options for controlling how the output is displayed. Many of the options are unique to the particular plot or chart. However, some general options apply to most of the basic plots and charts.

For example, options enable you to do the following:

- specify colors, line attributes, and other visual features.
- group the data by the values of a variable. A separate plot is created for each unique value of the grouping variable. The plot elements for each group value are automatically distinguished by different visual attributes.
- use a secondary axis (X2 or Y2). This option is available only for the SGPLOT procedure.
- reference an ID variable in attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

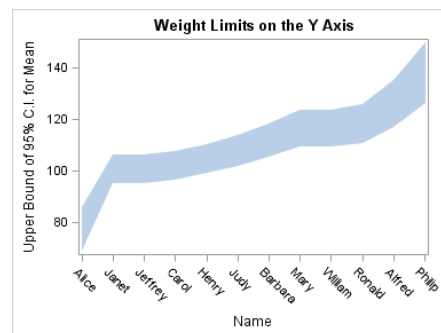
The basic plots and charts are described in the following sections. If you run the examples, your output might differ somewhat depending on the size of your graphics. The examples here were specified to be a particular size using the following statement:

```
ods graphics on / width=4in;
```

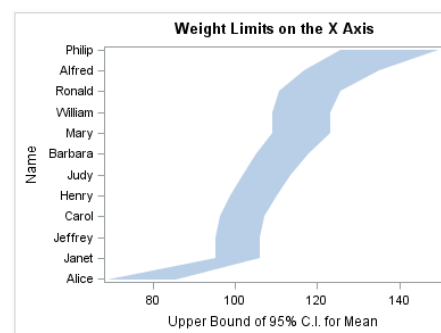
About Band Plots

A band plot creates a band that highlights part of the plot and shows upper and lower limits. The input data should be sorted by the X or Y variable.

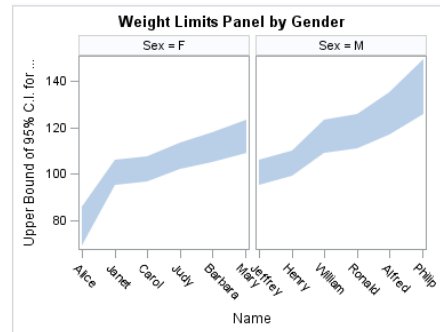
The following examples show upper and lower mean weight values for a class of students. The first two examples use the SGPLOT procedure to show the same band plotted along the X axis and the Y axis, respectively. The third example uses the SGPANEL procedure to show a matrix that is paneled by gender.



```
title "Weight Limits on the Y Axis";
proc sgplot data=sashelp.classfit;
  where age > 12;
  band x=name lower=lowermean upper=uppermean;
run;
title;
```



```
title "Weight Limits on the X Axis";
proc sgplot data=sashelp.classfit;
  where age > 12;
  band y=name lower=lowermean upper=uppermean;
run;
title;
```



```

title "Weight Limits Panel by Gender";
proc sgpanel data=sashelp.classfit;
  panelby sex / uniscale=row
              spacing=8;
  where age > 12;
  band x=name lower=lowermean upper=uppermean;
run;
title;

```

Options are available that enable you to customize the band plot and enhance its appearance. For example, you can do the following:

- add labels to the upper and lower edges of the band, specify how the labels are positioned, and set other attributes for the labels
- specify fill and outline attributes
- specify legend labels and plot transparency

Note: This list does not include all available options.

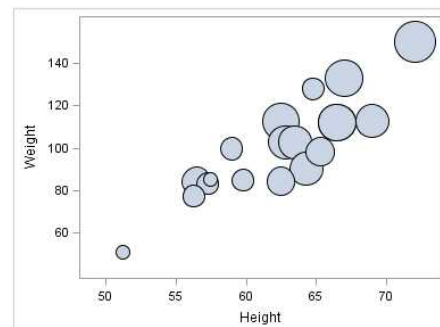
See Also

- [“BAND Statement” on page 90](#) (SGPANEL procedure)
- [“BAND Statement” on page 284](#) (SGPLOT procedure)

About Bubble Plots

Bubble plots show the relative magnitude of the values of a variable. The values of two variables determine the position of the bubble on the plot, and the value of a third variable determines the size of the bubble.

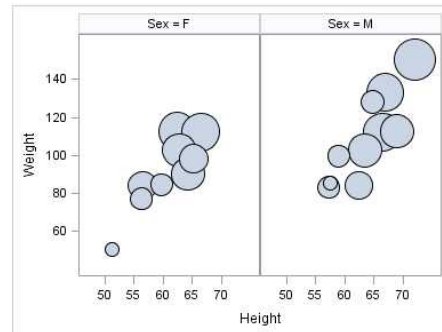
The following examples show the height and weight values for a class. The size of each bubble is determined by the student’s age. Examples are provided for the SGPLOT and the SGPANEL procedures.



```

proc sgplot data=sashelp.class;
  bubble x=height y=weight size=age;
run;

```



```
proc sgpanel data=sashelp.class;
  panelby sex;
  bubble x=height y=weight size=age;
run;
```

Options are available that enable you to customize the bubble plot and enhance its appearance. For example, you can do the following:

- control the size of the largest and the smallest bubble
- specify fill and outline attributes, and data labels and their attributes
- specify legend labels, plot transparency, and URLs for Web pages to be displayed when parts of the plot are selected within an HTML page

Note: This list does not include all available options.

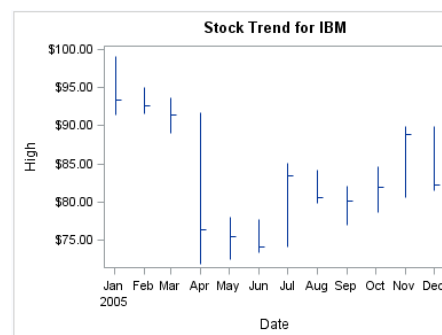
See Also

- [“BUBBLE Statement” on page 96](#) (SGPANEL procedure)
- [“BUBBLE Statement” on page 290](#) (SGPLOT procedure)

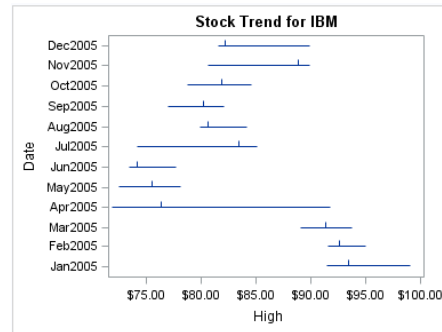
About High-Low Charts

High-low charts show how several values of one variable relate to one value of another variable. Typically, each variable value on the horizontal axis has several corresponding values on the vertical axis.

The following examples show the stock trend for IBM during a particular year. The first two examples use the SGPLOT procedure to show the same plot along the X axis and the Y axis, respectively. The third example uses the SGPANEL procedure to show a paneled graph for Intel and Microsoft stock prices in the same year. Optional values have been specified for the closing stock prices, which are represented as tick marks on the high-low lines.



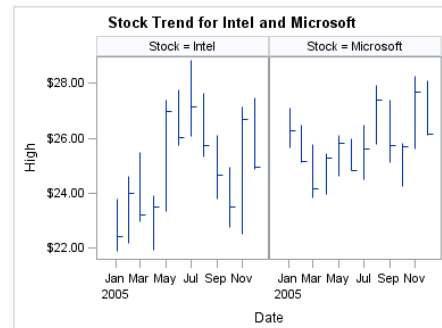
```
title "Stock Trend for IBM";
proc sgplot data=sashelp.stocks
  (where=(date >= "01jan2005"d and stock = "IBM"));
  highlow x=date high=high low=low
    / close=close;
run;
title;
```



```

title "Stock Trend for IBM";
proc sgplot data=sashelp.stocks
  (where=(date >= "01jan2005"d and stock = "IBM"));
  highlow y=date high=high low=low
    / close=close;
run;
title;

```



```

title "Stock Trend for Intel and Microsoft";
proc sgpanel data=sashelp.stocks
  (where=(date >= "01jan2005"d and
    (stock = "Intel" or stock = "Microsoft")));
  panelby stock;
  highlow x=date high=high low=low
    / close=close;
run;
title;

```

Options are available that enable you to customize the high-low chart and enhance its appearance. For example, you can do the following:

- use bars instead of lines to represent the data. If you use bars, then you can specify the fill and outline attributes for the bars.
- show tick marks for the open and closing values.
- specify labels and arrowheads for the high and low values.
- control the display of grouped data. For example, you can specify whether the groups are overlaid or clustered, the width of each cluster, and the order of lines or bars within a group.
- specify legend labels, plot transparency, and URLs for Web pages to be displayed when parts of the plot are selected within an HTML page.

Note: This list does not include all available options.

See Also

- [“HIGHLOW Statement” on page 137](#) (SGPANEL procedure)
- [“HIGHLOW Statement” on page 335](#) (SGPLOT procedure)

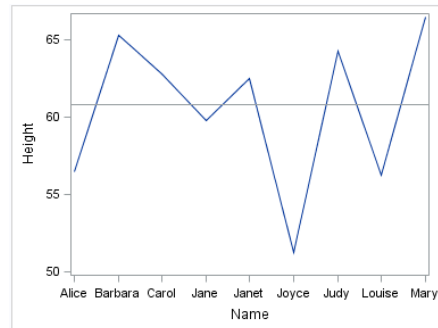
About Lines

About Reference Lines

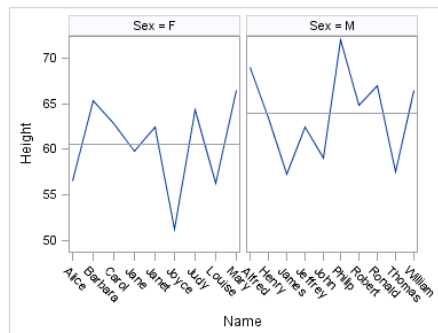
You can add horizontal or vertical reference lines to your graphics. You can draw a reference line for each value of a specified variable. Or you can specify one or more explicit values for the reference lines.

The following examples show the height values for a class of students. A horizontal reference line is overlaid on a series plot to show the average height. Examples are provided for the SGPLOT and the SGPANEL procedures.

In the first example, a value of 60.8 is specified for the reference line. The second example uses the MEANS procedure to obtain the averages for males and females in the class. The SGPANEL procedure then specifies the variable that contains these averages in order to obtain the reference lines.



```
proc sgplot data=sashelp.class;
  where (sex="F");
  series x=name y=height;
  refile 60.8;
run;
```



```
proc means data=sashelp.class mean;
  var height;
  class sex;
  output out= classAverages
    mean=averageWeight;
run;

data classHeightAverage;
  set sashelp.class work.classAverages;
run;

proc sgpanel data=classHeightAverage;
  panelby sex / uniscale=row
    spacing=8;
  series x=name y=height;
  refile averageWeight;
run;
```

Options are available that enable you to customize the reference line and enhance its appearance. For example, you can do the following:

- specify a horizontal or vertical line. In the SGPLOT procedure, you can associate the line with a secondary axis.
- specify line attributes, labels, and label attributes.
- specify legend labels and line transparency.
- specify an amount to offset all lines from discrete axis values.
- extend the plot axes to contain the reference lines.

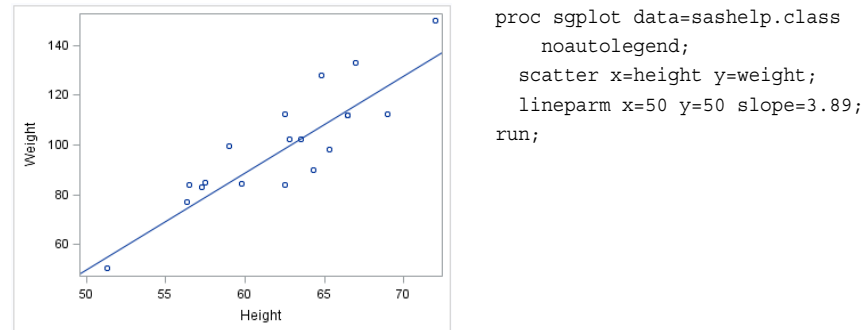
Note: This list does not include all available options.

About Parameterized Lines

Parameterized lines are straight lines specified by a point and a slope. The statement must be used with another plot statement that is derived from data values that provide

boundaries for the axis area. For example, the LINEPARM statement can be used with a scatter plot or a histogram.

The following example shows weight with respect to height for a class of students. A single line is generated by specifying values for the point and for the slope. The line in the example approximates a line of best fit.



You can generate multiple lines by specifying a numeric variable for any or all required arguments. Examples are provided for the SGPLOT and the SG PANEL procedures. The following two examples create lines of best fit for male and female participants in a heart disease study. The lines show weight with respect to height.

The examples first sort the data set by male and female participants. The sorted data is output to a data set named HEART.

```
proc sort data=sashelp.heart(keep=height weight sex)
    out=heart;
    by sex;
run;
```

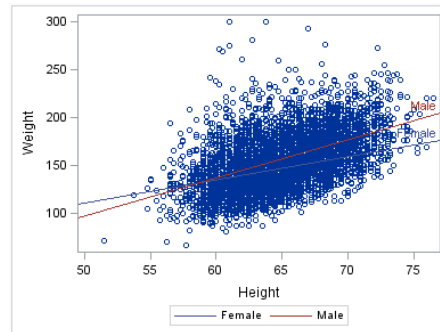
The examples then use the REG procedure and output the regression statistics to a data set named STATS. The STATS data set includes the slope and the Y-intercept for the regression.

```
proc reg data=heart
    outest=stats(rename=(height=slope));
    by sex;
    model weight=height;
run;
```

Finally, the examples merge the HEART and the STATS data sets.

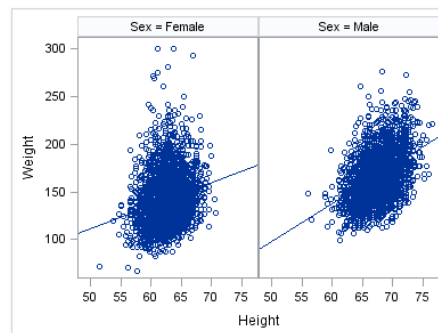
```
data heartStats;
    merge heart stats(keep=intercept slope sex);
run;
```

The first example uses the SGPLOT procedure to show lines of best fit for females and males in the study. The regression lines are labeled and have their own legend.



```
proc sgplot data=heartStats;
  scatter x=height y=weight;
  lineparm x=0 y=intercept slope=slope /
    name="Line" group=sex
    curvelabel;
  keylegend "Line";
run;
```

The following example uses the SGPANEL procedure to create the same information, which is paneled by gender.



```
proc sgpanel data=heartStats
  noautolegend;
  panelby sex;
  scatter x=height y=weight;
  lineparm x=0 y=intercept slope=slope;
run;
```

Options are available that enable you to customize the line and enhance its appearance. For example, you can do the following:

- specify line attributes, labels, and label attributes
- specify legend labels and line transparency
- prevent the line from being extended beyond the axis offset

Note: This list does not include all available options.

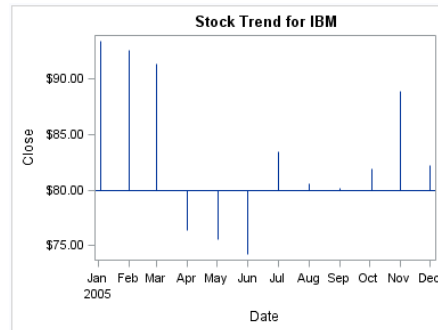
See Also

- [“REFLINE Statement” on page 183](#) (SGPANEL procedure)
- [“REFLINE Statement” on page 390](#) (SGPLOT procedure)
- [“LINEPARM Statement” on page 161](#) (SGPANEL procedure)
- [“LINEPARM Statement” on page 366](#) (SGPLOT procedure)

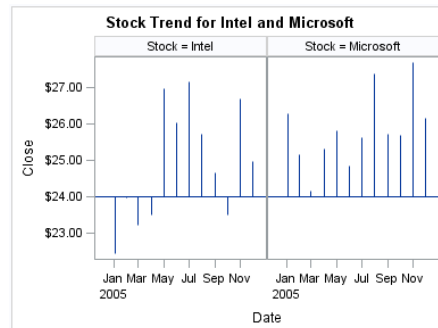
About Needle Plots

Needle plots use vertical line segments, or needles, to connect each data point to a baseline.

The following examples show the stock trend during a particular year. Examples are provided for the SGPLOT and the SGPANEL procedures. Each example specifies an optional baseline value on the Y axis.



```
title "Stock Trend for IBM";
proc sgplot data=sashelp.stocks
  (where=(date >= "01jan2005"d and stock = "IBM"));
  needle x=date y=close / baseline=80;
run;
title;
```



```
title "Stock Trend for Intel and Microsoft";
proc sgpanel data=sashelp.stocks
  (where=(date >= "01jan2005"d and
    (stock = "Intel" or stock = "Microsoft")));
  panelby stock;
  needle x=date y=close / baseline=24;
run;
title;
```

Options are available that enable you to customize the needle plot and enhance its appearance. For example, you can do the following:

- specify a baseline value, as shown in the example.
- add markers to the tips of the needles and specify marker attributes.
- add data labels and specify label attributes.
- control the display of grouped data. For example, you can specify whether the groups are overlaid or clustered, the width of each cluster, and the order of needles within a group.
- specify an amount to offset all needle lines from discrete X values.
- specify legend labels, plot transparency, and URLs for Web pages to be displayed when parts of the plot are selected within an HTML page.

Note: This list does not include all available options.

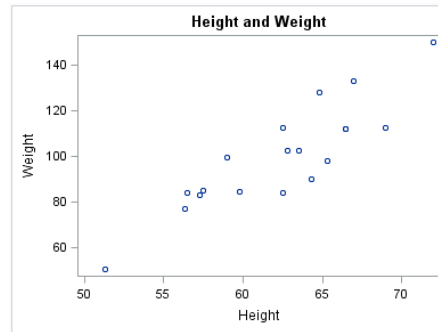
See Also

- [“NEEDLE Statement” on page 171](#) (SGPANEL procedure)
- [“NEEDLE Statement” on page 377](#) (SGPLOT procedure)

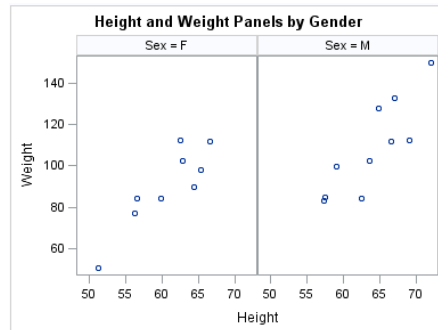
About Scatter Plots

Scatter plots show the relationship of one variable to another, often revealing concentrations or trends in the data. Typically, each variable value on the horizontal axis can have any number of corresponding values on the vertical axis.

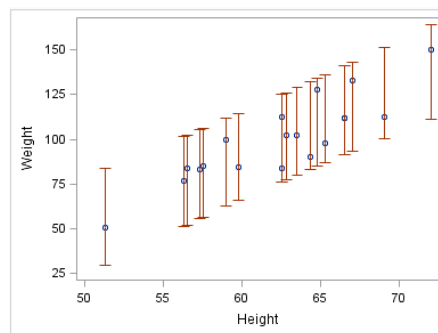
The following examples show the relationship of height to weight for a class of students. Examples are provided for the SGPLOT and the SGPANEL procedures. The third example includes error bars.



```
title "Height and Weight";
proc sgplot data=sashelp.class;
  scatter x=height y=weight;
run;
title;
```



```
title "Height and Weight Panels by Gender";
proc sgpanel data=sashelp.class;
  panelby sex;
  scatter x=height y=weight;
run;
title;
```



```
proc sgplot data=sashelp.classfit;
  scatter x=height y=weight /
    yerrorlower=lower yerrorupper=upper;
run;
```

Options are available that enable you to customize the scatter plot and enhance its appearance. For example, you can do the following:

- add and customize error bars. The previous example shows error bars.
- specify how many times observations are repeated for computational purposes.
- specify marker attributes, data labels, and label attributes.
- specify an amount to offset all markers from discrete axis values.
- control the display of grouped data. For example, you can specify whether the groups are overlaid or clustered, the width of each cluster, and the order of markers within a group
- specify legend labels, plot transparency, and URLs for Web pages to be displayed when parts of the plot are selected within an HTML page.

Note: This list does not include all available options.

See Also

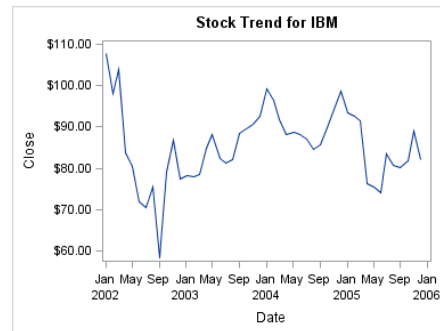
- [“SCATTER Statement” on page 192](#) (SGPANEL procedure)

- “SCATTER Statement” on page 399 (SGPLOT procedure)

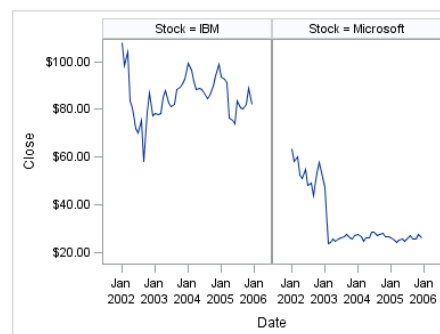
About Series Plots

Series plots display a series of line segments that connect observations of input data.

The following examples show series plots of stock trends. Examples are provided for the SGPLOT and the SGPPANEL procedures.



```
title "Stock Trend for IBM";
proc sgplot data=sashelp.stocks
  (where=(date >= "01jan2002"d and stock = "IBM"));
  series x=date y=close;
run;
title;
```



```
title "Stock Trend for IBM and Microsoft";
proc sgpanel data=sashelp.stocks
  (where=(date >= "01jan2002"d and
    (stock = "IBM" or stock = "Microsoft")));
  panelby stock;
  series x=date y=close;
run;
title;
```

Options are available that enable you to customize the series plot and enhance its appearance. For example, you can do the following:

- specify line and marker attributes, data labels, curve labels, and label attributes.
- specify an amount to offset all markers from discrete axis values.
- control the display of grouped data. For example, you can specify whether the groups are overlaid or clustered, the width of each cluster, and the order of lines within a group
- specify legend labels, plot transparency, and URLs for Web pages to be displayed when parts of the plot are selected within an HTML page.

Note: This list does not include all available options.

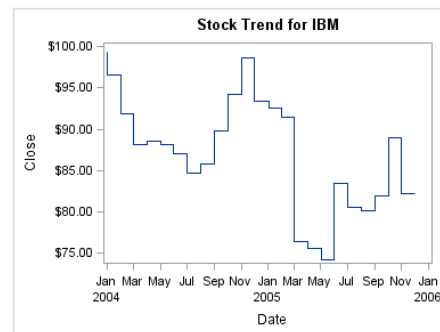
See Also

- “SERIES Statement” on page 198 (SGPPANEL procedure)
- “SERIES Statement” on page 406 (SGPLOT procedure)

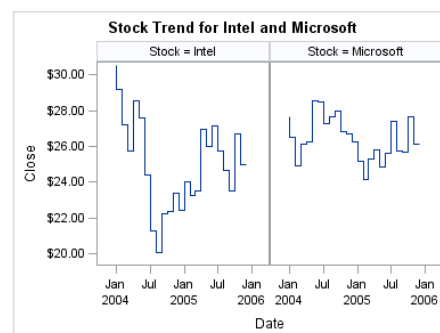
About Step Plots

Step plots display a series of horizontal and vertical line segments that connect observations of input data. The plots use a step function to connect the data points. The vertical line can change at each step.

The following examples show step plots of stock trends. Examples are provided for the SGPLOT and the SGPANEL procedures.



```
title "Stock Trend for IBM";
proc sgplot data=sashelp.stocks
  (where=(date >= "01jan2004"d and stock = "IBM"));
  step x=date y=close;
run;
```



```
title "Stock Trend for Intel and Microsoft";
proc sgpanel data=sashelp.stocks
  (where=(date >= "01jan2004"d and
    (stock = "Intel" or stock = "Microsoft")));
  panelby stock;
  step x=date y=close;
run;
title;
```

Options are available that enable you to customize the step plot and enhance its appearance. For example, you can do the following:

- add and customize markers and error bars.
- specify line attributes, data labels, curve labels, and label attributes.
- specify an amount to offset all step lines from discrete axis values.
- control the display of grouped data. For example, you can specify whether the groups are overlaid or clustered, the width of each cluster, and the order of lines within a group
- specify legend labels, plot transparency, and URLs for Web pages to be displayed when parts of the plot are selected within an HTML page.

Note: This list does not include all available options.

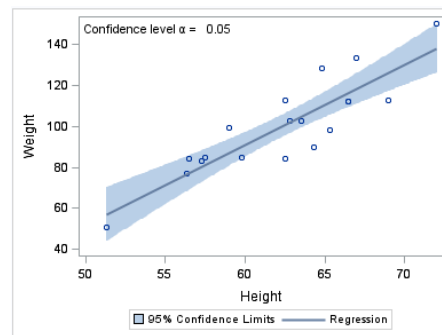
See Also

- [“STEP Statement” on page 204](#) (SGPANEL procedure)
- [“STEP Statement” on page 413](#) (SGPLOT procedure)

About Text Insets

A text inset provides an easy way to add text to a graphic. You can insert a text string as well as a series of label-value pairs.

The following example shows a linear regression curve with a text inset in the upper left corner. Text insets are available only for the SGPLOT procedure. The SGPANEL procedure does not support text insets.



```
proc sgplot data=sashelp.class;
  reg x=height y=weight / CLM;
  inset ("Confidence level (*ESC*)
    {unicode alpha} =" = "0.05");
run;
```

Options are available that enable you to customize the text inset and enhance its appearance. For example, you can do the following:

- show or hide a border
- position the text box within the plot
- specify text attributes, add a title, and specify title attributes
- align the labels and values when you specify label-value pairs

Note: This list does not include all available options.

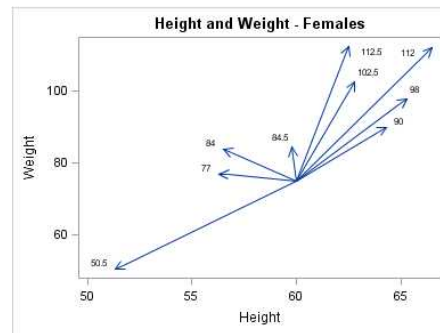
See Also

[“INSET Statement” on page 356](#) (SGPLOT procedure)

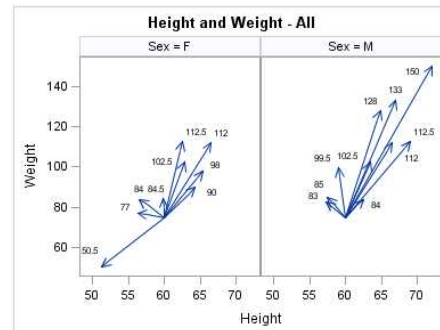
About Vector Plots

Vectors are directed line segments. A vector plot is a two-dimensional graphic that uses vectors to represent both direction and magnitude at each point.

The following examples show the relationship of height to weight for a class of students. Examples are provided for the SGPLOT and the SGPANEL procedures. Both examples specify optional X and Y origins and data labels.



```
title "Height and Weight - Females";
proc sgplot data=sashelp.class;
  where (sex="F");
  vector x=height y=weight
    / yorigin=75 xorigin=60
    datalabel;
run;
title;
```



```
title "Height and Weight - All";
proc sgpanel data=sashelp.class;
  panelby sex;
  vector x=height y=weight
    / yorigin=75 xorigin=60
    datalabel;
run;
title;
```

Options are available that enable you to customize the vector plot and enhance its appearance. For example, you can do the following:

- specify the origin, as shown in the examples.
- specify line attributes, data labels, and data label attributes. You can also show or hide the arrows, change the arrowhead shape, and the change the arrow direction.
- specify legend labels and plot transparency.

Note: This list does not include all available options.

See Also

- [“VECTOR Statement” on page 238](#) (SGPANEL procedure)
- [“VECTOR Statement” on page 447](#) (SGPLOT procedure)

Fit and Confidence Plots

About Fit and Confidence Plots

You can use the SGPLOT and SGPANEL procedures to produce fit plots and ellipses (the ellipses plot is available with the SGPLOT procedure only). Fit plots represent the line of best fit (trend line) with confidence limits.

The plot statements include many options for controlling how the output is displayed. The options that are available depend on the plot type. However, some general options

apply to most of the fit and confidence plots. For example, options enable you to do the following:

- add confidence limits (CLM) to the plot. When you add CLM limits, you can specify the confidence level, the transparency for the confidence limits, and other visual attributes. You can add CLM limits to loess, penalized B-spline, and regression plots.
- add prediction limits (CLI) for the individual predicted values. When you add CLI limits, you can specify the text that appears for the limits and other visual attributes. You can add CLI limits to penalized B-spline and regression plots.
- control the appearance of the markers and the fit line. You can also specify a smoothing parameter.
- add and customize curve and data labels.
- specify legend labels. You can also show or hide the legend entries for the CLM limits, the CLI limits, and the fit line.
- group the data by the values of a variable. A separate plot is created for each unique value of the grouping variable. The plot elements for each group value are automatically distinguished by different visual attributes.
- specify the value of an ID variable in an attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

Note: Not all of these features are available for all of the plots. Also, the list does not include all available options.

The fit and confidence plots are described in the following sections. If you run the examples, your output might differ somewhat depending on the size of your graphics. The examples here were specified to be a particular size using the following statement:

```
ods graphics on / width=4in;
```

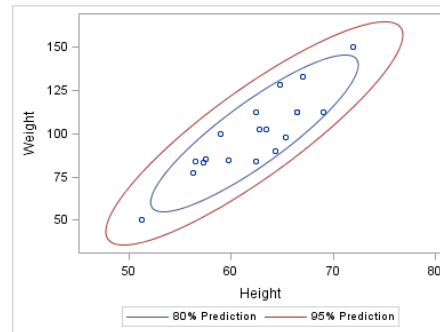
About Ellipse Plots

Ellipse plots create a confidence elliptical curve computed from input data. In order to produce useful output, the ELLIPSE statement should be used with another plot statement that uses numeric axes. Ellipses are available only for the SGPLOT procedure. The SGPANEL procedure does not support ellipses.

The following example shows the relationship of height to weight for a class of students. The example consists of a scatter plot and two ellipses.

Here are the noteworthy features of the example:

- Both ELLIPSE statements use TYPE=PREDICTED. This is the default.
- One ELLIPSE statement uses ALPHA=.2 and the other uses ALPHA=.05.
- The automatically generated legend, which contains an entry for each ellipse and for the scatter plot, has been suppressed. Only the ellipses require a legend.
- A legend was created that contains entries only for the ellipses. Each ELLIPSE statement specifies a legend label and a plot name. The KEYLEGEND statement uses the NAME value to determine the entries in the legend.



```
proc sgplot data=sashelp.class
    noautolegend;
    scatter x=height y=weight;
    ellipse x=height y=weight /
        alpha=.2
        name="eighty"
        legendlabel="80% Prediction";
    ellipse x=height y=weight /
        alpha=.05
        name="ninetyfive"
        legendlabel="95% Prediction";
    keylegend "eighty" "ninetyfive";
run;
```

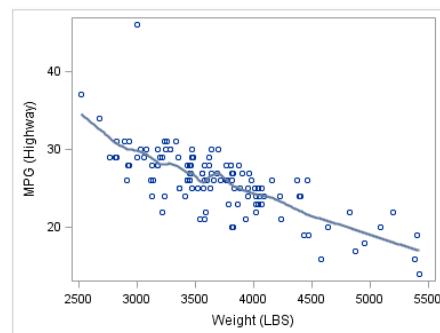
See Also

“[ELLIPSE Statement](#)” on page 306 (SGPLOT procedure)

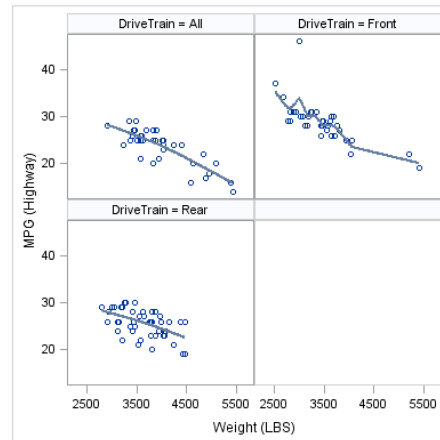
About Loess Plots

A loess plot includes a scatter plot of two numeric variables along with an overlaid nonlinear fit line that enables you to perform locally weighted polynomial regression. You can specify the degree of the local polynomials to use for each local regression. You can also change the default smoothing technique that is applied to the fit.

The following examples show the relationship of height to weight and the line of best fit for a class of students. Examples are provided for the SGPLOT and the SGPanel procedures. In both examples, the automatically generated legend for the fit line is not needed and has been suppressed.



```
proc sgplot data=sashelp.cars
    noautolegend;
    where (origin = "Europe");
    loess x=weight y=mpg_highway;
run;
```



```
proc sgpanel data=sashelp.cars
  noautolegend;
  where (origin = "Europe");
  panelby drivetrain;
  loess x=weight y=mpg_highway;
run;
```

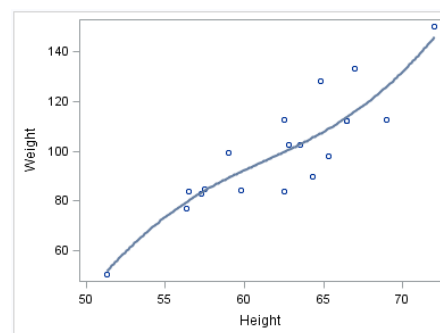
See Also

- [“LOESS Statement” on page 165](#) (SGPANEL procedure)
- [“LOESS Statement” on page 371](#) (SGPLOT procedure)

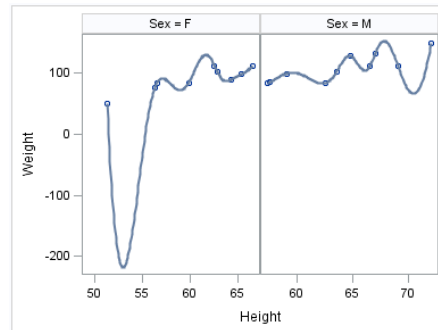
About Penalized B-Spline Plots

A penalized B-spline curve includes a scatter plot of two numeric variables along with an overlaid nonlinear fit line. You can specify the degree of the local polynomials to use for each local regression. You can also change the default smoothing technique that is applied to the fit.

The following examples show the relationship of height to weight and the line of best fit for a class of students. Examples are provided for the SGPLOT and the SGPANEL procedures. In both examples, the automatically generated legend for the fit line is not needed and has been suppressed.



```
proc sgplot data=sashelp.class
  noautolegend;
  pbspline x=height y=weight;
run;
```



```
proc sgpanel data=sashelp.class
  noautolegend;
  panelby sex / uniscale=row;
  pbspline x=height y=weight;
run;
```

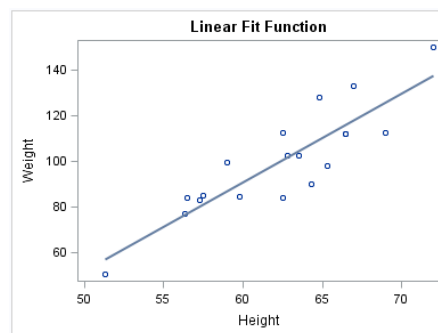
See Also

- “PBSPLINE Statement” on page 176 (SGPANEL procedure)
- “PBSPLINE Statement” on page 383 (SGPLOT procedure)

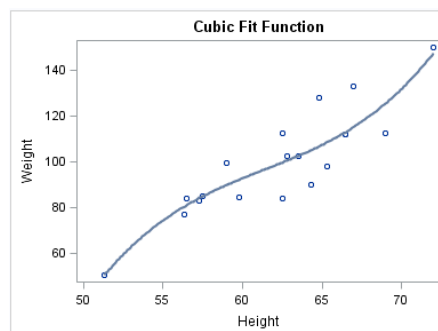
About Regression Plots

A regression plot includes a scatter plot of two numeric variables along with an overlaid linear or nonlinear fit line that enables you to perform a regression analysis. You can specify one of three types of regression equation: linear, quadratic, or cubic. You can display confidence limits for mean predicted values or individual predicted values.

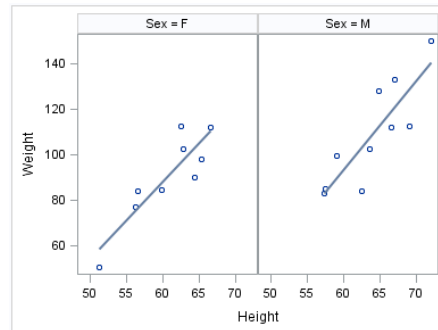
The following examples show the relationship of height to weight and the line of best fit for a class of students. Examples are provided for the SGPLOT and the SGPANEL procedures. The first two examples show the same plot with a linear and a cubic fit line, respectively. The third example shows a paneled graph. In all three examples, the automatically generated legend for the fit line is not needed and has been suppressed.



```
title "Linear Fit Function";
proc sgplot data=sashelp.class
  noautolegend;
  reg x=height y=weight;
run;
title;
```



```
title "Cubic Fit Function";
proc sgplot data=sashelp.class
  noautolegend;
  reg x=height y=weight / degree=3;
run;
title;
```



```
proc sgpanel data=sashelp.class
  noautolegend;
  panelby sex;
  reg x=height y=weight;
run;
```

See Also

- [“REG Statement” on page 185](#) (SGPANEL procedure)
- [“REG Statement” on page 392](#) (SGPLOT procedure)

Distribution Plots

About Distribution Plots

You can use the SGPLOT and SGPPANEL procedures to produce plots that characterize the frequency or the distribution of your data.

The plot statements include many options for controlling how the output is displayed. The options that are available depend on the plot type. The following sections describe each plot and the options that are available.

The distribution plots are described in the following sections. If you run the examples, your output might differ somewhat depending on the size of your graphics. The examples here were specified to be a particular size using the following statement:

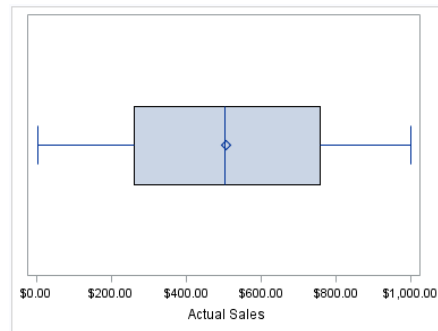
```
ods graphics on / width=4in;
```

About Box Plots

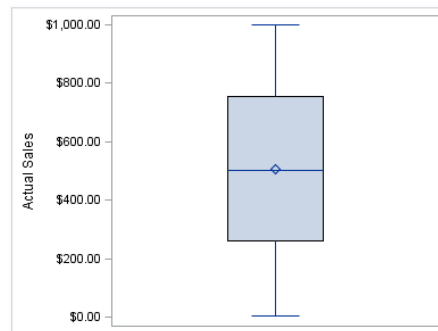
A box plot summarizes the data and indicates the median, upper and lower quartiles, and minimum and maximum values. The plot provides a quick visual summary that easily shows center, spread, range, and any outliers. The SGPLOT and the SGPPANEL procedures have separate statements for creating horizontal and vertical box plots.

The following examples show product sales summaries. Examples are provided for the SGPLOT and the SGPPANEL procedures.

The following two examples use the SGPLOT procedure to create a horizontal and a vertical plot, respectively.

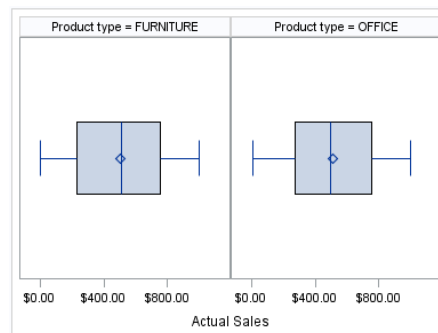


```
proc sgplot data=sashelp.prdsale;
  hbox actual;
run;
```

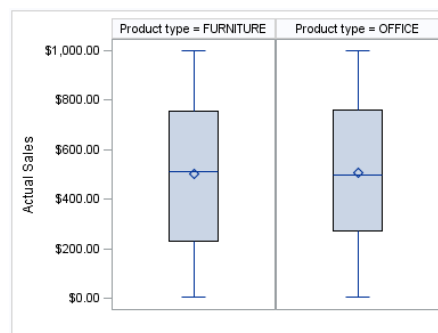


```
proc sgplot data=sashelp.prdsale;
  vbox actual;
run;
```

The following two examples use the SGPANEL procedure to create a horizontal and a vertical plot, respectively. The box plots are paneled by product type.



```
proc sgpanel data=sashelp.prdsale;
  panelby prodtype;
  hbox actual;
run;
```



```
proc sgpanel data=sashelp.prdsale;
  panelby prodtype;
  vbox actual;
run;
```

Options are available that enable you to customize the box plot and enhance its appearance. For example, you can do the following:

- control the box width, the whisker cap shape, and the visual attributes for the mean marker, median line, and the connect lines. You can also hide the whisker caps, mean marker, median line, and the outliers.
- specify data labels and font attributes for the labels.
- specify the method to use for computing the percentiles for the plot.
- group the data by the values of a variable. A separate plot is created for each unique value of the grouping variable. The plot elements for each group value are automatically distinguished by different visual attributes.
- control the display of grouped boxes. For example, you can specify whether the boxes are overlaid or clustered, and the width of each cluster.
- specify an amount to offset graph elements from the category midpoints or from the discrete axis tick marks.
- specify legend labels and plot transparency.
- assign the analysis variable to the secondary axis (X2 or Y2). This option is available only for the SGPLOT procedure.
- specify the value of an ID variable in an attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

Note: This list does not include all available options.

See Also

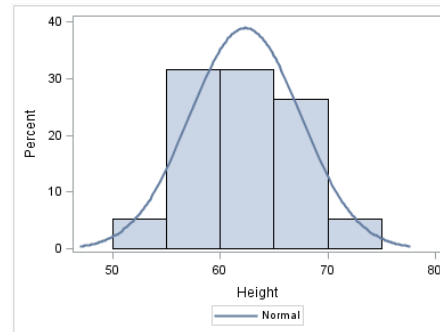
- [“HBOX Statement” on page 126](#) (SGPANEL procedure)
- [“VBOX Statement” on page 227](#) (SGPANEL procedure)
- [“HBOX Statement” on page 324](#) (SGPLOT procedure)
- [“VBOX Statement” on page 436](#) (SGPLOT procedure)

About Density Plots

After creating a histogram, you might use a density plot to fit various distributions to the data. The most common density plot uses the normal distribution, which is defined by the mean and the standard deviation.

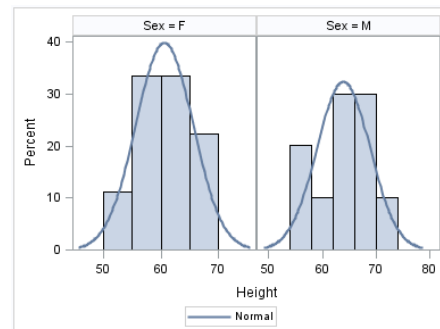
A density plot can be used by itself, combined with another density plot, and overlaid on a histogram.

The following examples show a density plot overlaid on a histogram. Examples are provided for the SGPLOT and the SGPANEL procedures.



```
proc sgplot data=sashelp.class;
  histogram height;
  density height;
run;
```

The SG PANEL example shows output that is paneled by gender. The UNISCALE=ROW option specifies that only the shared row axes are identical. The column axes vary based on the values of the height for the respective genders.



```
proc sgpanel data=sashelp.class;
  panelby sex /
    uniscale=row;
  histogram height;
  density height;
run;
```

Options are available that enable you to customize the density plot and enhance its appearance. For example, you can do the following:

- control the visual attributes of the density line.
- specify a kernel distribution instead of normal. You can also specify the scaling that is used for the response axis.
- specify legend labels and plot transparency.

Note: This list does not include all available options.

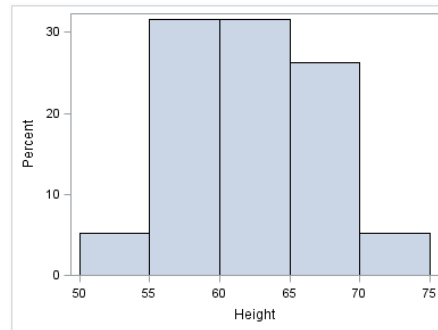
See Also

- “[DENSITY Statement](#)” on page 101 (SGPANEL procedure)
- “[DENSITY Statement](#)” on page 295 (SGPLOT procedure)

About Histograms

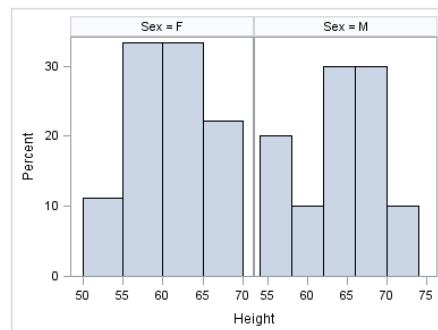
Histograms consist of a series of columns representing the frequency of a variable over a discrete interval or class.

The following examples show the height distribution for a class of students. Examples are provided for the SGPLOT and the SGPANEL procedures.



```
proc sgplot data=sashelp.class;
  histogram height;
run;
```

The SG PANEL example shows output that is paneled by gender. The UNISCALE=ROW option ensures that only the shared row axes are identical. The column axes vary based on the values of the height for the respective genders.



```
proc sgpanel data=sashelp.class;
  panelby sex /
    uniscale=row;
  histogram height;
run;
```

Options are available that enable you to customize the histogram and enhance its appearance. For example, you can do the following:

- control the visual attributes of the bins, such as fill color and outlines.
- specify the number of bins, their width, and the X coordinate of the first bin.
- specify legend labels and plot transparency.
- assign the response variable and the calculated values to the secondary axis (X2 or Y2). This option is available only for the SG PLOT procedure.

Note: This list does not include all available options.

See Also

- [“HISTOGRAM Statement” on page 145](#) (SG PANEL procedure)
- [“HISTOGRAM Statement” on page 343](#) (SG PLOT procedure)

Categorization Plots and Charts

About Categorization Plots and Charts

Categorization plots and charts produce a series of graph elements, one for each selected category of cases. For example, the relation between the age and the risk of a heart attack might differ between males and females. Categorization plots and charts can reveal patterns, complex interactions, exceptions, and anomalies.

You can use the SGPLOT and SGPANEL procedures to produce a variety of categorization plots and charts. The plot and chart statements include many options for controlling how the output is displayed. The options that are available depend on the plot type. The following sections describe each type and the options that are available.

The categorization plots are described in the following sections. If you run the examples, your output might differ somewhat depending on the size of your graphics. The examples here were specified to be a particular size using the following statement:

```
ods graphics on / width=4in;
```

About Bar Charts

Overview of Standard and Parameterized Bar Charts

Bar charts use bars to represent statistics based on the values of a category variable. Bar charts are useful for displaying magnitudes and emphasizing differences.

You can use the SGPLOT and SGPANEL procedures to create the following:

- horizontal and vertical bar charts that summarize the values of a category variable.
- parameterized horizontal and vertical bar charts that require a response variable in addition to the category variable. The response variable contains pre-summarized computed values such as a sum or a mean for each unique value of the category variable.

Options are available that enable you to customize both types of bar charts and enhance their appearance. For example, you can do the following:

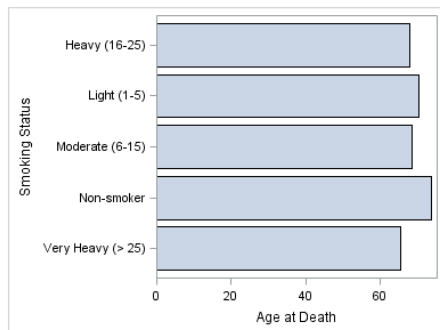
- control the visual attributes of the bars, such as bar width, fill color, fill skin, and outlines.
- add data labels and specify font attributes for the labels.
- group the data by the values of a variable. A separate plot is created for each unique value of the grouping variable. The plot elements for each group value are automatically distinguished by different visual attributes.
- control the display of grouped bars. For example, you can specify the width of each cluster.
- specify an amount to offset graph elements from the category midpoints or from the discrete axis tick marks.
- specify legend labels, plot transparency, and URLs for Web pages to be displayed when parts of the plot are selected within an HTML page.

- specify the value of an ID variable in an attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

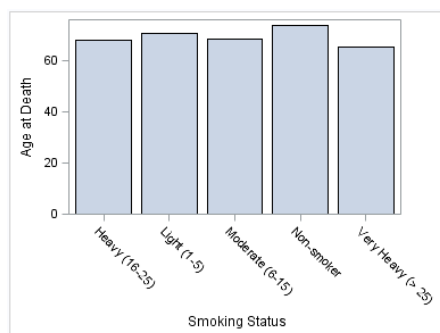
Note: This list does not include all available options.

Bar Chart Examples

The following examples show statistics for different categories of smokers. The examples use the SGPLOT procedure to create a horizontal and a vertical bar chart, respectively. By default, the charts show the frequency for each category. The examples specify an optional response variable to show the average age at death for each category rather than the frequency.

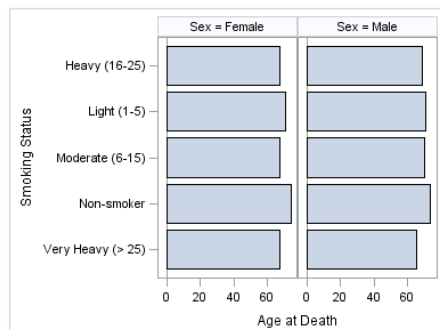


```
proc sgplot data=sashelp.heart;
  hbar smoking_status /
    response=ageatdeath
    stat=mean;
run;
```

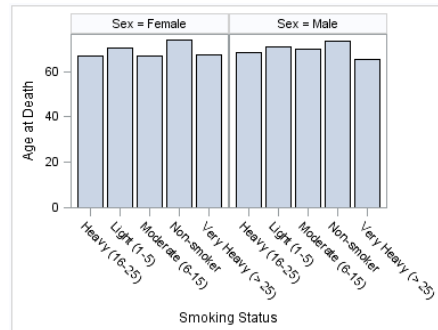


```
proc sgplot data=sashelp.heart;
  vbar smoking_status /
    response=ageatdeath
    stat=mean;
run;
```

The following two examples use the SGPANEL procedure to create a horizontal and a vertical chart, respectively. The bar charts are paneled by gender.



```
proc sgpanel data=sashelp.heart;
  panelby sex;
  hbar smoking_status /
    response=ageatdeath
    stat=mean;
run;
```



```
proc sgpanel data=sashelp.heart;
  panelby sex;
  vbar smoking_status /
    response=ageatdeath
    stat=mean;
run;
```

Bar charts includes options that are not applicable to parameterized bar charts. For example, you can do the following:

- specify the response variable and the statistic to use for its axis
- specify the order in which the response values are arranged
- show limit lines, specify the statistic to use for the limit lines, and specify the confidence level
- for grouped data, you can specify whether the bars are stacked or clustered
- specify how many times observations are repeated for computational purposes

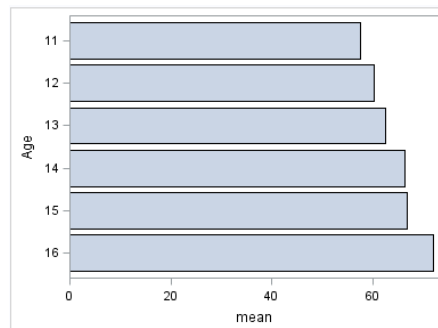
Note: This list does not include all available options.

Parameterized Bar Chart Examples

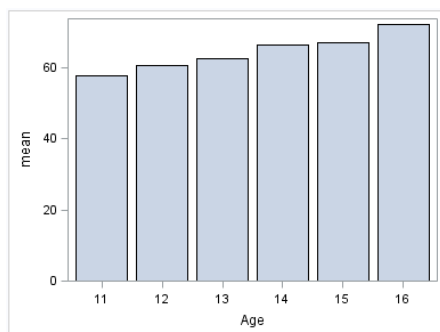
The following examples show height averages for a class of students. The averages are obtained using the following program.

```
proc means data=sashelp.class alpha=.05 clm mean std;
  class age sex;
  var height;
  output out=classMean uclm=uclm lclm=lclm mean=mean;
run;
```

The following two examples use the SGPLOT procedure to create a horizontal and a vertical chart, respectively. The response variable contains the computed mean values that were created with the MEANS procedure.

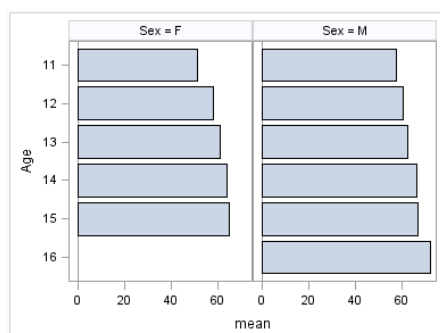


```
proc sgplot data=classMean;
  hbarparm category=age response=mean;
run;
```

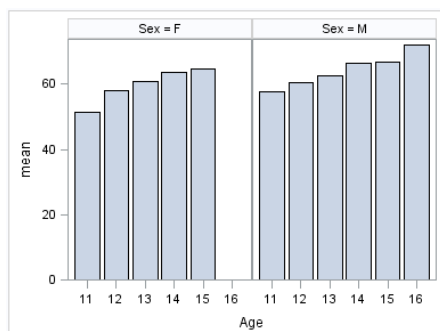


```
proc sgplot data=classMean;
  vbarparm category=age response=mean;
run;
```

The following two examples use the SG PANEL procedure to create horizontal and vertical bar charts, respectively. The charts are paneled by gender.

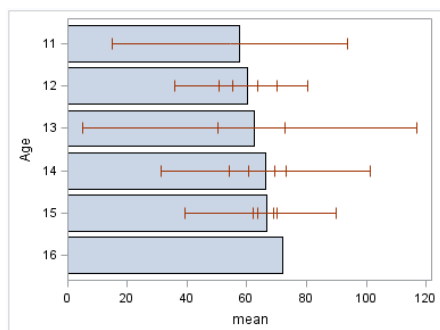


```
proc sgpanel data=classMean;
  panelby sex;
  hbarparm category=age response=mean;
run;
```



```
proc sgpanel data=classMean;
  panelby sex;
  vbarparm category=age response=mean;
run;
```

You can also assign variables to the upper and lower limits of the bar chart. Parameterized bar charts enable you to pass in your own precomputed limits.



```
proc sgplot data=classMean;
  hbarparm category=age response=mean /
    limitlower=lclm
    limitupper=uclm;
run;
```

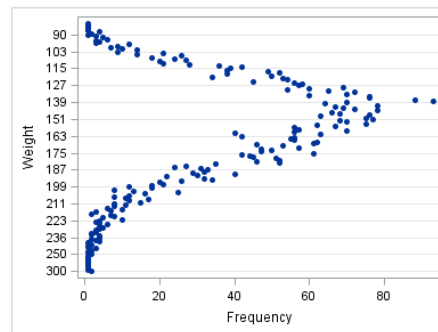
See Also

- “HBAR Statement” on page 111 (SGPANEL procedure)
- “VBAR Statement” on page 212 (SGPANEL procedure)
- “HBAR Statement” on page 309 (SGPLOT procedure)
- “VBAR Statement” on page 420 (SGPLOT procedure)
- “HBARPARM Statement” on page 120 (SGPANEL procedure)
- “VBARPARM Statement” on page 220 (SGPANEL procedure)
- “HBARPARM Statement” on page 317 (SGPLOT procedure)
- “VBARPARM Statement” on page 429 (SGPLOT procedure)

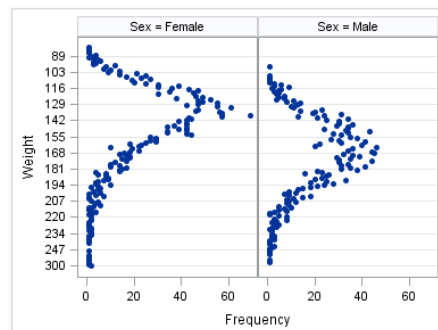
About Dot Plots

Dot plots summarize horizontally the values of a category variable. By default, each dot represents the frequency for each value of the category variable.

The following examples show the frequency of different weights of patients in a study. The examples use the SGPLOT and the SGPANEL procedures.



```
proc sgplot data=sashelp.heart;
    dot weight;
run;
```



```
proc sgpanel data=sashelp.heart;
    panelby sex;
    dot weight;
run;
```

Options are available that enable you to customize the dot plot and enhance its appearance. For example, you can do the following:

- specify an optional response variable and show the mean, the sum, or the frequency for that variable. You can also specify the order in which the response values are arranged.
- show limits for the plot. You can also specify the statistic for the limit lines and visual attributes of the lines.

- specify the color, size, and symbol for the markers.
- add data labels and specify font attributes for the labels.
- control the display of grouped markers, lines, and bars. For example, you can specify whether the groups are overlaid or clustered, and the ordering of dots within a group.
- specify an amount to offset graph elements from the category midpoints or from the discrete axis tick marks.
- specify legend labels, plot transparency, and URLs for Web pages to be displayed when parts of the plot are selected within an HTML page.

Note: This list does not include all available options.

See Also

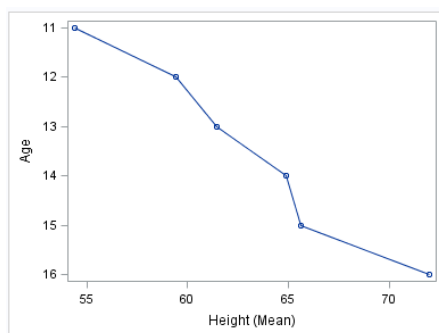
- [“DOT Statement” on page 104](#) (SGPANEL procedure)
- [“DOT Statement” on page 299](#) (SGPLOT procedure)

About Line Charts

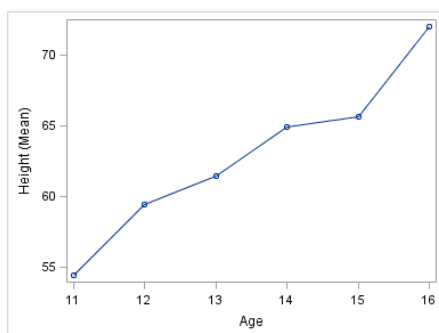
Line charts displays information as a series of data points connected by straight line segments. The SGPLOT and the SGPANEL procedures have separate statements for creating horizontal and vertical line charts.

The following examples show mean weight values for a class. Examples are provided for the SGPLOT and the SGPANEL procedures. The examples specify an optional response variable and use the mean statistic for that variable. The examples also add data point markers.

These two examples use the SGPLOT procedure to create a horizontal and a vertical chart, respectively.

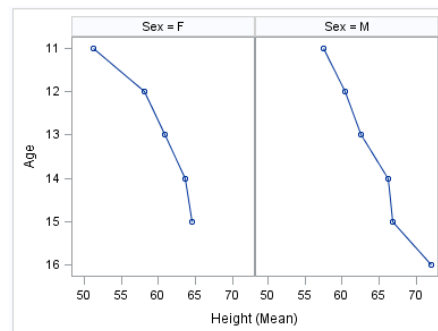


```
proc sgplot data=sashelp.class;
  hline age /
    response=height
    stat=mean
    markers;
run;
```

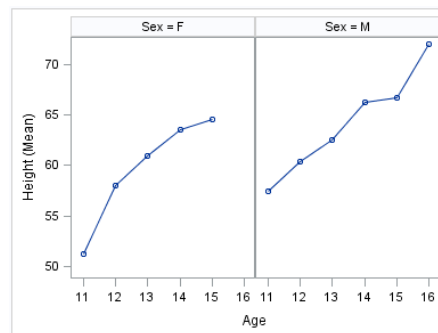


```
proc sgplot data=sashelp.class;
  vline age /
    response=height
    stat=mean
    markers;
run;
```

The following two examples use the SG PANEL procedure to create panels of horizontal and vertical charts, respectively.



```
proc sgpanel data=sashelp.class;
  panelby sex;
  hline age /
    response=height
    stat=mean
    markers;
run;
```



```
proc sgpanel data=sashelp.class;
  panelby sex;
  vline age /
    response=height
    stat=mean
    markers;
run;
```

Options are available that enable you to customize the line chart and enhance its appearance. For example, you can do the following:

- specify an optional response variable and show the mean, the sum, or the frequency for that variable. You can also specify the order in which the response values are arranged.
- show limits for the chart. You can also specify the statistic for the limit lines and visual attributes of the lines.
- add data point markers and specify the color, size, and symbol for the markers.
- add curve and data labels and specify font attributes for the labels.
- control the display of grouped lines. For example, you can specify whether the groups are overlaid or clustered, the width of each cluster, and the ordering of lines within a group.
- specify an amount to offset graph elements from the category midpoints or from the discrete axis tick marks.
- specify legend labels, plot transparency, and URLs for Web pages to be displayed when parts of the plot are selected within an HTML page.
- assign the category variable, the response variable, or both variables to the secondary axis (X2 or Y2). This option is available only for the SG PLOT procedure.
- specify the value of an ID variable in an attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

Note: This list does not include all available options.

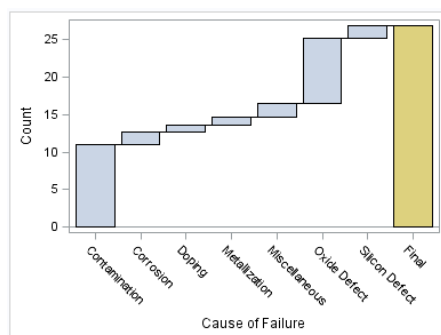
See Also

- “HLINE Statement” on page 148 (SGPANEL procedure)
- “VLINE Statement” on page 242 (SGPANEL procedure)
- “HLINE Statement” on page 346 (SGPLOT procedure)
- “VLINE Statement” on page 451 (SGPLOT procedure)

About Waterfall Charts (Preproduction)

Waterfall charts show how the value of a variable increases or decreases until it reaches a final value. In the chart, bars represent an initial value of Y and a series of intermediate values identified by X leading to a final value of Y. Waterfall charts are available only for the SGPLOT procedure.

The following example shows average failure counts for capacitors.



```
proc sgplot data=sashelp.failure;
  waterfall category=cause response=count
    / stat=mean;
run;
```

Options are available that enable you to customize the waterfall chart and enhance its appearance. For example, you can do the following:

- specify the statistic for the response variable.
- specify an initial bar for the chart. You can also specify the tick value that is used for the initial bar and visual attributes of the bar.
- control the appearance of the bars. For example, you can do the following:
 - show or hide the bar outline
 - show or hide the bar fill
 - use a special effect (data skin) for the fill
 - specify a variable to use for the bar colors
 - specify attributes separately for the final bar
- add data labels and specify font attributes for the labels.
- specify plot transparency and URLs for Web pages to be displayed when parts of the plot are selected within an HTML page.

Note: This list does not include all available options.

See Also

“WATERFALL (Preproduction) Statement” on page 462 (SGPLOT procedure)

Chapter 4

SAS Statements That Are Used with ODS Graphics Procedures

Overview of SAS Statements That Are Used with ODS Graphics Procedures . . .	51
Dictionary	52
BY Statement	52
FOOTNOTE Statement	53
FORMAT Statement	54
LABEL Statement	57
ODS GRAPHICS Statement	57
TITLE and FOOTNOTE Statements	58
WHERE Statement	63

Overview of SAS Statements That Are Used with ODS Graphics Procedures

The SAS ODS Graphics procedures support these statements in addition to statements that are unique to each procedure:

BY

processes your data by using one or more classification variables, and produces a separate graph for each unique combination of values.

FORMAT

associates SAS formats or user-defined formats with variables.

FOOTNOTE

adds footnotes to your graphs.

LABEL

associates descriptive labels with variables.

ODS GRAPHICS

enables you manage the settings for your graphics output.

TITLE

adds titles to your graphs.

WHERE

selects observations from SAS data sets that meet a particular condition.

The ODS GRAPHICS, TITLE, and FOOTNOTE statements are global statements. That is, they can be specified anywhere in your program and they remain in effect until you explicitly cancel or change them. The BY, FORMAT, and LABEL statements are associated with a specific procedure step.

Note: Some of the statements that can be used with traditional SAS procedures are not used with the ODS Graphics procedures.

Dictionary

BY Statement

Creates a separate graph for each BY group.

Used by: SGDESIGN, SGPANEL, SGPLOT, SGRENDER, and SGSCATTER procedures

Syntax

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

Required Argument

variable

specifies the variable that the procedure uses to form BY groups. You can specify more than one variable. By default, observations in the data set must either be sorted in ascending order by all the variables that you specify, or be indexed appropriately.

Optional Arguments

DESCENDING

specifies that the data set is sorted in descending order by the specified variable. This option affects only the variable that immediately follows it—you must specify the DESCENDING option before each variable that is sorted in descending order. For example, the following code specifies a BY group that uses two variables that are both sorted in descending order:

```
by descending variable1 descending variable2;
```

NOTSORTED

specifies that the observations in the data set that have the same BY values are grouped together, but are not necessarily sorted in alphabetical or numeric order. For example, the observations might be sorted in chronological order using a date format such as DDMMYY.

The NOTSORTED option applies to all of the variables in the BY statement. You can specify the NOTSORTED option anywhere within the BY statement.

The requirement for ordering or indexing observations according to the values of BY variables is suspended when you use the NOTSORTED option. In fact, the procedure does not use an index if you specify the NOTSORTED option. For the NOTSORTED option, the procedure defines a BY group as a set of contiguous observations that have the same values for all BY variables. If observations that have the same value for the BY variables are not contiguous, then the procedure treats each new value that it encounters as the first observation in a new BY group. The procedure then creates a graph for that value.

Restriction: The NOTSORTED option is not supported by the SGPANEL procedure.

Details

Preparing Data for BY-Group Processing

Unless you specify the NOTSORTED or DESCENDING options, observations in the input data set must be in ascending numeric or alphabetic order. To prepare the data set, sort it with the SORT procedure using the same BY statement that you plan to use in the target procedure. Alternatively, you can create an appropriate index on the BY variables. For more information about indexes, see “Understanding SAS Indexes” in Chapter 26 of *SAS Language Reference: Concepts*.

If the procedure encounters an observation that is out of order, an error message is generated.

If you need to group data in some other order, such as chronological order, you can still use BY-group processing. To do so, process the data so that observations are arranged in contiguous groups that have the same BY-variable values and specify the NOTSORTED option in the BY statement.

Controlling BY Lines

By default, the BY statement prints a BY line above each graph that contains the variable name followed by an equal sign and the variable value. For example, if you specify BY SITE in the procedure, the default heading when the value of SITE is London would be SITE=London.

To suppress the BY line, use the NOBYLINE option in an OPTION statement.

To display only the BY value, use the NOBYLINE option and then use the #BYVAL1 substitution in a TITLE statement.

Using the BY Statement with the SGPLOT Procedure

You can use the UNIFORM= option in the PROC SGPLOT statement to produce the same group markers, the same axis scaling, or both for all graphs in a BY group. By default, the group markers and axis scales might vary from graph to graph.

Using the BY Statement with the TITLE and FOOTNOTE Statements

The TITLE and FOOTNOTE statements can automatically include the BY variable name, BY variable values, or BY lines in the text that they produce. To insert BY variable information into the text strings used by these statements, use the #BYVAR, #BYVAL, and #BYLINE substitution options. For more information, see the description for the *text-string* argument in [“TITLE and FOOTNOTE Statements” on page 58](#).

FOOTNOTE Statement

Writes up to 10 lines of text at the bottom of the graph.

Valid in: anywhere in your program

Used by: SGPLOT, SGPANEL, and SGSCATTER procedures

See: [“TITLE and FOOTNOTE Statements” on page 58](#)

Syntax

```
FOOTNOTE<1...10> <text-options> <"text-string-1"> ... <text-options> <"text-string-n"> ;
```

FORMAT Statement

Associates SAS formats or user-defined formats with variables.

Used by: SGDESIGN, SGPANEL, SGPLOT, SGRENDER, and SGSCATTER procedures

Details

All features of the FORMAT statement are supported. For more information, see “FORMAT Statement” in *SAS Statements: Reference*.

The following SAS formats are supported by the ODS Graphics procedures:

Table 4.1 Character Formats Supported By Java

\$	\$ASCII	\$BINARY	\$CHAR
\$F	\$HEX	\$OCTAL	

Table 4.2 Numeric Formats Supported By Java

BEST	BINARY	COMMA	COMMAX	COMMAX
D	DOLLAR	DOLLARX	E	EURO
EUROX	F	HEX	LOGPROB	NEGPAREN
NLBEST	NLD	NLMNIAED	NLMNIAUD	NLMNIBGN
NLMNIBRL	NLMNICAD	NLMNICHF	NLMNICNY	NLMNICZK
NLMNIDKK	NLMNIEEK	NLMNIEGP	NLMNIEUR	NLMNIGBP
NLMNIHKD	NLMNIHRK	NLMNIHUF	NLMNIIDR	NLMNIILS
NLMNIINR	NLMNIJPY	NLMNIKRW	NLMNITL	NLMNILVL
NLMNIMOP	NLMNIMXN	NLMNIMYR	NLMNINOK	NLMNINZD
NLMNIPLN	NLMNIROL	NLMNIRUB	NLMNIRUR	NLMNISEK
NLMNISGD	NLMNISKK	NLMNITHB	NLMNITRY	NLMNITWD
NLMNIUSD	NLMNIZAR	NLMNLAED	NLMNLAUD	NLMNLBGN
NLMNLBRL	NLMNLCAD	NLMNLCHF	NLMNLCNY	NLMNLCZK
NLMNLDDK	NLMNLEEK	NLMNLEGP	NLMNLEUR	NLMNLGBP
NLMNLHKD	NLMNLHRK	NLMNLHUF	NLMNLIDR	NLMNLILS

NLMNLINR	NLMNLJPY	NLMNLKRW	NLMNLLTL	NLMNLLVL
NLMNLMOP	NLMNLMXN	NLMNLMYR	NLMNLNOK	NLMNLNZD
NLMNLPLN	NLMNLROL	NLMNLRUB	NLMNLRUR	NLMNLSEK
NLMNLSGD	NLMNLSKK	NLMNLTHB	NLMNLTRY	NLMNLTWD
NLMNLUSD	NLMNLZAR	NLMNY	NLMNYI	NLNUM
NLNUMI	NLPCT	NLPCTI	NLPVALUE	NUMX
OCTAL	PERCENT	PERCENTN	PVALUE	ROMAN
RSTDOCNY	RSTDOCYY	RSTDONYN	RSTDOPNY	RSTDOPYN
RSTDOPYY	YEN			

Table 4.3 Date and Time Formats Supported By Java

AFRDFDD	AFRDFDE	AFRDFDN	AFRDFDT	AFRDFDWN
AFRDFMN	AFRDFMY	AFRDFWDX	AFRDFWKX	CATDFDD
CATDFDE	CATDFDN	CATDFDT	CATDFDWN	CATDFMN
CATDFMY	CATDFWDX	CATDFWKX	CRODFDD	CRODFDE
CRODFDN	CRODFDT	CRODFDWN	CRODFMN	CRODFMY
CRODFWDX	CRODFWKX	CSYDFDD	CSYDFDE	CSYDFDN
CSYDFDT	CSYDFDWN	CSYDFMN	CSYDFMY	CSYDFWDX
CSYDFWKX	DANDFDD	DANDFDE	DANDFDN	DANDFDT
DANDFDWN	DANDFMN	DANDFMY	DANDFWDX	DANDFWKX
DATE	DATEAMPM	DATETIME	DAY	DDMMYY
DDMMYYN	DESDFDD	DESDFDE	DESDFDN	DESDFDT
DESDFDWN	DESDFMN	DESDFMY	DESDFWDX	DESDFWKX
DEUDFDD	DEUDFDE	DEUDFDN	DEUDFDT	DEUDFDWN
DEUDFMN	DEUDFMY	DEUDFWDX	DEUDFWKX	DOWNAME
DTDATE	DTMONYY	DTWKDATX	DTYEAR	DTYYQC
ENGDFDD	ENGDFDE	ENGDFDN	ENGDFDT	ENGDFDWN
ENGDFMN	ENGDFMY	ENGDFWDX	ENGDFWKX	ESPDFDD

ESPDFDE	ESPDFDN	ESPDFDT	ESPDFDWN	ESPDMN
ESPDMY	ESPFDWX	ESPFDWKX	EURDFDD	EURDFDE
EURDFDN	EURDFDT	EURDFDWN	EURDFMN	EURDFMY
EURDFWX	EURDFWKX	FINDFDD	FINDFDE	FINDFDN
FINDFDT	FINDFDWN	FINDFMN	FINDFMY	FINDFDWX
FINDFWKX	FRADFDD	FRADFDE	FRADFND	FRADFDT
FRADFND	FRADFMN	FRADFMY	FRADFWDX	FRADFWKX
FRSDFDD	FRSDFDE	FRSDFDN	FRSDFDT	FRSDFDWN
FRSDFMN	FRSDFMY	FRSDFWDX	FRSDFWKX	HHMM
HOUR	HUNDFDD	HUNDFDE	HUNDFDN	HUNDFDT
HUNDFDWN	HUNDFMN	HUNDFMY	HUNDFWDX	HUNDFWKX
ITADFDD	ITADFDE	ITADFND	ITADFDT	ITADFND
ITADFMN	ITADFMY	ITADFWDX	ITADFWKX	JDATEMD
JDATEMON	JDATEQRW	JDATEQTR	JDATESEM	JDATESMW
JULDATE	JULDAY	JULIAN	MACDFDD	MACDFDE
MACDFND	MACDFDT	MACDFDWN	MACDFMN	MACDFMY
MACDFWDX	MACDFWKX	MMDDYY	MMDDYYN	MMSS
MMYY	MMYYN	MONNAME	MONTH	MONYY
NLDATE	NLDATEMD	NLDATEMN	NLDATEW	NLDATEWN
NLDATEYM	NLDATEYQ	NLDATEYR	NLDATEYW	NLDATM
NLDATMAP	NLDATMDT	NLDATMMD	NLDATMTM	NLDATMTZ
NLDATMW	NLDATMWN	NLDATMWZ	NLDATMYM	NLDATMYQ
NLDATMYR	NLDATMYW	NLDATMZ	NLDDFDD	NLDDFDE
NLDDFDN	NLDDFDT	NLDDFDWN	NLDDFMN	NLDDFMY
NLDDFDWX	NLDDFWKX	NLSTRMON	NLSTRQTR	NLSTRWK
NLTIMAP	NLTIME	NORDFDD	NORDFDE	NORDFDN
NORDFDT	NORDFDWN	NORDFMN	NORDFMY	NORDFDWX

NORDFWKX	POLDFD	POLDFDE	POLDFDN	POLDFDT
POLDFDWN	POLDFMN	POLDFMY	POLDFWDX	POLDFWKX
PTGDFD	PTGDFDE	PTGDFDN	PTGDFDT	PTGDFDWN
PTGDFMN	PTGDFMY	PTGDFWDX	PTGDFWKX	QTR
QTRR	RUSDFD	RUSDFDE	RUSDFDN	RUSDFDT
RUSDFDWN	RUSDFMN	RUSDFMY	RUSDFWDX	RUSDFWKX
SLODFD	SLODFDE	SLODFDN	SLODFDT	SLODFDWN
SLODFMN	SLODFMY	SLODFWDX	SLODFWKX	SVEDFDD
SVEDFDE	SVEDFDN	SVEDFDT	SVEDFDWN	SVEDFMN
SVEDFMY	SVEDFWDX	SVEDFWKX	TIME	TIMEAMPM
TOD	WEEKDATE	WEEKDATX	WEEKDAY	WEEKU
WEEKV	WEEKW	WORDDATE	WORDDATX	YEAR
YYMM	YYMMDD	YYMMDDN	YYMMN	YYMON
YYQ	YYQN	YYQR	YYQRN	YYWEEKU
YYWEEKV	YYWEEKW			

LABEL Statement

Associates descriptive labels with variables.

Used by: SGDESIGN, SGPANEL, SGPLOT, SGRENDER, and SGSCATTER procedures

Details

All features of the LABEL statement are supported. For more information, see “LABEL Statement” in *SAS Statements: Reference*.

ODS GRAPHICS Statement

Specifies the settings for your graphics output.

Valid in: anywhere in your program

Used by: SGDESIGN, SGPANEL, SGPLOT, SGRENDER, and SGSCATTER procedures

Details

For information about using the ODS GRAPHICS statement, see [“Using the ODS GRAPHICS Statement” on page 651](#). For the complete statement syntax, see “ODS GRAPHICS Statement” in *SAS Output Delivery System: User's Guide*.

TITLE and FOOTNOTE Statements

The TITLE and FOOTNOTE statements control the content, appearance, and placement of title and footnote text.

Valid in: anywhere in your program

Used by: SGPLOT, SGPANEL, and SGSCATTER procedures

Syntax

TITLE<1...10> <text-options> <"text-string-1"> ... <text-options> <"text-string-n"> ;

FOOTNOTE<1...10> <text-options> <"text-string-1"> ... <text-options> <"text-string-n"> ;

Required Argument

text-string

is a text string that can contain up to 200 characters. You must enclose text strings in either single or double quotation marks. The text appears exactly as you type it in the statement, including uppercase and lowercase characters and spaces. Titles and footnotes automatically wrap to additional lines if necessary.

To use single quotation marks or apostrophes within the title, you can either:

- use a pair of single quotation marks together:

```
footnote 'All''s well that ends well';
```

- enclose the text in double quotation marks:

```
footnote "All's well that ends well";
```

Because the FOOTNOTE and TITLE statements concatenate all text strings, the strings must contain the correct spacing. With a series of strings, add spaces to the beginning of a text string rather than at the end, as in this example:

```
footnote color=red "Sales:" color=blue " 2000";
```

With fonts that support Unicode, you can produce specific characters by specifying a hexadecimal value. A trailing **x** identifies a string as a hexadecimal value. You must also enclose the character specification in a special ODS handler string, in the format **(*ESC*){unicode 'hexadecimal-value'x}**. For example:

```
title "Regression with Confidence Limits ( (*ESC*){unicode '03B1'x}=.05 )";
```

This statement produces the title, "Regression with Confidence Limits ($\alpha = .05$)" because '03B1'x is the hexadecimal value for the lowercase Greek letter alpha in all Unicode fonts.

In addition, if you are using a BY statement, then you can include special options. For more information, see [“Substituting BY Line Values in a Text String” on page 62](#).

Note: The LISTING destination does not honor the (*ESC*) statement.

Optional Arguments

BOLD

specifies that the font weight is bold for the text string.

Default: For titles, the default font weight is specified by the FONTWEIGHT attribute of the GraphTitleText style element in the current style.

For footnotes, the default font weight is specified by the FONTWEIGHT attribute of the GraphFootnoteText style element in the current style.

BCOLOR= *color*

specifies the background color for a box that you created with the BOX= option. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

This option has no effect if you do not also specify the BOX= option. By default, the background color is the same color as the background of the graph.

Alias: BC=

BOX= 1 | 2 | 3 | 4

draws a box around one line of text. Specify a value between 1 and 4, where 1 specifies the thinnest line and 4 specifies the thickest line. Only the last BOX= option is used. The color of the box outline is determined by the GraphBorderLines element of the current style.

Alias: BO

BSPACE=*numeric-value*<*units*>

specifies the amount of space between the text and the border of a box that you create with the BOX= option.

You can also specify the unit of measure. See “[Measurement Units for TITLE and FOOTNOTE Statement Options](#)” on page 63 for a list of the units that are supported.

If you do not specify a unit, then the size of the space is approximately 12*n* points. For example, if you specify BSPACE=2, then the space is approximately 24 points.

Alias: BS=

Default: 0

COLOR= *color*

specifies the color for the text. The COLOR= option affects all of the text strings that follow it in your TITLE or FOOTNOTE statement. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

You can use multiple colors by specifying multiple COLOR= options. For example, the following code produces a title where the first word is red and the second word is blue:

```
title color=red "Red" color=blue " Blue";
```

Alias: C=

Default: For titles, the default text color is specified by the COLOR attribute of the GraphTitleText style element in the current style.

For footnotes, the default font color is specified by the COLOR attribute of the GraphFootnoteText style element in the current style.

FONT= “*system-font*”

specifies a system font for the text string.

Alias: F=

Default: For titles, the default font is specified by the FONTFAMILY attribute of the GraphTitleText style element in the current style.

For footnotes, the default font is specified by the FONTFAMILY attribute of the GraphFootnoteText style element in the current style.

Note: SAS/GRAPH software fonts such as SWISS cannot be used with statistical graphics procedures.

HEIGHT= *numeric-value* *<units>*

specifies the size of the text. You can also specify the unit of measurement. The following table lists the measurement units that are supported:

You can also specify the unit of measure. See [“Measurement Units for TITLE and FOOTNOTE Statement Options” on page 63](#) for a list of the units that are supported.

If you do not specify a unit, then the size of the text is approximately 12*n* points. For example, if you specify HEIGHT=2, then the text size is approximately 24 points.

Alias: H=

Default: For titles, the default font size is specified by the FONTSIZE attribute of the GraphTitleText style element in the current style.

For footnotes, the default font size is specified by the FONTSIZE attribute of the GraphFootnoteText style element in the current style.

ITALIC

specifies that the font style is italic for the text string.

Default: For titles, the default font style is specified by the FONTSTYLE attribute of the GraphTitleText style element in the current style.

For footnotes, the default font style is specified by the FONTSTYLE attribute of the GraphFootnoteText style element in the current style.

JUSTIFY= LEFT | CENTER | RIGHT

specifies the alignment of the text string. You can specify one of the following values:

LEFT | L

aligns the text to the left.

CENTER | C

aligns the text in the center.

RIGHT | R

aligns the text to the right.

The JUSTIFY= option affects all of the text strings that follow it in your TITLE or FOOTNOTE statement. You can specify multiple alignments by using more than one JUSTIFY= option. For example, the following code creates a footnote where the first string is aligned to the left and the second string is aligned to the right:

```
footnote justify=left "Example 2" justify=right "Graph 3";
```

Alias: J=

LSPACE= *numeric-value* *<units>*

specifies the amount of space above the title text and below the footnote text.

You can also specify the unit of measure. See [“Measurement Units for TITLE and FOOTNOTE Statement Options” on page 63](#) for a list of the units that are supported.

If you do not specify a unit, then the size of the space is approximately $12n$ points. For example, if you specify `LSPACE=2`, then the space is approximately 24 points.

Alias: LS

Default: 0

Interaction: The `LSPACE=` option has no effect if you also specify the `BOX=` option.

Details

Summary of Text Options

text-options can be one or more of the following:

- appearance options:
 - BOLD
 - COLOR= *color*
 - FONT= “*system-font*”
 - HEIGHT= *numeric-value* <*units*>
 - ITALIC
- placement and spacing options:
 - JUSTIFY= LEFT | CENTER | RIGHT
 - LSPACE= *numeric-value* <*units*>
- boxing and drawing options:
 - BCOLOR= *color*
 - BOX= *numeric-value*
 - BSPACE= *numeric-value* <*units*>

The following options are not supported by statistical graphics procedures:

- ANGLE=
- BLANK=
- DRAW=
- LANGLE=
- LINK=
- MOVE=
- ROTATE=
- UNDERLIN=
- WRAP

Using TITLE and FOOTNOTE Statements

You can define TITLE and FOOTNOTE statements anywhere in your SAS program. They are global and remain in effect until you cancel them or until you end your SAS session. All currently defined FOOTNOTE and TITLE statements are displayed automatically.

You can define up to ten TITLE statements and ten FOOTNOTE statements in your SAS session. A TITLE or FOOTNOTE statement without a number is equivalent to a

TITLE1 or FOOTNOTE1 statement. It is not necessary to use sequential statement numbers—skipping a number in the sequence leaves a blank line.

You can use an unlimited number of text strings and options. Ensure that each option is placed before the text strings that you want it to modify.

The most recently specified TITLE or FOOTNOTE statement of any number completely replaces any other TITLE or FOOTNOTE statement of that number. In addition, it cancels all TITLE or FOOTNOTE statements of a higher number. For example, if you define TITLE1, TITLE2, and TITLE3, then submitting a new TITLE2 statement cancels TITLE3.

The most recently specified TITLE or FOOTNOTE statement of any number completely replaces any other TITLE or FOOTNOTE statement of that number. In addition, it cancels all TITLE or FOOTNOTE statements of a higher number. For example, if you define TITLE1, TITLE2, and TITLE3, resubmitting the TITLE2 statement cancels TITLE3.

```
title4;
```

But remember that this cancels all other existing statements of a higher number.

To cancel all current TITLE or FOOTNOTE statements, use the **TITLE1;** or **FOOTNOTE1;** statement:

Substituting BY Line Values in a Text String

These options are available if a BY statement is in effect:

#BYLINE

substitutes the entire BY line without leading or trailing blanks for #BYLINE in the text string. The BY line uses the format *variable-name=value*.

#BYVAL*n* | #BYVAL(*BY-variable-name*)

substitutes the current value of the specified BY variable for #BYVAL in the text string. Specify the variable with one of these:

n

specifies a variable by its position in the BY statement. For example, #BYVAL2 specifies the second variable in the BY statement.

BY-variable-name

specifies a variable from the BY statement by its name. For example, #BYVAL(YEAR) specifies the BY variable, YEAR. *variable-name* is not case sensitive.

#BYVAR*n* | #BYVAR(*BY-variable-name*)

substitutes the name of the BY-variable or the label associated with the variable (whatever the BY line would normally display) for #BYVAR in the text string. Specify the variable with one of these:

n

specifies a variable by its position in the BY statement. For example, #BYVAR2 specifies the second variable in the BY statement.

BY-variable-name

specifies a variable from the BY statement by its name. For example, #BYVAR(SITES) specifies the BY variable, SITES. *Variable-name* is not case sensitive.

Note: A BY variable name displayed in a title or footnote is always in uppercase. If a label is used, then it appears as specified in the LABEL statement.

To use the #BYVAR and #BYVAL substitutions, insert the item in the text string at the position where you want the substitution text to appear. Both #BYVAR and #BYVAL specifications must be followed by a delimiting character. The character can be either a space or other non-alphanumeric character, such as a quotation mark. If no delimiting character is provided, then the specification is ignored and its text remains intact and is displayed with the rest of the string. To allow a #BYVAR or #BYVAL substitution to be followed immediately by other text, with no delimiter, use a trailing dot (as with macro variables). The trailing dot is not displayed in the resolved text. If you want a period to be displayed as the last character in the resolved text, use two dots after the #BYVAR or #BYVAL substitution.

The substitution for #BYVAR or #BYVAL does not occur in the following cases:

- if you use a #BYVAR or #BYVAL specification for a variable that is not named in the BY statement. For example, you might use #BYVAL2 when there is only one BY-variable or #BYVAL(ABC) when ABC is non-existent or is not a BY-variable.
- if there is no BY statement at all

For both cases, no error or warning message is issued. The option specification is displayed with the rest of the string. The graph continues to display a BY line at the top of the page unless you suppress it by using the NOBYLINE option in an OPTION statement.

Measurement Units for TITLE and FOOTNOTE Statement Options

Some of the options in the TITLE and FOOTNOTE statements give you the option to specify the unit of measurement. The following table lists the units that are supported:

Table 4.4 Measurement Units

Unit	Description
CM	centimeters
IN	inches
PCT or %	percentage
PT	point size, calculated at 100 dots per inch

WHERE Statement

Selects observations from SAS data sets that meet a particular condition.

Used by: SGDESIGN, SGPANEL, SGPLOT, SGRENDER, and SGSCATTER procedures

Details

All features of the WHERE statement are supported. For more information, see “WHERE Statement” in *SAS Statements: Reference*.

Note: When the WHERE statement is used with the SGDESIGN procedure, you must specify the data set. This is required even though a data set has already been defined for the ODS Graphics Designer (SGD) file that is rendered by the procedure.

Part 2

SAS ODS Graphics Procedures

Chapter 5	
SGDESIGN Procedure	67
Chapter 6	
SGPANEL Procedure	77
Chapter 7	
SGPLOT Procedure	275
Chapter 8	
SGRENDER Procedure	513
Chapter 9	
SGSCATTER Procedure	523

Chapter 5

SGDESIGN Procedure

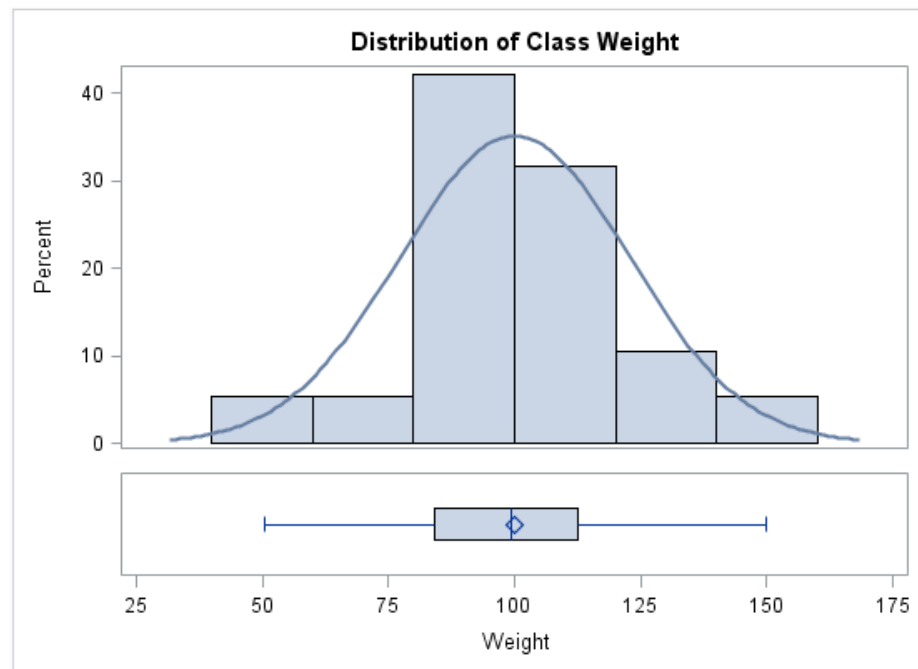
Overview: SGDESIGN Procedure	67
Concepts: SGDESIGN Procedure	68
General Concepts	68
Dynamic Variable Concepts	69
Syntax: SGDESIGN Procedure	70
PROC SGDESIGN Statement	70
DYNAMIC Statement	72
Examples: SGDESIGN Procedure	73
Example 1: Creating a Graph and Subsetting the Data	73
Example 2: Creating a Graph That Uses a Different Data Set	74
Example 3: Using the DYNAMIC Statement to Specify a Column and a Character String	74

Overview: SGDESIGN Procedure

The SGDESIGN procedure produces a graph from one or more input SAS data sets and a user-defined ODS Graphics Designer (SGD) file. The SGD file is created with the SAS ODS Graphics Designer application.

Here is example output:

Display 5.1 Example PROC SGDESIGN Output



In the example, the data columns and title text are provided dynamically with the DYNAMIC statement.

By default, the SGDESIGN procedure uses the data set or data sets that are currently defined in the SGD file. If the SGD file has been defined with dynamic variables, then you can substitute a different value for a variable by using the DYNAMIC statement. For more information, see [“Dynamic Variable Concepts” on page 69](#).

Note: The procedure applies the style of the active destination rather than the style that is currently defined in the SGD file.

For complete instructions on creating SGD files, see *SAS ODS Graphics Designer: User's Guide*.

Concepts: SGDESIGN Procedure

General Concepts

Here are the general concepts for the SGDESIGN procedure:

- The SGDESIGN procedure requires an SGD file that was created with the ODS Graphics Designer application.
- By default, the SGDESIGN procedure uses the data set or sets that are currently referenced by the SGD file. You have the option to specify an alternate data set.
- You can produce a graph from a different data set as long as the new data set has variables of the same name and type. For an example, see [“Example 2: Creating a Graph That Uses a Different Data Set” on page 74](#).

Note: If the SGD file was created using shared variables, then you can create a graph from a different data set regardless of the name of the variables. For more information, see [“Dynamic Variable Concepts” on page 69](#).

- You can render the graph to any ODS destination by using standard ODS syntax. When the graph is rendered, the procedure applies the style of the active destination rather than the style that was used in the SGD file.
- As with the other ODS Graphics procedures, the ODS GRAPHICS statement is always ON. However, you can use the ODS GRAPHICS statement options to control many aspects of your graphics. For more information, see [“Using the ODS GRAPHICS Statement” on page 651](#).
- The SGDESIGN procedure supports SAS statements such as FORMAT, LABEL, BY, and WHERE. These statements can be applied only if the DATA= option is used with the procedure.

For an example that uses the WHERE statement, see [“Example 1: Creating a Graph and Subsetting the Data” on page 73](#).

Dynamic Variable Concepts

A dynamic variable is a variable that is defined in a template with the DYNAMIC statement and that can be initialized at template run time. If the SGD file has been defined with dynamic variables, then you can set or initialize these variables by using the DYNAMIC statement with the SGDESIGN procedure.

For more information about dynamic variables, see *SAS Graph Template Language: Reference*.

Dynamic variables can be created in ODS Graphics Designer in two ways:

- shared variables

In ODS Graphics Designer, graphs can be defined to use shared variables. A shared variable is a feature of ODS Graphics Designer that enables users to reuse graphs and specify different variables from the same or from a different data set. The shared variable is a type of dynamic variable that can be assigned the name of a data column.

In ODS Graphics Designer, shared variables are named V1, V2, V3, and so on. Each shared variable corresponds to a column in the data set.

If the SGD file has been defined with a shared variable, you can substitute a different column for the shared variable when you run the SGDESIGN procedure. When you use the SGDESIGN procedure, the column that you substitute must be the same type as the column specified in the SGD file.

For example, suppose that your SGD file is a shared-variable graph in which V1 is assigned MPG_City, which is a numeric data type. When you run the SGDESIGN procedure, you can substitute MPG_Highway because it is also a numeric data type. In the procedure, you would specify **DYNAMIC V1="MPG_Highway"**.

The column that you substitute can reside in the same or in a different data set. (To use a different data set, you specify the data set by using the DATA= option.)

- dynamic content in text elements

In ODS Graphics Designer, you can insert dynamic content into textual elements such as titles and footnotes. This dynamic content is specified by using the expression `dyn(DNAME)` where *DNAME* is a name that you want to associate with

the text that is generated. You then substitute a character string or numeric constant when you run the graph using the SGDESIGN procedure.

For example, suppose the SGD file has the following title: *Weight in dyn(DWEIGHT)*. When you run the graph with the SGDESIGN procedure, you can specify **DYNAMIC DWEIGHT="Pounds"**. The entire string *dyn(DWEIGHT)* is replaced with the specified value. In the resulting graph, the title becomes *Weight in Pounds*.

Note:

- In ODS Graphics Designer, you can see the dynamic variables that have been defined for a graph when you view the graph's template code (from the **View** menu).
- For every graph that you create in ODS Graphics Designer, the designer defines dynamic variables automatically for every role assignment in the graph. In the template code, the names of these automatic variables begin with an underscore (for example, **dynamic _HEIGHT**).

CAUTION:

Substituting columns for these automatic dynamic variables can be complex and prone to error, and is not a supported usage of the SGDESIGN procedure. When using the SGDESIGN procedure, it is strongly recommended that you limit your use of dynamic variables to the shared variables and dynamic content that are described in this topic.

In summary, you can use dynamic variables to generate the same graph with different data variables, a different data set, and different text elements.

For more information about shared variables and dynamic content that are defined in the ODS Graphics Designer, see Chapter 19, “Using Shared Variables in Graphs,” in *SAS ODS Graphics Designer: User's Guide*.

Syntax: SGDESIGN Procedure

Global statements: BY, FORMAT, LABEL, ODS GRAPHICS, WHERE

```
PROC SGDESIGN SGD= "SGD-file-specification" < option(s)>;
DYNAMIC dynamic-var-1="assigned-value-1" <dynamic-var-n="assigned-value-n">;
```

PROC SGDESIGN Statement

Renders a graph (SGD file) that was created by using ODS Graphics Designer. The statement also enables you to specify a data set and other options.

Restrictions: On z/OS systems, the SGDESIGN procedure does not render SGD files that were generated with the previous release of the ODS Graphics Designer. You must open the SGD file in the 9.3 version of the ODS Graphics Designer (on a Windows or UNIX system). Then save the file in the 9.3 format.

On z/OS systems, SGD files must be transferred to the HFS file system of UNIX System Services in order to be rendered by the SGDESIGN procedure.

Requirement: An input SGD file specification is required. If the SGD file was created to use a temporary data set, such as a data set in the SAS Work library, then that data set must exist in the current SAS session in order for the graph to be rendered. The data set might need to be re-created in the current SAS session.

Syntax

PROC SGDESIGN SGD= *"SGD-file-specification"*

```
<CONTENT>
<DATA= input-data-set>
< DESCRIPTION= "text-string">
< LIBNAME= library-name>
```

;

Summary of Optional Arguments

CONTENT

outputs important information about the SGD file into the SAS log.

DATA= *input-data-set*

specifies the SAS data set that contains the variables to process.

DESCRIPTION= "*text-string*"

specifies a description for the output image.

LIBNAME= *library-name*

specifies an alternate library for all data sets that are associated with the SGD file.

Required Argument

SGD= *"SGD-file-specification"*

specifies the SGD file to use for the graph. Include the complete path and filename enclosed in single or double quotation marks.

If you do not specify the full path, then the procedure looks for the file in the SAS current working directory. The SAS current directory is the same directory in which you start your SAS session. If you are running SAS with the Windowing environment in the Windows operating system, then the current directory is displayed in the status bar at the bottom of the main SAS window.

For complete instructions on creating SGD files, see *SAS ODS Graphics Designer: User's Guide*.

Optional Arguments

CONTENT

outputs important information about the SGD file into the SAS log. The information includes data set names, library names, variable names (including those for dynamic variables), and the ODS style.

Note: When the CONTENT option is used, no graphics output is produced.

DATA= *input-data-set*

specifies the SAS data set that contains the variables to process.

By default, the SGDESIGN procedure uses the data set or data sets that are currently defined in the SGD file. You have the option to specify an alternate data set. Only one data set can be specified here. If the SGD file uses columns from multiple data

sets, then the specified data set should contain all the columns that are required to render the graph.

Specify the full name of the SAS data set in the format *libref.data-set*. If you do not specify the library, then the procedure looks for the data set in the SAS Work library.

DESCRIPTION= *“text-string”*

specifies a description for the output image.

The description identifies the image in the following locations:

- the Results window
- the alternate text for the image in HTML output
- the table of contents that is created by the CONTENTS option in an ODS statement

Alias: DES

Default: The default description is “The SGDesign Procedure”.

LIBNAME= *library-name*

specifies an alternate library for all data sets that are associated with the SGD file.

If no library is specified, then the procedure uses the library that is currently defined in the SGD file.

The data sets in the SGD file can come from multiple libraries. However, when you use the LIBNAME= option, you override all of those library names with the specified library.

DYNAMIC Statement

Assigns values to dynamic variables that have been defined in the SGD file. You can either specify variables in multiple DYNAMIC statements or specify all of your variables in a single DYNAMIC statement.

See: [“Example 3: Using the DYNAMIC Statement to Specify a Column and a Character String” on page 74](#)

Syntax

DYNAMIC *dynamic-var-1*=*“assigned-value-1”* <*dynamic-var-n*=*“assigned-value-n”*>;

Required Argument

dynamic-var= *assigned-value*

specifies the values for one or more dynamic variables. For more information about dynamic variables, see [“Dynamic Variable Concepts” on page 69](#).

dynamic-var

specifies the name of a dynamic variable.

The variable must be declared in a DYNAMIC statement within the SGD file's template. (In ODS Graphics Designer, you can see the dynamic variables that have been declared for a graph when you view the graph's template code from the **View** menu.)

assigned-value

specifies the value for the dynamic variable. If the value is a character string or a column name, then it must be enclosed in single or double quotation marks. For example: `V1="year"` and `MYTITLE="Distribution of Patient Weight"`.

If the value is a numeric value, then you should supply a number. In this case, no quotation marks are required. For example: `BINS=5`.

When the value refers to a column name, then the new column must be the same type as the column that it replaces. For example, if `V1` is a numeric type column, then the column name that you specify must also be numeric.

When the value is a column name, the case of the string value does not matter. For example, `V1="Weight"` is effectively the same as `V1="WEIGHT"`. When the value resolves to text that appears in the graph (for example, titles and axis labels) the value entered is case-sensitive.

Details

When you use dynamic variables, be aware that some variables must be initialized, whereas for others, initialization is optional. For example, suppose that the graph file `Histogram.sgd` is defined with dynamic content in the title. In order to execute this SGD file with the correct title, the dynamic variable used in the title must be initialized. (In the example, the dynamic variable is named `TITLE`.)

```
proc sgdesign sgd = "Histogram.sgd";
  dynamic TITLE = "This is a Histogram";
run;
```

However, if the SGD file was created with a shared variable, then that shared variable has already been assigned a column value in the SGD file. You do not need to specify a value for the variable unless you want to change the column. For more information about shared variables, see [“Dynamic Variable Concepts” on page 69](#).

If you have multiple dynamic variables, you can either specify variables in multiple `DYNAMIC` statements or specify all of your variables in a single `DYNAMIC` statement. For example, the following statement specifies two variables.

```
dynamic V1="cholesterol" EXAMPLESV="Cholesterol";
```

As an alternative, you can specify the dynamic variables in multiple `DYNAMIC` statements.

```
dynamic V1="cholesterol";
dynamic EXAMPLESV="Cholesterol";
```

Examples: SGDESIGN Procedure

Example 1: Creating a Graph and Subsetting the Data

Features: WHERE statement

This example uses a `WHERE` statement to subset the data. The example assumes the existence of a graph named `CarsLattice.sgd` that was created based on the `SASHELP.CARS` data set.

Although a data set has already been defined for the SGD file, you must specify the data set in the procedure when you use the WHERE statement.

```
proc sgdesign sgd="CarsLattice.sgd" 1
  data=sashelp.cars;
  where Origin="Asia";
run;
```

- 1 Specify the path and name that you used when you saved the SGD file. For example, the path might be "C:\SGDFiles\CarsLattice.sgd."

Example 2: Creating a Graph That Uses a Different Data Set

Features: subset data

You can produce a graph from a different data set as long as the new data set has variables of the same name and type. (If the SGD file was created using shared variables, then you can create a graph from a different data set regardless of the name of the variables. For more information, see [“Dynamic Variable Concepts” on page 69](#).)

The following example creates a new data set from an existing data set. This example assumes the existence of a graph named CarsLattice.sgd that was created based on the SASHELP.CARS data set.

```
data sedans;
  set sashelp.cars;
  where type="Sedan";
run;

proc sgdesign sgd="CarsLattice.sgd" 1 data=sedans;
run;
```

- 1 Specify the path and name that you used when you saved the SGD file. For example, the path might be "C:\SGDFiles\CarsLattice.sgd."

Example 3: Using the DYNAMIC Statement to Specify a Column and a Character String

Features: DYNAMIC statement

This example substitutes a column and also initializes the dynamic text for a title. The example assumes the following:

- the existence of a shared-variable graph named svExample.sgd that uses the SASHELP.CLASSFIT data set. To create this SGD file, follow the steps provided in “Example: Create a Shared-Variable Graph and Add a Dynamic Title” in Chapter 23 of *SAS ODS Graphics Designer: User's Guide*.
- in the SGD file, the shared variable V1 is set to WEIGHT.
- in ODS Graphics Designer, the title of the graph is *Distribution of dyn(EXAMPLESV)*.

Create the graph using the default data. You generate this graph only for comparison purposes.

```
proc sgdesign 1sgd="svExample.sgd";  
  2dynamic EXAMPLESV="Class Weight";  
run;
```

- 1 Specify the path and name that you used when you saved the SGD file. For example, the path might be "C:\SGDFiles\svExample.sgd."
- 2 Though this example uses the default value for the shared variable V1 (WEIGHT), there is no default value for the dyn(EXAMPLESV) function that is used in the title. To execute this SGD file with a correct title, the dynamic expression used in the title must be initialized, as shown in the code. In the output, the title becomes *Distribution of Class Weight*. The "Distribution of" portion of the title was defined in the SGD file. The "Class Weight" portion is generated dynamically.

Create the graph using different data. You can create the graph using a different data set and substitute a different column for V1. You can also change the dynamic variable that is used in the title. In the output, the title is *Distribution of Cholesterol*.

```
proc sgdesign 1sgd="svExample.sgd"  
  data=sashelp.heart;  
  dynamic 2V1="cholesterol" EXAMPLESV="Cholesterol";  
run;
```

- 1 Specify the path and name that you used when you saved the SGD file.
- 2 The CHOLESTEROL column is the same type as the WEIGHT column that was defined in the SGD file. The columns can be different as long as they are the same type (numeric in this case).

Chapter 6

SGPANEL Procedure

Overview: SGPANEL Procedure	78
Concepts: SGPANEL Procedure	79
Panel Creation	79
Plot Type Compatibility	81
Plot Axes	82
Panel Legends	82
Automatic Differentiation of Visual Attributes	83
Syntax: SGPANEL Procedure	83
PROC SGPANEL Statement	84
PANELBY Statement	87
BAND Statement	90
BUBBLE Statement	96
DENSITY Statement	101
DOT Statement	104
HBAR Statement	111
HBARPARM Statement	120
HBOX Statement	126
HIGHLOW Statement	137
HISTOGRAM Statement	145
HLINE Statement	148
KEYLEGEND Statement	157
LINEPARM Statement	161
LOESS Statement	165
NEEDLE Statement	171
PBSPLINE Statement	176
REFLINE Statement	183
REG Statement	185
SCATTER Statement	192
SERIES Statement	198
STEP Statement	204
VBAR Statement	212
VBARPARM Statement	220
VBOX Statement	227
VECTOR Statement	238
VLINE Statement	242
COLAXIS Statement	251
ROWAXIS Statement	259
Examples: SGPANEL Procedure	267
Example 1: Creating a Panel of Graph Cells with Histograms and Density Plots	267
Example 2: Creating a Panel of Regression Curves	269

Example 3: Creating a Panel of Bar Charts	270
Example 4: Creating a Panel of Line Charts	272

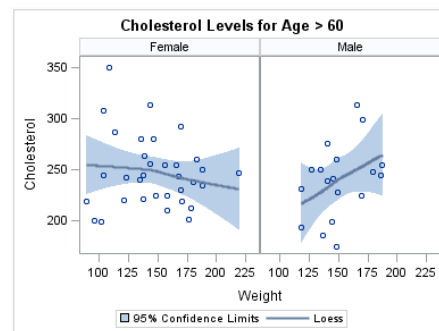
Overview: SGPANEL Procedure

The SGPANEL procedure creates a panel of graph cells for the values of one or more classification variables. For example, if a data set contains three variables (A, B and C) and you want to compare the scatter plots of B*C for each value of A, then you can use the SGPANEL to create this panel. The SGPANEL procedure creates a layout for you automatically and splits the panel into multiple graphs if necessary.

The SGPANEL procedure can create a wide variety of plot types, and overlay multiple plots together in each graph cell in the panel. It can also produce several types of layout.

Here are some examples of panels that the SGPANEL procedure can create.

Table 6.1 Examples of Panels That Can Be Generated by the SGPANEL Procedure



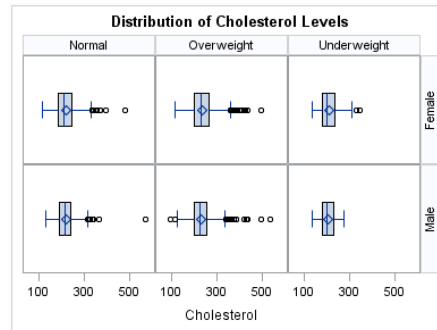
The following code creates a panel of loess curves:

```
title1 "Cholesterol Levels for Age > 60";
proc sgpanel data=sashelp.heart(
    where=(AgeAtStart > 60)) ;
    panelby sex / novarname;
    loess x=weight y=cholesterol / clm;
run;
title1;
```



The following code creates a panel of vertical bar charts:

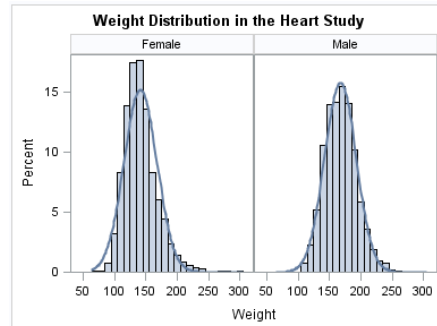
```
title1 "Product Sales";
proc sgpanel data=sashelp.prdsale;
    panelby quarter;
    rowaxis label="Sales";
    vbar product / response=predict stat=mean
        transparency=0.3;
    vbar product / response=actual stat=mean
        barwidth=0.5 transparency=0.3;
run;
title1;
```



The following code creates a panel of box plots in a lattice layout:

```
title1 "Distribution of Cholesterol Levels";
proc sgpanel data=sashelp.heart;
  panelby weight_status sex / layout=lattice
                                novarname;

  hbox cholesterol;
run;
title1;
```



The following code creates a panel of cells with a histogram and a normal density curve:

```
title1 "Weight Distribution in the Heart Study";
proc sgpanel data=sashelp.heart noautolegend;
  panelby sex / novarname;
  histogram weight;
  density weight;
run;
title1;
```

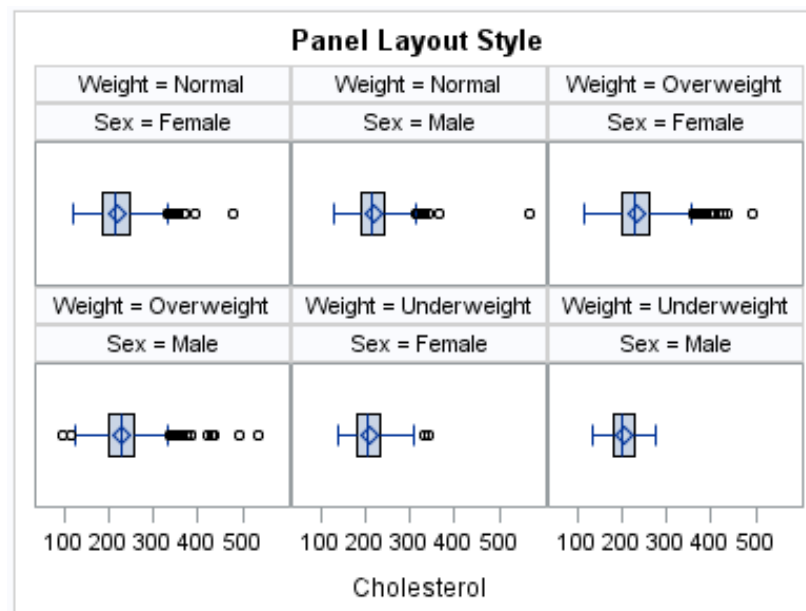
Concepts: SGPanel Procedure

Panel Creation

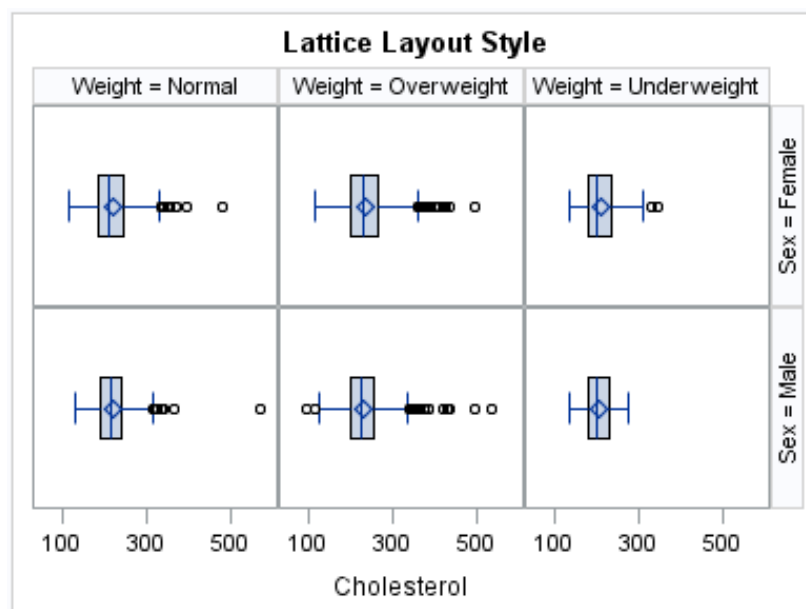
The SGPanel procedure has a required PANELBY statement that is used to define the classifier variables for the panel. This statement must be specified before any plot, axis, or legend statement or else an error occurs. You can use options in the PANELBY statement to control the attributes of the panel. For example, you can use the COLUMNS= option to specify the number of columns in the panel.

SGPanel can use four different layouts, which are specified by the LAYOUT= option in the PANELBY statement. The layout determines how your classifier variables are used to create the panel, and also affects the number of classifier variables that you can specify.

The default layout is PANEL. With this layout, you can specify any number of classifier variables. The graph cells in the panel are arranged automatically, and the classifier values are displayed above each graph cell in the panel. When you specify multiple classifier variables, the order of the classifier variables determines how the graph cells are sorted.

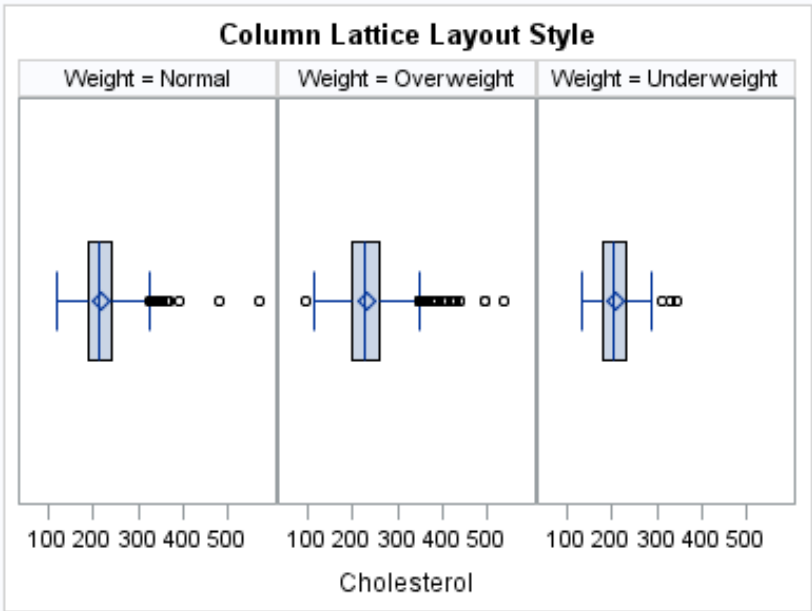
Figure 6.1 Example of the PANEL Layout

Another layout is called LATTICE. This layout requires exactly two classifier variables. The values of the first variable are assigned as columns, and the values of the second variable are assigned as rows. The classifier values are displayed above the columns and to the right side of the rows.

Figure 6.2 Example of the LATTICE Layout

Two additional layouts are available, which are called COLUMNLATTICE and ROWLATTICE. These layouts require exactly one classifier variable. The values of the classifier variable are assigned as cells in a single row or column.

Figure 6.3 Example of the COLUMNLATTICE Layout



If you have a large number of classifier variables, then the best method for creating a panel is to choose one or two classifiers for the PANELBY statement. Then specify the remaining variables in a BY statement. This method maximizes the space for the plots and generates results that are easier to interpret.

Plot Type Compatibility

Each graph cell in your panel contains one or more plots, and there are four basic types of plots that you can create with the SGPanel procedure.

- Basic plots
 - scatter, series, step, band, bubble, high-low, needle plots, and vector plots
- Fit and confidence plots
 - loess, regression, and penalized B-spline curves
- Distribution plots
 - box plots, histograms, normal density curves, and kernel density estimates
- Categorization plots
 - dot plots, bar charts, and line plots

Not all of the plot types can be used together in the same PROC SGPanel step. The following table shows which of the plot types can be used together:

Table 6.2 Plot Type Compatibility

	Basic	Fit and Confidence	Distribution	Categorization
Basic	x	x		
Fit and Confidence	x	x		

	Basic	Fit and Confidence	Distribution	Categorization
Distribution			x	
Categorization				x

Note: Box plots cannot be combined with any other plot types. Box plots can be overlaid with other box plots. However, overlaid box plots must have the same category variables.

If you submit a PROC SGPANEL step that combines two incompatible plot statements, then an error appears in the log.

The SGPANEL procedure draws the plots in your graph in the same order that you specify the plot statements. Because of this, it is important to consider the order of your plot statements so that your plots do not obscure one another. For example, if you specify a BAND statement after a SCATTER statement, then the band plot might obscure the markers in your scatter plot. You can also avoid obscuring your data by using the TRANSPARENCY= option to make your plots partially transparent.

Plot Axes

The SGPANEL procedure contains two statements that enable you to change the type and appearance for the axes of the graph cells in your panel: COLAXIS and ROWAXIS.

By default, the type of each axis is determined by the types of plots that use the axis and the data that is applied to the axis.

The SGPANEL procedure supports the following axis types:

Discrete

The axis contains independent data values rather than a range of numeric values. Each distinct value is represented by a tick mark. Discrete is the default axis type for character data.

Linear

The axis contains a linear range of numeric values. Linear is the default axis type for numeric data.

Logarithmic

The axis contains a logarithmic range of values. The logarithmic axis type is not used as a default.

Time

The axis contains a range of time values. Time is the default axis type for data that uses a SAS time, date, or datetime format.

Some types of plot do not support all of the axis types. For example, needle plots cannot use a discrete vertical axis. See the documentation for each plot statement to determine whether any axis type restrictions apply.

Panel Legends

The SGPANEL procedure creates a legend automatically based on the plot statements and options that you specify. The automatic legend functionality determines which information is likely to be useful in the legend. You can override this behavior by

defining your own legend with the KEYLEGEND statement or by specifying the NOAUTOLEGEND option in the PROC SGPANEL statement.

You can specify the labels that represent your plots in the legend by using the LEGENDLABEL= option in the corresponding plot statements.

You can create customized legends by using one or more KEYLEGEND statements. You can specify which plot statements are assigned to the legend, and use options to control the title, location, and border of the legend. For more information, see [“KEYLEGEND Statement” on page 157](#).

Automatic Differentiation of Visual Attributes

Depending on the plots and options that you specify, the SGPANEL procedure can automatically assign different style attributes to the plots in your graph. For example, if you specify two series plots, then each series plot automatically uses a different line pattern and line color by default. If different attributes are not assigned by default, you can force the procedure to assign different style attributes by using the CYCLEATTRS option in the PROC SGPANEL statement. You can also disable automatic attribute differentiation by using the NOCYCLEATTRS option in the PROC SGPANEL statement.

Syntax: SGPANEL Procedure

Requirement:	The PANELBY statement and at least one plot statement are required.
Global statements:	BY, FORMAT, LABEL, ODS GRAPHICS, TITLE and FOOTNOTE, WHERE

```

PROC SGPPANEL < option(s)>;
  PANELBY variable(s) < /option(s)>;
  BAND X= variable | Y= variable
  UPPER= numeric-value | numeric-variable LOWER= numeric-value | numeric-variable
  < /option(s)>;
  BUBBLE X= variable Y= variable SIZE= numeric-variable < / option(s)>;
  DENSITY response-variable < /option(s)>;
  DOT category-variable < /option(s)>;
  HBAR category-variable < /option(s)>;
  HBARPARM CATEGORY= category-variable RESPONSE= numeric-variable < / option(s)>;
  HBOX analysis-variable < /option(s)>;
  HIGHLOW X= variable | Y= variable
  HIGH= numeric-variable LOW= numeric-variable
  < /option(s)>;
  HISTOGRAM response-variable < /option(s)>;
  HLINE category-variable < /option(s)>;
  KEYLEGEND < "name(s)" > < /option(s)>;
  LINEPARM X= numeric-value |numeric-variable
  Y= numeric-value |numeric-variable
  SLOPE= numeric-value |numeric-variable
  < /option(s)>;
  LOESS X= numeric-variable Y= numeric-variable < /option(s)>;
  NEEDLE X= variable Y= numeric-variable < /option(s)>;
  PBSPLINE X= numeric-variable Y= numeric-variable < /option(s)>;
  REFLINE value(s) < /option(s)>;
  REG X= numeric-variable Y= numeric-variable < /option(s)>;
  SCATTER X= variable Y= variable < /option(s)>;
  SERIES X= variable Y= variable < /option(s)>;
  STEP X= variable Y= variable < /option(s)>;
  VBAR category-variable < /option(s)>;
  VBARPARM CATEGORY= category-variable RESPONSE= numeric-variable < / option(s)>;
  VBOX analysis-variable < /option(s)>;
  VECTOR X= numeric-variable Y= numeric-variable < /option(s)>;
  VLINE category-variable < /option(s)>;
  COLAXIS < /option(s)>;
  ROWAXIS < /option(s)>;

```

PROC SGPPANEL Statement

Identifies the data set that contains the plot variables. The statement also gives you the option to specify a description, and control automatic legends and automatic attributes.

Requirement: An input data set is required.

Syntax

```
PROC SGPANEL <DATA= input-data-set>
  <CYCLEATTRS | NOCYCLEATTRS>
  < DATTRMAP= "attribute-map-data-set" >
  < DESCRIPTION= "text-string" >
  <NOAUTOLEGEND>
  < PAD= dimension<units> | (pad-options)>
  < SGANNO= "annotation-data-set" >
;
```

Summary of Optional Arguments

CYCLEATTRS | NOCYCLEATTRS

specifies whether plots are drawn with unique attributes in the graph.

DATA=*input-data-set*

specifies the SAS data set that contains the variables to process.

DATTRMAP= *attribute-map-data-set*

specifies the SG attribute map data set that you want to use with the SGSCATTER procedure.

DESCRIPTION= "*text-string*"

specifies a description for the output image.

NOAUTOLEGEND

disables automatic legends from being generated.

PAD= *dimension*<units> | (*pad-options*)

specifies the amount of extra space that is reserved inside the border of an annotated graph.

SGANNO= *annotation-data-set*

specifies the SG annotation data set that you want to use.

Optional Arguments

CYCLEATTRS | NOCYCLEATTRS

specifies whether plots are drawn with unique attributes in the graph. By default, the SGPANEL procedure automatically assigns unique attributes in many situations, depending on the types of plots that you specify. If the plots do not have unique attributes by default, then the CYCLEATTRS option assigns unique attributes to each plot in the graph. The NOCYCLEATTRS option prevents the procedure from assigning unique attributes.

For example, if you specify the CYCLEATTRS option and you create a graph with a SERIES statement and a SCATTER statement, then the two plots have different colors.

If you specify the NOCYCLEATTRS option, then plots have the same attributes unless you specify appearance options such as the LINEATTRS= option.

DATA=*input-data-set*

specifies the SAS data set that contains the variables to process. By default, the procedure uses the most recently created SAS data set.

DATTRMAP= *attribute-map-data-set*

specifies the SG attribute map data set that you want to use with the SGSCATTER procedure. You specify this option only if you are using an attribute map to control

visual attributes of the graph. For more information, see [“Using SG Attribute Maps to Control Visual Attributes” on page 605](#).

DESCRIPTION= “text-string”

specifies a description for the output image. The description identifies the image in the following locations:

- the **Results** window
- the alternate text for the image in HTML output
- the table of contents that is created by the CONTENTS option in an ODS statement

The default description is “The SGPANEL Procedure”.

Alias: DES

Note: The name of the output image is specified by the IMAGENAME= option in the ODS GRAPHICS statement.

Tip: You can disable the alternate text in HTML output by specifying an empty string. That is, **DESCRIPTION= “ ”**.

NOAUTOLEGEND

disables automatic legends from being generated. By default, legends are created automatically for some plots, depending on their content. This option has no effect if you specify a KEYLEGEND statement.

PAD= dimension<units> | (pad-options)

specifies the amount of extra space that is reserved inside the border of an annotated graph.

You specify this option only if you are using the SG annotation feature to annotate your graph. For more information, see [Chapter 10, “Annotating ODS Graphics,” on page 557](#).

This option creates margins around the graph for company logos, annotated notes, and so on. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see [“Units of Measurement” on page 659](#).

Use pad options to create non-uniform padding. Edges that are not assigned padding are padded with the default amount.

pad-options can be one or more of the following:

LEFT= *dimension<units>*

specifies the amount of extra space to add to the left edge.

RIGHT= *dimension<units>*

specifies the amount of extra space to add to the right edge.

TOP= *dimension<units>*

specifies the amount of extra space to add to the top edge.

BOTTOM= *dimension<units>*

specifies the amount of extra space to add to the bottom edge.

SGANNO= annotation-data-set

specifies the SG annotation data set that you want to use. You specify this option only if you are using the SG annotation feature to annotate your graph. For more information, see [Chapter 10, “Annotating ODS Graphics,” on page 557](#).

PANELBY Statement

Specifies one or more classification variables for the panel, the layout type, and other options for the panel.

Syntax

PANELBY *variable(s)* </ *option(s)*>;

Summary of Optional Arguments

BORDER | NOBORDER

specifies whether borders are displayed around each cell in the panel.

COLHEADERPOS= TOP | BOTTOM | BOTH

specifies the location of the column headings in the panel.

COLUMNS= *n*

specifies the number of columns in the panel.

LAYOUT= LATTICE | PANEL | COLUMNLATTICE | ROWLATTICE

specifies the type of layout that is used for the panel.

MISSING

processes missing values as a valid classification value and creates cells for it.

NOVARNAME

removes the variable names from the cell headings of a panel layout, or from the row and column headings of a lattice layout.

ONEPANEL

places the entire panel in a single output image.

ROWHEADERPOS= LEFT | RIGHT | BOTH

specifies the location of the row headings in the panel.

ROWS= *n*

specifies the number of rows in the panel.

SPACING= *n*

specifies the number of pixels between the rows and columns in the panel.

SPARSE

enables the SGPPANEL procedure to create empty cells for crossings of the classification variables that are not present in the input data set.

START= TOPLEFT | BOTTOMLEFT

specifies whether the first cell in the panel is placed at the upper left corner or the lower left corner.

UNISCALE= COLUMN | ROW | ALL

scales the shared axes in the panel to be identical.

Required Argument

variable(s)

specifies one or more classification variables for the panel.

Optional Arguments**BORDER | NOBORDER**

specifies whether borders are displayed around each cell in the panel. BORDER adds the borders. NOBORDER removes the borders. Depending on the current ODS style, the borders might be present by default.

COLHEADERPOS= TOP | BOTTOM | BOTH

specifies the location of the column headings in the panel. Specify one of the following values:

TOP

places column headings at the top of each column.

BOTTOM

places column headings at the bottom of each column.

BOTH

places column headings at the top and bottom of each column.

Default: TOP

Interaction: This option has no effect if the panel uses the PANEL layout.

COLUMNS= *n*

specifies the number of columns in the panel. By default, the number of columns is determined automatically based on the number of classifier values and the layout type.

Tip: The SGPPANEL procedure automatically splits the panel into multiple graphs (pages) as needed when your panel contains a large number of cells. You can control the number of cells in each graph by using the COLUMNS= and the ROWS= options.

LAYOUT= LATTICE | PANEL | COLUMNLATTICE | ROWLATTICE

specifies the type of layout that is used for the panel. Select one of the following values:

LATTICE

when you specify two classification variables, arranges the cells so that the values of the first variable are columns and the values of the second variable are rows. You can use LATTICE only when you specify exactly two classification variables.

PANEL

arranges the cells in rows and columns. The headings for each cell are placed at the top of the cell.

COLUMNLATTICE

arranges the cells in a single row. You can use the COLUMNLATTICE layout only with a single classification variable.

ROWLATTICE

arranges the cells in a single column. You can use the ROWLATTICE layout only with a single classification variable.

Default: PANEL**MISSING**

processes missing values as a valid classification value and creates cells for it. By default, missing values are not processed as a classification value.

NOVARNAME

removes the variable names from the cell headings of a panel layout, or from the row and column headings of a lattice layout. For example, a row heading might

“NorthEast” instead of “Region=NorthEast” when you specify the NOVARNAM option.

ONEPANEL

places the entire panel in a single output image. If you do not specify this option, then the panel is automatically split into multiple images as appropriate.

Interactions:

When you use ONEPANEL with the PANEL layout, only one of the ROWS= and COLUMNS= options can be used. If you specify both options, then the value for COLUMNS= is used.

When you use ONEPANEL with the LATTICE layout, the ROWS= and COLUMNS= options have no effect.

Note: This option is recommended only for panels with a small number of cells. If your panel is too large for the output image, then a blank image is created.

ROWHEADERPOS= LEFT | RIGHT | BOTH

specifies the location of the row headings in the panel. Specify one of the following values:

LEFT

places row headings at the left side of each row.

RIGHT

places row headings at the right side of each row.

BOTH

places row headings at the left side and the right side of each row.

Default: RIGHT

Interaction: This option has no effect if the panel uses the PANEL layout.

ROWS= *n*

specifies the number of rows in the panel. By default, the number of rows is determined automatically based on the number of classifier values and the layout type.

Tip: The SGPPANEL procedure automatically splits the panel into multiple graphs (pages) as needed when your panel contains a large number of cells. You can control the number of cells in each graph by using the COLUMNS= and the ROWS= options.

SPACING= *n*

specifies the number of pixels between the rows and columns in the panel.

Default: 0

SPARSE

enables the SGPPANEL procedure to create empty cells for crossings of the classification variables that are not present in the input data set. By default, empty cells are not created for the panel layout.

Interaction: This option has no effect if you specify LAYOUT=LATTICE.

START= TOPLEFT | BOTTOMLEFT

specifies whether the first cell in the panel is placed at the upper left corner or the lower left corner. Specify one of the following values:

TOPLEFT

places the cell for the first data crossing in the upper left corner.

Cells are placed from left to right, starting in the top row. Each additional row is placed below the previous row.

The following figure shows the placement of nine cells in a panel where START= TOPLEFT:

1	2	3
4	5	6
7	8	9

BOTTOMLEFT

places the cell for the first data crossing in the lower left corner. Cells are placed from left to right, starting in the bottom row. Each additional row is placed above the previous row.

The following figure shows the placement of nine cells in a panel where START=BOTTOMLEFT:

7	8	9
4	5	6
1	2	3

Default: TOPLEFT

UNISCALE= COLUMN | ROW | ALL

scales the shared axes in the panel to be identical. Specify one of the following values:

COLUMN

scales all of the column axes in the panel to be identical.

ROW

scales all of the row axes in the panel to be identical.

ALL

scales all of the column axes to be identical, and also scales all of the row axes to be identical.

Default: ALL

BAND Statement

Creates a band that highlights part of the plot.

Restriction: The axis that the UPPER and LOWER values are placed on cannot be a discrete axis. For example, if you specify a variable for Y, the plot cannot use a discrete horizontal axis.

Note: The input data should be sorted by the X or Y variable. If the data is not sorted, the graph might produce unpredictable results.

Example: [“About Band Plots” on page 20](#)

Syntax

BAND *X= variable* | *Y= variable*

UPPER= numeric-value | *numeric-variable* *LOWER= numeric-value* | *numeric-variable*

</option(s)>;

Summary of Optional Arguments

Band options

DISCRETEOFFSET= *numeric-value*

specifies an amount to offset all bands from discrete X or Y tick values.

FILL | **NOFILL**

specifies whether the area fill is visible.

FILLATTRS= *style-element* | (**COLOR=** *color*)

specifies the appearance of the area fill for the band.

LINEATTRS= *style-element* <(options)> | (options)

specifies the appearance of the lines in the plot.

MODELNAME= "*plot-name*"

specifies that the band should be interpolated in the same way as the specified plot.

NOEXTEND

when you specify numeric values for **UPPER=** and **LOWER=**, specifies that the band does not extend beyond the first and last data points in the plot.

NOMISSINGGROUP

specifies that missing values of the group variable are not included in the plot.

OUTLINE | **NOOUTLINE**

specifies whether the outlines of the band are visible.

TYPE= **SERIES** | **STEP**

specifies how the data points for the lower and upper band boundaries are connected.

Label options

CURVELABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you specify a curve label.

CURVELABELLOWER = "*text-string*"

adds a label to the lower edge of the band.

CURVELABELPOS= **MIN** | **MAX** | **START** | **END**

specifies the location of the curve label.

CURVELABELUPPER = "*text-string*"

adds a label to the upper edge of the band.

Plot options

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set.

GROUP= *variable*

specifies a variable that is used to group the data.

LEGENDLABEL= "*text-string*"

specifies a label that identifies the elements from the band plot in the legend.

NAME= "*text-string*"

specifies a name for the plot.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the area fill.

Required Arguments

X= *variable* | **Y=** *variable*

specifies a variable that is used to plot the band along the x or y axis.

LOWER= *numeric-value* | *numeric-variable*

specifies the lower value for the band. You can specify either a constant numeric value or a numeric variable.

UPPER= *numeric-value* | *numeric-variable*

specifies the upper value for the band. You can specify either a constant numeric value or a numeric variable.

Optional Arguments

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph. For more information, see [“Using SG Attribute Maps to Control Visual Attributes” on page 605](#).

CURVELABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you specify a curve label. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

Here is an example that specifies a style element:

CURVELABELATTRS=GraphTitleText

options can be one or more of the following:

COLOR= *color*

specifies the color of the line label characters. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphValueText style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

FAMILY= “*font-family*”

specifies the font family for the label characters. The SAS ODS styles use TrueType system fonts. For more information, see “TrueType Fonts Supplied by SAS” in Chapter 15 of *SAS Language Reference: Concepts*.

Default: The default font family is specified by the FontFamily attribute of the GraphValueText style element in the current style.

Restriction: You cannot specify a list of font family names.

SIZE= *n* <*units*>

specifies the font size of the label characters. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see [“Units of Measurement” on page 659](#).

Default: The default font size is specified by the `FontSize` attribute of the `GraphValueText` style element in the current style.

`STYLE= ITALIC | NORMAL`

specifies whether the label characters are italic or normal.

Default: The default font style is specified by the `FontStyle` attribute of the `GraphValueText` style element in the current style.

`WEIGHT= BOLD | NORMAL`

specifies whether the label characters are bold or normal.

Default: The default font weight is specified by the `FontWeight` attribute of the `GraphValueText` style element in the current style.

Here is an example expression:

```
CURVELABELATTRS=(Color=Green Family=Arial Size=8 Style=Italic Weight=Bold)
```

Default: The default style element is `GraphValueText`.

Interaction: This option has no effect unless `CURVELABELLOWER` or `CURVELABELUPPER` is also specified.

CURVELABELLOWER = “*text-string*”

adds a label to the lower edge of the band. Specify the label text.

CURVELABELPOS= MIN | MAX | START | END

specifies the location of the curve label. Specify one of the following values:

MIN

places the label at the part of the curve closest to the minimum X axis value.

MAX

places the label at the part of the curve closest to the maximum X axis value.

START

places the curve label at the first point on the curve.

END

places the curve label at the last point on the curve.

Default: END

Interaction: This option has no effect unless the `CURVELABEL` option is also specified.

CURVELABELUPPER = “*text-string*”

adds a label to the upper edge of the band. Specify the label text.

DISCRETEOFFSET= *numeric-value*

specifies an amount to offset all bands from discrete X or Y tick values. Specify a value from -0.5 (left offset) to +0.5 (right offset).

Default: 0.0 (no offset)

Requirement: This option is applicable only when the X or Y axis is discrete.

FILL | NOFILL

specifies whether the area fill is visible. The `FILL` option shows the area fill. The `NOFILL` option hides the area fill.

Default: The default status of the area fill is specified by the `DisplayOpts` style attribute of the `GraphBand` style element in the current style.

Interactions:

Specifying `FILL` also hides any visible outlines.

If `NOFILL` and `NOOUTLINE` are both specified, then both options are ignored.

FILLATTRS= *style-element* | (COLOR= *color*)

specifies the appearance of the area fill for the band. You can specify the color of the fill by using a style element or by using the COLOR= suboption. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphConfidence style element in the current style.

For grouped data, the default color is specified by the Color attribute of the GraphData1 ... GraphData*n* style elements in the current style.

Interaction: This option has no effect if you specify the NOFILL option.

GROUP= *variable*

specifies a variable that is used to group the data. A separate band is created for each unique value of the grouping variable.

LEGENDLABEL= “*text-string*”

specifies a label that identifies the elements from the band plot in the legend. By default, the label “Band” is used for ungrouped data, and the group values are used for grouped data.

Interaction: The LEGENDLABEL= option has no effect if you also specify the GROUP= option in the same plot statement.

LINEATTRS= *style-element* <(options)> | (options)

specifies the appearance of the lines in the plot. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphDataDefault style element in the current style.

For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see “Line Patterns” on page 663.

Default: For ungrouped data, the default line pattern is specified by the LineStyle attribute of the GraphDataDefault style element in the current style.

For grouped data, the default line pattern is specified by the LineStyle attribute of the GraphData1 ... GraphData*n* style elements in the current style.

THICKNESS= *n* <units>

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: For ungrouped data, the default line thickness is specified by the LineThickness attribute of the GraphDataDefault style element in the current style.

For grouped data, the default line thickness is specified by the LineThickness attribute of the GraphData1 ... GraphData*n* style elements in the current style.

MODELNAME= "plot-name"

specifies that the band should be interpolated in the same way as the specified plot. If you do not specify the MODELNAME option, then the band is interpolated in the same way as a series plot.

NAME= "text-string"

specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note: The name is case-sensitive. When you refer to the specified name in other statements, you must use the same exact capitalization.

NOEXTEND

when you specify numeric values for UPPER= and LOWER=, specifies that the band does not extend beyond the first and last data points in the plot. By default, the band extends to the edges of the plot area.

Interaction: This option has no effect if you do not specify numeric values for the UPPER= and LOWER= options.

NOMISSINGGROUP

specifies that missing values of the group variable are not included in the plot.

OUTLINE | NOOUTLINE

specifies whether the outlines of the band are visible. The OUTLINE option shows the outlines. The NOOUTLINE option hides the outlines.

Default: The default status of the band outlines is specified by the DisplayOpts attribute of the GraphBand style element in the current style.

Interactions:

Specifying OUTLINE also hides the fill color.

If NOOUTLINE and NOFILL are both specified, then both options are ignored.

TRANSPARENCY= numeric-value

specifies the degree of transparency for the area fill. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default: 0.0

TYPE= SERIES | STEP

specifies how the data points for the lower and upper band boundaries are connected. You can specify one of the following:

SERIES

the data points are connected directly using line segments, as in a series plot.

STEP

the data points are connected using a step function, as in a step plot.

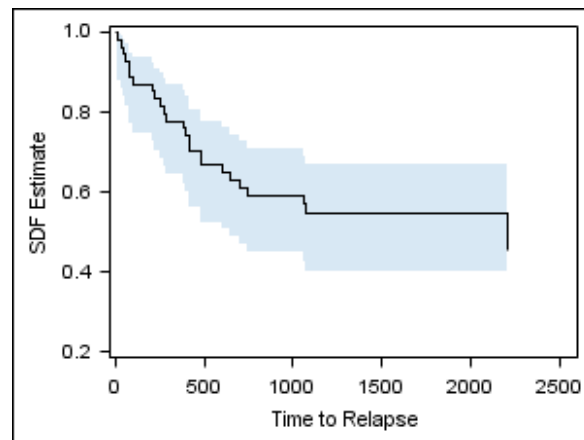
Default: SERIES

Details

The MODELNAME= option fits a band to another plot. This is particularly useful for plots that use a special interpolation such as step plots.

The following code fragment fits a band to a step plot:

```
band x=t upper=ucl lower=lcl / modelname="myname" transparency=.5;
step x=t y=survival / name="myname";
```

Figure 6.4 Fitted Band Plot Example

BUBBLE Statement

Creates a bubble plot in which two variables determine the location of the bubble centers and a third variable controls the size of the bubble.

Example: [“About Bubble Plots” on page 21](#)

Syntax

BUBBLE *X= variable Y= variable SIZE= numeric-variable* *</ option(s)>*;

Summary of Optional Arguments

Bubble options

BRADIUSMAX= *numeric-value*

specifies the size of the radius of the largest bubble.

BRADIUSMIN= *numeric-value*

specifies the size of the radius of the smallest bubble.

FILL | NOFILL

specifies whether the bubbles are filled.

FILLATTRS= *style-element* | (**COLOR=** *color*)

specifies the appearance of the fill for the bubbles.

LINEATTRS= *style-element* *<(options)>* | (*options*)

specifies the appearance of the outlines for the bubbles.

OUTLINE | NOOUTLINE

specifies whether the outlines of the bubbles are visible.

Group options

NOMISSINGGROUP

specifies that missing values of the group variable are not included in the plot.

Label options

DATALABEL *<= variable>*

displays a label for each data point.

DATALABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the DATALABEL= option.

Plot options

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set.

GROUP= *variable*

specifies a variable that is used to group the data.

LEGENDLABEL= *“text-string”*

specifies the label that identifies the bubble plot in the legend.

NAME= *“text-string”*

specifies a name for the plot.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the fill and outline of the bubbles.

URL= *character-variable*

specifies a character variable that contains URLs for Web pages to be displayed when parts of the plot are selected within an HTML page.

Required Arguments

X= *variable*

specifies the variable for the X axis.

Y= *variable*

specifies the variable for the Y axis.

SIZE= *numeric-variable*

specifies the variable that controls the size of the bubbles. The minimum and maximum values automatically provide the range that is used to determine bubble sizes. You can control this range manually by using the BRADIUSMAX and BRADIUSMIN options.

Optional Arguments

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph. For more information, see [“Using SG Attribute Maps to Control Visual Attributes” on page 605](#).

BRADIUSMAX= *numeric-value*

specifies the size of the radius of the largest bubble. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see [“Units of Measurement” on page 659](#).

Restriction: The BRADIUSMAX= value must be greater than the BRADIUSMIN= value if one is specified. If BRADIUSMAX is not greater, an error is generated and a message is written to the SAS log.

Note: If you specify the maximum size as a percentage, this is interpreted as a percentage of the graph's height.

BRADIUSMIN= *numeric-value*

specifies the size of the radius of the smallest bubble. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see [“Units of Measurement” on page 659](#).

Restriction: The BRADIUSMIN= value must be smaller than the BRADIUSMAX= value if one is specified. If BRADIUSMIN is not smaller, an error is generated and a message is written to the SAS log.

Note: If you specify the minimum size as a percentage, this is interpreted as a percentage of the graph's height.

DATALABEL <= *variable*>

displays a label for each data point. If you specify a variable, the values of that variable are used for the data labels. If you do not specify a variable, then the values of the Y variable are used for the data labels.

DATALABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the DATALABEL= option. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

Here is an example that specifies a style element:

DATALABELATTRS=GraphLabelText

options can be one or more of the following:

COLOR= *color*

specifies the color of the label characters. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphDataText style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

FAMILY= “*font-family*”

specifies the font family for the label characters. The SAS ODS styles use TrueType system fonts. For more information, see “TrueType Fonts Supplied by SAS” in Chapter 15 of *SAS Language Reference: Concepts*.

Default: The default font family is specified by the FontFamily attribute of the GraphDataText style element in the current style.

SIZE= *n* <*units*>

specifies the font size of the label characters. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default font size is specified by the FontSize attribute of the GraphDataText style element in the current style.

STYLE= ITALIC | NORMAL

specifies whether the label characters are italic (ITALIC) or normal (NORMAL).

Default: The default font style is specified by the FontStyle attribute of the GraphDataText style element in the current style.

WEIGHT= BOLD | NORMAL

specifies whether the label characters are bold (BOLD) or normal (NORMAL).

Default: The default font weight is specified by the FontWeight attribute of the GraphDataText style element in the current style.

Here is an example expression:

DATALABELATTRS=(Color=Green Family=Arial Size=8 Style=Italic Weight=Bold)

Default: The default style element is `GraphDataText`.

Interaction: This option has no effect unless the `DATALABEL` option is also specified.

FILL | NOFILL

specifies whether the bubbles are filled. The `FILL` option shows the fill color. The `NOFILL` option hides the fill color.

Default: `FILL`

Interactions:

Specifying `FILL` also hides the outlines.

If `NOFILL` and `NOOUTLINE` are both specified, then both options are ignored.

FILLATTRS= *style-element* | (COLOR= *color*)

specifies the appearance of the fill for the bubbles. You can specify the color of the fill by using a style element or by using the `COLOR=` suboption. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the `Color` attribute of the `GraphDataDefault` style element in the current style. For grouped data, the default color is specified by the `Color` attribute of the `GraphData1... GraphDatan` style elements in the current style.

Interaction: This option has no effect if you specify the `NOFILL` option.

GROUP= *variable*

specifies a variable that is used to group the data. The bubbles for each group value are automatically distinguished by different colors.

When this option is used, the bubble colors are taken from the `GraphData1... GraphDatan` style elements. If the bubbles are filled, then the `COLOR` attribute is used for bubble fill and `GRAPHOUTLINES` is used for the bubble outline. If the bubbles are not filled, then the `CONTRASTCOLOR` and `PATTERN` are used for the bubble outlines.

When this option is used and the value is a variable associated with an `ATTRID=` option, the attribute mapping defined by the associated attribute map is used.

LEGENDLABEL= “*text-string*”

specifies the label that identifies the bubble plot in the legend. By default, the label for the `Y` variable is used for ungrouped data, and the group values are used for grouped data.

Interaction: The `LEGENDLABEL=` option has no effect if you also specify the `GROUP=` option in the same plot statement.

LINEATTRS= *style-element* <(options)> | (options)

specifies the appearance of the outlines of the bubbles. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

Here is an example that specifies a style element: **LINEATTRS=GraphData3**

For more information about style elements, see “Style Elements for Use with ODS Graphics” on page 640.

options can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphDataDefault style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see [“Line Patterns” on page 663](#).

Default: For ungrouped data, the default line pattern is specified by the LineStyle attribute of the GraphDataDefault style element in the current style. For grouped data, the default line pattern is specified by the LineStyle attribute of the GraphData1 ... GraphData*n* style elements in the current style.

THICKNESS= *n* *<units>*

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see [“Units of Measurement” on page 659](#).

Default: For ungrouped data, the default line thickness is specified by the LineThickness attribute of the GraphDataDefault style element in the current style. For grouped data, the default line thickness is specified by the LineThickness attribute of the GraphData1 ... GraphData*n* style elements in the current style.

Here is an example expression:

```
LINEATTRS=(Color= "light green" Pattern= MediumDash Thickness= 4)
```

Default: For ungrouped data, line attributes are derived from the GraphDataDefault style element. For grouped data, the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements are used.

Interaction: This option has no effect if you also specify the NOOUTLINE option.

NAME= *“text-string”*

specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note: The name is case-sensitive. When you refer to the specified name in other statements, you must use the same exact capitalization.

NOMISSINGGROUP

specifies that missing values of the group variable are not included in the plot.

Interaction: This option has no effect unless GROUP= is also specified.

OUTLINE | NOOUTLINE

specifies whether the outlines of the bubbles are visible. The OUTLINE option shows the outlines. The NOOUTLINE option hides the outlines.

Default: OUTLINE

Interactions:

Specifying OUTLINE also hides the fill color.

If NOOUTLINE and NOFILL are both specified, then both options are ignored.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the fill and outline of the bubbles. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default: 0.0

URL= *character-variable*

specifies a character variable that contains URLs for Web pages to be displayed when parts of the plot are selected within an HTML page.

Default: By default, no HTML links are created.

Interactions:

This option affects graphics output that is created through the ODS HTML destination only. For more information about ODS destinations, see [“Specifying a Destination” on page 649](#).

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For more information, see [“Using the ODS GRAPHICS Statement” on page 651](#).

DENSITY Statement

Creates a density curve that shows the distribution of values in your data.

Interaction: The DENSITY statement can be combined only with the DENSITY and HISTOGRAM statements in the SGPanel procedure.

Examples: [“About Density Plots” on page 39](#)
[“Example 1: Creating a Panel of Graph Cells with Histograms and Density Plots” on page 267](#)

Syntax

DENSITY *response-variable* < / option(s)>;

Summary of Optional Arguments

DENSITY options

LINEATTRS= *style-element* <(options)> | (options)

specifies the appearance of the density line.

SCALE= COUNT | DENSITY | PERCENT | PROPORTION

specifies the scaling that is used for the response axis.

TYPE = NORMAL <(normal-opts)> | KERNEL <(kernel-opts)>

specifies the type of distribution curve that is used for the density plot.

Plot options

FREQ= *numeric-variable*

specifies how many times observations are repeated for computational purposes.

LEGENDLABEL= “*text-string*”

specifies a label that identifies the density plot in the legend.

NAME= “*text-string*”

specifies a name for the plot.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the density curve.

Required Argument

response-variable

specifies the variable for the x axis. The variable must be numeric.

Optional Arguments

FREQ= *numeric-variable*

specifies how many times observations are repeated for computational purposes. Each observation is repeated n times for computational purposes, where n is the value of the numeric variable. If n is not an integer, then it is truncated to an integer. If n is less than 1 or missing, then it is excluded from the analysis.

LEGENDLABEL= “*text-string*”

specifies a label that identifies the density plot in the legend. By default, the label identifies the type of density curve. If you specify TYPE=NORMAL, then the default label is “Normal.” If you specify TYPE=KERNEL, then the default label is “Kernel.”

Note: User-specified parameters from the TYPE= option are included in the label by default.

LINEATTRS= *style-element* <(options)> | (options)

specifies the appearance of the density line. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: The default color is specified by the ContrastColor attribute of the GraphFit style element in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see “Line Patterns” on page 663.

Default: The default line pattern is specified by the LineStyle attribute of the GraphFit style element in the current style.

THICKNESS= n <*units*>

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default line thickness is specified by the LineThickness attribute of the GraphFit style element in the current style.

NAME= “*text-string*”

specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note: The name is case-sensitive. When you refer to the specified name in other statements, you must use the same exact capitalization.

SCALE= COUNT | DENSITY | PERCENT | PROPORTION

specifies the scaling that is used for the response axis. Specify one of the following values:

COUNT

the axis displays the frequency count.

DENSITY

the axis displays the density estimate values.

PERCENT

the axis displays values as a percentage of the total.

PROPORTION

the axis displays values in proportion to the total.

Default: PERCENT

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the density curve. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default: 0.0

TYPE = NORMAL < (*normal-opts*)> | KERNEL < (*kernel-opts*)>

specifies the type of distribution curve that is used for the density plot. Specify one of the following keywords:

NORMAL < (*normal-opts*)>

specifies a normal density estimate, with a mean and a standard deviation.

normal-opts can be one or more of the following values:

MU= *numeric-value*

specifies the mean value that is used in the density function equation. By default, the mean value is calculated from the data.

SIGMA= *numeric-value*

specifies the standard deviation value that is used in the density function equation. The value that you specify for the SIGMA= suboption must be a positive number. By default, the standard deviation value is calculated from the data.

KERNEL < (*kernel-opts*)>

specifies a nonparametric kernel density estimate.

kernel-opts can be:

C= *numeric-value*

specifies the standardized bandwidth for a number that is greater than 0 and less than or equal to 100.

The value that you specify for the C= suboption affects the value of λ as shown in the following equation:

$$\lambda = cQn^{-\frac{1}{5}}$$

In this equation c is the standardized bandwidth, Q is the interquartile range, and n is the sample size.

WEIGHT= NORMAL | QUADRATIC | TRIANGULAR

specifies the weight function. You can specify either normal, quadratic, or triangular weight function.

Default: NORMAL

Default: NORMAL

Details

Normal Density Function

When the type of the density curve is NORMAL, the fitted density function equation is as follows.

$$p(x) = \frac{100h}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right) \text{ for } -\infty < x < \infty$$

In the equation, μ is the mean, and σ is the standard deviation. You can specify μ by using the MU= suboption and σ by using the SIGMA= suboption.

Kernel Density Function

When the TYPE of the density curve is KERNEL, 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, n is the sample size, and x_i is the i th observation. You can use the C= suboption to specify the bandwidth and the WEIGHT= suboption to specify the weight function $K_0(\cdot)$.

Kernel Density Weight Functions

The formulas for the weight functions are as follows.

NORMAL

$$K_0(t) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}t^2\right) \text{ for } -\infty < t < \infty$$

QUADRATIC

$$K_0(t) = \frac{3}{4}(1-t^2) \text{ for } |t| \leq 1$$

TRIANGULAR

$$K_0(t) = 1 - |t| \text{ for } |t| \leq 1$$

DOT Statement

Creates a dot plot that summarizes the values of a category variable.

Interaction: The DOT statement can be combined only with other horizontal categorization plot statements in the SGPPANEL procedure. See [“Plot Type Compatibility” on page 81](#).

Example: [“About Dot Plots” on page 46](#)

Syntax

DOT *category-variable* </option(s)>;

Summary of Optional Arguments

Dot options

DISCRETEOFFSET= *numeric-value*

specifies an amount to offset all dots from discrete category values.

LIMITATTRS= *style-element* <(options)> | (options)

specifies the appearance of the limit lines in the plot.

LIMITS= UPPER | LOWER | BOTH

specifies which limit lines to display.

LIMITSTAT= CLM | STDDEV | STDERR

specifies the statistic for the limit lines.

MARKERATTRS= *style-element* <(options)> | (options)

specifies the appearance of the markers in the plot.

MISSING

processes missing values as valid category value and creates a dot for it.

NOSTATLABEL

removes the statistic name from the axis and legend labels.

NUMSTD= *n*

specifies the number of standard units for the limit lines, when you specify LIMITSTAT= STDDEV or LIMITSTAT= STDERR.

RESPONSE= *response-variable*

specifies a numeric response variable for the plot.

STAT= FREQ | MEAN | SUM

specifies the statistic for the horizontal axis.

Group options

CLUSTERWIDTH= *numeric-value*

specifies the width of the group clusters as a fraction of the midpoint spacing.

GROUP= *variable*

specifies a classification variable to divide the values into groups.

GROUPDISPLAY= CLUSTER | OVERLAY

specifies how to display grouped dots.

GROUPORDER= ASCENDING | DESCENDING | DATA

specifies the ordering of dots within a group.

Label options

DATALABEL <= *variable*>

displays a label for each data point.

DATALABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the DATALABEL= option.

Plot options

ALPHA= *numeric-value*

specifies the confidence level for the confidence limits.

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set.

CATEGORYORDER= RESPASC | RESPDESC

specifies the order in which the response values are arranged.

FREQ= *numeric-variable*

specifies how many times observations are repeated for computational purposes.

LEGENDLABEL= "*text-string*"

specifies the label that identifies the dot plot in the legend.

NAME= "*text-string*"

specifies a name for the plot.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the plot.

URL= *character-variable*

specifies a character variable that contains URLs for Web pages to be displayed when parts of the plot are selected within an HTML page.

WEIGHT= *numeric-variable*

specifies how observations are weighted.

Required Argument

category-variable

specifies the variable that classifies the observations into distinct subsets.

Optional Arguments

ALPHA= *numeric-value*

specifies the confidence level for the confidence limits. Specify a number between 0.00 (100% confidence) and 1.00 (0% confidence).

Default: .05

Interactions:

This option has no effect if you do not specify LIMITSTAT=CLM.

If your plot is overlaid with other categorization plots, then the first ALPHA value that you specify is used for all of the plots.

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph. For more information, see [“Using SG Attribute Maps to Control Visual Attributes” on page 605](#).

CATEGORYORDER= RESPASC | RESPDESC

specifies the order in which the response values are arranged. Specify one of the following values:

RESPASC

sorts the response values in ascending order.

RESPDESC

sorts the response values in descending order.

Default: By default, the plot is sorted in ascending order based on the category values.

Restriction: This option takes effect only when the plot statement specifies a response variable and the axis for that variable is numeric. If the axis is not numeric, an error is generated and a message is written to the SAS log.

Requirement: This option requires that you configure the panel to use either one column or one row, depending on the orientation of your charts. Use the ROWS= or the COLUMNS= option in the PANELBY statement. If you do not use this option and your graph contains multiple cells, the specified sort order is not correctly applied to all cells.

Interaction: If a group variable is specified in the plot statement, the response sort option is ignored and a note is generated in the log. (You can use a BY statement in the procedure if you want to group data.)

Note: If CATEGORYORDER is specified in multiple statements, the procedure sorts by the last statement in which it is specified.

CLUSTERWIDTH= *numeric-value*

specifies the width of the group clusters as a fraction of the midpoint spacing. Specify a value from 0.0 (narrowest) to 1.0 (widest).

Default: 0.8

Interaction: This option is applicable only when the GROUP option is specified, when GROUPDISPLAY=CLUSTER, and when the category axis is discrete.

DATALABEL <= *variable*>

displays a label for each data point. If you specify a variable, then the values of that variable are used for the data labels. If you do not specify a variable, then the values of the calculated response are used for the data labels.

DATALABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the DATALABEL= option. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

Here is an example that specifies a style element:

DATALABELATTRS=GraphLabelText

options can be one or more of the following:

COLOR= *color*

specifies the color of the label characters. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphDataText style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

FAMILY= “*font-family*”

specifies the font family for the label characters. The SAS ODS styles use TrueType system fonts. For more information, see “TrueType Fonts Supplied by SAS” in Chapter 15 of *SAS Language Reference: Concepts*.

Default: The default font family is specified by the FontFamily attribute of the GraphDataText style element in the current style.

SIZE= *n* <*units*>

specifies the font size of the label characters. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default font size is specified by the FontSize attribute of the GraphDataText style element in the current style.

STYLE= ITALIC | NORMAL

specifies whether the label characters are italic (ITALIC) or normal (NORMAL).

Default: The default font style is specified by the FontStyle attribute of the GraphDataText style element in the current style.

WEIGHT= BOLD | NORMAL

specifies whether the label characters are bold (BOLD) or normal (NORMAL).

Default: The default font weight is specified by the FontWeight attribute of the GraphDataText style element in the current style.

Here is an example expression:

DATALABELATTRS=(Color=Green Family=Arial Size=8 Style=Italic Weight=Bold)

Default: The default style element is GraphDataText.

Interaction: This option has no effect unless the DATALABEL option is also specified.

DISCRETEOFFSET= *numeric-value*

specifies an amount to offset all dots from discrete category values. Specify a value from -0.5 (left offset) to +0.5 (right offset).

Default: 0.0 (no offset)

Requirement: This option is applicable only when the category axis is discrete.

FREQ= *numeric-variable*

specifies how many times observations are repeated for computational purposes. Each observation is repeated n times for computational purposes, where n is the value of the numeric variable. If n is not an integer, then it is truncated to an integer. If n is less than 1 or missing, then it is excluded from the analysis.

Interaction: If your plot is overlaid with other categorization plots, then the first FREQ variable that you specified is used for all of the plots.

GROUP= *variable*

specifies a classification variable to divide the values into groups. A separate plot is created for each unique value of the classification variable.

Interaction: If you specify more than one categorization plot statement, then all of the plots must specify the same GROUP variable. If you do not specify the same GROUP= option for all of the categorization plots, then the GROUP= option has no effect.

GROUPDISPLAY= CLUSTER | OVERLAY

specifies how to display grouped dots.

CLUSTER

grouped items are drawn adjacent to each other.

OVERLAY

grouped items are drawn at the exact coordinate and might overlap. Each group is represented by unique visual attributes derived from the GraphData1... GraphData n style elements in the current style.

Default: OVERLAY

Restriction: GROUPDISPLAY=CLUSTER can take effect only when the category axis is discrete. If necessary, use a TYPE=DISCRETE option on the axis statement.

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

GROUPORDER= ASCENDING | DESCENDING | DATA

specifies the ordering of dots within a group.

ASCENDING

displays each group in ascending group value.

DESCENDING

displays each group in descending group value.

DATA

displays each group in data order of the group variable.

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

LEGENDLABEL= “*text-string*”

specifies the label that identifies the dot plot in the legend. By default, the label of the RESPONSE= variable is used. If there is no response variable label, then the name of the response variable and the computed statistic (SUM or MEAN) is used. If the RESPONSE= option is not used, the legend label is “Frequency”.

Interaction: The LEGENDLABEL= option has no effect if you also specify the GROUP= option in the same plot statement.

LIMITATTRS= *style-element* <(options)> | (options)

specifies the appearance of the limit lines in the plot. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: The default color is specified by the ContrastColor attribute of the GraphError style element in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see “Line Patterns” on page 663.

Default: The default line pattern is specified by the LineStyle attribute of the GraphError style element in the current style.

THICKNESS= *n* <units>

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default line thickness is specified by the LineThickness attribute of the GraphError style element in the current style.

LIMITS= UPPER | LOWER | BOTH

specifies which limit lines to display. Limits are displayed as heavier line segments with a serif at the end extending horizontally from each dot. Upper limits extend to the right of the dot and lower limits extend to the left of the dot. By default, no limits are displayed unless you specify either the LIMITS= or LIMITSTAT= option. Specify one of the following values:

BOTH

adds lower and upper limit lines to the plot.

LOWER

adds lower limit lines to the plot.

UPPER

adds upper limit lines to the plot.

Interaction: Limit lines are displayed only when you specify STAT= MEAN.

LIMITSTAT= CLM | STDDEV | STDERR

specifies the statistic for the limit lines. Specify one of the following statistics:

CLM

confidence limits

STDDEV

standard deviation

STDERR

standard error

Default: CLM

Interaction: If you specify the LIMITSTAT= option only, then the default value for the LIMITS= option is BOTH. Limits lines are displayed only when you specify STAT=MEAN.

MARKERATTRS= *style-element* <(options)> | (options)

specifies the appearance of the markers in the plot. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the markers. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphDataDefault style element in the current style.

For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

SIZE= *n* <*units*>

specifies the size of the markers. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default marker size is specified by the MarkerSize attribute of the GraphDataDefault style element in the current style.

SYMBOL= *symbol-name*

specifies the symbol for the markers. For a list of valid marker symbols, see “Marker Symbols” on page 661.

Default: For ungrouped data, the default marker symbol is specified by the MarkerSymbol attribute of the GraphDataDefault style element in the current style.

For grouped data, the default marker symbol is specified by the MarkerSymbol attribute of the GraphData1 ... GraphData*n* style elements in the current style.

MISSING

processes missing values as valid category value and creates a dot for it.

NAME= “*text-string*”

specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note: The name is case-sensitive. When you refer to the specified name in other statements, you must use the same exact capitalization.

NOSTATLABEL

removes the statistic name from the axis and legend labels.

NUMSTD= *n*

specifies the number of standard units for the limit lines, when you specify LIMITSTAT= STDDEV or LIMITSTAT= STDERR. You can specify any positive number, including decimals.

Default: 1

RESPONSE= *response-variable*

specifies a numeric response variable for the plot. The summarized values of the response variable for each category value are displayed on the horizontal axis.

STAT= FREQ | MEAN | SUM

specifies the statistic for the horizontal axis. Specify one of the following statistics:

FREQ

the frequencies for the category variable. This is the default value when you do not specify the RESPONSE= option.

MEAN

the mean of the response variable.

SUM

the sum of the response variable. This is the default value when you specify the RESPONSE= option.

Interaction: If you do not specify the RESPONSE= option, then only the FREQ statistic can be used. If you specify the RESPONSE= option, then you can use either the SUM or MEAN statistics.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the plot. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default: 0.0

URL= *character-variable*

specifies a character variable that contains URLs for Web pages to be displayed when parts of the plot are selected within an HTML page.

Default: By default, no HTML links are created.

Interactions:

This option affects graphics output that is created through the ODS HTML destination only. For more information about ODS destinations, see [“Specifying a Destination” on page 649](#).

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For more information, see [“Using the ODS GRAPHICS Statement” on page 651](#).

WEIGHT= *numeric-variable*

specifies how observations are weighted. Each observation is weighted by a factor of w for computational purposes, where w is the value of the numeric variable. w can be any numeric value. If w is 0, negative or missing, then that observation is excluded from the analysis.

Interaction: If your plot is overlaid with other categorization plots, then the first WEIGHT variable that you specified is used for all of the plots.

HBAR Statement

Creates a bar chart that summarizes the values of a category variable.

Interaction: The HBAR statement can be combined only with other categorization plot statements in the SGPanel procedure. See [“Plot Type Compatibility” on page 81](#).

Examples: [“About Bar Charts” on page 42](#)
[“Example 3: Creating a Panel of Bar Charts” on page 270](#)

Syntax

HBAR *category-variable* </ option(s)>;

Summary of Optional Arguments

Bar options

BARWIDTH= *numeric-value*

specifies the width of the bars as a ratio of the maximum possible width.

DATASKIN= NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN

specifies a special effect to be used on all filled bars.

DISCRETEOFFSET= *numeric-value*

specifies an amount to offset all bars from the category midpoints.

FILL | **NOFILL**

specifies whether the bars are filled.

FILLATTRS= *style-element* | (COLOR= *color*)

specifies the appearance of the fill for the bars.

LIMITATTRS= *style-element* <(options)> | (options)

specifies the appearance of the limit lines in the plot.

LIMITS= BOTH | LOWER | UPPER

specifies which limit lines to display.

LIMITSTAT= CLM | STDDEV | STDERR

specifies the statistic for the limit lines.

MISSING

processes missing values as valid category value and creates a bar for it.

NOSTATLABEL

removes the statistic name from the axis and legend labels.

NUMSTD= *n*

specifies the number of standard units for the limit lines, when you specify **LIMITSTAT**= STDDEV or **LIMITSTAT**= STDERR.

OUTLINE | **NOOUTLINE**

specifies whether the bars have outlines.

RESPONSE= *response-variable*

specifies a numeric response variable for the plot.

STAT= FREQ | MEAN | SUM

specifies the statistic for the horizontal axis.

Group options

CLUSTERWIDTH= *numeric-value*

specifies the cluster width as a ratio of the maximum width.

GROUP= *variable*

specifies a variable that is used to group the data.

GROUPDISPLAY= STACK | CLUSTER

specifies how to display grouped bars.

GROUPORDER= ASCENDING | DESCENDING | DATA

specifies the ordering of bars within a group.

Label options

DATALABEL <= *variable*>

displays a label for each data point.

DATALABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the **DATALABEL**= option.

Plot options

ALPHA= *numeric-value*

specifies the confidence level for the confidence limits.

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set.

CATEGORYORDER= RESPASC | RESPDESC

specifies the order in which the response values are arranged.

FREQ= *numeric-variable*

specifies how many times observations are repeated for computational purposes.

LEGENDLABEL= "*text-string*"

specifies the label that identifies the bar chart in the legend.

NAME= "*text-string*"

specifies a name for the plot.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the bars and limits, if displayed.

URL= *character-variable*

specifies a character variable that contains URLs for Web pages to be displayed when parts of the plot are selected within an HTML page.

WEIGHT= *numeric-variable*

specifies how observations are weighted.

Required Argument

category-variable

specifies the variable that classifies the observations into distinct subsets.

Optional Arguments

ALPHA= *numeric-value*

specifies the confidence level for the confidence limits. Specify a number between 0.00 (100% confidence) and 1.00 (0% confidence).

Default: .05

Interactions:

This option has no effect if you do not specify LIMITSTAT=CLM.

If your plot is overlaid with other categorization plots, then the first ALPHA value that you specify is used for all of the plots.

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph. For more information, see [“Using SG Attribute Maps to Control Visual Attributes”](#) on page 605.

BARWIDTH= *numeric-value*

specifies the width of the bars as a ratio of the maximum possible width. The maximum width is equal to the distance between the center of each bar and the centers of the adjacent bars. Specify a value from 0.0 (narrowest) to 1.0 (widest).

For example, if you specify a width of 1, then there is no space between the bars. If you specify a width of .5, then the width of the bars is equal to the space between the bars.

Default: .8

1.0 when the GROUP option is specified and GROUPDISPLAY=CLUSTER

Interaction: When the GROUP option is specified, the bar width is determined by the maximum number of bars in any one group cluster. All bars are drawn with the same width, with the cluster being positioned symmetrically around the midpoint.

CATEGORYORDER= RESPASC | RESPDESC

specifies the order in which the response values are arranged. Specify one of the following values:

RESPASC

sorts the response values in ascending order.

RESPDESC

sorts the response values in descending order.

Default: By default, the plot is sorted in ascending order based on the category values.

Restriction: This option takes effect only when the plot statement specifies a response variable and the axis for that variable is numeric. If the axis is not numeric, an error is generated and a message is written to the SAS log.

Requirement: This option requires that you configure the panel to use either one column or one row, depending on the orientation of your charts. Use the ROWS= or the COLUMNS= option in the PANELBY statement. If you do not use this option and your graph contains multiple cells, the specified sort order is not correctly applied to all cells.

Interaction: If a group variable is specified in the plot statement, the response sort option is ignored and a note is generated in the log. (You can use a BY statement in the procedure if you want to group data.)

Note: If CATEGORYORDER is specified in multiple statements, the procedure sorts by the last statement in which it is specified.

CLUSTERWIDTH= *numeric-value*

specifies the cluster width as a ratio of the maximum width. Specify a value from 0.0 (narrowest) to 1.0 (widest).

CLUSTERWIDTH is the fraction of the midpoint spacing used by all bars that are clustered around a midpoint (category value). The bar width is applied to the maximum bar spacing divided by the maximum number of bars in any one cluster.

Default: 0.8

Interaction: This option is applicable only when the GROUP option is specified, when GROUPDISPLAY=CLUSTER, and when the category axis is discrete.

DATALABEL <= *variable*>

displays a label for each data point. If you specify a variable, then the values of that variable are used for the data labels. If you do not specify a variable, then the values of the calculated response are used for the data labels.

DATALABELATTRS= *style-element* <(*options*)> | (*options*)

specifies the appearance of the labels in the plot when you use the DATALABEL= option. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

Here is an example that specifies a style element:

DATALABELATTRS=GraphLabelText

options can be one or more of the following:

COLOR= *color*

specifies the color of the label characters. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphDataText style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

FAMILY= “*font-family*”

specifies the font family for the label characters. The SAS ODS styles use TrueType system fonts. For more information, see “TrueType Fonts Supplied by SAS” in Chapter 15 of *SAS Language Reference: Concepts*.

Default: The default font family is specified by the FontFamily attribute of the GraphDataText style element in the current style.

SIZE= *n* <*units*>

specifies the font size of the label characters. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default font size is specified by the FontSize attribute of the GraphDataText style element in the current style.

STYLE= ITALIC | NORMAL

specifies whether the label characters are italic (ITALIC) or normal (NORMAL).

Default: The default font style is specified by the FontStyle attribute of the GraphDataText style element in the current style.

WEIGHT= BOLD | NORMAL

specifies whether the label characters are bold (BOLD) or normal (NORMAL).

Default: The default font weight is specified by the FontWeight attribute of the GraphDataText style element in the current style.

Here is an example expression:

```
DATALABELATTRS=(Color=Green Family=Arial Size=8 Style=Italic Weight=Bold)
```

Default: The default style element is GraphDataText.

Interaction: This option has no effect unless the DATALABEL option is also specified.

DATASKIN= NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN

specifies a special effect to be used on all filled bars. Specify one of the following (the samples were generated using the HTMLBlue style):

NONE



CRISP



GLOSS



MATTE



PRESSED



SHEEN

**Default:** NONE**Interactions:**

This option has no effect if the applied style uses a fill pattern and FILL is disabled for the style. If the style has both FILL and FILLPATTERN enabled, data skins will work. For more information about fill patterns, see [“Using Fill Patterns to Distinguish Grouped Bar Charts” on page 636](#). For instructions about viewing the contents of style templates, see [“About Styles and Style Elements” on page 620](#).

This option has no effect if you also specify NOFILL.

DISCRETEOFFSET= *numeric-value*

specifies an amount to offset all bars from the category midpoints. Specify a value from -0.5 (left offset) to +0.5 (right offset).

Default: 0.0 (no offset)**FILL | NOFILL**

specifies whether the bars are filled. The FILL option shows the fill color for the bars. The NOFILL option hides the fill color for the bars.

Default: FILL**Interactions:**

Specifying FILL also hides the outlines.

If NOFILL and NOOUTLINE are both specified, then both options are ignored.

FILLATTRS= *style-element* | (COLOR= *color*)

specifies the appearance of the fill for the bars. You can specify the color of the fill by using a style element or by using the COLOR= suboption. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphDataDefault style element in the current style.

For grouped data, the default color is specified by the Color attribute of the GraphData1 ... GraphData*n* style elements in the current style.

Interaction: This option has no effect if you specify the NOFILL option.

FREQ= *numeric-variable*

specifies how many times observations are repeated for computational purposes. Each observation is repeated *n* times for computational purposes, where *n* is the value of the numeric variable. If *n* is not an integer, then it is truncated to an integer. If *n* is less than 1 or missing, then it is excluded from the analysis.

Interaction: If your plot is overlaid with other categorization plots, then the first FREQ variable that you specified is used for all of the plots.

GROUP= *variable*

specifies a variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes.

GROUPDISPLAY= STACK | CLUSTER

specifies how to display grouped bars.

STACK

groups are overlaid without any clustering. Each group is represented by unique visual attributes derived from the GraphData1... GraphData*n* style elements in the current style.

CLUSTER

displays group values as separate adjacent bars that replace the single category bar. Each set of group values is centered at the midpoint tick mark for the category.

Default: STACK

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

GROUPORDER= ASCENDING | DESCENDING | DATA

specifies the ordering of bars within a group.

ASCENDING

displays each group in ascending group value.

DESCENDING

displays each group in descending group value.

DATA

displays each group in data order of the group variable.

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

LEGENDLABEL= “*text-string*”

specifies the label that identifies the bar chart in the legend. By default, the label of the RESPONSE= variable is used. If there is no response variable label, the name of the response variable and the computed statistic (SUM or MEAN) is used. If the RESPONSE= option is not used, the legend label is “Frequency”.

Interaction: The LEGENDLABEL= option has no effect if you also specify the GROUP= option in the same plot statement.

LIMITATTRS= *style-element* <(options)> | (options)

specifies the appearance of the limit lines in the plot. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: The default color is specified by the ContrastColor attribute of the GraphError style element in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see “[Line Patterns](#)” on page 663.

Default: The default line pattern is specified by the LineStyle attribute of the GraphError style element in the current style.

THICKNESS= *n* <*units*>

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default line thickness is specified by the LineThickness attribute of the GraphError style element in the current style.

Interaction: This option has no effect unless you also specify either the LIMITS= or LIMITSTAT= option.

LIMITS= BOTH | LOWER | UPPER

specifies which limit lines to display. Limits are displayed as heavier line segments with a serif at the end extending from each bar. Upper limits extend to the right of the bar and lower limits extend to the left of the bar. By default, no limits are displayed unless you specify either the LIMITS= or LIMITSTAT= option. If you specify the LIMITSTAT= option only, then LIMITS=BOTH is the default. Specify one of the following values:

BOTH

adds lower and upper limit lines to the plot.

LOWER

adds lower limit lines to the plot.

UPPER

adds upper limit lines to the plot.

Default: By default, no limit lines are displayed. However, if you specify the LIMITSTAT= option, then the default is BOTH.

Interactions:

Limit lines are displayed only when you specify STAT= MEAN.

The LIMITS= option has no effect if you also specify the GROUP= option in the plot statement.

LIMITSTAT= CLM | STDDEV | STDERR

specifies the statistic for the limit lines. Specify one of the following statistics:

CLM

confidence limits

STDDEV

standard deviation

STDERR

standard error

Default: CLM

Interactions:

If you specify the LIMITSTAT= option only, then the default value for the LIMITS= option is BOTH.

Limits lines are displayed only when you specify STAT=MEAN.

The LIMITSTAT= option has no effect if you also specify the GROUP= option in the plot statement.

MISSING

processes missing values as valid category value and creates a bar for it.

NAME= *“text-string”*

specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note: The name is case-sensitive. When you refer to the specified name in other statements, you must use the same exact capitalization.

NOSTATLABEL

removes the statistic name from the axis and legend labels.

NUMSTD= *n*

specifies the number of standard units for the limit lines, when you specify LIMITSTAT= STDDEV or LIMITSTAT= STDERR. You can specify any positive number, including decimals.

Default: 1

OUTLINE | NOOUTLINE

specifies whether the bars have outlines. The OUTLINE option shows the outlines. The NOOUTLINE option hides the outlines.

Default: OUTLINE

Interactions:

Specifying OUTLINE also hides the fill color.

If NOOUTLINE and NOFILL are both specified, then both options are ignored.

RESPONSE= *response-variable*

specifies a numeric response variable for the plot. The summarized values of the response variable are displayed on the horizontal axis.

STAT= **FREQ | MEAN | SUM**

specifies the statistic for the horizontal axis. Specify one of the following statistics:

FREQ

the frequencies for the category variable. This is the default value when you do not specify the RESPONSE= option.

MEAN

the mean of the response variable.

SUM

the sum of the response variable. This is the default value when you specify the RESPONSE= option.

Interaction: If you do not specify the RESPONSE= option, then only the FREQ statistic can be used. If you specify the RESPONSE= option, then you can use either the SUM or MEAN statistics.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the bars and limits, if displayed. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default: 0.0

URL= *character-variable*

specifies a character variable that contains URLs for Web pages to be displayed when parts of the plot are selected within an HTML page.

Default: By default, no HTML links are created.

Interactions:

This option affects graphics output that is created through the ODS HTML destination only. For more information about ODS destinations, see [“Specifying a Destination” on page 649](#).

This option has no effect unless you also specify `IMAGEMAP` in the ODS `GRAPHICS` statement. For more information, see [“Using the ODS GRAPHICS Statement” on page 651](#).

WEIGHT= *numeric-variable*

specifies how observations are weighted. Each observation is weighted by a factor of w for computational purposes, where w is the value of the numeric variable. w can be any numeric value. If w is 0, negative or missing, then that observation is excluded from the analysis.

Interaction: If your plot is overlaid with other categorization plots, then the first `WEIGHT` variable that you specified is used for all of the plots.

HBARPARM Statement

Creates a horizontal bar chart based on a pre-summarized response value for each unique value of the category variable. You can also assign variables to the upper and lower limits.

Requirement: The data must contain only one response value per unique category variable. If more than one value is found, a warning is written to the SAS log, and the graph might produce unpredictable results.

Interactions: The `HBARPARM` statement can be combined only with other basic plot statements in the `SGPLOT` procedure. See [“Plot Type Compatibility” on page 277](#).
When used with particular styles, the `HBARPARM` statement produces fill patterns for grouped bars. For more information about fill patterns, see [“Using Fill Patterns to Distinguish Grouped Bar Charts” on page 636](#).

Note: An important distinction between `HBARPARM` and `HBAR` is that the response variable is required for `HBARPARM`. In addition, the response variable should contain pre-summarized computed values such as a sum or a mean.

Example: [“About Bar Charts” on page 42](#)

Syntax

HBARPARM `CATEGORY=` *category-variable* `RESPONSE=` *numeric-variable* `</ option(s)>`;

Summary of Optional Arguments

Bar options

BARWIDTH= *numeric-value*

specifies the width of the bars as a ratio of the maximum possible width.

DATASKIN= `NONE` | `CRISP` | `GLOSS` | `MATTE` | `PRESSED` | `SHEEN`

specifies a special effect to be used on all filled bars.

DISCRETEOFFSET= *numeric-value*

specifies an amount to offset all bars from the category midpoints.

FILL | **NOFILL**

specifies whether the bars are filled.

FILLATTRS= *style-element* | (`COLOR=` *color*)

specifies the appearance of the fill for the bars.

LIMITATTRS= *style-element* `<(options)>` | `(options)`

specifies the appearance of the limit lines in the plot.

LIMITLOWER= *numeric-variable*

specifies values for the lower endpoints on the limit lines.

LIMITUPPER= *numeric-variable*

specifies values for the upper endpoints on the limit lines.

MISSING

processes missing values as a valid category value and creates a bar for it.

OUTLINE | NOOUTLINE

specifies whether the bars have outlines.

Group options

CLUSTERWIDTH= *numeric-value*

specifies the cluster width as a ratio of the maximum width.

GROUP= *variable*

specifies a variable that is used to group the data.

GROUPORDER= *ASCENDING | DESCENDING | DATA*

specifies the ordering of bars within a group.

Label options

DATALABEL *<= variable>*

displays a label for each data point.

DATALABELATTRS= *style-element <(options)> | (options)*

specifies the appearance of the labels in the plot when you use the **DATALABEL=** option.

Plot options

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set.

LEGENDLABEL= *"text-string"*

specifies the label that identifies the bar chart in the legend.

NAME= *"text-string"*

specifies a name for the plot.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the bars and limits, if displayed.

URL= *character-variable*

specifies a character variable that contains URLs for Web pages to be displayed when parts of the plot are selected within an HTML page.

Required Arguments

CATEGORY= *category-variable*

specifies the variable that categorizes the data. All values are treated as discrete values. The input data for this variable should contain unique values. When the category values are not unique, a warning is logged, and multiple bars are superimposed at the duplicated category values.

The **CATEGORY** axis is always discrete.

RESPONSE= *numeric-variable*

specifies a numeric response variable. The input data is expected to be pre-summarized computed values (sum, mean, and so on).

Optional Arguments

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph. For more information, see [“Using SG Attribute Maps to Control Visual Attributes”](#) on page 605.

BARWIDTH= *numeric-value*

specifies the width of the bars as a ratio of the maximum possible width. The maximum width is equal to the distance between the center of each bar and the centers of the adjacent bars. Specify a value from 0.0 (narrowest) to 1.0 (widest).

For example, if you specify a width of 1, then there is no space between the bars. If you specify a width of .5, then the width of the bars is equal to the space between the bars.

Default: .8

1.0 when the GROUP option is specified and GROUPDISPLAY=CLUSTER

Interaction: When the GROUP option is specified, the bar width is determined by the maximum number of bars in any one group cluster. All bars are drawn with the same width, with the cluster being positioned symmetrically around the midpoint.

CLUSTERWIDTH= *numeric-value*

specifies the cluster width as a ratio of the maximum width. Specify a value from 0.0 (narrowest) to 1.0 (widest).

CLUSTERWIDTH is the fraction of the midpoint spacing used by all bars that are clustered around a midpoint (category value). The bar width is applied to the maximum bar spacing divided by the maximum number of bars in any one cluster.

Default: 0.8

Interaction: This option is applicable only when the GROUP option is specified, when GROUPDISPLAY=CLUSTER, and when the category axis is discrete.

DATALABEL <= *variable*>

displays a label for each data point. If you specify a variable, then the values of that variable are used for the data labels. If you do not specify a variable, then the values of the calculated response are used for the data labels.

DATALABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the DATALABEL= option. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

Here is an example that specifies a style element:

```
DATALABELATTRS=GraphLabelText
```

options can be one or more of the following:

COLOR= *color*

specifies the color of the label characters. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see [“Color-Naming Schemes”](#) in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphDataText style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

FAMILY= “*font-family*”
 specifies the font family for the label characters. The SAS ODS styles use TrueType system fonts. For more information, see “TrueType Fonts Supplied by SAS” in Chapter 15 of *SAS Language Reference: Concepts*.
Default: The default font family is specified by the FontFamily attribute of the GraphDataText style element in the current style.


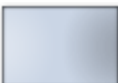
SIZE= *n* <*units*>
 specifies the font size of the label characters. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.
Default: The default font size is specified by the FontSize attribute of the GraphDataText style element in the current style.

STYLE= ITALIC | NORMAL
 specifies whether the label characters are italic (ITALIC) or normal (NORMAL).
Default: The default font style is specified by the FontStyle attribute of the GraphDataText style element in the current style.

WEIGHT= BOLD | NORMAL
 specifies whether the label characters are bold (BOLD) or normal (NORMAL).
Default: The default font weight is specified by the FontWeight attribute of the GraphDataText style element in the current style.

Here is an example expression:
 DATALABELATTRS=(Color=Green Family=Arial Size=8 Style=Italic Weight=Bold)
Default: The default style element is GraphDataText.
Interaction: This option has no effect unless the DATALABEL option is also specified.

DATASKIN= NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
 specifies a special effect to be used on all filled bars. Specify one of the following (the samples were generated using the HTMLBlue style):

NONE	
CRISP	
GLOSS	
MATTE	
PRESSED	

SHEEN

**Default:** NONE**Interactions:**

This option has no effect if the applied style uses a fill pattern and FILL is disabled for the style. If the style has both FILL and FILLPATTERN enabled, data skins will work. For more information about fill patterns, see [“Using Fill Patterns to Distinguish Grouped Bar Charts” on page 636](#). For instructions about viewing the contents of style templates, see [“About Styles and Style Elements” on page 620](#).

This option has no effect if you also specify NOFILL.

DISCRETEOFFSET= *numeric-value*

specifies an amount to offset all bars from the category midpoints. Specify a value from -0.5 (left offset) to +0.5 (right offset).

Default: 0.0 (no offset)**FILL | NOFILL**

specifies whether the bars are filled. The FILL option shows the fill color for the bars. The NOFILL option hides the fill color for the bars.

Default: FILL**Interactions:**

Specifying FILL also hides the outlines.

If NOFILL and NOOUTLINE are both specified, then both options are ignored.

FILLATTRS= *style-element* | (COLOR= *color*)

specifies the appearance of the fill for the bars. You can specify the color of the fill by using a style element or by using the COLOR= suboption. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphDataDefault style element in the current style. For grouped data, the default color is specified by the Color attribute of the GraphData1... GraphData*n* style elements in the current style.

Interaction: This option has no effect if you specify the NOFILL option.

GROUP= *variable*

specifies a variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes.

GROUPORDER= ASCENDING | DESCENDING | DATA

specifies the ordering of bars within a group.

ASCENDING

displays each group in ascending group value.

DESCENDING

displays each group in descending group value.

DATA

displays each group in data order of the group variable.

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

LEGENDLABEL= “text-string”

specifies the label that identifies the bar chart in the legend. By default, the label of the RESPONSE= variable is used. If there is no response variable label, the name of the response variable is used.

Interaction: The LEGENDLABEL= option has no effect if you also specify the GROUP= option in the same plot statement.

LIMITATTRS= style-element <(options)> | (options)

specifies the appearance of the limit lines in the plot. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= color

specifies the color of the line.

You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: The default color is specified by the ContrastColor attribute of the GraphError style element in the current style.

PATTERN= line-pattern

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see “Line Patterns” on page 663.

Default: The default line pattern is specified by the LineStyle attribute of the GraphError style element in the current style.

THICKNESS= n <units>

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default line thickness is specified by the LineThickness attribute of the GraphError style element in the current style.

LIMITLOWER= numeric-variable

specifies values for the lower endpoints on the limit lines. Limit lines are displayed as line segments with a serif at the end.

Default: The lower segments of the limit lines are not displayed. (Limit lines are displayed only if either LIMITLOWER= or LIMITUPPER= is specified.)

Interactions:

If LIMITUPPER= is also specified, then the plot displays the lower and upper segments of the limit lines.

This option is ignored if GROUP= is specified and GROUPDISPLAY=STACK.

The appearance of the limit lines can be controlled by the LIMITATTRS= option.

LIMITUPPER= numeric-variable

specifies values for the upper endpoints on the limit lines. Limit lines are displayed as line segments with a serif at the end.

Default: The upper segments of the limit lines are not displayed. (Limit lines are displayed only if either LIMITLOWER= or LIMITUPPER= is specified.)

Interactions:

If LIMITLOWER= is also specified, then the plot displays the lower and upper segments of the limit lines.

This option is ignored if GROUP= is specified and GROUPDISPLAY=STACK.

The appearance of the limit lines can be controlled by the LIMITATTRS= option.

MISSING

processes missing values as a valid category value and creates a bar for it.

NAME= *“text-string”*

specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note: The name is case-sensitive. When you refer to the specified name in other statements, you must use the same exact capitalization.

OUTLINE | NOOUTLINE

specifies whether the bars have outlines. The OUTLINE option shows the outlines. The NOOUTLINE option hides the outlines.

Default: OUTLINE

Interactions:

Specifying OUTLINE also hides the fill color.

If NOOUTLINE and NOFILL are both specified, then both options are ignored.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the bars and limits, if displayed. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default: 0.0

URL= *character-variable*

specifies a character variable that contains URLs for Web pages to be displayed when parts of the plot are selected within an HTML page.

Default: By default, no HTML links are created.

Interactions:

This option affects graphics output that is created through the ODS HTML destination only. For more information about ODS destinations, see [“Specifying a Destination” on page 649](#).

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For more information, see [“Using the ODS GRAPHICS Statement” on page 651](#).

HBOX Statement

Creates a horizontal box plot that shows the distribution of your data.

Interaction: The HBOX statement cannot be used together with other plot statements in the SGPanel procedure. Box plots can be overlaid with other box plots. However, overlaid box plots must have the same category variables.

Example: [“About Box Plots” on page 37](#)

Syntax

HBOX *analysis-variable* </ option(s)>;

Summary of Optional Arguments

Box options

BOXWIDTH= *numeric-value*

specifies the width of the box.

CAPSHAPE= **BRACKET** | **LINE** | **SERIF**

specifies the shape of the whisker cap lines.

CLUSTERWIDTH= *numeric-value*

specifies the cluster width as a ratio of the midpoint spacing.

CONNECT= **MEAN** | **MEDIAN** | **Q1** | **Q3** | **MIN** | **MAX**

specifies that a connect line joins a statistic from box to box.

CONNECTATTRS= *style-element* <(options)> | (options)

specifies the appearance of the lines that connect multiple boxes.

DISCRETEOFFSET= *numeric-value*

specifies an amount to offset all boxes from the discrete tick marks.

EXTREME

specifies that the whiskers can extend to the maximum and minimum values for the analysis variable, and that outliers are not identified.

FILL | **NOFILL**

specifies whether the boxes are filled with color.

FILLATTRS= *style-element* | (**COLOR**= *color*)

specifies the appearance of the fill for the boxes.

LINEATTRS= *style-element* <(options)> | (options)

specifies the appearance of the box outlines.

MEANATTRS= *style-element* <(options)> | (options)

specifies the appearance of the marker that represents the mean in the box.

MEDIANATTRS= *style-element* <(options)> | (options)

specifies the appearance of the line that represents the median.

NOCAPS

hides the cap lines for the whiskers.

NOMEAN

hides the mean marker.

NOMEDIAN

hides the median line.

NOOUTLIERS

hides the outliers from the plot.

NOTCHES

specifies that the boxes be notched.

OUTLIERATTRS= *style-element* <(options)> | (options)

specifies the appearance of the marker that represents the outliers.

WHISKERATTRS= *style-element* <(options)> | (options)

specifies the appearance of the whisker and cap lines.

Group options

GROUP= *variable*

specifies a variable that is used to group the data.

GROUPDISPLAY= **CLUSTER** | **OVERLAY**

specifies how to display grouped boxes.

GROUPORDER= **ASCENDING** | **DESCENDING** | **DATA**

specifies the ordering of boxes within a group.

Label options

DATALABEL <= *variable*>

adds data labels for the outlier markers.

DATALABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the DATALABEL= option.

LABELFAR

specifies that only the far outliers have data labels.

Plot options

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set.

CATEGORY= *category-variable*

specifies the category variable for the plot.

FREQ= *numeric-variable*

specifies how many times observations are repeated for computational purposes.

LEGENDLABEL= *“text-string”*

specifies a label that identifies the box plot in the legend.

MISSING

processes missing values as a valid category value and creates a box for it.

NAME= *“text-string”*

specifies a name for the plot.

PERCENTILE= 1 | 2 | 3 | 4 | 5

specifies a method for computing the percentiles for the plot.

SPREAD

relocates outlier points that have identical values to prevent overlapping.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the plot.

Required Argument

analysis-variable

specifies the analysis variable for the plot. If you do not specify the CATEGORY= option, then one box is created for the analysis variable.

Optional Arguments

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph. For more information, see [“Using SG Attribute Maps to Control Visual Attributes” on page 605](#).

BOXWIDTH= *numeric-value*

specifies the width of the box. Specify a value between 0.0 (0% of the available width) and 1.0 (100% of the available width).

Default: 0.4

When GROUP is specified, the default box width is 0.6.

CAPSHAPE= BRACKET | LINE | SERIF

specifies the shape of the whisker cap lines. Specify one of the following values:

BRACKET

displays a straight line with brackets.

LINE

displays a straight line.

SERIF

displays a short straight line.

CATEGORY= *category-variable*

specifies the category variable for the plot. A box plot is created for each distinct value of the category variable.

If you explicitly set the category axis type to LINEAR and use a numeric category variable, the box plot becomes an interval plot. Otherwise, the box plot is discrete.

CLUSTERWIDTH= *numeric-value*

specifies the cluster width as a ratio of the midpoint spacing. Specify a value from 0.1 (narrowest) to 1.0 (widest).

Default: 0.7

Interaction: This option is applicable only when a GROUP is in effect and the category axis is discrete.

CONNECT= MEAN | MEDIAN | Q1 | Q3 | MIN | MAX

specifies that a connect line joins a statistic from box to box.

Interaction: This option applies only when the CATEGORY option is used to generate multiple boxes.

Tip: You can use the CONNECTATTRS option to specify attributes for the connect line.

CONNECTATTRS= *style-element* <(options)> | (options)

specifies the appearance of the lines that connect multiple boxes. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

Here is an example that specifies a style element: **CONNECTATTRS=GraphData3**

options can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphConnectLine style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see [“Line Patterns” on page 663](#).

Default: For ungrouped data, the default line pattern is specified by the LineStyle attribute of the GraphConnectLine style element in the current style. For grouped data, the default line pattern is specified by the LineStyle attribute of the GraphData1 ... GraphData*n* style elements in the current style.

THICKNESS= *n* <units>

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see [“Units of Measurement” on page 659](#).

Default: For ungrouped data, the default line thickness is specified by the LineThickness attribute of the GraphConnectLine style element in the current

style. For grouped data, the default line thickness is specified by the LineThickness attribute of the GraphData1 ... GraphData*n* style elements in the current style.

Here is an example expression:

```
CONNECTATTRS=(Color= "light green" Pattern= MediumDash Thickness= 4)
```

Default: The default style element is GraphConnectLine.

Interactions:

This option takes effect only if the CONNECT= option is also specified.

This option is ignored if the GROUP= option is also specified.

DATALABEL <= *variable*>

adds data labels for the outlier markers. If you specified a variable, then the values for that variable are used for the data labels. If you did not specify a variable, then the values of the analysis variable are used.

Note: This option has no effect unless the plot contains outlier points.

DATALABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the DATALABEL= option. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

Here is an example that specifies a style element:

```
DATALABELATTRS=GraphLabelText
```

options can be one or more of the following:

COLOR= *color*

specifies the color of the label characters. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphDataText style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

FAMILY= “*font-family*”

specifies the font family for the label characters. The SAS ODS styles use TrueType system fonts. For more information, see “TrueType Fonts Supplied by SAS” in Chapter 15 of *SAS Language Reference: Concepts*.

Default: The default font family is specified by the FontFamily attribute of the GraphDataText style element in the current style.

SIZE= *n* <*units*>

specifies the font size of the label characters. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default font size is specified by the FontSize attribute of the GraphDataText style element in the current style.

STYLE= ITALIC | NORMAL

specifies whether the label characters are italic (ITALIC) or normal (NORMAL).

Default: The default font style is specified by the FontStyle attribute of the GraphDataText style element in the current style.

WEIGHT= BOLD | NORMAL

specifies whether the label characters are bold (BOLD) or normal (NORMAL).

Default: The default font weight is specified by the FontWeight attribute of the GraphDataText style element in the current style.

Here is an example expression:

```
DATALABELATTRS=(Color=Green Family=Arial Size=8 Style=Italic Weight=Bold)
```

Default: The default style element is GraphDataText.

Interaction: This option has no effect unless the DATALABEL option is also specified.

DISCRETEOFFSET= *numeric-value*

specifies an amount to offset all boxes from the discrete tick marks.

Specify a value from -0.5 (left offset) to +0.5 (right offset). If you specify a value outside of this range, an error message appears in the SAS log and the graph is not produced.

Default: 0.0 (no offset)

EXTREME

specifies that the whiskers can extend to the maximum and minimum values for the analysis variable, and that outliers are not identified. When you do not specify the EXTREME option, the whiskers cannot be longer than 1.5 times the length of the box.

FILL | NOFILL

specifies whether the boxes are filled with color. The FILL option shows the fill color. The NOFILL option hides the fill color.

Default: FILL

FILLATTRS= *style-element* | (COLOR= *color*)

specifies the appearance of the fill for the boxes. You can specify the color of the fill by using a style element or by using the COLOR= suboption. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphDataDefault style element in the current style.

For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

Interaction: This option has no effect if you also specify the NOFILL option.

FREQ= *numeric-variable*

specifies how many times observations are repeated for computational purposes. Each observation is repeated *n* times for computational purposes, where *n* is the value of the numeric variable. If *n* is not an integer, then it is truncated to an integer. If *n* is less than 1 or missing, then it is excluded from the analysis.

Interaction: If your plot is overlaid with other categorization plots, then the first FREQ variable that you specified is used for all of the plots.

GROUP= *variable*

specifies a variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes.

GROUPDISPLAY= CLUSTER | OVERLAY

specifies how to display grouped boxes.

CLUSTER

the boxes are drawn adjacent to each other.

OVERLAY

all the boxes for a given group value are drawn at the exact coordinate and might overlap. Each group is represented by unique visual attributes derived from the GraphData1... GraphData*n* style elements in the current style.

Default: CLUSTER for a discrete category axis

OVERLAY for a linear axis

Restriction: GROUPDISPLAY=CLUSTER can take effect only when the category axis is discrete.

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

GROUPORDER= ASCENDING | DESCENDING | DATA

specifies the ordering of boxes within a group.

ASCENDING

displays each group in ascending group value.

DESCENDING

displays each group in descending group value.

DATA

displays each group in data order of the group variable.

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

LABELFAR

specifies that only the far outliers have data labels. Far outliers are points whose distance from the box is more than three times the length of the box.

Note: This option has no effect if you do not specify the DATALABEL option, or if there are no far outliers.

LEGENDLABEL= “text-string”

specifies a label that identifies the box plot in the legend. By default, the label of the analysis variable is used.

LINEATTRS= style-element <(options)> | (options)

specifies the appearance of the box outlines. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

Here is an example that specifies a style element: **LINEATTRS=GraphData3**

For more information about style elements, see [“Style Elements for Use with ODS Graphics” on page 640](#).

options can be one or more of the following:

COLOR= color

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphDataDefault style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

PATTERN= line-pattern

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see [“Line Patterns” on page 663](#).

Default: For ungrouped data, the default line pattern is specified by the LineStyle attribute of the GraphDataDefault style element in the current style. For grouped data, the default line pattern is specified by the LineStyle attribute of the GraphData1 ... GraphData*n* style elements in the current style.

THICKNESS= *n* <*units*>

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see [“Units of Measurement” on page 659](#).

Default: For ungrouped data, the default line thickness is specified by the LineThickness attribute of the GraphDataDefault style element in the current style. For grouped data, the default line thickness is specified by the LineThickness attribute of the GraphData1 ... GraphData*n* style elements in the current style.

Here is an example expression:

```
LINEATTRS=(Color= "light green" Pattern= MediumDash Thickness= 4)
```

Default: For ungrouped data, line attributes are derived from the GraphDataDefault style element. For grouped data, the ContrastColor attribute of the GraphData1... GraphData*n* style elements are used.

Interactions:

This option takes effect only if the CONNECT= option is also specified.

This option is ignored if the GROUP= option is also specified.

MEANATTRS= *style-element* <(options)> | (options)

specifies the appearance of the marker that represents the mean in the box. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the markers. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see [“Color-Naming Schemes” in Chapter 14 of SAS/GRAPH: Reference](#).

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphBoxMean style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

SIZE= *n* <*units*>

specifies the size of the markers. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see [“Units of Measurement” on page 659](#).

Default: The default marker size is specified by the MarkerSize attribute of the GraphBoxMean style element in the current style.

SYMBOL= *symbol-name*

specifies the symbol for the markers. For a list of valid marker symbols, see [“Marker Symbols” on page 661](#).

Default: For ungrouped data, the default marker symbol is specified by the MarkerSymbol attribute of the GraphBoxMean style element in the current style. For grouped data, the default marker symbol is specified by the MarkerSymbol attribute of the GraphData1 ... GraphData*n* style elements in the current style.

WEIGHT= NORMAL | BOLD

specifies the marker weight of the marker.

Default: NORMAL

Default: GraphBoxMean is the default style element.

MEDIANATTRS= *style-element* <(options)> | (options)

specifies the appearance of the line that represents the median. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphBoxMedian style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see “Line Patterns” on page 663.

Default: For ungrouped data, the default line pattern is specified by the LineStyle attribute of the GraphBoxMedian style element in the current style. For grouped data, the default line pattern is specified by the LineStyle attribute of the GraphData1 ... GraphData*n* style elements in the current style.

THICKNESS= *n* <*units*>

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: For ungrouped data, the default line thickness is specified by the LineThickness attribute of the GraphBoxMedian style element in the current style. For grouped data, the default line thickness is specified by the LineThickness attribute of the GraphData1 ... GraphData*n* style elements in the current style.

Default: GraphBoxMedian is the default style element.

Interaction: This option is ignored if the NOMEDIAN option is also specified.

MISSING

processes missing values as a valid category value and creates a box for it.

NAME= “*text-string*”

specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note: The name is case-sensitive. When you refer to the specified name in other statements, you must use the same exact capitalization.

NOCAPS

hides the cap lines for the whiskers.

NOMEAN

hides the mean marker.

NOMEDIAN

hides the median line.

NOOUTLIERS

hides the outliers from the plot.

NOTCHES

specifies that the boxes be notched. The endpoints of the notches are at the following computed locations:

$$\text{median} \pm 1.58 (IQR / \sqrt{N})$$

For a visual description of the parts of a box plot, see “Details” on page 136 .

OUTLIERATTRS= style-element <(options)> | (options)

specifies the appearance of the marker that represents the outliers. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= color

specifies the color of the markers. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphOutlier style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

SIZE= n <units>

specifies the size of the markers. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default marker size is specified by the MarkerSize attribute of the GraphOutlier style element in the current style.

SYMBOL= symbol-name

specifies the symbol for the markers. For a list of valid marker symbols, see “Marker Symbols” on page 661.

Default: For ungrouped data, the default marker symbol is specified by the MarkerSymbol attribute of the GraphOutlier style element in the current style. For grouped data, the default marker symbol is specified by the MarkerSymbol attribute of the GraphData1 ... GraphData*n* style elements in the current style.

WEIGHT= NORMAL | BOLD

specifies the marker weight of the marker.

Default: NORMAL

Default: GraphOutlier is the default style element.

PERCENTILE= 1 | 2 | 3 | 4 | 5

specifies a method for computing the percentiles for the plot. For descriptions of each method, see “Calculating Percentiles” in the UNIVARIATE Procedure chapter of *Base SAS Procedures Guide: Statistical Procedures*.

Default: 5

SPREAD

relocates outlier points that have identical values to prevent overlapping.

Note: This option has no effect if your data does not contain two or more outliers with identical values for the analysis variable.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the plot. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default: 0.0

WHISKERATTRS= *style-element* <(options)> | (options)

specifies the appearance of the whisker and cap lines. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphBoxWhisker style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see “Line Patterns” on page 663.

Default: For ungrouped data, the default line pattern is specified by the LineStyle attribute of the GraphBoxWhisker style element in the current style. For grouped data, the default line pattern is specified by the LineStyle attribute of the GraphData1 ... GraphData*n* style elements in the current style.

THICKNESS= *n* <units>

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: For ungrouped data, the default line thickness is specified by the LineThickness attribute of the GraphBoxWhisker style element in the current style. For grouped data, the default line thickness is specified by the LineThickness attribute of the GraphData1 ... GraphData*n* style elements in the current style.

Default: GraphBoxWhisker is the default style element.

Interaction: This option is ignored if the NOMEDIAN option is also specified.

Details

Horizontal and vertical box plots display the distribution of data by using a rectangular box and whiskers. Whiskers are lines that indicate a data range outside of the box.

Note: Though the following figure shows a vertical box plot, the basic concepts apply to horizontal box plots.

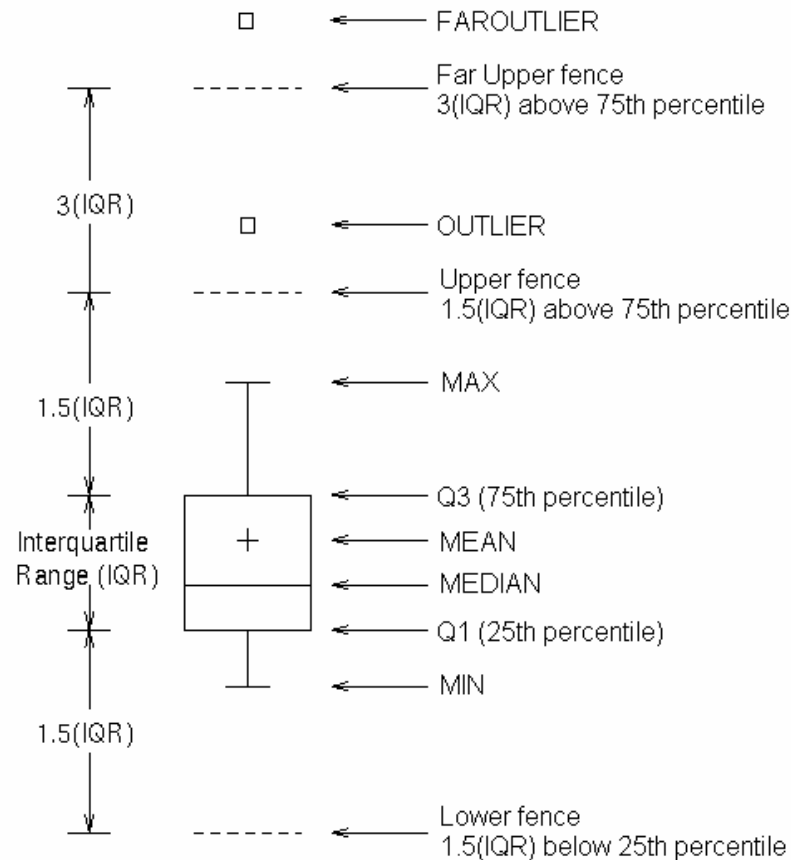
Figure 6.5 Parts of a Box Plot

Figure 6.5 on page 137 shows a diagram of a vertical box plot. The bottom and top edges of the box indicate the intra-quartile range (IQR). That is, the range of values between the first and third quartiles (the 25th and 75th percentiles). The marker inside the box indicates the mean value. The line inside the box indicates the median value.

The elements that are outside the box are dependent on your options. By default, the whiskers that extend from each box indicate the range of values that are outside of the intra-quartile range. However, the values are close enough not to be considered outliers (a distance less than or equal to $1.5 \times IQR$). If you specify the EXTREME option, then the whiskers indicate the entire range of values, including outliers.

Any points that are a distance of more than $1.5 \times IQR$ from the box are considered to be outliers. By default, these points are indicated by markers. If you specify DATALABEL= option, then the outlier points have data labels. If you also specify the LABELFAR option, then only outliers that are $3 \times IQR$ from the box have data labels.

HIGHLOW Statement

Creates a display of floating vertical or horizontal lines or bars that represent high and low values. The statement also gives you the option to display open and close values as tick marks and to specify a variety of plot attributes.

Note: This plot does not summarize data. If multiple observations have the same X or Y value, the observations are all plotted separately based on their values.

Example: [“About High-Low Charts” on page 22](#)

Syntax

HIGHLOW X= *variable* | Y= *variable*

HIGH= *numeric-variable* LOW= *numeric-variable*

</option(s)>;

Summary of Optional Arguments

Group options

CLUSTERWIDTH= *numeric-value*

specifies the cluster width as a ratio of the maximum width.

GROUP= *variable*

specifies a variable that is used to group the data.

GROUPDISPLAY= OVERLAY | CLUSTER

specifies how to display grouped data.

GROUPORDER= ASCENDING | DESCENDING | DATA

specifies the ordering of lines or bars within a group.

NOMISSINGGROUP

specifies that missing values of the group variable are not included in the plot.

HIGHLOW options

BARWIDTH= *numeric-value*

specifies the width of the bars as a ratio of the maximum possible width.

CLOSE= *numeric-variable*

specifies the data for the CLOSE tick on the bar or line.

DISCRETEOFFSET= *numeric-value*

specifies an amount to offset all lines or bars from the category midpoints.

FILL | NOFILL

specifies whether the area fill is visible for bars.

FILLATTRS= *style-element* | (COLOR= *color*)

specifies the appearance of the area fill for the bar.

HIGHCAP= *character-variable* | option | NONE

specifies the type of arrowhead used at the high end of the bar or line.

HIGHLABEL= *variable*

specifies the label to be shown at the high end of the line or bar.

INTERVALBARWIDTH= *numeric-value*

specifies the thickness of the bar when the X (or Y) data is numeric.

LABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the HIGHLABEL= option, the LOWLABEL= option, or both options.

LINEATTRS= *style-element* <(options)> | (options)

specifies the appearance of the outlines for the band.

LOWCAP= *character-variable* | option | NONE

specifies the type of arrowhead used at the low end of the bar or line.

LOWLABEL= *variable*

specifies the label to be shown at the low end of the line or bar.

OPEN= *numeric-variable*

specifies the data for the OPEN tick on the bar or line.

OUTLINE | NOOUTLINE

specifies whether the outlines of the bars are visible.

TYPE= BAR | LINE

specifies how the data is to be represented.

Plot options**ATTRID= *variable***

specifies the value of the ID variable in an attribute map data set.

LEGENDLABEL= "*text-string*"

specifies a label that identifies the elements from the band plot in the legend.

NAME= "*text-string*"

specifies a name for the plot.

TRANSPARENCY= *value*

specifies the degree of transparency for the plot.

URL= *character-variable*

specifies a character variable that contains URLs for Web pages to be displayed when parts of the plot are selected within an HTML page.

Required Arguments**X= *variable* | Y= *variable***

specifies a variable that is used to plot the values along the X or Y axis.

Note: If you specify X=*variable*, then the statement creates vertical lines or bars on the X axis, with the HIGH and LOW values plotted along the Y axis. Conversely, if you specify Y=*variable*, then the statement creates horizontal lines or bars on the Y axis, with the HIGH and LOW values plotted along the X axis.

HIGH= *numeric-variable*

specifies the upper value for the floating lines or bars.

LOW= *numeric-variable*

specifies the lower value for the floating lines or bars.

Optional Arguments**ATTRID= *variable***

specifies the value of the ID variable in an attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph. For more information, see [“Using SG Attribute Maps to Control Visual Attributes” on page 605](#).

BARWIDTH= *numeric-value*

specifies the width of the bars as a ratio of the maximum possible width. The maximum width is equal to the distance between the center of each bar and the centers of the adjacent bars. Specify a value from 0.0 (narrowest) to 1.0 (widest).

For example, if you specify a width of 1, then there is no distance between the bars. If you specify a width of .5, then the width of the bars is equal to the space between the bars.

Default: 0.85

Requirement: This option is applicable only when the X or Y axis is discrete.

Interaction: This option has no effect unless TYPE= BAR.

CLOSE= *numeric-variable*

specifies the data for the CLOSE tick on the bar or line. For a vertical plot, the tick value is represented by an indicator on the side of the bar or line that has higher X

values. For a horizontal plot, the value is represented by an indicator on the side with higher Y values.

CLUSTERWIDTH= *numeric-value*

specifies the cluster width as a ratio of the maximum width. Specify a value from 0.0 (narrowest) to 1.0 (widest).

Requirement: This option is applicable only when the X or Y axis is discrete.

Interactions:

This option is applicable only when TYPE= BAR.

This option is applicable only when the GROUP option is specified and GROUPDISPLAY= CLUSTER.

DISCRETEOFFSET= *numeric-value*

specifies an amount to offset all lines or bars from the category midpoints. Specify a value from -0.5 (left offset) to +0.5 (right offset).

Default: 0.0 (no offset from the category midpoints)

Requirement: This option is applicable only when the X or Y axis is discrete.

FILL | NOFILL

specifies whether the area fill is visible for bars. The FILL option shows the area fill. The NOFILL option hides the area fill.

Default: The default status of the area fill is specified by the DisplayOpts attribute of the GraphBand style element in the current style.

Interactions:

This option has no effect unless TYPE= BAR.

Specifying FILL also hides any visible outlines.

If NOFILL and NOOUTLINE are both specified, then both options are ignored.

FILLATTRS= *style-element* | (COLOR= *color*)

specifies the appearance of the area fill for the bar. You can specify the color of the fill by using a style element or by using the COLOR= suboption. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphDataConfidence style element in the current style. For grouped data, the default color is specified by the Color attribute of the GraphData1... GraphData*n* style elements in the current style.

Interactions:

This option is applicable only when TYPE= BAR.

This option has no effect if you also specify the NOFILL option.

GROUP= *variable*

specifies a variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes.

GROUPDISPLAY= OVERLAY | CLUSTER

specifies how to display grouped data.

OVERLAY

groups are overlaid without any clustering. Each group is represented by unique visual attributes derived from the GraphData1... GraphData*n* style elements in the current style.

CLUSTER

observations with different group values are displayed in adjacent clusters around the category value. Each set of group values is centered at the midpoint tick mark for the category.

Default: OVERLAY

Restriction: GROUPDISPLAY=CLUSTER has no effect unless the X or Y axis is discrete.

Interactions:

GROUPDISPLAY=CLUSTER is applicable only when TYPE=BAR.

This option is ignored unless GROUP= is specified.

GROUPORDER= ASCENDING | DESCENDING | DATA

specifies the ordering of lines or bars within a group.

ASCENDING

displays each group of lines or bars in ascending group value.

DESCENDING

displays each group of lines or bars in descending group value.

DATA

displays each group of lines or bars in data order of the group column.

Interactions:


This option is ignored unless GROUP= is specified.

This option is applicable only when GROUPDISPLAY=CLUSTER and TYPE=BAR.

HIGHCAP= *character-variable* | *option* | NONE


specifies the type of arrowhead used at the high end of the bar or line. You can specify one of the following options, or you can specify a character variable that contains one of the following options:

SERIF

a small serif. 


The serif is supported for TYPE=LINE only.

BARBEDARROW

a solid triangle with an indent at the base. 


The barbed arrow is supported for TYPE=LINE only.

FILLEDARROW

a solid triangle. 


The filled arrow is supported for TYPE=BAR and TYPE=LINE.

OPENARROW

resembles the letter "V". 

The open arrow is supported for TYPE=LINE only.

CLOSEDARROW

an outline of a triangle. 

The closed arrow is supported for TYPE=BAR only.

Default: NONE

Restriction: Arrow heads are not displayed for very short bars. Bar height must be at least twice the size of the arrow heads in order for the arrow heads to appear.

HIGHLABEL= *variable*

specifies the label to be shown at the high end of the line or bar.

INTERVALBARWIDTH= *numeric-value*

specifies the thickness of the bar when the X (or Y) data is numeric.

Default: The default thickness of the bar is derived from the minimum interval between the data values along X or Y.

Requirement: This option is applicable only when the X or Y axis is a linear or a TIME axis.

Interaction: This option has no effect unless TYPE=BAR.

LABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the HIGHLABEL= option, the LOWLABEL= option, or both options. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the characters. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphDataText style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

FAMILY= “*font-family*”

specifies the font family for the marker characters. The SAS ODS styles use TrueType system fonts. For more information, see “TrueType Fonts Supplied by SAS” in Chapter 15 of *SAS Language Reference: Concepts*.

Default: The default font family is specified by the FontFamily attribute of the GraphDataText style element in the current style.

SIZE= *n* <*units*>

specifies the font size of the characters. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default font size is specified by the FontSize attribute of the GraphDataText style element in the current style.

STYLE= ITALIC | NORMAL

specifies whether the characters are italic (ITALIC) or normal (NORMAL).

Default: The default font style is specified by the FontStyle attribute of the GraphDataText style element in the current style.

WEIGHT= BOLD | NORMAL

specifies whether the characters are bold (BOLD) or normal (NORMAL).

Default: The default font weight is specified by the FontWeight attribute of the GraphDataText style element in the current style.

LEGENDLABEL= “*text-string*”

specifies a label that identifies the elements from the band plot in the legend. By default, the label “band” is used for ungrouped data, and the group values are used for grouped data.

Interaction: The LEGENDLABEL= option has no effect if you also specify the GROUP= option.

LINEATTRS= *style-element* <(options)> | (options)

specifies the appearance of the outlines for the band. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphDataDefault style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see “Line Patterns” on page 663.

Default: For ungrouped data, the default line pattern is specified by the LineStyle attribute of the GraphDataDefault style element in the current style. For grouped data, the default line pattern is specified by the LineStyle attribute of the GraphData1 ... GraphData*n* style elements in the current style.

THICKNESS= *n* <units>

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.


Default: For ungrouped data, the default line thickness is specified by the LineThickness attribute of the GraphDataDefault style element in the current style. For grouped data, the default line thickness is specified by the LineThickness attribute of the GraphData1 ... GraphData*n* style elements in the current style.

Interaction: This option has no effect unless you also specify the OUTLINES option.

LOWCAP= *character-variable* | *option* | NONE


specifies the type of arrowhead used at the low end of the bar or line. You can specify one of the following options, or you can specify a character variable that contains one of the following options:

SERIF

a small serif. 


The serif is supported for TYPE=LINE only.

BARBEDARROW

a solid triangle with an indent at the base. 


The barbed arrow is supported for TYPE=LINE only.

FILLEDARROW

a solid triangle. 


The filled arrow is supported for TYPE=BAR and TYPE=LINE.

OPENARROW

resembles the letter "V". 

The open arrow is supported for TYPE=LINE only.

CLOSEDARROW

an outline of a triangle. 

The closed arrow is supported for TYPE=BAR only.

Default: NONE

Restriction: Arrow heads are not displayed for very short bars. Bar height must be at least twice the size of the arrow heads in order for the arrow heads to appear.

LOWLABEL= *variable*

specifies the label to be shown at the low end of the line or bar.

NOMISSINGGROUP

specifies that missing values of the group variable are not included in the plot.

Interaction: This option has no effect unless GROUP= is also specified.

NAME= "*text-string*"

specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note: The name is case-sensitive. When you refer to the specified name in other statements, you must use the same exact capitalization.

OPEN= *numeric-variable*

specifies the data for the OPEN tick on the bar or line.

For a vertical plot, the tick value is represented by an indicator on the side of the bar or line that has lower X values. For a horizontal plot, the value is represented by an indicator on the side with lower Y values.

OUTLINE | NOOUTLINE

specifies whether the outlines of the bars are visible. The OUTLINE option shows the outlines. The NOOUTLINE option hides the outlines.

Default: OUTLINE

Interactions:

This option has no effect unless TYPE= BAR.

Specifying OUTLINE also hides the fill color.

If NOOUTLINE and NOFILL are both specified, then both options are ignored.

TRANSPARENCY= *value*

specifies the degree of transparency for the plot. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default: 0.0

TYPE= BAR | LINE

specifies how the data is to be represented. BAR uses fill and outline attributes. LINE uses line attributes.

Default: LINE

URL= *character-variable*

specifies a character variable that contains URLs for Web pages to be displayed when parts of the plot are selected within an HTML page.

Default: By default, no HTML links are created.

Interactions:

This option affects graphics output that is created through the ODS HTML destination only. For more information about ODS destinations, see [“Specifying a Destination” on page 649](#).

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For more information, see [“Using the ODS GRAPHICS Statement” on page 651](#).

HISTOGRAM Statement

Creates a histogram that displays the frequency distribution of a numeric value.

- Interaction:** The HISTOGRAM statement can be combined only with DENSITY statements in the SGPanel procedure.
- Note:** The range of the response variable is automatically divided into an appropriate number of bins.
- Examples:** [“About Histograms” on page 40](#)
[“Example 1: Creating a Panel of Graph Cells with Histograms and Density Plots” on page 267](#)
-

Syntax

HISTOGRAM *response-variable* < / option(s)>;

Summary of Optional Arguments

Histogram options

- BINSTART=** *numeric-value*
 specifies the X coordinate of the first bin.
- BINWIDTH=** *numeric-value*
 specifies the bin width.
- BOUNDARY=** LOWER | UPPER
 specifies how boundary values are assigned to bins.
- FILL** | NOFILL
 specifies whether the area fill is visible.
- FILLATTRS=** *style-element* | (COLOR= *color*)
 specifies the appearance of the area fill.
- NBINS=** *numeric-value*
 specifies the number of bins.
- OUTLINE** | NOOUTLINE
 specifies whether outlines are displayed for the bars.
- SCALE=** COUNT | PERCENT | PROPORTION
 specifies the scaling that is applied to the vertical axis.

Plot options

- FREQ=** *numeric-variable*
 specifies how many times observations are repeated for computational purposes.
- LEGENDLABEL=** “*text-string*”
 specifies a label that identifies the histogram in the legend.

NAME= *“text-string”*

specifies a name for the plot.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the histogram.

Required Argument

response-variable

specifies a response variable for the histogram.

Optional Arguments

BINSTART= *numeric-value*

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: The default value is determined by the system.

BINWIDTH= *numeric-value*

specifies the bin width. The system determines the number of bins. The bins always span the range of the data.

Default: The default value is determined by the system.

Interaction: This option is ignored if the NBINS= option is also specified.

BOUNDARY= LOWER | UPPER

specifies how boundary values are assigned to bins.

LOWER

specifies that boundary values are assigned to the lower bin.

UPPER

specifies that boundary values are assigned to the upper bin.

Default: UPPER

FILL | NOFILL

specifies whether the area fill is visible. The FILL option shows the area fill. The NOFILL option hides the area fill.

Default: The default status of the area fill is specified by the DisplayOpts style attribute of the GraphHistogram style element in the current style.

Interactions:

Specifying FILL also hides any visible outlines.

If NOFILL and NOOUTLINE are both specified, then both options are ignored.

FILLATTRS= *style-element* | (**COLOR=** *color*)

specifies the appearance of the area fill. You can specify the color of the fill by using a style element or by using the COLOR= suboption. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphDataDefault style element in the current style.

For grouped data, the default color is specified by the Color attribute of the GraphData1 ... GraphData*n* style elements in the current style.

Interaction: This option has no effect if you specify the NOFILL option.

FREQ= *numeric-variable*

specifies how many times observations are repeated for computational purposes. Each observation is repeated n times for computational purposes, where n is the value of the numeric variable. If n is not an integer, then it is truncated to an integer. If n is less than 1 or missing, then it is excluded from the analysis.

LEGENDLABEL= "*text-string*"

specifies a label that identifies the histogram in the legend. By default, the label of the response variable is used.

NAME= "*text-string*"

specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note: The name is case-sensitive. When you refer to the specified name in other statements, you must use the same exact capitalization.

NBINS= *numeric-value*

specifies the number of bins. The system determines the BINWIDTH= value. The bins always span the range of the data.

The procedure attempts to produce tick values that are easily interpreted (for example, 5, 10, 15, 20). The procedure sometimes adjusts the location of the first bin and the bin width accordingly. As a result, the number of bins shown in the plot might not exactly match the number specified with the NBINS= option.

Default: The default number of bins is determined by the system.

OUTLINE | NOOUTLINE

specifies whether outlines are displayed for the bars. The OUTLINE option shows the outlines. The NOOUTLINE option hides the outlines.

Default: The default status of the bar outlines is specified by the DisplayOpts attribute of the GraphHistogram style element in the current style.

Interactions:

Specifying OUTLINE also hides the fill color.

If NOOUTLINE and NOFILL are both specified, then both options are ignored.

SCALE= COUNT | PERCENT | PROPORTION

specifies the scaling that is applied to the vertical axis. Specify one of the following values:

COUNT

the axis displays the frequency count.

PERCENT

the axis displays values as a percentage of the total.

PROPORTION

the axis displays values as proportions (0.0 to 1.0) of the total.

Default: PERCENT

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the histogram. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default: 0.0

HLINE Statement

Creates a horizontal line chart (the line is vertical). You can use the HLINE statement with the HBAR statement to create a horizontal bar-line chart.

Interaction: The HLINE statement can be combined only with other categorization plot statements in the SGPPANEL procedure. See [“Plot Type Compatibility” on page 81](#).

Example: [“About Line Charts” on page 47](#)

Syntax

HLINE *category-variable* < / option(s)>;

Summary of Optional Arguments

Group options

CLUSTERWIDTH= *numeric-value*

specifies the width of the group clusters as a fraction of the midpoint spacing.

GROUP= *variable*

specifies a category variable to divide the values into groups.

GROUPDISPLAY= **CLUSTER** | **OVERLAY**

specifies how to display grouped lines.

GROUPORDER= **ASCENDING** | **DESCENDING** | **DATA**

specifies the ordering of lines within a group.

Label options

CURVELABEL <=“*text-string*”>

adds a label for the line.

CURVELABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the **CURVELABEL**= option.

CURVELABELPOS= **MIN** | **MAX** | **START** | **END**

specifies the location of the curve label.

DATALABEL <= *variable*>

displays a label for each data point.

DATALABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the **DATALABEL**= option.

Line options

BREAK

creates a break in the line for each missing value of the response variable.

DISCRETEOFFSET= *numeric-value*

specifies an amount to offset all lines from the discrete category values.

LIMITATTRS= *style-element* <(options)> | (options)

specifies the appearance of the limit lines in the plot.

LIMITS= **BOTH** | **LOWER** | **UPPER**

adds limit lines to the plot.

LIMITSTAT= **CLM** | **STDDEV** | **STDERR**

specifies the statistic for the limit lines.

LINEATTRS= *style-element* <(options)> | (options)

specifies the appearance of the lines in the line plot.

MARKERATTRS= *style-element* <(options)> | (options)

specifies the appearance of the markers in the plot.

MARKERS

adds data point markers to the plot.

MISSING

processes missing values as a valid category value and creates a line for it.

NOSTATLABEL

removes the statistic name from the axis and legend labels.

NUMSTD= *n*

specifies the number of standard units for the limit lines, when you specify
LIMITSTAT= STDDEV or LIMITSTAT= STDERR.

RESPONSE= *response-variable*

specifies a numeric response variable for the plot.

STAT= FREQ | MEAN | SUM

specifies the statistic for the horizontal axis.

Plot options

ALPHA= *numeric-value*

specifies the confidence level for the confidence limits.

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set.

CATEGORYORDER= RESPASC | RESPDESC

specifies the order in which the response values are arranged.

FREQ= *numeric-variable*

specifies how many times observations are repeated for computational purposes.

LEGENDLABEL= "*text-string*"

specifies the label that identifies the line plot in the legend.

NAME= "*text-string*"

specifies a name for the plot.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the lines and markers.

URL= *character-variable*

specifies a character variable that contains URLs for Web pages to be displayed when parts of the plot are selected within an HTML page.

WEIGHT= *numeric-variable*

specifies how observations are weighted.

Required Argument

category-variable

specifies the variable that classifies the observations into distinct subsets.

Optional Arguments

ALPHA= *numeric-value*

specifies the confidence level for the confidence limits. Specify a number between 0.00 (100% confidence) and 1.00 (0% confidence).

Default: .05

Interactions:

This option has no effect if you do not specify LIMITSTAT=CLM.

If your plot is overlaid with other categorization plots, then the first ALPHA value that you specify is used for all of the plots.

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph. For more information, see [“Using SG Attribute Maps to Control Visual Attributes” on page 605](#).

BREAK

creates a break in the line for each missing value of the response variable.

Notes:

The observation is excluded from the graph when there is a missing value for the category variable without the specification of the MISSING option. (No break occurs in the line.)

The observation is excluded from the graph when there is a missing value for the FREQ variable.

CATEGORYORDER= RESPASC | RESPDESC

specifies the order in which the response values are arranged. Specify one of the following values:

RESPASC

sorts the response values in ascending order.

RESPDESC

sorts the response values in descending order.

Default: By default, the plot is sorted in ascending order based on the category values.

Restriction: This option takes effect only when the plot statement specifies a response variable and the axis for that variable is numeric. If the axis is not numeric, an error is generated and a message is written to the SAS log.

Requirement: This option requires that you configure the panel to use either one column or one row, depending on the orientation of your charts. Use the ROWS= or the COLUMNS= option in the PANELBY statement. If you do not use this option and your graph contains multiple cells, the specified sort order is not correctly applied to all cells.

Interaction: If a group variable is specified in the plot statement, the response sort option is ignored and a note is generated in the log. (You can use a BY statement in the procedure if you want to group data.)

Note: If CATEGORYORDER is specified in multiple statements, the procedure sorts by the last statement in which it is specified.

CLUSTERWIDTH= *numeric-value*

specifies the width of the group clusters as a fraction of the midpoint spacing. Specify a value from 0.0 (narrowest) to 1.0 (widest).

Default: 0.8

Interaction: This option is applicable only when the GROUP option is specified, when GROUPDISPLAY=CLUSTER, and when the category axis is discrete.

CURVELABEL <=“*text-string*”>

adds a label for the line. You can also specify the label text. If you do not specify a label, then the label from the response variable is used.

CURVELABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the CURVELABEL= option. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

Here is an example that specifies a style element:

CURVELABELATTRS=GraphTitleText

options can be one or more of the following:

COLOR= *color*

specifies the color of the line label characters. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphValueText style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

FAMILY= “*font-family*”

specifies the font family for the label characters. The SAS ODS styles use TrueType system fonts. For more information, see “TrueType Fonts Supplied by SAS” in Chapter 15 of *SAS Language Reference: Concepts*.

Default: The default font family is specified by the FontFamily attribute of the GraphValueText style element in the current style.

Restriction: You cannot specify a list of font family names.

SIZE= *n* <*units*>

specifies the font size of the label characters. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default font size is specified by the FontSize attribute of the GraphValueText style element in the current style.

STYLE= ITALIC | NORMAL

specifies whether the label characters are italic or normal.

Default: The default font style is specified by the FontStyle attribute of the GraphValueText style element in the current style.

WEIGHT= BOLD | NORMAL

specifies whether the label characters are bold or normal.

Default: The default font weight is specified by the FontWeight attribute of the GraphValueText style element in the current style.

Here is an example expression:

CURVELABELATTRS=(Color=Green Family=Arial Size=8 Style=Italic Weight=Bold)

Default: The default style element is GraphValueText.

Interaction: This option has no effect unless the CURVELABEL option is also specified.

CURVELABELPOS= MIN | MAX | START | END

specifies the location of the curve label. Specify one of the following values:

MIN

places the label at the part of the curve closest to the minimum X axis value.

MAX

places the label at the part of the curve closest to the maximum X axis value.

START

places the curve label at the first point on the curve.

END

places the curve label at the last point on the curve.

Default: END

Interaction: This option has no effect unless the CURVELABEL option is also specified.

DATALABEL <= *variable*>

displays a label for each data point. If you specify a variable, then the values of that variable are used for the data labels. If you do not specify a variable, then the values of the calculated response are used for the data labels.

DATALABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the DATALABEL= option. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

Here is an example that specifies a style element:

DATALABELATTRS=GraphLabelText

options can be one or more of the following:

COLOR= *color*

specifies the color of the label characters. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphDataText style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

FAMILY= “*font-family*”

specifies the font family for the label characters. The SAS ODS styles use TrueType system fonts. For more information, see “TrueType Fonts Supplied by SAS” in Chapter 15 of *SAS Language Reference: Concepts*.

Default: The default font family is specified by the FontFamily attribute of the GraphDataText style element in the current style.

SIZE= *n* <*units*>

specifies the font size of the label characters. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default font size is specified by the FontSize attribute of the GraphDataText style element in the current style.

STYLE= ITALIC | NORMAL

specifies whether the label characters are italic (ITALIC) or normal (NORMAL).

Default: The default font style is specified by the FontStyle attribute of the GraphDataText style element in the current style.

WEIGHT= BOLD | NORMAL

specifies whether the label characters are bold (BOLD) or normal (NORMAL).

Default: The default font weight is specified by the FontWeight attribute of the GraphDataText style element in the current style.

Here is an example expression:

```
DATALABELATTRS=(Color=Green Family=Arial Size=8 Style=Italic Weight=Bold)
```

Default: The default style element is GraphDataText.

Interaction: This option has no effect unless the DATALABEL option is also specified.

DISCRETEOFFSET= *numeric-value*

specifies an amount to offset all lines from the discrete category values. Specify a value from -0.5 (left offset) to +0.5 (right offset).

Default: 0.0 (no offset)

Requirement: This option is applicable only when the category axis is discrete.

FREQ= *numeric-variable*

specifies how many times observations are repeated for computational purposes. Each observation is repeated n times for computational purposes, where n is the value of the numeric variable. If n is not an integer, then it is truncated to an integer. If n is less than 1 or missing, then it is excluded from the analysis.

Interaction: If your plot is overlaid with other categorization plots, then the first FREQ variable that you specified is used for all of the plots.

GROUP= *variable*

specifies a category variable to divide the values into groups. A separate plot is created for each unique value of the category variable.

Interaction: If you specify more than one categorization plot statement, then all of the plots must specify the same GROUP variable. If you do not specify the same GROUP= option for all of the categorization plots, then the GROUP= option has no effect.

GROUPDISPLAY= CLUSTER | OVERLAY

specifies how to display grouped lines.

CLUSTER

grouped items are drawn adjacent to each other.

OVERLAY

grouped items are drawn at the exact coordinate and might overlap. Each group is represented by unique visual attributes derived from the GraphData1... GraphData n style elements in the current style.

Default: OVERLAY

Restriction: GROUPDISPLAY=CLUSTER can take effect only when the category axis is discrete. If necessary, use a TYPE=DISCRETE option on the axis statement.

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

GROUPORDER= ASCENDING | DESCENDING | DATA

specifies the ordering of lines within a group.

ASCENDING

displays each group in ascending group value.

DESCENDING

displays each group in descending group value.

DATA

displays each group in data order of the group variable.

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

LEGENDLABEL= *“text-string”*

specifies the label that identifies the line plot in the legend. By default, the label of the response variable is used. If there is no response variable label, then the name of the response variable and the computed statistic (SUM or MEAN) are used. If you do not specify a response variable, then the legend label is “Frequency”.

Interaction: The LEGENDLABEL= option has no effect if you also specify the GROUP= option in the same plot statement.

LIMITATTRS= *style-element* <(options)> | (options)

specifies the appearance of the limit lines in the plot. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: The default color is specified by the ContrastColor attribute of the GraphError style element in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see “Line Patterns” on page 663.

Default: The default line pattern is specified by the LineStyle attribute of the GraphError style element in the current style.

THICKNESS= *n* <*units*>

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default line thickness is specified by the LineThickness attribute of the GraphError style element in the current style.

LIMITS= BOTH | LOWER | UPPER

adds limit lines to the plot. Specify one of the following values:

BOTH

adds lower and upper limit lines to the plot.

LOWER

adds lower limit lines to the plot.

UPPER

adds upper limit lines to the plot.

Default: By default, no limit lines are displayed. However, if you specify the LIMITSTAT= option, then the default is BOTH.

Interaction: Limit lines are displayed only when you specify STAT= MEAN.

LIMITSTAT= CLM | STDDEV | STDERR

specifies the statistic for the limit lines. Specify one of the following

CLM

confidence limits

STDDEV

standard deviation

STDERR
standard error

Default: CLM

Interaction: If you specify the LIMITSTAT= option, then the default value for the LIMITS= option is BOTH.

LINEATTRS= *style-element* <(options)> | (options)

specifies the appearance of the lines in the line plot. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphDataDefault style element in the current style.

For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see “Line Patterns” on page 663.

Default: For ungrouped data, the default line pattern is specified by the LineStyle attribute of the GraphDataDefault style element in the current style.

For grouped data, the default line pattern is specified by the LineStyle attribute of the GraphData1 ... GraphData*n* style elements in the current style.

THICKNESS= *n* <*units*>

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: For ungrouped data, the default line thickness is specified by the LineThickness attribute of the GraphDataDefault style element in the current style.

For grouped data, the default line thickness is specified by the LineThickness attribute of the GraphData1 ... GraphData*n* style elements in the current style.

MARKERATTRS= *style-element* <(options)> | (options)

specifies the appearance of the markers in the plot. This option has no effect unless you also specify the MARKERS option. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the markers. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphDataDefault style element in the current style.

For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

SIZE= *n* <*units*>

specifies the size of the markers. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see [“Units of Measurement” on page 659](#).

Default: The default marker size is specified by the MarkerSize attribute of the GraphDataDefault style element in the current style.

SYMBOL= *symbol-name*

specifies the symbol for the markers. For a list of valid marker symbols, see [“Marker Symbols” on page 661](#).

Default: For ungrouped data, the default marker symbol is specified by the MarkerSymbol attribute of the GraphDataDefault style element in the current style.

For grouped data, the default marker symbol is specified by the MarkerSymbol attribute of the GraphData1 ... GraphData*n* style elements in the current style.

MARKERS

adds data point markers to the plot.

MISSING

processes missing values as a valid category value and creates a line for it.

NAME= *“text-string”*

specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note: The name is case-sensitive. When you refer to the specified name in other statements, you must use the same exact capitalization.

NOSTATLABEL

removes the statistic name from the axis and legend labels.

NUMSTD= *n*

specifies the number of standard units for the limit lines, when you specify LIMITSTAT= STDDEV or LIMITSTAT= STDERR. You can specify any positive number, including decimals.

Default: 1

RESPONSE= *response-variable*

specifies a numeric response variable for the plot. The summarized values of the response variable are displayed on the horizontal axis.

STAT= FREQ | MEAN | SUM

specifies the statistic for the horizontal axis. Specify one of the following:

FREQ

the frequencies for the category variable. This is the default value when you do not specify the RESPONSE= option.

MEAN

the mean of the response variable.

SUM

the sum of the response variable. This is the default value when you specify the RESPONSE= option.

Interaction: If you do not specify the RESPONSE= option, then only the FREQ statistic can be used. If you specify the RESPONSE= option, then you can use either the SUM or MEAN statistics.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the lines and markers. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default: 0.0

URL= *character-variable*

specifies a character variable that contains URLs for Web pages to be displayed when parts of the plot are selected within an HTML page.

Default: By default, no HTML links are created.

Interactions:

This option affects graphics output that is created through the ODS HTML destination only. For more information about ODS destinations, see [“Specifying a Destination” on page 649](#).

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For more information, see [“Using the ODS GRAPHICS Statement” on page 651](#).

WEIGHT= *numeric-variable*

specifies how observations are weighted. Each observation is weighted by a factor of w for computational purposes, where w is the value of the numeric variable. w can be any numeric value. If w is 0, negative or missing, then that observation is excluded from the analysis.

Interaction: If your plot is overlaid with other categorization plots, then the first WEIGHT variable that you specified is used for all of the plots.

KEYLEGEND Statement

Adds a legend to the plot.

Syntax

KEYLEGEND <“*name-l*” ... “*name-n*”> </ *option(s)*>;

Summary of Optional Arguments

ACROSS= *n*

specifies the number of columns in the legend.

BORDER | NOBORDER

specifies whether the border around the legend is visible.

DOWN= *n*

specifies the number of rows in the legend.

POSITION= *position-value*

specifies the position of the legend within the plot.

TITLE= “*text-string*”

adds a title to the legend.

TITLEATTRS= *style-element* <(options)> | (options)

specifies the appearance of the legend title.

VALUEATTRS= *style-element* <(options)> | (options)

specifies the appearance of the legend value labels.

“*name-l*” ... “*name-n*”

specifies the names of one or more plots that you want to include in legend.

Optional Arguments

“name-1” ... “name-n”

specifies the names of one or more plots that you want to include in legend. Each name that you specify must correspond to a value that you entered for the NAME= option in a plot statement.

Note: If you do not specify a name, then the legend contains references to all of the plots in the graph.

ACROSS= *n*

specifies the number of columns in the legend. By default, the number of columns is determined automatically.

Note: Depending on the number of legend entries and the number of columns and rows that you specify, the legend might not fit in your graph. If your legend does not appear, then you might need to specify a different value for the ACROSS= option.

BORDER | NOBORDER

specifies whether the border around the legend is visible. The BORDER option shows the border. The NOBORDER option hides the border.

Default: BORDER

DOWN= *n*

specifies the number of rows in the legend. By default, the number of rows is determined automatically.

Note: Depending on the number of legend entries and the number of columns and rows that you specify, the legend might not fit in your graph. If your legend does not appear, then you might need to specify a different value for the DOWN= option.

POSITION= *position-value*

specifies the position of the legend within the plot. The positions are as follows:

BOTTOM

places the legend at the bottom of the plot.

LEFT

places the legend at the left side of the plot.

RIGHT

places the legend at the right side of the plot.

TOP

places the legend at the top of the plot.

Default: BOTTOM

Note: By default, if you use more than one KEYLEGEND statement, then each legend is placed in a different position.

TITLE= *“text-string”*

adds a title to the legend.

TITLEATTRS= *style-element* <(options)> | (options)

specifies the appearance of the legend title. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

Here is an example that specifies a style element:

TITLEATTRS=GraphTitleText

options can be one or more of the following:

COLOR= *color*

specifies the color of the title characters. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphLabelText style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

FAMILY= “*font-family*”

specifies the font family for the title characters. The SAS ODS styles use TrueType system fonts. For more information, see “TrueType Fonts Supplied by SAS” in Chapter 15 of *SAS Language Reference: Concepts*.

Default: The default font family is specified by the FontFamily attribute of the GraphLabelText style element in the current style.

Restriction: You cannot specify a list of font family names.

SIZE= *n* <*units*>

specifies the font size of the title characters. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default font size is specified by the FontSize attribute of the GraphLabelText style element in the current style.

STYLE= ITALIC | NORMAL

specifies whether the title characters are italic or normal.

Default: The default font style is specified by the FontStyle attribute of the GraphLabelText style element in the current style.

WEIGHT= BOLD | NORMAL

specifies whether the title characters are bold or normal.

Default: The default font weight is specified by the FontWeight attribute of the GraphLabelText style element in the current style.

Here is an example expression:

```
TITLEATTRS=(Color=Green Family=Arial Size=8 Style=Italic Weight=Bold)
```

Default: The default appearance of the title text is specified by the GraphLabelText style element in the current style.

VALUEATTRS= *style-element* <(options)> | (options)

specifies the appearance of the legend value labels. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the label characters. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphValueText style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

FAMILY= “font-family”

specifies the font family for the label characters. The SAS ODS styles use TrueType system fonts. For more information, see “TrueType Fonts Supplied by SAS” in Chapter 15 of *SAS Language Reference: Concepts*.

Default: The default font family is specified by the FontFamily attribute of the GraphValueText style element in the current style.

Restriction: You cannot specify a list of font family names.

SIZE= *n* <units>

specifies the font size of the label characters. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default font size is specified by the FontSize attribute of the GraphValueText style element in the current style.

STYLE= ITALIC | NORMAL

specifies whether the label characters are italic or normal.

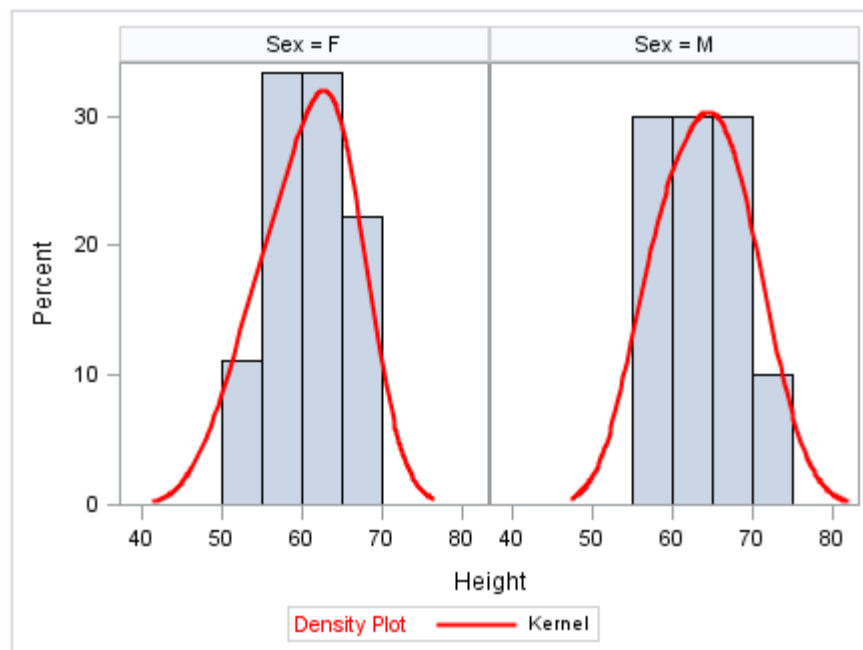
Default: The default font style is specified by the FontStyle attribute of the GraphValueText style element in the current style.

WEIGHT= BOLD | NORMAL

specifies whether the label characters are bold or normal.

Default: The default font weight is specified by the FontWeight attribute of the GraphValueText style element in the current style.

Example



```
proc sgpanel data=sashelp.class noautolegend;
  panelby sex;
  histogram height;
  density height /
    type=kernel
    name="kernel"
    lineattrs=(color = red);
```



```

keylegend "kernel" /
  title= "Density Plot"
  titleattrs=(color = red);
run;

```

LINEPARM Statement

Creates a straight line specified by a point and a slope. You can generate a single line by specifying a constant for each required argument. You can generate multiple lines by specifying a numeric variable for any or all required arguments.

Requirement: The statement must be used with another plot statement that is derived from data values that provide boundaries for the axis area. For example, the LINEPARM statement can be used with a scatter plot or a histogram.

Example: [“About Parameterized Lines” on page 24](#)

Syntax

```

LINEPARM X= numeric-value |numeric-variable
Y= numeric-value |numeric-variable
SLOPE= numeric-value |numeric-variable
</option(s)>;

```

Summary of Optional Arguments

Group options

NOMISSINGGROUP

specifies that missing values of the group variable are not included in the plot.

Label options

CURVELABEL <=“*text-string*”>

adds a label for the line.

CURVELABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the CURVELABEL= option.

CURVELABELPOS= MAX | MIN

specifies the location of the curve label.

Line options

CLIP

specifies that the data for the line is ignored when determining the data ranges for the axes.

LINEATTRS= *style-element* <(options)> | (options)

specifies the appearance of the line.

NOEXTEND

prevents the line from being extended beyond the axis offset.

Plot options

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set.

GROUP= *variable*

specifies a variable that is used to group the data.

LEGENDLABEL= *“text-string”*

specifies a label that identifies the plot in the legend.

NAME= *“text-string”*

specifies a name for the plot.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the lines.

Required Arguments

X= *numeric-value* | *numeric-variable*

specifies the X coordinate of a point.

Notes:

Values are in the units of the data.

If the value specified for the X= option is outside of the data range, then the data range is extended to include the specified point. This behavior can be changed with the CLIP= option.

If you specify a variable, and the variable contains any missing values, no line is drawn for the respective observation.

Y= *numeric-value* | *numeric-variable*

specifies the Y coordinate of a point.

Notes:

Values are in the units of the data.

If the value specified for the Y= option is outside of the data range, then the data range is extended to include the specified point. This behavior can be changed with the CLIP= option.

If you specify a variable, and the variable contains any missing values, no line is drawn for the respective observation.

SLOPE= *numeric-value* | *numeric-variable*

specifies the slope of the line. The slope can be positive or negative.

Tips:

SLOPE= 0 creates a horizontal line (parallel to the X axis).

SLOPE= . (missing value) creates a vertical line (parallel to the Y axis).

Optional Arguments

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph. For more information, see [“Using SG Attribute Maps to Control Visual Attributes” on page 605](#).

CLIP

specifies that the data for the line is ignored when determining the data ranges for the axes. Each axis scale is determined by the other plots in the overlay. This might result in the line not being displayed if its data range is not within the data ranges of the other plots.

CURVELABEL <=“*text-string*”>

adds a label for the line. You can also specify the label text. If you do not specify a label, then **SLOPE=***value* is used. If you specify a **GROUP** variable, the group value is shown instead of the slope.

CURVELABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the **CURVELABEL**= option. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

Here is an example that specifies a style element:

```
CURVELABELATTRS=GraphTitleText
```

options can be one or more of the following:

COLOR= *color*

specifies the color of the line label characters. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the **Color** attribute of the **GraphValueText** style element in the current style. For grouped data, the default color is specified by the **ContrastColor** attribute of the **GraphData1 ... GraphData*n*** style elements in the current style.

FAMILY= “*font-family*”

specifies the font family for the label characters. The SAS ODS styles use TrueType system fonts. For more information, see “TrueType Fonts Supplied by SAS” in Chapter 15 of *SAS Language Reference: Concepts*.

Default: The default font family is specified by the **FontFamily** attribute of the **GraphValueText** style element in the current style.

Restriction: You cannot specify a list of font family names.

SIZE= *n* <*units*>

specifies the font size of the label characters. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default font size is specified by the **FontSize** attribute of the **GraphValueText** style element in the current style.

STYLE= ITALIC | NORMAL

specifies whether the label characters are italic or normal.

Default: The default font style is specified by the **FontStyle** attribute of the **GraphValueText** style element in the current style.

WEIGHT= BOLD | NORMAL

specifies whether the label characters are bold or normal.

Default: The default font weight is specified by the **FontWeight** attribute of the **GraphValueText** style element in the current style.

Here is an example expression:

```
CURVELABELATTRS=(Color=Green Family=Arial Size=8 Style=Italic Weight=Bold)
```

Default: The default style element is **GraphValueText**.

Interaction: This option has no effect unless the **CURVELABEL** option is also specified.

CURVELABELPOS= MAX | MIN

specifies the location of the curve label. Specify one of the following values:

MIN

places the curve label at the minimum value for the X axis.

MAX

places the curve label at the maximum value for the X axis.

Default: MAX

Interaction: This option has no effect unless CURVELABEL is also specified.

GROUP= *variable*

specifies a variable that is used to group the data. A separate plot is created for each unique value of the grouping variable. The plot elements for each group value are automatically distinguished by different visual attributes.

LEGENDLABEL= “*text-string*”

specifies a label that identifies the plot in the legend. By default, the label “LineParm” is used (if you specify a numeric value for Y) or the Y variable name is used (if you specify a variable for Y).

Interaction: The LEGENDLABEL= option has no effect if you also specify the GROUP= option in the same plot statement.

LINEATTRS= *style-element* <(options)> | (options)

specifies the appearance of the line. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

Here is an example that specifies a style element: **LINEATTRS=GraphData3**

For more information about style elements, see [“Style Elements for Use with ODS Graphics” on page 640](#).

options can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphDataDefault style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see [“Line Patterns” on page 663](#).

Default: For ungrouped data, the default line pattern is specified by the LineStyle attribute of the GraphDataDefault style element in the current style. For grouped data, the default line pattern is specified by the LineStyle attribute of the GraphData1 ... GraphData*n* style elements in the current style.

THICKNESS= *n* <units>

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see [“Units of Measurement” on page 659](#).

Default: For ungrouped data, the default line thickness is specified by the LineThickness attribute of the GraphDataDefault style element in the current style. For grouped data, the default line thickness is specified by the

LineThickness attribute of the GraphData1 ... GraphData*n* style elements in the current style.

Here is an example expression:

```
LINEATTRS=(Color= "light green" Pattern= MediumDash Thickness= 4)
```

Default: GraphDataDefault is the default style element.

NAME= *“text-string”*

specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note: The name is case-sensitive. When you refer to the specified name in other statements, you must use the same exact capitalization.

NOEXTEND

prevents the line from being extended beyond the axis offset. When specified, there might be a 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.

NOMISSINGGROUP

specifies that missing values of the group variable are not included in the plot.

Interaction: This option has no effect unless GROUP= is also specified.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the lines. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default: 0.0

LOESS Statement

Creates a fitted loess curve.

Example: [“About Loess Plots” on page 34](#)

Syntax

LOESS X= *numeric-variable* Y= *numeric-variable* </ option(s)>;

Summary of Optional Arguments

Label options

CURVELABEL <= “*text-string*”>

adds a label for the curve.

CURVELABELPOS= MIN | MAX | START | END

specifies the location of the curve label.

DATALABEL <= *variable*>

displays a label for each data point.

Limit options

CLM <= “*text-string*”>

creates confidence limits.

CLMATTRS = *style-element* | (*CLMAttributeOptions*)

specifies the appearance of the mean value confidence limits by using an ODS style element or by specifying fill and line attributes.

CLMTRANSPARENCY= *numeric-value*

specifies the degree of transparency for the confidence limits.

LOESS options

DEGREE= 1 | 2

specifies the degree of the local polynomials to use for each local regression.

INTERPOLATION= CUBIC | LINEAR

specifies the degree of the interpolating polynomials that are used for blending local polynomial fits at the kd tree vertices.

LINEATTRS= *style-element* <(options)> | (options)

specifies the appearance of the fit curve.

MARKERATTRS= *style-element* <(options)> | (options)

specifies the appearance of the markers in the plot.

MAXPOINTS= *n*

specifies the maximum number of predicted points for the loess fit and the corresponding limits.

NOLEGCLM

hides the legend entry for the mean value confidence limits.

NOLEGFIT

hides the legend entry for the fit line.

NOMARKERS

removes the scatter markers from the plot.

REWEIGHT= *n*

specifies the number of iterative reweighting steps to apply to the data.

SMOOTH= *numeric-value*

specifies a smoothing parameter value.

Plot options

ALPHA= *numeric-value*

specifies the confidence level for the confidence limits.

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set.

GROUP= *variable*

specifies a classification variable to divide the values into groups.

LEGENDLABEL= "*text-string*"

specifies a label that identifies the fit line in the legend.

NAME= "*text-string*"

specifies a name for the plot.

WEIGHT= *numeric-variable*

specifies how observations are weighted.

Required Arguments

X= *numeric-variable*

specifies the variable for the x axis.

Y= *numeric-variable*

specifies the variable for the y axis.

Optional Arguments

ALPHA= *numeric-value*

specifies the confidence level for the confidence limits. Specify a number between 0.00 (100% confidence) and 1.00 (0% confidence).

Default: .05

Interaction: This option has no effect unless you also specify the CLM option.

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph. For more information, see [“Using SG Attribute Maps to Control Visual Attributes” on page 605](#).

CLM <= “*text-string*”>

creates confidence limits. The optional text string overrides the default legend label for the confidence limit.

CLMATTRS = *style-element* | (*CLMAttributeOptions*)

specifies the appearance of the mean value confidence limits by using an ODS style element or by specifying fill and line attributes. *CLMAttributeOptions* can be one or both of the following:

CLMFILLATTRS=*style-element* | (COLOR= *color*)

You can specify the color of the fill by using a style element or by using the COLOR= suboption. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

CLMLINEATTRS= *style-element* <(options)> | (options)

You can specify the appearance of the outline by using a style element or by using suboptions. If you specify a style element, then you can also specify suboptions to override specific appearance attributes.

The confidence limits have no outlines by default (for most styles). To make the outlines appear, you can change the DISPLAYOPTS in the GRAPHBAND style element. Then you can specify attributes for the outlines. The following code provides an example.

```
proc template;
  define style MyStyleDefault;
    parent=Styles.Default;
    style GraphBand from GraphBand /
      DisplayOpts = "Fill outline";
  end;
run;

ods listing style= MyStyleDefault;
proc sgplot data=sashelp.class;
  reg x=height y=weight / CLM
  CLMATTRS= (CLMLINEATTRS=
    (COLOR=Green PATTERN= ShortDash));
run;
```

The CLMLINEATTRS= *options* can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphConfidence style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see [“Line Patterns” on page 663](#).

Default: For ungrouped data, the default line pattern is specified by the LineStyle attribute of the GraphConfidence style element in the current style. For grouped data, the default line pattern is specified by the LineStyle attribute of the GraphData1 ... GraphData*n* style elements in the current style.

THICKNESS= *n* *<units>*

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see [“Units of Measurement” on page 659](#).

Default: For ungrouped data, the default line thickness is specified by the LineThickness attribute of the GraphConfidence style element in the current style. For grouped data, the default line thickness is specified by the LineThickness attribute of the GraphData1 ... GraphData*n* style elements in the current style.

Interaction: This suboption has no effect unless you change the display options in the style element to display outlines. See the preceding code example.

Default: The default appearance of the confidence limits is specified by the GraphConfidence style element in the current style.

Interaction: The CLMATTRS = option has no effect unless you also specify the CLM option.

CLMTRANSPARENCY= *numeric-value*

specifies the degree of transparency for the confidence limits. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default: 0.0

Interaction: This option takes effect only if you also specify the CLM option.

CURVELABEL *<=“text-string”>*

adds a label for the curve. You can also specify the label text. If you do not specify a label, the label from the Y variable is used.

CURVELABELPOS= MIN | MAX | START | END

specifies the location of the curve label. Specify one of the following values:

MIN

places the label at the part of the curve closest to the minimum X axis value.

MAX

places the label at the part of the curve closest to the maximum X axis value.

START

places the curve label at the first point on the curve.

END

places the curve label at the last point on the curve.

Default: END

Interaction: This option has no effect unless the CURVELABEL option is also specified.

DATALABEL *<= variable>*

displays a label for each data point. If you specify a variable, then the values of that variable are used for the data labels. If you do not specify a variable, then the values of the Y variable are used for the data labels.

Interaction: This option has no effect if you also specify the NOMARKERS option.

DEGREE = 1 | 2

specifies the degree of the local polynomials to use for each local regression. 1 specifies a linear fit and 2 specifies a quadratic fit.

Default: 1

GROUP = *variable*

specifies a classification variable to divide the values into groups. A separate plot is created for each unique value of the classification variable.

Interaction: If you specify the GROUP= option in multiple fit plot statements, then the first GROUP= variable is used for all of the fit plots that specify GROUP=.

INTERPOLATION = CUBIC | LINEAR

specifies the degree of the interpolating polynomials that are used for blending local polynomial fits at the kd tree vertices.

Default: CUBIC

LEGENDLABEL = *“text-string”*

specifies a label that identifies the fit line in the legend. By default, the label “Loess” is used, along with the value of the SMOOTH= option if specified.

LINEATTRS = *style-element* *<(options)>* | *(options)*

specifies the appearance of the fit curve. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR = *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphFit style element in the current style.

For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

PATTERN = *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see “[Line Patterns](#)” on page 663.

Default: For ungrouped data, the default line pattern is specified by the LineStyle attribute of the GraphFit style element in the current style.

For grouped data, the default line pattern is specified by the LineStyle attribute of the GraphData1 ... GraphData*n* style elements in the current style.

THICKNESS = *n* *<units>*

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see “[Units of Measurement](#)” on page 659.

Default: For ungrouped data, the default line thickness is specified by the LineThickness attribute of the GraphFit style element in the current style.

For grouped data, the default line thickness is specified by the LineThickness attribute of the GraphData1 ... GraphData*n* style elements in the current style.

MARKERATTRS= *style-element* <(options)> | (options)

specifies the appearance of the markers in the plot. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the markers. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphDataDefault style element in the current style.

For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

SIZE= *n* <*units*>

specifies the size of the markers. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default marker size is specified by the MarkerSize attribute of the GraphDataDefault style element in the current style.

SYMBOL= *symbol-name*

specifies the symbol for the markers. For a list of valid marker symbols, see “Marker Symbols” on page 661.

Default: For ungrouped data, the default marker symbol is specified by the MarkerSymbol attribute of the GraphDataDefault style element in the current style.

For grouped data, the default marker symbol is specified by the MarkerSymbol attribute of the GraphData1 ... GraphData*n* style elements in the current style.

MAXPOINTS= *n*

specifies the maximum number of predicted points for the loess fit and the corresponding limits.

Default: 201

NAME= “*text-string*”

specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note: The name is case-sensitive. When you refer to the specified name in other statements, you must use the same exact capitalization.

NOLEGCLM

hides the legend entry for the mean value confidence limits.

NOLEGFIT

hides the legend entry for the fit line.

NOMARKERS

removes the scatter markers from the plot.

REWEIGHT= *n*

specifies the number of iterative reweighting steps to apply to the data.

Default: 0

Interaction: This option has no affect if you do not specify the WEIGHT option.

SMOOTH= *numeric-value*

specifies a smoothing parameter value. If you do not specify this option, a smoothing value is determined automatically.

WEIGHT= *numeric-variable*

specifies how observations are weighted. Each observation is weighted by a factor of w for computational purposes, where w is the value of the numeric variable. w can be any numeric value. If w is 0, negative or missing, then that observation is excluded from the analysis.

Details

For the SMOOTH= option, the smoothing parameter value must be greater than the minimum value that is determined by the following equation.

$$\text{minimum} = \frac{\text{degree} + 1}{\text{number of observations}}$$

NEEDLE Statement

Creates a plot with needles connecting each point to the baseline.

Restriction: The vertical axis that is used with the NEEDLE statement cannot be a discrete axis.

Example: [“About Needle Plots” on page 26](#)

Syntax

NEEDLE X= *variable* Y= *numeric-variable* </ option(s)>;

Summary of Optional Arguments

Group options

CLUSTERWIDTH= *numeric-value*

specifies the width of the group clusters as a fraction of the midpoint spacing.

GROUP= *variable*

specifies a variable that is used to group the data.

GROUPDISPLAY= CLUSTER | OVERLAY

specifies how to display grouped needles.

GROUPORDER= ASCENDING | DESCENDING | DATA

specifies the ordering of needles within a group.

NOMISSINGGROUP

specifies that missing values of the group variable are not included in the plot.

Label options

DATALABEL <= *variable*>

displays a label for each data point.

DATALABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the DATALABEL= option.

Needle options**BASELINE=** *numeric-value*

specifies a value on the Y axis for the baseline.

DISCRETEOFFSET= *numeric-value*

specifies an amount to offset all needle lines from discrete X values.

LINEATTRS= *style-element* <(options)> | (options)

specifies the appearance of the needle lines.

MARKERATTRS= *style-element* <(options)> | (options)

specifies the appearance of the markers in the plot.

MARKERS

adds markers to the tips of the needles.

Plot options**ATTRID=** *variable*

specifies the value of the ID variable in an attribute map data set.

LEGENDLABEL= “*text-string*”

specifies a label that identifies the needle plot in the legend.

NAME= “*text-string*”

specifies a name for the plot.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the needle plot.

URL= *character-variable*

specifies a character variable that contains URLs for Web pages to be displayed when parts of the plot are selected within an HTML page.

Required Arguments**X=** *variable*

specifies the variable for the x axis.

Y= *numeric-variable*

specifies a numeric variable for the y axis.

Optional Arguments**ATTRID=** *variable*

specifies the value of the ID variable in an attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph. For more information, see “Using SG Attribute Maps to Control Visual Attributes” on page 605.

BASELINE= *numeric-value*

specifies a value on the Y axis for the baseline.

CLUSTERWIDTH= *numeric-value*

specifies the width of the group clusters as a fraction of the midpoint spacing. Specify a value from 0.0 (narrowest) to 1.0 (widest).

Default: 0.8**Interaction:** This option is applicable only when the GROUP option is specified, when GROUPDISPLAY=CLUSTER, and when the category axis is discrete.**DATALABEL** <= *variable*>

displays a label for each data point. If you specify a variable, the values of that variable are used for the data labels. If you do not specify a variable, then the values of the Y variable are used for the data labels.

DATALABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the DATALABEL= option. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

Here is an example that specifies a style element:

DATALABELATTRS=GraphLabelText

options can be one or more of the following:

COLOR= *color*

specifies the color of the label characters. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphDataText style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

FAMILY= “*font-family*”

specifies the font family for the label characters. The SAS ODS styles use TrueType system fonts. For more information, see “TrueType Fonts Supplied by SAS” in Chapter 15 of *SAS Language Reference: Concepts*.

Default: The default font family is specified by the FontFamily attribute of the GraphDataText style element in the current style.

SIZE= *n* <*units*>

specifies the font size of the label characters. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default font size is specified by the FontSize attribute of the GraphDataText style element in the current style.

STYLE= ITALIC | NORMAL

specifies whether the label characters are italic (ITALIC) or normal (NORMAL).

Default: The default font style is specified by the FontStyle attribute of the GraphDataText style element in the current style.

WEIGHT= BOLD | NORMAL

specifies whether the label characters are bold (BOLD) or normal (NORMAL).

Default: The default font weight is specified by the FontWeight attribute of the GraphDataText style element in the current style.

Here is an example expression:

DATALABELATTRS=(Color=Green Family=Arial Size=8 Style=Italic Weight=Bold)

Default: The default style element is GraphDataText.

Interaction: This option has no effect unless the DATALABEL option is also specified.

DISCRETEOFFSET= *numeric-value*

specifies an amount to offset all needle lines from discrete X values. Specify a value from -0.5 (left offset) to +0.5 (right offset).

Default: 0.0 (no offset)

Requirement: This option is applicable only when the X axis is discrete.

GROUP= *variable*

specifies a variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes.

GROUPDISPLAY= CLUSTER | OVERLAY

specifies how to display grouped needles.

CLUSTER

grouped items are drawn adjacent to each other.

OVERLAY

grouped items are drawn at the exact coordinate and might overlap. Each group is represented by unique visual attributes derived from the GraphData1... GraphData*n* style elements in the current style.

Default: OVERLAY

Restriction: GROUPDISPLAY=CLUSTER can take effect only when the category axis is discrete. If necessary, use a TYPE=DISCRETE option on the axis statement.

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

GROUPORDER= ASCENDING | DESCENDING | DATA

specifies the ordering of needles within a group.

ASCENDING

displays each group in ascending group value.

DESCENDING

displays each group in descending group value.

DATA

displays each group in data order of the group variable.

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

LEGENDLABEL= “*text-string*”

specifies a label that identifies the needle plot in the legend. By default, the label of the Y variable or the group value for each marker is used.

Interaction: The LEGENDLABEL= option has no effect if you also specify the GROUP= option in the same plot statement.

LINEATTRS= *style-element* <(options)> | (options)

specifies the appearance of the needle lines. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphDataDefault style element in the current style.

For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see “Line Patterns” on page 663.

Default: For ungrouped data, the default line pattern is specified by the `LineStyle` attribute of the `GraphDataDefault` style element in the current style.
For grouped data, the default line pattern is specified by the `LineStyle` attribute of the `GraphData1` ... `GraphDatan` style elements in the current style.

THICKNESS= *n* *<units>*

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see [“Units of Measurement” on page 659](#).

Default: For ungrouped data, the default line thickness is specified by the `LineThickness` attribute of the `GraphDataDefault` style element in the current style.

For grouped data, the default line thickness is specified by the `LineThickness` attribute of the `GraphData1` ... `GraphDatan` style elements in the current style.

MARKERATTRS= *style-element* *<(options)>* | *(options)*

specifies the appearance of the markers in the plot. This option has no effect unless you also specify the `MARKERS` option. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the markers. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see [“Color-Naming Schemes” in Chapter 14 of SAS/GRAPH: Reference](#).

Default: For ungrouped data, the default color is specified by the `ContrastColor` attribute of the `GraphDataDefault` style element in the current style.

For grouped data, the default color is specified by the `ContrastColor` attribute of the `GraphData1` ... `GraphDatan` style elements in the current style.

SIZE= *n* *<units>*

specifies the size of the markers. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see [“Units of Measurement” on page 659](#).

Default: The default marker size is specified by the `MarkerSize` attribute of the `GraphDataDefault` style element in the current style.

SYMBOL= *symbol-name*

specifies the symbol for the markers. For a list of valid marker symbols, see [“Marker Symbols” on page 661](#).

Default: For ungrouped data, the default marker symbol is specified by the `MarkerSymbol` attribute of the `GraphDataDefault` style element in the current style.

For grouped data, the default marker symbol is specified by the `MarkerSymbol` attribute of the `GraphData1` ... `GraphDatan` style elements in the current style.

MARKERS

adds markers to the tips of the needles.

NAME= *“text-string”*

specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note: The name is case-sensitive. When you refer to the specified name in other statements, you must use the same exact capitalization.

NOMISSINGGROUP

specifies that missing values of the group variable are not included in the plot.

Interaction: This option has no effect unless GROUP= is also specified.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the needle plot. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default: 0.0

URL= *character-variable*

specifies a character variable that contains URLs for Web pages to be displayed when parts of the plot are selected within an HTML page.

Default: By default, no HTML links are created.

Interactions:

This option affects graphics output that is created through the ODS HTML destination only. For more information about ODS destinations, see [“Specifying a Destination” on page 649](#).

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For more information, see [“Using the ODS GRAPHICS Statement” on page 651](#).

PBSPLINE Statement

Creates a fitted penalized B-spline curve.

Example: [“About Penalized B-Spline Plots” on page 35](#)

Syntax

PBSPLINE *x= numeric-variable y= numeric-variable* </ options>;

Summary of Optional Arguments

Label options

CURVELABEL <= “*text-string*”>

adds a label for the curve.

CURVELABELPOS= MIN | MAX | START | END

specifies the location of the curve label.

DATALABEL <= *variable*>

displays a label for each data point.

Limit options

CLI <= “*text-string*”>

creates prediction limits for the individual predicted values.

CLIATTRS = *style-element* | (CLILINEATTRS= *style-element* <(options)> | (options))

specifies the appearance of the individual value prediction limits by using an ODS style element or by specifying attributes for the outlines.

CLM <= “*text-string*”>

creates confidence limits.

CLMATTRS = *style-element* | (CLMAttributeOptions)

specifies the appearance of the mean value confidence limits by using an ODS style element or by specifying fill and line attributes.

CLMTRANSPARENCY= *numeric-value*

specifies the degree of transparency for the confidence limits.

PBSPLINE options

DEGREE= *n*

specifies the degree of the spline transformation.

LINEATTRS= *style-element* <(options)> | (options)

specifies the appearance of the fitted curve.

MARKERATTRS= *style-element* <(options)> | (options)

specifies the appearance of the markers in the plot.

MAXPOINTS= *n*

specifies the maximum number of predicted points for the spline curve and for any confidence limits.

NKNOTS= *n*

specifies the number of evenly spaced internal knots.

NOLEGCLI

hides the legend entry for the individual value prediction limits.

NOLEGCLM

hides the legend entry for the mean value confidence limits.

NOLEGFIT

hides the legend entry for the fit line.

NOMARKERS

removes the scatter markers from the plot.

SMOOTH= *numeric-value*

specifies a smoothing parameter value.

Plot options

ALPHA= *numeric-value*

specifies the confidence level for the confidence limits.

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set.

FREQ= *numeric-variable*

specifies how many times observations are repeated for computational purposes.

GROUP= *variable*

specifies a classification variable to divide the values into groups.

LEGENDLABEL= *"text-string"*

specifies a label that identifies the B-spline curve in the legend.

NAME= *"text-string"*

specifies a name for the plot.

WEIGHT= *numeric-variable*

specifies how observations are weighted.

Required Arguments

X= *numeric-variable*

specifies the variable for the x axis.

Y= *numeric-variable*

specifies the variable for the y axis.

Optional Arguments

ALPHA= *numeric-value*

specifies the confidence level for the confidence limits. Specify a number between 0.00 (100% confidence) and 1.00 (0% confidence).

Default: .05

Interaction: This option has no effect unless you specify the CLI or CLM option (or both).

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph. For more information, see [“Using SG Attribute Maps to Control Visual Attributes” on page 605](#).

CLI <= “*text-string*”>

creates prediction limits for the individual predicted values. The optional text string overrides the default legend label for the prediction limits.

CLIATTRS = *style-element* | (CLILINEATTRS= *style-element* <(options)> | (options))

specifies the appearance of the individual value prediction limits by using an ODS style element or by specifying attributes for the outlines. The outline attributes can be the following:

CLILINEATTRS= *style-element* <(options)> | (options)

You can specify the appearance of the outline by using a style element or by using suboptions. If you specify a style element, then you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see [“Color-Naming Schemes” in Chapter 14 of SAS/GRAPH: Reference](#).

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphPredictionLimits style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see [“Line Patterns” on page 663](#).

Default: For ungrouped data, the default line pattern is specified by the LineStyle attribute of the GraphPredictionLimits style element in the current style. For grouped data, the default line pattern is specified by the LineStyle attribute of the GraphData1 ... GraphData*n* style elements in the current style.

THICKNESS= *n* <*units*>

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see [“Units of Measurement” on page 659](#).

Default: For ungrouped data, the default line thickness is specified by the LineThickness attribute of the GraphPredictionLimits style element in the current style. For grouped data, the default line thickness is specified by

the LineThickness attribute of the GraphData1 ... GraphData*n* style elements in the current style.

Default: The default style element is GraphPredictionLimits.

Interaction: The CLIATTRS = option has no effect unless you also specify the CLI option.

CLM <= "*text-string*">

creates confidence limits. The optional text string overrides the default legend label for the confidence limit.

CLMATTRS = *style-element* | (*CLMAttributeOptions*)

specifies the appearance of the mean value confidence limits by using an ODS style element or by specifying fill and line attributes. *CLMAttributeOptions* can be one or both of the following:

CLMFILLATTRS=*style-element* | (COLOR= *color*)

You can specify the color of the fill by using a style element or by using the COLOR= suboption. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see "Color-Naming Schemes" in Chapter 14 of *SAS/GRAPH: Reference*.

CLMLINEATTRS= *style-element* <(options)> | (options)

You can specify the appearance of the outline by using a style element or by using suboptions. If you specify a style element, then you can also specify suboptions to override specific appearance attributes.

The confidence limits have no outlines by default (for most styles). To make the outlines appear, you can change the DISPLAYOPTS in the GRAPHBAND style element. Then you can specify attributes for the outlines. The following code provides an example.

```
proc template;
  define style MyStyleDefault;
    parent=Styles.Default;
    style GraphBand from GraphBand /
      DisplayOpts = "Fill outline";
  end;
run;

ods listing style= MyStyleDefault;
proc sgplot data=sashelp.class;
  reg x=height y=weight / CLM
  CLMATTRS=(CLMLINEATTRS=
    (COLOR=Green PATTERN= ShortDash));
run;
```

The CLMLINEATTRS= *options* can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see "Color-Naming Schemes" in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphConfidence style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see [“Line Patterns” on page 663](#).

Default: For ungrouped data, the default line pattern is specified by the `LineStyle` attribute of the `GraphConfidence` style element in the current style. For grouped data, the default line pattern is specified by the `LineStyle` attribute of the `GraphData1 ... GraphDatan` style elements in the current style.

THICKNESS= *n* *<units>*

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see [“Units of Measurement” on page 659](#).

Default: For ungrouped data, the default line thickness is specified by the `LineThickness` attribute of the `GraphConfidence` style element in the current style. For grouped data, the default line thickness is specified by the `LineThickness` attribute of the `GraphData1 ... GraphDatan` style elements in the current style.

Interaction: This suboption has no effect unless you change the display options in the style element to display outlines. See the preceding code example.

Default: The default appearance of the confidence limits is specified by the `GraphConfidence` style element in the current style.

Interaction: The `CLMATTRS =` option has no effect unless you also specify the `CLM` option.

CLMTRANSPARENCY= *numeric-value*

specifies the degree of transparency for the confidence limits. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default: 0.0

Interaction: This option takes effect only if you also specify the `CLM` option.

CURVELABEL *<=“text-string”>*

adds a label for the curve. You can also specify the label text. If you do not specify a label, the label from the Y variable is used.

CURVELABELPOS= `MIN` | `MAX` | `START` | `END`

specifies the location of the curve label. Specify one of the following values:

`MIN`

places the label at the part of the curve closest to the minimum X axis value.

`MAX`

places the label at the part of the curve closest to the maximum X axis value.

`START`

places the curve label at the first point on the curve.

`END`

places the curve label at the last point on the curve.

Default: `END`

Interaction: This option has no effect unless the `CURVELABEL` option is also specified.

DATALABEL <= *variable*>

displays a label for each data point. If you specify a variable, then the values of that variable are used for the data labels. If you do not specify a variable, then the values of the Y variable are used for the data labels.

Interaction: This option has no effect if you also specify the NOMARKERS option.

DEGREE= *n*

specifies the degree of the spline transformation.

Default: 3

FREQ= *numeric-variable*

specifies how many times observations are repeated for computational purposes. Each observation is repeated *n* times for computational purposes, where *n* is the value of the numeric variable. If *n* is not an integer, then it is truncated to an integer. If *n* is less than 1 or missing, then it is excluded from the analysis.

GROUP= *variable*

specifies a classification variable to divide the values into groups. A separate plot is created for each unique value of the classification variable.

Interaction: If you specify the GROUP= option in multiple fit plot statements, then the first GROUP= variable is used for all of the fit plots that specify GROUP=.

LEGENDLABEL= "*text-string*"

specifies a label that identifies the B-spline curve in the legend. By default, the label "Penalized B-Spline" is used with the SMOOTH= value if specified, or else the group value for each B-spline is used.

LINEATTRS= *style-element* <(options)> | (options)

specifies the appearance of the fitted curve. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see "Color-Naming Schemes" in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphFit style element in the current style.

For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see "[Line Patterns](#)" on page 663.

Default: For ungrouped data, the default line pattern is specified by the LineStyle attribute of the GraphFit style element in the current style.

For grouped data, the default line pattern is specified by the LineStyle attribute of the GraphData1 ... GraphData*n* style elements in the current style.

THICKNESS= *n* <*units*>

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see "[Units of Measurement](#)" on page 659.

Default: For ungrouped data, the default line thickness is specified by the LineThickness attribute of the GraphFit style element in the current style.

For grouped data, the default line thickness is specified by the LineThickness attribute of the GraphData1 ... GraphData*n* style elements in the current style.

MARKERATTRS= *style-element* <(options)> | (options)

specifies the appearance of the markers in the plot. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the markers. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphDataDefault style element in the current style.

For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

SIZE= *n* <*units*>

specifies the size of the markers. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default marker size is specified by the MarkerSize attribute of the GraphDataDefault style element in the current style.

SYMBOL= *symbol-name*

specifies the symbol for the markers. For a list of valid marker symbols, see “Marker Symbols” on page 661.

Default: For ungrouped data, the default marker symbol is specified by the MarkerSymbol attribute of the GraphDataDefault style element in the current style.

For grouped data, the default marker symbol is specified by the MarkerSymbol attribute of the GraphData1 ... GraphData*n* style elements in the current style.

MAXPOINTS= *n*

specifies the maximum number of predicted points for the spline curve and for any confidence limits.

Default: 201

NAME= “*text-string*”

specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note: The name is case-sensitive. When you refer to the specified name in other statements, you must use the same exact capitalization.

NKNOTS= *n*

specifies the number of evenly spaced internal knots.

Default: 100

NOLEGCLI

hides the legend entry for the individual value prediction limits.

NOLEGCLM

hides the legend entry for the mean value confidence limits.

NOLEGFIT

hides the legend entry for the fit line.

NOMARKERS

removes the scatter markers from the plot.

SMOOTH= *numeric-value*

specifies a smoothing parameter value. If you do not specify this option, a smoothing value is determined automatically.

WEIGHT= *numeric-variable*

specifies how observations are weighted. Each observation is weighted by a factor of w for computational purposes, where w is the value of the numeric variable. w can be any numeric value. If w is 0, negative or missing, then that observation is excluded from the analysis.

Details

For the SMOOTH= option, the smoothing parameter value must be greater than the minimum value that is determined by the following equation.

$$\text{minimum} = \frac{\text{degree} + 1}{\text{number of observations}}$$

REFLINE Statement

Creates a horizontal or vertical reference line.

Example: [“About Reference Lines” on page 23](#)

Syntax

REFLINE *variable* | *value-1* <... *value-n*> </ *option(s)*>;

Summary of Optional Arguments**Line options**

AXIS= X | Y

specifies the axis that contains the reference line values.

DISCRETEOFFSET= *numeric-value*

specifies an amount to offset all lines from discrete X or Y values.

LABEL <= (“*text-string-1*” ... “*text-string-n*”)>

creates labels for each reference line.

LABELPOS= MIN | MAX

specifies the position of the labels.

LINEATTRS= *style-element* <(options)> | (options)

specifies the appearance of the reference line.

NOCLIP

extends the plot axes to contain the reference lines.

Plot options

LEGENDLABEL= “*text-string*”

specifies a label that identifies the markers from the plot in the legend.

NAME= “*text-string*”

specifies a name for the plot.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the reference lines.

Required Arguments

variable

draws a reference line for each value of the specified variable.

value-1 <... value-n>

draws one or more reference lines at the values that you specify.

Optional Arguments

AXIS= X | Y

specifies the axis that contains the reference line values. For example, if you specify **AXIS= X**, vertical reference lines are drawn at points on the X axis.

Default: Y

DISCRETEOFFSET= *numeric-value*

specifies an amount to offset all lines from discrete X or Y values. Specify a value from -0.5 (left offset) to +0.5 (right offset).

Default: 0.0 (no offset)

Requirement: This option is applicable only when the X or Y axis is discrete.

LABEL <= (“*text-string-1*” ... “*text-string-n*”)>

creates labels for each reference line. If you do not specify a label for a line, the reference value for that line is used as the label.

LABELPOS= MIN | MAX

specifies the position of the labels. MIN specifies the label is placed at the minimum value of the data axis. MAX specifies that the label is placed at the maximum value of the data axis.

LEGENDLABEL= “*text-string*”

specifies a label that identifies the markers from the plot in the legend. By default, the label “Reference Line” is used.

Interaction: This option has no effect unless you also specify the **NAME=** option.

LINEATTRS= *style-element* <(options)> | (options)

specifies the appearance of the reference line. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: The default color is specified by the ContrastColor attribute of the GraphReference style element in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see “Line Patterns” on page 663.

Default: The default line pattern is specified by the LinePattern attribute of the GraphReference style element in the current style.

THICKNESS= *n* *<units>*

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see [“Units of Measurement” on page 659](#).

Default: The default line thickness is specified by the LineThickness attribute of the GraphReference style element in the current style.

NAME= *“text-string”*

specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note: The name is case-sensitive. When you refer to the specified name in other statements, you must use the same exact capitalization.

NOCLIP

extends the plot axes to contain the reference lines. By default, if a reference line is created outside of the data range, then the reference line is not visible. This option has no effect if you do not create reference lines that are outside of the data range.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the reference lines. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default: 0.0

REG Statement

Creates a fitted regression line or curve.

Interaction: A linear regression (DEGREE=1) cannot be used with logarithmic axes.

Examples: [“About Regression Plots” on page 36](#)
[“Example 2: Creating a Panel of Regression Curves” on page 269](#)

Syntax

REG *X= numeric-variable Y= numeric-variable* *</ option(s)>*;

Summary of Optional Arguments

Label options

CURVELABEL *<=“text-string”>*

adds a label for the curve.

CURVELABELPOS= MIN | MAX | START | END

specifies the location of the curve label.

DATALABEL *<= variable>*

displays a label for each data point.

Limit options

CLI *<= “text-string”>*

creates prediction limits for the individual predicted values.

CLIATTRS = *style-element* | (CLILINEATTRS= *style-element* *<(options)>* | *(options)*)

specifies the appearance of the individual value prediction limits by using an ODS style element or by specifying attributes for the outlines.

CLM <= *“text-string”*>

creates confidence limits.

CLMATTRS = *style-element* | (*CLMAttributeOptions*)

specifies the appearance of the mean value confidence limits by using an ODS style element or by specifying fill and line attributes.

CLMTRANSPARENCY = *numeric-value*

specifies the degree of transparency for the confidence limits.

Plot options

ALPHA = *numeric-value*

specifies the confidence level for the confidence limits.

ATTRID = *variable*

specifies the value of the ID variable in an attribute map data set.

FREQ = *numeric-variable*

specifies how many times observations are repeated for computational purposes.

GROUP = *variable*

specifies a classification variable to divide the values into groups.

LEGENDLABEL = *“text-string”*

specifies a label that identifies the regression curve in the legend.

NAME = *“text-string”*

specifies a name for the plot.

WEIGHT = *numeric-variable*

specifies how observations are weighted.

REG options

DEGREE = *n*

specifies the degree of the polynomial fit.

LINEATTRS = *style-element* <(options)> | (options)

specifies the appearance of the fit line.

MARKERATTRS = *style-element* <(options)> | (options)

specifies the appearance of the markers in the plot.

MAXPOINTS = *n*

specifies the maximum number of predicted points for the regression curve and for any confidence limits.

NOLEGCLI

hides the legend entry for the individual predicted value confidence limits.

NOLEGCLM

hides the legend entry for the mean value confidence limits.

NOLEGFIT

hides the legend entry for the fit line.

NOMARKERS

removes the scatter markers from the plot.

Required Arguments

X = *numeric-variable*

specifies the variable for the x axis.

Y = *numeric-variable*

specifies the variable for the y axis.

Optional Arguments

ALPHA= *numeric-value*

specifies the confidence level for the confidence limits. Specify a number between 0.00 (100% confidence) and 1.00 (0% confidence).

Default: .05

Interaction: This option has no effect unless you specify the CLI or CLM option (or both).

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph. For more information, see [“Using SG Attribute Maps to Control Visual Attributes” on page 605](#).

CLI <= “*text-string*”>

creates prediction limits for the individual predicted values. The optional text string overrides the default legend label for the prediction limits.

CLIATTRS = *style-element* | (CLILINEATTRS= *style-element* <(options)> | (options))

specifies the appearance of the individual value prediction limits by using an ODS style element or by specifying attributes for the outlines. The outline attributes can be the following:

CLILINEATTRS= *style-element* <(options)> | (options)

You can specify the appearance of the outline by using a style element or by using suboptions. If you specify a style element, then you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see [“Color-Naming Schemes” in Chapter 14 of SAS/GRAPH: Reference](#).

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphPredictionLimits style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see [“Line Patterns” on page 663](#).

Default: For ungrouped data, the default line pattern is specified by the LineStyle attribute of the GraphPredictionLimits style element in the current style. For grouped data, the default line pattern is specified by the LineStyle attribute of the GraphData1 ... GraphData*n* style elements in the current style.

THICKNESS= *n* <*units*>

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see [“Units of Measurement” on page 659](#).

Default: For ungrouped data, the default line thickness is specified by the LineThickness attribute of the GraphPredictionLimits style element in the current style. For grouped data, the default line thickness is specified by

the LineThickness attribute of the GraphData1 ... GraphData*n* style elements in the current style.

Default: The default style element is GraphPredictionLimits.

Interaction: The CLIATTRS = option has no effect unless you also specify the CLI option.

CLM <= "*text-string*">

creates confidence limits. The optional text string overrides the default legend label for the confidence limit.

CLMATTRS = *style-element* | (*CLMAttributeOptions*)

specifies the appearance of the mean value confidence limits by using an ODS style element or by specifying fill and line attributes. *CLMAttributeOptions* can be one or both of the following:

CLMFILLATTRS=*style-element* | (COLOR= *color*)

You can specify the color of the fill by using a style element or by using the COLOR= suboption. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see "Color-Naming Schemes" in Chapter 14 of *SAS/GRAPH: Reference*.

CLMLINEATTRS=*style-element* <(options)> | (options)

You can specify the appearance of the outline by using a style element or by using suboptions. If you specify a style element, then you can also specify suboptions to override specific appearance attributes.

The confidence limits have no outlines by default (for most styles). To make the outlines appear, you can change the DISPLAYOPTS in the GRAPHBAND style element. Then you can specify attributes for the outlines. The following code provides an example.

```
proc template;
  define style MyStyleDefault;
    parent=Styles.Default;
    style GraphBand from GraphBand /
      DisplayOpts = "Fill outline";
  end;
run;

ods listing style= MyStyleDefault;
proc sgplot data=sashelp.class;
  reg x=height y=weight / CLM
  CLMATTRS=(CLMLINEATTRS=
    (COLOR=Green PATTERN= ShortDash));
run;
```

The CLMLINEATTRS= *options* can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see "Color-Naming Schemes" in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphConfidence style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see [“Line Patterns” on page 663](#).

Default: For ungrouped data, the default line pattern is specified by the `LineStyle` attribute of the `GraphConfidence` style element in the current style. For grouped data, the default line pattern is specified by the `LineStyle` attribute of the `GraphData1 ... GraphDatan` style elements in the current style.

THICKNESS= *n <units>*

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see [“Units of Measurement” on page 659](#).

Default: For ungrouped data, the default line thickness is specified by the `LineThickness` attribute of the `GraphConfidence` style element in the current style. For grouped data, the default line thickness is specified by the `LineThickness` attribute of the `GraphData1 ... GraphDatan` style elements in the current style.

Interaction: This suboption has no effect unless you change the display options in the style element to display outlines. See the preceding code example.

Default: The default appearance of the confidence limits is specified by the `GraphConfidence` style element in the current style.

Interaction: The `CLMATTRS =` option has no effect unless you also specify the `CLM` option.

CLMTRANSPARENCY= *numeric-value*

specifies the degree of transparency for the confidence limits. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default: 0.0

Interaction: This option takes effect only if you also specify the `CLM` option.

CURVELABEL <=“*text-string*”>

adds a label for the curve. You can also specify the label text. If you do not specify a label, the label from the Y variable is used.

CURVELABELPOS= MIN | MAX | START | END

specifies the location of the curve label. Specify one of the following values:

MIN

places the label at the part of the curve closest to the minimum X axis value.

MAX

places the label at the part of the curve closest to the maximum X axis value.

START

places the curve label at the first point on the curve.

END

places the curve label at the last point on the curve.

Default: END

Interaction: This option has no effect unless the `CURVELABEL` option is also specified.

DATALABEL <= *variable*>

displays a label for each data point. If you specify a variable, then the values of that variable are used for the data labels. If you do not specify a variable, then the values of the Y variable are used for the data labels.

Interaction: This option has no effect if you also specify the NOMARKERS option.

DEGREE= *n*

specifies the degree of the polynomial fit. For example, 1 specifies a linear fit, 2 specifies a quadratic fit, and 3 specifies a cubic fit.

Default: 1

FREQ= *numeric-variable*

specifies how many times observations are repeated for computational purposes. Each observation is repeated *n* times for computational purposes, where *n* is the value of the numeric variable. If *n* is not an integer, then it is truncated to an integer. If *n* is less than 1 or missing, then it is excluded from the analysis.

GROUP= *variable*

specifies a classification variable to divide the values into groups. A separate plot is created for each unique value of the classification variable.

Interaction: If you specify the GROUP= option in multiple fit plot statements, then the first GROUP= variable is used for all of the fit plots that specify GROUP=.

LEGENDLABEL= "*text-string*"

specifies a label that identifies the regression curve in the legend. By default, the label "Regression" is used.

Interaction: The LEGENDLABEL= option has no effect if you also specify the GROUP= option in the same plot statement.

LINEATTRS= *style-element* <(options)> | (options)

specifies the appearance of the fit line. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see "Color-Naming Schemes" in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphFit style element in the current style.

For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see "[Line Patterns](#)" on page 663.

Default: For ungrouped data, the default line pattern is specified by the LineStyle attribute of the GraphFit style element in the current style.

For grouped data, the default line pattern is specified by the LineStyle attribute of the GraphData1 ... GraphData*n* style elements in the current style.

THICKNESS= *n* <*units*>

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see "[Units of Measurement](#)" on page 659.

Default: For ungrouped data, the default line thickness is specified by the LineThickness attribute of the GraphFit style element in the current style.

For grouped data, the default line thickness is specified by the LineThickness attribute of the GraphData1 ... GraphData*n* style elements in the current style.

MARKERATTRS= *style-element* <(options)> | (options)

specifies the appearance of the markers in the plot. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the markers. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphDataDefault style element in the current style.

For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

SIZE= *n* <units>

specifies the size of the markers. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default marker size is specified by the MarkerSize attribute of the GraphDataDefault style element in the current style.

SYMBOL= *symbol-name*

specifies the symbol for the markers. For a list of valid marker symbols, see “Marker Symbols” on page 661.

Default: For ungrouped data, the default marker symbol is specified by the MarkerSymbol attribute of the GraphDataDefault style element in the current style.

For grouped data, the default marker symbol is specified by the MarkerSymbol attribute of the GraphData1 ... GraphData*n* style elements in the current style.

MAXPOINTS= *n*

specifies the maximum number of predicted points for the regression curve and for any confidence limits.

Default: 10

NAME= “*text-string*”

specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note: The name is case-sensitive. When you refer to the specified name in other statements, you must use the same exact capitalization.

NOLEGCLI

hides the legend entry for the individual predicted value confidence limits.

NOLEGCLM

hides the legend entry for the mean value confidence limits.

NOLEGFIT

hides the legend entry for the fit line.

NOMARKERS

removes the scatter markers from the plot.

WEIGHT= *numeric-variable*

specifies how observations are weighted. Each observation is weighted by a factor of w for computational purposes, where w is the value of the numeric variable. w can be any numeric value. If w is 0, negative or missing, then that observation is excluded from the analysis.

SCATTER Statement

Creates a scatter plot.

Example: [“About Scatter Plots” on page 27](#)

Syntax

SCATTER X= *variable* Y= *variable* < / option(s)>;

Summary of Optional Arguments

Group options

CLUSTERWIDTH= *numeric-value*

specifies the width of the group clusters as a fraction of the midpoint spacing.

GROUP= *variable*

specifies a variable that is used to group the data.

GROUPDISPLAY= CLUSTER | OVERLAY

specifies how to display grouped markers.

GROUPORDER= ASCENDING | DESCENDING | DATA

specifies the ordering of markers within a group.

NOMISSINGGROUP

specifies that missing values of the group variable are not included in the plot.

Label options

DATALABEL <= *variable*>

displays a label for each data point.

DATALABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the DATALABEL= option.

Plot options

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set.

FREQ= *numeric-variable*

specifies how many times observations are repeated for computational purposes.

LEGENDLABEL= “*text-string*”

specifies a label that identifies the markers from the plot in the legend.

NAME= “*text-string*”

specifies a name for the plot.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the markers and error bars.

URL= *character-variable*

specifies a character variable that contains URLs for Web pages to be displayed when parts of the plot are selected within an HTML page.

Scatter options

DISCRETEOFFSET= *numeric-value*

specifies an amount to offset all markers from discrete X or Y values.

ERRORBARATTRS= *style-element* <(options)> | (options)

specifies the appearance of the error bars in the plot.

MARKERATTRS= *style-element* <(options)> | (options)

specifies the appearance of the markers in the plot.

MARKERCHAR= *variable*

specifies a variable whose values replace the marker symbols in the plot.

MARKERCHARATTRS= *style-element* <(options)> | (options)

specifies the appearance of the markers in the plot when you use the **MARKERCHAR=** option.

XERRORLOWER= *numeric-variable*

specifies a variable that contains the lower endpoints for the X error bars.

XERRORUPPER= *numeric-variable*

specifies a variable that contains the upper endpoints for the X error bars.

YERRORLOWER= *numeric-variable*

specifies a variable that contains the lower endpoints for the Y error bars.

YERRORUPPER= *numeric-variable*

specifies a variable that contains the upper endpoints for the Y error bars.

Required Arguments

X= *variable*

specifies the variable for the x axis.

Y= *variable*

specifies the variable for the y axis.

Optional Arguments

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph. For more information, see [“Using SG Attribute Maps to Control Visual Attributes” on page 605](#).

CLUSTERWIDTH= *numeric-value*

specifies the width of the group clusters as a fraction of the midpoint spacing. Specify a value from 0.0 (narrowest) to 1.0 (widest).

Default: 0.8

Interaction: This option is applicable only when the **GROUP** option is specified, when **GROUPDISPLAY=CLUSTER**, and when the category axis is discrete.

DATALABEL <=*variable*>

displays a label for each data point. If you specify a variable, the values of that variable are used for the data labels. If you do not specify a variable, then the values of the Y variable are used for the data labels.

DATALABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the DATALABEL= option. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

Here is an example that specifies a style element:

DATALABELATTRS=GraphLabelText

options can be one or more of the following:

COLOR= *color*

specifies the color of the label characters. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphDataText style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

FAMILY= “*font-family*”

specifies the font family for the label characters. The SAS ODS styles use TrueType system fonts. For more information, see “TrueType Fonts Supplied by SAS” in Chapter 15 of *SAS Language Reference: Concepts*.

Default: The default font family is specified by the FontFamily attribute of the GraphDataText style element in the current style.

SIZE= *n* <*units*>

specifies the font size of the label characters. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default font size is specified by the FontSize attribute of the GraphDataText style element in the current style.

STYLE= ITALIC | NORMAL

specifies whether the label characters are italic (ITALIC) or normal (NORMAL).

Default: The default font style is specified by the FontStyle attribute of the GraphDataText style element in the current style.

WEIGHT= BOLD | NORMAL

specifies whether the label characters are bold (BOLD) or normal (NORMAL).

Default: The default font weight is specified by the FontWeight attribute of the GraphDataText style element in the current style.

Here is an example expression:

DATALABELATTRS=(Color=Green Family=Arial Size=8 Style=Italic Weight=Bold)

Default: The default style element is GraphDataText.

Interaction: This option has no effect unless the DATALABEL option is also specified.

DISCRETEOFFSET= *numeric-value*

specifies an amount to offset all markers from discrete X or Y values. Specify a value from -0.5 (left offset) to +0.5 (right offset).

Default: 0.0 (no offset)

Requirement: This option is applicable only when the X or Y axis is discrete.

ERRORBARATTRS= *style-element* <(options)> | (options)

specifies the appearance of the error bars in the plot. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: The default color is specified by the ContrastColor attribute of the GraphError style element in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see “Line Patterns” on page 663.

Default: The default line pattern is specified by the LineStyle attribute of the GraphError style element in the current style.

THICKNESS= *n* <units>

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default line thickness is specified by the LineThickness attribute of the GraphError style element in the current style.

FREQ= *numeric-variable*

specifies how many times observations are repeated for computational purposes. Each observation is repeated *n* times for computational purposes, where *n* is the value of the numeric variable. If *n* is not an integer, then it is truncated to an integer. If *n* is less than 1 or missing, then it is excluded from the analysis.

GROUP= *variable*

specifies a variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes.

GROUPDISPLAY= CLUSTER | OVERLAY

specifies how to display grouped markers.

CLUSTER

grouped items are drawn adjacent to each other.

OVERLAY

grouped items are drawn at the exact coordinate and might overlap. Each group is represented by unique visual attributes derived from the GraphData1... GraphData*n* style elements in the current style.

Default: OVERLAY

Restriction: GROUPDISPLAY=CLUSTER can take effect only when the category axis is discrete. If necessary, use a TYPE=DISCRETE option on the axis statement.

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

GROUPORDER= ASCENDING | DESCENDING | DATA

specifies the ordering of markers within a group.

ASCENDING

displays each group in ascending group value.

DESCENDING

displays each group in descending group value.

DATA

displays each group in data order of the group variable.

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

LEGENDLABEL= “text-string”

specifies a label that identifies the markers from the plot in the legend. By default, the label of the Y variable or the group value for each marker is used.

Interaction: The LEGENDLABEL= option has no effect if you also specify the GROUP= option in the same plot statement.

MARKERATTRS= style-element <(options)> | (options)

specifies the appearance of the markers in the plot. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= color

specifies the color of the markers. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphDataDefault style element in the current style.

For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

SIZE= n <units>

specifies the size of the markers. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default marker size is specified by the MarkerSize attribute of the GraphDataDefault style element in the current style.

SYMBOL= symbol-name

specifies the symbol for the markers. For a list of valid marker symbols, see “Marker Symbols” on page 661.

Default: For ungrouped data, the default marker symbol is specified by the MarkerSymbol attribute of the GraphDataDefault style element in the current style.

For grouped data, the default marker symbol is specified by the MarkerSymbol attribute of the GraphData1 ... GraphData*n* style elements in the current style.

MARKERCHAR= variable

specifies a variable whose values replace the marker symbols in the plot. If a format is associated with the variable, then the formatted values are used as the marker symbols. If there is not a format associated with the variable and the variable contains numeric data, then the BEST6. format is used.

Interaction: The MARKERCHAR= option overrides the DATALABEL= option and the SYMBOL= suboption of the MARKERATTRS= option.

MARKERCHARATTRS= style-element <(options)> | (options)

specifies the appearance of the markers in the plot when you use the MARKERCHAR= option. You can specify the appearance by using a style element

or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the marker characters. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphDataText style element in the current style.

For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

FAMILY= “*font-family*”

specifies the font family for the marker characters. The SAS ODS styles use TrueType system fonts. For more information, see “TrueType Fonts Supplied by SAS” in Chapter 15 of *SAS Language Reference: Concepts*.

Default: The default font family is specified by the FontFamily attribute of the GraphDataText style element in the current style.

SIZE= *n* < *units* >

specifies the font size of the marker characters. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default font size is specified by the FontSize attribute of the GraphDataText style element in the current style.

STYLE= ITALIC | NORMAL

specifies whether the marker characters are italic (ITALIC) or normal (NORMAL).

Default: The default font style is specified by the FontStyle attribute of the GraphDataText style element in the current style.

WEIGHT= BOLD | NORMAL

specifies whether the marker characters are bold (BOLD) or normal (NORMAL).

Default: The default font weight is specified by the FontWeight attribute of the GraphDataText style element in the current style.

NAME= “*text-string*”

specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note: The name is case-sensitive. When you refer to the specified name in other statements, you must use the same exact capitalization.

NOMISSINGGROUP

specifies that missing values of the group variable are not included in the plot.

Interaction: This option has no effect unless GROUP= is also specified.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the markers and error bars. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default: 0.0

URL= *character-variable*

specifies a character variable that contains URLs for Web pages to be displayed when parts of the plot are selected within an HTML page.

Default: By default, no HTML links are created.

Interactions:

This option affects graphics output that is created through the ODS HTML destination only. For more information about ODS destinations, see [“Specifying a Destination” on page 649](#).

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For more information, see [“Using the ODS GRAPHICS Statement” on page 651](#).

XERRORLOWER= *numeric-variable*

specifies a variable that contains the lower endpoints for the X error bars.

XERRORUPPER= *numeric-variable*

specifies a variable that contains the upper endpoints for the X error bars.

YERRORLOWER= *numeric-variable*

specifies a variable that contains the lower endpoints for the Y error bars.

YERRORUPPER= *numeric-variable*

specifies a variable that contains the upper endpoints for the Y error bars.

SERIES Statement

Creates a series plot.

Example: [“About Series Plots” on page 29](#)

Syntax

SERIES X= *variable* Y= *variable* < / option(s)>;

Summary of Optional Arguments

Group options

CLUSTERWIDTH= *numeric-value*

specifies the width of the group clusters as a fraction of the midpoint spacing.

GROUP= *variable*

specifies a variable that is used to group the data.

GROUPDISPLAY= CLUSTER | OVERLAY

specifies how to display grouped lines.

GROUPORDER= ASCENDING | DESCENDING | DATA

specifies the ordering of lines within a group.

NOMISSINGGROUP

specifies that missing values of the group variable are not included in the plot.

Label options

CURVELABEL <=“*text-string*”>

adds a label for the curve.

CURVELABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the CURVELABEL= option.

CURVELABELPOS= MIN | MAX | START | END

specifies the location of the curve label.

DATALABEL <= *variable*>

displays a label for each data point.

DATALABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the DATALABEL= option.

Marker options

MARKERATTRS= *style-element* <(options)> | (options)

specifies the appearance of the markers in the plot.

MARKERS

adds data point markers to the series plot data points.

Plot options

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set.

LEGENDLABEL= “*text-string*”

specifies a label that identifies the series plot in the legend.

NAME= “*text-string*”

specifies a name for the plot.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the lines and markers.

URL= *character-variable*

specifies a character variable that contains URLs for Web pages to be displayed when parts of the plot are selected within an HTML page.

Series options

BREAK

creates a break in the line for each missing value for the Y variable.

DISCRETEOFFSET= *numeric-value*

specifies an amount to offset all lines from discrete X or Y values.

LINEATTRS= *style-element* <(options)> | (options)

specifies the appearance of the series line.

Required Arguments

X= *variable*

specifies the variable for the x axis.

Y= *variable*

specifies the variable for the y axis.

Optional Arguments

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph. For more information, see “Using SG Attribute Maps to Control Visual Attributes” on page 605.

BREAK

creates a break in the line for each missing value for the Y variable.

CLUSTERWIDTH= *numeric-value*

specifies the width of the group clusters as a fraction of the midpoint spacing. Specify a value from 0.0 (narrowest) to 1.0 (widest).

Default: 0.8

Interaction: This option is applicable only when the GROUP option is specified, when GROUPDISPLAY=CLUSTER, and when the category axis is discrete.

CURVELABEL <=“*text-string*”>

adds a label for the curve. You can also specify the label text. If you do not specify a label, the label from the Y variable is used.

CURVELABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the CURVELABEL= option. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

Here is an example that specifies a style element:

CURVELABELATTRS=GraphTitleText

options can be one or more of the following:

COLOR= *color*

specifies the color of the line label characters. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphValueText style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

FAMILY= “*font-family*”

specifies the font family for the label characters. The SAS ODS styles use TrueType system fonts. For more information, see “TrueType Fonts Supplied by SAS” in Chapter 15 of *SAS Language Reference: Concepts*.

Default: The default font family is specified by the FontFamily attribute of the GraphValueText style element in the current style.

Restriction: You cannot specify a list of font family names.

SIZE= *n* <*units*>

specifies the font size of the label characters. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default font size is specified by the FontSize attribute of the GraphValueText style element in the current style.

STYLE= ITALIC | NORMAL

specifies whether the label characters are italic or normal.

Default: The default font style is specified by the FontStyle attribute of the GraphValueText style element in the current style.

WEIGHT= BOLD | NORMAL

specifies whether the label characters are bold or normal.

Default: The default font weight is specified by the FontWeight attribute of the GraphValueText style element in the current style.

Here is an example expression:

CURVELABELATTRS=(Color=Green Family=Arial Size=8 Style=Italic Weight=Bold)

Default: The default style element is GraphValueText.

Interaction: This option has no effect unless the CURVELABEL option is also specified.

CURVELABELPOS= MIN | MAX | START | END

specifies the location of the curve label. Specify one of the following values:

MIN

places the label at the part of the curve closest to the minimum X axis value.

MAX

places the label at the part of the curve closest to the maximum X axis value.

START

places the curve label at the first point on the curve.

END

places the curve label at the last point on the curve.

Default: END

Interaction: This option has no effect unless the CURVELABEL option is also specified.

DATALABEL <= variable>

displays a label for each data point. If you specify a variable, the values of that variable are used for the data labels. If you do not specify a variable, then the values of the Y variable are used for the data labels.

DATALABELATTRS= style-element <(options)> | (options)

specifies the appearance of the labels in the plot when you use the DATALABEL= option. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

Here is an example that specifies a style element:

DATALABELATTRS=GraphLabelText

options can be one or more of the following:

COLOR= *color*

specifies the color of the label characters. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphDataText style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

FAMILY= “*font-family*”

specifies the font family for the label characters. The SAS ODS styles use TrueType system fonts. For more information, see “TrueType Fonts Supplied by SAS” in Chapter 15 of *SAS Language Reference: Concepts*.

Default: The default font family is specified by the FontFamily attribute of the GraphDataText style element in the current style.

SIZE= *n* <*units*>

specifies the font size of the label characters. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default font size is specified by the `FontSize` attribute of the `GraphDataText` style element in the current style.

`STYLE= ITALIC | NORMAL`

specifies whether the label characters are italic (`ITALIC`) or normal (`NORMAL`).

Default: The default font style is specified by the `FontStyle` attribute of the `GraphDataText` style element in the current style.

`WEIGHT= BOLD | NORMAL`

specifies whether the label characters are bold (`BOLD`) or normal (`NORMAL`).

Default: The default font weight is specified by the `FontWeight` attribute of the `GraphDataText` style element in the current style.

Here is an example expression:

```
DATALABELATTRS=(Color=Green Family=Arial Size=8 Style=Italic Weight=Bold)
```

Default: The default style element is `GraphDataText`.

Interaction: This option has no effect unless the `DATALABEL` option is also specified.

DISCRETEOFFSET= *numeric-value*

specifies an amount to offset all lines from discrete X or Y values. Specify a value from -0.5 (left offset) to +0.5 (right offset).

Default: 0.0 (no offset)

Requirement: This option is applicable only when the X or Y axis is discrete.

GROUP= *variable*

specifies a variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes.

GROUPDISPLAY= CLUSTER | OVERLAY

specifies how to display grouped lines.

CLUSTER

grouped items are drawn adjacent to each other.

OVERLAY

grouped items are drawn at the exact coordinate and might overlap. Each group is represented by unique visual attributes derived from the `GraphData1...` `GraphDatan` style elements in the current style.

Default: `OVERLAY`

Restriction: `GROUPDISPLAY=CLUSTER` can take effect only when the category axis is discrete. If necessary, use a `TYPE=DISCRETE` option on the axis statement.

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

GROUPORDER= ASCENDING | DESCENDING | DATA

specifies the ordering of lines within a group.

ASCENDING

displays each group in ascending group value.

DESCENDING

displays each group in descending group value.

DATA

displays each group in data order of the group variable.

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

LEGENDLABEL= “text-string”

specifies a label that identifies the series plot in the legend. By default, the label of the Y variable or the group value for each marker is used.

LINEATTRS= style-element <(options)> | (options)

specifies the appearance of the series line. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= color

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphDataDefault style element in the current style.

For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

PATTERN= line-pattern

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see “Line Patterns” on page 663.

Default: For ungrouped data, the default line pattern is specified by the LineStyle attribute of the GraphDataDefault style element in the current style.

For grouped data, the default line pattern is specified by the LineStyle attribute of the GraphData1 ... GraphData*n* style elements in the current style.

THICKNESS= n <units>

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: For ungrouped data, the default line thickness is specified by the LineThickness attribute of the GraphDataDefault style element in the current style.

For grouped data, the default line thickness is specified by the LineThickness attribute of the GraphData1 ... GraphData*n* style elements in the current style.

NAME= “text-string”

specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note: The name is case-sensitive. When you refer to the specified name in other statements, you must use the same exact capitalization.

MARKERATTRS= style-element <(options)> | (options)

specifies the appearance of the markers in the plot. This option has no effect unless you also specify the MARKERS option. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= color

specifies the color of the markers. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphDataDefault style element in the current style.

For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

SIZE= *n* <*units*>

specifies the size of the markers. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see [“Units of Measurement” on page 659](#).

Default: The default marker size is specified by the MarkerSize attribute of the GraphDataDefault style element in the current style.

SYMBOL= *symbol-name*

specifies the symbol for the markers. For a list of valid marker symbols, see [“Marker Symbols” on page 661](#).

Default: For ungrouped data, the default marker symbol is specified by the MarkerSymbol attribute of the GraphDataDefault style element in the current style.

For grouped data, the default marker symbol is specified by the MarkerSymbol attribute of the GraphData1 ... GraphData*n* style elements in the current style.

MARKERS

adds data point markers to the series plot data points.

NOMISSINGGROUP

specifies that missing values of the group variable are not included in the plot.

Interaction: This option has no effect unless GROUP= is also specified.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the lines and markers. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default: 0.0

URL= *character-variable*

specifies a character variable that contains URLs for Web pages to be displayed when parts of the plot are selected within an HTML page.

Default: By default, no HTML links are created.

Interactions:

This option affects graphics output that is created through the ODS HTML destination only. For more information about ODS destinations, see [“Specifying a Destination” on page 649](#).

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For more information, see [“Using the ODS GRAPHICS Statement” on page 651](#).

STEP Statement

Creates a step plot.

Restriction: The vertical axis that is used with the STEP statement cannot be a discrete axis.

Example: [“About Step Plots” on page 30](#)

Syntax

STEP X= *variable* Y= *numeric-variable* < / option(s)>;

Summary of Optional Arguments

Group options

CLUSTERWIDTH= *numeric-value*

specifies the width of the group clusters as a fraction of the midpoint spacing.

GROUP= *variable*

specifies a variable that is used to group the data.

GROUPDISPLAY= CLUSTER | OVERLAY

specifies how to display grouped step lines.

GROUPORDER= ASCENDING | DESCENDING | DATA

specifies the ordering of step lines within a group.

NOMISSINGGROUP

specifies that missing values of the group variable are not included in the plot.

Label options

CURVELABEL <="text-string">

adds a label for the curve.

CURVELABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the CURVELABEL= option.

CURVELABELPOS= MIN | MAX | START | END

specifies the location of the curve label.

DATALABEL <= *variable*>

displays a label for each data point.

DATALABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the DATALABEL= option.

Marker options

MARKERATTRS= *style-element* <(options)> | (options)

specifies the appearance of the markers in the plot.

MARKERS

adds markers to the step plot data points.

Plot options

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set.

LEGENDLABEL= "text-string"

specifies a label that identifies the step plot in the legend.

NAME= "text-string"

specifies a name for the plot.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the lines.

URL= *character-variable*

specifies a character variable that contains URLs for Web pages to be displayed when parts of the plot are selected within an HTML page.

Step options

BREAK

creates a break in the line for each missing value.

DISCRETEOFFSET= *numeric-value*

specifies an amount to offset all step lines from discrete X values.

ERRORBARATTRS= *style-element* <(options)> | (options)

specifies the appearance of the error bars in the plot.

JUSTIFY= LEFT | CENTER | RIGHT

specifies the location of each step relative to its data point.

LINEATTRS= *style-element* <(options)> | (options)

specifies the appearance of the step line.

YERRORLOWER= *numeric-variable*

specifies a variable that contains the lower endpoints for the Y error bars.

YERRORUPPER= *numeric-variable*

specifies a variable that contains the upper endpoints for the Y error bars.

Required Arguments

X= *variable*

specifies the variable for the x axis.

Y= *numeric-variable*

specifies the variable for the y axis.

Optional Arguments

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph. For more information, see [“Using SG Attribute Maps to Control Visual Attributes” on page 605](#).

BREAK

creates a break in the line for each missing value.

CLUSTERWIDTH= *numeric-value*

specifies the width of the group clusters as a fraction of the midpoint spacing. Specify a value from 0.0 (narrowest) to 1.0 (widest).

Default: 0.8

Interaction: This option is applicable only when the GROUP option is specified, when GROUPDISPLAY=CLUSTER, and when the category axis is discrete.

CURVELABEL <=“*text-string*”>

adds a label for the curve. You can also specify the label text. If you do not specify a label, the label from the Y variable is used.

CURVELABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the CURVELABEL= option. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

Here is an example that specifies a style element:

CURVELABELATTRS=GraphTitleText

options can be one or more of the following:

COLOR= *color*

specifies the color of the line label characters. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more

information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphValueText style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

FAMILY= “*font-family*”

specifies the font family for the label characters. The SAS ODS styles use TrueType system fonts. For more information, see “TrueType Fonts Supplied by SAS” in Chapter 15 of *SAS Language Reference: Concepts*.

Default: The default font family is specified by the FontFamily attribute of the GraphValueText style element in the current style.

Restriction: You cannot specify a list of font family names.

SIZE= *n* <*units*>

specifies the font size of the label characters. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default font size is specified by the FontSize attribute of the GraphValueText style element in the current style.

STYLE= ITALIC | NORMAL

specifies whether the label characters are italic or normal.

Default: The default font style is specified by the FontStyle attribute of the GraphValueText style element in the current style.

WEIGHT= BOLD | NORMAL

specifies whether the label characters are bold or normal.

Default: The default font weight is specified by the FontWeight attribute of the GraphValueText style element in the current style.

Here is an example expression:

```
CURVELABELATTRS=(Color=Green Family=Arial Size=8 Style=Italic Weight=Bold)
```

Default: The default style element is GraphValueText.

Interaction: This option has no effect unless the CURVELABEL option is also specified.

CURVELABELPOS= MIN | MAX | START | END

specifies the location of the curve label. Specify one of the following values:

MIN

places the label at the part of the curve closest to the minimum X axis value.

MAX

places the label at the part of the curve closest to the maximum X axis value.

START

places the curve label at the first point on the curve.

END

places the curve label at the last point on the curve.

Default: END

Interaction: This option has no effect unless the CURVELABEL option is also specified.

DATALABEL <= *variable*>

displays a label for each data point. If you specify a variable, the values of that variable are used for the data labels. If you do not specify a variable, then the values of the Y variable are used for the data labels.

DATALABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the DATALABEL= option. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

Here is an example that specifies a style element:

```
DATALABELATTRS=GraphLabelText
```

options can be one or more of the following:

COLOR= *color*

specifies the color of the label characters. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphDataText style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

FAMILY= “*font-family*”

specifies the font family for the label characters. The SAS ODS styles use TrueType system fonts. For more information, see “TrueType Fonts Supplied by SAS” in Chapter 15 of *SAS Language Reference: Concepts*.

Default: The default font family is specified by the FontFamily attribute of the GraphDataText style element in the current style.

SIZE= *n* <*units*>

specifies the font size of the label characters. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default font size is specified by the FontSize attribute of the GraphDataText style element in the current style.

STYLE= ITALIC | NORMAL

specifies whether the label characters are italic (ITALIC) or normal (NORMAL).

Default: The default font style is specified by the FontStyle attribute of the GraphDataText style element in the current style.

WEIGHT= BOLD | NORMAL

specifies whether the label characters are bold (BOLD) or normal (NORMAL).

Default: The default font weight is specified by the FontWeight attribute of the GraphDataText style element in the current style.

Here is an example expression:

```
DATALABELATTRS=(Color=Green Family=Arial Size=8 Style=Italic Weight=Bold)
```

Default: The default style element is GraphDataText.

Interaction: This option has no effect unless the DATALABEL option is also specified.

DISCRETEOFFSET= *numeric-value*

specifies an amount to offset all step lines from discrete X values. Specify a value from -0.5 (left offset) to +0.5 (right offset).

Default: 0.0 (no offset)

Requirement: This option is applicable only when the X axis is discrete.

ERRORBARATTRS= *style-element* <(options)> | (options)

specifies the appearance of the error bars in the plot. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: The default color is specified by the ContrastColor attribute of the GraphError style element in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see “Line Patterns” on page 663.

Default: The default line pattern is specified by the LineStyle attribute of the GraphError style element in the current style.

THICKNESS= *n* <units>

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default line thickness is specified by the LineThickness attribute of the GraphError style element in the current style.

GROUP= *variable*

specifies a variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes.

GROUPDISPLAY= CLUSTER | OVERLAY

specifies how to display grouped step lines.

CLUSTER

grouped items are drawn adjacent to each other.

OVERLAY

grouped items are drawn at the exact coordinate and might overlap. Each group is represented by unique visual attributes derived from the GraphData1...

GraphData*n* style elements in the current style.

Default: OVERLAY

Restriction: GROUPDISPLAY=CLUSTER can take effect only when the category axis is discrete. If necessary, use a TYPE=DISCRETE option on the axis statement.

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

GROUPORDER= ASCENDING | DESCENDING | DATA

specifies the ordering of step lines within a group.

ASCENDING

displays each group in ascending group value.

DESCENDING

displays each group in descending group value.

DATA

displays each group in data order of the group variable.

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

JUSTIFY= LEFT | CENTER | RIGHT

specifies the location of each step relative to its data point. [Figure 6.6 on page 210](#) shows the effect of each option:

Figure 6.6 Values for **JUSTIFY=**

**LEGENDLABEL= “text-string”**

specifies a label that identifies the step plot in the legend. By default, the label of the Y variable or the group value for each marker is used.

LINEATTRS= style-element <(options)> | (options)

specifies the appearance of the step line. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= color

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphDataDefault style element in the current style.

For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

PATTERN= line-pattern

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see “Line Patterns” on page 663.

Default: For ungrouped data, the default line pattern is specified by the LineStyle attribute of the GraphDataDefault style element in the current style.

For grouped data, the default line pattern is specified by the LineStyle attribute of the GraphData1 ... GraphData*n* style elements in the current style.

THICKNESS= n <units>

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: For ungrouped data, the default line thickness is specified by the LineThickness attribute of the GraphDataDefault style element in the current style.

For grouped data, the default line thickness is specified by the LineThickness attribute of the GraphData1 ... GraphData*n* style elements in the current style.

MARKERATTRS= *style-element* <(options)> | (options)

specifies the appearance of the markers in the plot. This option has no effect unless you also specify the MARKERS option. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the markers. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphDataDefault style element in the current style.

For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

SIZE= *n* <units>

specifies the size of the markers. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default marker size is specified by the MarkerSize attribute of the GraphDataDefault style element in the current style.

SYMBOL= *symbol-name*

specifies the symbol for the markers. For a list of valid marker symbols, see “Marker Symbols” on page 661.

Default: For ungrouped data, the default marker symbol is specified by the MarkerSymbol attribute of the GraphDataDefault style element in the current style.

For grouped data, the default marker symbol is specified by the MarkerSymbol attribute of the GraphData1 ... GraphData*n* style elements in the current style.

MARKERS

adds markers to the step plot data points.

NAME= “*text-string*”

specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note: The name is case-sensitive. When you refer to the specified name in other statements, you must use the same exact capitalization.

NOMISSINGGROUP

specifies that missing values of the group variable are not included in the plot.

Interaction: This option has no effect unless GROUP= is also specified.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the lines. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default: 0.0

URL= *character-variable*

specifies a character variable that contains URLs for Web pages to be displayed when parts of the plot are selected within an HTML page.

Default: By default, no HTML links are created.

Interactions:

This option affects graphics output that is created through the ODS HTML destination only. For more information about ODS destinations, see [“Specifying a Destination” on page 649](#).

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For more information, see [“Using the ODS GRAPHICS Statement” on page 651](#).

YERRORLOWER= *numeric-variable*

specifies a variable that contains the lower endpoints for the Y error bars.

YERRORUPPER= *numeric-variable*

specifies a variable that contains the upper endpoints for the Y error bars.

VBAR Statement

Creates a vertical bar chart that summarizes the values of a category variable.

Interaction: The VBAR statement can be combined only with other vertical categorization plot statements in the SG PANEL procedure. See [“Plot Type Compatibility” on page 81](#).

Example: [“About Bar Charts” on page 42](#)

Syntax

VBAR *category-variable* < / option(s) >;

Summary of Optional Arguments

Bar options

BARWIDTH= *numeric-value*

specifies the width of the bars as a ratio of the maximum possible width.

DATASKIN= NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN

specifies a special effect to be used on all filled bars.

DISCRETEOFFSET= *numeric-value*

specifies an amount to offset all bars from the category midpoints.

FILL | NOFILL

specifies whether the bars are filled.

FILLATTRS= *style-element* | (COLOR= *color*)

specifies the appearance of the fill for the bars.

LIMITATTRS= *style-element* <(options)> | (options)

specifies the appearance of the limit lines in the plot.

LIMITS= BOTH | LOWER | UPPER

specifies which limit lines to display.

LIMITSTAT= CLM | STDDEV | STDERR

specifies the statistic for the limit lines.

MISSING

processes missing values as a valid category value and creates a bar for it.

NOSTATLABEL

removes the statistic name from the axis and legend labels.

NUMSTD= *n*

specifies the number of standard units for the limit lines, when you specify LIMITSTAT= STDDEV or LIMITSTAT= STDERR.

OUTLINE | NOOUTLINE

specifies whether the bars have outlines.

RESPONSE= *response-variable*

specifies a numeric response variable for the plot.

STAT= FREQ | MEAN | SUM

specifies the statistic for the vertical axis.

Group options

CLUSTERWIDTH= *numeric-value*

specifies the cluster width as a ratio of the maximum width.

GROUP= *variable*

specifies a variable that is used to group the data.

GROUPDISPLAY= STACK | CLUSTER

specifies how to display grouped bars.

GROUPORDER= ASCENDING | DESCENDING | DATA

specifies the ordering of bars within a group.

Label options

DATALABEL <= *variable*>

displays a label for each data point.

DATALABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the DATALABEL= option.

Plot options

ALPHA= *numeric-value*

specifies the confidence level for the confidence limits.

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set.

CATEGORYORDER= RESPASC | RESPDESC

specifies the order in which the response values are arranged.

FREQ= *numeric-variable*

specifies how many times observations are repeated for computational purposes.

LEGENDLABEL= "*text-string*"

specifies the label that identifies the bar chart in the legend.

NAME= "*text-string*"

specifies a name for the plot.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the bars and limits, if displayed.

URL= *character-variable*

specifies a character variable that contains URLs for Web pages to be displayed when parts of the plot are selected within an HTML page.

WEIGHT= *numeric-variable*

specifies how observations are weighted.

Required Argument

category-variable

specifies the variable that classifies the observations into distinct subsets.

Optional Arguments

ALPHA= *numeric-value*

specifies the confidence level for the confidence limits. Specify a number between 0.00 (100% confidence) and 1.00 (0% confidence).

Default: .05

Interactions:

This option has no effect if you do not specify LIMITSTAT=CLM.

If your plot is overlaid with other categorization plots, then the first ALPHA value that you specify is used for all of the plots.

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph. For more information, see [“Using SG Attribute Maps to Control Visual Attributes” on page 605](#).

BARWIDTH= *numeric-value*

specifies the width of the bars as a ratio of the maximum possible width. The maximum width is equal to the distance between the center of each bar and the centers of the adjacent bars. Specify a value from 0.0 (narrowest) to 1.0 (widest).

For example, if you specify a width of 1, then there is no space between the bars. If you specify a width of .5, then the width of the bars is equal to the space between the bars.

Default: .8

1.0 when the GROUP option is specified and GROUPDISPLAY=CLUSTER

Interaction: When the GROUP option is specified, the bar width is determined by the maximum number of bars in any one group cluster. All bars are drawn with the same width, with the cluster being positioned symmetrically around the midpoint.

CATEGORYORDER= RESPASC | RESPDESC

specifies the order in which the response values are arranged. Specify one of the following values:

RESPASC

sorts the response values in ascending order.

RESPDESC

sorts the response values in descending order.

Default: By default, the plot is sorted in ascending order based on the category values.

Restriction: This option takes effect only when the plot statement specifies a response variable and the axis for that variable is numeric. If the axis is not numeric, an error is generated and a message is written to the SAS log.

Requirement: This option requires that you configure the panel to use either one column or one row, depending on the orientation of your charts. Use the ROWS= or the COLUMNS= option in the PANELBY statement. If you do not use this option and your graph contains multiple cells, the specified sort order is not correctly applied to all cells.

Interaction: If a group variable is specified in the plot statement, the response sort option is ignored and a note is generated in the log. (You can use a BY statement in the procedure if you want to group data.)

Note: If CATEGORYORDER is specified in multiple statements, the procedure sorts by the last statement in which it is specified.

CLUSTERWIDTH= *numeric-value*

specifies the cluster width as a ratio of the maximum width. Specify a value from 0.0 (narrowest) to 1.0 (widest).

CLUSTERWIDTH is the fraction of the midpoint spacing used by all bars that are clustered around a midpoint (category value). The bar width is applied to the maximum bar spacing divided by the maximum number of bars in any one cluster.

Default: 0.8

Interaction: This option is applicable only when the GROUP option is specified, when GROUPDISPLAY=CLUSTER, and when the category axis is discrete.

DATALABEL <= *variable*>

displays a label for each data point. If you specify a variable, then the values of that variable are used for the data labels. If you do not specify a variable, then the values of the calculated response are used for the data labels.

DATALABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the DATALABEL= option. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

Here is an example that specifies a style element:

DATALABELATTRS=GraphLabelText

options can be one or more of the following:

COLOR= *color*

specifies the color of the label characters. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphDataText style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

FAMILY= “*font-family*”

specifies the font family for the label characters. The SAS ODS styles use TrueType system fonts. For more information, see “TrueType Fonts Supplied by SAS” in Chapter 15 of *SAS Language Reference: Concepts*.

Default: The default font family is specified by the FontFamily attribute of the GraphDataText style element in the current style.

SIZE= *n* <*units*>

specifies the font size of the label characters. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default font size is specified by the FontSize attribute of the GraphDataText style element in the current style.

STYLE= ITALIC | NORMAL

specifies whether the label characters are italic (ITALIC) or normal (NORMAL).

Default: The default font style is specified by the FontStyle attribute of the GraphDataText style element in the current style.

WEIGHT= BOLD | NORMAL

specifies whether the label characters are bold (BOLD) or normal (NORMAL).

Default: The default font weight is specified by the FontWeight attribute of the GraphDataText style element in the current style.

Here is an example expression:

```
DATALABELATTRS=(Color=Green Family=Arial Size=8 Style=Italic Weight=Bold)
```

Default: The default style element is GraphDataText.

Interaction: This option has no effect unless the DATALABEL option is also specified.

DATASKIN= NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
specifies a special effect to be used on all filled bars. Specify one of the following (the samples were generated using the HTMLBlue style):

NONE



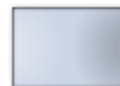
CRISP



GLOSS



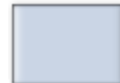
MATTE



PRESSED



SHEEN



Default: NONE

Interactions:

This option has no effect if the applied style uses a fill pattern and FILL is disabled for the style. If the style has both FILL and FILLPATTERN enabled, data skins will work. For more information about fill patterns, see [“Using Fill Patterns to Distinguish Grouped Bar Charts” on page 636](#). For instructions about viewing the contents of style templates, see [“About Styles and Style Elements” on page 620](#).

This option has no effect if you also specify NOFILL.

DISCRETEOFFSET= *numeric-value*

specifies an amount to offset all bars from the category midpoints. Specify a value from -0.5 (left offset) to +0.5 (right offset).

Default: 0.0 (no offset)

FILL | NOFILL

specifies whether the bars are filled. The FILL option shows the fill color for the bars. The NOFILL option hides the fill color for the bars.

Default: FILL

Interactions:

Specifying FILL also hides the outlines.

If NOFILL and NOOUTLINE are both specified, then both options are ignored.

FILLATTRS= *style-element* | (COLOR= *color*)

specifies the appearance of the fill for the bars. You can specify the color of the fill by using a style element or by using the COLOR= suboption. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphDataDefault style element in the current style.

For grouped data, the default color is specified by the Color attribute of the GraphData1... GraphData*n* style elements in the current style.

Interaction: This option has no effect if you specify the NOFILL option.

FREQ= *numeric-variable*

specifies how many times observations are repeated for computational purposes. Each observation is repeated *n* times for computational purposes, where *n* is the value of the numeric variable. If *n* is not an integer, then it is truncated to an integer. If *n* is less than 1 or missing, then it is excluded from the analysis.

Interaction: If your plot is overlaid with other categorization plots, then the first FREQ variable that you specified is used for all of the plots.

GROUP= *variable*

specifies a variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes.

GROUPDISPLAY= STACK | CLUSTER

specifies how to display grouped bars.

STACK

groups are overlaid without any clustering. Each group is represented by unique visual attributes derived from the GraphData1... GraphData*n* style elements in the current style.

CLUSTER

displays group values as separate adjacent bars that replace the single category bar. Each set of group values is centered at the midpoint tick mark for the category.

Default: STACK

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

GROUPORDER= ASCENDING | DESCENDING | DATA

specifies the ordering of bars within a group.

ASCENDING

displays each group in ascending group value.

DESCENDING

displays each group in descending group value.

DATA

displays each group in data order of the group variable.

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

LEGENDLABEL= *“text-string”*

specifies the label that identifies the bar chart in the legend. By default, the label of the RESPONSE= variable is used. If there is no response variable label, the name of the response variable and the computed statistic (SUM or MEAN) is used. If the RESPONSE= option is not used, the legend label is “Frequency”.

Interaction: The LEGENDLABEL= option has no effect if you also specify the GROUP= option in the same plot statement.

LIMITATTRS= *style-element* <(options)> | (options)

specifies the appearance of the limit lines in the plot. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: The default color is specified by the ContrastColor attribute of the GraphError style element in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see “Line Patterns” on page 663.

Default: The default line pattern is specified by the LineStyle attribute of the GraphError style element in the current style.

THICKNESS= *n* <*units*>

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default line thickness is specified by the LineThickness attribute of the GraphError style element in the current style.

LIMITS= BOTH | LOWER | UPPER

specifies which limit lines to display. Limits are displayed as heavier line segments with a serif at the end extending from each bar. Upper limits extend to the right of the bar and lower limits extend to the left of the bar. By default, no limits are displayed unless you specify either the LIMITS= or LIMITSTAT= option. If you specify the LIMITSTAT= option only, then LIMITS=BOTH is the default. Specify one of the following values:

BOTH

adds lower and upper limit lines to the plot.

LOWER

adds lower limit lines to the plot.

UPPER

adds upper limit lines to the plot.

Default: By default, no limit lines are displayed. However, if you specify the LIMITSTAT= option, then the default is BOTH.

Interactions:

Limit lines are displayed only when you specify STAT= MEAN.

The LIMITS= option has no effect if you also specify the GROUP= option in the plot statement.

LIMITSTAT= CLM | STDDEV | STDERR

specifies the statistic for the limit lines. Specify one of the following statistics:

CLM

confidence limits

STDDEV

standard deviation

STDERR

standard error

Default: CLM

Interactions:

If you specify the LIMITSTAT= option only, then the default value for the LIMITS= option is BOTH.

Limits lines are displayed only when you specify STAT=MEAN.

The LIMITSTAT= option has no effect if you also specify the GROUP= option in the plot statement.

MISSING

processes missing values as a valid category value and creates a bar for it.

NAME= "text-string"

specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note: The name is case-sensitive. When you refer to the specified name in other statements, you must use the same exact capitalization.

NOSTATLABEL

removes the statistic name from the axis and legend labels.

NUMSTD= *n*

specifies the number of standard units for the limit lines, when you specify LIMITSTAT= STDDEV or LIMITSTAT= STDERR. You can specify any positive number, including decimals.

Default: 1

OUTLINE | NOOUTLINE

specifies whether the bars have outlines. The OUTLINE option shows the outlines. The NOOUTLINE option hides the outlines.

Default: OUTLINE

Interactions:

Specifying OUTLINE also hides the fill color.

If NOOUTLINE and NOFILL are both specified, then both options are ignored.

RESPONSE= *response-variable*

specifies a numeric response variable for the plot. The summarized values of the response variable are displayed on the vertical axis.

STAT= FREQ | MEAN | SUM

specifies the statistic for the vertical axis. Specify one of the following statistics:

FREQ

the frequencies for the category variable. This is the default value when you do not specify the RESPONSE= option.

MEAN

the mean of the response variable.

SUM

the sum of the response variable. This is the default value when you specify the RESPONSE= option.

Interaction: If you do not specify the RESPONSE= option, then only the FREQ statistic can be used. If you specify the RESPONSE= option, then you can use either the SUM or MEAN statistics.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the bars and limits, if displayed. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default: 0.0

URL= *character-variable*

specifies a character variable that contains URLs for Web pages to be displayed when parts of the plot are selected within an HTML page.

Default: By default, no HTML links are created.

Interactions:

This option affects graphics output that is created through the ODS HTML destination only. For more information about ODS destinations, see [“Specifying a Destination” on page 649](#).

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For more information, see [“Using the ODS GRAPHICS Statement” on page 651](#).

WEIGHT= *numeric-variable*

specifies how observations are weighted. Each observation is weighted by a factor of w for computational purposes, where w is the value of the numeric variable. w can be any numeric value. If w is 0, negative or missing, then that observation is excluded from the analysis.

Interaction: If your plot is overlaid with other categorization plots, then the first WEIGHT variable that you specified is used for all of the plots.

VBARPARM Statement

Creates a vertical bar chart based on a pre-summarized response value for each unique value of the category variable. You can also assign variables to the upper and lower limits.

Requirement: The data must contain only one response value per unique category variable. If more than one value is found, a warning is written to the SAS log, and the graph might produce unpredictable results.

Interactions: The VBARPARM statement can be combined only with other basic plot statements in the SGPPLOT procedure. See [“Plot Type Compatibility” on page 277](#).
When used with particular styles, the VBARPARM statement produces fill patterns for grouped bars. For more information about fill patterns, see [“Using Fill Patterns to Distinguish Grouped Bar Charts” on page 636](#).

Note: An important distinction between VBARPARM and VBAR is that the response variable is required for VBARPARM. In addition, the response variable should contain pre-summarized computed values such as a sum or a mean.

Example: [“About Bar Charts” on page 42](#)

Syntax

VBARPARG CATEGORY= *category-variable* RESPONSE= *numeric-variable* </ option(s)>;

Summary of Optional Arguments

Bar options

BARWIDTH= *numeric-value*

specifies the width of the bars as a ratio of the maximum possible width.

DATASKIN= NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN

specifies a special effect to be used on all filled bars.

DISCRETEOFFSET= *numeric-value*

specifies an amount to offset all bars from the category midpoints.

FILL | NOFILL

specifies whether the bars are filled.

FILLATTRS= *style-element* | (COLOR= *color*)

specifies the appearance of the fill for the bars.

LIMITATTRS= *style-element* <(options)> | (options)

specifies the appearance of the limit lines in the plot.

LIMITLOWER= *numeric-variable*

specifies values for the lower endpoints on the limit lines.

LIMITUPPER= *numeric-variable*

specifies values for the upper endpoints on the limit lines.

MISSING

processes missing values as a valid category value and creates a bar for it.

OUTLINE | NOOUTLINE

specifies whether the bars have outlines.

Group options

CLUSTERWIDTH= *numeric-value*

specifies the cluster width as a ratio of the maximum width.

GROUP= *variable*

specifies a variable that is used to group the data.

GROUPORDER= ASCENDING | DESCENDING | DATA

specifies the ordering of bars within a group.

Label options

DATALABEL <= *variable*>

displays a label for each data point.

DATALABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the DATALABEL= option.

Plot options

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set.

LEGENDLABEL= “*text-string*”

specifies the label that identifies the bar chart in the legend.

NAME= “*text-string*”

specifies a name for the plot.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the bars and limits, if displayed.

URL= *character-variable*

specifies a character variable that contains URLs for Web pages to be displayed when parts of the plot are selected within an HTML page.

Required Arguments

CATEGORY= *category-variable*

specifies the variable that categorizes the data. All values are treated as discrete values. The input data for this variable should contain unique values. When the category values are not unique, a warning is logged, and multiple bars are superimposed at the duplicated category values.

The CATEGORY axis is always discrete.

RESPONSE= *numeric-variable*

specifies a numeric response variable. The input data is expected to be pre-summarized computed values (sum, mean, and so on).

Optional Arguments

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph. For more information, see [“Using SG Attribute Maps to Control Visual Attributes” on page 605](#).

BARWIDTH= *numeric-value*

specifies the width of the bars as a ratio of the maximum possible width. The maximum width is equal to the distance between the center of each bar and the centers of the adjacent bars. Specify a value from 0.0 (narrowest) to 1.0 (widest).

For example, if you specify a width of **1**, then there is no space between the bars. If you specify a width of **.5**, then the width of the bars is equal to the space between the bars.

Default: .8

1.0 when the GROUP option is specified and GROUPDISPLAY=CLUSTER

Interaction: When the GROUP option is specified, the bar width is determined by the maximum number of bars in any one group cluster. All bars are drawn with the same width, with the cluster being positioned symmetrically around the midpoint.

CLUSTERWIDTH= *numeric-value*

specifies the cluster width as a ratio of the maximum width. Specify a value from 0.0 (narrowest) to 1.0 (widest).

CLUSTERWIDTH is the fraction of the midpoint spacing used by all bars that are clustered around a midpoint (category value). The bar width is applied to the maximum bar spacing divided by the maximum number of bars in any one cluster.

Default: 0.8

Interaction: This option is applicable only when the GROUP option is specified, when GROUPDISPLAY=CLUSTER, and when the category axis is discrete.

DATALABEL *<= variable>*

displays a label for each data point. If you specify a variable, then the values of that variable are used for the data labels. If you do not specify a variable, then the values of the calculated response are used for the data labels.

DATALABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the DATALABEL= option. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

Here is an example that specifies a style element:

DATALABELATTRS=GraphLabelText

options can be one or more of the following:

COLOR= *color*

specifies the color of the label characters. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphDataText style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

FAMILY= “*font-family*”

specifies the font family for the label characters. The SAS ODS styles use TrueType system fonts. For more information, see “TrueType Fonts Supplied by SAS” in Chapter 15 of *SAS Language Reference: Concepts*.

Default: The default font family is specified by the FontFamily attribute of the GraphDataText style element in the current style.

SIZE= *n* <*units*>

specifies the font size of the label characters. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default font size is specified by the FontSize attribute of the GraphDataText style element in the current style.

STYLE= ITALIC | NORMAL

specifies whether the label characters are italic (ITALIC) or normal (NORMAL).

Default: The default font style is specified by the FontStyle attribute of the GraphDataText style element in the current style.

WEIGHT= BOLD | NORMAL

specifies whether the label characters are bold (BOLD) or normal (NORMAL).

Default: The default font weight is specified by the FontWeight attribute of the GraphDataText style element in the current style.

Here is an example expression:







DATALABELATTRS=(Color=Green Family=Arial Size=8 Style=Italic Weight=Bold)

Default: The default style element is GraphDataText.

Interaction: This option has no effect unless the DATALABEL option is also specified.

DATASKIN= NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN

specifies a special effect to be used on all filled bars. Specify one of the following (the samples were generated using the HTMLBlue style):

NONE	
CRISP	
GLOSS	
MATTE	
PRESSED	
SHEEN	

Default: NONE

Interactions:

This option has no effect if the applied style uses a fill pattern and FILL is disabled for the style. If the style has both FILL and FILLPATTERN enabled, data skins will work. For more information about fill patterns, see [“Using Fill Patterns to Distinguish Grouped Bar Charts” on page 636](#). For instructions about viewing the contents of style templates, see [“About Styles and Style Elements” on page 620](#).

This option has no effect if you also specify NOFILL.

DISCRETEOFFSET= *numeric-value*

specifies an amount to offset all bars from the category midpoints. Specify a value from -0.5 (left offset) to +0.5 (right offset).

Default: 0.0 (no offset)

FILL | NOFILL

specifies whether the bars are filled. The FILL option shows the fill color for the bars. The NOFILL option hides the fill color for the bars.

Default: FILL

Interactions:

Specifying FILL also hides the outlines.

If NOFILL and NOOUTLINE are both specified, then both options are ignored.

FILLATTRS= *style-element* | (COLOR= *color*)

specifies the appearance of the fill for the bars. You can specify the color of the fill by using a style element or by using the COLOR= suboption. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphDataDefault style element in the current style. For grouped data, the default color is specified by the Color attribute of the GraphData1... GraphData*n* style elements in the current style.

Interaction: This option has no effect if you specify the NOFILL option.

GROUP= *variable*

specifies a variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes.

GROUPORDER= ASCENDING | DESCENDING | DATA

specifies the ordering of bars within a group.

ASCENDING

displays each group in ascending group value.

DESCENDING

displays each group in descending group value.

DATA

displays each group in data order of the group variable.

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

LEGENDLABEL= “*text-string*”

specifies the label that identifies the bar chart in the legend. By default, the label of the RESPONSE= variable is used. If there is no response variable label, the name of the response variable is used.

Interaction: The LEGENDLABEL= option has no effect if you also specify the GROUP= option in the same plot statement.

LIMITATTRS= *style-element* <(options)> | (options)

specifies the appearance of the limit lines in the plot. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: The default color is specified by the ContrastColor attribute of the GraphError style element in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see “Line Patterns” on page 663.

Default: The default line pattern is specified by the LineStyle attribute of the GraphError style element in the current style.

THICKNESS= *n* <*units*>

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default line thickness is specified by the LineThickness attribute of the GraphError style element in the current style.

LIMITLOWER= *numeric-variable*

specifies values for the lower endpoints on the limit lines. Limit lines are displayed as line segments with a serif at the end.

Default: The lower segments of the limit lines are not displayed. (Limit lines are displayed only if either LIMITLOWER= or LIMITUPPER= is specified.)

Interactions:

If LIMITUPPER= is also specified, then the plot displays the lower and upper segments of the limit lines.

This option is ignored if GROUP= is specified and GROUPDISPLAY=STACK.

The appearance of the limit lines can be controlled by the LIMITATTRS= option.

LIMITUPPER= *numeric-variable*

specifies values for the upper endpoints on the limit lines. Limit lines are displayed as line segments with a serif at the end.

Default: The upper segments of the limit lines are not displayed. (Limit lines are displayed only if either LIMITLOWER= or LIMITUPPER= is specified.)

Interactions:

If LIMITLOWER= is also specified, then the plot displays the lower and upper segments of the limit lines.

This option is ignored if GROUP= is specified and GROUPDISPLAY=STACK.

The appearance of the limit lines can be controlled by the LIMITATTRS= option.

MISSING

processes missing values as a valid category value and creates a bar for it.

NAME= "*text-string*"

specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note: The name is case-sensitive. When you refer to the specified name in other statements, you must use the same exact capitalization.

OUTLINE | NOOUTLINE

specifies whether the bars have outlines. The OUTLINE option shows the outlines. The NOOUTLINE option hides the outlines.

Default: OUTLINE

Interactions:

Specifying OUTLINE also hides the fill color.

If NOOUTLINE and NOFILL are both specified, then both options are ignored.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the bars and limits, if displayed. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default: 0.0

URL= *character-variable*

specifies a character variable that contains URLs for Web pages to be displayed when parts of the plot are selected within an HTML page.

Default: By default, no HTML links are created.

Interactions:

This option affects graphics output that is created through the ODS HTML destination only. For more information about ODS destinations, see [“Specifying a Destination” on page 649](#).

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For more information, see [“Using the ODS GRAPHICS Statement” on page 651](#).

VBOX Statement

Creates a vertical box plot that shows the distribution of your data.

Interaction: The VBOX statement cannot be used together with other plot statements in the SGPanel procedure. Box plots can be overlaid with other box plots. However, overlaid box plots must have the same category variables.

Example: [“About Box Plots” on page 37](#)

Syntax

VBOX *analysis-variable* *</ option(s)>*;

Summary of Optional Arguments

Box options

BOXWIDTH= *numeric-value*

specifies the width of the box.

CAPSHAPE= BRACKET | LINE | SERIF

specifies the shape of the whisker cap lines.

CONNECT= MEAN | MEDIAN | Q1 | Q3 | MIN | MAX

specifies that a connect line joins a statistic from box to box.

CONNECTATTRS= *style-element* *<(options)>* | *(options)*

specifies the appearance of the lines that connect multiple boxes.

DISCRETEOFFSET= *numeric-value*

specifies an amount to offset all boxes from the discrete tick marks.

EXTREME

specifies that the whiskers can extend to the maximum and minimum values for the analysis variable, and that outliers are not identified.

FILL | NOFILL

specifies whether the boxes are filled with color.

FILLATTRS= *style-element* | (COLOR= *color*)

specifies the appearance of the fill for the boxes.

LINEATTRS= *style-element* *<(options)>* | *(options)*

specifies the appearance of the box outlines.

MEANATTRS= *style-element* *<(options)>* | *(options)*

specifies the appearance of the marker that represents the mean in the box.

MEDIANATTRS= *style-element* *<(options)>* | *(options)*

specifies the appearance of the line that represents the median.

NOCAPS

hides the cap lines for the whiskers.

NOMEAN

hides the mean marker.

NOMEDIAN

hides the median line.

NOOUTLIERS

hides the outliers from the plot.

NOTCHES

specifies that the boxes be notched.

OUTLIERATTRS= *style-element* <(options)> | (options)

specifies the appearance of the marker that represents the outliers.

WHISKERATTRS= *style-element* <(options)> | (options)

specifies the appearance of the whisker and cap lines.

Group options

CLUSTERWIDTH= *numeric-value*

specifies the cluster width as a ratio of the midpoint spacing.

GROUP= *variable*

specifies a variable that is used to group the data.

GROUPDISPLAY= CLUSTER | OVERLAY

specifies how to display grouped boxes.

GROUPORDER= ASCENDING | DESCENDING | DATA

specifies the ordering of boxes within a group.

Label options

DATALABEL <= *variable*>

adds data labels for the outlier markers.

DATALABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the DATALABEL= option.

LABELFAR

specifies that only the far outliers have data labels.

Plot options

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set.

CATEGORY= *category-variable*

specifies the category variable for the plot.

FREQ= *numeric-variable*

specifies how many times observations are repeated for computational purposes.

LEGENDLABEL= “*text-string*”

specifies a label that identifies the box plot in the legend.

MISSING

processes missing values as a valid category value and creates a box for it.

NAME= “*text-string*”

specifies a name for the plot.

PERCENTILE= 1 | 2 | 3 | 4 | 5

specifies a method for computing the percentiles for the plot.

SPREAD

relocates outlier points that have identical values to prevent overlapping.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the plot.

Required Argument

analysis-variable

specifies the analysis variable for the plot. If you do not specify the CATEGORY= option, then one box is created for the analysis variable.

Optional Arguments

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph. For more information, see [“Using SG Attribute Maps to Control Visual Attributes” on page 605](#).

BOXWIDTH= *numeric-value*

specifies the width of the box. Specify a value between 0.0 (0% of the available width) and 1.0 (100% of the available width).

Default: 0.4

When GROUP is specified, the default box width is 0.6.

CAPSHAPE= BRACKET | LINE | SERIF

specifies the shape of the whisker cap lines. Specify one of the following values:

BRACKET

displays a straight line with brackets.

LINE

displays a straight line.

SERIF

displays a short straight line.

CATEGORY= *category-variable*

specifies the category variable for the plot. A box plot is created for each distinct value of the category variable.

If you explicitly set the category axis type to LINEAR and use a numeric category variable, the box plot becomes an interval plot. Otherwise, the box plot is discrete.

CLUSTERWIDTH= *numeric-value*

specifies the cluster width as a ratio of the midpoint spacing. Specify a value from 0.1 (narrowest) to 1.0 (widest).

Default: 0.7

Interaction: This option is applicable only when a GROUP is in effect and the category axis is discrete.

CONNECT= MEAN | MEDIAN | Q1 | Q3 | MIN | MAX

specifies that a connect line joins a statistic from box to box.

Interaction: This option applies only when the CATEGORY option is used to generate multiple boxes.

Tip: You can use the CONNECTATTRS option to specify attributes for the connect line.

CONNECTATTRS= *style-element* <(options)> | (options)

specifies the appearance of the lines that connect multiple boxes. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

Here is an example that specifies a style element: **CONNECTATTRS=GraphData3**

options can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphConnectLine style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see [“Line Patterns” on page 663](#).

Default: For ungrouped data, the default line pattern is specified by the LineStyle attribute of the GraphConnectLine style element in the current style. For grouped data, the default line pattern is specified by the LineStyle attribute of the GraphData1 ... GraphData*n* style elements in the current style.

THICKNESS= *n* *<units>*

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see [“Units of Measurement” on page 659](#).

Default: For ungrouped data, the default line thickness is specified by the LineThickness attribute of the GraphConnectLine style element in the current style. For grouped data, the default line thickness is specified by the LineThickness attribute of the GraphData1 ... GraphData*n* style elements in the current style.

Here is an example expression:

```
CONNECTATTRS=(Color= "light green" Pattern= MediumDash Thickness= 4)
```

Default: The default style element is GraphConnectLine.

Interactions:

This option takes effect only if the CONNECT= option is also specified.

This option is ignored if the GROUP= option is also specified.

DATALABEL *<= variable>*

adds data labels for the outlier markers. If you specified a variable, then the values for that variable are used for the data labels. If you did not specify a variable, then the values of the analysis variable are used.

Note: This option has no effect unless the plot contains outlier points.

DATALABELATTRS= *style-element* *<(options)>* | *(options)*

specifies the appearance of the labels in the plot when you use the DATALABEL= option. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

Here is an example that specifies a style element:

```
DATALABELATTRS=GraphLabelText
```

options can be one or more of the following:

COLOR= *color*

specifies the color of the label characters. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see [“Color-Naming Schemes” in Chapter 14 of SAS/GRAPH: Reference](#).

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphDataText style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

FAMILY= “*font-family*”

specifies the font family for the label characters. The SAS ODS styles use TrueType system fonts. For more information, see “TrueType Fonts Supplied by SAS” in Chapter 15 of *SAS Language Reference: Concepts*.

Default: The default font family is specified by the FontFamily attribute of the GraphDataText style element in the current style.

SIZE= *n* <*units*>

specifies the font size of the label characters. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default font size is specified by the FontSize attribute of the GraphDataText style element in the current style.

STYLE= ITALIC | NORMAL

specifies whether the label characters are italic (ITALIC) or normal (NORMAL).

Default: The default font style is specified by the FontStyle attribute of the GraphDataText style element in the current style.

WEIGHT= BOLD | NORMAL

specifies whether the label characters are bold (BOLD) or normal (NORMAL).

Default: The default font weight is specified by the FontWeight attribute of the GraphDataText style element in the current style.

Here is an example expression:

```
DATALABELATTRS=(Color=Green Family=Arial Size=8 Style=Italic Weight=Bold)
```

Default: The default style element is GraphDataText.

Interaction: This option has no effect unless the DATALABEL option is also specified.

DISCRETEOFFSET= *numeric-value*

specifies an amount to offset all boxes from the discrete tick marks.

Specify a value from -0.5 (left offset) to +0.5 (right offset). If you specify a value outside of this range, an error message appears in the SAS log and the graph is not produced.

Default: 0.0 (no offset)

EXTREME

specifies that the whiskers can extend to the maximum and minimum values for the analysis variable, and that outliers are not identified. When you do not specify the EXTREME option, the whiskers cannot be longer than 1.5 times the length of the box.

FILL | NOFILL

specifies whether the boxes are filled with color. The FILL option shows the fill color. The NOFILL option hides the fill color.

Default: FILL

FILLATTRS= *style-element* | (COLOR= *color*)

specifies the appearance of the fill for the boxes. You can specify the color of the fill by using a style element or by using the COLOR= suboption. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphDataDefault style element in the current style.

For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData n style elements in the current style.

Interaction: This option has no effect if you also specify the NOFILL option.

FREQ= *numeric-variable*

specifies how many times observations are repeated for computational purposes. Each observation is repeated n times for computational purposes, where n is the value of the numeric variable. If n is not an integer, then it is truncated to an integer. If n is less than 1 or missing, then it is excluded from the analysis.

Interaction: If your plot is overlaid with other categorization plots, then the first FREQ variable that you specified is used for all of the plots.

GROUP= *variable*

specifies a variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes.

GROUPDISPLAY= CLUSTER | OVERLAY

specifies how to display grouped boxes.

CLUSTER

the boxes are drawn adjacent to each other.

OVERLAY

all the boxes for a given group value are drawn at the exact coordinate and might overlap. Each group is represented by unique visual attributes derived from the GraphData1... GraphData n style elements in the current style.

Default: CLUSTER for a discrete category axis

OVERLAY for a linear axis

Restriction: GROUPDISPLAY=CLUSTER can take effect only when the category axis is discrete.

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

GROUPORDER= ASCENDING | DESCENDING | DATA

specifies the ordering of boxes within a group.

ASCENDING

displays each group in ascending group value.

DESCENDING

displays each group in descending group value.

DATA

displays each group in data order of the group variable.

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

LABELFAR

specifies that only the far outliers have data labels. Far outliers are points whose distance from the box is more than three times the length of the box.

Note: This option has no effect if you do not specify the DATALABEL option, or if there are no far outliers.

LEGENDLABEL= “*text-string*”

specifies a label that identifies the box plot in the legend. By default, the label of the analysis variable is used.

LINEATTRS= *style-element* <(options)> | (options)

specifies the appearance of the box outlines. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

Here is an example that specifies a style element: **LINEATTRS=GraphData3**

For more information about style elements, see [“Style Elements for Use with ODS Graphics” on page 640](#).

options can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphDataDefault style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see [“Line Patterns” on page 663](#).

Default: For ungrouped data, the default line pattern is specified by the LineStyle attribute of the GraphDataDefault style element in the current style. For grouped data, the default line pattern is specified by the LineStyle attribute of the GraphData1 ... GraphData*n* style elements in the current style.

THICKNESS= *n* *<units>*

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see [“Units of Measurement” on page 659](#).

Default: For ungrouped data, the default line thickness is specified by the LineThickness attribute of the GraphDataDefault style element in the current style. For grouped data, the default line thickness is specified by the LineThickness attribute of the GraphData1 ... GraphData*n* style elements in the current style.

Here is an example expression:

```
LINEATTRS=(Color= "light green" Pattern= MediumDash Thickness= 4)
```

Default: For ungrouped data, line attributes are derived from the GraphDataDefault style element. For grouped data, the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements are used.

Interactions:

This option takes effect only if the CONNECT= option is also specified.

This option is ignored if the GROUP= option is also specified.

MEANATTRS= *style-element* *<(options)>* | *(options)*

specifies the appearance of the marker that represents the mean in the box. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the markers. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphBoxMean style element in the current style. For

grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

SIZE= *n* <*units*>

specifies the size of the markers. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see [“Units of Measurement” on page 659](#).

Default: The default marker size is specified by the MarkerSize attribute of the GraphBoxMean style element in the current style.

SYMBOL= *symbol-name*

specifies the symbol for the markers. For a list of valid marker symbols, see [“Marker Symbols” on page 661](#).

Default: For ungrouped data, the default marker symbol is specified by the MarkerSymbol attribute of the GraphBoxMean style element in the current style. For grouped data, the default marker symbol is specified by the MarkerSymbol attribute of the GraphData1 ... GraphData*n* style elements in the current style.

WEIGHT= NORMAL | BOLD

specifies the marker weight of the marker.

Default: NORMAL

Default: GraphBoxMean is the default style element.

MEDIANATTRS= *style-element* <(options)> | (options)

specifies the appearance of the line that represents the median. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see [“Color-Naming Schemes” in Chapter 14 of SAS/GRAPH: Reference](#).

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphBoxMedian style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see [“Line Patterns” on page 663](#).

Default: For ungrouped data, the default line pattern is specified by the LineStyle attribute of the GraphBoxMedian style element in the current style. For grouped data, the default line pattern is specified by the LineStyle attribute of the GraphData1 ... GraphData*n* style elements in the current style.

THICKNESS= *n* <*units*>

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see [“Units of Measurement” on page 659](#).

Default: For ungrouped data, the default line thickness is specified by the LineThickness attribute of the GraphBoxMedian style element in the current style. For grouped data, the default line thickness is specified by the LineThickness attribute of the GraphData1 ... GraphData*n* style elements in the current style.

Default: GraphBoxMedian is the default style element.

Interaction: This option is ignored if the NOMEDIAN option is also specified.

MISSING

processes missing values as a valid category value and creates a box for it.

NAME= *“text-string”*

specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note: The name is case-sensitive. When you refer to the specified name in other statements, you must use the same exact capitalization.

NOCAPS

hides the cap lines for the whiskers.

NOMEAN

hides the mean marker.

NOMEDIAN

hides the median line.

NOOUTLIERS

hides the outliers from the plot.

NOTCHES

specifies that the boxes be notched. The endpoints of the notches are at the following computed locations:

$$median \pm 1.58 (IQR / \sqrt{N})$$

For a visual description of the parts of a box plot, see [“Details” on page 237](#).

OUTLIERATTRS= *style-element* <(options)> | (options)

specifies the appearance of the marker that represents the outliers. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the markers. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphOutlier style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

SIZE= *n* <*units*>

specifies the size of the markers. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see [“Units of Measurement” on page 659](#).

Default: The default marker size is specified by the MarkerSize attribute of the GraphOutlier style element in the current style.

SYMBOL= *symbol-name*

specifies the symbol for the markers. For a list of valid marker symbols, see [“Marker Symbols” on page 661](#).

Default: For ungrouped data, the default marker symbol is specified by the MarkerSymbol attribute of the GraphOutlier style element in the current style. For grouped data, the default marker symbol is specified by the

MarkerSymbol attribute of the GraphData1 ... GraphData*n* style elements in the current style.

WEIGHT= NORMAL | BOLD

specifies the marker weight of the marker.

Default: NORMAL

Default: GraphOutlier is the default style element.

PERCENTILE= 1 | 2 | 3 | 4 | 5

specifies a method for computing the percentiles for the plot. For descriptions of each method, see “Calculating Percentiles” in the UNIVARIATE Procedure chapter of *Base SAS Procedures Guide: Statistical Procedures*.

Default: 5

SPREAD

relocates outlier points that have identical values to prevent overlapping.

Note: This option has no effect if your data does not contain two or more outliers with identical values for the analysis variable.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the plot. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default: 0.0

WHISKERATTRS= *style-element* <(options)> | (options)

specifies the appearance of the whisker and cap lines. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphBoxWhisker style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see “Line Patterns” on page 663.

Default: For ungrouped data, the default line pattern is specified by the LineStyle attribute of the GraphBoxWhisker style element in the current style. For grouped data, the default line pattern is specified by the LineStyle attribute of the GraphData1 ... GraphData*n* style elements in the current style.

THICKNESS= *n* <*units*>

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: For ungrouped data, the default line thickness is specified by the LineThickness attribute of the GraphBoxWhisker style element in the current style. For grouped data, the default line thickness is specified by the LineThickness attribute of the GraphData1 ... GraphData*n* style elements in the current style.

Default: GraphBoxWhisker is the default style element.

Interaction: This option is ignored if the NOMEDIAN option is also specified.

Details

Horizontal and vertical box plots display the distribution of data by using a rectangular box and whiskers. Whiskers are lines that indicate a data range outside of the box.

Figure 6.7 Parts of a Box Plot

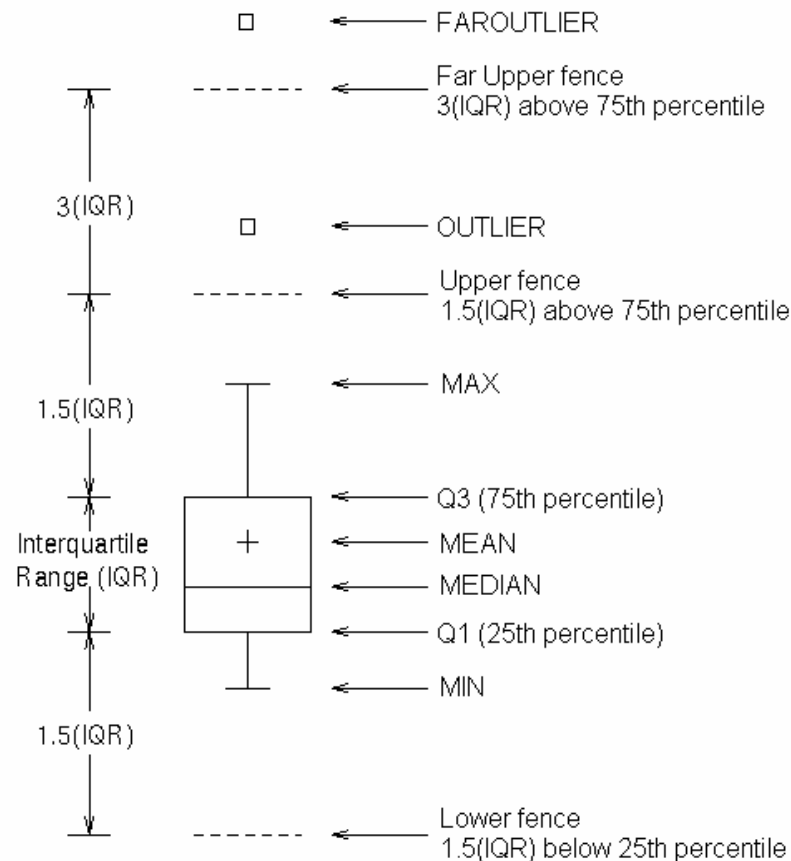


Figure 6.7 on page 237 shows a diagram of a vertical box plot. The bottom and top edges of the box indicate the intra-quartile range (IQR). That is, the range of values between the first and third quartiles (the 25th and 75th percentiles). The marker inside the box indicates the mean value. The line inside the box indicates the median value.

The elements that are outside the box are dependent on your options. By default, the whiskers that extend from each box indicate the range of values that are outside of the intra-quartile range. However, the values are close enough not to be considered outliers (a distance less than or equal to $1.5 \times IQR$). If you specify the EXTREME option, then the whiskers indicate the entire range of values, including outliers.

Any points that are a distance of more than $1.5 \times IQR$ from the box are considered to be outliers. By default, these points are indicated by markers. If you specify DATALABEL= option, then the outlier points have data labels. If you also specify the LABELFAR option, then only outliers that are $3 \times IQR$ from the box have data labels.

VECTOR Statement

Creates a vector plot that draws arrows from a point of origin to each data point.

Example: [“About Vector Plots” on page 31](#)

Syntax

VECTOR *X= numeric-variable Y= numeric-variable* *</ option(s)>*;

Summary of Optional Arguments

Label options

DATALABEL *<= variable>*

displays a label for each data point.

DATALABELATTRS *= style-element <(options)> | (options)*

specifies the appearance of the labels in the plot when you use the **DATALABEL=** option.

Plot options

ATTRID *= variable*

specifies the value of the ID variable in an attribute map data set.

GROUP *= variable*

specifies a variable that is used to group the data.

LEGENDLABEL *= “text-string”*

specifies a label that identifies the vector plot in the legend.

NAME *= “text-string”*

specifies a name for the plot.

TRANSPARENCY *= numeric-value*

specifies the degree of transparency for the lines.

Vector options

ARROWDIRECTION *= IN | OUT | BOTH*

specifies the location of the arrowheads for the vectors.

ARROWHEADSHAPE *= shape*

specifies the shape of the arrowheads for the vectors.

LINEATTRS *= style-element <(options)> | (options)*

specifies the appearance of the vector line.

NOARROWHEADS

removes the arrowheads from the vectors.

NOMISSINGGROUP

specifies that missing values of the group variable are not included in the plot.

XORIGIN *= numeric-value | numeric-variable*

specifies the X coordinate of the origin for the vectors.

YORIGIN *= numeric-value | numeric-variable*

specifies the Y coordinate of the origin for the vectors.

Required Arguments

X= *numeric-variable*

specifies a numeric variable for the x axis.

Y= *numeric-variable*

specifies numeric variable for the y axis.

Optional Arguments

ARROWDIRECTION= IN | OUT | BOTH

specifies the location of the arrowheads for the vectors. Specify one of the following:

IN

places the arrowheads at the origin of the vector.

OUT

places the arrowheads at the ending point of the vector.

BOTH


places arrowheads at both the origin and the ending point of the vector.

Default: OUT


ARROWHEADSHAPE= *shape*

specifies the shape of the arrowheads for the vectors. Specify one of the following:


OPEN

resembles the letter "V". 


CLOSED

an outline of a triangle. 

FILLED

a solid triangle. 

BARBED

a solid triangle with an indent at the base. 

Default: OPEN

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph. For more information, see [“Using SG Attribute Maps to Control Visual Attributes” on page 605](#).

DATALABEL <= *variable*>

displays a label for each data point. If you specify a variable, the values of that variable are used for the data labels. If you do not specify a variable, then the values of the Y variable are used for the data labels.

DATALABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the DATALABEL= option. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

Here is an example that specifies a style element:

DATALABELATTRS=GraphLabelText

options can be one or more of the following:

COLOR= *color*

specifies the color of the label characters. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphDataText style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

FAMILY= “*font-family*”

specifies the font family for the label characters. The SAS ODS styles use TrueType system fonts. For more information, see “TrueType Fonts Supplied by SAS” in Chapter 15 of *SAS Language Reference: Concepts*.

Default: The default font family is specified by the FontFamily attribute of the GraphDataText style element in the current style.

SIZE= *n* <*units*>

specifies the font size of the label characters. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default font size is specified by the FontSize attribute of the GraphDataText style element in the current style.

STYLE= ITALIC | NORMAL

specifies whether the label characters are italic (ITALIC) or normal (NORMAL).

Default: The default font style is specified by the FontStyle attribute of the GraphDataText style element in the current style.

WEIGHT= BOLD | NORMAL

specifies whether the label characters are bold (BOLD) or normal (NORMAL).

Default: The default font weight is specified by the FontWeight attribute of the GraphDataText style element in the current style.

Here is an example expression:

```
DATALABELATTRS=(Color=Green Family=Arial Size=8 Style=Italic Weight=Bold)
```

Default: The default style element is GraphDataText.

Interaction: This option has no effect unless the DATALABEL option is also specified.

GROUP= *variable*

specifies a variable that is used to group the data. A separate plot is created for each unique value of the grouping variable. The plot elements for each group value are automatically distinguished by different visual attributes.

LEGENDLABEL= “*text-string*”

specifies a label that identifies the vector plot in the legend. By default, the label of the Y variable or the group value for each plot element is used.

Interaction: The LEGENDLABEL= option has no effect if you also specify the GROUP= option in the same plot statement.

LINEATTRS= *style-element* <(options)> | (options)

specifies the appearance of the vector line. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphDataDefault style element in the current style.

For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see “Line Patterns” on page 663.

Default: For ungrouped data, the default line pattern is specified by the LineStyle attribute of the GraphDataDefault style element in the current style.

For grouped data, the default line pattern is specified by the LineStyle attribute of the GraphData1 ... GraphData*n* style elements in the current style.

THICKNESS= *n* *<units>*

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: For ungrouped data, the default line thickness is specified by the LineThickness attribute of the GraphDataDefault style element in the current style.

For grouped data, the default line thickness is specified by the LineThickness attribute of the GraphData1 ... GraphData*n* style elements in the current style.

NAME= “*text-string*”

specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note: The name is case-sensitive. When you refer to the specified name in other statements, you must use the same exact capitalization.

NOARROWHEADS

removes the arrowheads from the vectors.

NOMISSINGGROUP

specifies that missing values of the group variable are not included in the plot.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the lines. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default: 0.0

XORIGIN= *numeric-value* | *numeric-variable*

specifies the X coordinate of the origin for the vectors. You can specify either a numeric value or a numeric variable.

Default: 0

YORIGIN= *numeric-value* | *numeric-variable*

specifies the Y coordinate of the origin for the vectors. You can specify either a numeric value or a numeric variable.

Default: 0

VLINE Statement

Creates a vertical line chart (the line is horizontal). You can use the VLINE statement with the VBAR statement to create a bar-line chart.

Interaction: The VLINE statement can be combined only with other categorization plot statements in the SGPPANEL procedure. See [“Plot Type Compatibility” on page 81](#).

See:

Examples: [“About Line Charts” on page 47](#)
[“Example 4: Creating a Panel of Line Charts” on page 272](#)

Syntax

VLINE *category-variable* < / option(s) >;

Summary of Optional Arguments

Group options

CLUSTERWIDTH= *numeric-value*

specifies the width of the group clusters as a fraction of the midpoint spacing.

GROUP= *variable*

specifies a category variable to divide the values into groups.

GROUPDISPLAY= CLUSTER | OVERLAY

specifies how to display grouped lines.

GROUPORDER= ASCENDING | DESCENDING | DATA

specifies the ordering of lines within a group.

Label options

CURVELABEL <=“*text-string*”>

adds a label for the line plot.

CURVELABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the CURVELABEL= option.

CURVELABELPOS= MIN | MAX | START | END

specifies the location of the curve label.

DATALABEL <= *variable* >

displays a label for each data point.

DATALABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the DATALABEL= option.

Line options

BREAK

creates a break in the line for each missing value of the response variable.

DISCRETEOFFSET= *numeric-value*

specifies an amount to offset all lines from discrete category values.

LIMITATTRS= *style-element* <(options)> | (options)

specifies the appearance of the limit lines in the plot.

LIMITS= BOTH | LOWER | UPPER

adds limit lines to the plot.

LIMITSTAT= CLM | STDDEV | STDERR

specifies the statistic for the limit lines.

LINEATTRS= *style-element* <(options)> | (options)

specifies the appearance of the lines in the line plot.

MARKERATTRS= *style-element* <(options)> | (options)

specifies the appearance of the markers in the plot.

MARKERS

adds data point markers to the plot.

MISSING

processes missing values as a valid category value and creates a line for it.

NOSTATLABEL

removes the statistic name from the axis and legend labels.

NUMSTD= *n*

specifies the number of standard units for the limit lines, when you specify LIMITSTAT= STDDEV or LIMITSTAT= STDERR.

RESPONSE= *response-variable*

specifies a numeric response variable for the plot.

STAT= FREQ | MEAN | SUM

specifies the statistic for the vertical axis.

Plot options

ALPHA= *numeric-value*

specifies the confidence level for the confidence limits.

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set.

CATEGORYORDER= RESPASC | RESPDESC

specifies the order in which the response values are arranged.

FREQ= *numeric-variable*

specifies how many times observations are repeated for computational purposes.

LEGENDLABEL= "*text-string*"

specifies the label that identifies the line plot in the legend.

NAME= "*text-string*"

specifies a name for the plot.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the lines and markers.

URL= *character-variable*

specifies a character variable that contains URLs for Web pages to be displayed when parts of the plot are selected within an HTML page.

WEIGHT= *numeric-variable*

specifies how observations are weighted.

Required Argument

category-variable

specifies the variable that classifies the observations into distinct subsets.

Optional Arguments

ALPHA= *numeric-value*

specifies the confidence level for the confidence limits. Specify a number between 0.00 (100% confidence) and 1.00 (0% confidence).

Default: .05

Interactions:

This option has no effect if you do not specify LIMITSTAT=CLM.

If your plot is overlaid with other categorization plots, then the first ALPHA value that you specify is used for all of the plots.

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph. For more information, see [“Using SG Attribute Maps to Control Visual Attributes” on page 605](#).

BREAK

creates a break in the line for each missing value of the response variable.

Notes:

The observation is excluded from the graph when there is a missing value for the category variable without the specification of the MISSING option. (No break occurs in the line.)

The observation is excluded from the graph when there is a missing value for the FREQ variable.

CATEGORYORDER= RESPASC | RESPDESC

specifies the order in which the response values are arranged. Specify one of the following values:

RESPASC

sorts the response values in ascending order.

RESPDESC

sorts the response values in descending order.

Default: By default, the plot is sorted in ascending order based on the category values.

Restriction: This option takes effect only when the plot statement specifies a response variable and the axis for that variable is numeric. If the axis is not numeric, an error is generated and a message is written to the SAS log.

Requirement: This option requires that you configure the panel to use either one column or one row, depending on the orientation of your charts. Use the ROWS= or the COLUMNS= option in the PANELBY statement. If you do not use this option and your graph contains multiple cells, the specified sort order is not correctly applied to all cells.

Interaction: If a group variable is specified in the plot statement, the response sort option is ignored and a note is generated in the log. (You can use a BY statement in the procedure if you want to group data.)

Note: If CATEGORYORDER is specified in multiple statements, the procedure sorts by the last statement in which it is specified.

CLUSTERWIDTH= *numeric-value*

specifies the width of the group clusters as a fraction of the midpoint spacing. Specify a value from 0.0 (narrowest) to 1.0 (widest).

Default: 0.8

Interaction: This option is applicable only when the GROUP option is specified, when GROUPDISPLAY=CLUSTER, and when the category axis is discrete.

CURVELABEL <=“*text-string*”>

adds a label for the line plot. You can also specify the label text. If you do not specify a label, the label from the response variable is used.

CURVELABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the CURVELABEL= option. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

Here is an example that specifies a style element:

CURVELABELATTRS=GraphTitleText

options can be one or more of the following:

COLOR= *color*

specifies the color of the line label characters. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphValueText style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

FAMILY= “*font-family*”

specifies the font family for the label characters. The SAS ODS styles use TrueType system fonts. For more information, see “TrueType Fonts Supplied by SAS” in Chapter 15 of *SAS Language Reference: Concepts*.

Default: The default font family is specified by the FontFamily attribute of the GraphValueText style element in the current style.

Restriction: You cannot specify a list of font family names.

SIZE= *n* <*units*>

specifies the font size of the label characters. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default font size is specified by the FontSize attribute of the GraphValueText style element in the current style.

STYLE= ITALIC | NORMAL

specifies whether the label characters are italic or normal.

Default: The default font style is specified by the FontStyle attribute of the GraphValueText style element in the current style.

WEIGHT= BOLD | NORMAL

specifies whether the label characters are bold or normal.

Default: The default font weight is specified by the FontWeight attribute of the GraphValueText style element in the current style.

Here is an example expression:

CURVELABELATTRS=(Color=Green Family=Arial Size=8 Style=Italic Weight=Bold)

Default: The default style element is GraphValueText.

Interaction: This option has no effect unless the CURVELABEL option is also specified.

CURVELABELPOS= MIN | MAX | START | END

specifies the location of the curve label. Specify one of the following values:

MIN

places the label at the part of the curve closest to the minimum X axis value.

MAX

places the label at the part of the curve closest to the maximum X axis value.

START

places the curve label at the first point on the curve.

END

places the curve label at the last point on the curve.

Default: END

Interaction: This option has no effect unless the CURVELABEL option is also specified.

DATALABEL <= variable>

displays a label for each data point. If you specify a variable, then the values of that variable are used for the data labels. If you do not specify a variable, then the values of the calculated response are used for the data labels.

DATALABELATTRS= style-element <(options)> | (options)

specifies the appearance of the labels in the plot when you use the DATALABEL= option. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

Here is an example that specifies a style element:

DATALABELATTRS=GraphLabelText

options can be one or more of the following:

COLOR= *color*

specifies the color of the label characters. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphDataText style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

FAMILY= “*font-family*”

specifies the font family for the label characters. The SAS ODS styles use TrueType system fonts. For more information, see “TrueType Fonts Supplied by SAS” in Chapter 15 of *SAS Language Reference: Concepts*.

Default: The default font family is specified by the FontFamily attribute of the GraphDataText style element in the current style.

SIZE= *n* <*units*>

specifies the font size of the label characters. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default font size is specified by the FontSize attribute of the GraphDataText style element in the current style.

STYLE= ITALIC | NORMAL

specifies whether the label characters are italic (ITALIC) or normal (NORMAL).

Default: The default font style is specified by the `FontStyle` attribute of the `GraphDataText` style element in the current style.

WEIGHT= BOLD | NORMAL

specifies whether the label characters are bold (BOLD) or normal (NORMAL).

Default: The default font weight is specified by the `FontWeight` attribute of the `GraphDataText` style element in the current style.

Here is an example expression:

```
DATALABELATTRS=(Color=Green Family=Arial Size=8 Style=Italic Weight=Bold)
```

Default: The default style element is `GraphDataText`.

Interaction: This option has no effect unless the `DATALABEL` option is also specified.

DISCRETEOFFSET= *numeric-value*

specifies an amount to offset all lines from discrete category values. Specify a value from -0.5 (left offset) to +0.5 (right offset).

Default: 0.0 (no offset)

Requirement: This option is applicable only when the category axis is discrete.

FREQ= *numeric-variable*

specifies how many times observations are repeated for computational purposes. Each observation is repeated n times for computational purposes, where n is the value of the numeric variable. If n is not an integer, then it is truncated to an integer. If n is less than 1 or missing, then it is excluded from the analysis.

Interaction: If your plot is overlaid with other categorization plots, then the first FREQ variable that you specified is used for all of the plots.

GROUP= *variable*

specifies a category variable to divide the values into groups. A separate plot is created for each unique value of the category variable.

Interaction: If you specify more than one categorization plot statement, then all of the plots must specify the same GROUP variable. If you do not specify the same GROUP= option for all of the categorization plots, then the GROUP= option has no effect.

GROUPDISPLAY= CLUSTER | OVERLAY

specifies how to display grouped lines.

CLUSTER

grouped items are drawn adjacent to each other.

OVERLAY

grouped items are drawn at the exact coordinate and might overlap. Each group is represented by unique visual attributes derived from the `GraphData1...` `GraphData n` style elements in the current style.

Default: OVERLAY

Restriction: GROUPDISPLAY=CLUSTER can take effect only when the category axis is discrete. If necessary, use a TYPE=DISCRETE option on the axis statement.

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

GROUPORDER= ASCENDING | DESCENDING | DATA

specifies the ordering of lines within a group.

ASCENDING

displays each group in ascending group value.

DESCENDING

displays each group in descending group value.

DATA

displays each group in data order of the group variable.

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

LEGENDLABEL= *“text-string”*

specifies the label that identifies the line plot in the legend. By default, the label of the response variable is used. If there is no response variable label, then the name of the response variable and the computed statistic (SUM or MEAN) are used. If you do not specify a response variable, then the legend label is “Frequency”.

Interaction: The LEGENDLABEL= option has no effect if you also specify the GROUP= option in the same plot statement.

LIMITATTRS= *style-element* <(options)> | (options)

specifies the appearance of the limit lines in the plot. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: The default color is specified by the ContrastColor attribute of the GraphError style element in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see “Line Patterns” on page 663.

Default: The default line pattern is specified by the LineStyle attribute of the GraphError style element in the current style.

THICKNESS= *n* <units>

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default line thickness is specified by the LineThickness attribute of the GraphError style element in the current style.

LIMITS= BOTH | LOWER | UPPER

adds limit lines to the plot. Specify one of the following values:

BOTH

adds lower and upper limit lines to the plot.

LOWER

adds lower limit lines to the plot.

UPPER

adds upper limit lines to the plot.

Default: By default, no limit lines are displayed. However, if you specify the LIMITSTAT= option, then the default is BOTH.

Interaction: Limit lines are displayed only when you specify STAT= MEAN.

LIMITSTAT= CLM | STDDEV | STDERR

specifies the statistic for the limit lines. Specify one of the following statistics:

CLM
confidence limits

STDDEV
standard deviation

STDERR
standard error

Default: CLM

Interaction: If you specify the LIMITSTAT= option, then the default value for the LIMITS= option is BOTH.

LINEATTRS= *style-element* <(options)> | (options)

specifies the appearance of the lines in the line plot. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphDataDefault style element in the current style.

For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see “Line Patterns” on page 663.

Default: For ungrouped data, the default line pattern is specified by the LineStyle attribute of the GraphDataDefault style element in the current style.

For grouped data, the default line pattern is specified by the LineStyle attribute of the GraphData1 ... GraphData*n* style elements in the current style.

THICKNESS= *n* <*units*>

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: For ungrouped data, the default line thickness is specified by the LineThickness attribute of the GraphDataDefault style element in the current style.

For grouped data, the default line thickness is specified by the LineThickness attribute of the GraphData1 ... GraphData*n* style elements in the current style.

MARKERATTRS= *style-element* <(options)> | (options)

specifies the appearance of the markers in the plot. This option has no effect unless you also specify the MARKERS option. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the markers. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphDataDefault style element in the current style.

For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

SIZE= *n* <*units*>

specifies the size of the markers. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see [“Units of Measurement” on page 659](#).

Default: The default marker size is specified by the MarkerSize attribute of the GraphDataDefault style element in the current style.

SYMBOL= *symbol-name*

specifies the symbol for the markers. For a list of valid marker symbols, see [“Marker Symbols” on page 661](#).

Default: For ungrouped data, the default marker symbol is specified by the MarkerSymbol attribute of the GraphDataDefault style element in the current style.

For grouped data, the default marker symbol is specified by the MarkerSymbol attribute of the GraphData1 ... GraphData*n* style elements in the current style.

MARKERS

adds data point markers to the plot.

MISSING

processes missing values as a valid category value and creates a line for it.

NAME= *“text-string”*

specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note: The name is case-sensitive. When you refer to the specified name in other statements, you must use the same exact capitalization.

NOSTATLABEL

removes the statistic name from the axis and legend labels.

NUMSTD= *n*

specifies the number of standard units for the limit lines, when you specify LIMITSTAT= STDDEV or LIMITSTAT= STDERR. You can specify any positive number, including decimals.

Default: 1

RESPONSE= *response-variable*

specifies a numeric response variable for the plot. The summarized values of the response variable are displayed on the vertical axis.

STAT= FREQ | MEAN | SUM

specifies the statistic for the vertical axis. Specify one of the following statistics:

FREQ

the frequencies for the category variable. This is the default value when you do not specify the RESPONSE= option.

MEAN

the mean of the response variable.

SUM

the sum of the response variable. This is the default value when you specify the RESPONSE= option.

Interaction: If you do not specify the RESPONSE= option, then only the FREQ statistic can be used. If you specify the RESPONSE= option, then you can use either the SUM or MEAN statistics.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the lines and markers. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default: 0.0

URL= *character-variable*

specifies a character variable that contains URLs for Web pages to be displayed when parts of the plot are selected within an HTML page.

Default: By default, no HTML links are created.

Interactions:

This option affects graphics output that is created through the ODS HTML destination only. For more information about ODS destinations, see [“Specifying a Destination” on page 649](#).

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For more information, see [“Using the ODS GRAPHICS Statement” on page 651](#).

WEIGHT= *numeric-variable*

specifies how observations are weighted. Each observation is weighted by a factor of w for computational purposes, where w is the value of the numeric variable. w can be any numeric value. If w is 0, negative or missing, then that observation is excluded from the analysis.

Interaction: If your plot is overlaid with other categorization plots, then the first WEIGHT variable that you specified is used for all of the plots.

COLAXIS Statement

Specifies the axis options for each X axis in the panel. You can control the features of the axis (for example, the axis label, grid lines, and minor tick marks). You can also control the structure of the axis (for example, the data range, data type, and tick mark values).

Syntax

COLAXIS *option(s)*;

Summary of Optional Arguments

ALTERNATE

adds reference ticks to each side of the panel and alternates the tick values for each row or column between the two sides.

DISCRETEORDER= DATA | FORMATTED | UNFORMATTED

specifies the order in which discrete tick values are placed on the axis.

DISPLAY= ALL | NONE | (*options*)

specifies which features of the axis are displayed.

FITPOLICY= *policy-value*

specifies the method that is used to fit tick mark values on a horizontal axis when there is not enough room to draw them normally.

GRID

creates grid lines at each tick on the axis.

INTEGER

specifies that only integers are used for tick mark values.

INTERVAL= *interval-value*

specifies the tick interval for a time axis.

LABEL= "*text-string*"

specifies a label for the axis.

LABELATTRS= *style-element* <(*options*)> | (*options*)

specifies the appearance of the axis labels.

LOGBASE= 2 | 10 | e

specifies the base value for the logarithmic scale.

LOGSTYLE= LINEAR | LOGEXPAND | LOGEXPONENT

specifies how to scale and format the values for the major tick marks for logarithmic axes.

MAX= *numeric-value*

specifies the maximum value for the axis.

MIN= *numeric-value*

specifies the minimum value for the axis.

MINOR

adds minor tick marks to a log or time axis.

NOTIMESPLIT

prevents a time axis from splitting the time, date, or datetime values into two rows.

OFFSETMAX= *numeric-value*

specifies an offset that follows the highest data value on the axis.

OFFSETMIN= *numeric-value*

specifies an offset that precedes the lowest data value on the axis.

REFTICKS <= (*options*)>

adds tick marks to the side of the panel that is opposite from the specified axis.

REVERSE

specifies that the tick values are displayed in reverse (descending) order.

THRESHOLDMAX= *numeric-value*

Specifies a threshold for displaying one more tick mark at the high end of the axis.

THRESHOLDMIN= *numeric-value*

Specifies a threshold for displaying one more tick mark at the low end of the axis.

TICKVALUEFORMAT= DATA | *SAS-format*

specifies the format for the axis tick values.

TYPE= DISCRETE | LINEAR | LOG | TIME

specifies the type of axis.

VALUEATTRS= *style-element* <(*options*)> | (*options*)

specifies the appearance of the axis tick value labels.

VALUES= (*values-list*)

specifies the values for the ticks on the axis.

VALUESHINT

specifies that the minimum and maximum axis values are determined independently of the values that you specify in the VALUES= option.

Optional Arguments

ALTERNATE

adds reference ticks to each side of the panel and alternates the tick values for each row or column between the two sides.

DISCRETEORDER= DATA | FORMATTED | UNFORMATTED

specifies the order in which discrete tick values are placed on the axis. Specify one of the following values:

DATA

places the values in the order in which they appear in the data.

FORMATTED

sorts the formatted values in ascending character order.

UNFORMATTED

sorts the unformatted values in ascending character order.

Default: UNFORMATTED

Restriction: This option affects only box plots, dot plots, bar charts, and line plots, or for any axis where TYPE=DISCRETE.

DISPLAY= ALL | NONE | (*options*)

specifies which features of the axis are displayed. ALL displays all of the features of the axis. NONE specifies that none of the features of the axis are displayed.

You can also hide specific features by specifying options. *Options* can be any of the following:

NOLABEL

hides the axis label

NOLINE

hides the axis line

NOTICKS

hides the tick marks on the axis

NOVALUES

hides the tick mark values on the axis

Default: ALL

Interaction: If the ALTERNATE option is also specified, the DISPLAY option affects both the primary and the alternate axis.

FITPOLICY= *policy-value*

specifies the method that is used to fit tick mark values on a horizontal axis when there is not enough room to draw them normally. Select one of the following values:

ROTATE

rotates the value text 45 degrees. This is the default for discrete axes.

ROTATETHIN

attempts to use ROTATE, and then THIN to fit the values.

STAGGER

shifts the values up and down.

STAGGERROTATE

attempts to use STAGGER, and then ROTATE to fit the values.

STAGGERTHIN

attempts to use STAGGER, and then THIN to fit the values.

THIN

removes some of the values from the axis. This is the default for linear and time axes.

Restriction: This option does not affect logarithmic axes.

Note: This option has no effect if used with the vertical (row) axes; only the THIN fit policy is used for vertical axes.

GRID

creates grid lines at each tick on the axis.

INTEGER

specifies that only integers are used for tick mark values. This option affects only linear axes.

INTERVAL= *interval-value*

specifies the tick interval for a time axis. Specify one of the following values:

AUTO

determines the tick interval automatically based on the data.

SECOND

places ticks one second apart. The default tick value format is TIME8.

MINUTE

places ticks one minute apart. The default tick value format is TIME8.

HOURL

places ticks one hour apart. The default tick value format is TIME8.

DAY

places ticks one day apart. The default tick value format is DATE9.

WEEK

places ticks one week apart. The default tick value format is DATE9.

TENDAY

places ticks ten days apart. The default tick value format is DATE9.

SEMIMONTH

places ticks at the first and sixteenth day of each month. The default tick value format is DATE9.

MONTH

places ticks one month apart. The default tick value format is MONYY7.

QUARTER

places ticks three months apart. The default tick value format is YYQC6.

SEMIYEAR

places ticks six months apart. The default tick value format is MONYY7.

YEAR

places ticks one year apart. The default tick value format is YEAR4.

Default: AUTO

LABEL= "*text-string*"

specifies a label for the axis.

LABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the axis labels. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the label characters. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphLabelText style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

FAMILY= “*font-family*”

specifies the font family for the label characters. The SAS ODS styles use TrueType system fonts. For more information, see “TrueType Fonts Supplied by SAS” in Chapter 15 of *SAS Language Reference: Concepts*.

Default: The default font family is specified by the FontFamily attribute of the GraphLabelText style element in the current style.

Restriction: You cannot specify a list of font family names.

SIZE= *n* <*units*>

specifies the font size of the label characters. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default font size is specified by the FontSize attribute of the GraphValueText style element in the current style.

STYLE= ITALIC | NORMAL

specifies whether the label characters are italic or normal.

Default: The default font style is specified by the FontStyle attribute of the GraphLabelText style element in the current style.

WEIGHT= BOLD | NORMAL

specifies whether the label characters are bold or normal.

Default: The default font weight is specified by the FontWeight attribute of the GraphLabelText style element in the current style.

Interaction: This option has no effect unless CURVELABEL is also specified.

LOGBASE= 2 | 10 | *e*

specifies the base value for the logarithmic scale.

Default: 10

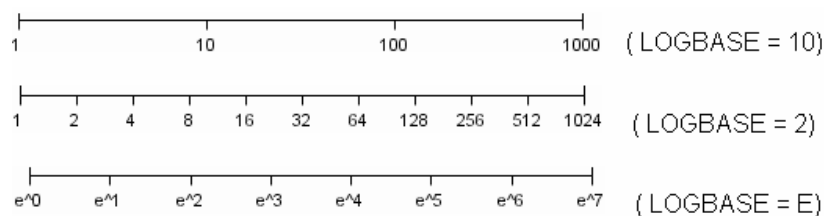
Interaction: This option has no effect unless you also specify TYPE=LOG.

LOGSTYLE= LINEAR | LOGEXPAND | LOGEXPONENT

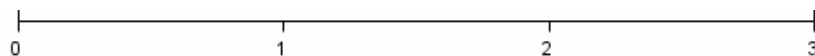
specifies how to scale and format the values for the major tick marks for logarithmic axes. Specify one of the following values:

LOGEXPAND

places the tick marks at integer powers of the base. For example, if you specified LOGBASE=2, the tick marks might be at 1, 2, 4, 8, 16. See [Figure 6.8 on page 256](#).

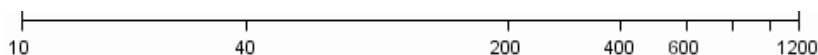
Figure 6.8 Graph Axes with LOGEXPAND**LOGEXPONENT**

places the tick marks at integer powers of the base, but identifies the values by the exponent. For example, if you specified **LOGBASE=10**, the tick marks might be at 1, 10, 100, 1000, but the tick values would read 0, 1, 2, 3. See [Figure 6.9 on page 256](#).

Figure 6.9 An Axis with LOGEXPONENT**LINEAR**

places the tick marks at uniform linear intervals, but spaces them logarithmically. In some cases an intermediate tick mark is placed between the first and second marks.

For example, if the data on this axis range from 14 to 1154, and you specify **LOGBASE=10**, then the tick marks might be at 10, 40, 200, 400, 600, 800, 1000, 1200. See [Figure 6.10 on page 256](#).

Figure 6.10 An Axis with LINEAR

Default: LOGEXPAND

Interaction: This option has no effect unless you also specify TYPE=LOG.

MAX= *numeric-value*

specifies the maximum value for the axis.

Restriction: This option affects linear, log, and time axes only.

Interaction: This option has no effect if you specify the VALUES= option and you do not also specify the VALUESHINT option.

MIN= *numeric-value*

specifies the minimum value for the axis.

Restriction: This option affects linear, log, and time axes only.

Interaction: This option has no effect if you specify the VALUES= option and you do not also specify the VALUESHINT option.

MINOR

adds minor tick marks to a log or time axis.

Restriction: This option has no effect on discrete or linear axes.

Interaction: This option has no effect if you specify the VALUES= option.

NOTIMESPLIT

prevents a time axis from splitting the time, date, or datetime values into two rows.

Restriction: This option applies to time axes only.

OFFSETMAX= *numeric-value*

specifies an offset that follows the highest data value on the axis. Specify a value between 0 and 1.

The value represents the offset as a proportion to the total length of the axis. For a continuous axis, the offset follows the highest data value or highest tick value, whichever is greater. For a discrete axis, the offset is applied to the end of the axis farther from the origin.

Default: The offset space is determined automatically based on the data values, tick mark values, markers, and labels that are inside of the plot area.

OFFSETMIN= *numeric-value*

specifies an offset that precedes the lowest data value on the axis. Specify a value between 0 and 1.

The value represents the offset as a proportion to the total length of the axis. For a continuous axis, the offset precedes the lowest data value or lowest tick value, whichever is less. For a discrete axis, the offset is applied to the end of the axis nearer to the origin.

Default: The offset space is determined automatically based on the data values, tick mark values, markers, and labels that are inside of the plot area.

REFTICKS <= (*options*)>

adds tick marks to the side of the panel that is opposite from the specified axis. You can also specify *options*:

LABEL

in addition to the tick marks, displays the axis label.

VALUES

in addition to the tick marks, displays the values that are represented by the tick marks.

REVERSE

specifies that the tick values are displayed in reverse (descending) order.

THRESHOLDMAX= *numeric-value*

Specifies a threshold for displaying one more tick mark at the high end of the axis.

THRESHOLDMIN= *numeric-value*

Specifies a threshold for displaying one more tick mark at the low end of the axis.

TICKVALUEFORMAT= DATA | *SAS-format*

specifies the format for the axis tick values. You can either specify a SAS format or specify DATA, which indicates that the format from axis variable is used.

TYPE= DISCRETE | LINEAR | LOG | TIME

specifies the type of axis. Specify one of the following values:

DISCRETE

specifies an axis with discrete values. If a character variable is assigned to an axis, then that the default type for that axis is discrete. In addition, all categorization plots use a discrete axis for the category variable.

LINEAR

specifies a linear scale for the axis. This is the default axis type for numeric variables, except when the data is discrete or when the numeric variable has a date or time format.

LOG

specifies a logarithmic scale for the axis. This axis type is never a default.

Interaction: A logarithmic scale cannot be used with linear regression plots (REG statement where DEGREE=1).

TIME

specifies a time scale for the axis. If the variable assigned to an axis has a time, date, or datetime format associated with it, then time is the default axis type.

VALUEATTRS= *style-element* <(options)> | (options)

specifies the appearance of the axis tick value labels. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the label characters. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphValueText style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

FAMILY= “*font-family*”

specifies the font family for the label characters. The SAS ODS styles use TrueType system fonts. For more information, see “TrueType Fonts Supplied by SAS” in Chapter 15 of *SAS Language Reference: Concepts*.

Default: The default font family is specified by the FontFamily attribute of the GraphValueText style element in the current style.

Restriction: You cannot specify a list of font family names.

SIZE= *n* <*units*>

specifies the font size of the label characters. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default font size is specified by the FontSize attribute of the GraphValueText style element in the current style.

STYLE= ITALIC | NORMAL

specifies whether the label characters are italic or normal.

Default: The default font style is specified by the FontStyle attribute of the GraphValueText style element in the current style.

WEIGHT= BOLD | NORMAL

specifies whether the label characters are bold or normal.

Default: The default font weight is specified by the FontWeight attribute of the GraphValueText style element in the current style.

VALUES= (*values-list*)

specifies the values for the ticks on the axis.

For values on a linear axis, the values list can be one of the following:

value <...*value-n*>

creates ticks for specific values. For example, **VALUES= (0 50 100)** places tick marks at 0, 50, and 100.

value-1 TO value-2 BY increment-value

creates ticks for a range of values. The start of the value range is specified by *value-1* and the end of the range is specified by *value-2*. The *increment-value* specifies the interval between the ticks. For example, **VALUES= (0 to 100 by 50)** creates tick marks at 0, 50, and 100.

<value ... value-n> value-1 TO value-2 BY increment-value <value ... value-n>

creates ticks for specific values, and also creates ticks for a range of values. The start of the value range is specified by *value-1* and the end of the range is specified by *value-2*. The *increment-value* specifies the interval between the ticks.

For example, **VALUES= (-5 10 to 50 by 20 75)** creates tick marks at -5, 10, 30, 50, and 75.

For values on a time axis, the values list can be one of the following:

value <...value-n>

creates ticks for specific values. For example, **VALUES= ("25MAY08"d "04JUL08"d "23AUG08"d)** places tick marks at 25MAY08, 04JUL08, and 23AUG08.

value-1 TO value-2 BY increment-value

creates ticks for a range of values. The start of the value range is specified by *value-1* and the end of the range is specified by *value-2*. The *increment-value* specifies the interval between the ticks. For example, **VALUES= ("01JAN08"d to "01MAY08"d by month)** creates tick marks at 01JAN08, 01FEB08, 01MAR08, 01APR08, and 01MAY08.

For a list of the interval values that you can specify, see the **INTERVAL=** option.

Restrictions:

This option has no effect on discrete and logarithmic axes.

If your **VALUES=** option creates more than 1000 values, then the option has no effect.

Interaction: If a custom format is applied to the value, the raw value is plotted first and then formatted.

VALUESHINT

specifies that the minimum and maximum axis values are determined independently of the values that you specify in the **VALUES=** option. The values from the **VALUES=** option are displayed only if they are located between the minimum and maximum values.

Interaction: This option has no effect unless you also specify the **VALUES=** option.

ROWAXIS Statement

Specifies the axis options for each Y axis in the panel. You can control the features of the axis (for example, the axis label, grid lines, and minor tick marks). You can also control the structure of the axis (for example, the data range, data type, and tick mark values).

Syntax

ROWAXIS *option(s)*;

Summary of Optional Arguments

ALTERNATE

adds reference ticks to each side of the panel and alternates the tick values for each row or column between the two sides.

DISCRETEORDER= DATA | FORMATTED | UNFORMATTED

specifies the order in which discrete tick values are placed on the axis.

DISPLAY= ALL | NONE | (*options*)

specifies which features of the axis are displayed.

GRID

creates grid lines at each tick on the axis.

INTEGER

specifies that only integers are used for tick mark values.

INTERVAL= *interval-value*

specifies the tick interval for a time axis.

LABEL= "*text-string*"

specifies a label for the axis.

LABELATTRS= *style-element* <(*options*)> | (*options*)

specifies the appearance of the axis labels.

LOGBASE= 2 | 10 | e

specifies the base value for the logarithmic scale.

LOGSTYLE= LINEAR | LOGEXPAND | LOGEXPONENT

specifies how to scale and format the values for the major tick marks for logarithmic axes.

MAX= *numeric-value*

specifies the maximum value for the axis.

MIN= *numeric-value*

specifies the minimum value for the axis.

MINOR

adds minor tick marks to a log or time axis.

NOTIMESPLIT

prevents a time axis from splitting the time, date, or datetime values into two rows.

OFFSETMAX= *numeric-value*

specifies an offset that follows the highest data value on the axis.

OFFSETMIN= *numeric-value*

specifies an offset that precedes the lowest data value on the axis.

REFTICKS <= (*options*)>

adds tick marks to the side of the panel that is opposite from the specified axis.

REVERSE

specifies that the tick values are displayed in reverse (descending) order.

THRESHOLDMAX= *numeric-value*

Specifies a threshold for displaying one more tick mark at the high end of the axis.

THRESHOLDMIN= *numeric-value*

Specifies a threshold for displaying one more tick mark at the low end of the axis.

TICKVALUEFORMAT= DATA | *SAS-format*

specifies the format for the axis tick values.

TYPE= DISCRETE | LINEAR | LOG | TIME

specifies the type of axis.

VALUEATTRS= *style-element* <(options)> | (options)

specifies the appearance of the axis tick value labels.

VALUES= (*values-list*)

specifies the values for the ticks on the axis.

VALUESHINT

specifies that the minimum and maximum axis values are determined independently of the values that you specify in the VALUES= option.

Optional Arguments

ALTERNATE

adds reference ticks to each side of the panel and alternates the tick values for each row or column between the two sides.

DISCRETEORDER= DATA | FORMATTED | UNFORMATTED

specifies the order in which discrete tick values are placed on the axis. Specify one of the following values:

DATA

places the values in the order in which they appear in the data.

FORMATTED

sorts the formatted values in ascending character order.

UNFORMATTED

sorts the unformatted values in ascending character order.

Default: UNFORMATTED

Restriction: This option affects only box plots, dot plots, bar charts, and line plots, or for any axis where TYPE=DISCRETE.

DISPLAY= ALL | NONE | (options)

specifies which features of the axis are displayed. ALL displays all of the features of the axis. NONE specifies that none of the features of the axis are displayed.

You can also hide specific features by specifying options. *Options* can be any of the following:

NOLABEL

hides the axis label

NOLINE

hides the axis line

NOTICKS

hides the tick marks on the axis

NOVALUES

hides the tick mark values on the axis

Default: ALL

Interaction: If the ALTERNATE option is also specified, the DISPLAY option affects both the primary and the alternate axis.

GRID

creates grid lines at each tick on the axis.

INTEGER

specifies that only integers are used for tick mark values. This option affects only linear axes.

INTERVAL= *interval-value*

specifies the tick interval for a time axis. Specify one of the following values:

AUTO

determines the tick interval automatically based on the data.

SECOND

places ticks one second apart. The default tick value format is TIME8.

MINUTE

places ticks one minute apart. The default tick value format is TIME8.

HOURL

places ticks one hour apart. The default tick value format is TIME8.

DAY

places ticks one day apart. The default tick value format is DATE9.

WEEK

places ticks one week apart. The default tick value format is DATE9.

TENDAY

places ticks ten days apart. The default tick value format is DATE9.

SEMIMONTH

places ticks at the first and sixteenth day of each month. The default tick value format is DATE9.

MONTH

places ticks one month apart. The default tick value format is MONYY7.

QUARTER

places ticks three months apart. The default tick value format is YYQC6.

SEMIYEAR

places ticks six months apart. The default tick value format is MONYY7.

YEAR

places ticks one year apart. The default tick value format is YEAR4.

Default: AUTO

LABEL= "*text-string*"

specifies a label for the axis.

LABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the axis labels. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the label characters. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see "Color-Naming Schemes" in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphLabelText style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

FAMILY= “font-family”

specifies the font family for the label characters. The SAS ODS styles use TrueType system fonts. For more information, see “TrueType Fonts Supplied by SAS” in Chapter 15 of *SAS Language Reference: Concepts*.

Default: The default font family is specified by the FontFamily attribute of the GraphLabelText style element in the current style.

Restriction: You cannot specify a list of font family names.

SIZE= *n* <units>

specifies the font size of the label characters. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default font size is specified by the FontSize attribute of the GraphValueText style element in the current style.

STYLE= ITALIC | NORMAL

specifies whether the label characters are italic or normal.

Default: The default font style is specified by the FontStyle attribute of the GraphLabelText style element in the current style.

WEIGHT= BOLD | NORMAL

specifies whether the label characters are bold or normal.

Default: The default font weight is specified by the FontWeight attribute of the GraphLabelText style element in the current style.

Interaction: This option has no effect unless CURVELABEL is also specified.

LOGBASE= 2 | 10 | e

specifies the base value for the logarithmic scale.

Default: 10

Interaction: This option has no effect unless you also specify TYPE=LOG.

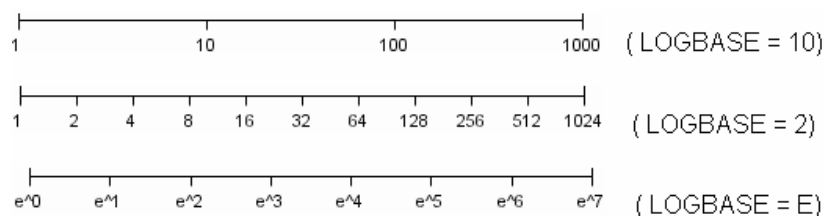
LOGSTYLE= LINEAR | LOGEXPAND | LOGEXPONENT

specifies how to scale and format the values for the major tick marks for logarithmic axes. Specify one of the following values:

LOGEXPAND

places the tick marks at integer powers of the base. For example, if you specified **LOGBASE=2**, the tick marks might be at 1, 2, 4, 8, 16. See Figure 6.8 on page 256.

Figure 6.11 Graph Axes with LOGEXPAND



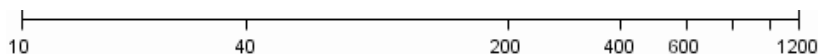
LOGEXPONENT

places the tick marks at integer powers of the base, but identifies the values by the exponent. For example, if you specified **LOGBASE=10**, the tick marks might be at 1, 10, 100, 1000, but the tick values would read 0, 1, 2, 3. See Figure 6.9 on page 256.

Figure 6.12 An Axis with LOGEXPONENT**LINEAR**

places the tick marks at uniform linear intervals, but spaces them logarithmically. In some cases an intermediate tick mark is placed between the first and second marks.

For example, if the data on this axis range from 14 to 1154, and you specify **LOGBASE=10**, then the tick marks might be at 10, 40, 200, 400, 600, 800, 1000, 1200. See [Figure 6.10 on page 256](#).

Figure 6.13 An Axis with LINEAR**Default:** LOGEXPAND

Interaction: This option has no effect unless you also specify **TYPE=LOG**.

MAX= numeric-value

specifies the maximum value for the axis.

Restriction: This option affects linear, log, and time axes only.

Interaction: This option has no effect if you specify the **VALUES=** option and you do not also specify the **VALUESHINT** option.

MIN= numeric-value

specifies the minimum value for the axis.

Restriction: This option affects linear, log, and time axes only.

Interaction: This option has no effect if you specify the **VALUES=** option and you do not also specify the **VALUESHINT** option.

MINOR

adds minor tick marks to a log or time axis.

Restriction: This option has no effect on discrete or linear axes.

Interaction: This option has no effect if you specify the **VALUES=** option.

NOTIMESPLIT

prevents a time axis from splitting the time, date, or datetime values into two rows.

Restriction: This option applies to time axes only.

OFFSETMAX= numeric-value

specifies an offset that follows the highest data value on the axis. Specify a value between 0 and 1.

The value represents the offset as a proportion to the total length of the axis. For a continuous axis, the offset follows the highest data value or highest tick value, whichever is greater. For a discrete axis, the offset is applied to the end of the axis farther from the origin.

Default: The offset space is determined automatically based on the data values, tick mark values, markers, and labels that are inside of the plot area.

Interaction: For **HBOX**, **HBAR**, **HLINE**, and **DOT** plot statements, the Y axis is reversed by default, so the axis origin is at the top. For these plots, **OFFSETMAX=** creates an offset at the end of the Y axis nearer to the origin.

OFFSETMIN= *numeric-value*

specifies an offset that precedes the lowest data value on the axis. Specify a value between 0 and 1.

The value represents the offset as a proportion to the total length of the axis. For a continuous axis, the offset precedes the lowest data value or lowest tick value, whichever is less. For a discrete axis, the offset is applied to the end of the axis nearer to the origin.

Default: The offset space is determined automatically based on the data values, tick mark values, markers, and labels that are inside of the plot area.

Interaction: For HBOX, HBAR, HLINE, and DOT plot statements, the Y axis is reversed by default, so the axis origin is at the top. For these plots, OFFSETMIN= creates an offset at the end of the Y axis farther from the origin.

REFTICKS <= (*options*)>

adds tick marks to the side of the panel that is opposite from the specified axis. You can also specify *options*:

LABEL

in addition to the tick marks, displays the axis label.

VALUES

in addition to the tick marks, displays the values that are represented by the tick marks.

REVERSE

specifies that the tick values are displayed in reverse (descending) order.

THRESHOLDMAX= *numeric-value*

Specifies a threshold for displaying one more tick mark at the high end of the axis.

THRESHOLDMIN= *numeric-value*

Specifies a threshold for displaying one more tick mark at the low end of the axis.

TICKVALUEFORMAT= DATA | *SAS-format*

specifies the format for the axis tick values. You can either specify a SAS format or specify DATA, which indicates that the format from axis variable is used.

TYPE= DISCRETE | LINEAR | LOG | TIME

specifies the type of axis. Specify one of the following values:

DISCRETE

specifies an axis with discrete values. If a character variable is assigned to an axis, then that the default type for that axis is discrete. In addition, all categorization plots use a discrete axis for the category variable.

LINEAR

specifies a linear scale for the axis. This is the default axis type for numeric variables, except when the data is discrete or when the numeric variable has a date or time format.

LOG

specifies a logarithmic scale for the axis. This axis type is never a default.

Interaction: A logarithmic scale cannot be used with linear regression plots (REG statement where DEGREE=1).

TIME

specifies a time scale for the axis. If the variable assigned to an axis has a time, date, or datetime format associated with it, then time is the default axis type.

VALUEATTRS= *style-element* <(options)> | (options)

specifies the appearance of the axis tick value labels. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the label characters. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphValueText style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

FAMILY= “*font-family*”

specifies the font family for the label characters. The SAS ODS styles use TrueType system fonts. For more information, see “TrueType Fonts Supplied by SAS” in Chapter 15 of *SAS Language Reference: Concepts*.

Default: The default font family is specified by the FontFamily attribute of the GraphValueText style element in the current style.

Restriction: You cannot specify a list of font family names.

SIZE= *n* <units>

specifies the font size of the label characters. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default font size is specified by the FontSize attribute of the GraphValueText style element in the current style.

STYLE= ITALIC | NORMAL

specifies whether the label characters are italic or normal.

Default: The default font style is specified by the FontStyle attribute of the GraphValueText style element in the current style.

WEIGHT= BOLD | NORMAL

specifies whether the label characters are bold or normal.

Default: The default font weight is specified by the FontWeight attribute of the GraphValueText style element in the current style.

VALUES= (*values-list*)

specifies the values for the ticks on the axis.

For values on a linear axis, the values list can be one of the following:

***value* <...*value-n*>**

creates ticks for specific values. For example, **VALUES= (0 50 100)** places tick marks at 0, 50, and 100.

value-1* TO *value-2* BY *increment-value

creates ticks for a range of values. The start of the value range is specified by *value-1* and the end of the range is specified by *value-2*. The *increment-value* specifies the interval between the ticks. For example, **VALUES= (0 to 100 by 50)** creates tick marks at 0, 50, and 100.

<value ... value-n> *value-1* TO *value-2* BY *increment-value* <value ... value-n>
creates ticks for specific values, and also creates ticks for a range of values. The start of the value range is specified by *value-1* and the end of the range is

specified by *value-2*. The *increment-value* specifies the interval between the ticks.

For example, **VALUES= (-5 10 to 50 by 20 75)** creates tick marks at -5, 10, 30, 50, and 75.

For values on a time axis, the values list can be one of the following:

value <...value-n>

creates ticks for specific values. For example, **VALUES= ("25MAY08"d "04JUL08"d "23AUG08"d)** places tick marks at 25MAY08, 04JUL08, and 23AUG08.

value-1 TO value-2 BY increment-value

creates ticks for a range of values. The start of the value range is specified by *value-1* and the end of the range is specified by *value-2*. The *increment-value* specifies the interval between the ticks. For example, **VALUES= ("01JAN08"d to "01MAY08"d by month)** creates tick marks at 01JAN08, 01FEB08, 01MAR08, 01APR08, and 01MAY08.

For a list of the interval values that you can specify, see the **INTERVAL=** option.

Restrictions:

This option has no effect on discrete and logarithmic axes.

If your **VALUES=** option creates more than 1000 values, then the option has no effect.

Interaction: If a custom format is applied to the value, the raw value is plotted first and then formatted.

VALUESHINT

specifies that the minimum and maximum axis values are determined independently of the values that you specify in the **VALUES=** option. The values from the **VALUES=** option are displayed only if they are located between the minimum and maximum values.

Interaction: This option has no effect unless you also specify the **VALUES=** option.

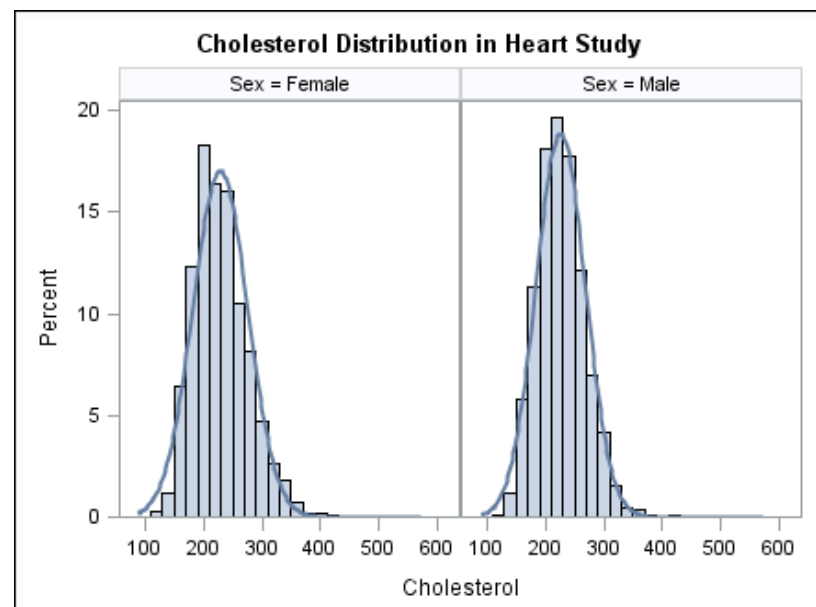
Examples: SGPanel Procedure

Example 1: Creating a Panel of Graph Cells with Histograms and Density Plots

Features: HISTOGRAM statement
DENSITY statement

Sample library member: SGPNHST

This example shows a panel of graph cells with histograms and density plots.

Output**Program**

```
proc sgpanel data=sashelp.heart noautolegend;
  title "Cholesterol Distribution in Heart Study";

  panelby sex;

  histogram cholesterol;
  density cholesterol;
run;

title;
```

Program Description**Create the panel and specify the title.**

```
proc sgpanel data=sashelp.heart noautolegend;
  title "Cholesterol Distribution in Heart Study";
```

Specify the classification variable for the panel.

```
  panelby sex;
```

Create the histogram and density plots.

```
  histogram cholesterol;
  density cholesterol;
run;
```

Cancel the title.

```
title;
```

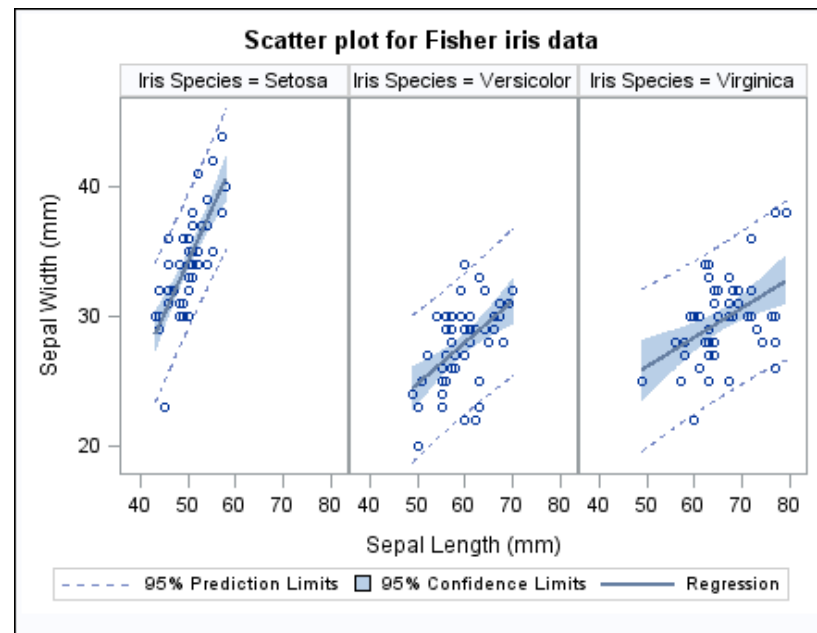
Example 2: Creating a Panel of Regression Curves

Features: REG statement

Sample library member: SGPNREG

This example shows a panel of regression curves. The COLUMNS= option in the PANELBY statement specifies that the panel has three columns of graph cells.

Output



Program

```
proc sgpanel data=sashelp.iris;
  title "Scatter plot for Fisher iris data";
  panelby species / columns=3;

  reg x=sepalwidth y=sepalwidth / cli clm;
run;

title;
```

Program Description

Create the panel and specify the title.

```
proc sgpanel data=sashelp.iris;
  title "Scatter plot for Fisher iris data";
```

Specify the classification variable for the panel. The COLUMNS= option specifies the number of columns in the panel.

```
panelby species / columns=3;
```

Create the regression curve. The CLI option creates individual predicted value confidence limits. The CLM option creates mean value confidence limits.

```
reg x=sepalwidth y=sepalwidth / cli clm;  
run;
```

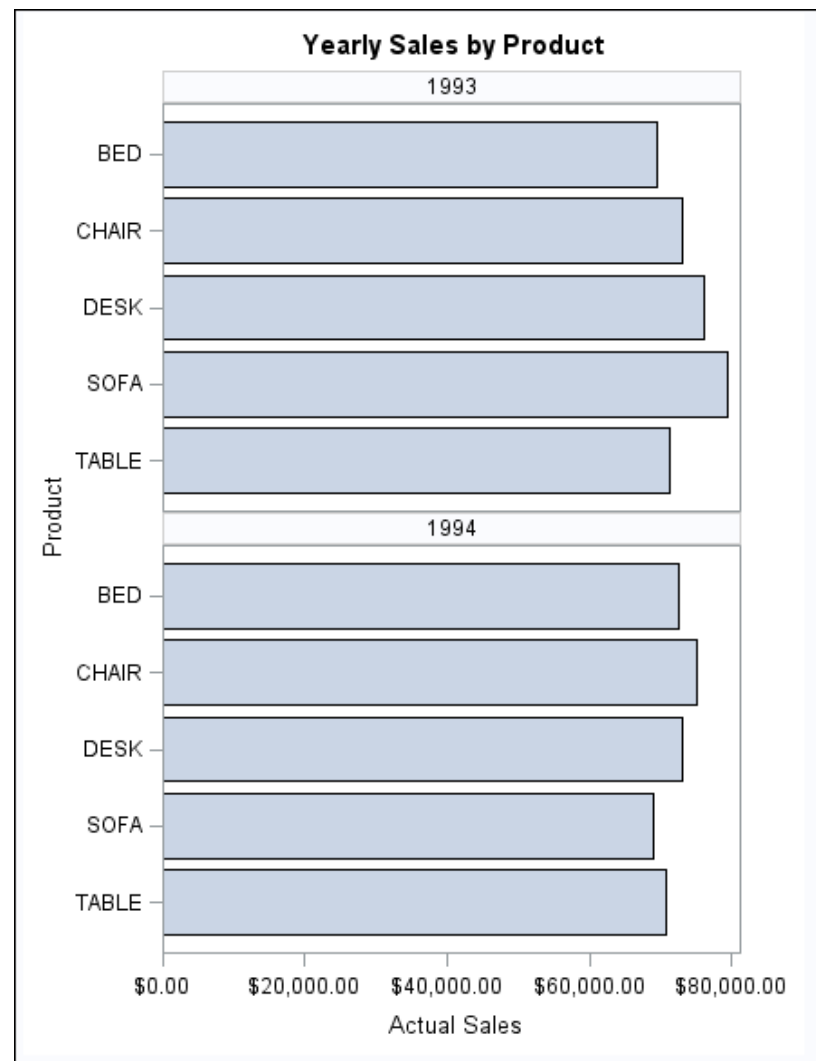
Cancel the title.

```
title;
```

Example 3: Creating a Panel of Bar Charts

Features:	HBAR statement
Sample library member:	SGPNBAR

This example shows a panel of bar charts. The COLUMNS= option in the PANELBY statement specifies that the panel contains a single column of cells.

Output**Program**

```
proc sgpanel data=sashelp.prdsale;
  title "Yearly Sales by Product";

  panelby year / novarname columns=1;

  hbar product / response=actual;
run;

title;
```

Program Description

Create the panel and set the title.

```
proc sgpanel data=sashelp.prdsale;
  title "Yearly Sales by Product";
```

Specify the classification variable for the panel. The NOVARNAM option specifies that the variable name is not shown in the heading for each cell. The COLUMNS= option specifies the number of columns in the panel.

```
panelby year / novarname columns=1;
```

Create the horizontal bar chart. The RESPONSE= option specifies the response variable for the chart.

```
hbar product / response=actual;  
run;
```

Cancel the title.

```
title;
```

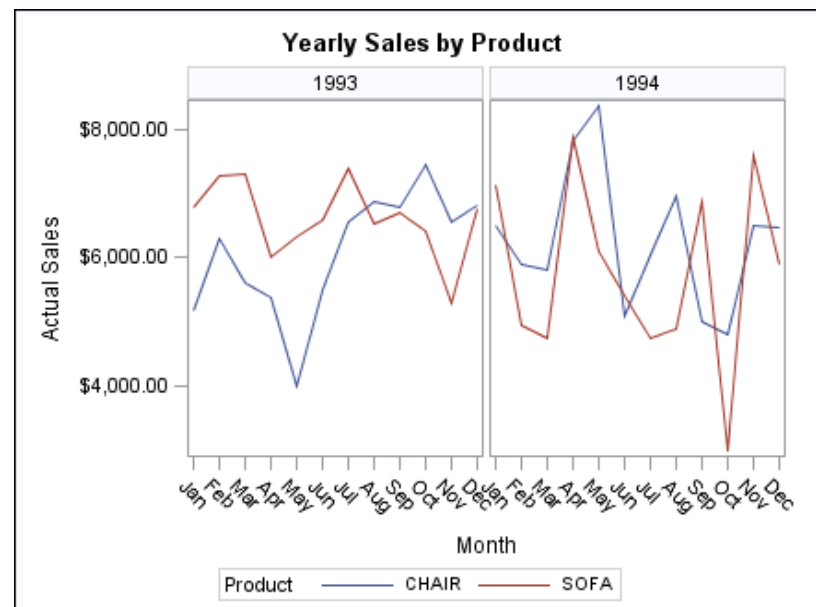
Example 4: Creating a Panel of Line Charts

Features: VLINE statement

Sample library member: SGPNNLIN

This example shows a panel of line plots with grouped data.

Output



Program

```
proc sgpanel data=sashelp.prdsale;  
  where product in ("CHAIR" "SOFA");  
  title "Yearly Sales by Product";  
  
  panelby year / spacing=5 novarname;
```



```

vline month / response=actual group=product;
run;

title;

```

Program Description

Create the panel and specify a title.

```

proc sgpanel data=sashelp.prdsale;
  where product in ("CHAIR" "SOFA");
  title "Yearly Sales by Product";

```

Specify the classification variable for the panel. The SPACING= option specifies the number of pixels between the panels in the plot. The NOVARNAME option specifies that the classification variable name is not shown in the headings for each cell.

```

panelby year / spacing=5 novarname;

```

Create the vertical line plot. The RESPONSE= option specifies the response variable. The GROUP= option specifies the group variable.

```

vline month / response=actual group=product;
run;

```

Cancel the title.

```

title;

```


Chapter 7

SGPLOT Procedure

Overview: SGPLOT Procedure	276
Concepts: SGPLOT Procedure	277
Plot Type Compatibility	277
Plot Axes	278
Legends in the SGPLOT Procedure	279
Automatic Differentiation of Visual Attributes	279
Syntax: SGPLOT Procedure	279
PROC SGPLOT Statement	281
BAND Statement	284
BUBBLE Statement	290
DENSITY Statement	295
DOT Statement	299
ELLIPSE Statement	306
HBAR Statement	309
HBARPARM Statement	317
HBOX Statement	324
HIGHLOW Statement	335
HISTOGRAM Statement	343
HLINE Statement	346
INSET Statement	356
KEYLEGEND Statement	360
LINEPARM Statement	366
LOESS Statement	371
NEEDLE Statement	377
PBSPLINE Statement	383
REFLINE Statement	390
REG Statement	392
SCATTER Statement	399
SERIES Statement	406
STEP Statement	413
VBAR Statement	420
VBARPARM Statement	429
VBOX Statement	436
VECTOR Statement	447
VLINE Statement	451
WATERFALL (Preproduction) Statement	462
XAXIS Statement	467
X2AXIS Statement	475
YAXIS Statement	483
Y2AXIS Statement	491

Examples: SGPLOT Procedure	498
Example 1: Grouping a Scatter Plot	498
Example 2: Clustering a Grouped Scatter Plot	499
Example 3: Plotting Three Series	500
Example 4: Adding Prediction and Confidence Bands to a Regression Plot	502
Example 5: Adding a Prediction Ellipse to a Scatter Plot	503
Example 6: Creating Lines and Bands from Pre-Computed Data	504
Example 7: Adding Statistical Limits to a Dot Plot	506
Example 8: Combining Histograms with Density Plots	507
Example 9: Creating a Horizontal Box Plot	508
Example 10: Creating a Bar-Line Chart	509
Example 11: Creating a High-Low Chart	510

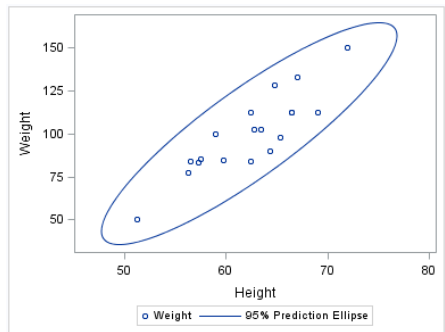
Overview: SGPLOT Procedure

The SGPLOT procedure creates one or more plots and overlays them on a single set of axes. You can use the SGPLOT procedure to create statistical graphics such as histograms and regression plots, in addition to simple graphics such as scatter plots and line plots. Statements and options enable you to control the appearance of your graph and add additional features such as legends and reference lines.

The SGPLOT procedure can create a wide variety of plot types, and can overlay plots together to produce many different types of graphs.

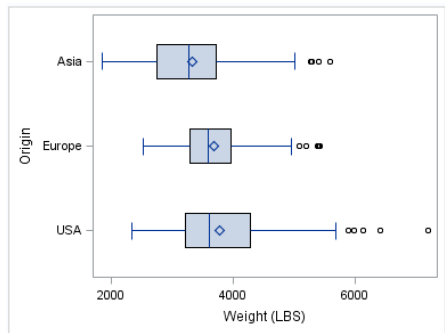
Here are some examples of graphs that the SGPLOT procedure can create.

Table 7.1 Examples of Graphs That Can Be Generated by the SGPLOT Procedure



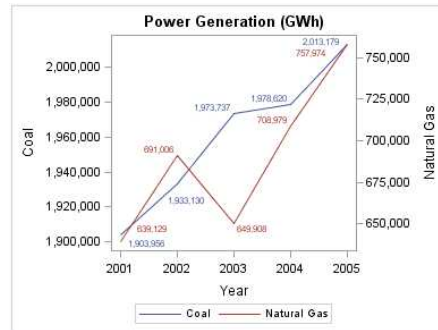
The following code creates an ellipse plot:

```
proc sgplot data=sashelp.class;
  scatter x=height y=weight;
  ellipse x=height y=weight;
run;
```



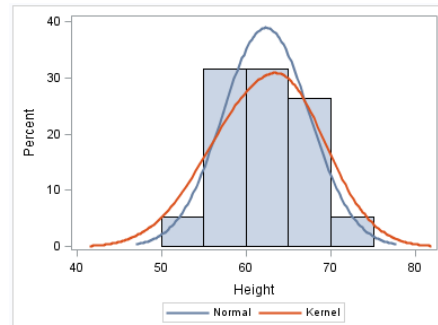
The following code creates a horizontal box plot:

```
proc sgplot data=sashelp.cars;
  hbox weight / category=origin;
run;
```



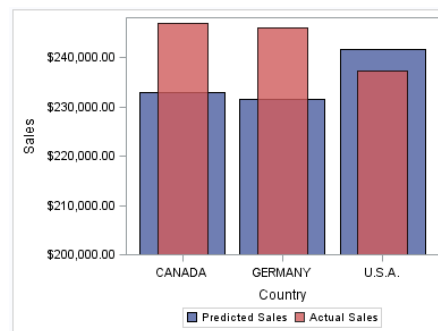
The following code creates a graph with two series plots:

```
title "Power Generation (GWh)";
proc sgplot data=sashelp.electric(where=
  (year >= 2001 and customer="Residential"));
  axis type=discrete;
  series x=year y=coal / datalabel;
  series x=year y=naturalgas /
    datalabel y2axis;
run;
title;
```



The following code creates a graph with a histogram, a normal density curve, and a kernel density curve:

```
proc sgplot data=sashelp.class;
  histogram height;
  density height;
  density height / type=kernel;
run;
```



The following code creates a graph with two bar charts:

```
proc sgplot data=sashelp.prdsale;
  yaxis label="Sales" min=200000;
  vbar country / response=predict;
  vbar country / response=actual
    barwidth=0.5
    transparency=0.2;
run;
```

Concepts: SGPLOT Procedure

Plot Type Compatibility

There are four basic types of plots that you can create with the SGPLOT procedure.

Basic plots

scatter, series, step, band, bubble, high-low, needle, and vector plots

Fit and confidence plots

loess, regression, and penalized B-spline curves, and ellipses

Distribution plots

box plots, histograms, and normal and kernel density estimates

Categorization plots

dot plots, bar charts, line plots, and waterfall plots (preproduction)

Not all of the plot types can be used together in the same PROC SGPLOT step. The following table shows which of the plot types can be used together:

Table 7.2 Plot Type Compatibility

	Basic	Fit and Confidence	Distribution	Categorization
Basic	x	x		
Fit and Confidence	x	x		
Distribution			x	
Categorization				x

Note: Box plots cannot be combined with any other plot types. Box plots can be overlaid with other box plots. However, overlaid box plots must have the same category variables.

If you submit a PROC SGPLOT step that combines two incompatible plot statements, then an error appears in the log.

The SGPLOT procedure draws the plots in your graph in the same order that you specify the plot statements. Because of this, it is important to consider the order of your plot statements so that your plots do not obscure one another. For example, if you specify a BAND statement after a SCATTER statement, then the band plot might obscure the markers in your scatter plot. You can also avoid obscuring your data by using the TRANSPARENCY= option to make your plots partially transparent.

Plot Axes

The SGPLOT procedure contains statements that enable you to change the type and appearance of your axes: XAXIS, X2AXIS, YAXIS, and Y2AXIS.

By default, the type of each axis is determined by the types of plots that use the axis and the data that is applied to the axis.

The SGPLOT procedure supports the following axis types:

Discrete

The axis contains independent data values rather than a range of numeric values. Each distinct value is represented by a tick mark. Discrete is the default axis type for character data.

Linear

The axis contains a linear range of numeric values. Linear is the default axis type for numeric data.

Logarithmic

The axis contains a logarithmic range of values. The logarithmic axis type is not used as a default.

Time

The axis contains a range of time values. Time is the default axis type for data that uses a SAS time, date, or datetime format.

Some types of plot do not support all of the axis types. For example, needle plots cannot use a discrete vertical axis. See the documentation for each plot statement to determine whether any axis type restrictions apply.

Legends in the SGPLOT Procedure

The SGPLOT procedure creates a legend automatically based on the plot statements and options that you specify. The automatic legend functionality determines which information is likely to be useful in the legend. You can override the automatic legend functionality by defining your own legend with the KEYLEGEND statement or by specifying the NOAUTOLEGEND option in the PROC SGPLOT statement.

You can create customized legends by using one or more KEYLEGEND statements. You can use the KEYLEGEND statement to control the contents, title, location, and border of the legend. See [“KEYLEGEND Statement” on page 360](#).

You can specify the labels that represent your plots in the legend by using the LEGENDLABEL= option in the corresponding plot statements.

Automatic Differentiation of Visual Attributes

Depending on the plots and options that you specify, the SGPLOT procedure can automatically assign different style attributes to the plots in your graph. For example, if you specify two series plots, then each series plot automatically uses a different line color by default. If different attributes are not assigned by default, then you can force the procedure to use different style attributes by using the CYCLEATTRS option in the PROC SGPLOT statement. For example, you can use the CYCLEATTRS option to assign different colors to a series plot and a scatter plot. You can also disable automatic attribute differentiation by using the NOCYCLEATTRS option in the PROC SGPLOT statement.

Syntax: SGPLOT Procedure

Requirement: At least one plot statement is required.

Global statements: BY, FORMAT, LABEL, ODS GRAPHICS, TITLE and FOOTNOTE, WHERE

```

PROC SGPLOT < option(s)>;
  BAND X= variable | Y= variable
  UPPER= numeric-value | numeric-variable LOWER= numeric-value | numeric-variable
  </option(s)>;
  BUBBLE X= variable Y= variable SIZE= numeric-variable </ option(s)>;
  DENSITY response-variable </option(s)>;
  DOT category-variable </option(s)>;
  ELLIPSE X= numeric-variable Y= numeric-variable </option(s)>;
  HBAR category-variable </option(s)>;
  HBARPARM CATEGORY= category-variable RESPONSE= numeric-variable </ option(s)>;
  HBOX analysis-variable </option(s)>;
  HIGHLOW X= variable | Y= variable
  HIGH= numeric-variable LOW= numeric-variable
  </option(s)>;
  HISTOGRAM response-variable </option(s)>;
  HLINE category-variable </option(s)>;
  INSET "text-string-1" <... "text-string-n"> | (label-list) </option(s)>;
  KEYLEGEND <"name-1" ... "name-n"> </option(s)>;
  LINEPARM X= numeric-value |numeric-variable
  Y= numeric-value |numeric-variable
  SLOPE= numeric-value |numeric-variable
  </option(s)>;
  LOESS X= numeric-variable Y= numeric-variable </option(s)>;
  NEEDLE X= variable Y= numeric-variable </option(s)>;
  PBSPLINE X= numeric-variable Y= numeric-variable </option(s)>;
  REFLINE value(s) </option(s)>;
  REG X= numeric-variable Y= numeric-variable </option(s)>;
  SCATTER X= variable Y= variable </option(s)>;
  SERIES X= variable Y= variable </option(s)>;
  STEP X= variable Y= variable </option(s)>;
  VBAR category-variable </option(s)>;
  VBARPARM CATEGORY= category-variable RESPONSE= numeric-variable </ option(s)>;
  VBOX analysis-variable </option(s)>;
  VECTOR X= numeric-variable Y= numeric-variable </option(s)>;
  VLINE category-variable </option(s)>;
  WATERFALL CATEGORY=variable RESPONSE= numeric-variable </ option(s)>;
  XAXIS <option(s)>;
  X2AXIS <option(s)>;
  YAXIS <option(s)>;
  Y2AXIS <option(s)>;

```


PROC SGPLOT Statement

Identifies the data set that contains the plot variables. The statement also gives you the option to specify a description, write template code to a file, control the uniformity of axes, and control automatic legends and automatic attributes.

Requirement: An input data set is required.

Syntax

```
PROC SGPLOT <DATA= input-data-set>
  <CYCLEATTRS | NOCYCLEATTRS>
  <DATTRMAP= "attribute-map-data-set" >
  <DESCRIPTION= "text-string" >
  <NOAUTOLEGEND>
  <PAD= dimension<units> | (pad-options)>
  <SGANNO= "annotation-data-set" >
  <TMPLOUT= "filename">
  <UNIFORM= GROUP | SCALE | ALL | XSCALE | YSCALE | XSCALEGROUP | YSCALEGROUP>
```

Summary of Optional Arguments

CYCLEATTRS | NOCYCLEATTRS

specifies whether plots are drawn with unique attributes in the graph.

DATA=*input-data-set*

specifies the SAS data set that contains the variables to process.

DATTRMAP= *attribute-map-data-set*

specifies the SG attribute map data set that you want to use with the SGSCATTER procedure.

DESCRIPTION= "*text-string*"

specifies a description for the output image.

NOAUTOLEGEND

disables automatic legends from being generated.

PAD= *dimension*<units> | (*pad-options*)

specifies the amount of extra space that is reserved inside the border of an annotated graph.

SGANNO= *annotation-data-set*

specifies the SG annotation data set that you want to use.

TMPLOUT= "*filename*"

writes the Graph Template Language code for your graph to a file.

UNIFORM= GROUP | SCALE | ALL | XSCALE | YSCALE | XSCALEGROUP | YSCALEGROUP

specifies how to control axis scaling and marker attributes when you use a BY statement.

Optional Arguments

CYCLEATTRS | NOCYCLEATTRS

specifies whether plots are drawn with unique attributes in the graph. By default, the SGPLOT procedure automatically assigns unique attributes in many situations, depending on the types of plots that you specify. If the plots do not have unique attributes by default, then the CYCLEATTRS option assigns unique attributes to each plot in the graph. The NOCYCLEATTRS option prevents the procedure from assigning unique attributes.

For example, if you specify the CYCLEATTRS option and you create a graph with a SERIES statement and a SCATTER statement, then the two plots will have different colors.

If you specify the NOCYCLEATTRS option, then plots have the same attributes unless you specify appearance options such as the LINEATTRS= option.

DATA=*input-data-set*

specifies the SAS data set that contains the variables to process. By default, the procedure uses the most recently created SAS data set.

DATTRMAP= *attribute-map-data-set*

specifies the SG attribute map data set that you want to use with the SGSCATTER procedure. You specify this option only if you are using an attribute map to control visual attributes of the graph. For more information, see [“Using SG Attribute Maps to Control Visual Attributes”](#) on page 605.

DESCRIPTION= “*text-string*”

specifies a description for the output image. The description identifies the image in the following locations:

- the Results window
- the alternate text for the image in HTML output
- the table of contents that is created by the CONTENTS option in an ODS statement

The default description is “The SGPLOT Procedure”.

Alias: DES

Notes:

The name of the output image is specified by the IMAGENAME= option in the ODS GRAPHICS statement.

You can disable the alternate text in HTML output by specifying an empty string. That is, **DESCRIPTION= " "**.

NOAUTOLEGEND

disables automatic legends from being generated. By default, legends are created automatically for some plots, depending on their content. This option has no effect if you specify a KEYLEGEND statement.

PAD= *dimension<units>* | (*pad-options*)

specifies the amount of extra space that is reserved inside the border of an annotated graph.

You specify this option only if you are using the SG annotation feature to annotate your graph. For more information, see [Chapter 10, “Annotating ODS Graphics,”](#) on page 557.

This option creates margins around the graph for company logos, annotated notes, and so on. You can also specify the unit of measurement. The default unit is pixels.

For a list of measurement units that are supported, see [“Units of Measurement” on page 659](#).

Use pad options to create non-uniform padding. Edges that are not assigned padding are padded with the default amount.

pad-options can be one or more of the following:

LEFT= *dimension*<*units*>

specifies the amount of extra space to add to the left edge.

RIGHT= *dimension*<*units*>

specifies the amount of extra space to add to the right edge.

TOP= *dimension*<*units*>

specifies the amount of extra space to add to the top edge.

BOTTOM= *dimension*<*units*>

specifies the amount of extra space to add to the bottom edge.

SGANNO= *annotation-data-set*

specifies the SG annotation data set that you want to use. You specify this option only if you are using the SG annotation feature to annotate your graph. For more information, see [Chapter 10, “Annotating ODS Graphics,” on page 557](#).

TMPLOUT= “*filename*”

writes the Graph Template Language code for your graph to a file. Writing your template code to a file can be useful for building larger Graph Template Language definitions.

Restriction: Graphs that contain summarized data (for example, bar, line, and dot) do not generate a useable template. For summarized data, the procedure sets internal values for one or more variables, and these internal values do not exist in the data set that is used with the procedure.

UNIFORM= GROUP | SCALE | ALL | XSCALE | YSCALE | XSCALEGROUP | YSCALEGROUP

specifies how to control axis scaling and marker attributes when you use a BY statement.

Specify one of the following values:

ALL

specifies that both the legend group values and the axis scaling are shared between all of the levels of the BY variable or variables.

GROUP

specifies that the legend group values are shared between all of the levels of the BY variable or variables. This option is applicable only when the GROUP= option is specified for the plot.

SCALE

specifies that the axis scaling is shared between all of the levels of the BY variable or variables.

XSCALE

specifies that the X axis scaling is shared between all of the levels of the BY variable or variables. The Y axes are unaffected.

YSCALE

specifies that the Y axis scaling is shared between all of the levels of the BY variable or variables. The X axes are unaffected.

XSCALEGROUP

specifies that both the legend group values and the X axis scaling are shared between all of the levels of the BY variable or variables. The Y axes are unaffected. This option is applicable only when the GROUP= option is specified for the plot.

YSCALEGROUP

specifies that both the legend group values and the Y axis scaling are shared between all of the levels of the BY variable or variables. The X axes are unaffected. This option is applicable only when the GROUP= option is specified for the plot.

By default, each level of the BY variable or variables can have different legend group values and different axis scaling.

Interaction: This option has no effect if you do not use a BY statement.

BAND Statement

Creates a band that highlights part of the plot.

Restriction: The axis that the UPPER and LOWER values are placed on cannot be a discrete axis. For example, if you specify a variable for Y, the plot cannot use a discrete horizontal axis.

Note: The input data should be sorted by the X or Y variable. If the data is not sorted, the graph might produce unpredictable results.

Examples: [“About Band Plots” on page 20](#)
[“Example 6: Creating Lines and Bands from Pre-Computed Data” on page 504](#)

Syntax

BAND X= *variable* | Y= *variable*

UPPER= *numeric-value* | *numeric-variable* LOWER= *numeric-value* | *numeric-variable*
 </option(s)>;

Summary of Optional Arguments

Band options

DISCRETEOFFSET= *numeric-value*

specifies an amount to offset all bands from discrete X or Y tick values.

FILL | **NOFILL**

specifies whether the area fill is visible.

FILLATTRS= *style-element* | (COLOR= *color*)

specifies the appearance of the area fill for the band.

LINEATTRS= *style-element* <(options)> | (options)

specifies the appearance of the outlines for the band.

MODELNAME= “*plot-name*”

specifies a plot that is used as a model for the interpolation for the band.

NOEXTEND

specifies that the band does not extend beyond the first and last data points in the plot.

OUTLINE | NOOUTLINE

specifies whether the outlines of the band are visible.

TYPE= SERIES | STEP

specifies how the data points for the lower and upper band boundaries are connected.

Group options**GROUP= *variable***

specifies a variable that is used to group the data.

NOMISSINGGROUP

specifies that missing values of the group variable are not included in the plot.

Label options**CURVELABELATTRS= *style-element* <(*options*)> | (*options*)**

specifies the appearance of the labels in the plot when you specify a curve label.

CURVELABELLOC= OUTSIDE | INSIDE

specifies whether the curve label is placed inside the plot axes (INSIDE) or outside of the plot axes (OUTSIDE).

CURVELABELLOWER = “*text-string*”

adds a label to the lower edge of the band.

CURVELABELPOS= AUTO | END | MAX | MIN | START

specifies the location of the curve label.

CURVELABELUPPER = “*text-string*”

adds a label to the upper edge of the band.

Plot options**ATTRID= *variable***

specifies the value of the ID variable in an attribute map data set.

LEGENDLABEL= “*text-string*”

specifies a label that identifies the elements from the band plot in the legend.

NAME= “*text-string*”

specifies a name for the plot.

TRANSPARENCY= *value*

specifies the degree of transparency for the plot.

X2AXIS

assigns the variable that is assigned to the primary (bottom) horizontal axis to the secondary (top) horizontal axis.

Y2AXIS

assigns the variable that is assigned to the primary (left) vertical axis to the secondary (right) vertical axis.

Required Arguments**X= *variable* | Y= *variable***

specifies a variable that is used to plot the band along the x or y axis.

LOWER= *numeric-value* | *numeric-variable*

specifies the lower value for the band. You can specify either a constant numeric value or a numeric variable.

UPPER= *numeric-value* | *numeric-variable*

specifies the upper value for the band. You can specify either a constant numeric value or a numeric variable.

Optional Arguments

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph. For more information, see [“Using SG Attribute Maps to Control Visual Attributes” on page 605](#).

CURVELABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you specify a curve label. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

Here is an example that specifies a style element:

CURVELABELATTRS=GraphTitleText

options can be one or more of the following:

COLOR= *color*

specifies the color of the line label characters. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphValueText style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

FAMILY= “*font-family*”

specifies the font family for the label characters. The SAS ODS styles use TrueType system fonts. For more information, see “TrueType Fonts Supplied by SAS” in Chapter 15 of *SAS Language Reference: Concepts*.

Default: The default font family is specified by the FontFamily attribute of the GraphValueText style element in the current style.

Restriction: You cannot specify a list of font family names.

SIZE= *n* <*units*>

specifies the font size of the label characters. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see [“Units of Measurement” on page 659](#).

Default: The default font size is specified by the FontSize attribute of the GraphValueText style element in the current style.

STYLE= ITALIC | NORMAL

specifies whether the label characters are italic or normal.

Default: The default font style is specified by the FontStyle attribute of the GraphValueText style element in the current style.

WEIGHT= BOLD | NORMAL

specifies whether the label characters are bold or normal.

Default: The default font weight is specified by the FontWeight attribute of the GraphValueText style element in the current style.

Here is an example expression:

CURVELABELATTRS=(Color=Green Family=Arial Size=8 Style=Italic Weight=Bold)

Default: The default style element is GraphValueText.

Interaction: This option has no effect unless CURVELABELLOWER or CURVELABELUPPER is also specified.

CURVELABELLOC= OUTSIDE | INSIDE

specifies whether the curve label is placed inside the plot axes (INSIDE) or outside of the plot axes (OUTSIDE).

Default: INSIDE

CURVELABELLOWER = “text-string”

adds a label to the lower edge of the band. Specify the label text.

CURVELABELPOS= AUTO | END | MAX | MIN | START

specifies the location of the curve label. Specify one of the following values:

AUTO

places the curve label outside the plot area near the end of the curve along unused axes whenever possible (typically Y2 or X2).

Interaction: This value takes effect only when CURVELABELLOC=OUTSIDE.

END

places the curve label at the last point on the curve.

MAX

places the label at the part of the curve closest to the maximum X axis value.

MIN

places the label at the part of the curve closest to the minimum X axis value.

START

places the curve label at the first point on the curve.

Default: END

Interactions:

This option has no effect unless the CURVELABEL option is also specified.

The START and END suboptions take effect only when CURVELABELLOC=INSIDE.

CURVELABELUPPER = “text-string”

adds a label to the upper edge of the band. Specify the label text.

DISCRETEOFFSET= numeric-value

specifies an amount to offset all bands from discrete X or Y tick values. Specify a value from -0.5 (left offset) to +0.5 (right offset).

Default: 0.0 (no offset)

Requirement: This option is applicable only when the X or Y axis is discrete.

FILL | NOFILL

specifies whether the area fill is visible. The FILL option shows the area fill. The NOFILL option hides the area fill.

Default: The default status of the area fill is specified by the DisplayOpts attribute of the GraphBand style element in the current style.

Interactions:

Specifying FILL also hides any visible outlines.

If NOFILL and NOOUTLINE are both specified, then both options are ignored.

FILLATTRS= *style-element* | (COLOR= *color*)

specifies the appearance of the area fill for the band. You can specify the color of the fill by using a style element or by using the COLOR= suboption. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphDataConfidence style element in the current style. For grouped data, the default color is specified by the Color attribute of the GraphData1... GraphData*n* style elements in the current style.

Interaction: This option has no effect if you specify the NOFILL option.

GROUP= *variable*

specifies a variable that is used to group the data. A separate band is created for each unique value of the grouping variable. The plot elements for each group value are automatically distinguished by different visual attributes.

LEGENDLABEL= “*text-string*”

specifies a label that identifies the elements from the band plot in the legend. By default, the label “band” is used for ungrouped data, and the group values are used for grouped data.

Interaction: The LEGENDLABEL= option has no effect if you also specify the GROUP= option in the same plot statement.

LINEATTRS= *style-element* <(options)> | (options)

specifies the appearance of the outlines for the band. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphConfidence style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see “[Line Patterns](#)” on page 663.

Default: For ungrouped data, the default line pattern is specified by the LineStyle attribute of the GraphConfidence style element in the current style. For grouped data, the default line pattern is specified by the LineStyle attribute of the GraphData1 ... GraphData*n* style elements in the current style.

THICKNESS= *n* <units>

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see “[Units of Measurement](#)” on page 659.

Default: For ungrouped data, the default line thickness is specified by the LineThickness attribute of the GraphDataDefault style element in the current style. For grouped data, the default line thickness is specified by the LineThickness attribute of the GraphData1 ... GraphData*n* style elements in the current style.

Interaction: This option has no effect unless you also specify the OUTLINES option.

MODELNAME= “*plot-name*”

specifies a plot that is used as a model for the interpolation for the band. If you do not specify the MODELNAME= option, then the band is interpolated in the same way as a series plot.

NAME= “*text-string*”

specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note: The name is case-sensitive. When you refer to the specified name in other statements, you must use the same exact capitalization.

NOEXTEND

specifies that the band does not extend beyond the first and last data points in the plot. By default, the band extends to the edges of the plot area.

Interaction: This option has no effect if you do not specify numeric values for the UPPER= and LOWER= options.

NOMISSINGGROUP

specifies that missing values of the group variable are not included in the plot.

OUTLINE | NOOUTLINE

specifies whether the outlines of the band are visible. The OUTLINE option shows the outlines. The NOOUTLINE option hides the outlines.

Default: The default status of the band outlines is specified by the DisplayOpts attribute of the GraphBand.

Interactions:

Specifying OUTLINE also hides the fill color.

If NOOUTLINE and NOFILL are both specified, then both options are ignored.

TRANSPARENCY= *value*

specifies the degree of transparency for the plot. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default: 0.0

TYPE= SERIES | STEP

specifies how the data points for the lower and upper band boundaries are connected. You can specify one of the following:

SERIES

the data points are connected directly using line segments, as in a series plot.

STEP

the data points are connected using a step function, as in a step plot.

Default: SERIES

X2AXIS

assigns the variable that is assigned to the primary (bottom) horizontal axis to the secondary (top) horizontal axis.

Y2AXIS

assigns the variable that is assigned to the primary (left) vertical axis to the secondary (right) vertical axis.

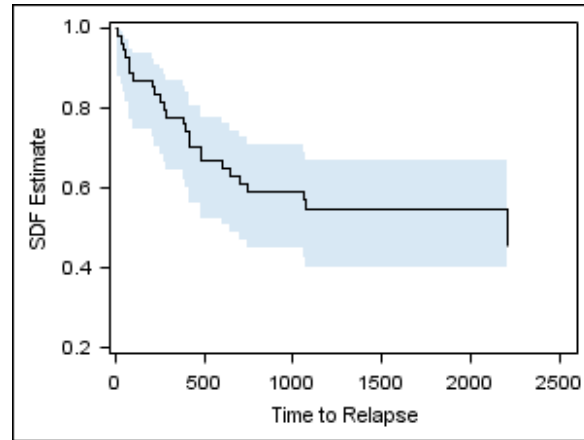
Details

The MODELNAME= option fits a band to another plot. This is particularly useful for plots that use a special interpolation such as step plots.

The following code fragment fits a band to a step plot:

```
band x=t upper=ucl lower=lcl / modelname="myname" transparency=.5;
step x=t y=survival / name="myname";
```

Figure 7.1 Fitted Band Plot Example



BUBBLE Statement

Creates a bubble plot in which two variables determine the location of the bubble centers and a third variable controls the size of the bubble.

Example: [“About Bubble Plots” on page 21](#)

Syntax

BUBBLE X= *variable* Y= *variable* SIZE= *numeric-variable* </ option(s)>;

Summary of Optional Arguments

Bubble options

BRADIUSMAX= *numeric-value*

specifies the size of the radius of the largest bubble.

BRADIUSMIN= *numeric-value*

specifies the size of the radius of the smallest bubble.

FILL | **NOFILL**

specifies whether the bubbles are filled.

FILLATTRS= *style-element* | (**COLOR**= *color*)

specifies the appearance of the fill for the bubbles.

LINEATTRS= *style-element* <(options)> | (options)

specifies the appearance of the outlines for the bubbles.

OUTLINE | **NOOUTLINE**

specifies whether the outlines of the bubbles are visible.

Group options

GROUP= *variable*

specifies a variable that is used to group the data.

NOMISSINGGROUP

specifies that missing values of the group variable are not included in the plot.

Label options

DATALABEL <= *variable*>

displays a label for each data point.

DATALABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the DATALABEL= option.

Plot options

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set.

LEGENDLABEL= "*text-string*"

specifies the label that identifies the bubble plot in the legend.

NAME= "*text-string*"

specifies a name for the plot.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the fill and outline of the bubbles.

URL= *character-variable*

specifies a character variable that contains URLs for Web pages to be displayed when parts of the plot are selected within an HTML page.

X2AXIS

assigns the variables that are assigned to the primary (bottom) horizontal axis to the secondary (top) horizontal axis.

Y2AXIS

assigns the variables that are assigned to the primary (left) vertical axis to the secondary (right) vertical axis.

Required Arguments

X= *variable*

specifies the variable for the X axis.

Y= *variable*

specifies the variable for the Y axis.

SIZE= *numeric-variable*

specifies the variable that controls the size of the bubbles. The minimum and maximum values automatically provide the range that is used to determine bubble sizes. You can control this range manually by using the BRADIUSMAX and BRADIUSMIN options.

Optional Arguments

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph. For more information, see [“Using SG Attribute Maps to Control Visual Attributes”](#) on page 605.

BRADIUSMAX= *numeric-value*

specifies the size of the radius of the largest bubble. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see [“Units of Measurement” on page 659](#).

Restriction: The BRADIUSMAX= value must be greater than the BRADIUSMIN= value if one is specified. If BRADIUSMAX is not greater, an error is generated and a message is written to the SAS log.

Note: If you specify the maximum size as a percentage, this is interpreted as a percentage of the graph's height.

BRADIUSMIN= *numeric-value*

specifies the size of the radius of the smallest bubble. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see [“Units of Measurement” on page 659](#).

Restriction: The BRADIUSMIN= value must be smaller than the BRADIUSMAX= value if one is specified. If BRADIUSMIN is not smaller, an error is generated and a message is written to the SAS log.

Note: If you specify the minimum size as a percentage, this is interpreted as a percentage of the graph's height.

DATALABEL <= *variable*>

displays a label for each data point. If you specify a variable, the values of that variable are used for the data labels. If you do not specify a variable, then the values of the Y variable are used for the data labels.

DATALABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the DATALABEL= option. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

Here is an example that specifies a style element:

```
DATALABELATTRS=GraphLabelText
```

options can be one or more of the following:

COLOR= *color*

specifies the color of the label characters. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see [“Color-Naming Schemes” in Chapter 14 of SAS/GRAPH: Reference](#).

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphDataText style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

FAMILY= *“font-family”*

specifies the font family for the label characters. The SAS ODS styles use TrueType system fonts. For more information, see [“TrueType Fonts Supplied by SAS” in Chapter 15 of SAS Language Reference: Concepts](#).

Default: The default font family is specified by the FontFamily attribute of the GraphDataText style element in the current style.

SIZE= *n* <units>

specifies the font size of the label characters. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see [“Units of Measurement” on page 659](#).

Default: The default font size is specified by the FontSize attribute of the GraphDataText style element in the current style.

STYLE= ITALIC | NORMAL

specifies whether the label characters are italic (ITALIC) or normal (NORMAL).

Default: The default font style is specified by the FontStyle attribute of the GraphDataText style element in the current style.

WEIGHT= BOLD | NORMAL

specifies whether the label characters are bold (BOLD) or normal (NORMAL).

Default: The default font weight is specified by the FontWeight attribute of the GraphDataText style element in the current style.

Here is an example expression:

```
DATALABELATTRS=(Color=Green Family=Arial Size=8 Style=Italic Weight=Bold)
```

Default: The default style element is GraphDataText.

Interaction: This option has no effect unless the DATALABEL option is also specified.

FILL | NOFILL

specifies whether the bubbles are filled. The FILL option shows the fill color. The NOFILL option hides the fill color.

Default: FILL

Interactions:

Specifying FILL also hides the outlines.

If NOFILL and NOOUTLINE are both specified, then both options are ignored.

FILLATTRS= *style-element* | (COLOR= *color*)

specifies the appearance of the fill for the bubbles. You can specify the color of the fill by using a style element or by using the COLOR= suboption. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphDataDefault style element in the current style. For grouped data, the default color is specified by the Color attribute of the GraphData1... GraphData*n* style elements in the current style.

Interaction: This option has no effect if you specify the NOFILL option.

GROUP= *variable*

specifies a variable that is used to group the data. The bubbles for each group value are automatically distinguished by different colors.

When this option is used, the bubble colors are taken from the GraphData1... GraphData*n* style elements. If the bubbles are filled, then the COLOR attribute is used for bubble fill and GRAPHOUTLINES is used for the bubble outline. If the bubbles are not filled, then the CONTRASTCOLOR and PATTERN are used for the bubble outlines.

When this option is used and the value is a variable associated with an ATTRID= option, the attribute mapping defined by the associated attribute map is used.

LEGENDLABEL= “*text-string*”

specifies the label that identifies the bubble plot in the legend. By default, the label for the Y variable is used for ungrouped data, and the group values are used for grouped data.

Interaction: The LEGENDLABEL= option has no effect if you also specify the GROUP= option in the same plot statement.

LINEATTRS= *style-element* <(options)> | (options)

specifies the appearance of the outlines for the bubbles. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

Here is an example that specifies a style element: **LINEATTRS=GraphData3**

For more information about style elements, see [“Style Elements for Use with ODS Graphics” on page 640](#).

options can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see [“Color-Naming Schemes” in Chapter 14 of SAS/GRAPH: Reference](#).

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphDataDefault style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see [“Line Patterns” on page 663](#).

Default: For ungrouped data, the default line pattern is specified by the LineStyle attribute of the GraphDataDefault style element in the current style. For grouped data, the default line pattern is specified by the LineStyle attribute of the GraphData1 ... GraphData*n* style elements in the current style.

THICKNESS= *n* <units>

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see [“Units of Measurement” on page 659](#).

Default: For ungrouped data, the default line thickness is specified by the LineThickness attribute of the GraphDataDefault style element in the current style. For grouped data, the default line thickness is specified by the LineThickness attribute of the GraphData1 ... GraphData*n* style elements in the current style.

Here is an example expression:

```
LINEATTRS=(Color= "light green" Pattern= MediumDash Thickness= 4)
```

Default: For ungrouped data, line attributes are derived from the GraphDataDefault style element. For grouped data, the ContrastColor attribute of the GraphData1... GraphData*n* style elements are used.

Interaction: This option has no effect if you also specify the NOOUTLINE option.

NAME= “*text-string*”

specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note: The name is case-sensitive. When you refer to the specified name in other statements, you must use the same exact capitalization.

NOMISSINGGROUP

specifies that missing values of the group variable are not included in the plot.

Interaction: This option has no effect unless GROUP= is also specified.

OUTLINE | NOOUTLINE

specifies whether the outlines of the bubbles are visible. The OUTLINE option shows the outlines. The NOOUTLINE option hides the outlines.

Default: OUTLINE

Interactions:

Specifying OUTLINE also hides the fill color.

If NOOUTLINE and NOFILL are both specified, then both options are ignored.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the fill and outline of the bubbles. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default: 0.0

URL= *character-variable*

specifies a character variable that contains URLs for Web pages to be displayed when parts of the plot are selected within an HTML page.

Default: By default, no HTML links are created.

Interactions:

This option affects graphics output that is created through the ODS HTML destination only. For more information about ODS destinations, see [“Specifying a Destination” on page 649](#).

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For more information, see [“Using the ODS GRAPHICS Statement” on page 651](#).

X2AXIS

assigns the variables that are assigned to the primary (bottom) horizontal axis to the secondary (top) horizontal axis.

Y2AXIS

assigns the variables that are assigned to the primary (left) vertical axis to the secondary (right) vertical axis.

DENSITY Statement

Creates a density curve that shows the distribution of values for a numeric variable.

Restriction: The DENSITY statement cannot be used with discrete axes.

Interaction: The DENSITY statement can be combined only with DENSITY and HISTOGRAM statements in the SGPLOT procedure.

Examples: [“About Density Plots” on page 39](#)
[“Example 8: Combining Histograms with Density Plots” on page 507](#)

Syntax

DENSITY *response-variable* </ *option(s)*>;

Summary of Optional Arguments

Density options

LINEATTRS= *style-element* <(options)> | (options)
 specifies the appearance of the density line.

SCALE= *scaling-type*

specifies the scaling that is used for the response axis.

TYPE = NORMAL < (*normal-opts*)> | KERNEL < (*kernel-opts*)>

specifies the type of distribution curve that is used for the density plot.

Plot options

FREQ= *numeric-variable*

specifies how many times observations are repeated for computational purposes.

LEGENDLABEL= "*text-string*"

specifies a label that identifies the density plot in the legend.

NAME= "*text-string*"

specifies a name for the plot.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the density curve.

X2AXIS

assigns the X variable to the secondary (top) horizontal axis.

Y2AXIS

assigns the calculated result to the secondary (right) vertical axis.

Required Argument

response-variable

specifies the variable for the x axis. The variable must be numeric.

Optional Arguments

FREQ= *numeric-variable*

specifies how many times observations are repeated for computational purposes. Each observation is repeated *n* times for computational purposes, where *n* is the value of the numeric variable. If *n* is not an integer, then it is truncated to an integer. If *n* is less than 1 or missing, then it is excluded from the analysis.

LEGENDLABEL= "*text-string*"

specifies a label that identifies the density plot in the legend. By default, the label identifies the type of density curve. If you specify TYPE=NORMAL, then the default label is "Normal." If you specify TYPE=KERNEL, then the default label is "Kernel."

Note: User-specified parameters from the TYPE= option are included in the label by default.

LINEATTRS= *style-element* <(*options*)> | (*options*)

specifies the appearance of the density line. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see "Color-Naming Schemes" in Chapter 14 of *SAS/GRAPH: Reference*.

Default: The default color is specified by the ContrastColor attribute of the GraphFit style element in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see [“Line Patterns” on page 663](#).

Default: The default line pattern is specified by the LineStyle attribute of the GraphFit style element in the current style.

THICKNESS= *n* *<units>*

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see [“Units of Measurement” on page 659](#).

Default: The default line thickness is specified by the LineThickness attribute of the GraphFit style element in the current style.

NAME= *“text-string”*

specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note: The name is case-sensitive. When you refer to the specified name in other statements, you must use the same exact capitalization.

SCALE= *scaling-type*

specifies the scaling that is used for the response axis. Specify one of the following values:

COUNT

the axis displays the frequency count.

DENSITY

the axis displays the density estimate values.

PERCENT

the axis displays values as a percentage of the total.

PROPORTION

the axis displays values in proportion to the total.

Default: DENSITY

Note: The PROPORTION scale can be used only when you combine a density plot and a histogram together.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the density curve. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default: 0.0

TYPE = NORMAL *<(normal-opts)>* | KERNEL *<(kernel-opts)>*

specifies the type of distribution curve that is used for the density plot. Specify one of the following keywords:

NORMAL *<(normal-opts)>*

specifies a normal density estimate, with a mean and a standard deviation.

normal-opts can be one or more of the following values:

MU= *numeric-value*

specifies the mean value that is used in the density function equation. By default, the mean value is calculated from the data.

SIGMA= *numeric-value*

specifies the standard deviation value that is used in the density function equation. The value that you specify for the SIGMA= suboption must be a positive number. By default, the standard deviation value is calculated from the data.

KERNEL <(kernel-opts)>

specifies a nonparametric kernel density estimate.

kernel-opts can be:

C= *numeric-value*

specifies the standardized bandwidth for a number that is greater than 0 and less than or equal to 100.

The value that you specify for the C= suboption affects the value of λ as shown in the following equation:

$$\lambda = cQn^{-\frac{1}{5}}$$

In this equation c is the standardized bandwidth, Q is the interquartile range, and n is the sample size.

WEIGHT= NORMAL | QUADRATIC | TRIANGULAR

specifies the weight function. You can specify either normal, quadratic, or triangular weight function.

Default: NORMAL

Default: NORMAL

X2AXIS

assigns the X variable to the secondary (top) horizontal axis.

Y2AXIS

assigns the calculated result to the secondary (right) vertical axis.

Details

Normal Density Function

When the type of the density curve is NORMAL, the fitted density function equation is as follows.

$$p(x) = \frac{100h}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right) \text{ for } -\infty < x < \infty$$

In the equation, μ is the mean, and σ is the standard deviation. You can specify μ by using the MU= suboption and σ by using the SIGMA= suboption.

Kernel Density Function

When the TYPE of the density curve is KERNEL, 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, n is the sample size, and x_i is the i th observation. You can use the C= suboption to specify the bandwidth and the WEIGHT= suboption to specify the weight function $K_0(\cdot)$.

Kernel Density Weight Functions

The formulas for the weight functions are as follows.

NORMAL

$$K_0(t) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}t^2\right) \text{ for } -\infty < t < \infty$$

QUADRATIC

$$K_0(t) = \frac{3}{4}(1 - t^2) \text{ for } |t| \leq 1$$

TRIANGULAR

$$K_0(t) = 1 - |t| \text{ for } -|t| \leq 1$$

DOT Statement

Creates a dot plot that summarizes the values of a category variable.

Interaction: The DOT statement can be combined only with other horizontal categorization plot statements in the SGPLOT procedure. See [“Plot Type Compatibility” on page 277](#).

Examples: [“About Dot Plots” on page 46](#)
[“Example 7: Adding Statistical Limits to a Dot Plot” on page 506](#)

Syntax

DOT *category-variable* </option(s)>;

Summary of Optional Arguments**Dot options**

DISCRETEOFFSET= *numeric-value*

specifies an amount to offset all dots from discrete category values.

LIMITATTRS= *style-element* <(options)> | (options)

specifies the appearance of the limit lines in the plot.

LIMITS= BOTH | LOWER | UPPER

specifies which limit lines to display.

LIMITSTAT= CLM | STDDEV | STDERR

specifies the statistic for the limit lines.

MARKERATTRS= *style-element* <(options)> | (options)

specifies the appearance of the markers in the plot.

NOSTATLABEL

removes the statistic name from the axis and legend labels.

NUMSTD= *n*

specifies the number of standard units for the limit lines when you specify LIMITSTAT= STDDEV or LIMITSTAT= STDERR.

RESPONSE= *response-variable*

specifies a numeric response variable for the plot.

STAT= FREQ | MEAN | SUM

specifies the statistic for the horizontal axis.

Group options**CLUSTERWIDTH=** *numeric-value*

specifies the width of the group clusters as a fraction of the midpoint spacing.

GROUP= *variable*

specifies a variable that is used to group the data.

GROUPDISPLAY= CLUSTER | OVERLAY

specifies how to display grouped dots.

GROUPORDER= ASCENDING | DESCENDING | DATA

specifies the ordering of dots within a group.

Label options**DATALABEL** <= *variable*>

displays a label for each data point.

DATALABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the DATALABEL= option.

Plot options**ALPHA=** *numeric-value*

specifies the confidence level for the confidence limits.

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set.

CATEGORYORDER= RESPASC | RESPDESC

specifies the order in which the response values are arranged.

FREQ= *numeric-variable*

specifies how many times observations are repeated for computational purposes.

LEGENDLABEL= “*text-string*”

specifies the label that identifies the dot plot in the legend.

MISSING

processes missing values as valid category value and creates a dot for it.

NAME= “*text-string*”

specifies a name for the plot.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the plot.

URL= *character-variable*

specifies a character variable that contains URLs for Web pages to be displayed when parts of the plot are selected within an HTML page.

WEIGHT= *numeric-variable*

specifies how observations are weighted.

Required Argument***category-variable***

specifies the variable that classifies the observations into distinct subsets.

Optional Arguments**ALPHA=** *numeric-value*

specifies the confidence level for the confidence limits. Specify a number between 0.00 (100% confidence) and 1.00 (0% confidence).

Default: .05

Interactions:

This option has no effect if you do not specify LIMITSTAT=CLM.

If your plot is overlaid with other categorization plots, then the first ALPHA value that you specify is used for all of the plots.

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph. For more information, see [“Using SG Attribute Maps to Control Visual Attributes” on page 605](#).

CATEGORYORDER= RESPASC | RESPDESC

specifies the order in which the response values are arranged. Specify one of the following values:

RESPASC

sorts the response values in ascending order.

RESPDESC

sorts the response values in descending order.

Default: By default, the plot is sorted in ascending order based on the category values.

Restrictions:

This option takes effect only when the plot statement specifies a response variable and the axis for that variable is numeric. If the axis is not numeric, an error is generated and a message is written to the SAS log.

Uniform scaling and response sorting cannot occur on the same axis. If the UNIFORM= option is used in the SGPLOT statement, the UNIFORM option is ignored for the sorted response axis and a note is generated in the log. The UNIFORM= option is applied to the other axes and groups if requested. Note that the UNIFORM= option can selectively apply scaling to only the X or Y axis.

Interaction: If a group variable is specified in the plot statement, the response sort option is ignored and a note is generated in the log. (You can use a BY statement in the procedure if you want to group data.)

Note: If CATEGORYORDER is specified in multiple statements, the procedure sorts by the last statement in which it is specified.

CLUSTERWIDTH= *numeric-value*

specifies the width of the group clusters as a fraction of the midpoint spacing. Specify a value from 0.0 (narrowest) to 1.0 (widest).

Default: 0.8

Interaction: This option is applicable only when the GROUP option is specified, when GROUPDISPLAY=CLUSTER, and when the category axis is discrete.

DATALABEL <= *variable*>

displays a label for each data point. If you specify a variable, then the values of that variable are used for the data labels. If you do not specify a variable, then the values of the calculated response are used for the data labels.

DATALABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the DATALABEL= option. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

Here is an example that specifies a style element:

DATALABELATTRS=GraphLabelText

options can be one or more of the following:

COLOR= *color*

specifies the color of the label characters. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphDataText style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

FAMILY= “*font-family*”

specifies the font family for the label characters. The SAS ODS styles use TrueType system fonts. For more information, see “TrueType Fonts Supplied by SAS” in Chapter 15 of *SAS Language Reference: Concepts*.

Default: The default font family is specified by the FontFamily attribute of the GraphDataText style element in the current style.

SIZE= *n* <*units*>

specifies the font size of the label characters. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default font size is specified by the FontSize attribute of the GraphDataText style element in the current style.

STYLE= ITALIC | NORMAL

specifies whether the label characters are italic (ITALIC) or normal (NORMAL).

Default: The default font style is specified by the FontStyle attribute of the GraphDataText style element in the current style.

WEIGHT= BOLD | NORMAL

specifies whether the label characters are bold (BOLD) or normal (NORMAL).

Default: The default font weight is specified by the FontWeight attribute of the GraphDataText style element in the current style.

Here is an example expression:

```
DATALABELATTRS=(Color=Green Family=Arial Size=8 Style=Italic Weight=Bold)
```

Default: The default style element is GraphDataText.

Interaction: This option has no effect unless the DATALABEL option is also specified.

DISCRETEOFFSET= *numeric-value*

specifies an amount to offset all dots from discrete category values. Specify a value from -0.5 (left offset) to +0.5 (right offset).

Default: 0.0 (no offset)

Requirement: This option is applicable only when the category axis is discrete.

FREQ= *numeric-variable*

specifies how many times observations are repeated for computational purposes. specifies that each observation is repeated *n* times for computational purposes, where *n* is the value of the numeric variable. If *n* is not an integer, then it is truncated to an integer. If *n* is less than 1 or missing, then it is excluded from the analysis.

Interaction: If your plot is overlaid with other categorization plots, then the first FREQ variable that you specified is used for all of the plots.

GROUP= *variable*

specifies a variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes.

Interaction: If you specify more than one categorization plot statement, then all of the plots must specify the same GROUP variable. If you do not specify the same GROUP= option for all of the categorization plots, then the GROUP= option has no effect.

GROUPDISPLAY= CLUSTER | OVERLAY

specifies how to display grouped dots.

CLUSTER

grouped items are drawn adjacent to each other.

OVERLAY

grouped items are drawn at the exact coordinate and might overlap. Each group is represented by unique visual attributes derived from the GraphData1...

GraphData*n* style elements in the current style.

Default: OVERLAY

Restriction: GROUPDISPLAY=CLUSTER can take effect only when the category axis is discrete. If necessary, use a TYPE=DISCRETE option on the axis statement.

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

GROUPORDER= ASCENDING | DESCENDING | DATA

specifies the ordering of dots within a group.

ASCENDING

displays each group in ascending group value.

DESCENDING

displays each group in descending group value.

DATA

displays each group in data order of the group variable.

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

LEGENDLABEL= “*text-string*”

specifies the label that identifies the dot plot in the legend. By default, the label of the response variable is used. If there is no response variable label, then the name of the response variable and the computed statistic (SUM or MEAN) is used. If the RESPONSE= option is not used, then the legend label is “Frequency”.

Interaction: The LEGENDLABEL= option has no effect if you also specify the GROUP= option in the same plot statement.

LIMITATTRS= *style-element* <(options)> | (options)

specifies the appearance of the limit lines in the plot. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: The default color is specified by the ContrastColor attribute of the GraphError style element in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see [“Line Patterns” on page 663](#).

Default: The default line pattern is specified by the LineStyle attribute of the GraphError style element in the current style.

THICKNESS= *n* *<units>*

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see [“Units of Measurement” on page 659](#).

Default: The default line thickness is specified by the LineThickness attribute of the GraphError style element in the current style.

LIMITS= BOTH | LOWER | UPPER

specifies which limit lines to display. Limits are displayed as heavier line segments with a serif at the end extending horizontally from each dot. Upper limits extend to the right of the dot and lower limits extend to the left of the dot. By default, no limits are displayed unless you specify either the LIMITS= or LIMITSTAT= option. Specify one of the following values:

BOTH

adds lower and upper limit lines to the plot.

LOWER

adds lower limit lines to the plot.

UPPER

adds upper limit lines to the plot.

Default: By default, no limit lines are displayed. However, if you specify the LIMITSTAT= option, then the default is BOTH.

Interaction: Limit lines are displayed only when you specify STAT= MEAN.

LIMITSTAT= CLM | STDDEV | STDERR

specifies the statistic for the limit lines. Specify one of the following statistics:

CLM

confidence limits

STDDEV

standard deviation

STDERR

standard error

Default: CLM

Interactions:

If you specify the LIMITSTAT= option only, then the default value for the LIMITS= option is BOTH.

Limits lines are displayed only when you specify STAT=MEAN.

MARKERATTRS= *style-element* *<(options)>* | *(options)*

specifies the appearance of the markers in the plot. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the markers. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see [“Color-Naming Schemes” in Chapter 14 of SAS/GRAPH: Reference](#).

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphDataDefault style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData n style elements in the current style.

SIZE= n <units>

specifies the size of the markers. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see [“Units of Measurement” on page 659](#).

Default: The default marker size is specified by the MarkerSize attribute of the GraphDataDefault style element in the current style.

SYMBOL= *symbol-name*

specifies the symbol for the markers. For a list of valid marker symbols, see [“Marker Symbols” on page 661](#).

Default: For ungrouped data, the default marker symbol is specified by the MarkerSymbol attribute of the GraphDataDefault style element in the current style. For grouped data, the default marker symbol is specified by the MarkerSymbol attribute of the GraphData1 ... GraphData n style elements in the current style.

MISSING

processes missing values as valid category value and creates a dot for it.

NAME= *“text-string”*

specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note: The name is case-sensitive. When you refer to the specified name in other statements, you must use the same exact capitalization.

NOSTATLABEL

removes the statistic name from the axis and legend labels.

NUMSTD= n

specifies the number of standard units for the limit lines when you specify LIMITSTAT= STDDEV or LIMITSTAT= STDERR. You can specify any positive number, including decimals.

Default: 1

RESPONSE= *response-variable*

specifies a numeric response variable for the plot. The summarized values of the response variable for each category value are displayed on the horizontal axis.

STAT= FREQ | MEAN | SUM

specifies the statistic for the horizontal axis. Specify one of the following:

FREQ

the frequencies for the category variable. This is the default value when you do not specify the RESPONSE= option.

MEAN

the mean of the response variable.

SUM

the sum of the response variable. This is the default value when you specify the RESPONSE= option.

If you do not specify the RESPONSE= option, then only the FREQ statistic can be used. If you specify RESPONSE=, then you can use either the SUM or MEAN statistics.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the plot. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default: 0.0

URL= *character-variable*

specifies a character variable that contains URLs for Web pages to be displayed when parts of the plot are selected within an HTML page.

Default: By default, no HTML links are created.

Interactions:

This option affects graphics output that is created through the ODS HTML destination only. For more information about ODS destinations, see [“Specifying a Destination” on page 649](#).

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For more information, see [“Using the ODS GRAPHICS Statement” on page 651](#).

WEIGHT= *numeric-variable*

specifies how observations are weighted. Each observation is weighted by a factor of w for computational purposes, where w is the value of the numeric variable. w can be any numeric value. If w is 0, negative or missing, then that observation is excluded from the analysis.

Interaction: If your plot is overlaid with other categorization plots, then the first WEIGHT variable that you specified is used for all of the plots.

ELLIPSE Statement

Adds a confidence or prediction ellipse to another plot.

Restriction: In order to produce useful output, the ELLIPSE statement should be used with another plot statement that uses numeric axes.

Examples: [“About Ellipse Plots” on page 33](#)
[“Example 5: Adding a Prediction Ellipse to a Scatter Plot” on page 503](#)

Syntax

ELLIPSE *X= numeric-variable Y= numeric-variable* </ option(s)>;

Summary of Optional Arguments

Ellipse options

ALPHA= *numeric-value*

specifies the confidence level for the ellipse.

CLIP

specifies that the data for the ellipse is ignored when determining the data ranges for the axes.

FILL | NOFILL

specifies whether the area fill is visible.

FILLATTRS= *style-element* | (COLOR= *color*)

specifies the appearance of the area fill for the ellipse.

LINEATTRS= *style-element* <(options)> | (options)

specifies the appearance of the outlines for the ellipse.

OUTLINE | NOOUTLINE

specifies whether the outlines of the ellipse are visible.

TYPE = MEAN | PREDICTED

specifies the type of ellipse.

Plot options

FREQ= *numeric-variable*

specifies how many times observations are repeated for computational purposes.

LEGENDLABEL= *“text-string”*

specifies a label that identifies the ellipse in the legend.

NAME= *“text-string”*

specifies a name for the plot.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the area fill and outline, if displayed.

X2AXIS

assigns the X variable to the secondary (top) horizontal axis.

Y2AXIS

assigns the Y variable to the secondary (right) vertical axis.

Required Arguments

X= *numeric-variable*

specifies a numeric variable for the X axis.

Y= *numeric-variable*

specifies a numeric variable for the Y axis.

Optional Arguments

ALPHA= *numeric-value*

specifies the confidence level for the ellipse. Specify a number between 0.00 (100% confidence) and 1.00 (0% confidence).

Default: .05

CLIP

specifies that the data for the ellipse is ignored when determining the data ranges for the axes. By default, the data for the ellipse is considered when determining the data ranges for the axes.

Interaction: This option is automatically set if you specify UNIFORM=SCALE or UNIFORM=ALL option in the PROC SGPLOT statement.

FILL | NOFILL

specifies whether the area fill is visible. The FILL option shows the area fill. The NOFILL option hides the area fill.

Default: The default status of the area fill is specified by the DisplayOpts attribute of the GraphEllipse style element in the current style.

Interactions:

Specifying FILL also hides any visible outlines.

If NOFILL and NOOUTLINE are both specified, then both options are ignored.

FILLATTRS= *style-element* | (**COLOR=** *color*)

specifies the appearance of the area fill for the ellipse. You can specify the color of the fill by using a style element or by using the COLOR= suboption. You can specify

colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: The default color is specified by the Color attribute of the GraphDataDefault style element in the current style.

Interaction: This option has no effect if you specify the NOFILL option.

FREQ= *numeric-variable*

specifies how many times observations are repeated for computational purposes. Each observation is repeated *n* times for computational purposes, where *n* is the value of the numeric variable. If *n* is not an integer, then it is truncated to an integer. If *n* is less than 1 or missing, then it is excluded from the analysis.

LEGENDLABEL= “*text-string*”

specifies a label that identifies the ellipse in the legend. By default, the label describes the confidence value of the ellipse and the type of ellipse. For example, “95% Prediction Ellipse.”

LINEATTRS= *style-element* <(options)> | (options)

specifies the appearance of the outlines for the ellipse. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: The default color is specified by the ContrastColor attribute of the GraphDataDefault style element in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see [“Line Patterns” on page 663](#).

Default: The default line pattern is specified by the LineStyle attribute of the GraphDataDefault style element in the current style.

THICKNESS= *n* <units>

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see [“Units of Measurement” on page 659](#).

Default: The default line thickness is specified by the LineThickness attribute of the GraphDataDefault style element in the current style.

NAME= “*text-string*”

specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note: The name is case-sensitive. When you refer to the specified name in other statements, you must use the same exact capitalization.

OUTLINE | NOOUTLINE

specifies whether the outlines of the ellipse are visible. The OUTLINE option shows the outlines and hides the fill. The NOOUTLINE option hides the outlines.

Default: The default status of the outlines is specified by the DisplayOpts attribute of the GraphEllipse style element in the current style.

Interactions:

Specifying OUTLINE also hides the fill color.

If NOOUTLINE and NOFILL are both specified, then both options are ignored.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the area fill and outline, if displayed. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default: 0.5

TYPE = MEAN | PREDICTED

specifies the type of ellipse. MEAN specifies a confidence ellipse for the population mean. PREDICTED specifies a prediction ellipse for a new observation. Both ellipse types assume a bivariate normal distribution.

Default: PREDICTED

X2AXIS

assigns the X variable to the secondary (top) horizontal axis.

Y2AXIS

assigns the Y variable to the secondary (right) vertical axis.

HBAR Statement

Creates a horizontal bar chart that summarizes the values of a category variable.

Interactions: The HBAR statement can be combined only with other categorization plot statements in the SGPLOT procedure. See [“Plot Type Compatibility” on page 277](#) . When used with particular styles, the HBAR statement produces fill patterns for grouped bars. For more information about fill patterns, see [“Using Fill Patterns to Distinguish Grouped Bar Charts” on page 636](#).

Example: [“About Bar Charts” on page 42](#)

Syntax

HBAR *category-variable* </ option(s)>;

Summary of Optional Arguments

Bar options

BARWIDTH= *numeric-value*

specifies the width of the bars as a ratio of the maximum possible width.

DATASKIN= NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN

specifies a special effect to be used on all filled bars.

DISCRETEOFFSET= *numeric-value*

specifies an amount to offset all bars from the category midpoints.

FILL | NOFILL

specifies whether the bars are filled.

FILLATTRS= *style-element* | (COLOR= *color*)

specifies the appearance of the fill for the bars.

LIMITATTRS= *style-element* <(options)> | (options)

specifies the appearance of the limit lines in the plot.

LIMITS= BOTH | LOWER | UPPER

specifies which limit lines to display.

LIMITSTAT= CLM | STDDEV | STDERR

specifies the statistic for the limit lines.

MISSING

processes missing values as a valid category value and creates a bar for it.

NOSTATLABEL

removes the statistic name from the axis and legend labels.

NUMSTD=*n*

specifies the number of standard units for the limit lines, when you specify LIMITSTAT= STDDEV or LIMITSTAT= STDERR.

OUTLINE | NOOUTLINE

specifies whether the bars have outlines.

RESPONSE= *response-variable*

specifies a numeric response variable for the plot.

STAT= FREQ | MEAN | SUM

specifies the statistic for the horizontal axis.

Group options

CLUSTERWIDTH= *numeric-value*

specifies the cluster width as a ratio of the maximum width.

GROUP= *variable*

specifies a variable that is used to group the data.

GROUPDISPLAY= STACK | CLUSTER

specifies how to display grouped bars.

GROUPORDER= ASCENDING | DESCENDING | DATA

specifies the ordering of bars within a group.

Label options

DATALABEL <= *variable*>

displays a label for each data point.

DATALABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the DATALABEL= option.

Plot options

ALPHA= *numeric-value*

specifies the confidence level for the confidence limits.

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set.

CATEGORYORDER= RESPASC | RESPDESC

specifies the order in which the response values are arranged.

FREQ= *numeric-variable*

specifies how many times observations are repeated for computational purposes.

LEGENDLABEL= "*text-string*"

specifies the label that identifies the bar chart in the legend.

NAME= "*text-string*"

specifies a name for the plot.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the bars and limits, if displayed.

URL= *character-variable*

specifies a character variable that contains URLs for Web pages to be displayed when parts of the plot are selected within an HTML page.

WEIGHT= *numeric-variable*

specifies how observations are weighted.

Required Argument

category-variable

specifies the variable that classifies the observations into distinct subsets.

Optional Arguments

ALPHA= *numeric-value*

specifies the confidence level for the confidence limits. Specify a number between 0.00 (100% confidence) and 1.00 (0% confidence).

Default: .05

Interactions:

This option has no effect if you do not specify LIMITSTAT=CLM.

If your plot is overlaid with other categorization plots, then the first ALPHA value that you specify is used for all of the plots.

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph. For more information, see [“Using SG Attribute Maps to Control Visual Attributes” on page 605](#).

BARWIDTH= *numeric-value*

specifies the width of the bars as a ratio of the maximum possible width. The maximum width is equal to the distance between the center of each bar and the centers of the adjacent bars. Specify a value from 0.0 (narrowest) to 1.0 (widest).

For example, if you specify a width of **1**, then there is no space between the bars. If you specify a width of **.5**, then the width of the bars is equal to the space between the bars.

Default: .8

1.0 when the GROUP option is specified and GROUPDISPLAY=CLUSTER

Interaction: When the GROUP option is specified, the bar width is determined by the maximum number of bars in any one group cluster. All bars are drawn with the same width, with the cluster being positioned symmetrically around the midpoint.

CATEGORYORDER= RESPASC | RESPDESC

specifies the order in which the response values are arranged. Specify one of the following values:

RESPASC

sorts the response values in ascending order.

RESPDESC

sorts the response values in descending order.

Default: By default, the plot is sorted in ascending order based on the category values.

Restrictions:

This option takes effect only when the plot statement specifies a response variable and the axis for that variable is numeric. If the axis is not numeric, an error is generated and a message is written to the SAS log.

Uniform scaling and response sorting cannot occur on the same axis. If the UNIFORM= option is used in the SGPLOT statement, the UNIFORM option is ignored for the sorted response axis and a note is generated in the log. The UNIFORM= option is applied to the other axes and groups if requested. Note that the UNIFORM= option can selectively apply scaling to only the X or Y axis

Interaction: If a group variable is specified in the plot statement, the response sort option is ignored and a note is generated in the log. (You can use a BY statement in the procedure if you want to group data.)

Note: If CATEGORYORDER is specified in multiple statements, the procedure sorts by the last statement in which it is specified.

CLUSTERWIDTH= *numeric-value*

specifies the cluster width as a ratio of the maximum width. Specify a value from 0.0 (narrowest) to 1.0 (widest).

CLUSTERWIDTH is the fraction of the midpoint spacing used by all bars that are clustered around a midpoint (category value). The bar width is applied to the maximum bar spacing divided by the maximum number of bars in any one cluster.

Default: 0.8

Interaction: This option is applicable only when the GROUP option is specified and when GROUPDISPLAY=CLUSTER.

DATALABEL <= *variable*>

displays a label for each data point. If you specify a variable, then the values of that variable are used for the data labels. If you do not specify a variable, then the values of the calculated response are used for the data labels.

DATALABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the DATALABEL= option. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

Here is an example that specifies a style element:

DATALABELATTRS=GraphLabelText

options can be one or more of the following:

COLOR= *color*

specifies the color of the label characters. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphDataText style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

FAMILY= “*font-family*”

specifies the font family for the label characters. The SAS ODS styles use TrueType system fonts. For more information, see “TrueType Fonts Supplied by SAS” in Chapter 15 of *SAS Language Reference: Concepts*.

Default: The default font family is specified by the FontFamily attribute of the GraphDataText style element in the current style.

SIZE= *n* <*units*>

specifies the font size of the label characters. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default font size is specified by the `FontSize` attribute of the `GraphDataText` style element in the current style.

`STYLE= ITALIC | NORMAL`

specifies whether the label characters are italic (ITALIC) or normal (NORMAL).

Default: The default font style is specified by the `FontStyle` attribute of the `GraphDataText` style element in the current style.

`WEIGHT= BOLD | NORMAL`

specifies whether the label characters are bold (BOLD) or normal (NORMAL).

Default: The default font weight is specified by the `FontWeight` attribute of the `GraphDataText` style element in the current style.

Here is an example expression:

```
DATALABELATTRS=(Color=Green Family=Arial Size=8 Style=Italic Weight=Bold)
```

Default: The default style element is `GraphDataText`.

Interaction: This option has no effect unless the `DATALABEL` option is also specified.

DATASKIN= NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN

specifies a special effect to be used on all filled bars. Specify one of the following (the samples were generated using the `HTMLBlue` style):

NONE	
CRISP	
GLOSS	
MATTE	
PRESSED	
SHEEN	

Default: NONE

Interactions:

This option has no effect if the applied style uses a fill pattern and `FILL` is disabled for the style. If the style has both `FILL` and `FILLPATTERN` enabled, data skins will work. For more information about fill patterns, see [“Using Fill Patterns to Distinguish Grouped Bar Charts” on page 636](#). For instructions about

viewing the contents of style templates, see [“About Styles and Style Elements” on page 620](#).

This option has no effect if you also specify NOFILL.

DISCRETEOFFSET= *numeric-value*

specifies an amount to offset all bars from the category midpoints. Specify a value from -0.5 (left offset) to +0.5 (right offset).

Default: 0.0 (no offset)

FILL | NOFILL

specifies whether the bars are filled. The FILL option shows the fill color for the bars. The NOFILL option hides the fill color for the bars.

Default: FILL

Interactions:

Specifying FILL also hides the outlines.

If NOFILL and NOOUTLINE are both specified, then both options are ignored.

FILLATTRS= *style-element* | (COLOR= *color*)

specifies the appearance of the fill for the bars. You can specify the color of the fill by using a style element or by using the COLOR= suboption. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphDataDefault style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

Interaction: This option has no effect if you specify the NOFILL option.

FREQ= *numeric-variable*

specifies how many times observations are repeated for computational purposes. specifies that each observation is repeated *n* times for computational purposes, where *n* is the value of the numeric variable. If *n* is not an integer, then it is truncated to an integer. If *n* is less than 1 or missing, then it is excluded from the analysis.

Interaction: If your plot is overlaid with other categorization plots, then the first FREQ variable that you specified is used for all of the plots.

GROUP= *variable*

specifies a variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes.

GROUPDISPLAY= STACK | CLUSTER

specifies how to display grouped bars.

STACK

groups are overlaid without any clustering. Each group is represented by unique visual attributes derived from the GraphData1... GraphData*n* style elements in the current style.

CLUSTER

displays group values as separate adjacent bars that replace the single category bar. Each set of group values is centered at the midpoint tick mark for the category.

Default: STACK

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

GROUPORDER= ASCENDING | DESCENDING | DATA

specifies the ordering of bars within a group.

ASCENDING

displays each group in ascending group value.

DESCENDING

displays each group in descending group value.

DATA

displays each group in data order of the group variable.

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

LEGENDLABEL= “text-string”

specifies the label that identifies the bar chart in the legend. By default, the label of the RESPONSE= variable is used. If there is no response variable label, the name of the response variable and the computed statistic (SUM or MEAN) is used. If the RESPONSE= option is not used, the legend label is “Frequency”.

Interaction: The LEGENDLABEL= option has no effect if you also specify the GROUP= option in the same plot statement.

LIMITATTRS= style-element <(options)> | (options)

specifies the appearance of the limit lines in the plot. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= color

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: The default color is specified by the ContrastColor attribute of the GraphError style element in the current style.

PATTERN= line-pattern

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see [“Line Patterns” on page 663](#).

Default: The default line pattern is specified by the LineStyle attribute of the GraphError style element in the current style.

THICKNESS= n <units>

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see [“Units of Measurement” on page 659](#).

Default: The default line thickness is specified by the LineThickness attribute of the GraphError style element in the current style.

Interaction: This option has no effect unless you also specify either the LIMITS= or LIMITSTAT= option.

LIMITS= BOTH | LOWER | UPPER

specifies which limit lines to display. Limits are displayed as heavier line segments with a serif at the end extending from each bar. Upper limits extend to the right of the bar and lower limits extend to the left of the bar. Specify one of the following values:

BOTH

adds lower and upper limit lines to the plot.

LOWER

adds lower limit lines to the plot.

UPPER

adds upper limit lines to the plot.

Default: By default, no limits are displayed unless you specify either the LIMITS= or LIMITSTAT= option. If you specify the LIMITSTAT= option only, then LIMITS=BOTH is the default.

Interactions:

Limit lines are displayed only when you specify STAT= MEAN.

The LIMITS= option has no effect if you also specify the GROUP= option in the plot statement.

LIMITSTAT= CLM | STDDEV | STDERR

specifies the statistic for the limit lines. Specify one of the following statistics:

CLM

confidence limits

STDDEV

standard deviation

STDERR

standard error

Default: CLM

Interactions:

If you specify the LIMITSTAT= option only, then the default value for the LIMITS= option is BOTH.

Limits lines are displayed only when you specify STAT=MEAN.

The LIMITSTAT= option has no effect if you also specify the GROUP= option in the plot statement.

MISSING

processes missing values as a valid category value and creates a bar for it.

NAME= “text-string”

specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note: The name is case-sensitive. When you refer to the specified name in other statements, you must use the same exact capitalization.

NOSTATLABEL

removes the statistic name from the axis and legend labels.

NUMSTD= *n*

specifies the number of standard units for the limit lines, when you specify LIMITSTAT= STDDEV or LIMITSTAT= STDERR. You can specify any positive number, including decimals.

Default: 1

OUTLINE | NOOUTLINE

specifies whether the bars have outlines. The OUTLINE option shows the outlines. The NOOUTLINE option hides the outlines.

Default: OUTLINE

Interactions:

Specifying OUTLINE also hides the fill color.

If NOOUTLINE and NOFILL are both specified, then both options are ignored.

RESPONSE= *response-variable*

specifies a numeric response variable for the plot. The summarized values of the response variable are displayed for each value on the horizontal axis.

STAT= FREQ | MEAN | SUM

specifies the statistic for the horizontal axis. Specify one of the following statistics:

FREQ

the frequencies for the category variable. This is the default value when you do not specify the RESPONSE= option.

MEAN

the mean of the response variable.

SUM

the sum of the response variable. This is the default value when you specify the RESPONSE= option.

If you do not specify the RESPONSE= option, then only the FREQ statistic can be used. If you specify RESPONSE=, then you can use either the SUM or MEAN statistics.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the bars and limits, if displayed. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default: 0.0

URL= *character-variable*

specifies a character variable that contains URLs for Web pages to be displayed when parts of the plot are selected within an HTML page.

Default: By default, no HTML links are created.

Interactions:

This option affects graphics output that is created through the ODS HTML destination only. For more information about ODS destinations, see [“Specifying a Destination” on page 649](#).

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For more information, see [“Using the ODS GRAPHICS Statement” on page 651](#).

WEIGHT= *numeric-variable*

specifies how observations are weighted. Each observation is weighted by a factor of w for computational purposes, where w is the value of the numeric variable. w can be any numeric value. If w is 0, negative or missing, then that observation is excluded from the analysis.

Interaction: If your plot is overlaid with other categorization plots, then the first WEIGHT variable that you specified is used for all of the plots.

HBARPARM Statement

Creates a horizontal bar chart based on a pre-summarized response value for each unique value of the category variable. You can also assign variables to the upper and lower limits.

Requirement: The data must contain only one response value per unique category variable. If more than one value is found, a warning is written to the SAS log, and the graph might produce unpredictable results.

Interactions: The HBARPARM statement can be combined only with other basic plot statements in the SGPLOT procedure. See [“Plot Type Compatibility” on page 277](#).

When used with particular styles, the HBARPARM statement produces fill patterns for grouped bars. For more information about fill patterns, see [“Using Fill Patterns to Distinguish Grouped Bar Charts” on page 636](#).

Note: An important distinction between HBARPARM and HBAR is that the response variable is required for HBARPARM. In addition, the response variable should contain pre-summarized computed values such as a sum or a mean.

Example: [“About Bar Charts” on page 42](#)

Syntax

HBARPARM CATEGORY= *category-variable* RESPONSE= *numeric-variable* </option(s)>;

Summary of Optional Arguments

Bar options

BARWIDTH= *numeric-value*

specifies the width of the bars as a ratio of the maximum possible width.

DATASKIN= NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN

specifies a special effect to be used on all filled bars.

DISCRETEOFFSET= *numeric-value*

specifies an amount to offset all bars from the category midpoints.

FILL | **NOFILL**

specifies whether the bars are filled.

FILLATTRS= *style-element* | (COLOR= *color*)

specifies the appearance of the fill for the bars.

LIMITATTRS= *style-element* <(options)> | (options)

specifies the appearance of the limit lines in the plot.

LIMITLOWER= *numeric-variable*

specifies values for the lower endpoints on the limit lines.

LIMITUPPER= *numeric-variable*

specifies values for the upper endpoints on the limit lines.

MISSING

processes missing values as a valid category value and creates a bar for it.

OUTLINE | **NOOUTLINE**

specifies whether the bars have outlines.

Group options

CLUSTERWIDTH= *numeric-value*

specifies the cluster width as a ratio of the maximum width.

GROUP= *variable*

specifies a variable that is used to group the data.

GROUPORDER= ASCENDING | DESCENDING | DATA

specifies the ordering of bars within a group.

Label options

DATALABEL <= *variable*>

displays a label for each data point.

DATALABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the DATALABEL= option.

Plot options**ATTRID=** *variable*

specifies the value of the ID variable in an attribute map data set.

LEGENDLABEL= *“text-string”*

specifies the label that identifies the bar chart in the legend.

NAME= *“text-string”*

specifies a name for the plot.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the bars and limits, if displayed.

URL= *character-variable*

specifies a character variable that contains URLs for Web pages to be displayed when parts of the plot are selected within an HTML page.

Required Arguments**CATEGORY=** *category-variable*

specifies the variable that categorizes the data. All values are treated as discrete values. The input data for this variable should contain unique values. When the category values are not unique, a warning is logged, and multiple bars are superimposed at the duplicated category values.

The CATEGORY axis is always discrete.

RESPONSE= *numeric-variable*

specifies a numeric response variable. The input data is expected to be pre-summarized computed values (sum, mean, and so on).

Optional Arguments**ATTRID=** *variable*specifies the value of the ID variable in an attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph. For more information, see [“Using SG Attribute Maps to Control Visual Attributes”](#) on page 605.**BARWIDTH=** *numeric-value*

specifies the width of the bars as a ratio of the maximum possible width. The maximum width is equal to the distance between the center of each bar and the centers of the adjacent bars. Specify a value from 0.0 (narrowest) to 1.0 (widest).

For example, if you specify a width of **1**, then there is no space between the bars. If you specify a width of **.5**, then the width of the bars is equal to the space between the bars.**Default:** .8

1.0 when the GROUP option is specified and GROUPDISPLAY=CLUSTER

Interaction: When the GROUP option is specified, the bar width is determined by the maximum number of bars in any one group cluster. All bars are drawn with the same width, with the cluster being positioned symmetrically around the midpoint.**CLUSTERWIDTH=** *numeric-value*

specifies the cluster width as a ratio of the maximum width. Specify a value from 0.0 (narrowest) to 1.0 (widest).

CLUSTERWIDTH is the fraction of the midpoint spacing used by all bars that are clustered around a midpoint (category value). The bar width is applied to the maximum bar spacing divided by the maximum number of bars in any one cluster.

Default: 0.8

Interaction: This option is applicable only when the GROUP option is specified and when GROUPDISPLAY=CLUSTER.

DATALABEL <= *variable*>

displays a label for each data point. If you specify a variable, then the values of that variable are used for the data labels. If you do not specify a variable, then the values of the calculated response are used for the data labels.

DATALABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the DATALABEL= option. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

Here is an example that specifies a style element:

DATALABELATTRS=GraphLabelText

options can be one or more of the following:

COLOR= *color*

specifies the color of the label characters. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphDataText style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

FAMILY= “*font-family*”

specifies the font family for the label characters. The SAS ODS styles use TrueType system fonts. For more information, see “TrueType Fonts Supplied by SAS” in Chapter 15 of *SAS Language Reference: Concepts*.

Default: The default font family is specified by the FontFamily attribute of the GraphDataText style element in the current style.

SIZE= *n* <*units*>

specifies the font size of the label characters. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default font size is specified by the FontSize attribute of the GraphDataText style element in the current style.

STYLE= ITALIC | NORMAL

specifies whether the label characters are italic (ITALIC) or normal (NORMAL).

Default: The default font style is specified by the FontStyle attribute of the GraphDataText style element in the current style.

WEIGHT= BOLD | NORMAL

specifies whether the label characters are bold (BOLD) or normal (NORMAL).

Default: The default font weight is specified by the FontWeight attribute of the GraphDataText style element in the current style.



Here is an example expression:

DATALABELATTRS=(Color=Green Family=Arial Size=8 Style=Italic Weight=Bold)

Default: The default style element is GraphDataText.

Interaction: This option has no effect unless the DATALABEL option is also specified.

DATASKIN= NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
 specifies a special effect to be used on all filled bars. Specify one of the following (the samples were generated using the HTMLBlue style):

NONE	
CRISP	
GLOSS	
MATTE	
PRESSED	
SHEEN	

Default: NONE

Interactions:
 This option has no effect if the applied style uses a fill pattern and FILL is disabled for the style. If the style has both FILL and FILLPATTERN enabled, data skins will work. For more information about fill patterns, see [“Using Fill Patterns to Distinguish Grouped Bar Charts” on page 636](#). For instructions about viewing the contents of style templates, see [“About Styles and Style Elements” on page 620](#).
 This option has no effect if you also specify NOFILL.

DISCRETEOFFSET= *numeric-value*
 specifies an amount to offset all bars from the category midpoints. Specify a value from -0.5 (left offset) to +0.5 (right offset).
Default: 0.0 (no offset)

FILL | NOFILL
 specifies whether the bars are filled. The FILL option shows the fill color for the bars. The NOFILL option hides the fill color for the bars.
Default: FILL

Interactions:
 Specifying FILL also hides the outlines.
 If NOFILL and NOOUTLINE are both specified, then both options are ignored.

FILLATTRS= *style-element* | (COLOR= *color*)

specifies the appearance of the fill for the bars. You can specify the color of the fill by using a style element or by using the COLOR= suboption. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphDataDefault style element in the current style. For grouped data, the default color is specified by the Color attribute of the GraphData1... GraphData*n* style elements in the current style.

Interaction: This option has no effect if you specify the NOFILL option.

GROUP= *variable*

specifies a variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes.

GROUPORDER= ASCENDING | DESCENDING | DATA

specifies the ordering of bars within a group.

ASCENDING

displays each group in ascending group value.

DESCENDING

displays each group in descending group value.

DATA

displays each group in data order of the group variable.

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

LEGENDLABEL= “*text-string*”

specifies the label that identifies the bar chart in the legend. By default, the label of the RESPONSE= variable is used. If there is no response variable label, the name of the response variable is used.

Interaction: The LEGENDLABEL= option has no effect if you also specify the GROUP= option in the same plot statement.

LIMITATTRS= *style-element* <(options)> | (options)

specifies the appearance of the limit lines in the plot. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: The default color is specified by the ContrastColor attribute of the GraphError style element in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see “[Line Patterns](#)” on page 663.

Default: The default line pattern is specified by the LineStyle attribute of the GraphError style element in the current style.

THICKNESS= *n* <units>

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see “[Units of Measurement](#)” on page 659.

Default: The default line thickness is specified by the LineThickness attribute of the GraphError style element in the current style.

LIMITLOWER= *numeric-variable*

specifies values for the lower endpoints on the limit lines. Limit lines are displayed as line segments with a serif at the end.

Default: The lower segments of the limit lines are not displayed. (Limit lines are displayed only if either LIMITLOWER= or LIMITUPPER= is specified.)

Interactions:

If LIMITUPPER= is also specified, then the plot displays the lower and upper segments of the limit lines.

This option is ignored if GROUP= is specified and GROUPDISPLAY=STACK.

The appearance of the limit lines can be controlled by the LIMITATTRS= option.

LIMITUPPER= *numeric-variable*

specifies values for the upper endpoints on the limit lines. Limit lines are displayed as line segments with a serif at the end.

Default: The upper segments of the limit lines are not displayed. (Limit lines are displayed only if either LIMITLOWER= or LIMITUPPER= is specified.)

Interactions:

If LIMITLOWER= is also specified, then the plot displays the lower and upper segments of the limit lines.

This option is ignored if GROUP= is specified and GROUPDISPLAY=STACK.

The appearance of the limit lines can be controlled by the LIMITATTRS= option.

MISSING

processes missing values as a valid category value and creates a bar for it.

NAME= *“text-string”*

specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note: The name is case-sensitive. When you refer to the specified name in other statements, you must use the same exact capitalization.

OUTLINE | NOOUTLINE

specifies whether the bars have outlines. The OUTLINE option shows the outlines. The NOOUTLINE option hides the outlines.

Default: OUTLINE

Interactions:

Specifying OUTLINE also hides the fill color.

If NOOUTLINE and NOFILL are both specified, then both options are ignored.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the bars and limits, if displayed. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default: 0.0

URL= *character-variable*

specifies a character variable that contains URLs for Web pages to be displayed when parts of the plot are selected within an HTML page.

Default: By default, no HTML links are created.

Interactions:

This option affects graphics output that is created through the ODS HTML destination only. For more information about ODS destinations, see [“Specifying a Destination” on page 649](#).

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For more information, see [“Using the ODS GRAPHICS Statement” on page 651](#).

HBOX Statement

Creates a horizontal box plot that shows the distribution of your data.

Interaction: The HBOX statement cannot be used with other plot statements in the SGPLOT procedure. Box plots can be overlaid with other box plots. However, overlaid box plots must have the same category variables.

Examples: [“About Box Plots” on page 37](#)
[“Example 9: Creating a Horizontal Box Plot” on page 508](#)

Syntax

HBOX *analysis-variable* *</ option(s)>*;

Summary of Optional Arguments

Box options

BOXWIDTH= *numeric-value*

specifies the width of the box.

CAPSHAPE= BRACKET | LINE | SERIF

specifies the shape of the whisker cap lines.

CONNECT= MEAN | MEDIAN | Q1 | Q3 | MIN | MAX

specifies that a connect line joins a statistic from box to box.

CONNECTATTRS= *style-element* *<(options)>* | *(options)*

specifies the appearance of the lines that connect multiple boxes.

DISCRETEOFFSET= *numeric-value*

specifies an amount to offset all boxes from the discrete tick marks.

EXTREME

specifies that the whiskers can extend to the maximum and minimum values for the analysis variable, and that outliers are not identified.

FILL | **NOFILL**

specifies whether the boxes are filled with color.

FILLATTRS= *style-element* | (COLOR= *color*)

specifies the appearance of the fill for the boxes.

LINEATTRS= *style-element* *<(options)>* | *(options)*

specifies the appearance of the box outlines.

MEANATTRS= *style-element* *<(options)>* | *(options)*

specifies the appearance of the marker that represents the mean in the box.

MEDIANATTRS= *style-element* *<(options)>* | *(options)*

specifies the appearance of the line that represents the median.

NOCAPS

hides the cap lines for the whiskers.

NOMEAN

hides the mean marker.

NOMEDIAN

hides the median line.

NOOUTLIERS

hides the outliers from the plot.

NOTCHES

specifies that the boxes be notched.

OUTLIERATTRS= *style-element* <(options)> | (options)

specifies the appearance of the marker that represents the outliers.

WHISKERATTRS= *style-element* <(options)> | (options)

specifies the appearance of the whisker and cap lines.

Group options

CLUSTERWIDTH= *numeric-value*

specifies the cluster width as a ratio of the midpoint spacing.

GROUP= *variable*

specifies a variable that is used to group the data.

GROUPDISPLAY= CLUSTER | OVERLAY

specifies how to display grouped boxes.

GROUPORDER= ASCENDING | DESCENDING | DATA

specifies the ordering of boxes within a group.

Label options

DATALABEL <= *variable*>

adds data labels for the outlier markers.

DATALABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the DATALABEL= option.

LABELFAR

specifies that only the far outliers have data labels.

Plot options

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set.

CATEGORY= *category-variable*

specifies the category variable for the plot.

FREQ= *numeric-variable*

specifies how many times observations are repeated for computational purposes.

LEGENDLABEL= "*text-string*"

specifies a label that identifies the box plot in the legend.

MISSING

processes missing values as a valid category value and creates a box for it.

NAME= "*text-string*"

specifies a name for the plot.

PERCENTILE= 1 | 2 | 3 | 4 | 5

specifies a method for computing the percentiles for the plot.

SPREAD

relocates outlier points that have identical values to prevent overlapping.

TRANSPARENCY= *value*

specifies the degree of transparency for the plot.

X2AXIS

assigns the analysis variable to the secondary (top) horizontal axis.

Y2AXIS

assigns the analysis variable to the secondary (right) vertical axis.

Required Argument

analysis-variable

specifies the analysis variable for the plot. If you do not specify the CATEGORY= option, then one box is created for the analysis variable.

Optional Arguments

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph. For more information, see [“Using SG Attribute Maps to Control Visual Attributes” on page 605](#).

BOXWIDTH= *numeric-value*

specifies the width of the box. Specify a value between 0.0 (0% of the available width) and 1.0 (100% of the available width).

Default: 0.4

When GROUP is specified, the default box width is 0.6.

CAPSHAPE= BRACKET | LINE | SERIF

specifies the shape of the whisker cap lines. Specify one of the following values:

BRACKET

displays a straight line with brackets.

LINE

displays a straight line.

SERIF

displays a short straight line.

CATEGORY= *category-variable*

specifies the category variable for the plot. A box plot is created for each distinct value of the category variable.

If you explicitly set the category axis type to LINEAR and use a numeric category variable, the box plot becomes an interval plot. Otherwise, the box plot is discrete.

CLUSTERWIDTH= *numeric-value*

specifies the cluster width as a ratio of the midpoint spacing. Specify a value from 0.1 (narrowest) to 1.0 (widest).

Default: 0.7

Interaction: This option is applicable only when a GROUP is in effect and the category axis is discrete.

CONNECT= MEAN | MEDIAN | Q1 | Q3 | MIN | MAX

specifies that a connect line joins a statistic from box to box.

Interaction: This option applies only when the CATEGORY option is used to generate multiple boxes.

Tip: You can use the CONNECTATTRS option to specify attributes for the connect line.

CONNECTATTRS= *style-element* <(options)> | (options)

specifies the appearance of the lines that connect multiple boxes. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

Here is an example that specifies a style element: **CONNECTATTRS=GraphData3**

options can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphConnectLine style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see “Line Patterns” on page 663.

Default: For ungrouped data, the default line pattern is specified by the LineStyle attribute of the GraphConnectLine style element in the current style. For grouped data, the default line pattern is specified by the LineStyle attribute of the GraphData1 ... GraphData*n* style elements in the current style.

THICKNESS= *n* <units>

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: For ungrouped data, the default line thickness is specified by the LineThickness attribute of the GraphConnectLine style element in the current style. For grouped data, the default line thickness is specified by the LineThickness attribute of the GraphData1 ... GraphData*n* style elements in the current style.

Here is an example expression:

```
CONNECTATTRS=(Color= "light green" Pattern= MediumDash Thickness= 4)
```

Default: The default style element is GraphConnectLine.

Interactions:

This option takes effect only if the CONNECT= option is also specified.

This option is ignored if the GROUP= option is also specified.

DATALABEL <= *variable*>

adds data labels for the outlier markers. If you specified a variable, then the values for that variable are used for the data labels. If you did not specify a variable, then the values of the analysis variable are used.

Note: This option has no effect unless the plot contains outlier points.

DATALABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the DATALABEL= option. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

Here is an example that specifies a style element:

```
DATALABELATTRS=GraphLabelText
```

options can be one or more of the following:

COLOR= *color*

specifies the color of the label characters. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphDataText style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

FAMILY= “*font-family*”

specifies the font family for the label characters. The SAS ODS styles use TrueType system fonts. For more information, see “TrueType Fonts Supplied by SAS” in Chapter 15 of *SAS Language Reference: Concepts*.

Default: The default font family is specified by the FontFamily attribute of the GraphDataText style element in the current style.

SIZE= *n* <*units*>

specifies the font size of the label characters. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default font size is specified by the FontSize attribute of the GraphDataText style element in the current style.

STYLE= ITALIC | NORMAL

specifies whether the label characters are italic (ITALIC) or normal (NORMAL).

Default: The default font style is specified by the FontStyle attribute of the GraphDataText style element in the current style.

WEIGHT= BOLD | NORMAL

specifies whether the label characters are bold (BOLD) or normal (NORMAL).

Default: The default font weight is specified by the FontWeight attribute of the GraphDataText style element in the current style.

Here is an example expression:

```
DATALABELATTRS=(Color=Green Family=Arial Size=8 Style=Italic Weight=Bold)
```

Default: The default style element is GraphDataText.

Interaction: This option has no effect unless the DATALABEL option is also specified.

DISCRETEOFFSET= *numeric-value*

specifies an amount to offset all boxes from the discrete tick marks.

Specify a value from -0.5 (left offset) to +0.5 (right offset). If you specify a value outside of this range, an error message appears in the SAS log and the graph is not produced.

Default: 0.0 (no offset)

EXTREME

specifies that the whiskers can extend to the maximum and minimum values for the analysis variable, and that outliers are not identified. When you do not specify the EXTREME option, the whiskers cannot be longer than 1.5 times the length of the box.

FILL | NOFILL

specifies whether the boxes are filled with color. The FILL option shows the fill color. The NOFILL option hides the fill color.

Default: FILL

FILLATTRS= *style-element* | (COLOR= *color*)

specifies the appearance of the fill for the boxes. You can specify the color of the fill by using a style element or by using the COLOR= suboption. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphDataDefault style element in the current style.

For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

Interaction: This option has no effect if you also specify the NOFILL option.

FREQ= *numeric-variable*

specifies how many times observations are repeated for computational purposes. specifies that each observation is repeated *n* times for computational purposes, where *n* is the value of the numeric variable. If *n* is not an integer, then it is truncated to an integer. If *n* is less than 1 or missing, then it is excluded from the analysis.

Interaction: If your plot is overlaid with other categorization plots, then the first FREQ variable that you specified is used for all of the plots.

GROUP= *variable*

specifies a variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes.

GROUPDISPLAY= CLUSTER | OVERLAY

specifies how to display grouped boxes.

CLUSTER

the boxes are drawn adjacent to each other.

OVERLAY

all the boxes for a given group value are drawn at the exact coordinate and might overlap. Each group is represented by unique visual attributes derived from the GraphData1 ... GraphData*n* style elements in the current style.

Default: CLUSTER for a discrete category axis

OVERLAY for a linear axis

Restriction: GROUPDISPLAY=CLUSTER can take effect only when the category axis is discrete.

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

GROUPORDER= ASCENDING | DESCENDING | DATA

specifies the ordering of boxes within a group.

ASCENDING

displays each group in ascending group value.

DESCENDING

displays each group in descending group value.

DATA

displays each group in data order of the group variable.

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

LABELFAR

specifies that only the far outliers have data labels. Far outliers are points whose distance from the box is more than three times the length of the box.

Note: This option has no effect if you do not specify the DATALABEL option, or if there are no far outliers.

LEGENDLABEL= *“text-string”*

specifies a label that identifies the box plot in the legend. By default, the label of the analysis variable is used.

LINEATTRS= *style-element* *<(options)>* | *(options)*

specifies the appearance of the box outlines. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

Here is an example that specifies a style element: **LINEATTRS=GraphData3**

For more information about style elements, see [“Style Elements for Use with ODS Graphics” on page 640](#).

options can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see [“Color-Naming Schemes” in Chapter 14 of SAS/GRAPH: Reference](#).

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphDataDefault style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see [“Line Patterns” on page 663](#).

Default: For ungrouped data, the default line pattern is specified by the LineStyle attribute of the GraphDataDefault style element in the current style. For grouped data, the default line pattern is specified by the LineStyle attribute of the GraphData1 ... GraphData*n* style elements in the current style.

THICKNESS= *n* *<units>*

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see [“Units of Measurement” on page 659](#).

Default: For ungrouped data, the default line thickness is specified by the LineThickness attribute of the GraphDataDefault style element in the current style. For grouped data, the default line thickness is specified by the LineThickness attribute of the GraphData1 ... GraphData*n* style elements in the current style.

Here is an example expression:

```
LINEATTRS=(Color= "light green" Pattern= MediumDash Thickness= 4)
```

Default: GraphDataDefault is the default style element.

Interactions:

This option takes effect only if the CONNECT= option is also specified.

This option is ignored if the GROUP= option is also specified.

MEANATTRS= *style-element* <(options)> | (options)

specifies the appearance of the marker that represents the mean in the box. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the markers. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphBoxMean style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

SIZE= *n* <units>

specifies the size of the markers. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default marker size is specified by the MarkerSize attribute of the GraphBoxMean style element in the current style.

SYMBOL= *symbol-name*

specifies the symbol for the markers. For a list of valid marker symbols, see “Marker Symbols” on page 661.

Default: For ungrouped data, the default marker symbol is specified by the MarkerSymbol attribute of the GraphBoxMean style element in the current style. For grouped data, the default marker symbol is specified by the MarkerSymbol attribute of the GraphData1 ... GraphData*n* style elements in the current style.

WEIGHT= NORMAL | BOLD

specifies the marker weight of the marker.

Default: NORMAL

Default: GraphBoxMean is the default style element.

MEDIANATTRS= *style-element* <(options)> | (options)

specifies the appearance of the line that represents the median. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphBoxMedian style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see “Line Patterns” on page 663.

Default: For ungrouped data, the default line pattern is specified by the `LineStyle` attribute of the `GraphBoxMedian` style element in the current style. For grouped data, the default line pattern is specified by the `LineStyle` attribute of the `GraphData1` ... `GraphDatan` style elements in the current style.

THICKNESS= *n* *<units>*

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see [“Units of Measurement” on page 659](#).

Default: For ungrouped data, the default line thickness is specified by the `LineThickness` attribute of the `GraphBoxMedian` style element in the current style. For grouped data, the default line thickness is specified by the `LineThickness` attribute of the `GraphData1` ... `GraphDatan` style elements in the current style.

Default: `GraphBoxMedian` is the default style element.

Interaction: This option is ignored if the `NOMEDIAN` option is also specified.

MISSING

processes missing values as a valid category value and creates a box for it.

NAME= *“text-string”*

specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note: The name is case-sensitive. When you refer to the specified name in other statements, you must use the same exact capitalization.

NOCAPS

hides the cap lines for the whiskers.

NOMEAN

hides the mean marker.

NOMEDIAN

hides the median line.

NOOUTLIERS

hides the outliers from the plot.

NOTCHES

specifies that the boxes be notched. The endpoints of the notches are at the following computed locations:

$$\text{median} \pm 1.58 (IQR / \sqrt{N})$$

For a visual description of the parts of a box plot, see [“Details” on page 334](#).

OUTLIERATTRS= *style-element* *<(options)>* | *(options)*

specifies the appearance of the marker that represents the outliers. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the markers. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see [“Color-Naming Schemes” in Chapter 14 of SAS/GRAPH: Reference](#).

Default: For ungrouped data, the default color is specified by the `ContrastColor` attribute of the `GraphOutlier` style element in the current style. For grouped

data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

SIZE= *n* <*units*>

specifies the size of the markers. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see [“Units of Measurement” on page 659](#).

Default: The default marker size is specified by the MarkerSize attribute of the GraphOutlier style element in the current style.

SYMBOL= *symbol-name*

specifies the symbol for the markers. For a list of valid marker symbols, see [“Marker Symbols” on page 661](#).

Default: For ungrouped data, the default marker symbol is specified by the MarkerSymbol attribute of the GraphOutlier style element in the current style. For grouped data, the default marker symbol is specified by the MarkerSymbol attribute of the GraphData1 ... GraphData*n* style elements in the current style.

WEIGHT= NORMAL | BOLD

specifies the marker weight of the marker.

Default: NORMAL

Default: GraphOutlier is the default style element.

PERCENTILE= 1 | 2 | 3 | 4 | 5

specifies a method for computing the percentiles for the plot. For descriptions of each method, see “Calculating Percentiles” in the UNIVARIATE Procedure chapter of *Base SAS Procedures Guide: Statistical Procedures*.

Default: 5

SPREAD

relocates outlier points that have identical values to prevent overlapping.

Note: This option has no effect if your data does not contain two or more outliers with identical values for the analysis variable.

TRANSPARENCY= *value*

specifies the degree of transparency for the plot. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default: 0.0

WHISKERATTRS= *style-element* <(options)> | (options)

specifies the appearance of the whisker and cap lines. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see [“Color-Naming Schemes” in Chapter 14 of SAS/GRAPH: Reference](#).

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphBoxWhisker style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see [“Line Patterns” on page 663](#).

Default: For ungrouped data, the default line pattern is specified by the `LineStyle` attribute of the `GraphBoxWhisker` style element in the current style. For grouped data, the default line pattern is specified by the `LineStyle` attribute of the `GraphData1` ... `GraphDatan` style elements in the current style.

THICKNESS= *n* *<units>*

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see [“Units of Measurement” on page 659](#).

Default: For ungrouped data, the default line thickness is specified by the `LineThickness` attribute of the `GraphBoxWhisker` style element in the current style. For grouped data, the default line thickness is specified by the `LineThickness` attribute of the `GraphData1` ... `GraphDatan` style elements in the current style.

Default: `GraphBoxWhisker` is the default style element.

Interaction: This option is ignored if the `NOMEDIAN` option is also specified.

X2AXIS

assigns the analysis variable to the secondary (top) horizontal axis.

Y2AXIS

assigns the analysis variable to the secondary (right) vertical axis.

Details

Horizontal and vertical box plots display the distribution of data by using a rectangular box and whiskers. Whiskers are lines that indicate a data range outside of the box.

Note: Though the following figure shows a vertical box plot, the basic concepts apply to horizontal box plots.

Figure 7.2 Parts of a Box Plot

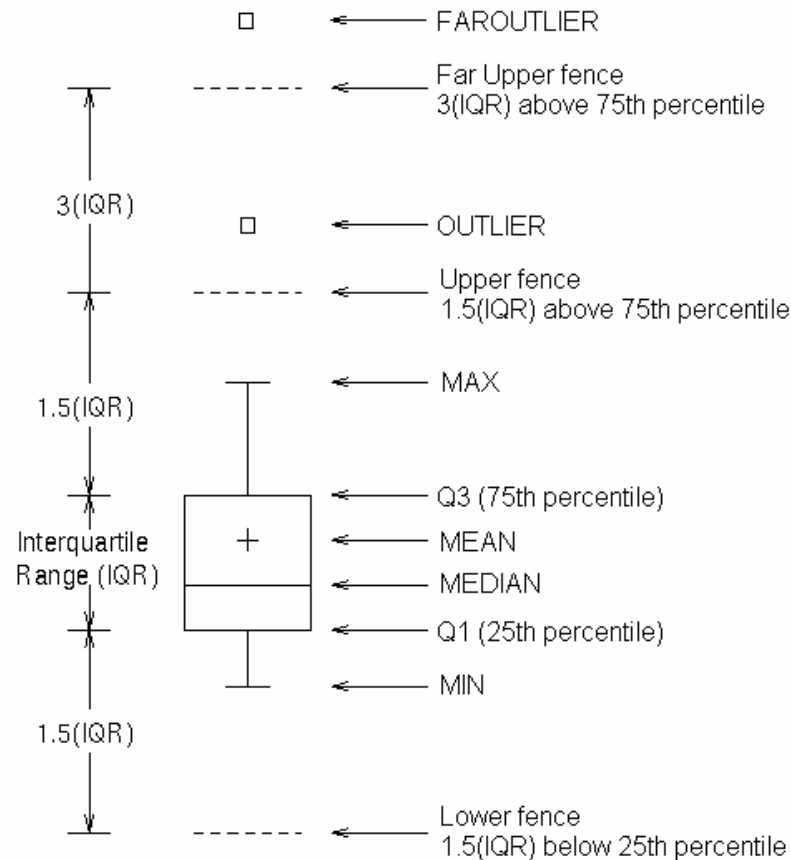


Figure 7.2 on page 335 shows a diagram of a vertical box plot. The bottom and top edges of the box indicate the intra-quartile range (IQR). That is, the range of values between the first and third quartiles (the 25th and 75th percentiles). The marker inside the box indicates the mean value. The line inside the box indicates the median value.

The elements that are outside the box are dependent on your options. By default, the whiskers that extend from each box indicate the range of values that are outside of the intra-quartile range. However, they are close enough not to be considered outliers (a distance less than or equal to $1.5 \times IQR$). If you specify the EXTREME option, then the whiskers indicate the entire range of values, including outliers.

Any points that are a distance of more than $1.5 \times IQR$ from the box are considered to be outliers. By default, these points are indicated by markers. If you specify `DATALABEL=` option, then the outlier points have data labels. If you also specify the `LABELFAR` option, then only outliers that are $3 \times IQR$ from the box have data labels.

HIGHLOW Statement

Creates a display of floating vertical or horizontal lines or bars that represent high and low values. The statement also gives you the option to display open and close values as tick marks and to specify a variety of plot attributes.

Note: This plot does not summarize data. If multiple observations have the same X or Y value, the observations are all plotted separately based on their values.

Examples: [“About High-Low Charts” on page 22](#)

[“Example 11: Creating a High-Low Chart” on page 510](#)

Syntax

HIGHLOW X= *variable* | Y= *variable*

HIGH= *numeric-variable* LOW= *numeric-variable*

</option(s)>;

Summary of Optional Arguments

Group options

CLUSTERWIDTH= *numeric-value*

specifies the cluster width as a ratio of the maximum width.

GROUP= *variable*

specifies a variable that is used to group the data.

GROUPDISPLAY= OVERLAY | CLUSTER

specifies how to display grouped data.

GROUPORDER= ASCENDING | DESCENDING | DATA

specifies the ordering of lines or bars within a group.

NOMISSINGGROUP

specifies that missing values of the group variable are not included in the plot.

HIGHLOW options

BARWIDTH= *numeric-value*

specifies the width of the bars as a ratio of the maximum possible width.

CLOSE= *numeric-variable*

specifies the data for the CLOSE tick on the bar or line.

DISCRETEOFFSET= *numeric-value*

specifies an amount to offset all lines or bars from the category midpoints.

FILL | NOFILL

specifies whether the area fill is visible for bars.

FILLATTRS= *style-element* | (COLOR= *color*)

specifies the appearance of the area fill for the bar.

HIGHCAP= *character-variable* | option | NONE

specifies the type of arrowhead used at the high end of the bar or line.

HIGHLABEL= *variable*

specifies the label to be shown at the high end of the line or bar.

INTERVALBARWIDTH= *numeric-value*

specifies the thickness of the bar when the X (or Y) data is numeric.

LABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the HIGHLABEL= option, the LOWLABEL= option, or both options.

LINEATTRS= *style-element* <(options)> | (options)

specifies the appearance of the outlines for the band.

LOWCAP= *character-variable* | option | NONE

specifies the type of arrowhead used at the low end of the bar or line.

LOWLABEL= *variable*

specifies the label to be shown at the low end of the line or bar.

OPEN= *numeric-variable*

specifies the data for the OPEN tick on the bar or line.

OUTLINE | NOOUTLINE

specifies whether the outlines of the bars are visible.

TYPE= BAR | LINE

specifies how the data is to be represented.

Plot options

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set.

LEGENDLABEL= "*text-string*"

specifies a label that identifies the elements from the band plot in the legend.

NAME= "*text-string*"

specifies a name for the plot.

TRANSPARENCY= *value*

specifies the degree of transparency for the plot.

URL= *character-variable*

specifies a character variable that contains URLs for Web pages to be displayed when parts of the plot are selected within an HTML page.

Required Arguments

X= *variable* | Y= *variable*

specifies a variable that is used to plot the values along the N or Y axis.

Note: If you specify X=*variable*, then the statement creates vertical lines or bars on the X axis, with the HIGH and LOW values plotted along the Y axis. Conversely, if you specify Y=*variable*, then the statement creates horizontal lines or bars on the Y axis, with the HIGH and LOW values plotted along the X axis.

HIGH= *numeric-variable*

specifies the upper value for the floating lines or bars.

LOW= *numeric-variable*

specifies the lower value for the floating lines or bars.

Optional Arguments

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph. For more information, see [“Using SG Attribute Maps to Control Visual Attributes” on page 605](#).

BARWIDTH= *numeric-value*

specifies the width of the bars as a ratio of the maximum possible width. The maximum width is equal to the distance between the center of each bar and the centers of the adjacent bars. Specify a value from 0.0 (narrowest) to 1.0 (widest).

For example, if you specify a width of 1, then there is no distance between the bars. If you specify a width of .5, then the width of the bars is equal to the space between the bars.

Default: 0.85

Requirement: This option is applicable only when the X or Y axis is discrete.

Interaction: This option has no effect unless TYPE= BAR.

CLOSE= *numeric-variable*

specifies the data for the CLOSE tick on the bar or line. For a vertical plot, the tick value is represented by an indicator on the side of the bar or line that has higher X values. For a horizontal plot, the value is represented by an indicator on the side with higher Y values.

CLUSTERWIDTH= *numeric-value*

specifies the cluster width as a ratio of the maximum width. Specify a value from 0.0 (narrowest) to 1.0 (widest).

Requirement: This option is applicable only when the X or Y axis is discrete.

Interactions:

This option is applicable only when TYPE= BAR.

This option is applicable only when the GROUP option is specified and GROUPDISPLAY= CLUSTER.

DISCRETEOFFSET= *numeric-value*

specifies an amount to offset all lines or bars from the category midpoints. Specify a value from -0.5 (left offset) to +0.5 (right offset).

Default: 0.0 (no offset from the category midpoints)

Requirement: This option is applicable only when the X or Y axis is discrete.

FILL | NOFILL

specifies whether the area fill is visible for bars. The FILL option shows the area fill. The NOFILL option hides the area fill.

Default: The default status of the area fill is specified by the DisplayOpts attribute of the GraphBand style element in the current style.

Interactions:

This option has no effect unless TYPE= BAR.

Specifying FILL also hides any visible outlines.

If NOFILL and NOOUTLINE are both specified, then both options are ignored.

FILLATTRS= *style-element* | (COLOR= *color*)

specifies the appearance of the area fill for the bar. You can specify the color of the fill by using a style element or by using the COLOR= suboption. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphDataConfidence style element in the current style. For grouped data, the default color is specified by the Color attribute of the GraphData1... GraphData*n* style elements in the current style.

Interactions:

This option is applicable only when TYPE= BAR.

This option has no effect if you also specify the NOFILL option.

GROUP= *variable*

specifies a variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes.

GROUPDISPLAY= OVERLAY | CLUSTER

specifies how to display grouped data.

OVERLAY

groups are overlaid without any clustering. Each group is represented by unique visual attributes derived from the GraphData1... GraphData*n* style elements in the current style.

CLUSTER

observations with different group values are displayed in adjacent clusters around the category value. Each set of group values is centered at the midpoint tick mark for the category.

Default: OVERLAY

Restriction: GROUPDISPLAY=CLUSTER has no effect unless the X or Y axis is discrete.

Interactions:

GROUPDISPLAY=CLUSTER is applicable only when TYPE=BAR.

This option is ignored unless GROUP= is specified.

GROUPORDER= ASCENDING | DESCENDING | DATA

specifies the ordering of lines or bars within a group.

ASCENDING

displays each group of lines or bars in ascending group value.

DESCENDING

displays each group of lines or bars in descending group value.

DATA

displays each group of lines or bars in data order of the group column.

Interactions:


This option is ignored unless GROUP= is specified.

This option is applicable only when GROUPDISPLAY=CLUSTER and TYPE=BAR.

HIGHCAP= *character-variable* | *option* | NONE


specifies the type of arrowhead used at the high end of the bar or line. You can specify one of the following options, or you can specify a character variable that contains one of the following options:

SERIF

a small serif. 


The serif is supported for TYPE=LINE only.

BARBEDARROW

a solid triangle with an indent at the base. 


The barbed arrow is supported for TYPE=LINE only.

FILLEDARROW

a solid triangle. 


The filled arrow is supported for TYPE=BAR and TYPE=LINE.

OPENARROW

resembles the letter "V". 

The open arrow is supported for TYPE=LINE only.

CLOSEDARROW

an outline of a triangle. 

The closed arrow is supported for TYPE=BAR only.

Default: NONE

Restriction: Arrow heads are not displayed for very short bars. Bar height must be at least twice the size of the arrow heads in order for the arrow heads to appear.

HIGHLABEL= *variable*

specifies the label to be shown at the high end of the line or bar.

INTERVALBARWIDTH= *numeric-value*

specifies the thickness of the bar when the X (or Y) data is numeric.

Default: The default thickness of the bar is derived from the minimum interval between the data values along X or Y.

Requirement: This option is applicable only when the X or Y axis is a linear or a TIME axis.

Interaction: This option has no effect unless TYPE=BAR.

LABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the HIGHLABEL= option, the LOWLABEL= option, or both options. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the characters. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphDataText style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

FAMILY= “*font-family*”

specifies the font family for the marker characters. The SAS ODS styles use TrueType system fonts. For more information, see “TrueType Fonts Supplied by SAS” in Chapter 15 of *SAS Language Reference: Concepts*.

Default: The default font family is specified by the FontFamily attribute of the GraphDataText style element in the current style.

SIZE= *n* <*units*>

specifies the font size of the characters. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default font size is specified by the FontSize attribute of the GraphDataText style element in the current style.

STYLE= ITALIC | NORMAL

specifies whether the characters are italic (ITALIC) or normal (NORMAL).

Default: The default font style is specified by the FontStyle attribute of the GraphDataText style element in the current style.

WEIGHT= BOLD | NORMAL

specifies whether the characters are bold (BOLD) or normal (NORMAL).

Default: The default font weight is specified by the FontWeight attribute of the GraphDataText style element in the current style.

LEGENDLABEL= “*text-string*”

specifies a label that identifies the elements from the band plot in the legend. By default, the label “band” is used for ungrouped data, and the group values are used for grouped data.

Interaction: The LEGENDLABEL= option has no effect if you also specify the GROUP= option.

LINEATTRS= *style-element* <(options)> | (options)

specifies the appearance of the outlines for the band. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphDataDefault style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see “Line Patterns” on page 663.

Default: For ungrouped data, the default line pattern is specified by the LineStyle attribute of the GraphDataDefault style element in the current style. For grouped data, the default line pattern is specified by the LineStyle attribute of the GraphData1 ... GraphData*n* style elements in the current style.

THICKNESS= *n* <units>

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.


Default: For ungrouped data, the default line thickness is specified by the LineThickness attribute of the GraphDataDefault style element in the current style. For grouped data, the default line thickness is specified by the LineThickness attribute of the GraphData1 ... GraphData*n* style elements in the current style.

Interaction: This option has no effect unless you also specify the OUTLINES option.

LOWCAP= *character-variable* | *option* | NONE


specifies the type of arrowhead used at the low end of the bar or line. You can specify one of the following options, or you can specify a character variable that contains one of the following options:

SERIF

a small serif. 


The serif is supported for TYPE=LINE only.

BARBEDARROW

a solid triangle with an indent at the base. 


The barbed arrow is supported for TYPE=LINE only.

FILLEDARROW

a solid triangle. 


The filled arrow is supported for TYPE=BAR and TYPE=LINE.

OPENARROW

resembles the letter "V". 

The open arrow is supported for TYPE=LINE only.

CLOSEDARROW

an outline of a triangle. 

The closed arrow is supported for TYPE=BAR only.

Default: NONE

Restriction: Arrow heads are not displayed for very short bars. Bar height must be at least twice the size of the arrow heads in order for the arrow heads to appear.

LOWLABEL= *variable*

specifies the label to be shown at the low end of the line or bar.

NOMISSINGGROUP

specifies that missing values of the group variable are not included in the plot.

Interaction: This option has no effect unless GROUP= is also specified.

NAME= "*text-string*"

specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note: The name is case-sensitive. When you refer to the specified name in other statements, you must use the same exact capitalization.

OPEN= *numeric-variable*

specifies the data for the OPEN tick on the bar or line.

For a vertical plot, the tick value is represented by an indicator on the side of the bar or line that has lower X values. For a horizontal plot, the value is represented by an indicator on the side with lower Y values.

OUTLINE | NOOUTLINE

specifies whether the outlines of the bars are visible. The OUTLINE option shows the outlines. The NOOUTLINE option hides the outlines.

Default: OUTLINE

Interactions:

This option has no effect unless TYPE= BAR.

Specifying OUTLINE also hides the fill color.

If NOOUTLINE and NOFILL are both specified, then both options are ignored.

TRANSPARENCY= *value*

specifies the degree of transparency for the plot. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default: 0.0

TYPE= BAR | LINE

specifies how the data is to be represented. BAR uses fill and outline attributes. LINE uses line attributes.

Default: LINE

URL= *character-variable*

specifies a character variable that contains URLs for Web pages to be displayed when parts of the plot are selected within an HTML page.

Default: By default, no HTML links are created.

Interactions:

This option affects graphics output that is created through the ODS HTML destination only. For more information about ODS destinations, see [“Specifying a Destination” on page 649](#).

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For more information, see [“Using the ODS GRAPHICS Statement” on page 651](#).

HISTOGRAM Statement

Creates a histogram that displays the frequency distribution of a numeric variable.

Interaction: The HISTOGRAM statement can be combined only with DENSITY statements in the SGPlot procedure.

Note: The range of the response variable is automatically divided into an appropriate number of bins.

Examples: [“About Histograms” on page 40](#)
[“Example 8: Combining Histograms with Density Plots” on page 507](#)

Syntax

HISTOGRAM *response-variable* *</ option(s)>*;

Summary of Optional Arguments

Histogram options

BINSTART= *numeric-value*

specifies the X coordinate of the first bin.

BINWIDTH= *numeric-value*

specifies the bin width.

BOUNDARY= LOWER | UPPER

specifies how boundary values are assigned to bins.

FILL | NOFILL

specifies whether the area fill is visible.

FILLATTRS= *style-element* | (COLOR= *color*)

specifies the appearance of the area fill.

NBINS= *numeric-value*

specifies the number of bins.

OUTLINE | NOOUTLINE

specifies whether outlines are displayed for the bars.

SCALE= COUNT | PERCENT | PROPORTION

specifies the scaling that is applied to the vertical axis.

SHOWBINS

specifies that the midpoints of the value bins are used to create the tick marks for the horizontal axis.

Plot options

FREQ= *numeric-variable*

specifies how many times observations are repeated for computational purposes.

LEGENDLABEL= *“text-string”*

specifies a label that identifies the histogram in the legend.

NAME= *“text-string”*

specifies a name for the plot.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the histogram.

X2AXIS

assigns the response variable to the secondary (top) horizontal axis.

Y2AXIS

assigns the calculated values to the secondary (right) vertical axis.

Required Argument

response-variable

specifies the response variable for the histogram.

Optional Arguments

BINSTART= *numeric-value*

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: The default value is determined by the system.

BINWIDTH= *numeric-value*

specifies the bin width. The system determines the number of bins. The bins always span the range of the data.

Default: The default value is determined by the system.

Interactions:

This option is ignored if the **NBINS=** option is also specified.

BOUNDARY= LOWER | UPPER

specifies how boundary values are assigned to bins.

LOWER

specifies that boundary values are assigned to the lower bin.

UPPER

specifies that boundary values are assigned to the upper bin.

Default: UPPER

FILL | NOFILL

specifies whether the area fill is visible. The **FILL** option shows the area fill. The **NOFILL** option hides the area fill.

Default: The default status of the area fill is specified by the **DisplayOpts** attribute of the **GraphHistogram** style element in the current style.

Interactions:

Specifying **FILL** also hides any visible outlines.

If **NOFILL** and **NOOUTLINE** are both specified, then both options are ignored.

FILLATTRS= *style-element* | (**COLOR=** *color*)

specifies the appearance of the area fill. You can specify the color of the fill by using a style element or by using the **COLOR=** suboption. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more

information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: The default color is specified by the Color attribute of the GraphDataDefault style element in the current style.

Interaction: This option has no effect if you specify the NOFILL option.

FREQ= *numeric-variable*

specifies how many times observations are repeated for computational purposes. Each observation is repeated n times for computational purposes, where n is the value of the numeric variable. If n is not an integer, then it is truncated to an integer. If n is less than 1 or missing, then it is excluded from the analysis.

LEGENDLABEL= “*text-string*”

specifies a label that identifies the histogram in the legend. By default, the label of the response variable is used.

NAME= “*text-string*”

specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note: The name is case-sensitive. When you refer to the specified name in other statements, you must use the same exact capitalization.

NBINS= *numeric-value*

specifies the number of bins. The system determines the BINWIDTH= value. The bins always span the range of the data.

The procedure attempts to produce tick values that are easily interpreted (for example, 5, 10, 15, 20). The procedure sometimes adjusts the location of the first bin and the bin width accordingly. As a result, the number of bins shown in the plot might not exactly match the number specified with NBINS=.

Default: The default number of bins is determined by the system.

OUTLINE | NOOUTLINE

specifies whether outlines are displayed for the bars. The OUTLINE option shows the outlines. The NOOUTLINE option hides the outlines.

Default: The default status of the outlines is specified by the DisplayOpts attribute of the GraphHistogram style element in the current style.

Interactions:

Specifying OUTLINE also hides the fill color.

If NOOUTLINE and NOFILL are both specified, then both options are ignored.

SCALE= COUNT | PERCENT | PROPORTION

specifies the scaling that is applied to the vertical axis. Specify one of the following values:

COUNT

the axis displays the frequency count.

PERCENT

the axis displays values as a percentage of the total.

PROPORTION

the axis displays values as proportions (0.0 to 1.0) of the total.

Default: PERCENT

SHOWBINS

specifies that the midpoints of the value bins are used to create the tick marks for the horizontal axis. By default, the tick marks are created at regular intervals based on the minimum and maximum values.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the histogram. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default: 0.0

X2AXIS

assigns the response variable to the secondary (top) horizontal axis.

Y2AXIS

assigns the calculated values to the secondary (right) vertical axis.

HLINE Statement

Creates a horizontal line chart. You can use the HLINE statement with the HBAR statement to create a horizontal bar-line chart.

Interaction: The HLINE statement can be combined only with other categorization plot statements in the SGLOT procedure. See [“Plot Type Compatibility” on page 277](#).

Example: [“About Line Charts” on page 47](#)

Syntax

HLINE *category-variable* < / option(s) >;

Summary of Optional Arguments

Group options

CLUSTERWIDTH= *numeric-value*

specifies the width of the group clusters as a fraction of the midpoint spacing.

GROUP= *variable*

specifies a variable that is used to group the data.

GROUPDISPLAY= CLUSTER | OVERLAY

specifies how to display grouped lines.

GROUPORDER= ASCENDING | DESCENDING | DATA

specifies the ordering of lines within a group.

Label options

CURVELABEL <=“*text-string*”>

adds a label for the line.

CURVELABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the CURVELABEL= option.

CURVELABELLOC= OUTSIDE | INSIDE

specifies whether the curve label is placed inside the plot axes (INSIDE) or outside of the plot axes (OUTSIDE).

CURVELABELPOS= AUTO | END | MAX | MIN | START

specifies the location of the curve label.

DATALABEL <= *variable*>

displays a label for each data point.

DATALABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the DATALABEL= option.

Line options

BREAK

creates a break in the line for each missing value of the response variable.

DISCRETEOFFSET= *numeric-value*

specifies an amount to offset all lines from the discrete category values.

LIMITATTRS= *style-element* <(options)> | (options)

specifies the appearance of the limit lines in the plot.

LIMITS= BOTH | LOWER | UPPER

adds limit lines to the plot.

LIMITSTAT= CLM | STDDEV | STDERR

specifies the statistic for the limit lines.

LINEATTRS= *style-element* <(options)> | (options)

specifies the appearance of the lines in the line plot.

MARKERATTRS= *style-element* <(options)> | (options)

specifies the appearance of the markers in the plot.

MARKERS

adds markers to the plot.

NOSTATLABEL

removes the statistic name from the axis and legend labels.

NUMSTD= *n*

specifies the number of standard units for the limit lines, when you specify LIMITSTAT= STDDEV or LIMITSTAT= STDERR.

RESPONSE= *response-variable*

specifies a numeric response variable for the plot.

STAT= FREQ | MEAN | SUM

specifies the statistic for the horizontal axis.

Plot options

ALPHA= *numeric-value*

specifies the confidence level for the confidence limits.

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set.

CATEGORYORDER= RESPASC | RESPDESC

specifies the order in which the response values are arranged.

FREQ= *numeric-variable*

specifies how many times observations are repeated for computational purposes.

LEGENDLABEL= "*text-string*"

specifies the label that identifies the line plot in the legend.

MISSING

processes missing values as a valid category value and creates a line for it.

NAME= "*text-string*"

specifies a name for the plot.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the lines and markers.

URL= *character-variable*

specifies a character variable that contains URLs for Web pages to be displayed when parts of the plot are selected within an HTML page.

WEIGHT= *numeric-variable*

specifies how observations are weighted.

X2AXIS

assigns the response variable to the secondary (top) horizontal axis.

Y2AXIS

assigns the category variable to the secondary (right) vertical axis.

Required Argument

category-variable

specifies the variable that classifies the observations into distinct subsets.

Optional Arguments

ALPHA= *numeric-value*

specifies the confidence level for the confidence limits. Specify a number between 0.00 (100% confidence) and 1.00 (0% confidence).

Default: .05

Interactions:

This option has no effect if you do not specify LIMITSTAT=CLM.

If your plot is overlaid with other categorization plots, then the first ALPHA value that you specify is used for all of the plots.

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph. For more information, see [“Using SG Attribute Maps to Control Visual Attributes” on page 605](#).

BREAK

creates a break in the line for each missing value of the response variable.

Notes:

The observation is excluded from the graph when there is a missing value for the category variable without the specification of the MISSING option. (No break occurs in the line.)

The observation is excluded from the graph when there is a missing value for the FREQ variable.

CATEGORYORDER= RESPASC | RESPDESC

specifies the order in which the response values are arranged. Specify one of the following values:

RESPASC

sorts the response values in ascending order.

RESPDESC

sorts the response values in descending order.

Default: By default, the plot is sorted in ascending order based on the category values.

Restrictions:

This option takes effect only when the plot statement specifies a response variable and the axis for that variable is numeric. If the axis is not numeric, an error is generated and a message is written to the SAS log.

Uniform scaling and response sorting cannot occur on the same axis. If the UNIFORM= option is used in the SGPLOT statement, the UNIFORM option is ignored for the sorted response axis and a note is generated in the log. The

UNIFORM= option is applied to the other axes and groups if requested. Note that the UNIFORM= option can selectively apply scaling to only the X or Y axis

Interaction: If a group variable is specified in the plot statement, the response sort option is ignored and a note is generated in the log. (You can use a BY statement in the procedure if you want to group data.)

Note: If CATEGORYORDER is specified in multiple statements, the procedure sorts by the last statement in which it is specified.

CLUSTERWIDTH= *numeric-value*

specifies the width of the group clusters as a fraction of the midpoint spacing. Specify a value from 0.0 (narrowest) to 1.0 (widest).

Default: 0.8

Interaction: This option is applicable only when the GROUP option is specified, when GROUPDISPLAY=CLUSTER, and when the category axis is discrete.

CURVELABEL <=“*text-string*”>

adds a label for the line. You can also specify the label text. If you do not specify a label, then the label from the response variable is used.

CURVELABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the CURVELABEL= option. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

Here is an example that specifies a style element:

CURVELABELATTRS=GraphTitleText

options can be one or more of the following:

COLOR= *color*

specifies the color of the label characters. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphValueText style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

FAMILY= “*font-family*”

specifies the font family for the label characters. The SAS ODS styles use TrueType system fonts. For more information, see “TrueType Fonts Supplied by SAS” in Chapter 15 of *SAS Language Reference: Concepts*.

Default: The default font family is specified by the FontFamily attribute of the GraphValueText style element in the current style.

Restriction: You cannot specify a list of font family names.

SIZE= *n* <*units*>

specifies the font size of the label characters. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default font size is specified by the FontSize attribute of the GraphValueText style element in the current style.

STYLE= ITALIC | NORMAL

specifies whether the label characters are italic or normal.

Default: The default font style is specified by the `FontStyle` attribute of the `GraphValueText` style element in the current style.

WEIGHT= BOLD | NORMAL

specifies whether the label characters are bold or normal.

Default: The default font weight is specified by the `FontWeight` attribute of the `GraphValueText` style element in the current style.

Here is an example expression:

```
CURVELABELATTRS=(Color=Green Family=Arial Size=8 Style=Italic Weight=Bold)
```

Default: The default style element is `GraphValueText`.

Interaction: This option has no effect unless the `CURVELABEL` option is also specified.

CURVELABELLOC= OUTSIDE | INSIDE

specifies whether the curve label is placed inside the plot axes (`INSIDE`) or outside of the plot axes (`OUTSIDE`).

Default: `INSIDE`

CURVELABELPOS= AUTO | END | MAX | MIN | START

specifies the location of the curve label. Specify one of the following values:

AUTO

places the curve label outside the plot area near the end of the curve along unused axes whenever possible (typically `Y2` or `X2`).

Interaction: This value takes effect only when `CURVELABELLOC=OUTSIDE`.

END

places the curve label at the last point on the curve.

MAX

places the label at the part of the curve closest to the maximum X axis value.

MIN

places the label at the part of the curve closest to the minimum X axis value.

START

places the curve label at the first point on the curve.

Default: `END`

Interactions:

This option has no effect unless the `CURVELABEL` option is also specified.

The `START` and `END` suboptions take effect only when `CURVELABELLOC=INSIDE`.

DATALABEL <= variable>

displays a label for each data point. If you specify a variable, then the values of that variable are used for the data labels. If you do not specify a variable, then the values of the calculated response are used for the data labels.

DATALABELATTRS= style-element <(options)> | (options)

specifies the appearance of the labels in the plot when you use the `DATALABEL=` option. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

Here is an example that specifies a style element:

```
DATALABELATTRS=GraphLabelText
```

options can be one or more of the following:

COLOR= *color*

specifies the color of the label characters. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphDataText style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

FAMILY= “*font-family*”

specifies the font family for the label characters. The SAS ODS styles use TrueType system fonts. For more information, see “TrueType Fonts Supplied by SAS” in Chapter 15 of *SAS Language Reference: Concepts*.

Default: The default font family is specified by the FontFamily attribute of the GraphDataText style element in the current style.

SIZE= *n* <*units*>

specifies the font size of the label characters. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default font size is specified by the FontSize attribute of the GraphDataText style element in the current style.

STYLE= ITALIC | NORMAL

specifies whether the label characters are italic (ITALIC) or normal (NORMAL).

Default: The default font style is specified by the FontStyle attribute of the GraphDataText style element in the current style.

WEIGHT= BOLD | NORMAL

specifies whether the label characters are bold (BOLD) or normal (NORMAL).

Default: The default font weight is specified by the FontWeight attribute of the GraphDataText style element in the current style.

Here is an example expression:

```
DATALABELATTRS=(Color=Green Family=Arial Size=8 Style=Italic Weight=Bold)
```

Default: The default style element is GraphDataText.

Interaction: This option has no effect unless the DATALABEL option is also specified.

DISCRETEOFFSET= *numeric-value*

specifies an amount to offset all lines from the discrete category values. Specify a value from -0.5 (left offset) to +0.5 (right offset).

Default: 0.0 (no offset)

Requirement: This option is applicable only when the category axis is discrete.

FREQ= *numeric-variable*

specifies how many times observations are repeated for computational purposes. specifies that each observation is repeated *n* times for computational purposes, where *n* is the value of the numeric variable. If *n* is not an integer, then it is truncated to an integer. If *n* is less than 1 or missing, then it is excluded from the analysis.

Interaction: If your plot is overlaid with other categorization plots, then the first FREQ variable that you specified is used for all of the plots.

GROUP= *variable*

specifies a variable that is used to group the data. A separate plot is created for each unique value of the grouping variable. The plot elements for each group value are automatically distinguished by different visual attributes.

Interaction: If you specify more than one categorization plot statement, then all of the plots must specify the same GROUP variable. If you do not specify the same GROUP= option for all of the categorization plots, then the GROUP= option has no effect.

GROUPDISPLAY= CLUSTER | OVERLAY

specifies how to display grouped lines.

CLUSTER

grouped items are drawn adjacent to each other.

OVERLAY

grouped items are drawn at the exact coordinate and might overlap. Each group is represented by unique visual attributes derived from the GraphData1... GraphData*n* style elements in the current style.

Default: OVERLAY

Restriction: GROUPDISPLAY=CLUSTER can take effect only when the category axis is discrete. If necessary, use a TYPE=DISCRETE option on the axis statement.

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

GROUPORDER= ASCENDING | DESCENDING | DATA

specifies the ordering of lines within a group.

ASCENDING

displays each group in ascending group value.

DESCENDING

displays each group in descending group value.

DATA

displays each group in data order of the group variable.

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

LEGENDLABEL= “*text-string*”

specifies the label that identifies the line plot in the legend. By default, the label of the response variable is used. If there is no response variable label, then the name of the response variable and the computed statistic (SUM or MEAN) are used. If you do not specify a response variable, then the legend label is “Frequency”.

Interaction: The LEGENDLABEL= option has no effect if you also specify the GROUP= option in the same plot statement.

LIMITATTRS= *style-element* <(options)> | (options)

specifies the appearance of the limit lines in the plot. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: The default color is specified by the ContrastColor attribute of the GraphError style element in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see [“Line Patterns” on page 663](#).

Default: The default line pattern is specified by the LineStyle attribute of the GraphError style element in the current style.

THICKNESS= *n* *<units>*

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see [“Units of Measurement” on page 659](#).

Default: The default line thickness is specified by the LineThickness attribute of the GraphError style element in the current style.

LIMITS= BOTH | LOWER | UPPER

adds limit lines to the plot. Specify one of the following values:

BOTH

adds lower and upper limit lines to the plot.

LOWER

adds lower limit lines to the plot.

UPPER

adds upper limit lines to the plot.

By default, no limit lines are displayed. However, if you specify the LIMITSTAT= option, then the default is BOTH.

Interaction: Limit lines are displayed only when you specify STAT= MEAN.

LIMITSTAT= CLM | STDDEV | STDERR

specifies the statistic for the limit lines. Specify one of the following statistics:

CLM

confidence limits

STDDEV

standard deviation

STDERR

standard error

Default: CLM

Interaction: If you specify the LIMITSTAT= option, then the default value for the LIMITS= option is BOTH.

LINEATTRS= *style-element* *<(options)>* | *(options)*

specifies the appearance of the lines in the line plot. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see [“Color-Naming Schemes” in Chapter 14 of SAS/GRAPH: Reference](#).

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphDataDefault style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see [“Line Patterns” on page 663](#).

Default: For ungrouped data, the default line pattern is specified by the `LineStyle` attribute of the `GraphDataDefault` style element in the current style. For grouped data, the default line pattern is specified by the `LineStyle` attribute of the `GraphData1 ... GraphDatan` style elements in the current style.

THICKNESS= *n <units>*

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see [“Units of Measurement” on page 659](#).

Default: For ungrouped data, the default line thickness is specified by the `LineThickness` attribute of the `GraphDataDefault` style element in the current style. For grouped data, the default line thickness is specified by the `LineThickness` attribute of the `GraphData1 ... GraphDatan` style elements in the current style.

MARKERATTRS= *style-element <(options)> | (options)*

specifies the appearance of the markers in the plot. This option has no effect unless you also specify the `MARKERS` option. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the markers. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see [“Color-Naming Schemes” in Chapter 14 of SAS/GRAPH: Reference](#).

Default: For ungrouped data, the default color is specified by the `ContrastColor` attribute of the `GraphDataDefault` style element in the current style. For grouped data, the default color is specified by the `ContrastColor` attribute of the `GraphData1 ... GraphDatan` style elements in the current style.

SIZE= *n <units>*

specifies the size of the markers. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see [“Units of Measurement” on page 659](#).

Default: The default marker size is specified by the `MarkerSize` attribute of the `GraphDataDefault` style element in the current style.

SYMBOL= *symbol-name*

specifies the symbol for the markers. For a list of valid marker symbols, see [“Marker Symbols” on page 661](#).

Default: For ungrouped data, the default marker symbol is specified by the `MarkerSymbol` attribute of the `GraphDataDefault` style element in the current style. For grouped data, the default marker symbol is specified by the `MarkerSymbol` attribute of the `GraphData1 ... GraphDatan` style elements in the current style.

MARKERS

adds markers to the plot.

MISSING

processes missing values as a valid category value and creates a line for it.

NAME= *“text-string”*

specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note: The name is case-sensitive. When you refer to the specified name in other statements, you must use the same exact capitalization.

NOSTATLABEL

removes the statistic name from the axis and legend labels.

NUMSTD= *n*

specifies the number of standard units for the limit lines, when you specify LIMITSTAT= STDDEV or LIMITSTAT= STDERR. You can specify any positive number, including decimals.

Default: 1

RESPONSE= *response-variable*

specifies a numeric response variable for the plot. The summarized values of the response variable are displayed for each value on the horizontal axis.

STAT= **FREQ | **MEAN** | **SUM****

specifies the statistic for the horizontal axis. Specify one of the following statistics:

FREQ

the frequencies for the category variable

MEAN

the mean of the response variable

SUM

the sum of the response variable

Default: If you do not specify the RESPONSE= option, then the FREQ statistic is the default. If you specify RESPONSE=, then the SUM statistic is the default.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the lines and markers. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default: 0.0

URL= *character-variable*

specifies a character variable that contains URLs for Web pages to be displayed when parts of the plot are selected within an HTML page.

Default: By default, no HTML links are created.

Interactions:

This option affects graphics output that is created through the ODS HTML destination only. For more information about ODS destinations, see [“Specifying a Destination” on page 649](#).

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For more information, see [“Using the ODS GRAPHICS Statement” on page 651](#).

WEIGHT= *numeric-variable*

specifies how observations are weighted. Each observation is weighted by a factor of w for computational purposes, where w is the value of the numeric variable. w can be any numeric value. If w is 0, negative or missing, then that observation is excluded from the analysis.

Interaction: If your plot is overlaid with other categorization plots, then the first WEIGHT variable that you specified is used for all of the plots.

X2AXIS

assigns the response variable to the secondary (top) horizontal axis.

Y2AXIS

assigns the category variable to the secondary (right) vertical axis.

INSET Statement

Adds a text box inside of the axes of the plot.

Example: [“About Text Insets” on page 31](#)

Syntax

INSET *“text-string-1”* <... *“text-string-n”*></ *option(s)*>;

INSET (*label-list*) </ *option(s)*>;

Summary of Optional Arguments**Inset options**

BORDER | **NOBORDER**

specifies whether to display a border around the text box.

LABELALIGN= **LEFT** | **CENTER** | **RIGHT**

specifies how the labels are aligned when you specify label-value pairs.

POSITION= *position-value*

specifies the position of the text box within the plot.

TEXTATTRS= *style-element* <(options)> | (options)

specifies the appearance of the text in the text box.

VALUEALIGN= **LEFT** | **CENTER** | **RIGHT**

specifies how the values are aligned when you specify label-value pairs.

Title options

TITLE= *“text-string”*

specifies a title for the text box.

TITLEATTRS= *style-element* <(options)> | (options)

specifies the appearance of the title.

Required Argument

text-string-list | (*label-list*)

you must specify one of the following arguments:

text-string-list

specifies one or more quoted text strings. Each string is placed on a separate line in the text box (for example, **“My line 1” “My line 2”**).

(*label-list*)

specifies a series of quoted labels and values for the text box.

Specify your label-value pairs as follows:

(*“label-1”* = *“value-1”* ... *“label-n”* = *“value-n”*)

Here is an example: ("My label 1" = "My value 1" "My label 2"
= "My value 2").

You can format the text in insets and add special characters using the following syntax:

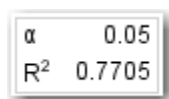
(*ESC*) {sup 'string'}	Specifies that the string appears as a superscript.
(*ESC*) {sub 'string'}	Specifies that the string appears as a subscript.
(*ESC*) {unicode 'hexadecimal-value'x } keyword }	Specifies a glyph (graphical character) to be displayed using its Unicode hexadecimal value or a SAS keyword equivalent. For more information about SAS keywords, see Appendix 1, "SAS Keywords for Unicode Glyphs," in <i>SAS Graph Template Language: User's Guide</i> .

Note: (*ESC*) is the default ODS escape character. You can set your own escape character with the ODS **ESCAPECHAR="character"**; statement.

In the following inset statement, the Greek letter alpha denotes the alpha value for the confidence limits in a graph. The R-Square value is displayed using the superscript format:

```
inset ("(*ESC*){unicode alpha}" = "0.05"  
      "R(*ESC*){sup '2'}" = "0.7705") / border;
```

Here is the text inset for this example:



Optional Arguments

BORDER | NOBORDER

specifies whether to display a border around the text box. The BORDER option displays the border. The NOBORDER option hides the border.

LABELALIGN= LEFT | CENTER | RIGHT

specifies how the labels are aligned when you specify label-value pairs. Specify one of the following:

LEFT

aligns the text to the left.

CENTER

aligns the text to the center.

RIGHT

aligns the text to the right.

Default: LEFT

POSITION= position-value

specifies the position of the text box within the plot. The position values are as follows:

BOTTOM

places the text box at the bottom of the graph.

BOTTOMLEFT

places the text box at the bottom left corner of the graph.

BOTTOMRIGHT

places the text box at the bottom right corner of the graph.

LEFT

places the text box at the left side of the graph.

RIGHT

places the text box at the right side of the graph.

TOP

places the text box at the top of the graph.

TOPLEFT

places the text box at the top left corner of the graph.

TOPRIGHT

places the text box at the top right corner of the graph.

If you do not specify a position, then a position is determined automatically.

TEXTATTRS= *style-element* <(options)> | (options)

specifies the appearance of the text in the text box. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

Here is an example that specifies a style element: **TEXTATTRS=GraphDataText**

options can be one or more of the following:

COLOR= *color*

specifies the color of the label characters. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphValueText style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

FAMILY= “*font-family*”

specifies the font family for the label characters. The SAS ODS styles use TrueType system fonts. For more information, see “TrueType Fonts Supplied by SAS” in Chapter 15 of *SAS Language Reference: Concepts*.

Default: The default font family is specified by the FontFamily attribute of the GraphValueText style element in the current style.

Restriction: You cannot specify a list of font family names.

SIZE= *n* <units>

specifies the font size of the label characters. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default font size is specified by the FontSize attribute of the GraphValueText style element in the current style.

STYLE= ITALIC | NORMAL

specifies whether the label characters are italic or normal.

Default: The default font style is specified by the FontStyle attribute of the GraphValueText style element in the current style.

WEIGHT= BOLD | NORMAL

specifies whether the label characters are bold or normal.

Default: The default font weight is specified by the FontWeight attribute of the GraphValueText style element in the current style.

Here is an example expression:

```
TEXTATTRS=(Color=Green Family=Arial Size=8 Style=Italic Weight=Bold)
```

Default: The default appearance of the text is specified by the GraphValueText style element in the current style.

Tip: The GraphDataText style element makes the text slightly smaller. The GraphLabelText style element makes the text slightly larger.

TITLE= *“text-string”*

specifies a title for the text box. The title text is always center-aligned.

TITLEATTRS= *style-element* <(options)> | (options)

specifies the appearance of the title. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

Here is an example that specifies a style element:

```
TITLEATTRS=GraphTitleText
```

options can be one or more of the following:

COLOR= *color*

specifies the color of the title characters. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphLabelText style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

FAMILY= *“font-family”*

specifies the font family for the title characters. The SAS ODS styles use TrueType system fonts. For more information, see “TrueType Fonts Supplied by SAS” in Chapter 15 of *SAS Language Reference: Concepts*.

Default: The default font family is specified by the FontFamily attribute of the GraphLabelText style element in the current style.

Restriction: You cannot specify a list of font family names.

SIZE= *n* <*units*>

specifies the font size of the title characters. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default font size is specified by the FontSize attribute of the GraphLabelText style element in the current style.

STYLE= ITALIC | NORMAL

specifies whether the title characters are italic or normal.

Default: The default font style is specified by the FontStyle attribute of the GraphLabelText style element in the current style.

WEIGHT= BOLD | NORMAL

specifies whether the title characters are bold or normal.

Default: The default font weight is specified by the `FontWeight` attribute of the `GraphLabelText` style element in the current style.

Here is an example expression:

```
TITLEATTRS=(Color=Green Family=Arial Size=8 Style=Italic Weight=Bold)
```

Default: The default appearance of the title text is specified by the `GraphLabelText` style element in the current style.

VALUEALIGN= LEFT | CENTER | RIGHT

specifies how the values are aligned when you specify label-value pairs. Specify one of the following values.

LEFT

aligns the text to the left.

CENTER

aligns the text to the center.

RIGHT

aligns the text to the right.

Default: RIGHT

KEYLEGEND Statement

Adds a legend to the plot.

Examples: [“Example 5: Adding a Prediction Ellipse to a Scatter Plot” on page 503](#)
[“Example 6: Creating Lines and Bands from Pre-Computed Data” on page 504](#)
[“Example 8: Combining Histograms with Density Plots” on page 507](#)

Syntax

```
KEYLEGEND <“name-1” ... “name-n”> </ option(s)>;
```

Summary of Optional Arguments

Legend options

ACROSS= *n*

specifies the number of columns in the legend.

BORDER | NOBORDER

specifies whether the border around the legend is visible.

DOWN= *n*

specifies the number of rows in the legend.

LOCATION= OUTSIDE | INSIDE

specifies whether the legend is placed outside or inside of the axis area.

POSITION= *position-value*

specifies the position of the legend within the graph.

VALUEATTRS= *style-element* <(options)> | (options)

specifies the appearance of the legend value labels.

“name-1” ... “name-n”

specifies the names of one or more plots that you want to include in legend.

Title options

TITLE= *“text-string”*

specifies a title for the legend.

TITLEATTRS= *style-element* <*options*> | (*options*)

specifies the appearance of the title.

Optional Arguments

“name-1” ... “name-n”

specifies the names of one or more plots that you want to include in legend. Each name that you specify must correspond to a value that you entered for the NAME= option in a plot statement.

Note: If you do not specify a name, then the legend contains references to all of the plots in the graph.

ACROSS= *n*

specifies the number of columns in the legend. By default, the number of columns is determined automatically.

Note: Depending on the number of legend entries and the number of columns and rows that you specify, the legend might not fit in your graph. If your legend does not appear, then you might need to specify a different value for the ACROSS= option.

BORDER | NOBORDER

specifies whether the border around the legend is visible. The BORDER option shows the border. The NOBORDER option hides the border.

Default: BORDER

DOWN= *n*

specifies the number of rows in the legend. By default, the number of rows is determined automatically.

Note: Depending on the number of legend entries and the number of columns and rows that you specify, the legend might not fit in your graph. If your legend does not appear, then you might need to specify a different value for the DOWN= option.

LOCATION= OUTSIDE | INSIDE

specifies whether the legend is placed outside or inside of the axis area. The OUTSIDE option places the legend outside of the axis area. The INSIDE option places the legend inside of the axis area.

Default: OUTSIDE

Interaction: If LOCATION=INSIDE and POSITION= is also specified with any value, you might need to offset the axis in order to make space for the legend within the axis area. For more information, see [“Example 2: Legend with a Position Specified inside the Axis Area” on page 364](#).

POSITION= *position-value*

specifies the position of the legend within the graph. The positions are as follows:

BOTTOM

places the legend at the bottom of the graph.

BOTTOMLEFT

places the legend at the bottom left corner of the graph.

BOTTOMRIGHT

places the legend at the bottom right corner of the graph.

LEFT

places the legend at the left side of the graph.

RIGHT

places the legend at the right side of the graph.

TOP

places the legend at the top of the graph.

TOPLEFT

places the legend at the top left corner of the graph.

TOPRIGHT

places the legend at the top right corner of the graph.

Default: BOTTOM

If LOCATION=INSIDE is specified, the legend is automatically positioned near an edge of the plot area that has the least amount of collision with the data.

Interaction: If LOCATION=INSIDE and POSITION= is also specified with any value, you might need to offset the axis in order to make space for the legend within the axis area. For more information, see [“Example 2: Legend with a Position Specified inside the Axis Area” on page 364](#).

Notes:

By default, if you use more than one KEYLEGEND statement, then each legend is placed in a different position.

If you specify more than one legend with the same position, then all of your legends are placed at that position.

TITLE= “text-string”

specifies a title for the legend.

TITLEATTRS= style-element <(options)> | (options)

specifies the appearance of the title. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

Here is an example that specifies a style element:

```
TITLEATTRS=GraphTitleText
```

options can be one or more of the following:

COLOR= color

specifies the color of the title characters. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphLabelText style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

FAMILY= “font-family”

specifies the font family for the title characters. The SAS ODS styles use TrueType system fonts. For more information, see “TrueType Fonts Supplied by SAS” in Chapter 15 of *SAS Language Reference: Concepts*.

Default: The default font family is specified by the FontFamily attribute of the GraphLabelText style element in the current style.

Restriction: You cannot specify a list of font family names.

SIZE= *n* <*units*>

specifies the font size of the title characters. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see [“Units of Measurement” on page 659](#).

Default: The default font size is specified by the FontSize attribute of the GraphLabelText style element in the current style.

STYLE= ITALIC | NORMAL

specifies whether the title characters are italic or normal.

Default: The default font style is specified by the FontStyle attribute of the GraphLabelText style element in the current style.

WEIGHT= BOLD | NORMAL

specifies whether the title characters are bold or normal.

Default: The default font weight is specified by the FontWeight attribute of the GraphLabelText style element in the current style.

Here is an example expression:

```
TITLEATTRS=(Color=Green Family=Arial Size=8 Style=Italic Weight=Bold)
```

Default: The default appearance of the title text is specified by the GraphLabelText style element in the current style.

VALUEATTRS= *style-element* <(options)> | (options)

specifies the appearance of the legend value labels. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the label characters. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see [“Color-Naming Schemes” in Chapter 14 of SAS/GRAPH: Reference](#).

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphValueText style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

FAMILY= *“font-family”*

specifies the font family for the label characters. The SAS ODS styles use TrueType system fonts. For more information, see [“TrueType Fonts Supplied by SAS” in Chapter 15 of SAS Language Reference: Concepts](#).

Default: The default font family is specified by the FontFamily attribute of the GraphValueText style element in the current style.

Restriction: You cannot specify a list of font family names.

SIZE= *n* <*units*>

specifies the font size of the label characters. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see [“Units of Measurement” on page 659](#).

Default: The default font size is specified by the FontSize attribute of the GraphValueText style element in the current style.

STYLE= ITALIC | NORMAL

specifies whether the label characters are italic or normal.

Default: The default font style is specified by the `FontStyle` attribute of the `GraphValueText` style element in the current style.

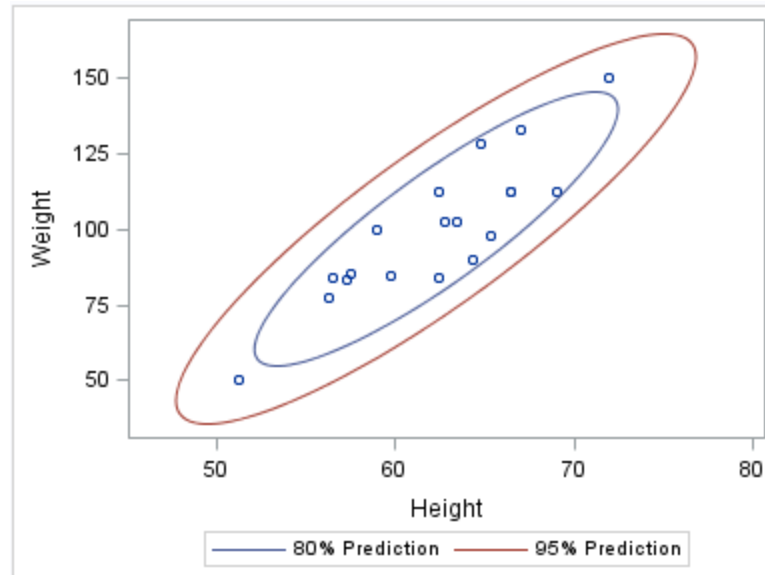
`WEIGHT= BOLD | NORMAL`

specifies whether the label characters are bold or normal.

Default: The default font weight is specified by the `FontWeight` attribute of the `GraphValueText` style element in the current style.

Examples

Example 1: Legend for Two Ellipses



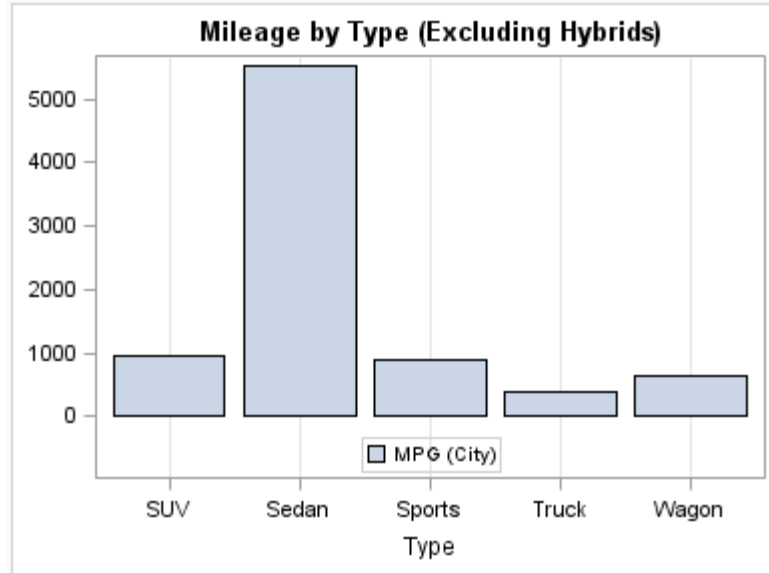
```
ods graphics / reset width=4in;
proc sgplot data=sashelp.class
    noautolegend;
    scatter x=height y=weight;
    ellipse x=height y=weight /
        alpha=.2
        name="eighty"
        legendlabel="80% Prediction";
    ellipse x=height y=weight /
        alpha=.05
        name="ninetyfive"
        legendlabel="95% Prediction";
    keylegend "eighty" "ninetyfive";
run;
```

Example 2: Legend with a Position Specified inside the Axis Area

If `POSITION=` and `LOCATION= INSIDE` are both specified, you might need to offset the axis in order to make space for the legend within the axis area.

The following example specifies `LOCATION= INSIDE` and `POSITION= BOTTOM`. In the graph, the legend collides with the data. To avoid this problem, the graph specifies

an OFFSETMIN= value for the Y axis. (If the legend's position was at the top of the plot, an OFFSETMAX= value would be specified instead.)

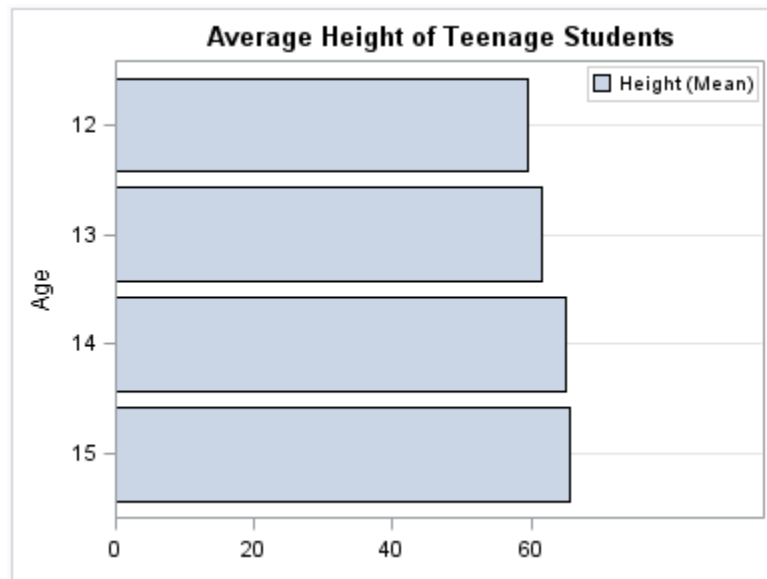


```
ods graphics / reset width=4in;
title "Mileage by Type (Excluding Hybrids)";
proc sgplot data=sashelp.cars;
  where type ne "Hybrid";
  vbar type / response=mpg_city;
  keylegend / location=inside position=bottom;
  xaxis grid;
  yaxis offsetmin=0.15 display=(nolabel);
run;
title;
```

Note: For HBOX, HBAR, HLINE, and DOT plot statements, the Y axis is reversed by default, so the axis origin is at the top. As a result, you use OFFSETMAX= to create an offset at the bottom of the axis. OFFSETMIN= is used to create an offset at the top of the axis for these plot types.

The following example specifies LOCATION= INSIDE and POSITION= TOPRIGHT. To avoid having the legend collide with the graph, the code specifies an OFFSETMAX=

value for the X axis. The OFFSETMAX= value is applied to the end of the axis farther from the origin.



```
ods graphics / reset width=4in;
title "Average Height of Teenage Students";
proc sgplot data=sashelp.class;
  where age >= 12 and age < 16;
  hbar age / response=height stat=mean;
  keylegend / location=inside
    position=topright;
  xaxis offsetmax=0.3 display=(nolabel);
  yaxis grid;
run;
title;
```

LINEPARM Statement

Creates a straight line specified by a point and a slope. You can generate a single line by specifying a constant for each required argument. You can generate multiple lines by specifying a numeric variable for any or all required arguments.

Requirement: The statement must be used with another plot statement that is derived from data values that provide boundaries for the axis area. For example, the LINEPARM statement can be used with a scatter plot or a histogram.

Example: [“About Parameterized Lines” on page 24](#)

Syntax

```
LINEPARM X= numeric-value | numeric-variable
Y= numeric-value | numeric-variable
SLOPE= numeric-value | numeric-variable
</option(s)>;
```

Summary of Optional Arguments

Group options

GROUP= *variable*

specifies a variable that is used to group the data.

NOMISSINGGROUP

specifies that missing values of the group variable are not included in the plot.

Label options

CURVELABEL <="text-string">

adds a label for the line.

CURVELABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the CURVELABEL= option.

CURVELABELLOC= OUTSIDE | INSIDE

specifies whether the curve label is placed inside the plot axes (INSIDE) or outside of the plot axes (OUTSIDE).

CURVELABELPOS= AUTO | MIN | MAX

specifies the location of the curve label.

Line options

CLIP

specifies that the data for the line is ignored when determining the data ranges for the axes.

LINEATTRS= *style-element* <(options)> | (options)

specifies the appearance of the line.

NOEXTEND

prevents the line from being extended beyond the axis offset.

Plot options

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set.

LEGENDLABEL= "text-string"

specifies a label that identifies the plot in the legend.

NAME= "text-string"

specifies a name for the plot.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the lines.

X2AXIS

assigns the X variable to the secondary (top) horizontal axis.

Y2AXIS

assigns the Y variable to the secondary (right) vertical axis.

Required Arguments

X= *numeric-value* | *numeric-variable*

specifies the X coordinate of a point.

Notes:

Values are in the units of the data.

If the value specified for the X= option is outside of the data range, then the data range is extended to include the specified point. This behavior can be changed with the CLIP= option.

If you specify a variable, and the variable contains any missing values, no line is drawn for the respective observation.

Y= *numeric-value* | *numeric-variable*
specifies the Y coordinate of a point.

Notes:

Values are in the units of the data.

If the value specified for the Y= option is outside of the data range, then the data range is extended to include the specified point. This behavior can be changed with the CLIP= option.

If you specify a variable, and the variable contains any missing values, no line is drawn for the respective observation.

SLOPE= *numeric-value* | *numeric-variable*
specifies the slope of the line. The slope can be positive or negative.

Tips:

SLOPE= 0 creates a horizontal line (parallel to the X axis).

SLOPE= . (missing value) creates a vertical line (parallel to the Y axis).

Optional Arguments

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph. For more information, see [“Using SG Attribute Maps to Control Visual Attributes” on page 605](#).

CLIP

specifies that the data for the line is ignored when determining the data ranges for the axes. Each axis scale is determined by the other plots in the overlay. This might result in the line not being displayed if its data range is not within the data ranges of the other plots.

CURVELABEL <=“*text-string*”>

adds a label for the line. You can also specify the label text. If you do not specify a label, then SLOPE=**value** is used. If you specify a GROUP variable, the group value is shown instead of the slope.

CURVELABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the CURVELABEL= option. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

Here is an example that specifies a style element:

CURVELABELATTRS=GraphTitleText

options can be one or more of the following:

COLOR= *color*

specifies the color of the label characters. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphValueText style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

FAMILY= “*font-family*”

specifies the font family for the label characters. The SAS ODS styles use TrueType system fonts. For more information, see “TrueType Fonts Supplied by SAS” in Chapter 15 of *SAS Language Reference: Concepts*.

Default: The default font family is specified by the FontFamily attribute of the GraphValueText style element in the current style.

Restriction: You cannot specify a list of font family names.

SIZE= *n* <*units*>

specifies the font size of the label characters. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default font size is specified by the FontSize attribute of the GraphValueText style element in the current style.

STYLE= ITALIC | NORMAL

specifies whether the label characters are italic or normal.

Default: The default font style is specified by the FontStyle attribute of the GraphValueText style element in the current style.

WEIGHT= BOLD | NORMAL

specifies whether the label characters are bold or normal.

Default: The default font weight is specified by the FontWeight attribute of the GraphValueText style element in the current style.

Here is an example expression:

```
CURVELABELATTRS=(Color=Green Family=Arial Size=8 Style=Italic Weight=Bold)
```

Default: The default style element is GraphValueText.

Interaction: This option has no effect unless the CURVELABEL option is also specified.

CURVELABELLOC= OUTSIDE | INSIDE

specifies whether the curve label is placed inside the plot axes (INSIDE) or outside of the plot axes (OUTSIDE).

Default: INSIDE

Interactions:

This option is used in conjunction with the CURVELABELPOS= option to determine where the labels appear.

This option has no effect unless CURVELABEL is also specified.

CURVELABELPOS= AUTO | MIN | MAX

specifies the location of the curve label. Specify one of the following values:

AUTO

determines the best label position automatically. The line label is positioned near the line boundary along unused axes whenever possible (typically Y2 and X2).

Interaction: This value is available only when CURVELABELLOC=OUTSIDE. If CURVELABELLOC=INSIDE, you must specify either MIN or MAX.

MIN

places the curve label at the minimum value for the X axis.

MAX

places the curve label at the maximum value for the X axis.

Default: AUTO when CURVELABELLOC=OUTSIDE

MAX when CURVELABELLOC=INSIDE

Interaction: This option has no effect unless CURVELABEL is also specified.

GROUP= *variable*

specifies a variable that is used to group the data. A separate plot is created for each unique value of the grouping variable. The plot elements for each group value are automatically distinguished by different visual attributes.

LEGENDLABEL= “*text-string*”

specifies a label that identifies the plot in the legend. By default, the label “LineParm” is used (if you specify a numeric value for Y) or the Y variable name is used (if you specify a variable for Y).

Interaction: The LEGENDLABEL= option has no effect if you also specify the GROUP= option in the same plot statement.

LINEATTRS= *style-element* <(options)> | (options)

specifies the appearance of the line. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

Here is an example that specifies a style element: **LINEATTRS=GraphData3**

For more information about style elements, see [“Style Elements for Use with ODS Graphics” on page 640](#).

options can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphDataDefault style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see [“Line Patterns” on page 663](#).

Default: For ungrouped data, the default line pattern is specified by the LineStyle attribute of the GraphDataDefault style element in the current style. For grouped data, the default line pattern is specified by the LineStyle attribute of the GraphData1 ... GraphData*n* style elements in the current style.

THICKNESS= *n* <units>

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see [“Units of Measurement” on page 659](#).

Default: For ungrouped data, the default line thickness is specified by the LineThickness attribute of the GraphDataDefault style element in the current style. For grouped data, the default line thickness is specified by the LineThickness attribute of the GraphData1 ... GraphData*n* style elements in the current style.

Here is an example expression:

LINEATTRS=(Color= "light green" Pattern= MediumDash Thickness= 4)

Default: GraphDataDefault is the default style element.

NAME= *“text-string”*

specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note: The name is case-sensitive. When you refer to the specified name in other statements, you must use the same exact capitalization.

NOEXTEND

prevents the line from being extended beyond the axis offset. When specified, there might be a 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.

NOMISSINGGROUP

specifies that missing values of the group variable are not included in the plot.

Interaction: This option has no effect unless GROUP= is also specified.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the lines. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default: 0.0

X2AXIS

assigns the X variable to the secondary (top) horizontal axis.

Y2AXIS

assigns the Y variable to the secondary (right) vertical axis.

LOESS Statement

Creates a fitted loess curve.

Example: [“About Loess Plots” on page 34](#)

Syntax

LOESS X= *numeric-variable* Y= *numeric-variable* </ option(s)>;

Summary of Optional Arguments

Label options

CURVELABEL <=“*text-string*”>

adds a label for the curve.

CURVELABELLOC= OUTSIDE | INSIDE

specifies whether the curve label is placed inside the plot axes (INSIDE) or outside of the plot axes (OUTSIDE).

CURVELABELPOS= AUTO | END | MAX | MIN | START

specifies the location of the curve label.

DATALABEL <= *variable*>

displays a label for each data point.

Limit options

CLM <= “*text-string*”>

creates confidence limits.

CLMATTRS = *style-element* | (*CLMAttributeOptions*)

specifies the appearance of the mean value confidence limits by using an ODS style element or by specifying fill and line attributes.

CLMTRANSPARENCY = *numeric-value*

specifies the degree of transparency for the confidence limits.

LOESS options

DEGREE = 1 | 2

specifies the degree of the local polynomials to use for each local regression.

INTERPOLATION = CUBIC | LINEAR

specifies the degree of the interpolating polynomials that are used for blending local polynomial fits at the kd tree vertices.

LINEATTRS = *style-element* <(options)> | (options)

specifies the appearance of the fit curve.

MARKERATTRS = *style-element* <(options)> | (options)

specifies the appearance of the markers in the plot.

MAXPOINTS = *n*

specifies the maximum number of predicted points for the loess fit and the corresponding limits.

NOLEGCLM

hides the legend entry for the mean value confidence limits.

NOLEGFIT

hides the legend entry for the fit line.

NOMARKERS

removes the scatter markers from the plot.

REWEIGHT = *n*

specifies the number of iterative reweighting steps to apply to the data.

SMOOTH = *numeric-value*

specifies a smoothing parameter value.

Plot options

ALPHA = *numeric-value*

specifies the confidence level for the confidence limits.

ATTRID = *variable*

specifies the value of the ID variable in an attribute map data set.

GROUP = *variable*

specifies a variable that is used to group the data.

LEGENDLABEL = “*text-string*”

specifies a label that identifies the fit line in the legend.

NAME = “*text-string*”

specifies a name for the plot.

WEIGHT = *numeric-variable*

specifies how observations are weighted.

X2AXIS

assigns the X variable to the secondary (top) horizontal axis.

Y2AXIS

assigns the Y variable to the secondary (right) vertical axis.

Required Arguments

X= *numeric-variable*
specifies the variable for the x axis.

Y= *numeric-variable*
specifies the variable for the y axis.

Optional Arguments

ALPHA= *numeric-value*
specifies the confidence level for the confidence limits. Specify a number between 0.00 (100% confidence) and 1.00 (0% confidence).

Default: .05

Interaction: This option has no effect unless you also specify the CLM option.

ATTRID= *variable*
specifies the value of the ID variable in an attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph. For more information, see [“Using SG Attribute Maps to Control Visual Attributes” on page 605](#).

CLM <= “*text-string*”>
creates confidence limits. The optional text string overrides the default legend label for the confidence limit.

CLMATTRS = *style-element* | (CLMAttributeOptions)
specifies the appearance of the mean value confidence limits by using an ODS style element or by specifying fill and line attributes. *CLMAttributeOptions* can be one or both of the following:

CLMFILLATTRS= *style-element* | (COLOR= *color*)

You can specify the color of the fill by using a style element or by using the COLOR= suboption. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

CLMLINEATTRS= *style-element* <(options)> | (options)

You can specify the appearance of the outline by using a style element or by using suboptions. If you specify a style element, then you can also specify suboptions to override specific appearance attributes.

The confidence limits have no outlines by default (for most styles). To make the outlines appear, you can change the DISPLAYOPTS in the GRAPHBAND style element. Then you can specify attributes for the outlines. The following code provides an example.

```
proc template;
  define style MyStyleDefault;
    parent=Styles.Default;
    style GraphBand from GraphBand /
      DisplayOpts = "Fill outline";
  end;
run;

ods listing style= MyStyleDefault;
proc sgplot data=sashelp.class;
  reg x=height y=weight / CLM
  CLMATTRS=(CLMLINEATTRS=
```

```
(COLOR=Green PATTERN= ShortDash));
run;
```

The CLMLINEATTRS= *options* can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphConfidence style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see “Line Patterns” on page 663.

Default: For ungrouped data, the default line pattern is specified by the LineStyle attribute of the GraphConfidence style element in the current style. For grouped data, the default line pattern is specified by the LineStyle attribute of the GraphData1 ... GraphData*n* style elements in the current style.

THICKNESS= *n* <units>

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: For ungrouped data, the default line thickness is specified by the LineThickness attribute of the GraphConfidence style element in the current style. For grouped data, the default line thickness is specified by the LineThickness attribute of the GraphData1 ... GraphData*n* style elements in the current style.

Interaction: This suboption has no effect unless you change the display options in the style element to display outlines. See the preceding code example.

Default: The default appearance of the confidence limits is specified by the GraphConfidence style element in the current style.

Interaction: The CLMATTRS = option has no effect unless you also specify the CLM option.

CLMTRANSPARENCY= *numeric-value*

specifies the degree of transparency for the confidence limits. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default: 0.0

Interaction: This option takes effect only if you also specify the CLM option.

CURVELABEL <=“*text-string*”>

adds a label for the curve. You can also specify the label text. If you do not specify a label, the label from the Y variable is used.

CURVELABELLOC= OUTSIDE | INSIDE

specifies whether the curve label is placed inside the plot axes (INSIDE) or outside of the plot axes (OUTSIDE).

Default: INSIDE

CURVELABELPOS= AUTO | END | MAX | MIN | START

specifies the location of the curve label. Specify one of the following values:

AUTO

places the curve label outside the plot area near the end of the curve along unused axes whenever possible (typically Y2 or X2).

Interaction: This value takes effect only when CURVELABELLOC=OUTSIDE.

END

places the curve label at the last point on the curve.

MAX

places the label at the part of the curve closest to the maximum X axis value.

MIN

places the label at the part of the curve closest to the minimum X axis value.

START

places the curve label at the first point on the curve.

Default: END

Interactions:

This option has no effect unless the CURVELABEL option is also specified.

The START and END suboptions take effect only when CURVELABELLOC=INSIDE.

DATALABEL <= variable>

displays a label for each data point. If you specify a variable, then the values of that variable are used for the data labels. If you do not specify a variable, then the values of the Y variable are used for the data labels.

Interaction: This option has no effect if you also specify the NOMARKERS option.

DEGREE= 1 | 2

specifies the degree of the local polynomials to use for each local regression. 1 specifies a linear fit and 2 specifies a quadratic fit.

Default: 1

GROUP= variable

specifies a variable that is used to group the data. A separate plot is created for each unique value of the category variable. The plot elements for each group value are automatically distinguished by different visual attributes.

Interaction: If you specify the GROUP= option in multiple fit plot statements, then the first GROUP= variable is used for all of the fit plots that specify GROUP=.

INTERPOLATION= CUBIC | LINEAR

specifies the degree of the interpolating polynomials that are used for blending local polynomial fits at the kd tree vertices.

Default: CUBIC

LEGENDLABEL= "text-string"

specifies a label that identifies the fit line in the legend. By default, the label "Loess" is used, along with the value of the SMOOTH= option if specified.

LINEATTRS= style-element <(options)> | (options)

specifies the appearance of the fit curve. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphFit style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see “Line Patterns” on page 663.

Default: For ungrouped data, the default line pattern is specified by the LineStyle attribute of the GraphFit style element in the current style. For grouped data, the default line pattern is specified by the LineStyle attribute of the GraphData1 ... GraphData*n* style elements in the current style.

THICKNESS= *n* *<units>*

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: For ungrouped data, the default line thickness is specified by the LineThickness attribute of the GraphFit style element in the current style. For grouped data, the default line thickness is specified by the LineThickness attribute of the GraphData1 ... GraphData*n* style elements in the current style.

MARKERATTRS= *style-element* *<(options)>* | *(options)*

specifies the appearance of the markers in the plot. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the markers. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphDataDefault style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

SIZE= *n* *<units>*

specifies the size of the markers. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default marker size is specified by the MarkerSize attribute of the GraphDataDefault style element in the current style.

SYMBOL= *symbol-name*

specifies the symbol for the markers. For a list of valid marker symbols, see “Marker Symbols” on page 661.

Default: For ungrouped data, the default marker symbol is specified by the MarkerSymbol attribute of the GraphDataDefault style element in the current style. For grouped data, the default marker symbol is specified by the MarkerSymbol attribute of the GraphData1 ... GraphData*n* style elements in the current style.

MAXPOINTS= *n*

specifies the maximum number of predicted points for the loess fit and the corresponding limits.

Default: 201

NAME= “*text-string*”

specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note: The name is case-sensitive. When you refer to the specified name in other statements, you must use the same exact capitalization.

NOLEGCLM

hides the legend entry for the mean value confidence limits.

NOLEGFIT

hides the legend entry for the fit line.

NOMARKERS

removes the scatter markers from the plot.

REWEIGHT= *n*

specifies the number of iterative reweighting steps to apply to the data.

Default: 0

Interaction: This option has no affect if you do not specify the WEIGHT option.

SMOOTH= *numeric-value*

specifies a smoothing parameter value. If you do not specify this option, a smoothing value is determined automatically.

WEIGHT= *numeric-variable*

specifies how observations are weighted. Each observation is weighted by a factor of w for computational purposes, where w is the value of the numeric variable. w can be any numeric value. If w is 0, negative or missing, then that observation is excluded from the analysis.

X2AXIS

assigns the X variable to the secondary (top) horizontal axis.

Y2AXIS

assigns the Y variable to the secondary (right) vertical axis.

Details

For the SMOOTH= option, the smoothing parameter value must be greater than the minimum value that is determined by the following equation.

$$\text{minimum} = \frac{\text{degree} + 1}{\text{number of observations}}$$

NEEDLE Statement

Creates a plot with needles connecting each point to the baseline.

Restriction: The vertical axis that is used with the NEEDLE statement cannot be a discrete axis.

Example: [“About Needle Plots” on page 26](#)

Syntax

NEEDLE X= *variable* Y= *numeric-variable* </ *option(s)*>;

Summary of Optional Arguments

Group options

CLUSTERWIDTH= *numeric-value*

specifies the width of the group clusters as a fraction of the midpoint spacing.

GROUP= *variable*

specifies a variable that is used to group the data.

GROUPDISPLAY= **CLUSTER** | **OVERLAY**

specifies how to display grouped needles.

GROUPORDER= **ASCENDING** | **DESCENDING** | **DATA**

specifies the ordering of needles within a group.

NOMISSINGGROUP

specifies that missing values of the group variable are not included in the plot.

Label options

DATALABEL <= *variable*>

displays a label for each data point.

DATALABELATTRS= *style-element* <(*options*)> | (*options*)

specifies the appearance of the labels in the plot when you use the **DATALABEL**= option.

Needle options

BASELINE= *numeric-value*

specifies a value on the Y axis for the baseline.

DISCRETEOFFSET= *numeric-value*

specifies an amount to offset all needle lines from discrete X values.

LINEATTRS= *style-element* <(*options*)> | (*options*)

specifies the appearance of the needle lines.

MARKERATTRS= *style-element* <(*options*)> | (*options*)

specifies the appearance of the markers in the plot.

MARKERS

adds markers to the tips of the needles.

Plot options

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set.

LEGENDLABEL= " *text-string* "

specifies a label that identifies the needle plot in the legend.

NAME= " *text-string* "

specifies a name for the plot.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the needle plot.

URL= *character-variable*

specifies a character variable that contains URLs for Web pages to be displayed when parts of the plot are selected within an HTML page.

X2AXIS

assigns the X variable to the secondary (top) horizontal axis.

Y2AXIS

assigns the Y variable to the secondary (right) vertical axis.

Required Arguments

X= *variable*

specifies the variable for the x axis.

Y= *numeric-variable*

specifies the variable for the y axis.

Optional Arguments

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph. For more information, see [“Using SG Attribute Maps to Control Visual Attributes” on page 605](#).

BASELINE= *numeric-value*

specifies a value on the Y axis for the baseline.

CLUSTERWIDTH= *numeric-value*

specifies the width of the group clusters as a fraction of the midpoint spacing. Specify a value from 0.0 (narrowest) to 1.0 (widest).

Default: 0.8

Interaction: This option is applicable only when the GROUP option is specified, when GROUPDISPLAY=CLUSTER, and when the category axis is discrete.

DATALABEL <= *variable*>

displays a label for each data point. If you specify a variable, the values of that variable are used for the data labels. If you do not specify a variable, then the values of the Y variable are used for the data labels.

DATALABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the DATALABEL= option. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

Here is an example that specifies a style element:

DATALABELATTRS=GraphLabelText

options can be one or more of the following:

COLOR= *color*

specifies the color of the label characters. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphDataText style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

FAMILY= *“font-family”*

specifies the font family for the label characters. The SAS ODS styles use TrueType system fonts. For more information, see “TrueType Fonts Supplied by SAS” in Chapter 15 of *SAS Language Reference: Concepts*.

Default: The default font family is specified by the FontFamily attribute of the GraphDataText style element in the current style.

SIZE= *n* *<units>*

specifies the font size of the label characters. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default font size is specified by the FontSize attribute of the GraphDataText style element in the current style.

STYLE= ITALIC | NORMAL

specifies whether the label characters are italic (ITALIC) or normal (NORMAL).

Default: The default font style is specified by the FontStyle attribute of the GraphDataText style element in the current style.

WEIGHT= BOLD | NORMAL

specifies whether the label characters are bold (BOLD) or normal (NORMAL).

Default: The default font weight is specified by the FontWeight attribute of the GraphDataText style element in the current style.

Here is an example expression:

```
DATALABELATTRS=(Color=Green Family=Arial Size=8 Style=Italic Weight=Bold)
```

Default: The default style element is GraphDataText.

Interaction: This option has no effect unless the DATALABEL option is also specified.

DISCRETEOFFSET= *numeric-value*

specifies an amount to offset all needle lines from discrete X values. Specify a value from -0.5 (left offset) to +0.5 (right offset).

Default: 0.0 (no offset)

Requirement: This option is applicable only when the X axis is discrete.

GROUP= *variable*

specifies a variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes.

GROUPDISPLAY= CLUSTER | OVERLAY

specifies how to display grouped needles.

CLUSTER

grouped items are drawn adjacent to each other.

OVERLAY

grouped items are drawn at the exact coordinate and might overlap. Each group is represented by unique visual attributes derived from the GraphData1... GraphData*n* style elements in the current style.

Default: OVERLAY

Restriction: GROUPDISPLAY=CLUSTER can take effect only when the category axis is discrete. If necessary, use a TYPE=DISCRETE option on the axis statement.

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

GROUPORDER= ASCENDING | DESCENDING | DATA

specifies the ordering of needles within a group.

ASCENDING

displays each group in ascending group value.

DESCENDING

displays each group in descending group value.

DATA

displays each group in data order of the group variable.

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

LEGENDLABEL= "text-string"

specifies a label that identifies the needle plot in the legend. By default, the label of the Y variable or the group value for each marker is used.

Interaction: The LEGENDLABEL= option has no effect if you also specify the GROUP= option in the same plot statement.

LINEATTRS= style-element <(options)> | (options)

specifies the appearance of the needle lines. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= color

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see "Color-Naming Schemes" in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphDataDefault style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

PATTERN= line-pattern

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see "[Line Patterns](#)" on page 663.

Default: For ungrouped data, the default line pattern is specified by the LineStyle attribute of the GraphDataDefault style element in the current style. For grouped data, the default line pattern is specified by the LineStyle attribute of the GraphData1 ... GraphData*n* style elements in the current style.

THICKNESS= n <units>

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see "[Units of Measurement](#)" on page 659.

Default: For ungrouped data, the default line thickness is specified by the LineThickness attribute of the GraphDataDefault style element in the current style. For grouped data, the default line thickness is specified by the LineThickness attribute of the GraphData1 ... GraphData*n* style elements in the current style.

MARKERATTRS= style-element <(options)> | (options)

specifies the appearance of the markers in the plot. This option has no effect unless you also specify the MARKERS option. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the markers. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphDataDefault style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

SIZE= *n* <*units*>

specifies the size of the markers. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default marker size is specified by the MarkerSize attribute of the GraphDataDefault style element in the current style.

SYMBOL= *symbol-name*

specifies the symbol for the markers. For a list of valid marker symbols, see “Marker Symbols” on page 661.

Default: For ungrouped data, the default marker symbol is specified by the MarkerSymbol attribute of the GraphDataDefault style element in the current style. For grouped data, the default marker symbol is specified by the MarkerSymbol attribute of the GraphData1 ... GraphData*n* style elements in the current style.

MARKERS

adds markers to the tips of the needles.

NAME= “*text-string*”

specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note: The name is case-sensitive. When you refer to the specified name in other statements, you must use the same exact capitalization.

NOMISSINGGROUP

specifies that missing values of the group variable are not included in the plot.

Interaction: This option has no effect unless GROUP= is also specified.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the needle plot. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default: 0.0

URL= *character-variable*

specifies a character variable that contains URLs for Web pages to be displayed when parts of the plot are selected within an HTML page.

Default: By default, no HTML links are created.

Interactions:

This option affects graphics output that is created through the ODS HTML destination only. For more information about ODS destinations, see “Specifying a Destination” on page 649.

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For more information, see “Using the ODS GRAPHICS Statement” on page 651.

X2AXIS

assigns the X variable to the secondary (top) horizontal axis.

Y2AXIS

assigns the Y variable to the secondary (right) vertical axis.

PBSPLINE Statement

Creates a fitted penalized B-spline curve.

Example: [“About Penalized B-Spline Plots” on page 35](#)

Syntax

PBSPLINE *x= numeric-variable y= numeric-variable* </ options>;

Summary of Optional Arguments**Label options**

CURVELABEL <=“*text-string*”>

adds a label for the curve.

CURVELABELLOC= OUTSIDE | INSIDE

specifies whether the curve label is placed inside the plot axes (INSIDE) or outside of the plot axes (OUTSIDE).

CURVELABELPOS= AUTO | END | MAX | MIN | START

specifies the location of the curve label.

DATALABEL <= *variable*>

displays a label for each data point.

Limit options

CLI <= “*text-string*”>

creates prediction limits for the individual predicted values.

CLIATTRS = *style-element* | (CLILINEATTRS= *style-element* <(options)> | (options))

specifies the appearance of the individual value prediction limits by using an ODS style element or by specifying attributes for the outlines.

CLM <= “*text-string*”>

creates confidence limits.

CLMATTRS = *style-element* | (CLMAttributeOptions)

specifies the appearance of the mean value confidence limits by using an ODS style element or by specifying fill and line attributes.

CLMTRANSPARENCY= *numeric-value*

specifies the degree of transparency for the confidence limits.

PBSPLINE options

DEGREE= *n*

specifies the degree of the spline transformation.

LINEATTRS= *style-element* <(options)> | (options)

specifies the appearance of the fitted curve.

MARKERATTRS= *style-element* <(options)> | (options)

specifies the appearance of the markers in the plot.

MAXPOINTS= *n*

specifies the maximum number of predicted points for the spline curve and for any confidence limits.

NKNOTS= *n*

specifies the number of evenly spaced internal knots.

NOLEGCLI

hides the legend entry for the individual value prediction limits.

NOLEGCLM

hides the legend entry for the mean value confidence limits.

NOLEGFIT

hides the legend entry for the fit line.

NOMARKERS

removes the scatter markers from the plot.

SMOOTH= *numeric-value*

specifies a smoothing parameter value.

Plot options

ALPHA= *numeric-value*

specifies the confidence level for the confidence limits.

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set.

FREQ= *numeric-variable*

specifies how many times observations are repeated for computational purposes.

GROUP= *variable*

specifies a variable that is used to group the data.

LEGENDLABEL= *“text-string”*

specifies a label that identifies the B-spline curve in the legend.

NAME= *“text-string”*

specifies a name for the plot.

WEIGHT= *numeric-variable*

specifies how observations are weighted.

X2AXIS

assigns the X variable to the secondary (top) horizontal axis.

Y2AXIS

assigns the Y variable to the secondary (right) vertical axis.

Required Arguments

X= *numeric-variable*

specifies the variable for the x axis.

Y= *numeric-variable*

specifies the variable for the y axis.

Optional Arguments

ALPHA= *numeric-value*

specifies the confidence level for the confidence limits. Specify a number between 0.00 (100% confidence) and 1.00 (0% confidence).

Default: .05

Interaction: This option has no effect unless you specify the CLI or CLM option (or both).

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph. For more information, see [“Using SG Attribute Maps to Control Visual Attributes” on page 605](#).

CLI <= “*text-string*”>

creates prediction limits for the individual predicted values. The optional text string overrides the default legend label for the prediction limits.

CLIATTRS = *style-element* | (CLILINEATTRS= *style-element* <(*options*)> | (*options*))

specifies the appearance of the individual value prediction limits by using an ODS style element or by specifying attributes for the outlines. The outline attributes can be the following:

CLILINEATTRS= *style-element* <(*options*)> | (*options*)

You can specify the appearance of the outline by using a style element or by using suboptions. If you specify a style element, then you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see [“Color-Naming Schemes” in Chapter 14 of SAS/GRAPH: Reference](#).

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphPredictionLimits style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see [“Line Patterns” on page 663](#).

Default: For ungrouped data, the default line pattern is specified by the LineStyle attribute of the GraphPredictionLimits style element in the current style. For grouped data, the default line pattern is specified by the LineStyle attribute of the GraphData1 ... GraphData*n* style elements in the current style.

THICKNESS= *n* <*units*>

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see [“Units of Measurement” on page 659](#).

Default: For ungrouped data, the default line thickness is specified by the LineThickness attribute of the GraphPredictionLimits style element in the current style. For grouped data, the default line thickness is specified by the LineThickness attribute of the GraphData1 ... GraphData*n* style elements in the current style.

Default: The default style element is GraphPredictionLimits.

Interaction: The CLIATTRS = option has no effect unless you also specify the CLI option.

CLM <= "*text-string*">

creates confidence limits. The optional text string overrides the default legend label for the confidence limit.

CLMATTRS = *style-element* | (*CLMAttributeOptions*)

specifies the appearance of the mean value confidence limits by using an ODS style element or by specifying fill and line attributes. *CLMAttributeOptions* can be one or both of the following:

CLMFILLATTRS=*style-element* | (COLOR= *color*)

You can specify the color of the fill by using a style element or by using the COLOR= suboption. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see "Color-Naming Schemes" in Chapter 14 of *SAS/GRAPH: Reference*.

CLMLINEATTRS= *style-element* <(options)> | (options)

You can specify the appearance of the outline by using a style element or by using suboptions. If you specify a style element, then you can also specify suboptions to override specific appearance attributes.

The confidence limits have no outlines by default (for most styles). To make the outlines appear, you can change the DISPLAYOPTS in the GRAPHBAND style element. Then you can specify attributes for the outlines. The following code provides an example.

```
proc template;
  define style MyStyleDefault;
    parent=Styles.Default;
    style GraphBand from GraphBand /
      DisplayOpts = "Fill outline";
  end;
run;

ods listing style= MyStyleDefault;
proc sgplot data=sashelp.class;
  reg x=height y=weight / CLM
  CLMATTRS=(CLMLINEATTRS=
    (COLOR=Green PATTERN= ShortDash));
run;
```

The CLMLINEATTRS= *options* can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see "Color-Naming Schemes" in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphConfidence style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see "[Line Patterns](#)" on page 663.

Default: For ungrouped data, the default line pattern is specified by the LineStyle attribute of the GraphConfidence style element in the current style. For grouped data, the default line pattern is specified by the

LineStyle attribute of the GraphData1 ... GraphData*n* style elements in the current style.

THICKNESS= *n* <units>

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: For ungrouped data, the default line thickness is specified by the LineThickness attribute of the GraphConfidence style element in the current style. For grouped data, the default line thickness is specified by the LineThickness attribute of the GraphData1 ... GraphData*n* style elements in the current style.

Interaction: This suboption has no effect unless you change the display options in the style element to display outlines. See the preceding code example.

Default: The default appearance of the confidence limits is specified by the GraphConfidence style element in the current style.

Interaction: The CLMATTRS = option has no effect unless you also specify the CLM option.

CLMTRANSPARENCY= *numeric-value*

specifies the degree of transparency for the confidence limits. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default: 0.0

Interaction: This option takes effect only if you also specify the CLM option.

CURVELABEL <=“*text-string*”>

adds a label for the curve. You can also specify the label text. If you do not specify a label, the label from the Y variable is used.

CURVELABELLOC= OUTSIDE | INSIDE

specifies whether the curve label is placed inside the plot axes (INSIDE) or outside of the plot axes (OUTSIDE).

Default: INSIDE

CURVELABELPOS= AUTO | END | MAX | MIN | START

specifies the location of the curve label. Specify one of the following values:

AUTO

places the curve label outside the plot area near the end of the curve along unused axes whenever possible (typically Y2 or X2).

Interaction: This value takes effect only when CURVELABELLOC=OUTSIDE.

END

places the curve label at the last point on the curve.

MAX

places the label at the part of the curve closest to the maximum X axis value.

MIN

places the label at the part of the curve closest to the minimum X axis value.

START

places the curve label at the first point on the curve.

Default: END

Interactions:

This option has no effect unless the CURVELABEL option is also specified.

The START and END suboptions take effect only when CURVELABELLOC=INSIDE.

DATALABEL <= *variable*>

displays a label for each data point. If you specify a variable, then the values of that variable are used for the data labels. If you do not specify a variable, then the values of the Y variable are used for the data labels.

Interaction: This option has no effect if you also specify the NOMARKERS option.

DEGREE= *n*

specifies the degree of the spline transformation.

Default: 3

FREQ= *numeric-variable*

specifies how many times observations are repeated for computational purposes. Each observation is repeated *n* times for computational purposes, where *n* is the value of the numeric variable. If *n* is not an integer, then it is truncated to an integer. If *n* is less than 1 or missing, then it is excluded from the analysis.

GROUP= *variable*

specifies a variable that is used to group the data. A separate plot is created for each unique value of the grouping variable. The plot elements for each group value are automatically distinguished by different visual attributes.

Interaction: If you specify the GROUP= option in multiple fit plot statements, then the first GROUP= variable is used for all of the fit plots that specify GROUP=.

LEGENDLABEL= "*text-string*"

specifies a label that identifies the B-spline curve in the legend. By default, the label "Penalized B-Spline" is used with the SMOOTH= value if specified, or else the group value for each B-spline is used.

LINEATTRS= *style-element* <(options)> | (options)

specifies the appearance of the fitted curve. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see "Color-Naming Schemes" in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphFit style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see "[Line Patterns](#)" on page 663.

Default: For ungrouped data, the default line pattern is specified by the LineStyle attribute of the GraphFit style element in the current style. For grouped data, the default line pattern is specified by the LineStyle attribute of the GraphData1 ... GraphData*n* style elements in the current style.

THICKNESS= *n* <*units*>

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see "[Units of Measurement](#)" on page 659.

Default: For ungrouped data, the default line thickness is specified by the LineThickness attribute of the GraphFit style element in the current style. For grouped data, the default line thickness is specified by the LineThickness attribute of the GraphData1 ... GraphData*n* style elements in the current style.

MARKERATTRS= *style-element* <(options)> | (options)

specifies the appearance of the markers in the plot. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the markers. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphDataDefault style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

SIZE= *n* <*units*>

specifies the size of the markers. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default marker size is specified by the MarkerSize attribute of the GraphDataDefault style element in the current style.

SYMBOL= *symbol-name*

specifies the symbol for the markers. For a list of valid marker symbols, see “Marker Symbols” on page 661.

Default: For ungrouped data, the default marker symbol is specified by the MarkerSymbol attribute of the GraphDataDefault style element in the current style. For grouped data, the default marker symbol is specified by the MarkerSymbol attribute of the GraphData1 ... GraphData*n* style elements in the current style.

MAXPOINTS= *n*

specifies the maximum number of predicted points for the spline curve and for any confidence limits.

Default: 201

NAME= “*text-string*”

specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note: The name is case-sensitive. When you refer to the specified name in other statements, you must use the same exact capitalization.

NKNOTS= *n*

specifies the number of evenly spaced internal knots.

Default: 100

Range: 1 to 1000

NOLEGCLI

hides the legend entry for the individual value prediction limits.

NOLEGCLM

hides the legend entry for the mean value confidence limits.

NOLEGFIT

hides the legend entry for the fit line.

NOMARKERS

removes the scatter markers from the plot.

SMOOTH= *numeric-value*

specifies a smoothing parameter value. If you do not specify this option, a smoothing value is determined automatically.

WEIGHT= *numeric-variable*

specifies how observations are weighted. Each observation is weighted by a factor of w for computational purposes, where w is the value of the numeric variable. w can be any numeric value. If w is 0, negative or missing, then that observation is excluded from the analysis.

X2AXIS

assigns the X variable to the secondary (top) horizontal axis.

Y2AXIS

assigns the Y variable to the secondary (right) vertical axis.

Details

For the SMOOTH= option, the smoothing parameter value must be greater than the minimum value that is determined by the following equation.

$$\text{minimum} = \frac{\text{degree} + 1}{\text{number of observations}}$$

REFLINE Statement

Creates a horizontal or vertical reference line.

Example: [“About Reference Lines” on page 23](#)

Syntax

REFLINE *variable* | *value-1* <... *value-n*> </ *option(s)*>;

Summary of Optional Arguments**Line options**

AXIS= X | X2 | Y | Y2

specifies the axis that contains the reference line values.

DISCRETEOFFSET= *numeric-value*

specifies an amount to offset all lines from discrete X or Y values.

LABEL <= (“*text-string-1*” ... “*text-string-n*”)>

creates labels for each reference line.

LABELLOC= INSIDE | OUTSIDE

specifies whether the labels are placed inside or outside of the plot area.

LABELPOS= AUTO | MIN | MAX

specifies the position of the labels.

LINEATTRS= *style-element* <(options)> | (options)

specifies the appearance of the reference line.

NOCLIP

extends the plot axes to contain the reference lines.

Plot options

LEGENDLABEL= *“text-string”*

specifies a label that identifies the markers from the plot in the legend.

NAME= *“text-string”*

specifies a name for the plot.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the reference lines.

Required Arguments**variable**

draws a reference line for each value of the specified variable.

value-1 <... value-n>

draws one or more reference lines at the values that you specify.

Optional Arguments

AXIS= X | X2 | Y | Y2

specifies the axis that contains the reference line values. For example, if you specify **AXIS**= X, vertical reference lines are drawn at points on the X axis.

Default: Y

DISCRETEOFFSET= *numeric-value*

specifies an amount to offset all lines from discrete X or Y values. Specify a value from -0.5 (left offset) to +0.5 (right offset).

Default: 0.0 (no offset)

Requirement: This option is applicable only when the X or Y axis is discrete.

LABEL <= (*“text-string-1”* ... *“text-string-n”*)>

creates labels for each reference line. If you do not specify a label for a line, the reference value for that line is used as the label.

LABELLOC= INSIDE | OUTSIDE

specifies whether the labels are placed inside or outside of the plot area. The INSIDE option places the labels inside of the plot area. The OUTSIDE option places the labels outside of the plot area.

Default: OUTSIDE

LABELPOS= AUTO | MIN | MAX

specifies the position of the labels. MIN specifies the label is placed at the minimum value of the data axis. MAX specifies that the label is placed at the maximum value of the data axis. AUTO places the label automatically.

Default: AUTO

LEGENDLABEL= *“text-string”*

specifies a label that identifies the markers from the plot in the legend. By default, the label “Reference Line” is used.

Interaction: This option has no effect unless you also specify the **NAME**= option.

LINEATTRS= *style-element* <(options)> | (options)

specifies the appearance of the reference line. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: The default color is specified by the ContrastColor attribute of the GraphReference style element in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see “Line Patterns” on page 663.

Default: The default line pattern is specified by the LineStyle attribute of the GraphReference style element in the current style.

THICKNESS= *n* *<units>*

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default line thickness is specified by the LineThickness attribute of the GraphReference style element in the current style.

NAME= “*text-string*”

specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note: The name is case-sensitive. When you refer to the specified name in other statements, you must use the same exact capitalization.

NOCLIP

extends the plot axes to contain the reference lines. By default, if a reference line is created outside of the data range, then the reference line is not visible. This option has no effect if you do not create reference lines that are outside of the data range.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the reference lines. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default: 0.0

REG Statement

Creates a fitted regression line or curve.

Interaction: A linear regression (DEGREE=1) cannot be used with logarithmic axes.

Examples: “About Regression Plots” on page 36
 “Example 4: Adding Prediction and Confidence Bands to a Regression Plot” on page 502

Syntax

REG *X= numeric-variable Y= numeric-variable* *</ option(s)>*;

Summary of Optional Arguments

Label options

CURVELABEL <= “*text-string*”>

adds a label for the curve.

CURVELABELLOC= OUTSIDE | INSIDE

specifies whether the curve label is placed inside the plot axes (INSIDE) or outside of the plot axes (OUTSIDE).

CURVELABELPOS= AUTO | END | MAX | MIN | START

specifies the location of the curve label.

DATALABEL <= *variable*>

displays a label for each data point.

Limit options

CLI <= “*text-string*”>

creates prediction limits for the individual predicted values.

CLIATTRS = *style-element* | (CLILINEATTRS= *style-element* <(options)> | (options))

specifies the appearance of the individual value prediction limits by using an ODS style element or by specifying attributes for the outlines.

CLM <= “*text-string*”>

creates confidence limits.

CLMATTRS = *style-element* | (CLMAttributeOptions)

specifies the appearance of the mean value confidence limits by using an ODS style element or by specifying fill and line attributes.

CLMTRANSPARENCY= *numeric-value*

specifies the degree of transparency for the confidence limits.

Plot options

ALPHA= *numeric-value*

specifies the confidence level for the confidence limits.

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set.

FREQ= *numeric-variable*

specifies how many times observations are repeated for computational purposes.

GROUP= *variable*

specifies a variable that is used to group the data.

LEGENDLABEL= “*text-string*”

specifies a label that identifies the regression curve in the legend.

NAME= “*text-string*”

specifies a name for the plot.

WEIGHT= *numeric-variable*

specifies how observations are weighted.

X2AXIS

assigns the X variable to the secondary (top) horizontal axis.

Y2AXIS

assigns the Y variable to the secondary (right) vertical axis.

REG options

DEGREE= *n*

specifies the degree of the polynomial fit.

LINEATTRS= *style-element* <(options)> | (options)

specifies the appearance of the fit line.

MARKERATTRS= *style-element* <(options)> | (options)
specifies the appearance of the markers in the plot.

MAXPOINTS= *n*
specifies the maximum number of predicted points for the regression curve and for any confidence limits.

NOLEGCLI
hides the legend entry for the individual value confidence limits.

NOLEGCLM
hides the legend entry for the mean value confidence limits.

NOLEGFIT
hides the legend entry for the fit line.

NOMARKERS
removes the scatter markers from the plot.

Required Arguments

X= *numeric-variable*
specifies the variable for the x axis.

Y= *numeric-variable*
specifies the variable for the y axis.

Optional Arguments

ALPHA= *numeric-value*
specifies the confidence level for the confidence limits. Specify a number between 0.00 (100% confidence) and 1.00 (0% confidence).

Default: .05

Interaction: This option has no effect unless you specify the CLI or CLM option (or both).

ATTRID= *variable*
specifies the value of the ID variable in an attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph. For more information, see [“Using SG Attribute Maps to Control Visual Attributes” on page 605](#).

CLI <= “*text-string*”>
creates prediction limits for the individual predicted values. The optional text string overrides the default legend label for the prediction limits.

CLIATTRS = *style-element* | (**CLILINEATTRS=** *style-element* <(options)> | (options))

specifies the appearance of the individual value prediction limits by using an ODS style element or by specifying attributes for the outlines. The outline attributes can be the following:

CLILINEATTRS= *style-element* <(options)> | (options)

You can specify the appearance of the outline by using a style element or by using suboptions. If you specify a style element, then you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*
specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see [“Color-Naming Schemes”](#) in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphPredictionLimits style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see [“Line Patterns” on page 663](#).

Default: For ungrouped data, the default line pattern is specified by the LineStyle attribute of the GraphPredictionLimits style element in the current style. For grouped data, the default line pattern is specified by the LineStyle attribute of the GraphData1 ... GraphData*n* style elements in the current style.

THICKNESS= *n* *<units>*

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see [“Units of Measurement” on page 659](#).

Default: For ungrouped data, the default line thickness is specified by the LineThickness attribute of the GraphPredictionLimits style element in the current style. For grouped data, the default line thickness is specified by the LineThickness attribute of the GraphData1 ... GraphData*n* style elements in the current style.

Default: The default style element is GraphPredictionLimits.

Interaction: The CLIATTRS = option has no effect unless you also specify the CLI option.

CLM *<= “text-string”>*

creates confidence limits. The optional text string overrides the default legend label for the confidence limit.

CLMATTRS = *style-element* | (*CLMAttributeOptions*)

specifies the appearance of the mean value confidence limits by using an ODS style element or by specifying fill and line attributes. *CLMAttributeOptions* can be one or both of the following:

CLMFILLATTRS=*style-element* | (COLOR= *color*)

You can specify the color of the fill by using a style element or by using the COLOR= suboption. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

CLMLINEATTRS= *style-element* *<(options)>* | (*options*)

You can specify the appearance of the outline by using a style element or by using suboptions. If you specify a style element, then you can also specify suboptions to override specific appearance attributes.

The confidence limits have no outlines by default (for most styles). To make the outlines appear, you can change the DISPLAYOPTS in the GRAPHBAND style element. Then you can specify attributes for the outlines. The following code provides an example.

```
proc template;
  define style MyStyleDefault;
    parent=Styles.Default;
    style GraphBand from GraphBand /
      DisplayOpts = "Fill outline";
```

```

end;
run;

ods listing style= MyStyleDefault;
proc sgplot data=sashelp.class;
  reg x=height y=weight / CLM
  CLMATTRS=(CLMLINEATTRS=
    (COLOR=Green PATTERN= ShortDash));
run;

```

The CLMLINEATTRS= *options* can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphConfidence style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see “Line Patterns” on page 663.

Default: For ungrouped data, the default line pattern is specified by the LineStyle attribute of the GraphConfidence style element in the current style. For grouped data, the default line pattern is specified by the LineStyle attribute of the GraphData1 ... GraphData*n* style elements in the current style.

THICKNESS= *n* <units>

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: For ungrouped data, the default line thickness is specified by the LineThickness attribute of the GraphConfidence style element in the current style. For grouped data, the default line thickness is specified by the LineThickness attribute of the GraphData1 ... GraphData*n* style elements in the current style.

Interaction: This suboption has no effect unless you change the display options in the style element to display outlines. See the preceding code example.

Default: The default appearance of the confidence limits is specified by the GraphConfidence style element in the current style.

Interaction: The CLMATTRS = option has no effect unless you also specify the CLM option.

CLMTRANSPARENCY= *numeric-value*

specifies the degree of transparency for the confidence limits. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default: 0.0

Interaction: This option takes effect only if you also specify the CLM option.

CURVELABEL <=“*text-string*”>

adds a label for the curve. You can also specify the label text. If you do not specify a label, the label from the Y variable is used.

CURVELABELLOC= OUTSIDE | INSIDE

specifies whether the curve label is placed inside the plot axes (INSIDE) or outside of the plot axes (OUTSIDE).

Default: INSIDE

CURVELABELPOS= AUTO | END | MAX | MIN | START

specifies the location of the curve label. Specify one of the following values:

AUTO

places the curve label outside the plot area near the end of the curve along unused axes whenever possible (typically Y2 or X2).

Interaction: This value takes effect only when CURVELABELLOC=OUTSIDE.

END

places the curve label at the last point on the curve.

MAX

places the label at the part of the curve closest to the maximum X axis value.

MIN

places the label at the part of the curve closest to the minimum X axis value.

START

places the curve label at the first point on the curve.

Default: END

Interactions:

This option has no effect unless the CURVELABEL option is also specified.

The START and END suboptions take effect only when CURVELABELLOC=INSIDE.

DATALABEL <= *variable*>

displays a label for each data point. If you specify a variable, then the values of that variable are used for the data labels. If you do not specify a variable, then the values of the Y variable are used for the data labels.

Interaction: This option has no effect if you also specify the NOMARKERS option.

DEGREE= *n*

specifies the degree of the polynomial fit. For example, 1 specifies a linear fit, 2 specifies a quadratic fit, and 3 specifies a cubic fit.

Default: 1

FREQ= *numeric-variable*

specifies how many times observations are repeated for computational purposes. Each observation is repeated *n* times for computational purposes, where *n* is the value of the numeric variable. If *n* is not an integer, then it is truncated to an integer. If *n* is less than 1 or missing, then it is excluded from the analysis.

GROUP= *variable*

specifies a variable that is used to group the data. A separate plot is created for each unique value of the grouping variable. The plot elements for each group value are automatically distinguished by different visual attributes.

Interaction: If you specify the GROUP= option in multiple fit plot statements, then the first GROUP= variable is used for all of the fit plots that specify GROUP=.

LEGENDLABEL= “text-string”

specifies a label that identifies the regression curve in the legend. By default, the label “Regression” is used.

Interaction: The LEGENDLABEL= option has no effect if you also specify the GROUP= option in the same plot statement.

LINEATTRS= style-element <(options)> | (options)

specifies the appearance of the fit line. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, then you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= color

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphFit style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

PATTERN= line-pattern

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see “Line Patterns” on page 663.

Default: For ungrouped data, the default line pattern is specified by the LineStyle attribute of the GraphFit style element in the current style. For grouped data, the default line pattern is specified by the LineStyle attribute of the GraphData1 ... GraphData*n* style elements in the current style.

THICKNESS= n <units>

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: For ungrouped data, the default line thickness is specified by the LineThickness attribute of the GraphFit style element in the current style. For grouped data, the default line thickness is specified by the LineThickness attribute of the GraphData1 ... GraphData*n* style elements in the current style.

MARKERATTRS= style-element <(options)> | (options)

specifies the appearance of the markers in the plot. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= color

specifies the color of the markers. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphDataDefault style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

SIZE= n <units>

specifies the size of the markers. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default marker size is specified by the MarkerSize attribute of the GraphDataDefault style element in the current style.

SYMBOL= *symbol-name*

specifies the symbol for the markers. For a list of valid marker symbols, see [“Marker Symbols” on page 661](#).

Default: For ungrouped data, the default marker symbol is specified by the MarkerSymbol attribute of the GraphDataDefault style element in the current style. For grouped data, the default marker symbol is specified by the MarkerSymbol attribute of the GraphData1 ... GraphData*n* style elements in the current style.

MAXPOINTS= *n*

specifies the maximum number of predicted points for the regression curve and for any confidence limits.

Default: 10

NAME= *“text-string”*

specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note: The name is case-sensitive. When you refer to the specified name in other statements, you must use the same exact capitalization.

NOLEGCLI

hides the legend entry for the individual value confidence limits.

NOLEGCLM

hides the legend entry for the mean value confidence limits.

NOLEGFIT

hides the legend entry for the fit line.

NOMARKERS

removes the scatter markers from the plot.

WEIGHT= *numeric-variable*

specifies how observations are weighted. Each observation is weighted by a factor of w for computational purposes, where w is the value of the numeric variable. w can be any numeric value. If w is 0, negative or missing, then that observation is excluded from the analysis.

X2AXIS

assigns the X variable to the secondary (top) horizontal axis.

Y2AXIS

assigns the Y variable to the secondary (right) vertical axis.

SCATTER Statement

Creates a scatter plot.

Examples: [“About Scatter Plots” on page 27](#)

[“Example 1: Grouping a Scatter Plot” on page 498](#)

[“Example 5: Adding a Prediction Ellipse to a Scatter Plot” on page 503](#)

[“Example 6: Creating Lines and Bands from Pre-Computed Data” on page 504](#)

Syntax

SCATTER X= *variable* Y= *variable* < / option(s)>;

Summary of Optional Arguments

Group options

CLUSTERWIDTH= *numeric-value*

specifies the width of the group clusters as a fraction of the midpoint spacing.

GROUP= *variable*

specifies a variable that is used to group the data.

GROUPDISPLAY= **CLUSTER** | **OVERLAY**

specifies how to display grouped markers.

GROUPORDER= **ASCENDING** | **DESCENDING** | **DATA**

specifies the ordering of markers within a group.

NOMISSINGGROUP

specifies that missing values of the group variable are not included in the plot.

Label options

DATALABEL <= *variable*>

displays a label for each data point.

DATALABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the **DATALABEL**= option.

Plot options

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set.

FREQ= *numeric-variable*

specifies how many times observations are repeated for computational purposes.

LEGENDLABEL= “*text-string*”

specifies a label that identifies the markers from the plot in the legend.

NAME= “*text-string*”

specifies a name for the plot.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the markers and error bars.

URL= *character-variable*

specifies a character variable that contains URLs for Web pages to be displayed when parts of the plot are selected within an HTML page.

X2AXIS

assigns the X variable to the secondary (top) horizontal axis.

Y2AXIS

assigns the Y variable to the secondary (right) vertical axis.

Scatter options

DISCRETEOFFSET= *numeric-value*

specifies an amount to offset all markers from discrete X or Y values.

ERRORBARATTRS= *style-element* <(options)> | (options)

specifies the appearance of the error bars in the plot.

MARKERATTRS= *style-element* <(options)> | (options)
 specifies the appearance of the markers in the plot.

MARKERCHAR= *variable*
 specifies a variable whose values replace the marker symbols in the plot.

MARKERCHARATTRS= *style-element* <(options)> | (options)
 specifies the appearance of the markers in the plot when you use the
 MARKERCHAR= option.

XERRORLOWER= *numeric-variable*
 specifies a variable that contains the lower endpoints for the X error bars.

XERRORUPPER= *numeric-variable*
 specifies a variable that contains the upper endpoints for the X error bars.

YERRORLOWER= *numeric-variable*
 specifies a variable that contains the lower endpoints for the Y error bars.

YERRORUPPER= *numeric-variable*
 specifies a variable that contains the upper endpoints for the Y error bars.

Required Arguments

X= *variable*
 specifies the variable for the x axis.

Y= *variable*
 specifies the variable for the y axis.

Optional Arguments

ATTRID= *variable*
 specifies the value of the ID variable in an attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph. For more information, see [“Using SG Attribute Maps to Control Visual Attributes” on page 605](#).

CLUSTERWIDTH= *numeric-value*
 specifies the width of the group clusters as a fraction of the midpoint spacing. Specify a value from 0.0 (narrowest) to 1.0 (widest).

Default: 0.8

Interaction: This option is applicable only when the GROUP option is specified, when GROUPDISPLAY=CLUSTER, and when the category axis is discrete.

DATALABEL <=*variable*>
 displays a label for each data point. If you specify a variable, the values of that variable are used for the data labels. If you do not specify a variable, then the values of the Y variable are used for the data labels.

DATALABELATTRS= *style-element* <(options)> | (options)
 specifies the appearance of the labels in the plot when you use the DATALABEL= option. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

Here is an example that specifies a style element:

DATALABELATTRS=GraphLabelText

options can be one or more of the following:

COLOR= *color*

specifies the color of the label characters. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphDataText style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

FAMILY= “*font-family*”

specifies the font family for the label characters. The SAS ODS styles use TrueType system fonts. For more information, see “TrueType Fonts Supplied by SAS” in Chapter 15 of *SAS Language Reference: Concepts*.

Default: The default font family is specified by the FontFamily attribute of the GraphDataText style element in the current style.

SIZE= *n* <*units*>

specifies the font size of the label characters. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default font size is specified by the FontSize attribute of the GraphDataText style element in the current style.

STYLE= ITALIC | NORMAL

specifies whether the label characters are italic (ITALIC) or normal (NORMAL).

Default: The default font style is specified by the FontStyle attribute of the GraphDataText style element in the current style.

WEIGHT= BOLD | NORMAL

specifies whether the label characters are bold (BOLD) or normal (NORMAL).

Default: The default font weight is specified by the FontWeight attribute of the GraphDataText style element in the current style.

Here is an example expression:

```
DATALABELATTRS=(Color=Green Family=Arial Size=8 Style=Italic Weight=Bold)
```

Default: The default style element is GraphDataText.

Interaction: This option has no effect unless the DATALABEL option is also specified.

DISCRETEOFFSET= *numeric-value*

specifies an amount to offset all markers from discrete X or Y values. Specify a value from -0.5 (left offset) to +0.5 (right offset).

Default: 0.0 (no offset)

Requirement: This option is applicable only when the X or Y axis is discrete.

ERRORBARATTRS= *style-element* <(options)> | (options)

specifies the appearance of the error bars in the plot. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: The default color is specified by the ContrastColor attribute of the GraphError style element in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see [“Line Patterns” on page 663](#).

Default: The default line pattern is specified by the LineStyle attribute of the GraphError style element in the current style.

THICKNESS= *n* *<units>*

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see [“Units of Measurement” on page 659](#).

Default: The default line thickness is specified by the LineThickness attribute of the GraphError style element in the current style.

FREQ= *numeric-variable*

specifies how many times observations are repeated for computational purposes. Each observation is repeated *n* times for computational purposes, where *n* is the value of the numeric variable. If *n* is not an integer, then it is truncated to an integer. If *n* is less than 1 or missing, then it is excluded from the analysis.

GROUP= *variable*

specifies a variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes.

GROUPDISPLAY= CLUSTER | OVERLAY

specifies how to display grouped markers.

CLUSTER

grouped items are drawn adjacent to each other.

OVERLAY

grouped items are drawn at the exact coordinate and might overlap. Each group is represented by unique visual attributes derived from the GraphData1... GraphData*n* style elements in the current style.

Default: OVERLAY

Restriction: GROUPDISPLAY=CLUSTER can take effect only when the category axis is discrete. If necessary, use a TYPE=DISCRETE option on the axis statement.

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

GROUPORDER= ASCENDING | DESCENDING | DATA

specifies the ordering of markers within a group.

ASCENDING

displays each group in ascending group value.

DESCENDING

displays each group in descending group value.

DATA

displays each group in data order of the group variable.

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

LEGENDLABEL= “text-string”

specifies a label that identifies the markers from the plot in the legend. By default, the label of the Y variable or the group value for each marker is used.

Interaction: The LEGENDLABEL= option has no effect if you also specify the GROUP= option in the same plot statement.

MARKERATTRS= *style-element* <(options)> | (options)

specifies the appearance of the markers in the plot. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the markers. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphDataDefault style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

SIZE= *n* <units>

specifies the size of the markers. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default marker size is specified by the MarkerSize attribute of the GraphDataDefault style element in the current style.

SYMBOL= *symbol-name*

specifies the symbol for the markers. For a list of valid marker symbols, see “Marker Symbols” on page 661.

Default: For ungrouped data, the default marker symbol is specified by the MarkerSymbol attribute of the GraphDataDefault style element in the current style. For grouped data, the default marker symbol is specified by the MarkerSymbol attribute of the GraphData1 ... GraphData*n* style elements in the current style.

MARKERCHAR= *variable*

specifies a variable whose values replace the marker symbols in the plot. If a format is associated with the variable, then the formatted values are used as the marker symbols. If there is not a format associated with the variable and the variable contains numeric data, then the BEST6. format is used.

Interaction: The MARKERCHAR= option overrides the DATALABEL= option and the SYMBOL= suboption of the MARKERATTRS= option.

MARKERCHARATTRS= *style-element* <(options)> | (options)

specifies the appearance of the markers in the plot when you use the MARKERCHAR= option. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the marker characters. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphDataText style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

FAMILY= *“font-family”*

specifies the font family for the marker characters. The SAS ODS styles use TrueType system fonts. For more information, see “TrueType Fonts Supplied by SAS” in Chapter 15 of *SAS Language Reference: Concepts*.

Default: The default font family is specified by the FontFamily attribute of the GraphDataText style element in the current style.

SIZE= *n* *<units>*

specifies the font size of the marker characters. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default font size is specified by the FontSize attribute of the GraphDataText style element in the current style.

STYLE= ITALIC | NORMAL

specifies whether the marker characters are italic (ITALIC) or normal (NORMAL).

Default: The default font style is specified by the FontStyle attribute of the GraphDataText style element in the current style.

WEIGHT= BOLD | NORMAL

specifies whether the marker characters are bold (BOLD) or normal (NORMAL).

Default: The default font weight is specified by the FontWeight attribute of the GraphDataText style element in the current style.

NAME= *“text-string”*

specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note: The name is case-sensitive. When you refer to the specified name in other statements, you must use the same exact capitalization.

NOMISSINGGROUP

specifies that missing values of the group variable are not included in the plot.

Interaction: This option has no effect unless GROUP= is also specified.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the markers and error bars. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default: 0.0

URL= *character-variable*

specifies a character variable that contains URLs for Web pages to be displayed when parts of the plot are selected within an HTML page.

Default: By default, no HTML links are created.

Interactions:

This option affects graphics output that is created through the ODS HTML destination only. For more information about ODS destinations, see “Specifying a Destination” on page 649.

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For more information, see “Using the ODS GRAPHICS Statement” on page 651.

X2AXIS

assigns the X variable to the secondary (top) horizontal axis.

XERRORLOWER= *numeric-variable*

specifies a variable that contains the lower endpoints for the X error bars.

XERRORUPPER= *numeric-variable*

specifies a variable that contains the upper endpoints for the X error bars.

Y2AXIS

assigns the Y variable to the secondary (right) vertical axis.

YERRORLOWER= *numeric-variable*

specifies a variable that contains the lower endpoints for the Y error bars.

YERRORUPPER= *numeric-variable*

specifies a variable that contains the upper endpoints for the Y error bars.

SERIES Statement

Creates a line plot.

Examples: [“About Series Plots” on page 29](#)
 [“Example 3: Plotting Three Series” on page 500](#)

Syntax

SERIES X= *variable* Y= *variable* < / option(s)>;

Summary of Optional Arguments

Group options

CLUSTERWIDTH= *numeric-value*

specifies the width of the group clusters as a fraction of the midpoint spacing.

GROUP= *variable*

specifies a variable that is used to group the data.

GROUPDISPLAY= CLUSTER | OVERLAY

specifies how to display grouped lines.

GROUPORDER= ASCENDING | DESCENDING | DATA

specifies the ordering of lines within a group.

NOMISSINGGROUP

specifies that missing values of the group variable are not included in the plot.

Label options

CURVELABEL <=“*text-string*”>

adds a label for the curve.

CURVELABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the CURVELABEL= option.

CURVELABELLOC= OUTSIDE | INSIDE

specifies whether the curve label is placed inside the plot axes (INSIDE) or outside of the plot axes (OUTSIDE).

CURVELABELPOS= AUTO | END | MAX | MIN | START

specifies the location of the curve label.

DATALABEL <= *variable*>

displays a label for each data point.

DATALABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the DATALABEL= option.

Marker options

MARKERATTRS= *style-element* <(options)> | (options)
specifies the appearance of the markers in the plot.

MARKERS

adds data point markers to the series plot data points.

Plot options

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set.

LEGENDLABEL= "text-string"

specifies a label that identifies the series plot in the legend.

NAME= "text-string"

specifies a name for the plot.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the lines and markers.

URL= *character-variable*

specifies a character variable that contains URLs for Web pages to be displayed when parts of the plot are selected within an HTML page.

X2AXIS

assigns the X variable to the secondary (top) horizontal axis.

Y2AXIS

assigns the Y variable to the secondary (right) vertical axis.

Series options

BREAK

creates a break in the line for each missing value for the Y variable.

DISCRETEOFFSET= *numeric-value*

specifies an amount to offset all lines from discrete X or Y values.

LINEATTRS= *style-element* <(options)> | (options)

specifies the appearance of the series line.

Required Arguments

X= *variable*

specifies the variable for the x axis.

Y= *variable*

specifies the variable for the y axis.

Optional Arguments

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph. For more information, see "Using SG Attribute Maps to Control Visual Attributes" on page 605.

BREAK

creates a break in the line for each missing value for the Y variable.

CLUSTERWIDTH= *numeric-value*

specifies the width of the group clusters as a fraction of the midpoint spacing. Specify a value from 0.0 (narrowest) to 1.0 (widest).

Default: 0.8

Interaction: This option is applicable only when the GROUP option is specified, when GROUPDISPLAY=CLUSTER, and when the category axis is discrete.

CURVELABEL <=“*text-string*”>

adds a label for the curve. You can also specify the label text. If you do not specify a label, the label from the Y variable is used.

CURVELABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the CURVELABEL= option. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

Here is an example that specifies a style element:

CURVELABELATTRS=GraphTitleText

options can be one or more of the following:

COLOR= *color*

specifies the color of the label characters. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphValueText style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

FAMILY= “*font-family*”

specifies the font family for the label characters. The SAS ODS styles use TrueType system fonts. For more information, see “TrueType Fonts Supplied by SAS” in Chapter 15 of *SAS Language Reference: Concepts*.

Default: The default font family is specified by the FontFamily attribute of the GraphValueText style element in the current style.

Restriction: You cannot specify a list of font family names.

SIZE= *n* <*units*>

specifies the font size of the label characters. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default font size is specified by the FontSize attribute of the GraphValueText style element in the current style.

STYLE= ITALIC | NORMAL

specifies whether the label characters are italic or normal.

Default: The default font style is specified by the FontStyle attribute of the GraphValueText style element in the current style.

WEIGHT= BOLD | NORMAL

specifies whether the label characters are bold or normal.

Default: The default font weight is specified by the FontWeight attribute of the GraphValueText style element in the current style.

Here is an example expression:

CURVELABELATTRS=(Color=Green Family=Arial Size=8 Style=Italic Weight=Bold)

Default: The default style element is GraphValueText.

Interaction: This option has no effect unless the CURVELABEL option is also specified.

CURVELABELLOC= OUTSIDE | INSIDE

specifies whether the curve label is placed inside the plot axes (INSIDE) or outside of the plot axes (OUTSIDE).

Default: INSIDE

CURVELABELPOS= AUTO | END | MAX | MIN | START

specifies the location of the curve label. Specify one of the following values:

AUTO

places the curve label outside the plot area near the end of the curve along unused axes whenever possible (typically Y2 or X2).

Interaction: This value takes effect only when CURVELABELLOC=OUTSIDE.

END

places the curve label at the last point on the curve.

MAX

places the label at the part of the curve closest to the maximum X axis value.

MIN

places the label at the part of the curve closest to the minimum X axis value.

START

places the curve label at the first point on the curve.

Default: END

Interactions:

This option has no effect unless the CURVELABEL option is also specified.

The START and END suboptions take effect only when CURVELABELLOC=INSIDE.

DATALABEL <= variable>

displays a label for each data point. If you specify a variable, the values of that variable are used for the data labels. If you do not specify a variable, then the values of the Y variable are used for the data labels.

DATALABELATTRS= style-element <(options)> | (options)

specifies the appearance of the labels in the plot when you use the DATALABEL= option. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

Here is an example that specifies a style element:

DATALABELATTRS=GraphLabelText

options can be one or more of the following:

COLOR= *color*

specifies the color of the label characters. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphDataText style element in the current style. For grouped data, the

default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

FAMILY= “*font-family*”

specifies the font family for the label characters. The SAS ODS styles use TrueType system fonts. For more information, see “TrueType Fonts Supplied by SAS” in Chapter 15 of *SAS Language Reference: Concepts*.

Default: The default font family is specified by the FontFamily attribute of the GraphDataText style element in the current style.

SIZE= *n* <*units*>

specifies the font size of the label characters. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default font size is specified by the FontSize attribute of the GraphDataText style element in the current style.

STYLE= ITALIC | NORMAL

specifies whether the label characters are italic (ITALIC) or normal (NORMAL).

Default: The default font style is specified by the FontStyle attribute of the GraphDataText style element in the current style.

WEIGHT= BOLD | NORMAL

specifies whether the label characters are bold (BOLD) or normal (NORMAL).

Default: The default font weight is specified by the FontWeight attribute of the GraphDataText style element in the current style.

Here is an example expression:

```
DATALABELATTRS=(Color=Green Family=Arial Size=8 Style=Italic Weight=Bold)
```

Default: The default style element is GraphDataText.

Interaction: This option has no effect unless the DATALABEL option is also specified.

DISCRETEOFFSET= *numeric-value*

specifies an amount to offset all lines from discrete X or Y values. Specify a value from -0.5 (left offset) to +0.5 (right offset).

Default: 0.0 (no offset)

Requirement: This option is applicable only when the X or Y axis is discrete.

GROUP= *variable*

specifies a variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes.

GROUPDISPLAY= CLUSTER | OVERLAY

specifies how to display grouped lines.

CLUSTER

grouped items are drawn adjacent to each other.

OVERLAY

grouped items are drawn at the exact coordinate and might overlap. Each group is represented by unique visual attributes derived from the GraphData1... GraphData*n* style elements in the current style.

Default: OVERLAY

Restriction: GROUPDISPLAY=CLUSTER can take effect only when the category axis is discrete. If necessary, use a TYPE=DISCRETE option on the axis statement.

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

GROUPORDER= ASCENDING | DESCENDING | DATA

specifies the ordering of lines within a group.

ASCENDING

displays each group in ascending group value.

DESCENDING

displays each group in descending group value.

DATA

displays each group in data order of the group variable.

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

LEGENDLABEL= “text-string”

specifies a label that identifies the series plot in the legend. By default, the label of the Y variable or the group value for each marker is used.

LINEATTRS= style-element <(options)> | (options)

specifies the appearance of the series line. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= color

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphDataDefault style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

PATTERN= line-pattern

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see “Line Patterns” on page 663.

Default: For ungrouped data, the default line pattern is specified by the LineStyle attribute of the GraphDataDefault style element in the current style. For grouped data, the default line pattern is specified by the LineStyle attribute of the GraphData1 ... GraphData*n* style elements in the current style.

THICKNESS= *n* <units>

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: For ungrouped data, the default line thickness is specified by the LineThickness attribute of the GraphDataDefault style element in the current style. For grouped data, the default line thickness is specified by the LineThickness attribute of the GraphData1 ... GraphData*n* style elements in the current style.

MARKERATTRS= style-element <(options)> | (options)

specifies the appearance of the markers in the plot. This option has no effect unless you also specify the MARKERS option. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the markers. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphDataDefault style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

SIZE= *n* *<units>*

specifies the size of the markers. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default marker size is specified by the MarkerSize attribute of the GraphDataDefault style element in the current style.

SYMBOL= *symbol-name*

specifies the symbol for the markers. For a list of valid marker symbols, see “Marker Symbols” on page 661.

Default: For ungrouped data, the default marker symbol is specified by the MarkerSymbol attribute of the GraphDataDefault style element in the current style. For grouped data, the default marker symbol is specified by the MarkerSymbol attribute of the GraphData1 ... GraphData*n* style elements in the current style.

MARKERS

adds data point markers to the series plot data points.

NAME= “*text-string*”

specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note: The name is case-sensitive. When you refer to the specified name in other statements, you must use the same exact capitalization.

NOMISSINGGROUP

specifies that missing values of the group variable are not included in the plot.

Interaction: This option has no effect unless GROUP= is also specified.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the lines and markers. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default: 0.0

URL= *character-variable*

specifies a character variable that contains URLs for Web pages to be displayed when parts of the plot are selected within an HTML page.

Default: By default, no HTML links are created.

Interactions:

This option affects graphics output that is created through the ODS HTML destination only. For more information about ODS destinations, see “Specifying a Destination” on page 649.

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For more information, see “Using the ODS GRAPHICS Statement” on page 651.

X2AXIS

assigns the X variable to the secondary (top) horizontal axis.

Y2AXIS

assigns the Y variable to the secondary (right) vertical axis.

STEP Statement

Creates a step plot.

Restriction: The vertical axis that is used with the STEP statement cannot be a discrete axis.

Example: [“About Step Plots” on page 30](#)

Syntax

STEP X= *variable* Y= *numeric-variable* < / option(s)>;

Summary of Optional Arguments**Group options**

CLUSTERWIDTH= *numeric-value*

specifies the width of the group clusters as a fraction of the midpoint spacing.

GROUP= *variable*

specifies a variable that is used to group the data.

GROUPDISPLAY= **CLUSTER** | **OVERLAY**

specifies how to display grouped step lines.

GROUPORDER= **ASCENDING** | **DESCENDING** | **DATA**

specifies the ordering of step lines within a group.

NOMISSINGGROUP

specifies that missing values of the group variable are not included in the plot.

Label options

CURVELABEL <=“*text-string*”>

adds a label for the curve.

CURVELABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the **CURVELABEL**= option.

CURVELABELLOC= **OUTSIDE** | **INSIDE**

specifies whether the curve label is placed inside the plot axes (**INSIDE**) or outside of the plot axes (**OUTSIDE**).

CURVELABELPOS= **AUTO** | **END** | **MAX** | **MIN** | **START**

specifies the location of the curve label.

DATALABEL <= *variable*>

displays a label for each data point.

DATALABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the **DATALABEL**= option.

Marker options

MARKERATTRS= *style-element* <(options)> | (options)

specifies the appearance of the markers in the plot.

MARKERS

adds markers to the step plot data points.

Plot options

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set.

LEGENDLABEL= *“text-string”*

specifies a label that identifies the step plot in the legend.

NAME= *“text-string”*

specifies a name for the plot.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the lines.

URL= *character-variable*

specifies a character variable that contains URLs for Web pages to be displayed when parts of the plot are selected within an HTML page.

X2AXIS

assigns the X variable to the secondary (top) horizontal axis.

Y2AXIS

assigns the Y variable to the secondary (right) vertical axis.

Step options**BREAK**

creates a break in the line for each missing value for the Y variable.

DISCRETEOFFSET= *numeric-value*

specifies an amount to offset all step lines from discrete X values.

ERRORBARATTRS= *style-element* *<(options)>* | *(options)*

specifies the appearance of the error bars in the plot.

JUSTIFY= *LEFT* | *CENTER* | *RIGHT*

specifies the location of each step relative to its data point.

LINEATTRS= *style-element* *<(options)>* | *(options)*

specifies the appearance of the step line.

YERRORLOWER= *numeric-variable*

specifies a variable that contains the lower endpoints for the Y error bars.

YERRORUPPER= *numeric-variable*

specifies a variable that contains the upper endpoints for the Y error bars.

Required Arguments

X= *variable*

specifies the variable for the x axis.

Y= *numeric-variable*

specifies the variable for the y axis.

Optional Arguments

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph. For more information, see [“Using SG Attribute Maps to Control Visual Attributes”](#) on page 605.

BREAK

creates a break in the line for each missing value for the Y variable.

CLUSTERWIDTH= *numeric-value*

specifies the width of the group clusters as a fraction of the midpoint spacing. Specify a value from 0.0 (narrowest) to 1.0 (widest).

Default: 0.8

Interaction: This option is applicable only when the GROUP option is specified, when GROUPDISPLAY=CLUSTER, and when the category axis is discrete.

CURVELABEL <=“*text-string*”>

adds a label for the curve. You can also specify the label text. If you do not specify a label, the label from the Y variable is used.

CURVELABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the CURVELABEL= option. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

Here is an example that specifies a style element:

CURVELABELATTRS=GraphTitleText

options can be one or more of the following:

COLOR= *color*

specifies the color of the label characters. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphValueText style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

FAMILY= “*font-family*”

specifies the font family for the label characters. The SAS ODS styles use TrueType system fonts. For more information, see “TrueType Fonts Supplied by SAS” in Chapter 15 of *SAS Language Reference: Concepts*.

Default: The default font family is specified by the FontFamily attribute of the GraphValueText style element in the current style.

Restriction: You cannot specify a list of font family names.

SIZE= *n* <units>

specifies the font size of the label characters. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default font size is specified by the FontSize attribute of the GraphValueText style element in the current style.

STYLE= ITALIC | NORMAL

specifies whether the label characters are italic or normal.

Default: The default font style is specified by the FontStyle attribute of the GraphValueText style element in the current style.

WEIGHT= BOLD | NORMAL

specifies whether the label characters are bold or normal.

Default: The default font weight is specified by the FontWeight attribute of the GraphValueText style element in the current style.

Here is an example expression:

```
CURVELABELATTRS=(Color=Green Family=Arial Size=8 Style=Italic Weight=Bold)
```

Default: The default style element is GraphValueText.

Interaction: This option has no effect unless the CURVELABEL option is also specified.

CURVELABELLOC= OUTSIDE | INSIDE

specifies whether the curve label is placed inside the plot axes (INSIDE) or outside of the plot axes (OUTSIDE).

Default: INSIDE

CURVELABELPOS= AUTO | END | MAX | MIN | START

specifies the location of the curve label. Specify one of the following values:

AUTO

places the curve label outside the plot area near the end of the curve along unused axes whenever possible (typically Y2 or X2).

Interaction: This value takes effect only when CURVELABELLOC=OUTSIDE.

END

places the curve label at the last point on the curve.

MAX

places the label at the part of the curve closest to the maximum X axis value.

MIN

places the label at the part of the curve closest to the minimum X axis value.

START

places the curve label at the first point on the curve.

Default: END

Interactions:

This option has no effect unless the CURVELABEL option is also specified.

The START and END suboptions take effect only when CURVELABELLOC=INSIDE.

DATALABEL <= variable>

displays a label for each data point. If you specify a variable, the values of that variable are used for the data labels. If you do not specify a variable, then the values of the Y variable are used for the data labels.

DATALABELATTRS= style-element <(options)> | (options)

specifies the appearance of the labels in the plot when you use the DATALABEL= option. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

Here is an example that specifies a style element:

```
DATALABELATTRS=GraphLabelText
```

options can be one or more of the following:

COLOR= color

specifies the color of the label characters. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphDataText style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

FAMILY= *“font-family”*

specifies the font family for the label characters. The SAS ODS styles use TrueType system fonts. For more information, see “TrueType Fonts Supplied by SAS” in Chapter 15 of *SAS Language Reference: Concepts*.

Default: The default font family is specified by the FontFamily attribute of the GraphDataText style element in the current style.

SIZE= *n* *<units>*

specifies the font size of the label characters. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default font size is specified by the FontSize attribute of the GraphDataText style element in the current style.

STYLE= ITALIC | NORMAL

specifies whether the label characters are italic (ITALIC) or normal (NORMAL).

Default: The default font style is specified by the FontStyle attribute of the GraphDataText style element in the current style.

WEIGHT= BOLD | NORMAL

specifies whether the label characters are bold (BOLD) or normal (NORMAL).

Default: The default font weight is specified by the FontWeight attribute of the GraphDataText style element in the current style.

Here is an example expression:

```
DATALABELATTRS=(Color=Green Family=Arial Size=8 Style=Italic Weight=Bold)
```

Default: The default style element is GraphDataText.

Interaction: This option has no effect unless the DATALABEL option is also specified.

DISCRETEOFFSET= *numeric-value*

specifies an amount to offset all step lines from discrete X values. Specify a value from -0.5 (left offset) to +0.5 (right offset).

Default: 0.0 (no offset)

Requirement: This option is applicable only when the X axis is discrete.

ERRORBARATTRS= *style-element* *<(options)>* | *(options)*

specifies the appearance of the error bars in the plot. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: The default color is specified by the ContrastColor attribute of the GraphError style element in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see “Line Patterns” on page 663.

Default: The default line pattern is specified by the LineStyle attribute of the GraphError style element in the current style.

THICKNESS= *n* <*units*>

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default line thickness is specified by the LineThickness attribute of the GraphError style element in the current style.

GROUP= *variable*

specifies a variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes.

GROUPDISPLAY= CLUSTER | OVERLAY

specifies how to display grouped step lines.

CLUSTER

grouped items are drawn adjacent to each other.

OVERLAY

grouped items are drawn at the exact coordinate and might overlap. Each group is represented by unique visual attributes derived from the GraphData1... GraphData*n* style elements in the current style.

Default: OVERLAY

Restriction: GROUPDISPLAY=CLUSTER can take effect only when the category axis is discrete. If necessary, use a TYPE=DISCRETE option on the axis statement.

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

GROUPORDER= ASCENDING | DESCENDING | DATA

specifies the ordering of step lines within a group.

ASCENDING

displays each group in ascending group value.

DESCENDING

displays each group in descending group value.

DATA

displays each group in data order of the group variable.

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

JUSTIFY= LEFT | CENTER | RIGHT

specifies the location of each step relative to its data point. Figure 7.3 on page 418 shows the effect of each option:

Figure 7.3 Values for JUSTIFY=



LEGENDLABEL= “*text-string*”

specifies a label that identifies the step plot in the legend. By default, the label of the Y variable or the group value for each marker is used.

LINEATTRS= *style-element* <(options)> | (options)

specifies the appearance of the step line. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphDataDefault style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see “Line Patterns” on page 663.

Default: For ungrouped data, the default line pattern is specified by the LineStyle attribute of the GraphDataDefault style element in the current style. For grouped data, the default line pattern is specified by the LineStyle attribute of the GraphData1 ... GraphData*n* style elements in the current style.

THICKNESS= *n* <units>

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: For ungrouped data, the default line thickness is specified by the LineThickness attribute of the GraphDataDefault style element in the current style. For grouped data, the default line thickness is specified by the LineThickness attribute of the GraphData1 ... GraphData*n* style elements in the current style.

MARKERATTRS= *style-element* <(options)> | (options)

specifies the appearance of the markers in the plot. This option has no effect unless you also specify the MARKERS option. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the markers. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphDataDefault style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

SIZE= *n* <units>

specifies the size of the markers. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default marker size is specified by the MarkerSize attribute of the GraphDataDefault style element in the current style.

SYMBOL= *symbol-name*

specifies the symbol for the markers. For a list of valid marker symbols, see [“Marker Symbols” on page 661](#).

Default: For ungrouped data, the default marker symbol is specified by the MarkerSymbol attribute of the GraphDataDefault style element in the current style. For grouped data, the default marker symbol is specified by the MarkerSymbol attribute of the GraphData1 ... GraphData*n* style elements in the current style.

MARKERS

adds markers to the step plot data points.

NAME= *“text-string”*

specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note: The name is case-sensitive. When you refer to the specified name in other statements, you must use the same exact capitalization.

NOMISSINGGROUP

specifies that missing values of the group variable are not included in the plot.

Interaction: This option has no effect unless GROUP= is also specified.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the lines. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default: 0.0

URL= *character-variable*

specifies a character variable that contains URLs for Web pages to be displayed when parts of the plot are selected within an HTML page.

Default: By default, no HTML links are created.

Interactions:

This option affects graphics output that is created through the ODS HTML destination only. For more information about ODS destinations, see [“Specifying a Destination” on page 649](#).

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For more information, see [“Using the ODS GRAPHICS Statement” on page 651](#).

X2AXIS

assigns the X variable to the secondary (top) horizontal axis.

YERRORLOWER= *numeric-variable*

specifies a variable that contains the lower endpoints for the Y error bars.

YERRORUPPER= *numeric-variable*

specifies a variable that contains the upper endpoints for the Y error bars.

Y2AXIS

assigns the Y variable to the secondary (right) vertical axis.

VBAR Statement

Creates a vertical bar chart that summarizes the values of a category variable.

Interaction: The VBAR statement can be combined only with other categorization plot statements in the SGLOT procedure. See [“Plot Type Compatibility” on page 277](#).

Examples: “About Bar Charts” on page 42
 “Example 10: Creating a Bar-Line Chart” on page 509

Syntax

VBAR *category-variable* </ *option(s)*>;

Summary of Optional Arguments

Bar options

BARWIDTH= *numeric-value*

specifies the width of the bars as a ratio of the maximum possible width.

DATASKIN= NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN

specifies a special effect to be used on all filled bars.

DISCRETEOFFSET= *numeric-value*

specifies an amount to offset all bars from the category midpoints.

FILL | **NOFILL**

specifies whether the bars are filled.

FILLATTRS= *style-element* | (COLOR= *color*)

specifies the appearance of the fill for the bars.

MISSING

processes missing values as a valid category value and creates a bar for it.

NOSTATLABEL

removes the statistic name from the axis and legend labels.

NUMSTD= *n*

specifies the number of standard units for the limit lines, when you specify
LIMITSTAT= STDDEV or **LIMITSTAT**= STDERR.

OUTLINE | **NOOUTLINE**

specifies whether the bars have outlines.

RESPONSE= *response-variable*

specifies a numeric response variable for the plot.

STAT= FREQ | MEAN | SUM

specifies the statistic for the vertical axis.

Group options

CLUSTERWIDTH= *numeric-value*

specifies the cluster width as a ratio of the maximum width.

GROUP= *variable*

specifies a variable that is used to group the data.

GROUPDISPLAY= STACK | CLUSTER

specifies how to display grouped bars.

GROUPORDER= ASCENDING | DESCENDING | DATA

specifies the ordering of bars within a group.

Label options

DATALABEL <= *variable*>

displays a label for each data point.

DATALABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the
DATALABEL= option.

DATALABELPOS= DATA | BOTTOM | TOP

specifies the location of the data label.

Limit options

LIMITATTRS= *style-element* <(*options*)> | (*options*)

specifies the appearance of the limit lines in the plot.

LIMITS= BOTH | LOWER | UPPER

specifies which limit lines to display.

LIMITSTAT= CLM | STDDEV | STDERR

specifies the statistic for the limit lines.

Plot options

ALPHA= *numeric-value*

specifies the confidence level for the confidence limits.

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set.

CATEGORYORDER= RESPASC | RESPDESC

specifies the order in which the response values are arranged.

FREQ= *numeric-variable*

specifies how many times observations are repeated for computational purposes.

LEGENDLABEL= “*text-string*”

specifies the label that identifies the bar chart in the legend.

NAME= “*text-string*”

specifies a name for the plot.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the bars and limits, if displayed.

URL= *character-variable*

specifies a character variable that contains URLs for Web pages to be displayed when parts of the plot are selected within an HTML page.

WEIGHT= *numeric-variable*

specifies how observations are weighted.

Required Argument

category-variable

specifies the variable that classifies the observations into distinct subsets.

Optional Arguments

ALPHA= *numeric-value*

specifies the confidence level for the confidence limits. Specify a number between 0.00 (100% confidence) and 1.00 (0% confidence).

Default: .05

Interactions:

This option has no effect if you do not specify LIMITSTAT=CLM.

If your plot is overlaid with other categorization plots, then the first ALPHA value that you specify is used for all of the plots.

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

For more information, see [“Using SG Attribute Maps to Control Visual Attributes” on page 605](#).

BARWIDTH= *numeric-value*

specifies the width of the bars as a ratio of the maximum possible width. The maximum width is equal to the distance between the center of each bar and the centers of the adjacent bars. Specify a value from 0.0 (narrowest) to 1.0 (widest).

For example, if you specify a width of **1**, then there is no space between the bars. If you specify a width of **.5**, then the width of the bars is equal to the space between the bars.

Default: .8

1.0 when the GROUP option is specified and GROUPDISPLAY=CLUSTER

Interaction: When the GROUP option is specified, the bar width is determined by the maximum number of bars in any one group cluster. All bars are drawn with the same width, with the cluster being positioned symmetrically around the midpoint.

CATEGORYORDER= RESPASC | RESPDESC

specifies the order in which the response values are arranged. Specify one of the following values:

RESPASC

sorts the response values in ascending order.

RESPDESC

sorts the response values in descending order.

Default: By default, the plot is sorted in ascending order based on the category values.

Restrictions:

This option takes effect only when the plot statement specifies a response variable and the axis for that variable is numeric. If the axis is not numeric, an error is generated and a message is written to the SAS log.

Uniform scaling and response sorting cannot occur on the same axis. If the UNIFORM= option is used in the SGPLOT statement, the UNIFORM option is ignored for the sorted response axis and a note is generated in the log. The UNIFORM= option is applied to the other axes and groups if requested. Note that the UNIFORM= option can selectively apply scaling to only the X or Y axis

Interaction: If a group variable is specified in the plot statement, the response sort option is ignored and a note is generated in the log. (You can use a BY statement in the procedure if you want to group data.)

Note: If CATEGORYORDER is specified in multiple statements, the procedure sorts by the last statement in which it is specified.

CLUSTERWIDTH= *numeric-value*

specifies the cluster width as a ratio of the maximum width. Specify a value from 0.0 (narrowest) to 1.0 (widest).

CLUSTERWIDTH is the fraction of the midpoint spacing used by all bars that are clustered around a midpoint (category value). The bar width is applied to the maximum bar spacing divided by the maximum number of bars in any one cluster.

Default: 0.8

Interaction: This option is applicable only when the GROUP option is specified and when GROUPDISPLAY=CLUSTER.

DATALABEL <=*variable*>

displays a label for each data point. If you specify a variable, then the values of that variable are used for the data labels. If you do not specify a variable, then the values of the calculated response are used for the data labels.

DATALABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the DATALABEL= option. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

Here is an example that specifies a style element:

```
DATALABELATTRS=GraphLabelText
```

options can be one or more of the following:

COLOR= *color*

specifies the color of the label characters. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphDataText style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

FAMILY= “*font-family*”

specifies the font family for the label characters. The SAS ODS styles use TrueType system fonts. For more information, see “TrueType Fonts Supplied by SAS” in Chapter 15 of *SAS Language Reference: Concepts*.

Default: The default font family is specified by the FontFamily attribute of the GraphDataText style element in the current style.

SIZE= *n* <*units*>

specifies the font size of the label characters. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default font size is specified by the FontSize attribute of the GraphDataText style element in the current style.

STYLE= ITALIC | NORMAL

specifies whether the label characters are italic (ITALIC) or normal (NORMAL).

Default: The default font style is specified by the FontStyle attribute of the GraphDataText style element in the current style.

WEIGHT= BOLD | NORMAL

specifies whether the label characters are bold (BOLD) or normal (NORMAL).

Default: The default font weight is specified by the FontWeight attribute of the GraphDataText style element in the current style.

Here is an example expression:

```
DATALABELATTRS=(Color=Green Family=Arial Size=8 Style=Italic Weight=Bold)
```

Default: The default style element is GraphDataText.

Interaction: This option has no effect unless the DATALABEL option is also specified.

DATALABELPOS= DATA | BOTTOM | TOP

specifies the location of the data label. Specify one of the following values:

DATA

places the label on the data primitives (at the upper boundary of the bars).

BOTTOM

places the label below the bars.

TOP

places the label above the bars.

Default: DATA

Interactions:

This option displays limit information when limits are specified. When limits are specified, the default data label position is BOTTOM.

This option displays group values for each category when GROUP= is also specified.

This option displays response values for each overlaid chart.

This option has no effect unless you also specify the DATALABEL option.

DATASKIN= NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN

specifies a special effect to be used on all filled bars. Specify one of the following (the samples were generated using the HTMLBlue style):

NONE



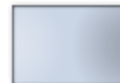
CRISP



GLOSS



MATTE



PRESSED



SHEEN



Default: NONE

Interactions:

This option has no effect if the applied style uses a fill pattern and FILL is disabled for the style. If the style has both FILL and FILLPATTERN enabled, data skins will work. For more information about fill patterns, see [“Using Fill Patterns to Distinguish Grouped Bar Charts” on page 636](#). For instructions about viewing the contents of style templates, see [“About Styles and Style Elements” on page 620](#).

This option has no effect if you also specify NOFILL.

DISCRETEOFFSET= *numeric-value*

specifies an amount to offset all bars from the category midpoints. Specify a value from -0.5 (left offset) to +0.5 (right offset).

Default: 0.0 (no offset)

FILL | NOFILL

specifies whether the bars are filled. The FILL option shows the fill color for the bars. The NOFILL option hides the fill color for the bars.

Default: FILL

Interactions:

Specifying FILL also hides the outlines.

If NOFILL and NOOUTLINE are both specified, then both options are ignored.

FILLATTRS= *style-element* | (COLOR= *color*)

specifies the appearance of the fill for the bars. You can specify the color of the fill by using a style element or by using the COLOR= suboption. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphDataDefault style element in the current style. For grouped data, the default color is specified by the Color attribute of the GraphData1... GraphData*n* style elements in the current style.

Interaction: This option has no effect if you specify the NOFILL option.

FREQ= *numeric-variable*

specifies how many times observations are repeated for computational purposes. specifies that each observation is repeated *n* times for computational purposes, where *n* is the value of the numeric variable. If *n* is not an integer, then it is truncated to an integer. If *n* is less than 1 or missing, then it is excluded from the analysis.

Interaction: If your plot is overlaid with other categorization plots, then the first FREQ variable that you specified is used for all of the plots.

GROUP= *variable*

specifies a variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes.

GROUPDISPLAY= STACK | CLUSTER

specifies how to display grouped bars.

STACK

groups are overlaid without any clustering. Each group is represented by unique visual attributes derived from the GraphData1... GraphData*n* style elements in the current style.

CLUSTER

displays group values as separate adjacent bars that replace the single category bar. Each set of group values is centered at the midpoint tick mark for the category.

Default: STACK

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

GROUPORDER= ASCENDING | DESCENDING | DATA

specifies the ordering of bars within a group.

ASCENDING

displays each group in ascending group value.

DESCENDING

displays each group in descending group value.

DATA

displays each group in data order of the group variable.

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

LEGENDLABEL= *“text-string”*

specifies the label that identifies the bar chart in the legend. By default, the label of the RESPONSE= variable is used. If there is no response variable label, the name of the response variable and the computed statistic (SUM or MEAN) is used. If the RESPONSE= option is not used, the legend label is “Frequency”.

Interaction: The LEGENDLABEL= option has no effect if you also specify the GROUP= option in the same plot statement.

LIMITATTRS= *style-element* <(options)> | (options)

specifies the appearance of the limit lines in the plot. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: The default color is specified by the ContrastColor attribute of the GraphError style element in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see [“Line Patterns” on page 663](#).

Default: The default line pattern is specified by the LineStyle attribute of the GraphError style element in the current style.

THICKNESS= *n* <units>

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see [“Units of Measurement” on page 659](#).

Default: The default line thickness is specified by the LineThickness attribute of the GraphError style element in the current style.

LIMITS= BOTH | LOWER | UPPER

specifies which limit lines to display. Limits are displayed as heavier line segments with a serif at the end extending from each bar. Upper limits extend to the right of the bar and lower limits extend to the left of the bar. By default, no limits are displayed unless you specify either the LIMITS= or LIMITSTAT= option. If you specify the LIMITSTAT= option only, then LIMITS=BOTH is the default. Specify one of the following values:

BOTH

adds lower and upper limit lines to the plot.

LOWER

adds lower limit lines to the plot.

UPPER

adds upper limit lines to the plot.

By default, no limit lines are displayed. However, if you specify the **LIMITSTAT=** option, then the default is **BOTH**.

Interactions:

Limit lines are displayed only when you specify **STAT= MEAN**.

The **LIMITS=** option has no effect if you also specify the **GROUP=** option in the plot statement.

LIMITSTAT= CLM | STDDEV | STDERR

specifies the statistic for the limit lines. Specify one of the following statistics:

CLM

confidence limits

STDDEV

standard deviation

STDERR

standard error

Default: CLM**Interactions:**

If you specify the **LIMITSTAT=** option only, then the default value for the **LIMITS=** option is **BOTH**.

Limits lines are displayed only when you specify **STAT=MEAN**.

The **LIMITSTAT=** option has no effect if you also specify the **GROUP=** option in the plot statement.

MISSING

processes missing values as a valid category value and creates a bar for it.

NAME= "text-string"

specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note: The name is case-sensitive. When you refer to the specified name in other statements, you must use the same exact capitalization.

NOSTATLABEL

removes the statistic name from the axis and legend labels.

NUMSTD= *n*

specifies the number of standard units for the limit lines, when you specify **LIMITSTAT= STDDEV** or **LIMITSTAT= STDERR**. You can specify any positive number, including decimals.

Default: 1**OUTLINE | NOOUTLINE**

specifies whether the bars have outlines. The **OUTLINE** option shows the outlines. The **NOOUTLINE** option hides the outlines.

Default: OUTLINE**Interactions:**

Specifying **OUTLINE** also hides the fill color.

If **NOOUTLINE** and **NOFILL** are both specified, then both options are ignored.

RESPONSE= response-variable

specifies a numeric response variable for the plot. The summarized values of the response variable are displayed on the vertical axis.

STAT= FREQ | MEAN | SUM

specifies the statistic for the vertical axis. Specify one of the following statistics:

FREQ

the frequencies for the category variable. This is the default value when you do not specify the RESPONSE= option.

MEAN

the mean of the response variable.

SUM

the sum of the response variable. This is the default value when you specify the RESPONSE= option.

Interaction: If you do not specify the RESPONSE= option, then only FREQ can be used. If you specify RESPONSE=, then you can use either SUM or MEAN.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the bars and limits, if displayed. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default: 0.0

URL= *character-variable*

specifies a character variable that contains URLs for Web pages to be displayed when parts of the plot are selected within an HTML page.

Default: By default, no HTML links are created.

Interactions:

This option affects graphics output that is created through the ODS HTML destination only. For more information about ODS destinations, see [“Specifying a Destination” on page 649](#).

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For more information, see [“Using the ODS GRAPHICS Statement” on page 651](#).

WEIGHT= *numeric-variable*

specifies how observations are weighted. Each observation is weighted by a factor of w for computational purposes, where w is the value of the numeric variable. w can be any numeric value. If w is 0, negative or missing, then that observation is excluded from the analysis.

Interaction: If your plot is overlaid with other categorization plots, then the first WEIGHT variable that you specified is used for all of the plots.

VBARPARM Statement

Creates a vertical bar chart based on a pre-summarized response value for each unique value of the category variable. You can also assign variables to the upper and lower limits.

Requirement: The data must contain only one response value per unique category variable. If more than one value is found, a warning is written to the SAS log, and the graph might produce unpredictable results.

Interactions: The HBARPARM statement can be combined only with other basic plot statements in the SGPLOT procedure. See [“Plot Type Compatibility” on page 277](#).
When used with particular styles, the VBARPARM statement produces fill patterns for grouped bars. For more information about fill patterns, see [“Using Fill Patterns to Distinguish Grouped Bar Charts” on page 636](#).

Note: An important distinction between VBARPARM and VBAR is that the response variable is required for VBARPARM. In addition, the response variable should contain pre-summarized computed values such as a sum or a mean.

Example: [“About Bar Charts” on page 42](#)

Syntax

VBARPARM CATEGORY= *category-variable* RESPONSE= *numeric-variable* </option(s)>;

Summary of Optional Arguments

Bar options

BARWIDTH= *numeric-value*

specifies the width of the bars as a ratio of the maximum possible width.

DATASKIN= NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN

specifies a special effect to be used on all filled bars.

DISCRETEOFFSET= *numeric-value*

specifies an amount to offset all bars from the category midpoints.

FILL | **NOFILL**

specifies whether the bars are filled.

FILLATTRS= *style-element* | (COLOR= *color*)

specifies the appearance of the fill for the bars.

LIMITATTRS= *style-element* <(options)> | (options)

specifies the appearance of the limit lines in the plot.

LIMITLOWER= *numeric-variable*

specifies values for the lower endpoints on the limit lines.

LIMITUPPER= *numeric-variable*

specifies values for the upper endpoints on the limit lines.

MISSING

processes missing values as a valid category value and creates a bar for it.

OUTLINE | **NOOUTLINE**

specifies whether the bars have outlines.

Group options

CLUSTERWIDTH= *numeric-value*

specifies the cluster width as a ratio of the maximum width.

GROUP= *variable*

specifies a variable that is used to group the data.

GROUPORDER= ASCENDING | DESCENDING | DATA

specifies the ordering of bars within a group.

Label options

DATALABEL <= *variable*>

displays a label for each data point.

DATALABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the DATALABEL= option.

DATALABELPOS= DATA | BOTTOM | TOP

specifies the location of the data label.

Plot options**ATTRID=** *variable*

specifies the value of the ID variable in an attribute map data set.

LEGENDLABEL= *“text-string”*

specifies the label that identifies the bar chart in the legend.

NAME= *“text-string”*

specifies a name for the plot.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the bars and limits, if displayed.

URL= *character-variable*

specifies a character variable that contains URLs for Web pages to be displayed when parts of the plot are selected within an HTML page.

Required Arguments**CATEGORY=** *category-variable*

specifies the variable that categorizes the data. All values are treated as discrete values. The input data for this variable should contain unique values. When the category values are not unique, a warning is logged, and multiple bars are superimposed at the duplicated category values.

The CATEGORY axis is always discrete.

RESPONSE= *numeric-variable*

specifies a numeric response variable. The input data is expected to be pre-summarized computed values (sum, mean, and so on).

Optional Arguments**ATTRID=** *variable*specifies the value of the ID variable in an attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph. For more information, see [“Using SG Attribute Maps to Control Visual Attributes” on page 605](#).**BARWIDTH=** *numeric-value*

specifies the width of the bars as a ratio of the maximum possible width. The maximum width is equal to the distance between the center of each bar and the centers of the adjacent bars. Specify a value from 0.0 (narrowest) to 1.0 (widest).

For example, if you specify a width of **1**, then there is no space between the bars. If you specify a width of **.5**, then the width of the bars is equal to the space between the bars.**Default:** .8

1.0 when the GROUP option is specified and GROUPDISPLAY=CLUSTER

Interaction: When the GROUP option is specified, the bar width is determined by the maximum number of bars in any one group cluster. All bars are drawn with the same width, with the cluster being positioned symmetrically around the midpoint.**CLUSTERWIDTH=** *numeric-value*

specifies the cluster width as a ratio of the maximum width. Specify a value from 0.0 (narrowest) to 1.0 (widest).

CLUSTERWIDTH is the fraction of the midpoint spacing used by all bars that are clustered around a midpoint (category value). The bar width is applied to the maximum bar spacing divided by the maximum number of bars in any one cluster.

Default: 0.8

Interaction: This option is applicable only when the GROUP option is specified and when GROUPDISPLAY=CLUSTER.

DATALABEL <= *variable*>

displays a label for each data point. If you specify a variable, then the values of that variable are used for the data labels. If you do not specify a variable, then the values of the calculated response are used for the data labels.

DATALABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the DATALABEL= option. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

Here is an example that specifies a style element:

DATALABELATTRS=GraphLabelText

options can be one or more of the following:

COLOR= *color*

specifies the color of the label characters. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphDataText style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

FAMILY= “*font-family*”

specifies the font family for the label characters. The SAS ODS styles use TrueType system fonts. For more information, see “TrueType Fonts Supplied by SAS” in Chapter 15 of *SAS Language Reference: Concepts*.

Default: The default font family is specified by the FontFamily attribute of the GraphDataText style element in the current style.

SIZE= *n* <*units*>

specifies the font size of the label characters. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default font size is specified by the FontSize attribute of the GraphDataText style element in the current style.

STYLE= ITALIC | NORMAL

specifies whether the label characters are italic (ITALIC) or normal (NORMAL).

Default: The default font style is specified by the FontStyle attribute of the GraphDataText style element in the current style.

WEIGHT= BOLD | NORMAL

specifies whether the label characters are bold (BOLD) or normal (NORMAL).

Default: The default font weight is specified by the FontWeight attribute of the GraphDataText style element in the current style.

Here is an example expression:

DATALABELATTRS=(Color=Green Family=Arial Size=8 Style=Italic Weight=Bold)

Default: The default style element is GraphDataText.

Interaction: This option has no effect unless the DATALABEL option is also specified.

DATALABELPOS= DATA | BOTTOM | TOP

specifies the location of the data label. Specify one of the following values:

DATA

places the label on the data primitives (at the upper boundary of the bars).

BOTTOM

places the label below the bars.

TOP

places the label above the bars.

Default: DATA

Interactions:

This option displays limit information when limits are specified. When limits are specified, the default data label position is BOTTOM.

This option displays group values for each category when GROUP= is also specified.

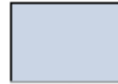
This option displays response values for each overlaid chart.

This option has no effect unless you also specify the DATALABEL option.

DATASKIN= NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN

specifies a special effect to be used on all filled bars. Specify one of the following (the samples were generated using the HTMLBlue style):

NONE



CRISP



GLOSS



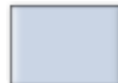
MATTE



PRESSED



SHEEN



Default: NONE

Interactions:

This option has no effect if the applied style uses a fill pattern and FILL is disabled for the style. If the style has both FILL and FILLPATTERN enabled, data skins will work. For more information about fill patterns, see [“Using Fill Patterns to Distinguish Grouped Bar Charts” on page 636](#). For instructions about viewing the contents of style templates, see [“About Styles and Style Elements” on page 620](#).

This option has no effect if you also specify NOFILL.

DISCRETEOFFSET= *numeric-value*

specifies an amount to offset all bars from the category midpoints. Specify a value from -0.5 (left offset) to +0.5 (right offset).

Default: 0.0 (no offset)

FILL | NOFILL

specifies whether the bars are filled. The FILL option shows the fill color for the bars. The NOFILL option hides the fill color for the bars.

Default: FILL

Interactions:

Specifying FILL also hides the outlines.

If NOFILL and NOOUTLINE are both specified, then both options are ignored.

FILLATTRS= *style-element* | (COLOR= *color*)

specifies the appearance of the fill for the bars. You can specify the color of the fill by using a style element or by using the COLOR= suboption. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see [“Color-Naming Schemes” in Chapter 14 of SAS/GRAPH: Reference](#).

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphDataDefault style element in the current style. For grouped data, the default color is specified by the Color attribute of the GraphData1... GraphData*n* style elements in the current style.

Interaction: This option has no effect if you specify the NOFILL option.

GROUP= *variable*

specifies a variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes.

GROUPORDER= ASCENDING | DESCENDING | DATA

specifies the ordering of bars within a group.

ASCENDING

displays each group in ascending group value.

DESCENDING

displays each group in descending group value.

DATA

displays each group in data order of the group variable.

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

LEGENDLABEL= “*text-string*”

specifies the label that identifies the bar chart in the legend. By default, the label of the RESPONSE= variable is used. If there is no response variable label, the name of the response variable is used.

Interaction: The LEGENDLABEL= option has no effect if you also specify the GROUP= option in the same plot statement.

LIMITATTRS= *style-element* <(options)> | (options)

specifies the appearance of the limit lines in the plot. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: The default color is specified by the ContrastColor attribute of the GraphError style element in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see “Line Patterns” on page 663.

Default: The default line pattern is specified by the LineStyle attribute of the GraphError style element in the current style.

THICKNESS= *n* <units>

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default line thickness is specified by the LineThickness attribute of the GraphError style element in the current style.

LIMITLOWER= *numeric-variable*

specifies values for the lower endpoints on the limit lines. Limit lines are displayed as line segments with a serif at the end.

Default: The lower segments of the limit lines are not displayed. (Limit lines are displayed only if either LIMITLOWER= or LIMITUPPER= is specified.)

Interactions:

If LIMITUPPER= is also specified, then the plot displays the lower and upper segments of the limit lines.

This option is ignored if GROUP= is specified and GROUPDISPLAY=STACK.

The appearance of the limit lines can be controlled by the LIMITATTRS= option.

LIMITUPPER= *numeric-variable*

specifies values for the upper endpoints on the limit lines. Limit lines are displayed as line segments with a serif at the end.

Default: The upper segments of the limit lines are not displayed. (Limit lines are displayed only if either LIMITLOWER= or LIMITUPPER= is specified.)

Interactions:

If LIMITLOWER= is also specified, then the plot displays the lower and upper segments of the limit lines.

This option is ignored if GROUP= is specified and GROUPDISPLAY=STACK.

The appearance of the limit lines can be controlled by the LIMITATTRS= option.

MISSING

processes missing values as a valid category value and creates a bar for it.

NAME= “*text-string*”

specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note: The name is case-sensitive. When you refer to the specified name in other statements, you must use the same exact capitalization.

OUTLINE | NOOUTLINE

specifies whether the bars have outlines. The OUTLINE option shows the outlines. The NOOUTLINE option hides the outlines.

Default: OUTLINE

Interactions:

Specifying OUTLINE also hides the fill color.

If NOOUTLINE and NOFILL are both specified, then both options are ignored.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the bars and limits, if displayed. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default: 0.0

URL= *character-variable*

specifies a character variable that contains URLs for Web pages to be displayed when parts of the plot are selected within an HTML page.

Default: By default, no HTML links are created.

Interactions:

This option affects graphics output that is created through the ODS HTML destination only. For more information about ODS destinations, see [“Specifying a Destination” on page 649](#).

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For more information, see [“Using the ODS GRAPHICS Statement” on page 651](#).

VBOX Statement

Creates a vertical box plot that shows the distribution of your data.

Interaction: The VBOX statement cannot be used together with other plot statements in the SGPLOT procedure. Box plots can be overlaid with other box plots. However, overlaid box plots must have the same category variables.

Example: [“About Box Plots” on page 37](#)

Syntax

VBOX *analysis-variable* </ option(s)>;

Summary of Optional Arguments

Box options

BOXWIDTH= *numeric-value*

specifies the width of the box.

CAPSHAPE= BRACKET | LINE | SERIF

specifies the shape of the whisker cap lines.

CONNECT= MEAN | MEDIAN | Q1 | Q3 | MIN | MAX

specifies that a connect line joins a statistic from box to box.

CONNECTATTRS= *style-element* <(options)> | (options)

specifies the appearance of the lines that connect multiple boxes.

DISCRETEOFFSET= *numeric-value*

specifies an amount to offset all boxes from the discrete tick marks.

EXTREME

specifies that the whiskers can extend to the maximum and minimum values for the analysis variable, and that outliers are not identified.

FILL | NOFILL

specifies whether the boxes are filled with color.

FILLATTRS= *style-element* | (COLOR= *color*)

specifies the appearance of the fill for the boxes.

LINEATTRS= *style-element* <(options)> | (options)

specifies the appearance of the box outlines.

MEANATTRS= *style-element* <(options)> | (options)

specifies the appearance of the marker that represents the mean in the box.

MEDIANATTRS= *style-element* <(options)> | (options)

specifies the appearance of the line that represents the median.

NOCAPS

hides the cap lines for the whiskers.

NOMEAN

hides the mean marker.

NOMEDIAN

hides the median line.

NOOUTLIERS

hides the outliers from the plot.

NOTCHES

specifies that the boxes be notched.

OUTLIERATTRS= *style-element* <(options)> | (options)

specifies the appearance of the marker that represents the outliers.

WHISKERATTRS= *style-element* <(options)> | (options)

specifies the appearance of the whisker and cap lines.

Group options

CLUSTERWIDTH= *numeric-value*

specifies the cluster width as a ratio of the midpoint spacing.

GROUP= *variable*

specifies a variable that is used to group the data.

GROUPDISPLAY= CLUSTER | OVERLAY

specifies how to display grouped boxes.

GROUPORDER= ASCENDING | DESCENDING | DATA

specifies the ordering of boxes within a group.

Label options

DATALABEL <= *variable*>

adds data labels for the outlier markers.

DATALABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the DATALABEL= option.

LABELFAR

specifies that only the far outliers have data labels.

Plot options

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set.

CATEGORY= *category-variable*

specifies the category variable for the plot.

FREQ= *numeric-variable*

specifies how many times observations are repeated for computational purposes.

LEGENDLABEL= *“text-string”*

specifies a label that identifies the box plot in the legend.

MISSING

processes missing values as a valid category value and creates a box for it.

NAME= *“text-string”*

specifies a name for the plot.

PERCENTILE= 1 | 2 | 3 | 4 | 5

specifies a method for computing the percentiles for the plot.

SPREAD

relocates outlier points that have identical values to prevent overlapping.

TRANSPARENCY= *value*

specifies the degree of transparency for the plot.

X2AXIS

assigns the category variable to the secondary (top) horizontal axis.

Y2AXIS

assigns the response variable to the secondary (right) vertical axis.

Required Argument

analysis-variablespecifies the analysis variable for the plot. If you do not specify the **CATEGORY=** option, then one box is created for the analysis variable.

Optional Arguments

ATTRID= *variable*specifies the value of the ID variable in an attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph. For more information, see [“Using SG Attribute Maps to Control Visual Attributes” on page 605](#).**BOXWIDTH=** *numeric-value*

specifies the width of the box. Specify a value between 0.0 (0% of the available width) and 1.0 (100% of the available width).

Default: 0.4When **GROUP** is specified, the default box width is 0.6.**CAPSHAPE=** **BRACKET** | **LINE** | **SERIF**

specifies the shape of the whisker cap lines. Specify one of the following values:

BRACKET

displays a straight line with brackets.

LINE

displays a straight line.

SERIF

displays a short straight line.

CATEGORY= *category-variable*

specifies the category variable for the plot. A box plot is created for each distinct value of the category variable.

If you explicitly set the category axis type to LINEAR and use a numeric category variable, the box plot becomes an interval plot. Otherwise, the box plot is discrete.

CLUSTERWIDTH= *numeric-value*

specifies the cluster width as a ratio of the midpoint spacing. Specify a value from 0.1 (narrowest) to 1.0 (widest).

Default: 0.7

Interaction: This option is applicable only when a GROUP is in effect and the category axis is discrete.

CONNECT= MEAN | MEDIAN | Q1 | Q3 | MIN | MAX

specifies that a connect line joins a statistic from box to box.

Interaction: This option applies only when the CATEGORY option is used to generate multiple boxes.

Tip: You can use the CONNECTATTRS option to specify attributes for the connect line.

CONNECTATTRS= *style-element* <(options)> | (options)

specifies the appearance of the lines that connect multiple boxes. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

Here is an example that specifies a style element: **CONNECTATTRS=GraphData3**

options can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphConnectLine style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see “Line Patterns” on page 663.

Default: For ungrouped data, the default line pattern is specified by the LineStyle attribute of the GraphConnectLine style element in the current style. For grouped data, the default line pattern is specified by the LineStyle attribute of the GraphData1 ... GraphData*n* style elements in the current style.

THICKNESS= *n* <units>

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: For ungrouped data, the default line thickness is specified by the LineThickness attribute of the GraphConnectLine style element in the current style. For grouped data, the default line thickness is specified by the LineThickness attribute of the GraphData1 ... GraphData*n* style elements in the current style.

Here is an example expression:

```
CONNECTATTRS=(Color= "light green" Pattern= MediumDash Thickness= 4)
```

Default: The default style element is `GraphConnectLine`.

Interactions:

This option takes effect only if the `CONNECT=` option is also specified.

This option is ignored if the `GROUP=` option is also specified.

DATALABEL *<= variable>*

adds data labels for the outlier markers. If you specified a variable, then the values for that variable are used for the data labels. If you did not specify a variable, then the values of the analysis variable are used.

Note: This option has no effect unless the plot contains outlier points.

DATALABELATTRS= *style-element* *<(options)>* | *(options)*

specifies the appearance of the labels in the plot when you use the `DATALABEL=` option. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

Here is an example that specifies a style element:

DATALABELATTRS=`GraphLabelText`

options can be one or more of the following:

COLOR= *color*

specifies the color of the label characters. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the `Color` attribute of the `GraphDataText` style element in the current style. For grouped data, the default color is specified by the `ContrastColor` attribute of the `GraphData1 ... GraphDatan` style elements in the current style.

FAMILY= “*font-family*”

specifies the font family for the label characters. The SAS ODS styles use TrueType system fonts. For more information, see “TrueType Fonts Supplied by SAS” in Chapter 15 of *SAS Language Reference: Concepts*.

Default: The default font family is specified by the `FontFamily` attribute of the `GraphDataText` style element in the current style.

SIZE= *n* *<units>*

specifies the font size of the label characters. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default font size is specified by the `FontSize` attribute of the `GraphDataText` style element in the current style.

STYLE= `ITALIC` | `NORMAL`

specifies whether the label characters are italic (`ITALIC`) or normal (`NORMAL`).

Default: The default font style is specified by the `FontStyle` attribute of the `GraphDataText` style element in the current style.

WEIGHT= `BOLD` | `NORMAL`

specifies whether the label characters are bold (`BOLD`) or normal (`NORMAL`).

Default: The default font weight is specified by the `FontWeight` attribute of the `GraphDataText` style element in the current style.

Here is an example expression:

`DATALABELATTRS=(Color=Green Family=Arial Size=8 Style=Italic Weight=Bold)`

Default: The default style element is GraphDataText.

Interaction: This option has no effect unless the DATALABEL option is also specified.

DISCRETEOFFSET= *numeric-value*

specifies an amount to offset all boxes from the discrete tick marks.

Specify a value from -0.5 (left offset) to +0.5 (right offset). If you specify a value outside of this range, an error message appears in the SAS log and the graph is not produced.

Default: 0.0 (no offset)

EXTREME

specifies that the whiskers can extend to the maximum and minimum values for the analysis variable, and that outliers are not identified. When you do not specify the EXTREME option, the whiskers cannot be longer than 1.5 times the length of the box.

FILL | NOFILL

specifies whether the boxes are filled with color. The FILL option shows the fill color. The NOFILL option hides the fill color.

Default: FILL

FILLATTRS= *style-element* | (COLOR= *color*)

specifies the appearance of the fill for the boxes. You can specify the color of the fill by using a style element or by using the COLOR= suboption. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphDataDefault style element in the current style.

For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

Interaction: This option has no effect if you also specify the NOFILL option.

FREQ= *numeric-variable*

specifies how many times observations are repeated for computational purposes. specifies that each observation is repeated *n* times for computational purposes, where *n* is the value of the numeric variable. If *n* is not an integer, then it is truncated to an integer. If *n* is less than 1 or missing, then it is excluded from the analysis.

Interaction: If your plot is overlaid with other categorization plots, then the first FREQ variable that you specified is used for all of the plots.

GROUP= *variable*

specifies a variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes.

GROUPDISPLAY= CLUSTER | OVERLAY

specifies how to display grouped boxes.

CLUSTER

the boxes are drawn adjacent to each other.

OVERLAY

all the boxes for a given group value are drawn at the exact coordinate and might overlap. Each group is represented by unique visual attributes derived from the GraphData1 ... GraphData*n* style elements in the current style.

Default: CLUSTER for a discrete category axis

OVERLAY for a linear axis

Restriction: GROUPDISPLAY=CLUSTER can take effect only when the category axis is discrete.

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

GROUPORDER= ASCENDING | DESCENDING | DATA

specifies the ordering of boxes within a group.

ASCENDING

displays each group in ascending group value.

DESCENDING

displays each group in descending group value.

DATA

displays each group in data order of the group variable.

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

LABELFAR

specifies that only the far outliers have data labels. Far outliers are points whose distance from the box is more than three times the length of the box.

Note: This option has no effect if you do not specify the DATALABEL option, or if there are no far outliers.

LEGENDLABEL= “text-string”

specifies a label that identifies the box plot in the legend. By default, the label of the analysis variable is used.

LINEATTRS= style-element <(options)> | (options)

specifies the appearance of the box outlines. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

Here is an example that specifies a style element: **LINEATTRS=GraphData3**

For more information about style elements, see [“Style Elements for Use with ODS Graphics” on page 640](#).

options can be one or more of the following:

COLOR= color

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphDataDefault style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

PATTERN= line-pattern

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see [“Line Patterns” on page 663](#).

Default: For ungrouped data, the default line pattern is specified by the LineStyle attribute of the GraphDataDefault style element in the current style. For grouped data, the default line pattern is specified by the LineStyle attribute of the GraphData1 ... GraphData*n* style elements in the current style.

THICKNESS= n <units>

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see [“Units of Measurement” on page 659](#).

Default: For ungrouped data, the default line thickness is specified by the LineThickness attribute of the GraphDataDefault style element in the current style. For grouped data, the default line thickness is specified by the LineThickness attribute of the GraphData1 ... GraphData*n* style elements in the current style.

Here is an example expression:

```
LINEATTRS=(Color= "light green" Pattern= MediumDash Thickness= 4)
```

Default: For ungrouped data, line attributes are derived from the GraphDataDefault style element. For grouped data, the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements are used.

Interactions:

This option takes effect only if the CONNECT= option is also specified.

This option is ignored if the GROUP= option is also specified.

MEANATTRS= *style-element* <(options)> | (options)

specifies the appearance of the marker that represents the mean in the box. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the markers. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphBoxMean style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

SIZE= *n* <units>

specifies the size of the markers. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default marker size is specified by the MarkerSize attribute of the GraphBoxMean style element in the current style.

SYMBOL= *symbol-name*

specifies the symbol for the markers. For a list of valid marker symbols, see “Marker Symbols” on page 661.

Default: For ungrouped data, the default marker symbol is specified by the MarkerSymbol attribute of the GraphBoxMean style element in the current style. For grouped data, the default marker symbol is specified by the MarkerSymbol attribute of the GraphData1 ... GraphData*n* style elements in the current style.

WEIGHT= NORMAL | BOLD

specifies the marker weight of the marker.

Default: NORMAL

Default: GraphBoxMean is the default style element.

MEDIANATTRS= *style-element* <(options)> | (options)

specifies the appearance of the line that represents the median. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphBoxMedian style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see “Line Patterns” on page 663.

Default: For ungrouped data, the default line pattern is specified by the LineStyle attribute of the GraphBoxMedian style element in the current style. For grouped data, the default line pattern is specified by the LineStyle attribute of the GraphData1 ... GraphData*n* style elements in the current style.

THICKNESS= *n* *<units>*

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: For ungrouped data, the default line thickness is specified by the LineThickness attribute of the GraphBoxMedian style element in the current style. For grouped data, the default line thickness is specified by the LineThickness attribute of the GraphData1 ... GraphData*n* style elements in the current style.

Default: GraphBoxMedian is the default style element.

Interaction: This option is ignored if the NOMEDIAN option is also specified.

MISSING

processes missing values as a valid category value and creates a box for it.

NAME= “*text-string*”

specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note: The name is case-sensitive. When you refer to the specified name in other statements, you must use the same exact capitalization.

NOCAPS

hides the cap lines for the whiskers.

NOMEAN

hides the mean marker.

NOMEDIAN

hides the median line.

NOOUTLIERS

hides the outliers from the plot.

NOTCHES

specifies that the boxes be notched. The endpoints of the notches are at the following computed locations:

$$\text{median} \pm 1.58(IQR/\sqrt{N})$$

For a visual description of the parts of a box plot, see “Details” on page 446 .

OUTLIERATTRS= *style-element* <(options)> | (options)

specifies the appearance of the marker that represents the outliers. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the markers. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphOutlier style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

SIZE= *n* <units>

specifies the size of the markers. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default marker size is specified by the MarkerSize attribute of the GraphOutlier style element in the current style.

SYMBOL= *symbol-name*

specifies the symbol for the markers. For a list of valid marker symbols, see “Marker Symbols” on page 661.

Default: For ungrouped data, the default marker symbol is specified by the MarkerSymbol attribute of the GraphOutlier style element in the current style. For grouped data, the default marker symbol is specified by the MarkerSymbol attribute of the GraphData1 ... GraphData*n* style elements in the current style.

WEIGHT= NORMAL | BOLD

specifies the marker weight of the marker.

Default: NORMAL

Default: GraphOutlier is the default style element.

PERCENTILE= 1 | 2 | 3 | 4 | 5

specifies a method for computing the percentiles for the plot. For descriptions of each method, see “Calculating Percentiles” in the UNIVARIATE Procedure chapter of *Base SAS Procedures Guide: Statistical Procedures*.

Default: 5

SPREAD

relocates outlier points that have identical values to prevent overlapping.

Note: This option has no effect if your data does not contain two or more outliers with identical values for the analysis variable.

TRANSPARENCY= *value*

specifies the degree of transparency for the plot. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default: 0.0

WHISKERATTRS= *style-element* <(options)> | (options)

specifies the appearance of the whisker and cap lines. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphBoxWhisker style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see “Line Patterns” on page 663.

Default: For ungrouped data, the default line pattern is specified by the LineStyle attribute of the GraphBoxWhisker style element in the current style. For grouped data, the default line pattern is specified by the LineStyle attribute of the GraphData1 ... GraphData*n* style elements in the current style.

THICKNESS= *n* *<units>*

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: For ungrouped data, the default line thickness is specified by the LineThickness attribute of the GraphBoxWhisker style element in the current style. For grouped data, the default line thickness is specified by the LineThickness attribute of the GraphData1 ... GraphData*n* style elements in the current style.

Default: GraphBoxWhisker is the default style element.

Interaction: This option is ignored if the NOMEDIAN option is also specified.

X2AXIS

assigns the category variable to the secondary (top) horizontal axis.

Y2AXIS

assigns the response variable to the secondary (right) vertical axis.

Details

Horizontal and vertical box plots display the distribution of data by using a rectangular box and whiskers. Whiskers are lines that indicate a data range outside of the box.

Figure 7.4 Parts of a Box Plot

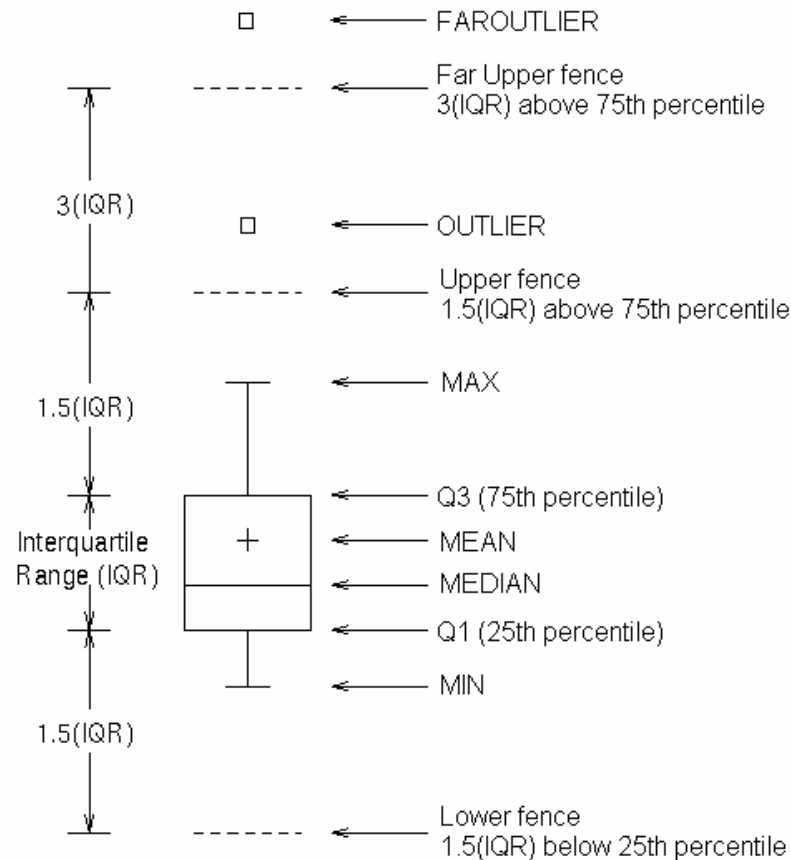


Figure 7.4 on page 447 shows a diagram of a vertical box plot. The bottom and top edges of the box indicate the intra-quartile range (IQR). That is, the range of values between the first and third quartiles (the 25th and 75th percentiles). The marker inside the box indicates the mean value. The line inside the box indicates the median value.

The elements that are outside the box are dependent on your options. By default, the whiskers that extend from each box indicate the range of values that are outside of the intra-quartile range. However, the values are close enough not to be considered outliers (a distance less than or equal to $1.5 \times IQR$). If you specify the EXTREME option, then the whiskers indicate the entire range of values, including outliers.

Any points that are a distance of more than $1.5 \times IQR$ from the box are considered to be outliers. By default, these points are indicated by markers. If you specify **DATALABEL=** option, then the outlier points have data labels. If you also specify the **LABELFAR** option, then only outliers that are $3 \times IQR$ from the box have data labels.

VECTOR Statement

Creates a vector plot that draws arrows from a point of origin to each data point.

Example: [“About Vector Plots” on page 31](#)

Syntax

VECTOR *X= numeric-variable Y= numeric-variable* *</ option(s)>*;

Summary of Optional Arguments

Group options

GROUP= *variable*

specifies a variable that is used to group the data.

Label options

DATALABEL *<= variable>*

displays a label for each data point.

DATALABELATTRS= *style-element <(options)> | (options)*

specifies the appearance of the labels in the plot when you use the DATALABEL= option.

Plot options

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set.

LEGENDLABEL= *“text-string”*

specifies a label that identifies the vector plot in the legend.

NAME= *“text-string”*

specifies a name for the plot.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the lines.

X2AXIS

assigns the X variable to the secondary (top) horizontal axis.

Y2AXIS

assigns the Y variable to the secondary (right) vertical axis.

Vector options

ARROWDIRECTION= *IN | OUT | BOTH*

specifies the location of the arrowheads for the vectors.

ARROWHEADSHAPE= *shape*

specifies the shape of the arrowheads for the vectors.

LINEATTRS= *style-element <(options)> | (options)*

specifies the appearance of the vector line.

NOARROWHEADS

removes the arrowheads from the vectors.

NOMISSINGGROUP

specifies that missing values of the group variable are not included in the plot.

XORIGIN= *numeric-value | numeric-variable*

specifies the X coordinate of the origin for the vectors.

YORIGIN= *numeric-value | numeric-variable*

specifies the Y coordinate of the origin for the vectors.

Required Arguments

X= *numeric-variable*

specifies a numeric variable for the x axis.

Y= *numeric-variable*

specifies numeric variable for the y axis.

Optional Arguments

ARROWDIRECTION= IN | OUT | BOTH

specifies the location of the arrowheads for the vectors. Specify one of the following:

IN

places the arrowheads at the origin of the vector.

OUT

places the arrowheads at the ending point of the vector.

BOTH


places arrowheads at both the origin and the ending point of the vector.

Default: OUT


ARROWHEADSHAPE= *shape*

specifies the shape of the arrowheads for the vectors. Specify one of the following:


OPEN

resembles the letter "V". 


CLOSED

an outline of a triangle. 

FILLED

a solid triangle. 

BARBED

a solid triangle with an indent at the base. 

Default: OPEN

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph. For more information, see [“Using SG Attribute Maps to Control Visual Attributes” on page 605](#).

DATALABEL <= *variable*>

displays a label for each data point. If you specify a variable, the values of that variable are used for the data labels. If you do not specify a variable, then the values of the Y variable are used for the data labels.

DATALABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the DATALABEL= option. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

Here is an example that specifies a style element:

DATALABELATTRS=GraphLabelText

options can be one or more of the following:

COLOR= *color*

specifies the color of the label characters. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphDataText style element in the current style. For grouped data, the

default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

FAMILY= *“font-family”*

specifies the font family for the label characters. The SAS ODS styles use TrueType system fonts. For more information, see “TrueType Fonts Supplied by SAS” in Chapter 15 of *SAS Language Reference: Concepts*.

Default: The default font family is specified by the FontFamily attribute of the GraphDataText style element in the current style.

SIZE= *n* *<units>*

specifies the font size of the label characters. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default font size is specified by the FontSize attribute of the GraphDataText style element in the current style.

STYLE= ITALIC | NORMAL

specifies whether the label characters are italic (ITALIC) or normal (NORMAL).

Default: The default font style is specified by the FontStyle attribute of the GraphDataText style element in the current style.

WEIGHT= BOLD | NORMAL

specifies whether the label characters are bold (BOLD) or normal (NORMAL).

Default: The default font weight is specified by the FontWeight attribute of the GraphDataText style element in the current style.

Here is an example expression:

```
DATALABELATTRS=(Color=Green Family=Arial Size=8 Style=Italic Weight=Bold)
```

Default: The default style element is GraphDataText.

Interaction: This option has no effect unless the DATALABEL option is also specified.

GROUP= *variable*

specifies a variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes.

LEGENDLABEL= *“text-string”*

specifies a label that identifies the vector plot in the legend. By default, the label of the Y variable or the group value for each plot element is used.

Interaction: The LEGENDLABEL= option has no effect if you also specify the GROUP= option in the same plot statement.

LINEATTRS= *style-element* *<(options)>* | *(options)*

specifies the appearance of the vector line. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphDataDefault style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see [“Line Patterns” on page 663](#).

Default: For ungrouped data, the default line pattern is specified by the LineStyle attribute of the GraphDataDefault style element in the current style. For grouped data, the default line pattern is specified by the LineStyle attribute of the GraphData1 ... GraphData*n* style elements in the current style.

THICKNESS= *n* *<units>*

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see [“Units of Measurement” on page 659](#).

Default: For ungrouped data, the default line thickness is specified by the LineThickness attribute of the GraphDataDefault style element in the current style. For grouped data, the default line thickness is specified by the LineThickness attribute of the GraphData1 ... GraphData*n* style elements in the current style.

NAME= *“text-string”*

specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note: The name is case-sensitive. When you refer to the specified name in other statements, you must use the same exact capitalization.

NOARROWHEADS

removes the arrowheads from the vectors.

NOMISSINGGROUP

specifies that missing values of the group variable are not included in the plot.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the lines. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default: 0.0

X2AXIS

assigns the X variable to the secondary (top) horizontal axis.

XORIGIN= *numeric-value* | *numeric-variable*

specifies the X coordinate of the origin for the vectors. You can specify either a numeric value or a numeric variable.

Default: 0

Y2AXIS

assigns the Y variable to the secondary (right) vertical axis.

YORIGIN= *numeric-value* | *numeric-variable*

specifies the Y coordinate of the origin for the vectors. You can specify either a numeric value or a numeric variable.

Default: 0

VLINE Statement

Creates a vertical line chart (the line is horizontal). You can use the VLINE statement with the VBAR statement to create a bar-line chart.

Interaction: The VLINE statement can be combined only with other categorization plot statements in the SGPLOT procedure. See [“Plot Type Compatibility” on page 277](#) .

Examples: [“About Line Charts” on page 47](#)
[“Example 10: Creating a Bar-Line Chart” on page 509](#)

Syntax

VLINE *category-variable* < / option(s)>;

Summary of Optional Arguments

Group options

CLUSTERWIDTH= *numeric-value*

specifies the width of the group clusters as a fraction of the midpoint spacing.

GROUP= *variable*

specifies a variable that is used to group the data.

GROUPDISPLAY= CLUSTER | OVERLAY

specifies how to display grouped lines.

GROUPORDER= ASCENDING | DESCENDING | DATA

specifies the ordering of lines within a group.

Label options

CURVELABEL <=“*text-string*”>

adds a label for the line plot.

CURVELABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the CURVELABEL= option.

CURVELABELLOC= OUTSIDE | INSIDE

specifies whether the curve label is placed inside the plot axes (INSIDE) or outside of the plot axes (OUTSIDE).

CURVELABELPOS= AUTO | END | MAX | MIN | START

specifies the location of the curve label.

DATALABEL <= *variable*>

displays a label for each data point.

DATALABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the DATALABEL= option.

DATALABELPOS= DATA | BOTTOM | TOP

specifies the location of the data label.

Limit options

LIMITATTRS= *style-element* <(options)> | (options)

specifies the appearance of the limit lines in the plot.

LIMITS= BOTH | LOWER | UPPER

adds limit lines to the plot.

LIMITSTAT= CLM | STDDEV | STDERR

specifies the statistic for the limit lines.

Line options

BREAK

creates a break in the line for each missing value of the response variable.

DISCRETEOFFSET= *numeric-value*

specifies an amount to offset all lines from discrete category values.

LINEATTRS= *style-element* <(options)> | (options)

specifies the appearance of the lines in the line plot.

MARKERATTRS= *style-element* <(options)> | (options)

specifies the appearance of the markers in the plot.

MARKERS

adds markers to the plot.

MISSING

processes missing values as a valid category value and creates a line for it.

NOSTATLABEL

removes the statistic name from the axis and legend labels.

NUMSTD= *n*

specifies the number of standard units for the limit lines, when you specify **LIMITSTAT**= **STDDEV** or **LIMITSTAT**= **STDERR**.

RESPONSE= *response-variable*

specifies a numeric response variable for the plot.

STAT= **FREQ** | **MEAN** | **SUM**

specifies the statistic for the vertical axis.

Plot options

ALPHA= *numeric-value*

specifies the confidence level for the confidence limits.

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set.

CATEGORYORDER= **RESPASC** | **RESPDESC**

specifies the order in which the response values are arranged.

FREQ= *numeric-variable*

specifies how many times observations are repeated for computational purposes.

LEGENDLABEL= "*text-string*"

specifies the label that identifies the line plot in the legend.

NAME= "*text-string*"

specifies a name for the plot.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the lines and markers.

URL= *character-variable*

specifies a character variable that contains URLs for Web pages to be displayed when parts of the plot are selected within an HTML page.

WEIGHT= *numeric-variable*

specifies how observations are weighted.

X2AXIS

assigns the category variable to the secondary (top) horizontal axis.

Y2AXIS

assigns the response variable to the secondary (right) vertical axis.

Required Argument

category-variable

specifies the variable that classifies the observations into distinct subsets.

Optional Arguments

ALPHA= *numeric-value*

specifies the confidence level for the confidence limits. Specify a number between 0.00 (100% confidence) and 1.00 (0% confidence).

Default: .05

Interactions:

This option has no effect if you do not specify LIMITSTAT=CLM.

If your plot is overlaid with other categorization plots, then the first ALPHA value that you specify is used for all of the plots.

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph. For more information, see [“Using SG Attribute Maps to Control Visual Attributes” on page 605](#).

BREAK

creates a break in the line for each missing value of the response variable.

Notes:

The observation is excluded from the graph when there is a missing value for the category variable without the specification of the MISSING option. (No break occurs in the line.)

The observation is excluded from the graph when there is a missing value for the FREQ variable.

CATEGORYORDER= RESPASC | RESPDESC

specifies the order in which the response values are arranged. Specify one of the following values:

RESPASC

sorts the response values in ascending order.

RESPDESC

sorts the response values in descending order.

Default: By default, the plot is sorted in ascending order based on the category values.

Restrictions:

This option takes effect only when the plot statement specifies a response variable and the axis for that variable is numeric. If the axis is not numeric, an error is generated and a message is written to the SAS log.

Uniform scaling and response sorting cannot occur on the same axis. If the UNIFORM= option is used in the SGPLOT statement, the UNIFORM option is ignored for the sorted response axis and a note is generated in the log. The UNIFORM= option is applied to the other axes and groups if requested. Note that the UNIFORM= option can selectively apply scaling to only the X or Y axis.

Interaction: If a group variable is specified in the plot statement, the response sort option is ignored and a note is generated in the log. (You can use a BY statement in the procedure if you want to group data.)

Note: If CATEGORYORDER is specified in multiple statements, the procedure sorts by the last statement in which it is specified.

CLUSTERWIDTH= *numeric-value*

specifies the width of the group clusters as a fraction of the midpoint spacing. Specify a value from 0.0 (narrowest) to 1.0 (widest).

Default: 0.8

Interaction: This option is applicable only when the GROUP option is specified, when GROUPDISPLAY=CLUSTER, and when the category axis is discrete.

CURVELABEL <=“*text-string*”>

adds a label for the line plot. You can also specify the label text. If you do not specify a label, the label from the response variable is used.

CURVELABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the CURVELABEL= option. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

Here is an example that specifies a style element:

CURVELABELATTRS=GraphTitleText

options can be one or more of the following:

COLOR= *color*

specifies the color of the label characters. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphValueText style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

FAMILY= “*font-family*”

specifies the font family for the label characters. The SAS ODS styles use TrueType system fonts. For more information, see “TrueType Fonts Supplied by SAS” in Chapter 15 of *SAS Language Reference: Concepts*.

Default: The default font family is specified by the FontFamily attribute of the GraphValueText style element in the current style.

Restriction: You cannot specify a list of font family names.

SIZE= *n* <*units*>

specifies the font size of the label characters. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default font size is specified by the FontSize attribute of the GraphValueText style element in the current style.

STYLE= ITALIC | NORMAL

specifies whether the label characters are italic or normal.

Default: The default font style is specified by the FontStyle attribute of the GraphValueText style element in the current style.

WEIGHT= BOLD | NORMAL

specifies whether the label characters are bold or normal.

Default: The default font weight is specified by the FontWeight attribute of the GraphValueText style element in the current style.

Here is an example expression:

CURVELABELATTRS=(Color=Green Family=Arial Size=8 Style=Italic Weight=Bold)

Default: The default style element is GraphValueText.

Interaction: This option has no effect unless the CURVELABEL option is also specified.

CURVELABELLOC= OUTSIDE | INSIDE

specifies whether the curve label is placed inside the plot axes (INSIDE) or outside of the plot axes (OUTSIDE).

Default: INSIDE

CURVELABELPOS= AUTO | END | MAX | MIN | START

specifies the location of the curve label. Specify one of the following values:

AUTO

places the curve label outside the plot area near the end of the curve along unused axes whenever possible (typically Y2 or X2).

Interaction: This value takes effect only when CURVELABELLOC=OUTSIDE.

END

places the curve label at the last point on the curve.

MAX

places the label at the part of the curve closest to the maximum X axis value.

MIN

places the label at the part of the curve closest to the minimum X axis value.

START

places the curve label at the first point on the curve.

Default: END

Interactions:

This option has no effect unless the CURVELABEL option is also specified.

The START and END suboptions take effect only when CURVELABELLOC=INSIDE.

DATALABEL <= variable>

displays a label for each data point. If you specify a variable, then the values of that variable are used for the data labels. If you do not specify a variable, then the values of the calculated response are used for the data labels.

DATALABELATTRS= style-element <(options)> | (options)

specifies the appearance of the labels in the plot when you use the DATALABEL= option. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

Here is an example that specifies a style element:

DATALABELATTRS=GraphLabelText

options can be one or more of the following:

COLOR= color

specifies the color of the label characters. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphDataText style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

FAMILY= *“font-family”*

specifies the font family for the label characters. The SAS ODS styles use TrueType system fonts. For more information, see “TrueType Fonts Supplied by SAS” in Chapter 15 of *SAS Language Reference: Concepts*.

Default: The default font family is specified by the FontFamily attribute of the GraphDataText style element in the current style.

SIZE= *n* *<units>*

specifies the font size of the label characters. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default font size is specified by the FontSize attribute of the GraphDataText style element in the current style.

STYLE= ITALIC | NORMAL

specifies whether the label characters are italic (ITALIC) or normal (NORMAL).

Default: The default font style is specified by the FontStyle attribute of the GraphDataText style element in the current style.

WEIGHT= BOLD | NORMAL

specifies whether the label characters are bold (BOLD) or normal (NORMAL).

Default: The default font weight is specified by the FontWeight attribute of the GraphDataText style element in the current style.

Here is an example expression:

```
DATALABELATTRS=(Color=Green Family=Arial Size=8 Style=Italic Weight=Bold)
```

Default: The default style element is GraphDataText.

Interaction: This option has no effect unless the DATALABEL option is also specified.

DATALABELPOS= DATA | BOTTOM | TOP

specifies the location of the data label. Specify one of the following values:

DATA

places the label at the data points of the lines.

BOTTOM

places the label below the lines.

TOP

places the label above the lines.

Default: DATA

Interactions:

This option displays limit information when limits are specified. When limits are specified, the default data label position is BOTTOM.

This option displays group values for each category when GROUP= is also specified.

This option displays response values for each overlaid chart.

This option has no effect unless you also specify the DATALABEL option.

DISCRETEOFFSET= *numeric-value*

specifies an amount to offset all lines from discrete category values. Specify a value from -0.5 (left offset) to +0.5 (right offset).

Default: 0.0 (no offset)

Requirement: This option is applicable only when the category axis is discrete.

FREQ= *numeric-variable*

specifies how many times observations are repeated for computational purposes. specifies that each observation is repeated n times for computational purposes, where n is the value of the numeric variable. If n is not an integer, then it is truncated to an integer. If n is less than 1 or missing, then it is excluded from the analysis.

Interaction: If your plot is overlaid with other categorization plots, then the first FREQ variable that you specified is used for all of the plots.

GROUP= *variable*

specifies a variable that is used to group the data. A separate plot is created for each unique value of the grouping variable. The plot elements for each group value are automatically distinguished by different visual attributes.

Interaction: If you specify more than one categorization plot statement, then all of the plots must specify the same GROUP variable. If you do not specify the same GROUP= option for all of the categorization plots, then the GROUP= option has no effect.

GROUPDISPLAY= CLUSTER | OVERLAY

specifies how to display grouped lines.

CLUSTER

grouped items are drawn adjacent to each other.

OVERLAY

grouped items are drawn at the exact coordinate and might overlap. Each group is represented by unique visual attributes derived from the GraphData1... GraphData n style elements in the current style.

Default: OVERLAY

Restriction: GROUPDISPLAY=CLUSTER can take effect only when the category axis is discrete. If necessary, use a TYPE=DISCRETE option on the axis statement.

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

GROUPORDER= ASCENDING | DESCENDING | DATA

specifies the ordering of lines within a group.

ASCENDING

displays each group in ascending group value.

DESCENDING

displays each group in descending group value.

DATA

displays each group in data order of the group variable.

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

LEGENDLABEL= "*text-string*"

specifies the label that identifies the line plot in the legend. By default, the label of the response variable is used. If there is no response variable label, then the name of the response variable and the computed statistic (SUM or MEAN) are used. If you do not specify a response variable, then the legend label is "Frequency".

Interaction: The LEGENDLABEL= option has no effect if you also specify the GROUP= option in the same plot statement.

LIMITATTRS= *style-element* <(options)> | (options)

specifies the appearance of the limit lines in the plot. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: The default color is specified by the ContrastColor attribute of the GraphError style element in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see “Line Patterns” on page 663.

Default: The default line pattern is specified by the LineStyle attribute of the GraphError style element in the current style.

THICKNESS= *n* *<units>*

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default line thickness is specified by the LineThickness attribute of the GraphError style element in the current style.

LIMITS= BOTH | LOWER | UPPER

adds limit lines to the plot. Specify one of the following values:

BOTH

adds lower and upper limit lines to the plot.

LOWER

adds lower limit lines to the plot.

UPPER

adds upper limit lines to the plot.

By default, no limit lines are displayed. However, if you specify the LIMITSTAT= option, then the default is BOTH.

Interaction: Limit lines are displayed only when you specify STAT= MEAN.

LIMITSTAT= CLM | STDDEV | STDERR

specifies the statistic for the limit lines. Specify one of the following statistics:

CLM

confidence limits

STDDEV

standard deviation

STDERR

standard error

Default: CLM

Interaction: If you specify the LIMITSTAT= option, then the default value for the LIMITS= option is BOTH.

LINEATTRS= *style-element* *<(options)>* | *(options)*

specifies the appearance of the lines in the line plot. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphDataDefault style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see “Line Patterns” on page 663.

Default: For ungrouped data, the default line pattern is specified by the LineStyle attribute of the GraphDataDefault style element in the current style. For grouped data, the default line pattern is specified by the LineStyle attribute of the GraphData1 ... GraphData*n* style elements in the current style.

THICKNESS= *n*<*units*>

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: For ungrouped data, the default line thickness is specified by the LineThickness attribute of the GraphDataDefault style element in the current style. For grouped data, the default line thickness is specified by the LineThickness attribute of the GraphData1 ... GraphData*n* style elements in the current style.

MARKERATTRS= *style-element* <(*options*)> | (*options*)

specifies the appearance of the markers in the plot. This option has no effect unless you also specify the MARKERS option. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the markers. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphDataDefault style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

SIZE= *n* <*units*>

specifies the size of the markers. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default marker size is specified by the MarkerSize attribute of the GraphDataDefault style element in the current style.

SYMBOL= *symbol-name*

specifies the symbol for the markers. For a list of valid marker symbols, see “Marker Symbols” on page 661.

Default: For ungrouped data, the default marker symbol is specified by the MarkerSymbol attribute of the GraphDataDefault style element in the current style. For grouped data, the default marker symbol is specified by the

MarkerSymbol attribute of the GraphData1 ... GraphData*n* style elements in the current style.

MARKERS

adds markers to the plot.

MISSING

processes missing values as a valid category value and creates a line for it.

NAME= “text-string”

specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note: The name is case-sensitive. When you refer to the specified name in other statements, you must use the same exact capitalization.

NOSTATLABEL

removes the statistic name from the axis and legend labels.

NUMSTD= *n*

specifies the number of standard units for the limit lines, when you specify LIMITSTAT= STDDEV or LIMITSTAT= STDERR. You can specify any positive number, including decimals.

Default: 1

RESPONSE= *response-variable*

specifies a numeric response variable for the plot. The summarized values of the response variable are displayed on the vertical axis.

STAT= FREQ | MEAN | SUM

specifies the statistic for the vertical axis. Specify one of the following statistics:

FREQ

the frequencies for the category variable. This is the default value when you do not specify the RESPONSE= option.

MEAN

the mean of the response variable.

SUM

the sum of the response variable. This is the default value when you specify the RESPONSE= option.

Interaction: If you do not specify the RESPONSE= option, then only FREQ can be used. If you specify RESPONSE=, then you can use either SUM or MEAN.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the lines and markers. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default: 0.0

URL= *character-variable*

specifies a character variable that contains URLs for Web pages to be displayed when parts of the plot are selected within an HTML page.

Default: By default, no HTML links are created.

Interactions:

This option affects graphics output that is created through the ODS HTML destination only. For more information about ODS destinations, see [“Specifying a Destination” on page 649](#).

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For more information, see [“Using the ODS GRAPHICS Statement” on page 651](#).

WEIGHT= *numeric-variable*

specifies how observations are weighted. Each observation is weighted by a factor of w for computational purposes, where w is the value of the numeric variable. w can be any numeric value. If w is 0, negative or missing, then that observation is excluded from the analysis.

Interaction: If your plot is overlaid with other categorization plots, then the first WEIGHT variable that you specified is used for all of the plots.

X2AXIS

assigns the category variable to the secondary (top) horizontal axis.

Y2AXIS

assigns the response variable to the secondary (right) vertical axis.

WATERFALL (Preproduction) Statement

Creates a waterfall chart computed from input data. In the chart, bars represent an initial value of Y and a series of intermediate values identified by X leading to a final value of Y.

Example: [“About Waterfall Charts \(Preproduction\)” on page 49](#)

Syntax

WATERFALL CATEGORY=*variable* RESPONSE= *numeric-variable* </ option(s)>;

Summary of Optional Arguments

Bar options

BARWIDTH= *numeric-value*

specifies the width of the bars as a ratio of the maximum possible width.

COLORGROUP= *variable*

specifies a variable that is used to discretely color the bars.

DATASKIN= NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN

specifies a special effect to be used on all filled bars.

FILL | NOFILL

specifies whether the bars are filled.

FILLATTRS= *style-element* | (COLOR= *color*)

specifies the color of the fill for the bars.

FINALBARATTRS= *style-element* | (COLOR= *color*)

specifies the appearance of the final bar, if displayed.

FINALBARTICKVALUE= “*text-string*”

specifies the tick value used on the X axis for the final bar, if displayed.

INITIALBARATTRS= *style-element* | (COLOR= *color*)

specifies the appearance of the initial bar, if displayed.

INITIALBARTICKVALUE= “*text-string*”

specifies the tick value used on the X axis for the initial bar, if displayed.

INITIALBARVALUE= *number*

specifies the value of the initial bar.

MISSING

processes missing values as a valid category value and creates a bar for it.

OUTLINE | NOOUTLINE

specifies whether the bars have outlines.

STAT= MEAN | SUM

specifies the statistic for the response axis.

Label options

DATALABEL <= *variable*>

displays a label for each data point.

DATALABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the labels in the plot when you use the DATALABEL= option.

Plot options

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set.

NAME= “*text-string*”

specifies a name for the plot.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the bars.

URL= *character-variable*

specifies a character variable that contains URLs for Web pages to be displayed when parts of the plot are selected within an HTML page.

X2AXIS

assigns the variables that are assigned to the primary (bottom) horizontal axis to the secondary (top) horizontal axis.

Y2AXIS

assigns the variables that are assigned to the primary (left) vertical axis to the secondary (right) vertical axis.

Required Arguments

CATEGORY= *variable*

specifies the variable for the category values. Duplicated category values are summarized into a unique value. All values are treated as discrete.

RESPONSE= *numeric-variable*

specifies the variable for the response values.

Optional Arguments

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph. For more information, see [“Using SG Attribute Maps to Control Visual Attributes” on page 605](#).

BARWIDTH= *numeric-value*

specifies the width of the bars as a ratio of the maximum possible width. The maximum width is equal to the distance between the center of each bar and the centers of the adjacent bars. Specify a value from 0.0 (narrowest) to 1.0 (widest).

For example, if you specify a width of 1, then there is no space between the bars. If you specify a width of .5, then the width of the bars is equal to the space between the bars.

Default: .8

COLORGROUP= *variable*

specifies a variable that is used to discretely color the bars. Once the variable values are found, the transaction bar colors are taken from the GraphData1 ... GraphData*n* style elements in the current style. If the transaction bars are filled, then the COLOR attribute is used for the fill. The CONTRASTCOLOR attribute is used for the transaction bar outlines.

DATALABEL <= *variable*>

displays a label for each data point. If you specify a variable, then the values of that variable are used for the data labels. If you do not specify a variable, then the values of the calculated response are used for the data labels.

DATALABELATTRS= *style-element* <(*options*)> | (*options*)

specifies the appearance of the labels in the plot when you use the DATALABEL= option. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

Here is an example that specifies a style element:

```
DATALABELATTRS=GraphLabelText
```

options can be one or more of the following:

COLOR= *color*

specifies the color of the label characters. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: The default color is specified by the Color attribute of the GraphDataText style element in the current style.

FAMILY= “*font-family*”

specifies the font family for the label characters. The SAS ODS styles use TrueType system fonts. For more information, see “TrueType Fonts Supplied by SAS” in Chapter 15 of *SAS Language Reference: Concepts*.

Default: The default font family is specified by the FontFamily attribute of the GraphDataText style element in the current style.

SIZE= *n* <*units*>

specifies the font size of the label characters. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default font size is specified by the FontSize attribute of the GraphDataText style element in the current style.

STYLE= ITALIC | NORMAL

specifies whether the label characters are italic (ITALIC) or normal (NORMAL).

Default: The default font style is specified by the FontStyle attribute of the GraphDataText style element in the current style.

WEIGHT= BOLD | NORMAL

specifies whether the label characters are bold (BOLD) or normal (NORMAL).

Default: The default font weight is specified by the FontWeight attribute of the GraphDataText style element in the current style.

Here is an example expression:

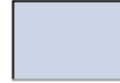
```
DATALABELATTRS=(Color=Green Family=Arial Size=8 Style=Italic Weight=Bold)
```

Default: The default style element is GraphDataText.

Interaction: This option has no effect unless the DATALABEL option is also specified.

DATASKIN= NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
specifies a special effect to be used on all filled bars. Specify one of the following (the samples were generated using the LISTING style):

NONE



CRISP



GLOSS



MATTE



PRESSED



SHEEN



Default: NONE

Interaction: This option has no effect if you specify the NOFILL option.

FILL | NOFILL

specifies whether the bars are filled. The FILL option shows the fill color for the bars. The NOFILL option hides the fill color for the bars.

Default: FILL

Interactions:

Specifying FILL also hides the outlines.

If NOFILL and NOOUTLINE are both specified, then both options are ignored.

FILLATTRS= *style-element* | (COLOR= *color*)

specifies the color of the fill for the bars.

You can specify the color of the fill by using a style element or by using the COLOR= suboption. For more information about specifying colors, see “Specifying Colors in SAS/GRAPH Programs” in Chapter 14 of *SAS/GRAPH: Reference* in the “Using Colors in SAS/GRAPH Programs” chapter in the *SAS/GRAPH: Reference*

Here is an example that specifies a style element: **FILLATTRS=GraphData3**

Here is an example that specifies a color name: **FILLATTRS= (COLOR= "Very light red")**

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphDataDefault style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

Interaction: This option has no effect if you also specify the NOFILL option.

FINALBARATTRS= *style-element* | (COLOR= *color*)

specifies the appearance of the final bar, if displayed. You can specify the color of the fill by using a style element or by using the COLOR= suboption. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: The default color is specified by the Color attribute of the GraphFinal style element in the current style.

Interaction: This option has no effect if you specify the NOFILL option.

FINALBARTICKVALUE= “*text-string*”

specifies the tick value used on the X axis for the final bar, if displayed.

Default: "Final"

INITIALBARATTRS= *style-element* | (COLOR= *color*)

specifies the appearance of the initial bar, if displayed. You can specify the color of the fill by using a style element or by using the COLOR= suboption. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: The default color is specified by the Color attribute of the GraphInitial style element in the current style.

Interactions:

This option has no effect if you specify the NOFILL option.

This option has no effect unless INITIALBARVALUE is also specified.

INITIALBARTICKVALUE= “*text-string*”

specifies the tick value used on the X axis for the initial bar, if displayed.

Default: "Initial"

Interaction: This option has no effect unless INITIALBARVALUE is also specified.

INITIALBARVALUE= *number*

specifies the value of the initial bar. When specified, the first transaction bar to be drawn starts at that initial value instead of zero.

Default: If the INITIALBARVALUE option is not specified, the initial bar is not included in the chart and the first transaction bar is drawn from zero.

MISSING

processes missing values as a valid category value and creates a bar for it.

NAME= “*text-string*”

specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note: The name is case-sensitive. When you refer to the specified name in other statements, you must use the same exact capitalization.

OUTLINE | NOOUTLINE

specifies whether the bars have outlines. The OUTLINE option shows the outlines. The NOOUTLINE option hides the outlines.

Default: OUTLINE

Interactions:

Specifying OUTLINE also hides the fill color.

If NOOUTLINE and NOFILL are both specified, then both options are ignored.

STAT= MEAN | SUM

specifies the statistic for the response axis. Specify one of the following statistics:

MEAN

the mean of the response variable.

SUM

the sum of the response variable.

Default: SUM

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the bars. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default: 0.0

URL= *character-variable*

specifies a character variable that contains URLs for Web pages to be displayed when parts of the plot are selected within an HTML page.

Default: By default, no HTML links are created.

Interactions:

This option affects graphics output that is created through the ODS HTML destination only. For more information about ODS destinations, see [“Specifying a Destination” on page 649](#).

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For more information, see [“Using the ODS GRAPHICS Statement” on page 651](#).

X2AXIS

assigns the variables that are assigned to the primary (bottom) horizontal axis to the secondary (top) horizontal axis.

Y2AXIS

assigns the variables that are assigned to the primary (left) vertical axis to the secondary (right) vertical axis.

XAXIS Statement

Specifies the axis options for the X axis. You can control the features of the axis (for example, the axis label, grid lines, and minor tick marks). You can also control the structure of the axis (for example, the data range, data type, and tick mark values).

Syntax

XAXIS *option(s)*;

Summary of Optional Arguments

Axis options

DISCRETEORDER= DATA | FORMATTED | UNFORMATTED

specifies the order in which discrete tick values are placed on the axis.

DISPLAY= ALL | NONE | (*options*)

specifies which features of the axis are displayed.

FITPOLICY= *policy-value*

specifies the method that is used to fit tick mark values on a horizontal axis when there is not enough room to draw them normally.

GRID

creates grid lines at each tick on the axis.

INTEGER

specifies that only integers are used for tick mark values.

INTERVAL= *interval-value*

specifies the tick interval for a time axis.

LABEL= “*text-string*”

specifies a label for the axis.

LABELATTRS= *style-element* <(*options*)> | (*options*)

specifies the appearance of the axis labels.

LOGBASE= 2 | 10 | e

specifies the base value for the logarithmic scale.

LOGSTYLE= LINEAR | LOGEXPAND | LOGEXPONENT

specifies how to scale and format the values for the major tick marks for logarithmic axes.

MAX= *numeric-value*

specifies the maximum value for the axis.

MIN= *numeric-value*

specifies the minimum value for the axis.

MINOR

adds minor tick marks to a log or time axis.

NOTIMESPLIT

prevents a time axis from splitting the time, date, or datetime values into two rows.

OFFSETMAX= *numeric-value*

specifies an offset that follows the highest data value on the axis.

OFFSETMIN= *numeric-value*

specifies an offset that precedes the lowest data value on the axis.

REFTICKS

adds tick marks to axis that is opposite from the specified axis.

REVERSE

specifies that the tick values are displayed in reverse (descending) order.

THRESHOLDMAX= *numeric-value*

Specifies a threshold for displaying one more tick mark at the high end of the axis.

THRESHOLDMIN= *numeric-value*

Specifies a threshold for displaying one more tick mark at the low end of the axis.

TICKVALUEFORMAT= DATA | *SAS-format*

specifies the format for the axis tick values.

TYPE= DISCRETE | LINEAR | LOG | TIME

specifies the type of axis.

VALUEATTRS= *style-element* <(*options*)> | (*options*)

specifies the appearance of the axis tick value labels.

VALUES= (*values-list*)

specifies the values for the ticks on an axis.

VALUESHINT

specifies that the minimum and maximum axis values are determined independently of the values that you specify in the VALUES= option.

Optional Arguments

DISCRETEORDER= DATA | FORMATTED | UNFORMATTED

specifies the order in which discrete tick values are placed on the axis. Specify one of the following values:

DATA

places the values in the order in which they appear in the data.

FORMATTED

sorts the formatted values in ascending character order.

UNFORMATTED

sorts the unformatted values in ascending character order.

Default: UNFORMATTED

Restriction: This option affects only box plots, dot plots, bar charts, and line plots, or for any axis where TYPE=DISCRETE.

DISPLAY= ALL | NONE | (*options*)

specifies which features of the axis are displayed. ALL displays all of the features of the axis. NONE specifies that none of the features of the axis are displayed. You can also hide specific features:

NOLABEL

hides the axis label

NOLINE

hides the axis line

NOTICKS

hides the tick marks on the axis

NOVALUES

hides the tick mark values on the axis

Default: ALL

FITPOLICY= *policy-value*

specifies the method that is used to fit tick mark values on a horizontal axis when there is not enough room to draw them normally. Select one of the following values:

ROTATE

rotates the value text 45 degrees. This is the default for discrete axes.

ROTATETHIN

attempts to use ROTATE, and then THIN to fit the values.

STAGGER

shifts the values up and down.

STAGGERROTATE

attempts to use STAGGER, and then ROTATE to fit the values.

STAGGERTHIN

attempts to use STAGGER, and then THIN to fit the values.

THIN

removes some of the values from the axis. This is the default for linear and time axes.

Note: This option has no effect if used with the vertical (Y or Y2) axes; only the THIN fit policy is used for vertical axes.

GRID

creates grid lines at each tick on the axis.

INTEGER

specifies that only integers are used for tick mark values. This option affects only linear axes.

INTERVAL= *interval-value*

specifies the tick interval for a time axis. Specify one of the following values:

AUTO

determines the tick interval automatically based on the data.

SECOND

places ticks one second apart. The default tick value format is TIME8.

MINUTE

places ticks one minute apart. The default tick value format is TIME8.

HOURL

places ticks one hour apart. The default tick value format is TIME8.

DAY

places ticks one day apart. The default tick value format is DATE9.

WEEK

places ticks one week apart. The default tick value format is DATE9.

TENDAY

places ticks ten days apart. The default tick value format is DATE9.

SEMIMONTH

places ticks at the first and sixteenth day of each month. The default tick value format is DATE9.

MONTH

places ticks one month apart. The default tick value format is MONYY7.

QUARTER

places ticks three months apart. The default tick value format is YYQC6.

SEMIYEAR

places ticks six months apart. The default tick value format is MONYY7.

YEAR

places ticks one year apart. The default tick value format is YEAR4.

Default: AUTO

LABEL= "*text-string*"

specifies a label for the axis.

LABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the axis labels. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the label characters. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphLabelText style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

FAMILY= “*font-family*”

specifies the font family for the label characters. The SAS ODS styles use TrueType system fonts. For more information, see “TrueType Fonts Supplied by SAS” in Chapter 15 of *SAS Language Reference: Concepts*.

Default: The default font family is specified by the FontFamily attribute of the GraphLabelText style element in the current style.

Restriction: You cannot specify a list of font family names.

SIZE= *n* <*units*>

specifies the font size of the label characters. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default font size is specified by the FontSize attribute of the GraphValueText style element in the current style.

STYLE= ITALIC | NORMAL

specifies whether the label characters are italic or normal.

Default: The default font style is specified by the FontStyle attribute of the GraphLabelText style element in the current style.

WEIGHT= BOLD | NORMAL

specifies whether the label characters are bold or normal.

Default: The default font weight is specified by the FontWeight attribute of the GraphLabelText style element in the current style.

Interaction: This option has no effect unless CURVELABEL is also specified.

LOGBASE= 2 | 10 | *e*

specifies the base value for the logarithmic scale.

Default: 10

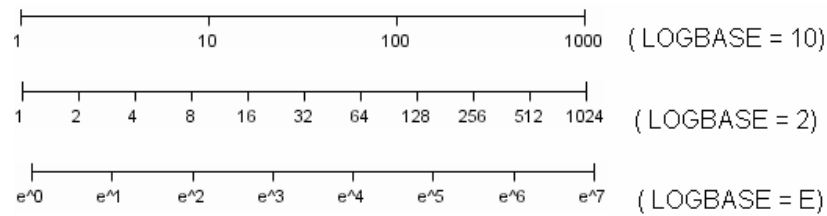
Interaction: This option has no effect unless you also specify TYPE=LOG.

LOGSTYLE= LINEAR | LOGEXPAND | LOGEXPONENT

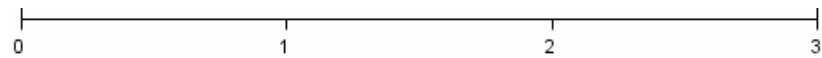
specifies how to scale and format the values for the major tick marks for logarithmic axes. Specify one of the following values:

LOGEXPAND

places the tick marks at integer powers of the base. For example, if you specified LOGBASE=2, the tick marks might be at 1, 2, 4, 8, 16. See [Figure 7.5 on page 472](#).

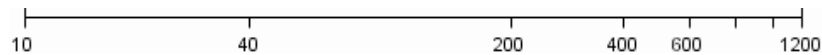
Figure 7.5 Graph Axes with LOGEXPAND**LOGEXPONENT**

places the tick marks at integer powers of the base, but identifies the values by the exponent. For example, if you specified **LOGBASE=10**, the tick marks might be at 1, 10, 100, 1000, but the tick values would read 0, 1, 2, 3. See [Figure 7.6 on page 472](#).

Figure 7.6 An Axis with LOGEXPONENT**LINEAR**

places the tick marks at uniform linear intervals, but spaces them logarithmically. In some cases an intermediate tick mark is placed between the first and second marks.

For example, if the data on this axis range from 14 to 1154, and you specify **LOGBASE=10**, then the tick marks might be at 10, 40, 200, 400, 600, 800, 1000, 1200. See [Figure 7.7 on page 472](#).

Figure 7.7 An Axis with LINEAR**Default:** LOGEXPAND

Interaction: This option has no effect unless you also specify **TYPE=LOG**.

MAX= *numeric-value*

specifies the maximum value for the axis.

Restriction: This option affects linear, log, and time axes only.

Interaction: This option has no effect if you specify the **VALUES=** option and you do not also specify the **VALUESHINT** option.

MIN= *numeric-value*

specifies the minimum value for the axis.

Restriction: This option affects linear, log, and time axes only.

Interaction: This option has no effect if you specify the **VALUES=** option and you do not also specify the **VALUESHINT** option.

MINOR

adds minor tick marks to a log or time axis.

Restriction: This option has no effect on linear or discrete axes.

Interaction: This option has no effect if you specify the **VALUES=** option.

NOTIMESPLIT

prevents a time axis from splitting the time, date, or datetime values into two rows.

Restriction: This option applies to time axes only.

OFFSETMAX= *numeric-value*

specifies an offset that follows the highest data value on the axis. Specify a value between 0 and 1.

The value represents the offset as a proportion to the total length of the axis. For a continuous axis, the offset follows the highest data value or highest tick value, whichever is greater. For a discrete axis, the offset is applied to the end of the axis farther from the origin.

Default: The offset space is determined automatically based on the data values, tick mark values, markers, and labels that are inside of the plot area.

OFFSETMIN= *numeric-value*

specifies an offset that precedes the lowest data value on the axis. Specify a value between 0 and 1.

The value represents the offset as a proportion to the total length of the axis. For a continuous axis, the offset precedes the lowest data value or lowest tick value, whichever is less. For a discrete axis, the offset is applied to the end of the axis nearer to the origin.

Default: The offset space is determined automatically based on the data values, tick mark values, markers, and labels that are inside of the plot area.

REFTICKS

adds tick marks to axis that is opposite from the specified axis. For example, if you specify the REFTICKS option in the XAXIS statement, then tick marks are added to the X2 axis.

Note: This option has no effect if the target axis already contains data.

REVERSE

specifies that the tick values are displayed in reverse (descending) order.

THRESHOLDMAX= *numeric-value*

Specifies a threshold for displaying one more tick mark at the high end of the axis.

THRESHOLDMIN= *numeric-value*

Specifies a threshold for displaying one more tick mark at the low end of the axis.

TICKVALUEFORMAT= DATA | *SAS-format*

specifies the format for the axis tick values. You can either specify a SAS format or specify DATA, which indicates that the format from the axis variable is used.

TYPE= DISCRETE | LINEAR | LOG | TIME

specifies the type of axis. Specify one of the following values:

DISCRETE

specifies an axis with discrete values. If a character variable is assigned to an axis, then the default type for that axis is discrete. In addition, all categorization plots use a discrete axis for the category variable.

LINEAR

specifies a linear scale for the axis. This is the default axis type for numeric variables, except when the data is discrete or when the numeric variable has a date or time format.

LOG

specifies a logarithmic scale for the axis. This axis type is never a default.

Interaction: A logarithmic scale cannot be used with linear regression plots (REG statement where DEGREE=1).

TIME

specifies a time scale for the axis. If the variable assigned to an axis has a time, date, or datetime format associated with it, then time is the default axis type.

VALUEATTRS= *style-element* <(options)> | (options)

specifies the appearance of the axis tick value labels. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the label characters. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphValueText style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

FAMILY= “*font-family*”

specifies the font family for the label characters. The SAS ODS styles use TrueType system fonts. For more information, see “TrueType Fonts Supplied by SAS” in Chapter 15 of *SAS Language Reference: Concepts*.

Default: The default font family is specified by the FontFamily attribute of the GraphValueText style element in the current style.

Restriction: You cannot specify a list of font family names.

SIZE= *n* <units>

specifies the font size of the label characters. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default font size is specified by the FontSize attribute of the GraphValueText style element in the current style.

STYLE= ITALIC | NORMAL

specifies whether the label characters are italic or normal.

Default: The default font style is specified by the FontStyle attribute of the GraphValueText style element in the current style.

WEIGHT= BOLD | NORMAL

specifies whether the label characters are bold or normal.

Default: The default font weight is specified by the FontWeight attribute of the GraphValueText style element in the current style.

VALUES= (*values-list*)

specifies the values for the ticks on an axis.

For values on a linear axis, the values list can be one of the following:

***value* <...*value-n*>**

creates ticks for specific values. For example, **VALUES= (0 50 100)** places tick marks at 0, 50, and 100.

value-1* TO *value-2* BY *increment-value

creates ticks for a range of values. The start of the value range is specified by *value-1* and the end of the range is specified by *value-2*. The *increment-value* specifies the interval between the ticks. For example, **VALUES= (0 to 100 by 50)** creates tick marks at 0, 50, and 100.

<value ... value-n>value-1 TO value-2 BY increment-value <value ... value-n>
 creates ticks for specific values, and also creates ticks for a range of values. The start of the value range is specified by *value-1* and the end of the range is specified by *value-2*. The *increment-value* specifies the interval between the ticks.

For example, **VALUES= (-5 10 to 50 by 20 75)** creates tick marks at -5, 10, 30, 50, and 75.

For values on a time axis, the values list can be one of the following:

value <...value-n>

creates ticks for specific values. For example, **VALUES= ("25MAY08"d "04JUL08"d "23AUG08"d)** places tick marks at 25MAY08, 04JUL08, and 23AUG08.

value-1 TO value-2 BY increment-value

creates ticks for a range of values. The start of the value range is specified by *value-1* and the end of the range is specified by *value-2*. The *increment-value* specifies the interval between the ticks. For example, **VALUES= ("01JAN08"d to "01MAY08"d by month)** creates tick marks at 01JAN08, 01FEB08, 01MAR08, 01APR08, and 01MAY08.

For a list of the interval values that you can specify, see the **INTERVAL=** option.

Restrictions:

This option has no effect on discrete and logarithmic axes.

If your **VALUES=** option creates more than 1000 values, then the option has no effect.

Interaction: If a custom format is applied to the value, the raw value is plotted first and then formatted.

VALUESHINT

specifies that the minimum and maximum axis values are determined independently of the values that you specify in the **VALUES=** option. The values from the **VALUES=** option are displayed only if they are located between the minimum and maximum values.

Interaction: This option has no effect unless you also specify the **VALUES=** option.

X2AXIS Statement

Specifies the axis options for the X2 axis. You can control the features of the axis (for example, the axis label, grid lines, and minor tick marks). You can also control the structure of the axis (for example, the data range, data type, and tick mark values).

Syntax

X2AXIS *option(s)*;

Summary of Optional Arguments

Axis options

DISCRETEORDER= [DATA](#) | [FORMATTED](#) | [UNFORMATTED](#)

specifies the order in which discrete tick values are placed on the axis.

DISPLAY= [ALL](#) | [NONE](#) | (*options*)

specifies which features of the axis are displayed.

FITPOLICY= *policy-value*

specifies the method that is used to fit tick mark values on a horizontal axis when there is not enough room to draw them normally.

GRID

creates grid lines at each tick on the axis.

INTEGER

specifies that only integers are used for tick mark values.

INTERVAL= *interval-value*

specifies the tick interval for a time axis.

LABEL= “*text-string*”

specifies a label for the axis.

LABELATTRS= *style-element* <(*options*)> | (*options*)

specifies the appearance of the axis labels.

LOGBASE= 2 | 10 | e

specifies the base value for the logarithmic scale.

LOGSTYLE= LINEAR | LOGEXPAND | LOGEXPONENT

specifies how to scale and format the values for the major tick marks for logarithmic axes.

MAX= *numeric-value*

specifies the maximum value for the axis.

MIN= *numeric-value*

specifies the minimum value for the axis.

MINOR

adds minor tick marks to a log or time axis.

NOTIMESPLIT

prevents a time axis from splitting the time, date, or datetime values into two rows.

OFFSETMAX= *numeric-value*

specifies an offset that follows the highest data value on the axis.

OFFSETMIN= *numeric-value*

specifies an offset that precedes the lowest data value on the axis.

REFTICKS

adds tick marks to axis that is opposite from the specified axis.

REVERSE

specifies that the tick values are displayed in reverse (descending) order.

THRESHOLDMAX= *numeric-value*

Specifies a threshold for displaying one more tick mark at the high end of the axis.

THRESHOLDMIN= *numeric-value*

Specifies a threshold for displaying one more tick mark at the low end of the axis.

TICKVALUEFORMAT= DATA | *SAS-format*

specifies the format for the axis tick values.

TYPE= DISCRETE | LINEAR | LOG | TIME

specifies the type of axis.

VALUEATTRS= *style-element* <(*options*)> | (*options*)

specifies the appearance of the axis tick value labels.

VALUES= (*values-list*)

specifies the values for the ticks on an axis.

VALUESHINT

specifies that the minimum and maximum axis values are determined independently of the values that you specify in the VALUES= option.

Optional Arguments**DISCRETEORDER= DATA | FORMATTED | UNFORMATTED**

specifies the order in which discrete tick values are placed on the axis. Specify one of the following values:

DATA

places the values in the order in which they appear in the data.

FORMATTED

sorts the formatted values in ascending character order.

UNFORMATTED

sorts the unformatted values in ascending character order.

Default: UNFORMATTED

Restriction: This option affects only box plots, dot plots, bar charts, and line plots, or for any axis where TYPE=DISCRETE.

DISPLAY= ALL | NONE | (options)

specifies which features of the axis are displayed. ALL displays all of the features of the axis. NONE specifies that none of the features of the axis are displayed. You can also hide specific features:

NOLABEL

hides the axis label

NOLINE

hides the axis line

NOTICKS

hides the tick marks on the axis

NOVALUES

hides the tick mark values on the axis

Default: ALL

FITPOLICY= policy-value

specifies the method that is used to fit tick mark values on a horizontal axis when there is not enough room to draw them normally. Select one of the following values:

ROTATE

rotates the value text 45 degrees. This is the default for discrete axes.

ROTATETHIN

attempts to use ROTATE, and then THIN to fit the values.

STAGGER

shifts the values up and down.

STAGGERROTATE

attempts to use STAGGER, and then ROTATE to fit the values.

STAGGERTHIN

attempts to use STAGGER, and then THIN to fit the values.

THIN

removes some of the values from the axis. This is the default for linear and time axes.

Note: This option has no effect if used with the vertical (Y or Y2) axes; only the THIN fit policy is used for vertical axes.

GRID

creates grid lines at each tick on the axis.

INTEGER

specifies that only integers are used for tick mark values. This option affects only linear axes.

INTERVAL= *interval-value*

specifies the tick interval for a time axis. Specify one of the following values:

AUTO

determines the tick interval automatically based on the data.

SECOND

places ticks one second apart. The default tick value format is TIME8.

MINUTE

places ticks one minute apart. The default tick value format is TIME8.

HOURL

places ticks one hour apart. The default tick value format is TIME8.

DAY

places ticks one day apart. The default tick value format is DATE9.

WEEK

places ticks one week apart. The default tick value format is DATE9.

TENDAY

places ticks ten days apart. The default tick value format is DATE9.

SEMIMONTH

places ticks at the first and sixteenth day of each month. The default tick value format is DATE9.

MONTH

places ticks one month apart. The default tick value format is MONYY7.

QUARTER

places ticks three months apart. The default tick value format is YYQC6.

SEMIYEAR

places ticks six months apart. The default tick value format is MONYY7.

YEAR

places ticks one year apart. The default tick value format is YEAR4.

Default: AUTO

LABEL= "*text-string*"

specifies a label for the axis.

LABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the axis labels. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the label characters. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more

information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphLabelText style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

FAMILY= “font-family”

specifies the font family for the label characters. The SAS ODS styles use TrueType system fonts. For more information, see “TrueType Fonts Supplied by SAS” in Chapter 15 of *SAS Language Reference: Concepts*.

Default: The default font family is specified by the FontFamily attribute of the GraphLabelText style element in the current style.

Restriction: You cannot specify a list of font family names.

SIZE= *n* <units>

specifies the font size of the label characters. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default font size is specified by the FontSize attribute of the GraphValueText style element in the current style.

STYLE= ITALIC | NORMAL

specifies whether the label characters are italic or normal.

Default: The default font style is specified by the FontStyle attribute of the GraphLabelText style element in the current style.

WEIGHT= BOLD | NORMAL

specifies whether the label characters are bold or normal.

Default: The default font weight is specified by the FontWeight attribute of the GraphLabelText style element in the current style.

Interaction: This option has no effect unless CURVELABEL is also specified.

LOGBASE= 2 | 10 | e

specifies the base value for the logarithmic scale.

Default: 10

Interaction: This option has no effect unless you also specify TYPE=LOG.

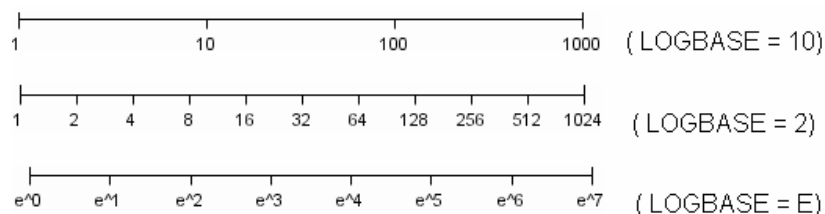
LOGSTYLE= LINEAR | LOGEXPAND | LOGEXPONENT

specifies how to scale and format the values for the major tick marks for logarithmic axes. Specify one of the following values:

LOGEXPAND

places the tick marks at integer powers of the base. For example, if you specified **LOGBASE=2**, the tick marks might be at 1, 2, 4, 8, 16. See Figure 7.5 on page 472.

Figure 7.8 Graph Axes with LOGEXPAND



LOGEXPONENT

places the tick marks at integer powers of the base, but identifies the values by the exponent. For example, if you specified **LOGBASE=10**, the tick marks might be at 1, 10, 100, 1000, but the tick values would read 0, 1, 2, 3. See [Figure 7.6 on page 472](#).

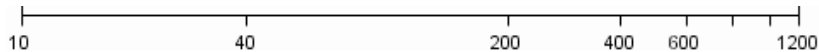
Figure 7.9 An Axis with **LOGEXPONENT**

**LINEAR**

places the tick marks at uniform linear intervals, but spaces them logarithmically. In some cases an intermediate tick mark is placed between the first and second marks.

For example, if the data on this axis range from 14 to 1154, and you specify **LOGBASE=10**, then the tick marks might be at 10, 40, 200, 400, 600, 800, 1000, 1200. See [Figure 7.7 on page 472](#).

Figure 7.10 An Axis with **LINEAR**

**Default:** LOGEXPAND

Interaction: This option has no effect unless you also specify **TYPE=LOG**.

MAX= *numeric-value*

specifies the maximum value for the axis.

Restriction: This option affects linear, log, and time axes only.

Interaction: This option has no effect if you specify the **VALUES=** option and you do not also specify the **VALUESHINT** option.

MIN= *numeric-value*

specifies the minimum value for the axis.

Restriction: This option affects linear, log, and time axes only.

Interaction: This option has no effect if you specify the **VALUES=** option and you do not also specify the **VALUESHINT** option.

MINOR

adds minor tick marks to a log or time axis.

Restriction: This option has no effect on linear or discrete axes.

Interaction: This option has no effect if you specify the **VALUES=** option.

NOTIMESPLIT

prevents a time axis from splitting the time, date, or datetime values into two rows.

Restriction: This option applies to time axes only.

OFFSETMAX= *numeric-value*

specifies an offset that follows the highest data value on the axis. Specify a value between 0 and 1.

The value represents the offset as a proportion to the total length of the axis. For a continuous axis, the offset follows the highest data value or highest tick value, whichever is greater. For a discrete axis, the offset is applied to the end of the axis farther from the origin.

Default: The offset space is determined automatically based on the data values, tick mark values, markers, and labels that are inside of the plot area.

OFFSETMIN= *numeric-value*

specifies an offset that precedes the lowest data value on the axis. Specify a value between 0 and 1.

The value represents the offset as a proportion to the total length of the axis. For a continuous axis, the offset precedes the lowest data value or lowest tick value, whichever is less. For a discrete axis, the offset is applied to the end of the axis nearer to the origin.

Default: The offset space is determined automatically based on the data values, tick mark values, markers, and labels that are inside of the plot area.

REFTICKS

adds tick marks to axis that is opposite from the specified axis. For example, if you specify the REFTICKS option in the XAXIS statement, then tick marks are added to the X2 axis.

Note: This option has no effect if the target axis already contains data.

REVERSE

specifies that the tick values are displayed in reverse (descending) order.

THRESHOLDMAX= *numeric-value*

Specifies a threshold for displaying one more tick mark at the high end of the axis.

THRESHOLDMIN= *numeric-value*

Specifies a threshold for displaying one more tick mark at the low end of the axis.

TICKVALUEFORMAT= DATA | *SAS-format*

specifies the format for the axis tick values. You can either specify a SAS format or specify DATA, which indicates that the format from the axis variable is used.

TYPE= DISCRETE | LINEAR | LOG | TIME

specifies the type of axis. Specify one of the following values:

DISCRETE

specifies an axis with discrete values. If a character variable is assigned to an axis, then the default type for that axis is discrete. In addition, all categorization plots use a discrete axis for the category variable.

LINEAR

specifies a linear scale for the axis. This is the default axis type for numeric variables, except when the data is discrete or when the numeric variable has a date or time format.

LOG

specifies a logarithmic scale for the axis. This axis type is never a default.

Interaction: A logarithmic scale cannot be used with linear regression plots (REG statement where DEGREE=1).

TIME

specifies a time scale for the axis. If the variable assigned to an axis has a time, date, or datetime format associated with it, then time is the default axis type.

VALUEATTRS= *style-element* <(options)> | (options)

specifies the appearance of the axis tick value labels. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the label characters. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphValueText style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

FAMILY= “*font-family*”

specifies the font family for the label characters. The SAS ODS styles use TrueType system fonts. For more information, see “TrueType Fonts Supplied by SAS” in Chapter 15 of *SAS Language Reference: Concepts*.

Default: The default font family is specified by the FontFamily attribute of the GraphValueText style element in the current style.

Restriction: You cannot specify a list of font family names.

SIZE= *n* <*units*>

specifies the font size of the label characters. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default font size is specified by the FontSize attribute of the GraphValueText style element in the current style.

STYLE= ITALIC | NORMAL

specifies whether the label characters are italic or normal.

Default: The default font style is specified by the FontStyle attribute of the GraphValueText style element in the current style.

WEIGHT= BOLD | NORMAL

specifies whether the label characters are bold or normal.

Default: The default font weight is specified by the FontWeight attribute of the GraphValueText style element in the current style.

VALUES= (*values-list*)

specifies the values for the ticks on an axis.

For values on a linear axis, the values list can be one of the following:

value <...*value-n*>

creates ticks for specific values. For example, **VALUES= (0 50 100)** places tick marks at 0, 50, and 100.

value-1 TO *value-2* BY *increment-value*

creates ticks for a range of values. The start of the value range is specified by *value-1* and the end of the range is specified by *value-2*. The *increment-value* specifies the interval between the ticks. For example, **VALUES= (0 to 100 by 50)** creates tick marks at 0, 50, and 100.

<*value* ... *value-n*>*value-1* TO *value-2* BY *increment-value* <*value* ... *value-n*>

creates ticks for specific values, and also creates ticks for a range of values. The start of the value range is specified by *value-1* and the end of the range is specified by *value-2*. The *increment-value* specifies the interval between the ticks.

For example, **VALUES= (-5 10 to 50 by 20 75)** creates tick marks at -5, 10, 30, 50, and 75.

For values on a time axis, the values list can be one of the following:

value <...*value-n*>

creates ticks for specific values. For example, **VALUES= ("25MAY08"d "04JUL08"d "23AUG08"d)** places tick marks at 25MAY08, 04JUL08, and 23AUG08.

value-1 TO *value-2* BY *increment-value*

creates ticks for a range of values. The start of the value range is specified by *value-1* and the end of the range is specified by *value-2*. The *increment-value* specifies the interval between the ticks. For example, **VALUES= ("01JAN08"d to "01MAY08"d by month)** creates tick marks at 01JAN08, 01FEB08, 01MAR08, 01APR08, and 01MAY08.

For a list of the interval values that you can specify, see the **INTERVAL=** option.

Restrictions:

This option has no effect on discrete and logarithmic axes.

If your **VALUES=** option creates more than 1000 values, then the option has no effect.

Interaction: If a custom format is applied to the value, the raw value is plotted first and then formatted.

VALUESHINT

specifies that the minimum and maximum axis values are determined independently of the values that you specify in the **VALUES=** option. The values from the **VALUES=** option are displayed only if they are located between the minimum and maximum values.

Interaction: This option has no effect unless you also specify the **VALUES=** option.

YAXIS Statement

Specifies the axis options for the Y axis. You can control the features of the axis (for example, the axis label, grid lines, and minor tick marks). You can also control the structure of the axis (for example, the data range, data type, and tick mark values).

Syntax

YAXIS *option(s)*;

Summary of Optional Arguments

Axis options

DISCRETEORDER= *DATA* | *FORMATTED* | *UNFORMATTED*

specifies the order in which discrete tick values are placed on the axis.

DISPLAY= *ALL* | *NONE* | (*options*)

specifies which features of the axis are displayed.

GRID

creates grid lines at each tick on the axis.

INTEGER

specifies that only integers are used for tick mark values.

INTERVAL= *interval-value*

specifies the tick interval for a time axis.

LABEL= "text-string"

specifies a label for the axis.

LABELATTRS= *style-element* <(*options*)> | (*options*)

specifies the appearance of the axis labels.

LOGBASE= 2 | 10 | e

specifies the base value for the logarithmic scale.

LOGSTYLE= LINEAR | LOGEXPAND | LOGEXPONENT

specifies how to scale and format the values for the major tick marks for logarithmic axes.

MAX= *numeric-value*

specifies the maximum value for the axis.

MIN= *numeric-value*

specifies the minimum value for the axis.

MINOR

adds minor tick marks to a log or time axis.

NOTIMESPLIT

prevents a time axis from splitting the time, date, or datetime values into two rows.

OFFSETMAX= *numeric-value*

specifies an offset that follows the highest data value on the axis.

OFFSETMIN= *numeric-value*

specifies an offset that precedes the lowest data value on the axis.

REFTICKS

adds tick marks to axis that is opposite from the specified axis.

REVERSE

specifies that the tick values are displayed in reverse (descending) order.

THRESHOLDMAX= *numeric-value*

Specifies a threshold for displaying one more tick mark at the high end of the axis.

THRESHOLDMIN= *numeric-value*

Specifies a threshold for displaying one more tick mark at the low end of the axis.

TICKVALUEFORMAT= DATA | *SAS-format*

specifies the format for the axis tick values.

TYPE= DISCRETE | LINEAR | LOG | TIME

specifies the type of axis.

VALUEATTRS= *style-element* <(*options*)> | (*options*)

specifies the appearance of the axis tick value labels.

VALUES= (*values-list*)

specifies the values for the ticks on an axis.

VALUESHINT

specifies that the minimum and maximum axis values are determined independently of the values that you specify in the VALUES= option.

Optional Arguments

DISCRETEORDER= DATA | FORMATTED | UNFORMATTED

specifies the order in which discrete tick values are placed on the axis. Specify one of the following values:

DATA

places the values in the order in which they appear in the data.

FORMATTED

sorts the formatted values in ascending character order.

UNFORMATTED

sorts the unformatted values in ascending character order.

Default: UNFORMATTED

Restriction: This option affects only box plots, dot plots, bar charts, and line plots, or for any axis where TYPE=DISCRETE.

DISPLAY= ALL | NONE | (options)

specifies which features of the axis are displayed. ALL displays all of the features of the axis. NONE specifies that none of the features of the axis are displayed. You can also hide specific features:

NOLABEL

hides the axis label

NOLINE

hides the axis line

NOTICKS

hides the tick marks on the axis

NOVALUES

hides the tick mark values on the axis

Default: ALL

GRID

creates grid lines at each tick on the axis.

INTEGER

specifies that only integers are used for tick mark values. This option affects only linear axes.

INTERVAL= interval-value

specifies the tick interval for a time axis. Specify one of the following values:

AUTO

determines the tick interval automatically based on the data.

SECOND

places ticks one second apart. The default tick value format is TIME8.

MINUTE

places ticks one minute apart. The default tick value format is TIME8.

HOURL

places ticks one hour apart. The default tick value format is TIME8.

DAY

places ticks one day apart. The default tick value format is DATE9.

WEEK

places ticks one week apart. The default tick value format is DATE9.

TENDAY

places ticks ten days apart. The default tick value format is DATE9.

SEMIMONTH

places ticks at the first and sixteenth day of each month. The default tick value format is DATE9.

MONTH

places ticks one month apart. The default tick value format is MONYY7.

QUARTER

places ticks three months apart. The default tick value format is YYQC6.

SEMIYEAR

places ticks six months apart. The default tick value format is MONYY7.

YEAR

places ticks one year apart. The default tick value format is YEAR4.

Default: AUTO

LABEL= “text-string”

specifies a label for the axis.

LABELATTRS= style-element <(options)> | (options)

specifies the appearance of the axis labels. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= color

specifies the color of the label characters. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphLabelText style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

FAMILY= “font-family”

specifies the font family for the label characters. The SAS ODS styles use TrueType system fonts. For more information, see “TrueType Fonts Supplied by SAS” in Chapter 15 of *SAS Language Reference: Concepts*.

Default: The default font family is specified by the FontFamily attribute of the GraphLabelText style element in the current style.

Restriction: You cannot specify a list of font family names.

SIZE= n <units>

specifies the font size of the label characters. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default font size is specified by the FontSize attribute of the GraphValueText style element in the current style.

STYLE= ITALIC | NORMAL

specifies whether the label characters are italic or normal.

Default: The default font style is specified by the FontStyle attribute of the GraphLabelText style element in the current style.

WEIGHT= BOLD | NORMAL

specifies whether the label characters are bold or normal.

Default: The default font weight is specified by the FontWeight attribute of the GraphLabelText style element in the current style.

Interaction: This option has no effect unless CURVELABEL is also specified.

LOGBASE= 2 | 10 | e

specifies the base value for the logarithmic scale.

Default: 10

Interaction: This option has no effect unless you also specify TYPE=LOG.

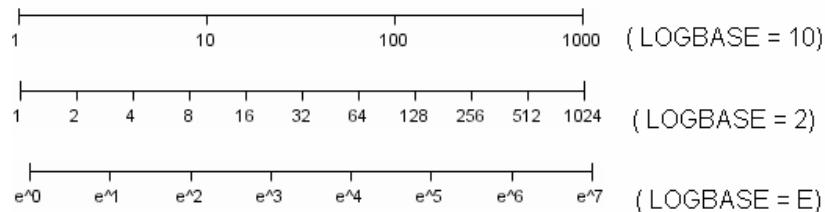
LOGSTYLE= LINEAR | LOGEXPAND | LOGEXPONENT

specifies how to scale and format the values for the major tick marks for logarithmic axes. Specify one of the following values:

LOGEXPAND

places the tick marks at integer powers of the base. For example, if you specified **LOGBASE=2**, the tick marks might be at 1, 2, 4, 8, 16. See [Figure 7.5 on page 472](#).

Figure 7.11 Graph Axes with LOGEXPAND



LOGEXPONENT

places the tick marks at integer powers of the base, but identifies the values by the exponent. For example, if you specified **LOGBASE=10**, the tick marks might be at 1, 10, 100, 1000, but the tick values would read 0, 1, 2, 3. See [Figure 7.6 on page 472](#).

Figure 7.12 An Axis with LOGEXPONENT

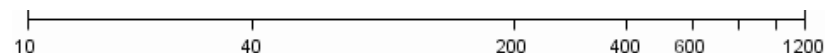


LINEAR

places the tick marks at uniform linear intervals, but spaces them logarithmically. In some cases an intermediate tick mark is placed between the first and second marks.

For example, if the data on this axis range from 14 to 1154, and you specify **LOGBASE=10**, then the tick marks might be at 10, 40, 200, 400, 600, 800, 1000, 1200. See [Figure 7.7 on page 472](#).

Figure 7.13 An Axis with LINEAR



Default: LOGEXPAND

Interaction: This option has no effect unless you also specify TYPE=LOG.

MAX= *numeric-value*

specifies the maximum value for the axis.

Restriction: This option affects linear, log, and time axes only.

Interaction: This option has no effect if you specify the VALUES= option and you do not also specify the VALUESHINT option.

MIN= *numeric-value*

specifies the minimum value for the axis.

Restriction: This option affects linear, log, and time axes only.

Interaction: This option has no effect if you specify the `VALUES=` option and you do not also specify the `VALUESHINT` option.

MINOR

adds minor tick marks to a log or time axis.

Restriction: This option has no effect on linear or discrete axes.

Interaction: This option has no effect if you specify the `VALUES=` option.

NOTIMESPLIT

prevents a time axis from splitting the time, date, or datetime values into two rows.

Restriction: This option applies to time axes only.

OFFSETMAX= *numeric-value*

specifies an offset that follows the highest data value on the axis. Specify a value between 0 and 1.

The value represents the offset as a proportion to the total length of the axis. For a continuous axis, the offset follows the highest data value or highest tick value, whichever is greater. For a discrete axis, the offset is applied to the end of the axis farther from the axis origin.

Default: The offset space is determined automatically based on the data values, tick mark values, markers, and labels that are inside of the plot area.

Interaction: For `HBOX`, `HBAR`, `HLINE`, and `DOT` plot statements, the Y axis is reversed by default, so the axis origin is at the top. For these plots, `OFFSETMAX=` creates an offset at the end of the Y axis nearer to the origin.

OFFSETMIN= *numeric-value*

specifies an offset that precedes the lowest data value on the axis. Specify a value between 0 and 1.

The value represents the offset as a proportion to the total length of the axis. For a continuous axis, the offset precedes the lowest data value or lowest tick value, whichever is less. For a discrete axis, the offset is applied to the end of the axis nearer to the axis origin.

Default: The offset space is determined automatically based on the data values, tick mark values, markers, and labels that are inside of the plot area.

Interaction: For `HBOX`, `HBAR`, `HLINE`, and `DOT` plot statements, the Y axis is reversed by default, so the axis origin is at the top. For these plots, `OFFSETMIN=` creates an offset at the end of the Y axis farther from the origin.

REFTICKS

adds tick marks to axis that is opposite from the specified axis. For example, if you specify the `REFTICKS` option in the `XAXIS` statement, then tick marks are added to the X2 axis.

Note: This option has no effect if the target axis already contains data.

REVERSE

specifies that the tick values are displayed in reverse (descending) order.

THRESHOLDMAX= *numeric-value*

Specifies a threshold for displaying one more tick mark at the high end of the axis.

THRESHOLDMIN= *numeric-value*

Specifies a threshold for displaying one more tick mark at the low end of the axis.

TICKVALUEFORMAT= DATA | *SAS-format*

specifies the format for the axis tick values. You can either specify a SAS format or specify `DATA`, which indicates that the format from the axis variable is used.

TYPE= DISCRETE | LINEAR | LOG | TIME

specifies the type of axis. Specify one of the following values:

DISCRETE

specifies an axis with discrete values. If a character variable is assigned to an axis, then the default type for that axis is discrete. In addition, all categorization plots use a discrete axis for the category variable.

LINEAR

specifies a linear scale for the axis. This is the default axis type for numeric variables, except when the data is discrete or when the numeric variable has a date or time format.

LOG

specifies a logarithmic scale for the axis. This axis type is never a default.

Interaction: A logarithmic scale cannot be used with linear regression plots (REG statement where DEGREE=1).

TIME

specifies a time scale for the axis. If the variable assigned to an axis has a time, date, or datetime format associated with it, then time is the default axis type.

VALUEATTRS= *style-element* <(options)> | (options)

specifies the appearance of the axis tick value labels. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the label characters. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphValueText style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

FAMILY= “*font-family*”

specifies the font family for the label characters. The SAS ODS styles use TrueType system fonts. For more information, see “TrueType Fonts Supplied by SAS” in Chapter 15 of *SAS Language Reference: Concepts*.

Default: The default font family is specified by the FontFamily attribute of the GraphValueText style element in the current style.

Restriction: You cannot specify a list of font family names.

SIZE= *n* <units>

specifies the font size of the label characters. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default font size is specified by the FontSize attribute of the GraphValueText style element in the current style.

STYLE= ITALIC | NORMAL

specifies whether the label characters are italic or normal.

Default: The default font style is specified by the FontStyle attribute of the GraphValueText style element in the current style.

WEIGHT= BOLD | NORMAL

specifies whether the label characters are bold or normal.

Default: The default font weight is specified by the `FontWeight` attribute of the `GraphValueText` style element in the current style.

VALUES= (*values-list*)

specifies the values for the ticks on an axis.

For values on a linear axis, the values list can be one of the following:

value <...*value-n*>

creates ticks for specific values. For example, **VALUES= (0 50 100)** places tick marks at 0, 50, and 100.

value-1 TO *value-2* BY *increment-value*

creates ticks for a range of values. The start of the value range is specified by *value-1* and the end of the range is specified by *value-2*. The *increment-value* specifies the interval between the ticks. For example, **VALUES= (0 to 100 by 50)** creates tick marks at 0, 50, and 100.

<*value* ... *value-n*>*value-1* TO *value-2* BY *increment-value* <*value* ... *value-n*>

creates ticks for specific values, and also creates ticks for a range of values. The start of the value range is specified by *value-1* and the end of the range is specified by *value-2*. The *increment-value* specifies the interval between the ticks.

For example, **VALUES= (-5 10 to 50 by 20 75)** creates tick marks at -5, 10, 30, 50, and 75.

For values on a time axis, the values list can be one of the following:

value <...*value-n*>

creates ticks for specific values. For example, **VALUES= ("25MAY08"d "04JUL08"d "23AUG08"d)** places tick marks at 25MAY08, 04JUL08, and 23AUG08.

value-1 TO *value-2* BY *increment-value*

creates ticks for a range of values. The start of the value range is specified by *value-1* and the end of the range is specified by *value-2*. The *increment-value* specifies the interval between the ticks. For example, **VALUES= ("01JAN08"d to "01MAY08"d by month)** creates tick marks at 01JAN08, 01FEB08, 01MAR08, 01APR08, and 01MAY08.

For a list of the interval values that you can specify, see the `INTERVAL=` option.

Restrictions:

This option has no effect on discrete and logarithmic axes.

If your `VALUES=` option creates more than 1000 values, then the option has no effect.

Interaction: If a custom format is applied to the value, the raw value is plotted first and then formatted.

VALUESHINT

specifies that the minimum and maximum axis values are determined independently of the values that you specify in the `VALUES=` option. The values from the `VALUES=` option are displayed only if they are located between the minimum and maximum values.

Interaction: This option has no effect unless you also specify the `VALUES=` option.

Y2AXIS Statement

Specifies the axis options for the Y2 axis. You can control the features of the axis (for example, the axis label, grid lines, and minor tick marks). You can also control the structure of the axis (for example, the data range, data type, and tick mark values).

Syntax

Y2AXIS *option(s)*;

Summary of Optional Arguments

Axis options

DISCRETEORDER= DATA | FORMATTED | UNFORMATTED

specifies the order in which discrete tick values are placed on the axis.

DISPLAY= ALL | NONE | (*options*)

specifies which features of the axis are displayed.

GRID

creates grid lines at each tick on the axis.

INTEGER

specifies that only integers are used for tick mark values.

INTERVAL= *interval-value*

specifies the tick interval for a time axis.

LABEL= "*text-string*"

specifies a label for the axis.

LABELATTRS= *style-element* <(*options*)> | (*options*)

specifies the appearance of the axis labels.

LOGBASE= 2 | 10 | e

specifies the base value for the logarithmic scale.

LOGSTYLE= LINEAR | LOGEXPAND | LOGEXPONENT

specifies how to scale and format the values for the major tick marks for logarithmic axes.

MAX= *numeric-value*

specifies the maximum value for the axis.

MIN= *numeric-value*

specifies the minimum value for the axis.

MINOR

adds minor tick marks to a log or time axis.

NOTIMESPLIT

prevents a time axis from splitting the time, date, or datetime values into two rows.

OFFSETMAX= *numeric-value*

specifies an offset that follows the highest data value on the axis.

OFFSETMIN= *numeric-value*

specifies an offset that precedes the lowest data value on the axis.

REFTICKS

adds tick marks to axis that is opposite from the specified axis.

REVERSE

specifies that the tick values are displayed in reverse (descending) order.

THRESHOLDMAX= *numeric-value*

Specifies a threshold for displaying one more tick mark at the high end of the axis.

THRESHOLDMIN= *numeric-value*

Specifies a threshold for displaying one more tick mark at the low end of the axis.

TICKVALUEFORMAT= *DATA* | *SAS-format*

specifies the format for the axis tick values.

TYPE= *DISCRETE* | *LINEAR* | *LOG* | *TIME*

specifies the type of axis.

VALUEATTRS= *style-element* <(*options*)> | (*options*)

specifies the appearance of the axis tick value labels.

VALUES= (*values-list*)

specifies the values for the ticks on an axis.

VALUESHINT

specifies that the minimum and maximum axis values are determined independently of the values that you specify in the **VALUES=** option.

Optional Arguments

DISCRETEORDER= *DATA* | *FORMATTED* | *UNFORMATTED*

specifies the order in which discrete tick values are placed on the axis. Specify one of the following values:

DATA

places the values in the order in which they appear in the data.

FORMATTED

sorts the formatted values in ascending character order.

UNFORMATTED

sorts the unformatted values in ascending character order.

Default: UNFORMATTED

Restriction: This option affects only box plots, dot plots, bar charts, and line plots, or for any axis where **TYPE=DISCRETE**.

DISPLAY= *ALL* | *NONE* | (*options*)

specifies which features of the axis are displayed. *ALL* displays all of the features of the axis. *NONE* specifies that none of the features of the axis are displayed. You can also hide specific features:

NOLABEL

hides the axis label

NOLINE

hides the axis line

NOTICKS

hides the tick marks on the axis

NOVALUES

hides the tick mark values on the axis

Default: ALL

GRID

creates grid lines at each tick on the axis.

INTEGER

specifies that only integers are used for tick mark values. This option affects only linear axes.

INTERVAL= *interval-value*

specifies the tick interval for a time axis. Specify one of the following values:

AUTO

determines the tick interval automatically based on the data.

SECOND

places ticks one second apart. The default tick value format is TIME8.

MINUTE

places ticks one minute apart. The default tick value format is TIME8.

HOURL

places ticks one hour apart. The default tick value format is TIME8.

DAY

places ticks one day apart. The default tick value format is DATE9.

WEEK

places ticks one week apart. The default tick value format is DATE9.

TENDAY

places ticks ten days apart. The default tick value format is DATE9.

SEMIMONTH

places ticks at the first and sixteenth day of each month. The default tick value format is DATE9.

MONTH

places ticks one month apart. The default tick value format is MONYY7.

QUARTER

places ticks three months apart. The default tick value format is YYQC6.

SEMIYEAR

places ticks six months apart. The default tick value format is MONYY7.

YEAR

places ticks one year apart. The default tick value format is YEAR4.

Default: AUTO

LABEL= "*text-string*"

specifies a label for the axis.

LABELATTRS= *style-element* <(options)> | (options)

specifies the appearance of the axis labels. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the label characters. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see "Color-Naming Schemes" in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphLabelText style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

FAMILY= “font-family”

specifies the font family for the label characters. The SAS ODS styles use TrueType system fonts. For more information, see “TrueType Fonts Supplied by SAS” in Chapter 15 of *SAS Language Reference: Concepts*.

Default: The default font family is specified by the FontFamily attribute of the GraphLabelText style element in the current style.

Restriction: You cannot specify a list of font family names.

SIZE= *n* <units>

specifies the font size of the label characters. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default font size is specified by the FontSize attribute of the GraphValueText style element in the current style.

STYLE= ITALIC | NORMAL

specifies whether the label characters are italic or normal.

Default: The default font style is specified by the FontStyle attribute of the GraphLabelText style element in the current style.

WEIGHT= BOLD | NORMAL

specifies whether the label characters are bold or normal.

Default: The default font weight is specified by the FontWeight attribute of the GraphLabelText style element in the current style.

Interaction: This option has no effect unless CURVELABEL is also specified.

LOGBASE= 2 | 10 | e

specifies the base value for the logarithmic scale.

Default: 10

Interaction: This option has no effect unless you also specify TYPE=LOG.

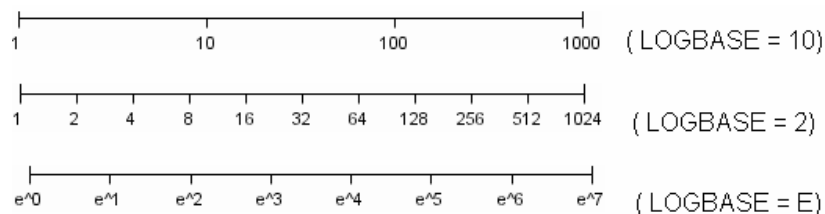
LOGSTYLE= LINEAR | LOGEXPAND | LOGEXPONENT

specifies how to scale and format the values for the major tick marks for logarithmic axes. Specify one of the following values:

LOGEXPAND

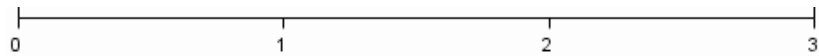
places the tick marks at integer powers of the base. For example, if you specified **LOGBASE=2**, the tick marks might be at 1, 2, 4, 8, 16. See Figure 7.5 on page 472.

Figure 7.14 Graph Axes with LOGEXPAND



LOGEXPONENT

places the tick marks at integer powers of the base, but identifies the values by the exponent. For example, if you specified **LOGBASE=10**, the tick marks might be at 1, 10, 100, 1000, but the tick values would read 0, 1, 2, 3. See Figure 7.6 on page 472.

Figure 7.15 An Axis with LOGEXPONENT**LINEAR**

places the tick marks at uniform linear intervals, but spaces them logarithmically. In some cases an intermediate tick mark is placed between the first and second marks.

For example, if the data on this axis range from 14 to 1154, and you specify **LOGBASE=10**, then the tick marks might be at 10, 40, 200, 400, 600, 800, 1000, 1200. See [Figure 7.7 on page 472](#).

Figure 7.16 An Axis with LINEAR**Default:** LOGEXPAND

Interaction: This option has no effect unless you also specify TYPE=LOG.

MAX= numeric-value

specifies the maximum value for the axis.

Restriction: This option affects linear, log, and time axes only.

Interaction: This option has no effect if you specify the VALUES= option and you do not also specify the VALUESHINT option.

MIN= numeric-value

specifies the minimum value for the axis.

Restriction: This option affects linear, log, and time axes only.

Interaction: This option has no effect if you specify the VALUES= option and you do not also specify the VALUESHINT option.

MINOR

adds minor tick marks to a log or time axis.

Restriction: This option has no effect on linear or discrete axes.

Interaction: This option has no effect if you specify the VALUES= option.

NOTIMESPLIT

prevents a time axis from splitting the time, date, or datetime values into two rows.

Restriction: This option applies to time axes only.

OFFSETMAX= numeric-value

specifies an offset that follows the highest data value on the axis. Specify a value between 0 and 1.

The value represents the offset as a proportion to the total length of the axis. For a continuous axis, the offset follows the highest data value or highest tick value, whichever is greater. For a discrete axis, the offset is applied to the end of the axis farther from the axis origin.

Default: The offset space is determined automatically based on the data values, tick mark values, markers, and labels that are inside of the plot area.

Interaction: For HBOX, HBAR, HLINE, and DOT plot statements, the Y axis is reversed by default, so the axis origin is at the top. For these plots, OFFSETMAX= creates an offset at the end of the Y axis nearer to the origin.

OFFSETMIN= *numeric-value*

specifies an offset that precedes the lowest data value on the axis. Specify a value between 0 and 1.

The value represents the offset as a proportion to the total length of the axis. For a continuous axis, the offset precedes the lowest data value or lowest tick value, whichever is less. For a discrete axis, the offset is applied to the end of the axis nearer to the axis origin.

Default: The offset space is determined automatically based on the data values, tick mark values, markers, and labels that are inside of the plot area.

Interaction: For HBOX, HBAR, HLINE, and DOT plot statements, the Y axis is reversed by default, so the axis origin is at the top. For these plots, OFFSETMIN= creates an offset at the end of the Y axis farther from the origin.

REFTICKS

adds tick marks to axis that is opposite from the specified axis. For example, if you specify the REFTICKS option in the XAXIS statement, then tick marks are added to the X2 axis.

Note: This option has no effect if the target axis already contains data.

REVERSE

specifies that the tick values are displayed in reverse (descending) order.

THRESHOLDMAX= *numeric-value*

Specifies a threshold for displaying one more tick mark at the high end of the axis.

THRESHOLDMIN= *numeric-value*

Specifies a threshold for displaying one more tick mark at the low end of the axis.

TICKVALUEFORMAT= DATA | *SAS-format*

specifies the format for the axis tick values. You can either specify a SAS format or specify DATA, which indicates that the format from the axis variable is used.

TYPE= DISCRETE | LINEAR | LOG | TIME

specifies the type of axis. Specify one of the following values:

DISCRETE

specifies an axis with discrete values. If a character variable is assigned to an axis, then the default type for that axis is discrete. In addition, all categorization plots use a discrete axis for the category variable.

LINEAR

specifies a linear scale for the axis. This is the default axis type for numeric variables, except when the data is discrete or when the numeric variable has a date or time format.

LOG

specifies a logarithmic scale for the axis. This axis type is never a default.

Interaction: A logarithmic scale cannot be used with linear regression plots (REG statement where DEGREE=1).

TIME

specifies a time scale for the axis. If the variable assigned to an axis has a time, date, or datetime format associated with it, then time is the default axis type.

VALUEATTRS= *style-element* <(options)> | (options)

specifies the appearance of the axis tick value labels. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the label characters. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the Color attribute of the GraphValueText style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

FAMILY= “*font-family*”

specifies the font family for the label characters. The SAS ODS styles use TrueType system fonts. For more information, see “TrueType Fonts Supplied by SAS” in Chapter 15 of *SAS Language Reference: Concepts*.

Default: The default font family is specified by the FontFamily attribute of the GraphValueText style element in the current style.

Restriction: You cannot specify a list of font family names.

SIZE= *n* <*units*>

specifies the font size of the label characters. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default font size is specified by the FontSize attribute of the GraphValueText style element in the current style.

STYLE= ITALIC | NORMAL

specifies whether the label characters are italic or normal.

Default: The default font style is specified by the FontStyle attribute of the GraphValueText style element in the current style.

WEIGHT= BOLD | NORMAL

specifies whether the label characters are bold or normal.

Default: The default font weight is specified by the FontWeight attribute of the GraphValueText style element in the current style.

VALUES= (*values-list*)

specifies the values for the ticks on an axis.

For values on a linear axis, the values list can be one of the following:

value <...*value-n*>

creates ticks for specific values. For example, **VALUES= (0 50 100)** places tick marks at 0, 50, and 100.

value-1 TO *value-2* BY *increment-value*

creates ticks for a range of values. The start of the value range is specified by *value-1* and the end of the range is specified by *value-2*. The *increment-value* specifies the interval between the ticks. For example, **VALUES= (0 to 100 by 50)** creates tick marks at 0, 50, and 100.

<*value* ... *value-n*>*value-1* TO *value-2* BY *increment-value* <*value* ... *value-n*>

creates ticks for specific values, and also creates ticks for a range of values. The start of the value range is specified by *value-1* and the end of the range is specified by *value-2*. The *increment-value* specifies the interval between the ticks.

For example, **VALUES= (-5 10 to 50 by 20 75)** creates tick marks at -5, 10, 30, 50, and 75.

For values on a time axis, the values list can be one of the following:

value <...*value-n*>

creates ticks for specific values. For example, **VALUES= ("25MAY08"d "04JUL08"d "23AUG08"d)** places tick marks at 25MAY08, 04JUL08, and 23AUG08.

value-1 TO *value-2* BY *increment-value*

creates ticks for a range of values. The start of the value range is specified by *value-1* and the end of the range is specified by *value-2*. The *increment-value* specifies the interval between the ticks. For example, **VALUES= ("01JAN08"d to "01MAY08"d by month)** creates tick marks at 01JAN08, 01FEB08, 01MAR08, 01APR08, and 01MAY08.

For a list of the interval values that you can specify, see the **INTERVAL=** option.

Restrictions:

This option has no effect on discrete and logarithmic axes.

If your **VALUES=** option creates more than 1000 values, then the option has no effect.

Interaction: If a custom format is applied to the value, the raw value is plotted first and then formatted.

VALUESHINT

specifies that the minimum and maximum axis values are determined independently of the values that you specify in the **VALUES=** option. The values from the **VALUES=** option are displayed only if they are located between the minimum and maximum values.

Interaction: This option has no effect unless you also specify the **VALUES=** option.

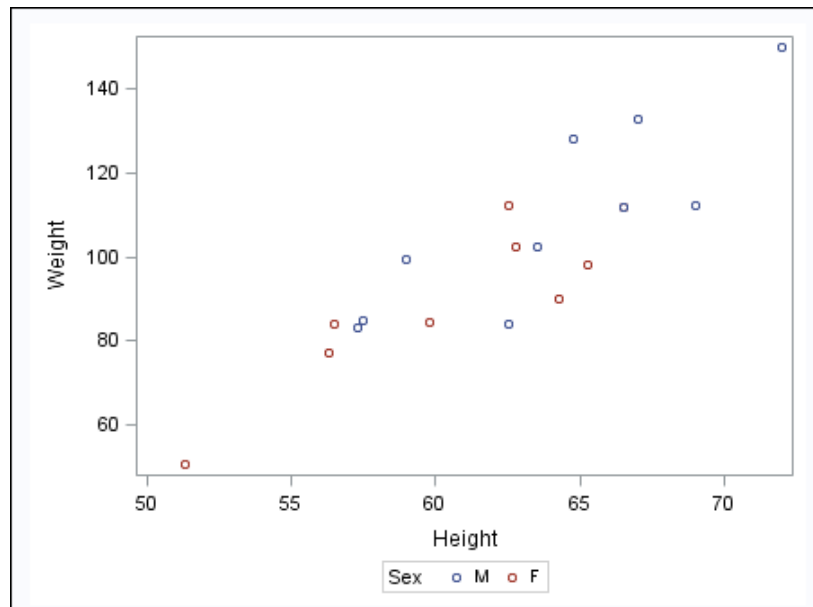
Examples: SGPLOT Procedure

Example 1: Grouping a Scatter Plot

Features: SCATTER statement

Sample library member: SGPLSCT

This example shows a simple scatter plot with grouped data.

Output**Program**

```
proc sgplot data=sashelp.class;
  scatter x=height y=weight / group=sex;
run;
```

Program Description

Create the scatter plot. In the SCATTER statement, the GROUP= option groups the data by the SEX variable.

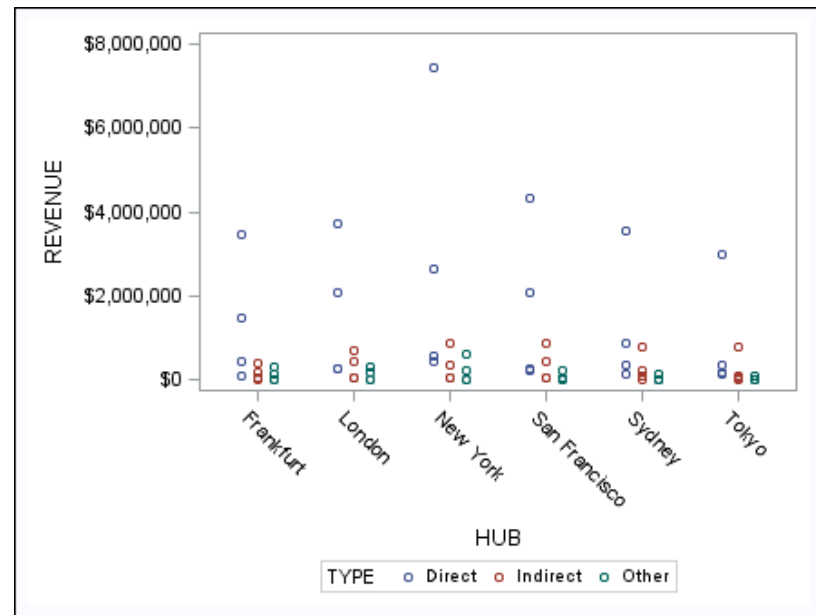
```
proc sgplot data=sashelp.class;
  scatter x=height y=weight / group=sex;
run;
```

Example 2: Clustering a Grouped Scatter Plot

Features: SCATTER statement
 GROUPDISPLAY option
 CLUSTERWIDTH option

Sample library member: SGPLCLU

This example shows a simple scatter plot with grouped data that is clustered.

Output**Program**

```
proc sgplot data=sashelp.revhub2;
  scatter x=hub y=revenue /
    group=type groupdisplay=cluster clusterwidth=0.5;
  xaxis type=discrete;
run;
```

Program Description

Create the scatter plot. In the SCATTER statement, the GROUP= option groups the data by the TYPE variable. The GROUPDISPLAY option specifies that the grouped markers are clustered. The CLUSTERWIDTH option specifies the width of the group clusters.

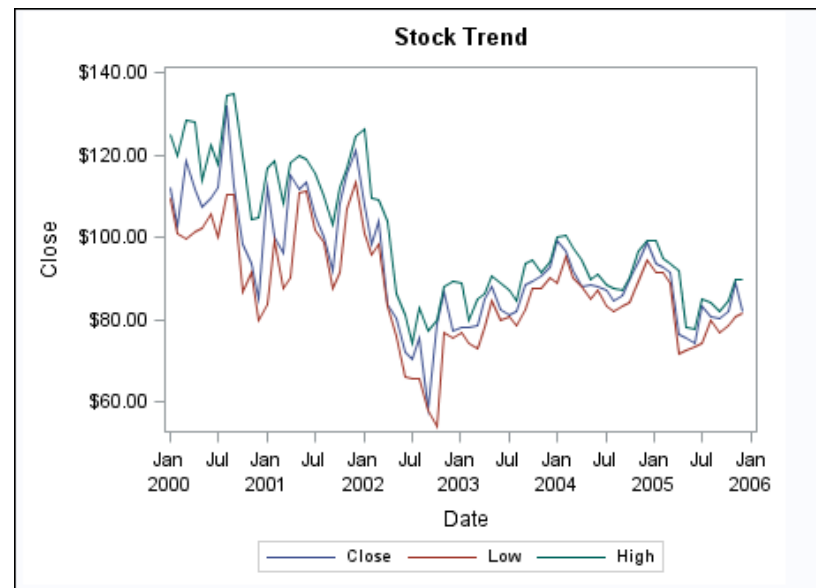
```
proc sgplot data=sashelp.revhub2;
  scatter x=hub y=revenue /
    group=type groupdisplay=cluster clusterwidth=0.5;
  xaxis type=discrete;
run;
```

Example 3: Plotting Three Series

Features: SERIES statement

Sample library member: SGPLSER

This example shows a series plot with three series on the Y axis.

Output**Program**

```
proc sgplot data=sashelp.stocks
  (where=(date >= "01jan2000"d and stock = "IBM"));
  title "Stock Trend";

  series x=date y=close;
  series x=date y=low;
  series x=date y=high;
run;

title;
```

Program Description**Specify the data set and the title.**

```
proc sgplot data=sashelp.stocks
  (where=(date >= "01jan2000"d and stock = "IBM"));
  title "Stock Trend";
```

Create the series plots.

```
  series x=date y=close;
  series x=date y=low;
  series x=date y=high;
run;
```

Cancel the title.

```
title;
```

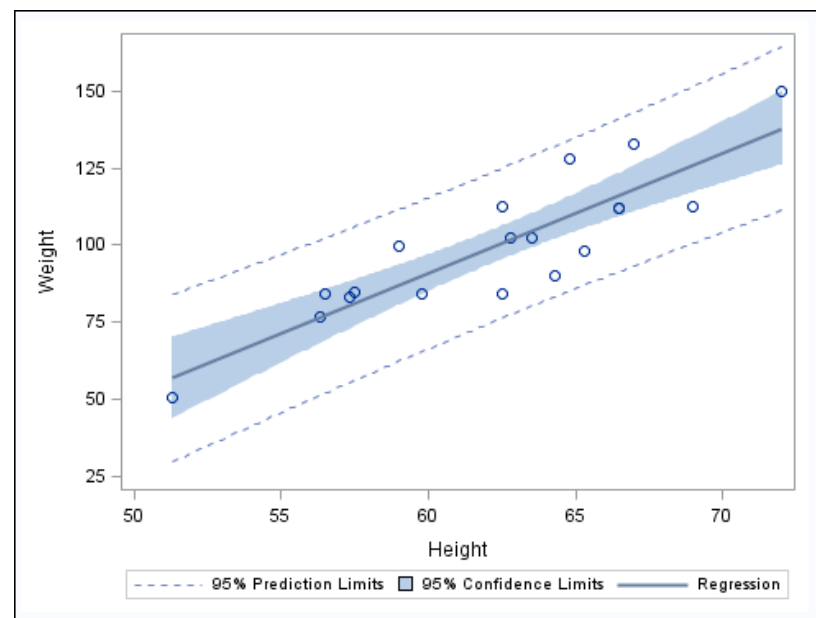
Example 4: Adding Prediction and Confidence Bands to a Regression Plot

Features: REG statement

Sample library member: SGPLREG

This example shows a regression plot with prediction and confidence limits.

Output



Program

```
proc sgplot data=sashelp.class;
  reg x=height y=weight / CLM CLI;
run;
```

Program Description

Create the regression plot. The CLM option adds confidence limits for the mean predicted values. The CLI option adds confidence limits for the individual predicted values.

```
proc sgplot data=sashelp.class;
  reg x=height y=weight / CLM CLI;
run;
```

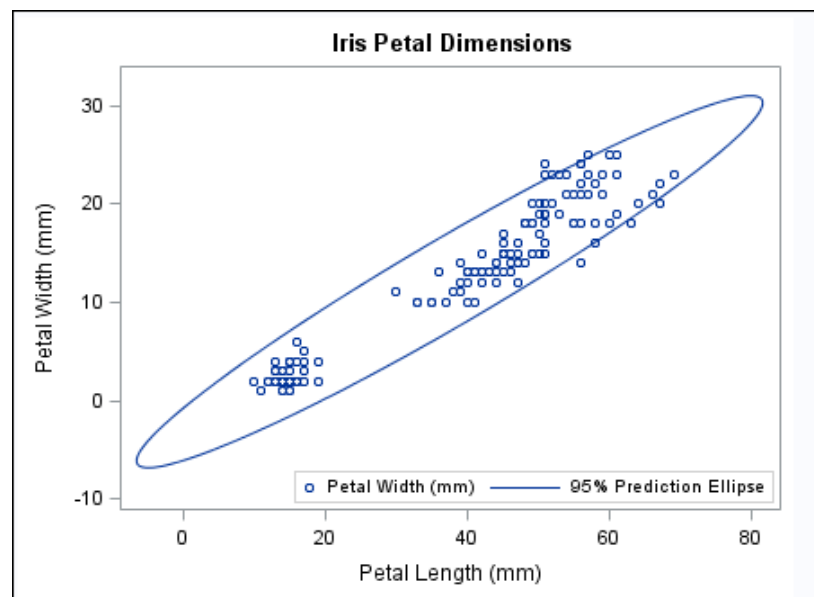
Example 5: Adding a Prediction Ellipse to a Scatter Plot

Features: SCATTER statement
ELLIPSE statement
KEYLEGEND statement

Sample library member: SGPLELI

This example shows a scatter plot with a prediction ellipse.

Output



Program

```
proc sgplot data=sashelp.iris;  
  title "Iris Petal Dimensions";  
  scatter x=petallength y=petalwidth;  
  
  ellipse x=petallength y=petalwidth;  
  
  keylegend / location=inside position=bottomright;  
run;  
  
title;
```

Program Description

Set the title and create the scatter plot.

```
proc sgplot data=sashelp.iris;  
  title "Iris Petal Dimensions";  
  scatter x=petallength y=petalwidth;
```

Create the ellipse.

```
ellipse x=petallength y=petalwidth;
```

Position the Legend. The `LOCATION=` option places the legend inside of the plot area. The `POSITION=` option places the legend at the bottom right.

```
keylegend / location=inside position=bottomright;
run;
```

Cancel the title.

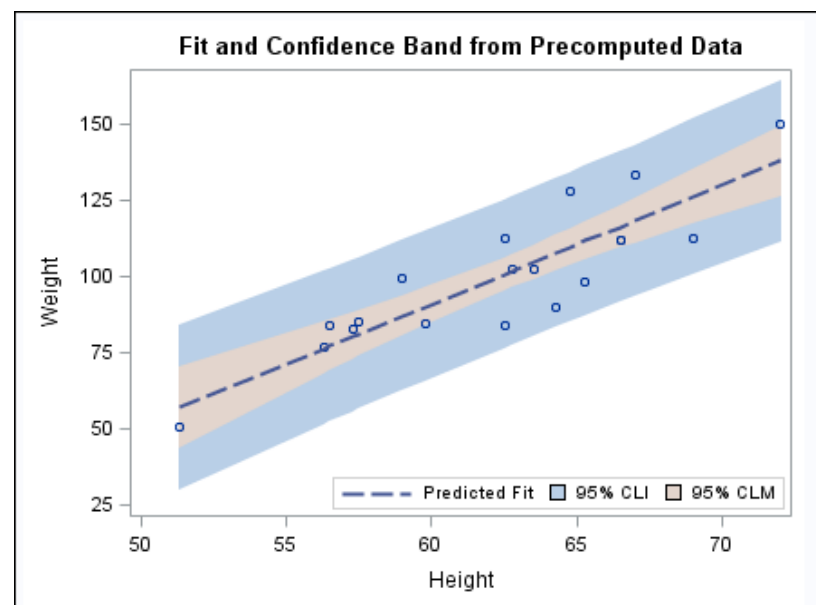
```
title;
```

Example 6: Creating Lines and Bands from Pre-Computed Data

Features: BAND statement
KEYLEGEND statement
SCATTER statement
SERIES statement

Sample library member: SGPLBND

This example shows how to use pre-computed data to create a scatter plot, fit line, and confidence bands. The data set was created by the REG procedure. This technique is useful for more complex fit models.

Output**Program**

```
proc sgplot data=sashelp.classfit;
  title "Fit and Confidence Band from Precomputed Data";
```



```

band x=height lower=lower upper=upper /
    legendlabel="95% CLI" name="band1";

band x=height lower=lowermean upper=uppermean /
    fillattrs=GraphConfidence2
    legendlabel="95% CLM" name="band2";
scatter x=height y=weight;
series x=height y=predict / lineattrs=GraphPrediction
    legendlabel="Predicted Fit" name="series";

keylegend "series" "band1" "band2" / location=inside
position=bottomright;
run;

title;

```

Program Description

Set the title and create the first band plot. The LEGENDLABEL= option in the BAND statement specifies the label for the band plot in the legend.

```

proc sgplot data=sashelp.classfit;
    title "Fit and Confidence Band from Precomputed Data";
    band x=height lower=lower upper=upper /
        legendlabel="95% CLI" name="band1";

```

Create the second band plot. The LEGENDLABEL= option specifies the label for the band plot in the legend. The FILLATTRS= option specifies the style element for the fill.

```

band x=height lower=lowermean upper=uppermean /
    fillattrs=GraphConfidence2
    legendlabel="95% CLM" name="band2";
scatter x=height y=weight;
series x=height y=predict / lineattrs=GraphPrediction
    legendlabel="Predicted Fit" name="series";

```

Create the scatter and series plots. The LINEATTRS= option in the SERIES statement specifies the style attribute for the series plot. The LEGENDLABEL= option in the SERIES statement specifies the legend label for the series plot.

Create a legend for the graph. The quoted strings specify the names of the plots that you want to include in the legend. The LOCATION= option places the legend inside of the plot area. The POSITION= option places the legend in the bottom right corner of the graph.

```

keylegend "series" "band1" "band2" / location=inside
position=bottomright;
run;

```

Cancel the title.

```

title;

```

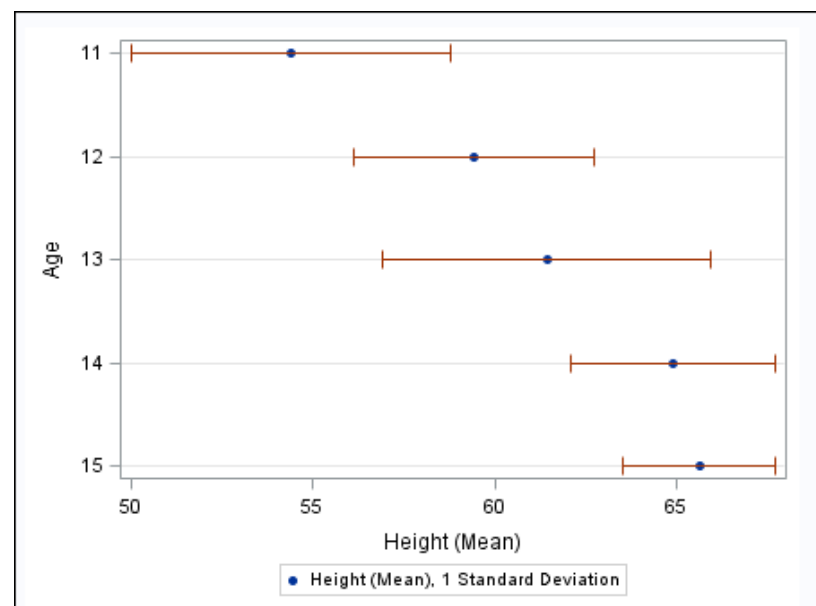
Example 7: Adding Statistical Limits to a Dot Plot

Features: DOT statement

Sample library member: SGPLDT1

This example shows a dot plot with a response variable and statistical limits. Each dot represents the mean for each value of the category variable, and bands represent the standard deviation.

Output



Program

```
proc sgplot data=sashelp.class(where=(age<16));
  dot age / response=height stat=mean
    limitstat=stddev numstd=1;
run;
```

Program Description

Create the dot plot. The RESPONSE= option specifies the response variable. The STAT= option specifies that the mean statistic is used to analyze the graph. The LIMITSTAT= option specifies that the limit statistic is the standard deviation. The NUMSTD= option specifies that one standard deviation is used.

```
proc sgplot data=sashelp.class(where=(age<16));
  dot age / response=height stat=mean
    limitstat=stddev numstd=1;
run;
```

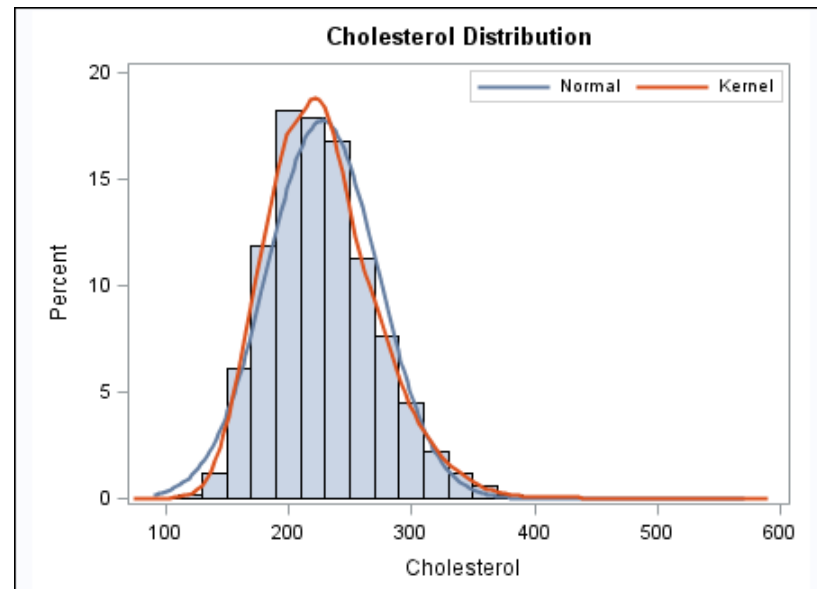
Example 8: Combining Histograms with Density Plots

Features: HISTOGRAM statement
DENSITY statement
KEYLEGEND statement

Sample library member: SGPLHST

This example shows a histogram combined with two density plots. One density plot uses a normal density estimate and the other density plot uses a kernel density estimate.

Output



Program

```
proc sgplot data=sashelp.heart;
  title "Cholesterol Distribution";
  histogram cholesterol;

  density cholesterol;
  density cholesterol / type=kernel;

  keylegend / location=inside position=topright;
run;

title;
```

Program Description

Set the title, set a label for the X axis, and create the histogram.

```
proc sgplot data=sashelp.heart;
  title "Cholesterol Distribution";
  histogram cholesterol;
```

Create the density plots. The TYPE= option specifies which density equation is used.

```
  density cholesterol;
  density cholesterol / type=kernel;
```

Position the Legend The LOCATION= option places the legend inside of the plot area. The POSITION= option places the legend at the top right.

```
  keylegend / location=inside position=topright;
run;
```

Cancel the title.

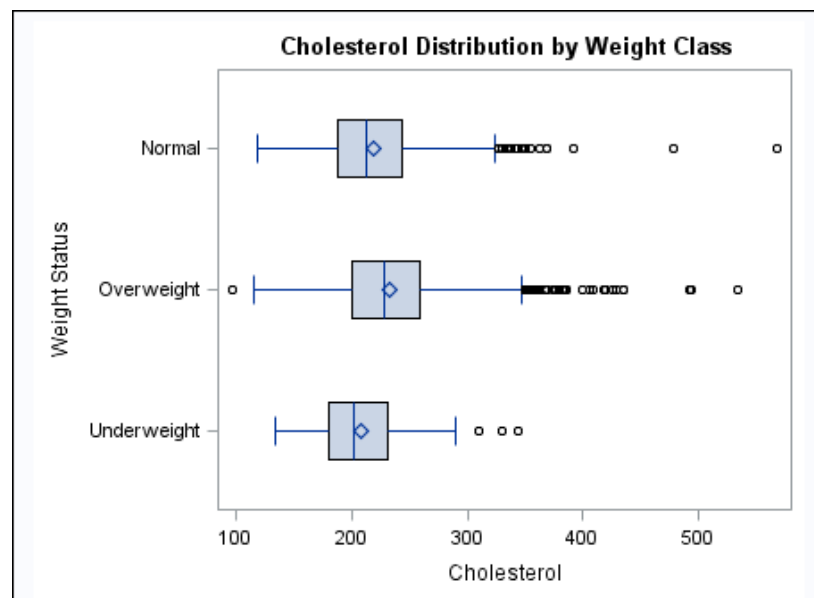
```
  title;
```

Example 9: Creating a Horizontal Box Plot

Features: HBOX statement
Sample library member: SGPLBOX

This example shows a horizontal box plot.

Output



Program

```
proc sgplot data=sashelp.heart;
  title "Cholesterol Distribution by Weight Class";
  hbox cholesterol / category=weight_status;
```

```
run;
title;
```

Program Description

Create the box plot. The CATEGORY= option specifies the category variable.

```
proc sgplot data=sashelp.heart;
  title "Cholesterol Distribution by Weight Class";
  hbox cholesterol / category=weight_status;
run;
title;
```

Example 10: Creating a Bar-Line Chart

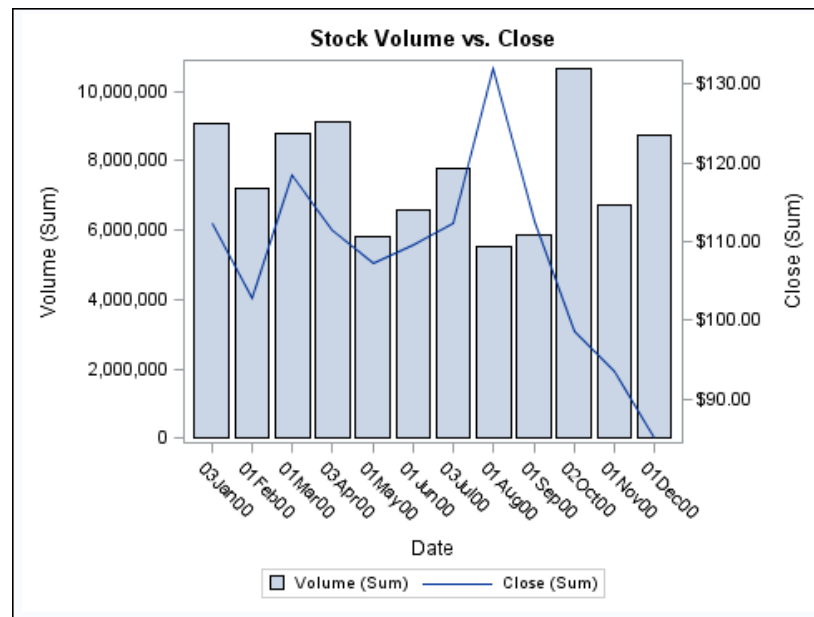
Features: VBAR statement
VLINE statement

Sample library member: SGPLBAR

Note: You can also combine the HBAR and HLINE statements to create a horizontal bar-line chart.

This example creates a bar-line chart.

Output



Program

```
proc sgplot data=sashelp.stocks (where=(date >= "01jan2000"d
  and date <= "01jan2001"d
  and stock = "IBM"));
```

```

title "Stock Volume vs. Close";
vbar date / response=volume;
vline date / response=close y2axis;
run;
title;

```

Program Description

Create the Bar-line Chart. The Y2AXIS option assigns the line plot to the Y2 axis.

```

proc sgplot data=sashelp.stocks (where=(date >= "01jan2000"d
                                     and date <= "01jan2001"d
                                     and stock = "IBM"));

title "Stock Volume vs. Close";
vbar date / response=volume;
vline date / response=close y2axis;
run;
title;

```

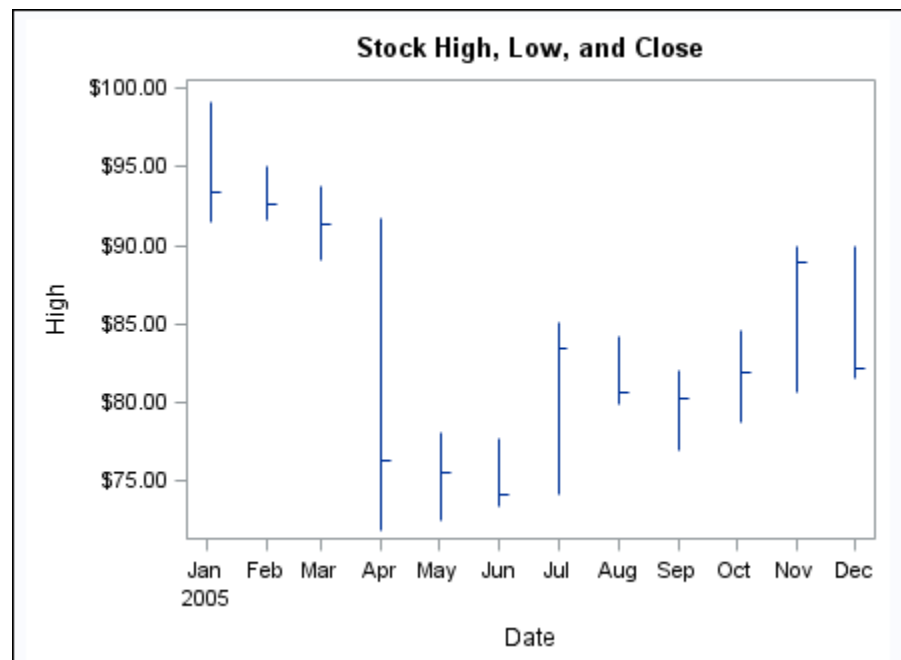
Example 11: Creating a High-Low Chart

Features: HIGHLOW statement

Sample library member: SGPLHILO

This example shows the high, low, and closing stock prices for a company during the year 2005.

Output



Program

```
title "Stock High, Low, and Close";
proc sgplot data=sashelp.stocks;
  where Date >= '01JAN2005'd and stock='IBM';
  highlow x=date high=high low=low
    / close=close;
run;
title;
```

Program Description

Create the high-low chart. The HIGH, LOW, and CLOSE variables are used in the HIGHLOW statement. In addition, the plot subsets the data by year and by company.

```
title "Stock High, Low, and Close";
proc sgplot data=sashelp.stocks;
  where Date >= '01JAN2005'd and stock='IBM';
  highlow x=date high=high low=low
    / close=close;
run;
title;
```


Chapter 8

SGRENDER Procedure

Overview: SGRENDER Procedure	513
Syntax: SGRENDER Procedure	513
PROC SGRENDER Statement	514
DYNAMIC Statement	515
Examples: SGRENDER Procedure	517
Example 1: Generating a Graph from a Simple GTL Template	517
Example 2: Defining Dynamic Variables	518

Overview: SGRENDER Procedure

The SGRENDER procedure produces graphical output from templates that are created with the Graph Template Language (GTL). The GTL is a comprehensive language for creating statistical graphics, which can be used to create customized layouts and graphs that are beyond the scope of the ODS Graphics procedures. For more information about the GTL, see the *SAS Graph Template Language: Reference* and the *SAS Graph Template Language: User's Guide*.

The SGRENDER procedure can also produce graphical output from SAS ODS Graphics Editor files. For more information about the editor, see the *SAS ODS Graphics Editor: User's Guide*.

Syntax: SGRENDER Procedure

Global statements:	BY, FORMAT, LABEL, ODS GRAPHICS, WHERE
---------------------------	--

```
PROC SGRENDER < option(s) >;
    DYNAMIC variable-assignment(s);
```

PROC SGRENDER Statement

Identifies the data set that contains the plot variables and the StatGraph template. You can also render a graph from a SAS ODS Graphics Editor (SGE) file. The statement also gives you the option to specify the name of the output object and the label for the output object.

Requirement: Requires an input data set and a template, or an SGE file.

Syntax

```
PROC SGRENDER TEMPLATE= statgraph-template <DATA= input-data-set>
    <OBJECT= object-name>
    <OBJECTLABEL=“text-string” > ;
PROC SGRENDER SGE= fileref | “file-name”
    <OBJECT= object-name>
    <OBJECTLABEL=“text-string” > ;
```

Summary of Optional Arguments

Render options

DATA=[input-data-set](#)

specifies the SAS data set that contains the variables to process.

OBJECT=[object-name](#)

specifies a name for the ODS output object.

OBJECTLABEL= “*text-string*”

specifies a description for the output image.

SGE=[fileref](#) | “*file-name*”

specifies the SAS ODS Graphics Editor (SGE) file to render.

Required Argument

TEMPLATE=[statgraph-template](#)

specifies a StatGraph template that defines one or more graphs. The SGRENDER procedure applies the StatGraph template to your data to create the output graphs.

Interaction: This argument is required except when you use the SGE= option.

Note: Specifying a template that is not a StatGraph template is not supported. If you specify a non-StatGraph template, then the SGRENDER procedure might produce unpredictable results.

Optional Arguments

DATA=[input-data-set](#)

specifies the SAS data set that contains the variables to process. By default, the procedure uses the most recently created SAS data set.

OBJECT=[object-name](#)

specifies a name for the ODS output object.

Alias: NAME=

Default: SGRender

Note: To specify the filename of the output image, use the IMAGENAME= option in the ODS GRAPHICS statement.

OBJECTLABEL= “*text-string*”

specifies a description for the output image. The description identifies the image in the following locations:

- the Results window
- the alternate text for the image in HTML output
- the table of contents that is created by the CONTENTS option in an ODS statement

The default description is “The SGRENDER Procedure”.

Alias: DES=, DESCRIPTION=

SGE= *fileref* | “*file-name*”

specifies the SAS ODS Graphics Editor (SGE) file to render. Specify a fileref or a filename for the SGE file. This option enables you to render an SGE file to any supported ODS destination. The output type depends on the OUTPUTFMT option of the ODS GRAPHICS statement and the open ODS destination.

When you use this option to render an input SGE file, ODS does not produce an output SGE file even if the SGE=ON option is specified in the ODS statement.

For information about generating SGE files, see the *SAS ODS Graphics Editor: User's Guide*.

Requirement: Although SGE rendering does not require a data set, the SGRENDER procedure requires a data reference. You might need to specify DATA=_NULL_ in the SGRENDER procedure if the _LAST_ data set variable has not been set by the SAS system. This typically happens at the beginning of a SAS session, before the creation of a data set.

DYNAMIC Statement

Defines the values for dynamic variables in your StatGraph template code.

See: [“Example 2: Defining Dynamic Variables” on page 518](#)

Syntax

DYNAMIC *variable-assignment(s)*;

Required Argument

variable-assignment(s)

specifies the values for one or more dynamic variables. The variables that you assign must also be declared in a DYNAMIC statement within the StatGraph template.

You can specify either a numeric value or a quoted text string. For example, the following statement defines one string variable and one numeric variable: **dynamic region="Europe" limit=3000;**

Details

Description

The DYNAMIC statement defines values for dynamic variables that are used in your StatGraph template.

You can either specify variables on multiple DYNAMIC statements, or specify all of your variables on a single DYNAMIC statement.

Note: If your template uses a dynamic variable to specify a required attribute, such as a variable name, and the name is misspelled or 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.

Special Dynamic Variables

Some special dynamic variables can be declared and referenced in a StatGraph template and do not need to be assigned in the DYNAMIC statement of the SGRENDER procedure. The following table describes those dynamic variables.

Table 8.1 Special Dynamic Variables

<code>_LIBNAME_</code>	Represents the name of the library that contains the data set.
<code>_MEMNAME_</code>	Represents the name of the library member that contains the data set.
<code>_BYLINE_</code>	Represents the complete BY line, when you specify a BY statement.
<code>_BYFOOTNOTE_</code>	When the BYLINE=FOOTNOTE option is specified in the ODS Graphics statement, the BY line information is set in this dynamic. Typically, you use this dynamic with the ENTRYFOOTNOTE statement in your template.
<code>_BYTITLE_</code>	When the BYLINE=TITLE option is specified in the ODS Graphics statement, the BY line information is set in this dynamic. Typically, you use this dynamic with the ENTRYTITLE statement in your template.
<code>_BYVAR_</code>	Represents the name of the first BY variable, when you specify a BY statement.
<code>_BYVARn_</code>	Represents the name of the n th BY variable, when you specify a BY statement with multiple variables.
<code>_BYVAL_</code>	Represents the first BY value, when you specify a BY statement.

`_BYVAL n _`

Represents the value of the n th BY variable, when you specify a BY statement with multiple variables.

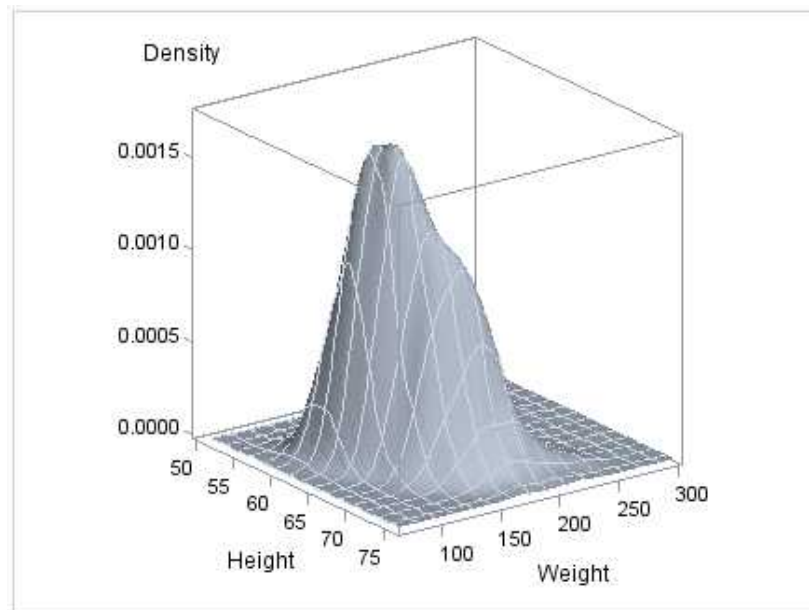
Examples: SGRENDER Procedure

Example 1: Generating a Graph from a Simple GTL Template

Features: GTL template
Sample library member: SGRENDR

This example creates a simple StatGraph template by using Graph Template Language and then generates a graph from the template by using the SGRENDER procedure.

Output



Program

```
proc template;
  define statgraph surface;
    begingraph;
      layout overlay3d;
      surfaceplotparm x=height y=weight z=density;
      endlayout;
    endgraph;
```

```

        end;
    run;

proc sgrender data=sashelp.gridded template=surface;
run;

```

Program Description

Create the Statgraph Template.

```

proc template;
    define statgraph surface;
        begingraph;
            layout overlay3d;
            surfaceplotparm x=height y=weight z=density;
            endlayout;
        endgraph;
    end;
run;

```

Generate the graphics output from the template.

```

proc sgrender data=sashelp.gridded template=surface;
run;

```

Example 2: Defining Dynamic Variables

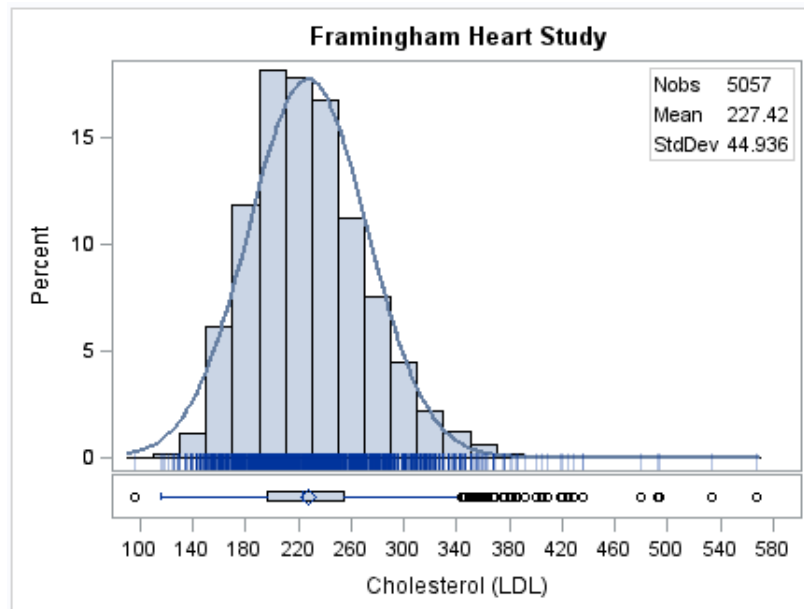
Features:	DYNAMIC Statement
Sample library member:	SGREND1

This example uses dynamic variables to set values within the StatGraph template. By using dynamic variables to set the variable names, variable labels, and other parameters, the StatGraph template can be used with different data sets.

The first PROC SGRENDER statement generates a graph for the SASHELP.HEART data set.

The second PROC SGRENDER statement generates multiple graph for the CARS data set by using BY grouping.

Output with Heart Data



Program Template and Heart Data

```
proc template;
  define statgraph distribution;
    dynamic VAR VARLABEL TITLE NORMAL _BYLINE_;
    begingraph;
      entrytitle TITLE;
      entrytitle _BYLINE_;
      layout lattice / columns=1 rows=2 rowgutter=2px
        rowweights=(.9 .1) columndatarange=union;
      columnaxes;
        columnaxis / label=VARLABEL;
      endcolumnaxes;
      layout overlay / yaxisopts=(offsetmin=.035);
        layout gridded / columns=2 border=true autoalign=(topleft topright);
        entry halign=left "Nobs";
          entry halign=left eval(strip(put(n(VAR),8.)));
        entry halign=left "Mean";
          entry halign=left eval(strip(put(mean(VAR),8.2)));
          entry halign=left "StdDev";
            entry halign=left eval(strip(put(stddev(VAR),8.3)));
        endlayout;
      histogram VAR / scale=percent;
      if (exists(NORMAL))
        densityplot VAR / normal( );
      endif;
      fringeplot VAR / datatransparency=.7;
      endlayout;
      boxplot y=VAR / orient=horizontal;
    endlayout;
  endgraph;
end;
run;
```

```
proc sgrender data=sashelp.heart template=distribution;
  dynamic var="cholesterol" varlabel="Cholesterol (LDL)" normal="yes"
    title="Framingham Heart Study";
run;
title;
```

Program Description

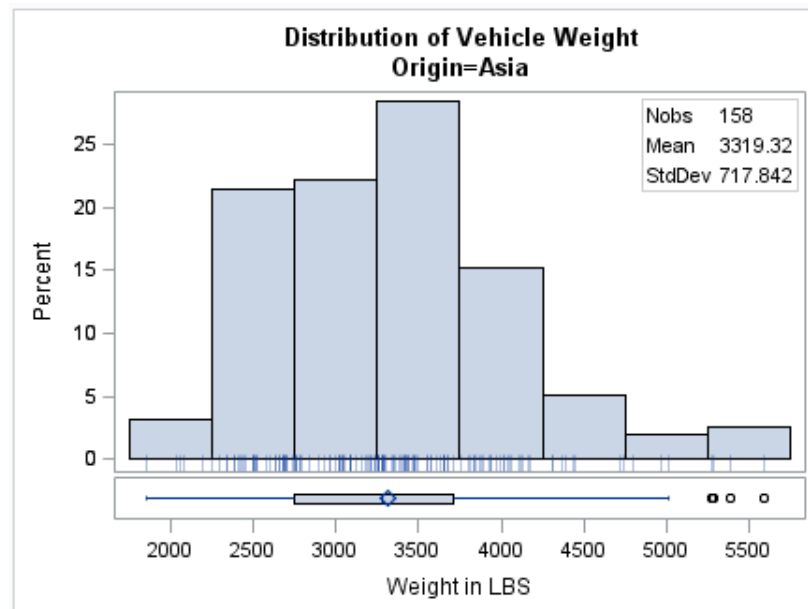
Create the Statgraph template.

```
proc template;
  define statgraph distribution;
    dynamic VAR VARLABEL TITLE NORMAL _BYLINE_;
    begingraph;
      entrytitle TITLE;
      entrytitle _BYLINE_;
      layout lattice / columns=1 rows=2 rowgutter=2px
        rowweights=(.9 .1) columndatarange=union;
      columnaxes;
        columnaxis / label=VARLABEL;
      endcolumnaxes;
      layout overlay / yaxisopts=(offsetmin=.035);
        layout gridded / columns=2 border=true autoalign=(topleft topright);
        entry halign=left "Nobs";
          entry halign=left eval(strip(put(n(VAR),8.)));
        entry halign=left "Mean";
          entry halign=left eval(strip(put(mean(VAR),8.2)));
          entry halign=left "StdDev";
          entry halign=left eval(strip(put(stddev(VAR),8.3)));
        endlayout;
      histogram VAR / scale=percent;
      if (exists(NORMAL))
        densityplot VAR / normal( );
      endif;
      fringeplot VAR / datatransparency=.7;
    endlayout;
    boxplot y=VAR / orient=horizontal;
  endlayout;
endgraph;
end;
run;
```

Generate the first graphics output from the template using the SASHELP.HEART data set. The DYNAMIC statement defines dynamic variables in the template.

```
proc sgrender data=sashelp.heart template=distribution;
  dynamic var="cholesterol" varlabel="Cholesterol (LDL)" normal="yes"
    title="Framingham Heart Study";
run;
title;
```


Output with Grouped Cars Data



Program for Grouped Cars Data

```
proc sort data=sashelp.cars out=cars;
  by origin;
run;

proc sgrender data=cars template=distribution;
  by origin;
  dynamic var="weight" varlabel="Weight in LBS"
           title="Distribution of Vehicle Weight";
run;
title;
```

Program Description

Sort the SASHELP.CARS data set. The data set must be sorted by the same variable that the following PROC SGRENDER block uses in its BY statement.

```
proc sort data=sashelp.cars out=cars;
  by origin;
run;
```

Generate the second graphics output from the template using the WORK.CARS data set. The BY statement generates multiple graphs for each value of the BY variable. The DYNAMIC statement defines dynamic variables in the template.

```
proc sgrender data=cars template=distribution;
  by origin;
  dynamic var="weight" varlabel="Weight in LBS"
           title="Distribution of Vehicle Weight";
run;
title;
```


Chapter 9

SGSCATTER Procedure

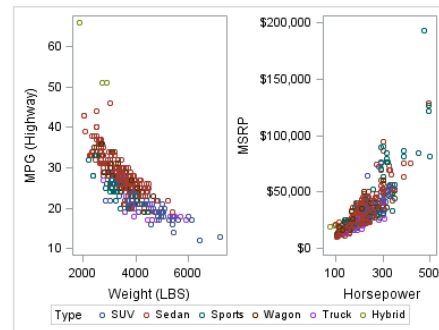
Overview: SGSCATTER Procedure	523
Concepts: SGSCATTER Procedure	525
Statements for Creating Panels	525
Legends in the SGSCATTER Procedure	527
Syntax: SGSCATTER Procedure	527
PROC SGSCATTER Statement	528
COMPARE Statement	529
MATRIX Statement	537
PLOT Statement	540
Examples: SGSCATTER Procedure	548
Example 1: Creating a Scatter Plot Matrix	548
Example 2: Creating a Graph with Multiple Independent Scatter Plots and Spline Curves	549
Example 3: Creating a Simple Comparative Panel	551
Example 4: Creating a Comparative Panel with Regression Fits and Confidence Ellipses	552

Overview: SGSCATTER Procedure

The SGSCATTER procedure creates a paneled graph of scatter plots for multiple combinations of variables, depending on the plot statement that you use. You can use options to overlay fit plots and ellipses on your scatter plots.

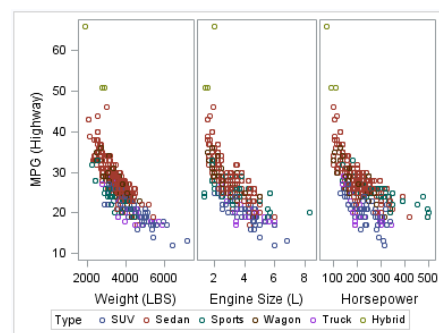
Here are examples of some types of graphs that the SGSCATTER procedure can create.

Table 9.1 Examples of Graphs that Can Be Generated by the SGSCATTER Procedure



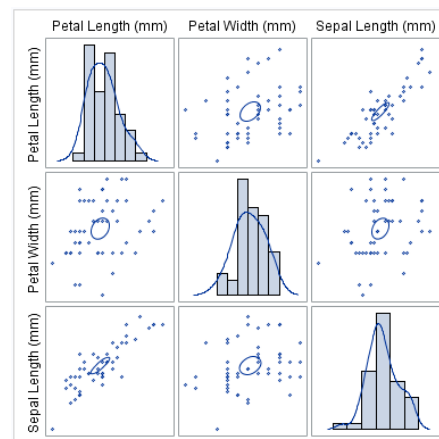
The following code creates a paneled graph of scatter plots:

```
proc sgscatter data=sashelp.cars;
  plot mpg_highway*weight msrp*horsepower
    / group=type;
run;
```



The following code creates a paneled graph of scatter plots with shared axes:

```
proc sgscatter data=sashelp.cars;
  compare y=mpg_highway
    x=(weight enginesize horsepower )
    / group=type;
run;
```



The following code creates a scatter plot matrix with prediction ellipses and a diagonal with histograms and density plots:

```
proc sgscatter data=sashelp.iris
  (where=(species eq "Virginica"));
matrix petallength petalwidth sepallength
  / ellipse=(type=mean)
  diagonal=(histogram kernel);
run;
```

Note: The graphs that you create with the SGSCATTER procedure can have many individual graph cells. As the number of cells increases, the overall graph size does not automatically increase. To increase the graph size, use the HEIGHT= and WIDTH= options of the ODS GRAPHICS statement.

Concepts: SGSCATTER Procedure

Statements for Creating Panels

About Creating Panels

The SGSCATTER procedure contains three statements that you can use to create a paneled graph of scatter plots:

- PLOT
- COMPARE
- MATRIX

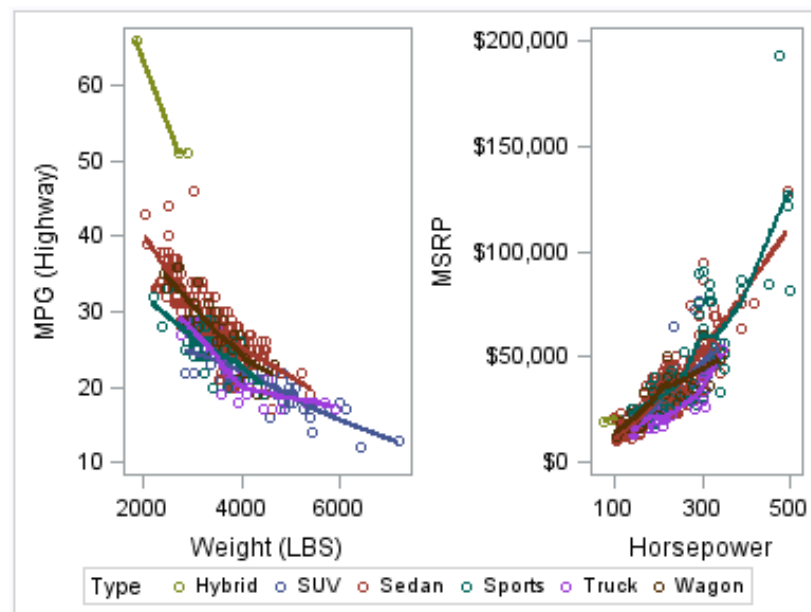
Each of the statements is specialized for creating a different type of paneled graph.

PLOT Statement

The PLOT statement is used to create a paneled graph of scatter plots where each graph cell has its own independent set of axes. Each variable pair that you specify in the PLOT statement creates an independent graph cell. You can also overlay fit plots and ellipses on each cell by using options.

For example, [Figure 9.1 on page 525](#) shows a graph that contains two independent cells. Each cell contains a scatter plot and a loess curve.

Figure 9.1 Example Graph from the PLOT Statement



By default, the axis ranges of each cell are independent from the other cells. However, you can use the UNISCALE= option to specify that all of the cells use the same axis ranges for the X axis, the Y axis, or both axes.

Note: It is possible to create a single scatter cell with the PLOT statement, but the SGPLOT procedure is better suited to creating a single-celled graph.

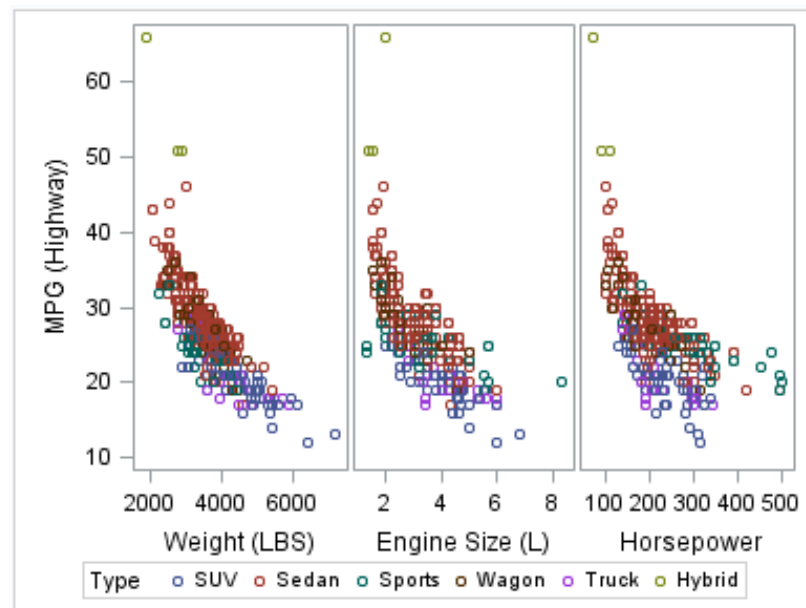
COMPARE Statement

The COMPARE statement is used to create a shared axis panel, also called an MxN matrix. The list of X and Y variables are crossed to create each cell in the graph. All cells in a row share the same row axis range. All cells in a column share the same column axis range.

When using the COMPARE statement, you can add fit plots and confidence ellipses to each cell in the panel by using options.

The COMPARE statement can also be used to do simple X or Y axis sharing by specifying only one X or Y variable, as in the following example:

Figure 9.2 Example Graph from the COMPARE Statement



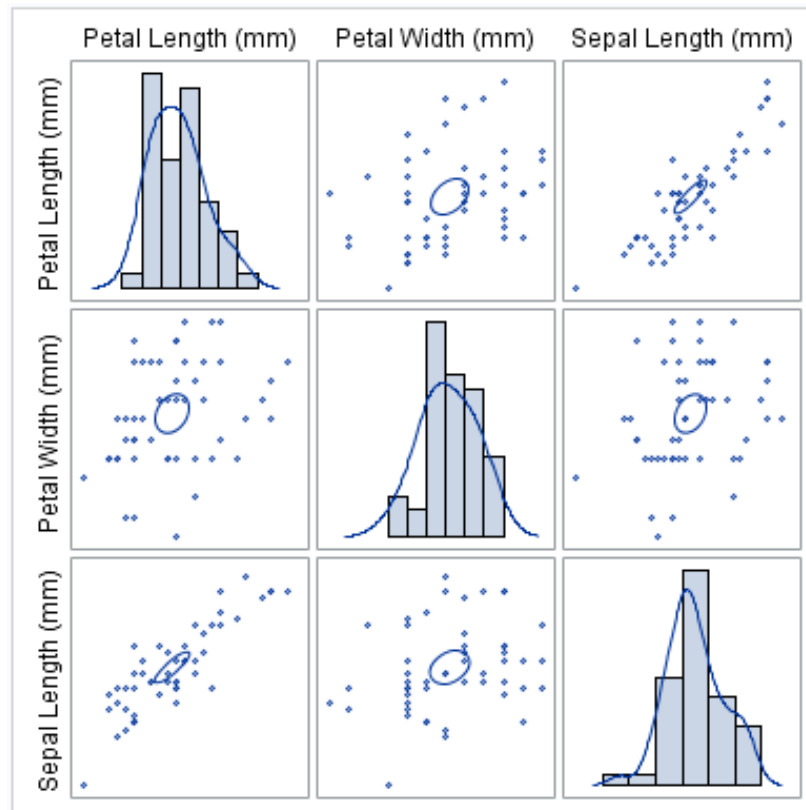
MATRIX Statement

The MATRIX statement is used to create a scatter plot matrix. Each of the variables that you specify are graphed against each other to create the graph. You can use the ELLIPSE option to overlay a confidence ellipse on each cell in the panel.

The MATRIX statement also has an option called DIAGONAL= that enables you to display the distribution of your variables in the diagonal cells of the matrix. You can place histograms in the diagonal cells, and overlay those histograms with normal density curves or kernel density estimates.

For example, the following example shows a graph with histograms and normal density curves in the diagonal cells.

Figure 9.3 Example Graph from the MATRIX Statement



Legends in the SGSCATTER Procedure

The SGSCATTER procedure creates a legend automatically when you specify a GROUP= variable. You can use the NOLEGEND option to disable the legend.

For all of the graph creation statements, you can use the LEGEND= option to specify the attributes of the legend.

Syntax: SGSCATTER Procedure

Requirement: One COMPARE, MATRIX, or PLOT statement is required.

Global statements: BY, FORMAT, LABEL, ODS GRAPHICS, TITLE and FOOTNOTE, WHERE

PROC SGSCATTER <options>;

COMPARE X= variable | (variable-1 ... variable-n) Y= variable |
(variable-1 ... variable-n) </options>;

MATRIX variable-1 variable-2 < ... variable-n> </options>;

PLOT plot-request(s) </options>;

PROC SGSCATTER Statement

Identifies the data set that contains the plot variables. The statement also gives you the option to specify a description and write template code to a file.

Requirement: An input data set is required.

Syntax

```
PROC SGSCATTER <DATA= input-data-set>
  < DATTRMAP= "attribute-map-data-set" >
  < DESCRIPTION= "text-string" >
  < PAD= dimension<units> | (pad-options)>
  < SGANNO= "annotation-data-set" >
  < TMPLOUT= "filename">;
```

Summary of Optional Arguments

DATA=*input-data-set*

specifies the SAS data set that contains the variables to process.

DATTRMAP=*attribute-map-data-set*

specifies the SG attribute map data set that you want to use with the SGSCATTER procedure.

DESCRIPTION=*"text-string"*

specifies a description for the output image.

PAD=*dimension*<*units*> | (*pad-options*)

specifies the amount of extra space that is reserved inside the border of an annotated graph.

SGANNO=*annotation-data-set*

specifies the SG annotation data set that you want to use.

TMPLOUT=*"filename"*

specifies a file destination for the template code that is generated by the SGSCATTER procedure.

Optional Arguments

DATA=*input-data-set*

specifies the SAS data set that contains the variables to process. By default, the procedure uses the most recently created SAS data set.

DATTRMAP=*attribute-map-data-set*

specifies the SG attribute map data set that you want to use with the SGSCATTER procedure. You specify this option only if you are using an attribute map to control visual attributes of the graph. For more information, see [“Using SG Attribute Maps to Control Visual Attributes”](#) on page 605.

DESCRIPTION=*"text-string"*

specifies a description for the output image. The description identifies the image in the following locations:

- the Results window

- the alternate text for the image in HTML output
- the table of contents that is created by the CONTENTS option in an ODS statement

The default description is “The SGSCATTER Procedure”.

Alias: DES

Notes:

You can disable the alternate text in HTML output by specifying an empty string. That is, **DESCRIPTION=** " ".

The name of the output image is specified by the IMAGENAME= option in the ODS GRAPHICS statement.

PAD= *dimension*<*units*> | (*pad-options*)

specifies the amount of extra space that is reserved inside the border of an annotated graph.

You specify this option only if you are using the SG annotation feature to annotate your graph. For more information, see [Chapter 10, “Annotating ODS Graphics,” on page 557](#).

This option creates margins around the graph for company logos, annotated notes, and so on. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see [“Units of Measurement” on page 659](#).

Use pad options to create non-uniform padding. Edges that are not assigned padding are padded with the default amount.

pad-options can be one or more of the following:

LEFT= *dimension*<*units*>

specifies the amount of extra space to add to the left edge.

RIGHT= *dimension*<*units*>

specifies the amount of extra space to add to the right edge.

TOP= *dimension*<*units*>

specifies the amount of extra space to add to the top edge.

BOTTOM= *dimension*<*units*>

specifies the amount of extra space to add to the bottom edge.

SGANNO= *annotation-data-set*

specifies the SG annotation data set that you want to use. You specify this option only if you are using the SG annotation feature to annotate your graph. For more information, see [Chapter 10, “Annotating ODS Graphics,” on page 557](#).

TMPLOUT= "*filename*"

specifies a file destination for the template code that is generated by the SGSCATTER procedure.

COMPARE Statement

Creates a comparative panel of scatter plots with shared axes.

Examples: [“Example 3: Creating a Simple Comparative Panel” on page 551](#)
 [“Example 4: Creating a Comparative Panel with Regression Fits and Confidence Ellipses” on page 552](#)

Syntax

COMPARE X= *variable* | (*variable-1* ... *variable-n*) Y= *variable* | (*variable-1* ... *variable-n*) </options>;

Summary of Optional Arguments

- DATALABEL** <= *variable*>
displays a label for each data point.
- ELLIPSE** <= (*options*)>
Adds a confidence or prediction ellipse to the scatter plot.
- GRID**
creates grid lines for each tick on both axes.
- GROUP**= *variable*
specifies a classification variable to divide the values into groups.
- JOIN** <= (*options*)>
specifies that line segments join all of the scatter points.
- LEGEND**= (*options*)
specifies the appearance of the legend for the scatter plot.
- LOESS** <= (*options*)>
Adds a loess fit to the scatter plot.
- MARKERATTRS**= *style-element* <(*options*)> | (*options*)
specifies the appearance of the markers in the plot.
- NOLEGEND**
removes the legend from the plot.
- PBSPLINE** <= (*options*)>
adds a fitted, penalized B-spline curve to the scatter plot.
- REFTICKS**
duplicates the tick marks from the X and Y axes on the opposite sides of the graph.
- REG** <= (*options*)>
adds a regression fit to the scatter plot.
- SPACING** = *n*
specifies the amount of spacing (in pixels) that is placed between the cells in the graph.
- TRANSPARENCY**= *numeric-value*
specifies the degree of transparency for the plot components.

Required Arguments

- X**= *variable* | (*variable-1*) ... (*variable-n*)
specifies one or more variables for the X axis. To specify more than one variable, enclose the list of variables in parentheses.
- Y**= *variable* | (*variable-1*) ... (*variable-n*)
specifies the one or more variables for the Y axis. To specify more than one variable, enclose the list of variables in parentheses.

Optional Arguments

- DATALABEL** <= *variable*>
displays a label for each data point. If you specify a variable, the values of that variable are used for the data labels. If you do not specify a variable, the value of the Y variable is used for the data label.

ELLIPSE <= (*options*)>

Adds a confidence or prediction ellipse to the scatter plot. *options* can be one or more of the following:

ALPHA= *numeric-value*

specifies the confidence level for the ellipse. Specify a number between 0.00 (100% confidence) and 1.00 (0% confidence).

Default: .05

TYPE= MEAN | PREDICTED

specifies the type of ellipse. MEAN specifies a confidence ellipse for the population mean. PREDICTED specifies a prediction ellipse for a new observation. Both ellipse types assume bivariate normal distribution.

Default: PREDICTED

Restriction: To use this option, all of the X and Y variables must be numeric.

Interaction: The GROUP option does not affect the creation of ellipses. The SGSCATTER procedure always uses all of the data points to calculate the confidence or prediction ellipse.

GRID

creates grid lines for each tick on both axes.

GROUP= *variable*

specifies a classification variable to divide the values into groups. If a fit line is requested, then the GROUP= variable is also applied to the fit plot unless you specify the NOGROUP suboption in the option for the fit plot.

JOIN <= (*options*)>

specifies that line segments join all of the scatter points. The line segments connect the scatter points in increasing order along the X axis. The data order of the input data set has no effect on the order of the lines.

You can specify the following options:

LINEATTRS= *style-element* <(options)> | (options)

specifies the appearance of the join lines. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphFit style element in the current style.

For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see “[Line Patterns](#)” on page 663.

Default: For ungrouped data, the default line pattern is specified by the LineStyle attribute of the GraphFit style element in the current style.

For grouped data, the default line pattern is specified by the `LineStyle` attribute of the `GraphData1` ... `GraphDatan` style elements in the current style.

THICKNESS= *n* *<units>*

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: For ungrouped data, the default line thickness is specified by the `LineThickness` attribute of the `GraphFit` style element in the current style. For grouped data, the default line thickness is specified by the `LineThickness` attribute of the `GraphData1` ... `GraphDatan` style elements in the current style.

LEGEND= (*options*)

specifies the appearance of the legend for the scatter plot.

options can be one or more of the following:

ACROSS= *n*

specifies the number of columns in the legend.

DOWN= *n*

specifies the number of rows in the legend.

Interaction: If you specify both the **ACROSS=** and **DOWN=** suboptions, then the **DOWN=** suboption has no effect.

NOBORDER

removes the border from the legend.

NOTITLE

removes the label from the legend.

POSITION= *position-value*

specifies the position of the legend within the graph. The positions are as follows:

BOTTOM

places the legend at the bottom of the graph.

LEFT

places the legend at the left side of the graph.

RIGHT

places the legend at the right side of the graph.

TOP

places the legend at the top of the graph.

Default: **BOTTOM**

Note: If you specify more than one legend with the same position, then all of your legends are placed at that position.

TITLE= *"text-string"*

specifies the label for the legend. By default, the label of the group variable is used.

LOESS *<= (options)>*

Adds a loess fit to the scatter plot. You can specify the following options:

ALPHA= *numeric-value*

specifies the confidence level for the confidence limits. Specify a number between 0.00 (100% confidence) and 1.00 (0% confidence).

Default: .05

Interaction: This option has no effect if you do not specify the CLM option.

CLM

creates confidence limits for a mean predicted value for each observation.

DEGREE= 1 | 2

specifies the degree of the local polynomials to use for each local regression. 1 specifies a linear fit and 2 specifies a quadratic fit.

Default: 1

INTERPOLATION= CUBIC | LINEAR

specifies the degree of the interpolating polynomials that are used for blending local polynomial fits at the kd tree vertices.

Default: CUBIC

LINEATTRS= *style-element* <(options)> | (options)

specifies the appearance of the fit line. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphFit style element in the current style.

For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see “Line Patterns” on page 663.

Default: For ungrouped data, the default line pattern is specified by the LineStyle attribute of the GraphFit style element in the current style.

For grouped data, the default line pattern is specified by the LineStyle attribute of the GraphData1 ... GraphData*n* style elements in the current style.

THICKNESS= *n* <*units*>

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: For ungrouped data, the default line thickness is specified by the LineThickness attribute of the GraphFit style element in the current style.

For grouped data, the default line thickness is specified by the LineThickness attribute of the GraphData1 ... GraphData*n* style elements in the current style.

NOGROUP

specifies that the fit does not use the group variable from the scatter plot.

SMOOTH= *numeric-value*

specifies a smoothing parameter value. If you do not specify a value, the value is determined automatically.

Restriction: To use this option, all of the X and Y variables must be numeric.

MARKERATTRS= *style-element* <(options)> | (options)

specifies the appearance of the markers in the plot. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the markers. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphDataDefault style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

SIZE= *n* <units>

specifies the size of the markers. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default marker size is specified by the MarkerSize attribute of the GraphDataDefault style element in the current style.

SYMBOL= *symbol-name*

specifies the symbol for the markers. For a list of valid marker symbols, see “Marker Symbols” on page 661.

Default: For ungrouped data, the default marker symbol is specified by the MarkerSymbol attribute of the GraphDataDefault style element in the current style. For grouped data, the default marker symbol is specified by the MarkerSymbol attribute of the GraphData1 ... GraphData*n* style elements in the current style.

NOLEGEND

removes the legend from the plot.

PBSPLINE <= (options)>

adds a fitted, penalized B-spline curve to the scatter plot. *options* can be one or more of the following:

ALPHA= *numeric-value*

specifies the confidence level for the confidence limits. Specify a number between 0.00 (100% confidence) and 1.00 (0% confidence).

Default: .05

Interaction: This option has no effect if you do not specify either the CLI option or the CLM option.

CLI

creates confidence limits for individual predicted values for each observation.

CLM

creates confidence limits for a mean predicted value for each observation.

DEGREE= *n*

specifies the degree of the spline transformation.

Default: 3

LINEATTRS= *style-element* <(options)> | (options)

specifies the appearance of the curve line. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphFit style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see “Line Patterns” on page 663.

Default: For ungrouped data, the default line pattern is specified by the LineStyle attribute of the GraphFit style element in the current style. For grouped data, the default line pattern is specified by the LineStyle attribute of the GraphData1 ... GraphData*n* style elements in the current style.

THICKNESS= *n* <*units*>

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: For ungrouped data, the default line thickness is specified by the LineThickness attribute of the GraphFit style element in the current style. For grouped data, the default line thickness is specified by the LineThickness attribute of the GraphData1 ... GraphData*n* style elements in the current style.

NKNOTS= *n*

specifies the number of evenly spaced internal knots.

Default: 100

NOGROUP

specifies that the curve does not use the group variable from the scatter plot.

SMOOTH= *numeric-value*

specifies a smoothing parameter value. If you do not specify this option, then a smoothing value is determined automatically.

Restriction: To use this option, all of the X and Y variables must be numeric.

REFTICKS

duplicates the tick marks from the X and Y axes on the opposite sides of the graph.

REG <= (options)>

adds a regression fit to the scatter plot. *options* can be one or more of the following:

ALPHA= *numeric-value*

specifies the confidence level for the confidence limits. Specify a number between 0.00 (100% confidence) and 1.00 (0% confidence).

Default: .05

Interaction: This option has no effect if you do not specify either the CLI option or the CLM option.

CLI

creates confidence limits for individual predicted values for each observation.

CLM

creates confidence limits for a mean predicted value for each observation.

DEGREE= *n*

specifies the degree of the polynomial fit. For example, 1 specifies a linear fit, 2 specifies a quadratic fit, and 3 specifies a cubic fit.

Default: 1

LINEATTRS= *style-element* <(options)> | (options)

specifies the appearance of the fit line. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphFit style element in the current style.
For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see “Line Patterns” on page 663.

Default: For ungrouped data, the default line pattern is specified by the LineStyle attribute of the GraphFit style element in the current style.
For grouped data, the default line pattern is specified by the LineStyle attribute of the GraphData1 ... GraphData*n* style elements in the current style.

THICKNESS= *n* <*units*>

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: For ungrouped data, the default line thickness is specified by the LineThickness attribute of the GraphFit style element in the current style.
For grouped data, the default line thickness is specified by the LineThickness attribute of the GraphData1 ... GraphData*n* style elements in the current style.

NOGROUP

specifies that the fit does not use the group variable from the scatter plot.

Restriction: To use this option, all of the X and Y variables must be numeric.

SPACING = *n*

specifies the amount of spacing (in pixels) that is placed between the cells in the graph.

Default: 0

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the plot components. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default: 0.0

MATRIX Statement

Creates a scatter plot matrix.

Example: [“Example 1: Creating a Scatter Plot Matrix” on page 548](#)

Syntax

MATRIX *numeric-variable-1 numeric-variable-2 < ... numeric-variable-n> </ options>;*

Summary of Optional Arguments

DATALABEL= *variable*

specifies a variable that is used to create data labels for each point in the plot.

DIAGONAL= (*graph-list*)

adds graphs to the diagonal cells of the plot matrix.

ELLIPSE <= (*options*)>

Adds a confidence or prediction ellipse to each cell that contains a scatter plot.

GROUP= *variable*

specifies a classification variable to divide the values into groups.

LEGEND= (*options*)

specifies the appearance of the legend for the scatter plot.

MARKERATTRS= *style-element* <(*options*)> | (*options*)

specifies the appearance of the markers in the plot.

NOLEGEND

removes the legend from the graph.

START= BOTTOMLEFT | TOPLEFT

specifies whether the diagonal starts in the top left corner or the bottom left corner.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the plot components.

Required Argument

numeric-variable-1 numeric-variable-2 < ... numeric-variable-n >

specifies two or more numeric variables for the matrix.

Optional Arguments**DATALABEL= *variable***

specifies a variable that is used to create data labels for each point in the plot.

DIAGONAL= (*graph-list*)

adds graphs to the diagonal cells of the plot matrix. If you do not specify the DIAGONAL option, the diagonal cells contain the variable names.

graph-list can be one or more of the following:

HISTOGRAM

specifies a histogram.

KERNEL

specifies a kernel density estimate.

NORMAL

specifies a normal density curve.

ELLIPSE <= (*options*)>

Adds a confidence or prediction ellipse to each cell that contains a scatter plot.

options can be one or more of the following:

ALPHA= *numeric-value*

specifies the confidence level for the ellipse. Specify a number between 0.00 (100% confidence) and 1.00 (0% confidence).

Default: .05

TYPE= MEAN | PREDICTED

specifies the type of ellipse. MEAN specifies a confidence ellipse for the population mean. PREDICTED specifies a prediction ellipse for a new observation. Both ellipse types assume bivariate normal distribution.

Default: PREDICTED

Interaction: The GROUP option does not affect the creation of ellipses. The SGSCATTER procedure always uses all of the data points to calculate the confidence or prediction ellipse.

GROUP= *variable*

specifies a classification variable to divide the values into groups. If a fit line is requested, then the GROUP= variable is also applied to the fit plot unless you specify the NOGROUP suboption in the option for the fit plot.

LEGEND= (*options*)

specifies the appearance of the legend for the scatter plot.

options can be one or more of the following:

ACROSS= *n*

specifies the number of columns in the legend.

DOWN= *n*

specifies the number of rows in the legend.

Interaction: If you specify both the ACROSS= and DOWN= suboptions, then the DOWN= suboption has no effect.

NOBORDER

removes the border from the legend.

NOTITLE

removes the label from the legend.

POSITION= *position-value*

specifies the position of the legend within the graph. The values are as follows:

BOTTOM

places the legend at the bottom of the graph.

LEFT

places the legend at the left side of the graph.

RIGHT

places the legend at the right side of the graph.

TOP

places the legend at the top of the graph.

Default: BOTTOM

Note: If you specify more than one legend with the same position, then all of your legends are placed at that position.

TITLE= "*text-string*"

specifies the label for the legend. By default, the label of the group variable is used.

MARKERATTRS= *style-element* <(options)> | (options)

specifies the appearance of the markers in the plot. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the markers. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see "Color-Naming Schemes" in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphDataDefault style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

SIZE= *n* <*units*>

specifies the size of the markers. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see "Units of Measurement" on page 659.

Default: The default marker size is specified by the MarkerSize attribute of the GraphDataDefault style element in the current style.

SYMBOL= *symbol-name*

specifies the symbol for the markers. For a list of valid marker symbols, see "Marker Symbols" on page 661.

Default: For ungrouped data, the default marker symbol is specified by the MarkerSymbol attribute of the GraphDataDefault style element in the current style. For grouped data, the default marker symbol is specified by the MarkerSymbol attribute of the GraphData1 ... GraphData*n* style elements in the current style.

NOLEGEND

removes the legend from the graph.

START= BOTTOMLEFT | TOPLEFT

specifies whether the diagonal starts in the top left corner or the bottom left corner.

Default: TOPLEFT

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the plot components. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default: 0.0

PLOT Statement

Creates a paneled graph that contains multiple independent scatter plots.

Example: [“Example 2: Creating a Graph with Multiple Independent Scatter Plots and Spline Curves” on page 549](#)

Syntax

PLOT *plot-request(s)* *</ options>*;

Summary of Optional Arguments

ATTRID= *variable*

specifies the value of the ID variable in an attribute map data set.

COLUMNS= *n*

specifies the number of columns in the graph.

DATALABEL *<= variable>*

displays a label for each data point.

ELLIPSE *<= (options)>*

adds a confidence or prediction ellipse to the scatter plot.

GRID

creates grid lines for each tick on both axes.

GROUP= *variable*

specifies a classification variable to divide the values into groups.

JOIN *<= (options)>*

specifies that line segments join all of the scatter points.

LEGEND= *(options)*

specifies the appearance of the legend for the scatter plot.

LOESS *<= (options)>*

adds a loess fit to the scatter plot.

MARKERATTRS= *style-element <(options)> | (options)*

specifies the appearance of the markers in the plot.

NOLEGEND

removes the legend from the graph.

PBSPLINE *<= (options)>*

adds a fitted, penalized B-spline curve to the scatter plot.

REFTICKS

duplicates the tick marks from the X and Y axes on the opposite sides of each cell.

REG *<= (options)>*

adds a regression fit to the scatter plot.

ROWS= *n*

specifies the number of rows in the graph.

SPACING = *n*

specifies the amount of spacing (in pixels) that is placed between the cells in the graph, if the PLOT statement creates multiple cells.

TRANSPARENCY = *numeric-value*

specifies the degree of transparency for the plot components.

UNISCALE = X | Y | ALL

specifies that the X axis, Y axis, or both axes are scaled uniformly for all of the cells in the graph.

Required Argument***plot-request-1 < ... plot-request-n>***

specifies one or more plot requests. Each plot request specifies the variables to plot and produces a separate cell. All variables must be in the input data set. Multiple plot requests are separated with blanks. You can plot character or numeric variables. A plot request can be either of these:

y-variable*x-variable

plots the values of two variables.

y-variable

variable plotted on the left vertical axis.

x-variable

variable plotted on the horizontal axis.

(y-variable(s))*(x-variable(s))

plots the values of two or more variables and produces a separate cell for each combination of Y and X variables. That is, each Y*X pair is plotted on a separate set of axes.

y-variable(s)

variables plotted on the left vertical axes.

x-variable(s)

variables plotted on the horizontal axes.

If you use only one *y-variable* or only one *x-variable*, omit the parentheses for that variable, for example,

```
plot (temp rain)*month;
```

This plot request produces two cells, one for TEMP and MONTH and one for RAIN and MONTH.

Optional Arguments**ATTRID = *variable***

specifies the value of the ID variable in an attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph. For more information, see [“Using SG Attribute Maps to Control Visual Attributes” on page 605](#).

COLUMNS = *n*

specifies the number of columns in the graph.

DATALABEL <= *variable*>

displays a label for each data point. If you specify a variable, the values of that variable are used for the data labels. If you do not specify a variable, the value of the Y variable is used for the data label.

ELLIPSE <= (*options*)>

adds a confidence or prediction ellipse to the scatter plot. *options* can be one or more of the following:

ALPHA= *n*

specifies the confidence level for the ellipse. Specify a number between 0.00 (100% confidence) and 1.00 (0% confidence).

Default: .05

TYPE= MEAN | PREDICTED

specifies the type of ellipse. MEAN specifies a confidence ellipse for the population mean. PREDICTED specifies a prediction ellipse for a new observation. Both ellipse types assume bivariate normal distribution.

Default: PREDICTED

Restriction: To use this option, all of the X and Y variables must be numeric.

Interaction: The GROUP option does not affect the creation of ellipses. The SGSCATTER procedure always uses all of the data points to calculate the confidence or prediction ellipse.

GRID

creates grid lines for each tick on both axes.

GROUP= *variable*

specifies a classification variable to divide the values into groups. If a fit line is requested, then the GROUP= variable is also applied to the fit plot unless you specify the NOGROUP suboption in the option for the fit plot.

JOIN <= (*options*)>

specifies that line segments join all of the scatter points. The line segments connect the scatter points in increasing order along the X axis. The data order of the input data set has no effect on the order of the lines.

You can specify the following options:

LINEATTRS= *style-element* <(options)> | (options)

specifies the appearance of the join lines. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphFit style element in the current style.

For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see “Line Patterns” on page 663.

Default: For ungrouped data, the default line pattern is specified by the LineStyle attribute of the GraphFit style element in the current style.

For grouped data, the default line pattern is specified by the `LineStyle` attribute of the `GraphData1 ... GraphData n` style elements in the current style.

THICKNESS= n <units>

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see [“Units of Measurement” on page 659](#).

Default: For ungrouped data, the default line thickness is specified by the `LineThickness` attribute of the `GraphFit` style element in the current style. For grouped data, the default line thickness is specified by the `LineThickness` attribute of the `GraphData1 ... GraphData n` style elements in the current style.

LEGEND= (*options*)

specifies the appearance of the legend for the scatter plot. *options* can be one or more of the following:

ACROSS= n

specifies the number of columns in the legend.

DOWN= n

specifies the number of rows in the legend.

Interaction: This suboption has no effect if you also specify the **ACROSS=** suboption.

LOCATION= CELL | OUTSIDE

specifies whether the legend is placed inside an empty cell (**CELL**) in the plot area or outside of the plot area (**OUTSIDE**). The **LOCATION=CELL** suboption should be specified only when the graph contains an empty cell.

NOBORDER

removes the border from the legend.

NOTITLE

removes the label from the legend.

POSITION= *position-value*

specifies the position of the legend within the graph. The positions are as follows:

BOTTOM

places the legend at the bottom of the graph.

LEFT

places the legend at the left side of the graph.

RIGHT

places the legend at the right side of the graph.

TOP

places the legend at the top of the graph.

Default: **BOTTOM**

Interaction: This suboption has no effect if you also specify **LOCATION=CELL**.

Note: If you specify more than one legend with the same position, then all of your legends are placed at that position.

TITLE= *"text-string"*

specifies the label for the legend. By default, the label of the group variable is used.

LOESS <= (*options*)>

adds a loess fit to the scatter plot. You can specify the following options:

ALPHA= *numeric-value*

specifies the confidence level for the confidence limits. Specify a number between 0.00 (100% confidence) and 1.00 (0% confidence).

Default: .05

Interaction: This option has no effect if you do not specify the CLM option.

CLM

creates confidence limits for a mean predicted value for each observation.

DEGREE= 1 | 2

specifies the degree of the local polynomials to use for each local regression. 1 specifies a linear fit and 2 specifies a quadratic fit.

Default: 1

INTERPOLATION= CUBIC | LINEAR

specifies the degree of the interpolating polynomials that are used for blending local polynomial fits at the kd tree vertices.

Default: CUBIC

LINEATTRS= *style-element* <(options)> | (options)

specifies the appearance of the fit line. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphFit style element in the current style.
For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see “Line Patterns” on page 663.

Default: For ungrouped data, the default line pattern is specified by the LineStyle attribute of the GraphFit style element in the current style.
For grouped data, the default line pattern is specified by the LineStyle attribute of the GraphData1 ... GraphData*n* style elements in the current style.

THICKNESS= *n* <*units*>

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: For ungrouped data, the default line thickness is specified by the LineThickness attribute of the GraphFit style element in the current style.

For grouped data, the default line thickness is specified by the LineThickness attribute of the GraphData1 ... GraphData*n* style elements in the current style.

NOGROUP

specifies that the fit does not use the group variable from the scatter plot.

SMOOTH= *numeric-value*

specifies a smoothing parameter value. If you do not specify a value, then a smoothing value is determined automatically.

Restriction: To use this option, all of the X and Y variables must be numeric.

MARKERATTRS= *style-element* <(options)> | (options)

specifies the appearance of the markers in the plot. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the markers. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphDataDefault style element in the current style. For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

SIZE= *n* <units>

specifies the size of the markers. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: The default marker size is specified by the MarkerSize attribute of the GraphDataDefault style element in the current style.

SYMBOL= *symbol-name*

specifies the symbol for the markers. For a list of valid marker symbols, see “Marker Symbols” on page 661.

Default: For ungrouped data, the default marker symbol is specified by the MarkerSymbol attribute of the GraphDataDefault style element in the current style. For grouped data, the default marker symbol is specified by the MarkerSymbol attribute of the GraphData1 ... GraphData*n* style elements in the current style.

NOLEGEND

removes the legend from the graph.

PBSPLINE <= (options)>

adds a fitted, penalized B-spline curve to the scatter plot. *options* can be one or more of the following:

ALPHA= *numeric-value*

specifies the confidence level for the confidence limits. Specify a number between 0.00 (100% confidence) and 1.00 (0% confidence).

Default: .05

Interaction: This option has no effect if you do not specify either the CLI option or the CLM option.

CLI

creates confidence limits for individual predicted values for each observation.

CLM

creates confidence limits for a mean predicted value for each observation.

DEGREE= *n*

specifies the degree of the spline transformation.

Default: 3

LINEATTRS= *style-element* <(options)> | (options)

specifies the appearance of the curve line. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphFit style element in the current style.
For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see “Line Patterns” on page 663.

Default: For ungrouped data, the default line pattern is specified by the LineStyle attribute of the GraphFit style element in the current style.
For grouped data, the default line pattern is specified by the LineStyle attribute of the GraphData1 ... GraphData*n* style elements in the current style.

THICKNESS= *n* <*units*>

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: For ungrouped data, the default line thickness is specified by the LineThickness attribute of the GraphFit style element in the current style.
For grouped data, the default line thickness is specified by the LineThickness attribute of the GraphData1 ... GraphData*n* style elements in the current style.

NKNOTS= *n*

specifies the number of evenly spaced internal knots.

Default: 100

NOGROUP

specifies that the curve does not use the group variable from the scatter plot.

SMOOTH= *numeric-value*

specifies a smoothing parameter value. If you do not specify this option, then a smoothing value is determined automatically.

Restriction: To use this option, all of the X and Y variables must be numeric.

REFTICKS

duplicates the tick marks from the X and Y axes on the opposite sides of each cell.

REG <=*options*>

adds a regression fit to the scatter plot. *options* can be one or more of the following:

ALPHA= *numeric-value*

specifies the confidence level for the confidence limits. Specify a number between 0.00 (100% confidence) and 1.00 (0% confidence).

Default: .05

Interaction: This option has no effect if you do not specify either the CLI option or the CLM option.

CLI

creates confidence limits for individual predicted values for each observation.

CLM

creates confidence limits for a mean predicted value for each observation.

DEGREE= *n*

specifies the degree of the polynomial fit. For example, 1 specifies a linear fit, 2 specifies a quadratic fit, and 3 specifies a cubic fit.

Default: 1

LINEATTRS= *style-element* <(options)> | (options)

specifies the appearance of the fit line. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

options can be one or more of the following:

COLOR= *color*

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: For ungrouped data, the default color is specified by the ContrastColor attribute of the GraphFit style element in the current style.
For grouped data, the default color is specified by the ContrastColor attribute of the GraphData1 ... GraphData*n* style elements in the current style.

PATTERN= *line-pattern*

specifies the line pattern for the line. You can reference SAS patterns by number or by name. For a list of line patterns, see “Line Patterns” on page 663.

Default: For ungrouped data, the default line pattern is specified by the LineStyle attribute of the GraphFit style element in the current style.
For grouped data, the default line pattern is specified by the LineStyle attribute of the GraphData1 ... GraphData*n* style elements in the current style.

THICKNESS= *n* <*units*>

specifies the thickness of the line. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: For ungrouped data, the default line thickness is specified by the LineThickness attribute of the GraphFit style element in the current style.

For grouped data, the default line thickness is specified by the LineThickness attribute of the GraphData1 ... GraphData n style elements in the current style.

NOGROUP

specifies that the fit does not use the group variable from the scatter plot.

Restriction: To use this option, all of the X and Y variables must be numeric.

ROWS= n

specifies the number of rows in the graph.

Interaction: This option has no effect if you specify the COLUMNS= option.

SPACING = n

specifies the amount of spacing (in pixels) that is placed between the cells in the graph, if the PLOT statement creates multiple cells.

Default: 10

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the plot components. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default: 0.0

UNISCALE= X | Y | ALL

specifies that the X axis, Y axis, or both axes are scaled uniformly for all of the cells in the graph. By default, no scaling is performed. If this option is used, then all variables on the specified axis must be of the same type (all numeric or all character).

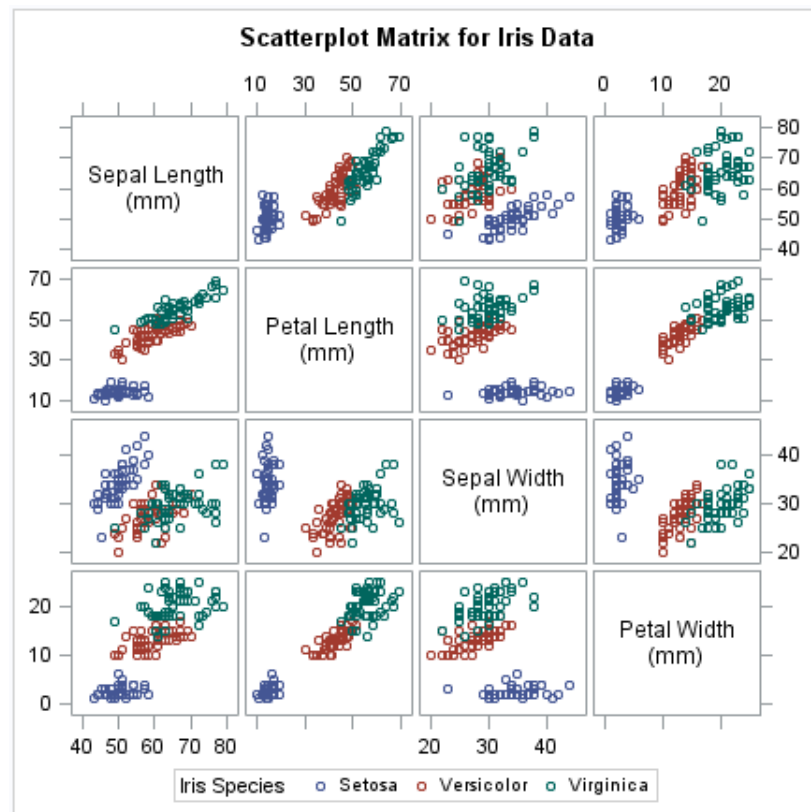
Examples: SGSCATTER Procedure

Example 1: Creating a Scatter Plot Matrix

Features: MATRIX statement
GROUP option

Sample library member: SGSCMAT

This example shows a scatter plot matrix with grouped data.

Output**Program**

```
proc sgscatter data=sashelp.iris;
  title "Scatterplot Matrix for Iris Data";
  matrix sepallength petallength sepalwidth petalwidth
    / group=species;
run;
title;
```

Program Description

Set the title and footnote and create the scatter plot matrix. In the MATRIX statement, the GROUP = option groups the data by the SPECIES variable.

```
proc sgscatter data=sashelp.iris;
  title "Scatterplot Matrix for Iris Data";
  matrix sepallength petallength sepalwidth petalwidth
    / group=species;
run;
title;
```

Example 2: Creating a Graph with Multiple Independent Scatter Plots and Spline Curves

Features: PLOT statement

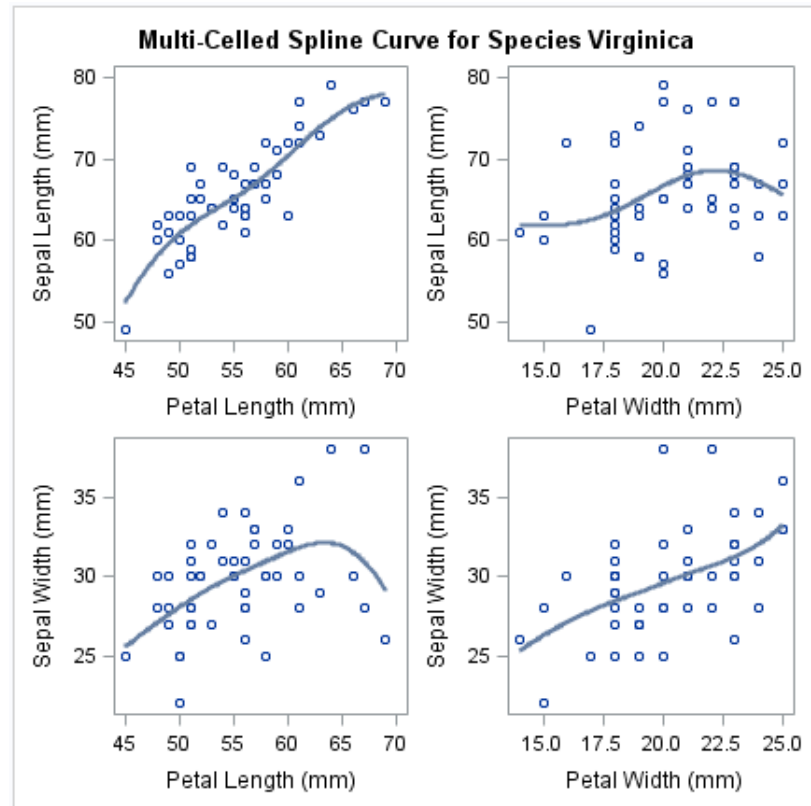
PBSPLINE option

Sample library
member:

SGSCPLT

This example shows a graph with multiple independent scatter plots with fitted splines.

Output



Program

```
proc sgscatter data=sashelp.iris(where=(species="Virginica"));
  title "Multi-Celled Spline Curve for Species Virginica";
  plot (sepallength sepalwidth)*(petallength petalwidth)
    / pbspline;
run;
title;
```

Program Description

Create the scatter plots. In the PLOT statement, the PBSPLINE option fits the spline curves to the scatter points.

```
proc sgscatter data=sashelp.iris(where=(species="Virginica"));
  title "Multi-Celled Spline Curve for Species Virginica";
  plot (sepallength sepalwidth)*(petallength petalwidth)
    / pbspline;
```

```
run;
title;
```

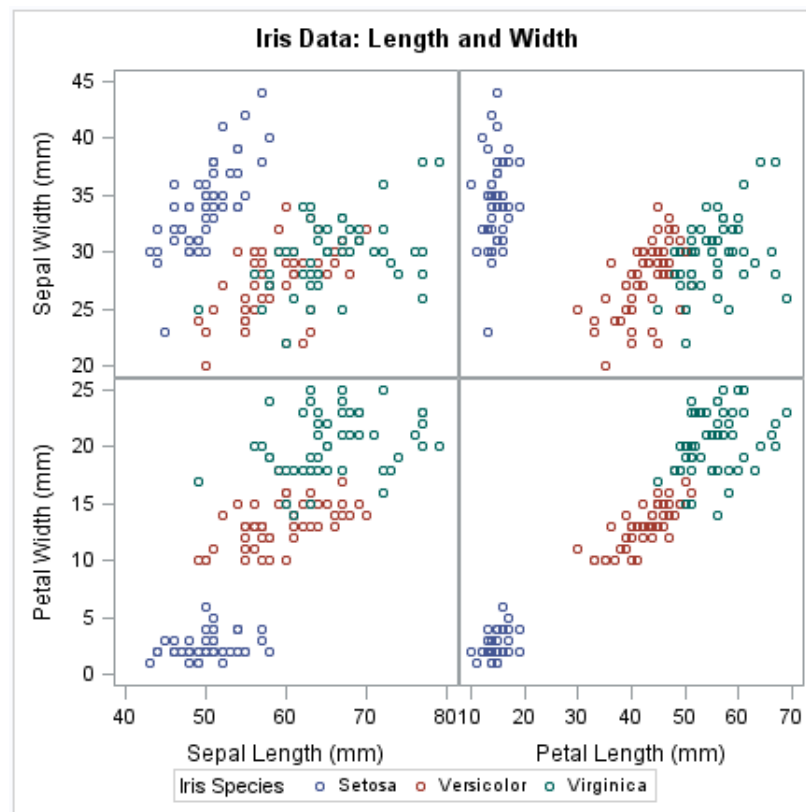
Example 3: Creating a Simple Comparative Panel

Features: COMPARE statement
GROUP option

Sample library member: SGSCCMP

This example shows a comparative scatter plot with grouped data.

Output



Program

```
proc sgscatter data=sashelp.iris;
  title "Iris Data: Length and Width";
  compare x=(sepallength petallength)
          y=(sepalwidth petalwidth)
          / group=species;
run;
title;
```

Program Description

Create the scatter plot. In the COMPARE statement, the GROUP= option groups the data by the SPECIES variable.

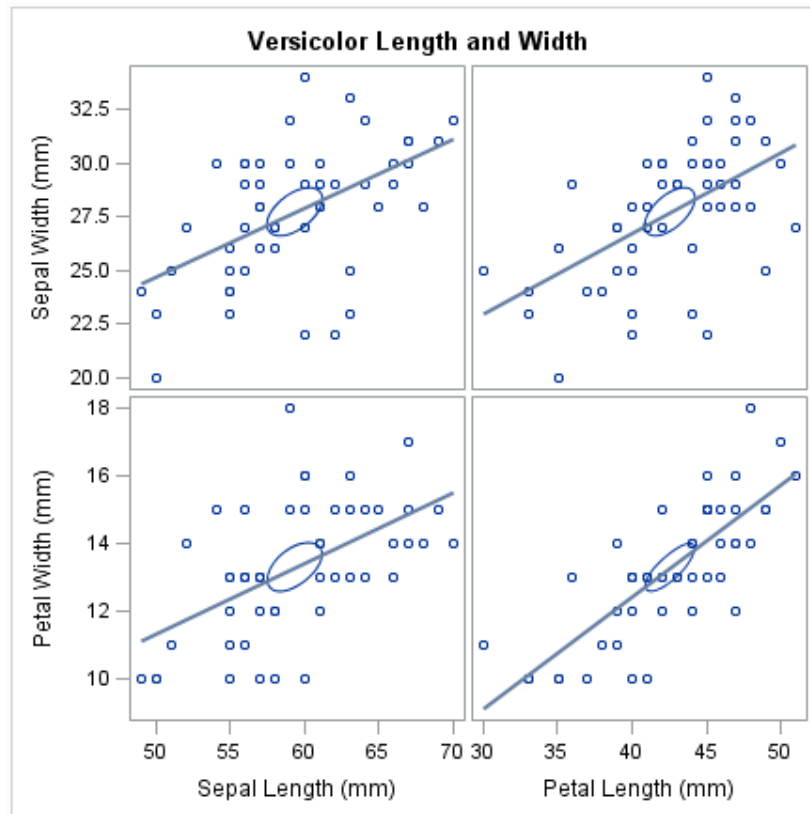
```
proc sgscatter data=sashelp.iris;
  title "Iris Data: Length and Width";
  compare x=(sepallength petallength)
          y=(sepalwidth petalwidth)
          / group=species;
run;
title;
```

Example 4: Creating a Comparative Panel with Regression Fits and Confidence Ellipses

Features: COMPARE statement
REG option
ELLIPSE option
SPACING option

Sample library member: SGSCCM1

This example shows a comparative panel of scatter plots with regression fits and confidence ellipses.

Output**Program**

```
proc sgscatter data=sashelp.iris(where=(species="Versicolor"));
  title "Versicolor Length and Width";
  compare y=(sepalwidth petalwidth)
          x=(sepalwidth petalwidth)
          / reg ellipse=(type=mean) spacing=4;
run;
title;
```

Program Description

Create the scatter plot. In the COMPARE statement, the REG option fits the regression lines and the ELLIPSE option creates the confidence ellipses and sets the ellipse type to MEAN. The SPACING= option adds spacing between plots.

```
proc sgscatter data=sashelp.iris(where=(species="Versicolor"));
  title "Versicolor Length and Width";
  compare y=(sepalwidth petalwidth)
          x=(sepalwidth petalwidth)
          / reg ellipse=(type=mean) spacing=4;
run;
title;
```


Part 3

SG Annotation

Chapter 10

Annotating ODS Graphics 557

Chapter 11

SG Annotation Function Dictionary 565

Chapter 10

Annotating ODS Graphics

Overview of SG Annotation	557
SG Annotation Data Sets	558
About the SG Annotation Data Set	558
SG Annotation Variables	559
SG Annotation Functions	560
Modifying an SG Procedure to Use the SG Annotation Data Set	561
Controlling the Drawing Space	561
About the Drawing Space	561
Drawing Space Variables	562
Drawing Space Values	562

Overview of SG Annotation

The ODS Statistical Graphics (SG) procedures provide a mechanism for adding shapes, images, and other annotations to graph output.

You can add the following annotation elements to a graph:

- text labels
- lines and arrows
- ovals (including circles)
- rectangles (including squares)
- polygons
- images

There are two main tasks required to add annotation elements to a graph:

1. Create an SG annotation data set, which contains the commands for creating the annotation elements. For more information, see [“About the SG Annotation Data Set” on page 558](#).
2. Modify the SG procedure to use the SG annotation data set. You can use annotation in the SGPLOT, SGPANEL, and SGSCATTER procedures. For more information, see [“Modifying an SG Procedure to Use the SG Annotation Data Set” on page 561](#).

SG Annotation Data Sets

About the SG Annotation Data Set

Once you have determined which annotation you want and how you want it to appear in the output, you can create the SG annotation data set. This data set contains the commands for creating one or more annotation elements. The annotations drawn by these commands can be added to SG procedure output.

Each observation represents a command to draw an annotation element or to continue an element. Reserved keywords are used for functions that specify the drawing operation.

You create the data set using the same methods that you use to create any SAS data set. The main distinctions are that the SG annotation data set uses reserved keywords for its variable names, and each observation represents a command to draw an annotation element.

Note: The most commonly used method for creating data sets is with a DATA step. For more information about the DATA step, see *SAS Language Reference: Concepts*.

Here is an example of an SG annotation data set named LINE. The observations in this data set contain the commands that create a blue text label and a blue line.

Display 10.1 LINE SG Annotation Data Set

Obs	function	x1	y1	label	x2	y2	textcolor	linecolor
1	text	20	70	Average Height 62 Inches	.	.	blue	
2	line	10	60		99	60		blue

A blank denotes a missing value for a character variable. A '.' denotes a missing value for a numeric variable.

Each observation in this data set contains complete instructions for drawing an annotation. The value of the FUNCTION variable determines what the observation does. Other variables control how the function is performed.

This list describes each observation in LINE and the task that it performs:

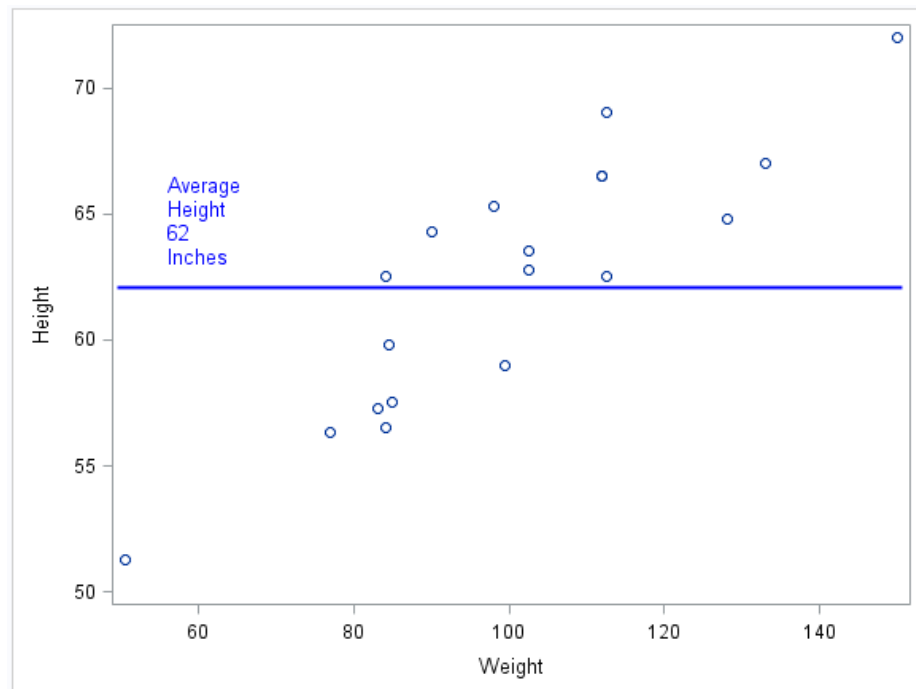
1. This instruction writes a blue text inset at position (20,70). The value of the FUNCTION variable (TEXT) tells the program what to do. The values of the coordinate variables X1 and Y1 tell where to do it. The value of the attribute variable TEXTCOLOR specifies the color of the text inset.

Note: The default drawing space for both observations is the graph area. For more information about drawing spaces, see [“Controlling the Drawing Space” on page 561](#).

2. This instruction draws a blue line that begins at position (10,60) and ends at (99,60). The value of the FUNCTION variable (LINE) tells the program what to do. The values of the coordinate variables X1, Y1, X2, and Y2 tell where to do it. The value of the attribute variable LINECOLOR specifies the color of the line.

The following figure shows the blue text and line annotations created by the LINE data set displayed with an SGPLOT procedure that specifies a scatter plot.

Display 10.2 LINE SG Annotation Output



Here is the SAS code step that creates the LINE data set and generates the graph.

```
data Line;
  infile datalines dlm='#';
  length label $ 27
         textcolor $ 9
         linecolor $ 9;

  input function $ x1 y1 label x2 y2 textcolor linecolor;
datalines;
text # 20 # 70 # Average Height 62 Inches # . # . # blue #
line # 10 # 60 #   # 99 # 60 #   # blue
;
run;

proc sgplot data=sashelp.class sganno=Line;
  scatter x=weight y=height;
run;
```

SG Annotation Variables

When an SG annotation data set is processed, the SG procedure looks at the values of specific variables in order to draw annotation elements. Variables in the SG annotation data set have predefined names. In each observation, the procedure looks only for variables with those names. Other variables can be present, but they are ignored.

The variables can be categorized into the following main types:

- A FUNCTION variable specifies which annotation element to draw. For more information, see [“SG Annotation Functions” on page 560](#).

- Position variables specify the point or points at which to draw the graphics element.
- Coordinate system variables specify the drawing space to use for the annotation. For more information, see [“Controlling the Drawing Space” on page 561](#).
- Attribute variables specify the characteristics of the graphics element (for example, color, size, line style, text font).

All variables are described in the context of the functions that support the variables. For more information about the functions, see [Chapter 11, “SG Annotation Function Dictionary,” on page 565](#).

SG Annotation Functions

In an SG annotation data set, the FUNCTION variable accepts a set of predefined values (functions) that insert annotation elements into the graph. The value of the FUNCTION variable specifies what drawing action the observation performs. These functions act in conjunction with other variables that determine where and how to perform the action.

The following table summarizes the functions:

Table 10.1 Summary of SG Annotation Functions

Function	Description
ARROW	Draws an arrow annotation. For more information, see “ARROW Function” on page 565 .
IMAGE	Specifies a graphic file to use for an image annotation. For more information, see “IMAGE Function” on page 570 .
LINE	Draws a line annotation. For more information, see “LINE Function” on page 573 .
OVAL	Draws an oval or circle annotation. For more information, see “OVAL Function” on page 577 .
POLYCONT	Continues drawing a polygon that was begun with the POLYGON function, or a line that was begun with the POLYLINE function. For more information, see “POLYCONT Function” on page 581 .
POLYGON	Specifies the beginning point of a polygon. For more information, see “POLYGON Function” on page 582 .
POLYLINE	Specifies the beginning point of a polyline, which is a connected series of line segments. For more information, see “POLYLINE Function” on page 585 .
RECTANGLE	Draws a rectangle or square annotation. For more information, see “RECTANGLE Function” on page 588 .
TEXT	Places text in the graph output. For more information, see “TEXT Function” on page 592 .

Function	Description
TEXTCONT	Continues a text string. For more information, see “TEXTCONT Function” on page 596 .

Modifying an SG Procedure to Use the SG Annotation Data Set

After you have created an SG annotation data set, you can modify the SG procedure to reference the annotation data. You can use annotation in the SGPLOT, SGPANEL, and SGSCATTER procedures.

To modify the procedure to use the SG annotation data set, add the following option to the procedure statement.

SGANNO= *annotation-data-set*

specifies the SG annotation data set that you want to use. For more information about SG annotation data sets, see [“SG Annotation Data Sets” on page 558](#).

You can also create margins around the graph by using the PAD= option.

PAD= *dimension<units>* | (*pad-options*)

specifies the amount of extra space that is reserved along the edges of the graph. This option creates margins around the graph for company logos, annotated notes, and so on. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see [“Units of Measurement” on page 659](#).

Use pad options to create non-uniform padding. Edges that are not assigned padding are padded with the default amount.

pad-options can be one or more of the following:

LEFT= *dimension<units>*

specifies the amount of extra space to add to the left edge.

RIGHT= *dimension<units>*

specifies the amount of extra space to add to the right edge.

TOP= *dimension<units>*

specifies the amount of extra space to add to the top edge.

BOTTOM= *dimension<units>*

specifies the amount of extra space to add to the bottom edge.

Controlling the Drawing Space

About the Drawing Space

You can control the position and scaling of your annotations by specifying the drawing space and units in the SG annotation data set. You have the option to specify the drawing space for the X or Y axes individually, or for both axes.

All annotations are rendered in one of four drawing areas:

graph area

the entire region of the graph image. The graph area contains the axes, titles, footnotes, legends, and one or more cells.

layout area

the area of the layout, not including any titles and footnotes.

wall area

the area within the axes, including offsets. (This area is not available when using the SG PANEL and SG SCATTER procedures.)

data area

the area within the axes, not including offsets. (This area is not available when using the SG PANEL and SG SCATTER procedures.)

Annotations can be rendered in the drawing areas using one of three units:

- percentage
- pixels
- data value (data area only)

Drawing Space Variables

You can specify the drawing space values for the following variables in the SG annotation data set:

Table 10.2 Data Set Variables That Are Used to Specify Drawing Space

Variable	Description	Supported Functions
DRAWSPACE	specifies the drawing space for the annotation	all except POLYCONT and TEXTCONT
X1SPACE	specifies the drawing space of the annotation's X coordinate.	all except TEXTCONT
X2SPACE	specifies the drawing space of the annotation's second X coordinate.	ARROW, LINE
Y1SPACE	specifies the drawing space of the annotation's Y coordinate.	all except TEXTCONT
Y2SPACE	specifies the drawing space of the annotation's second Y coordinate.	ARROW, LINE

Drawing Space Values

Here are the values that are used in the SG annotation data set to control the drawing space.

Note: For more information about the data, graph, layout, or wall areas, see [“About the Drawing Space” on page 561](#).

DATAPERCENT

positioned and scaled as a percentage with respect to the data area.

DATAPIXEL

positioned and scaled as pixels with respect to the data area.

DATAVALUE

positioned and scaled with respect to the data values.

GRAPHPERCENT

positioned and scaled as a percentage of the graph area.

Note: GRAPHPERCENT is the default value.

GRAPHPIXEL

positioned and scaled as pixels with respect to the graph area.

LAYOUTPERCENT

positioned and scaled as a percentage of the layout area.

LAYOUTPIXEL

positioned and scaled as pixels with respect to the layout area.

WALLPERCENT

positioned and scaled as a percentage of the wall area.

WALLPIXEL

positioned and scaled as pixels with respect to the wall area.

Note: For the SGPanel and SGScatter procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

Chapter 11

SG Annotation Function Dictionary

Dictionary	565
ARROW Function	565
IMAGE Function	570
LINE Function	573
OVAL Function	577
POLYCONT Function	581
POLYGON Function	582
POLYLINE Function	585
RECTANGLE Function	588
TEXT Function	592
TEXTCONT Function	596
Examples	597
Example 1: Create an Arrow in a Paneled Graph	597
Example 2: Create a Tabular Text Annotation	599

Dictionary

ARROW Function

Draws an arrow annotation from the specified (X1, Y1) coordinates to the specified (X2,Y2) coordinates.

Syntax

FUNCTION= "ARROW";

Required Variables

X1= *numeric-value*

specifies the first X coordinate of the annotation.

Default: none

X2= *numeric-value*

specifies the second X coordinate of the annotation.

Default: none

Y1= *numeric-value*

specifies the first Y coordinate of the annotation.

Default: none

Y2= *numeric-value*

specifies the second Y coordinate of the annotation.

Default: none

Optional Variables**DIRECTION= “BOTH” | “IN” | “OUT”**

specifies the direction for arrows.

“BOTH”

Places the arrowhead at both ends of the line.

“IN”

Places the arrowhead at the source (X1 or Y1 coordinate) of the line.

“OUT”

Places the arrowhead at the tail end (X2 or Y2 coordinate) of the line.

Default: OUT

DISCRETEOFFSET= *numeric-value*

specifies an amount to offset the annotation from a discrete value in data space. Specify a value from -0.5 (left offset) to +0.5 (right offset).

Default: 0.0 (no offset)

DRAWSPACE= “*draw-space-value*”

specifies the drawing space and units for the annotation. DRAWSPACE can be used rather than specify individual values for X1SPACE, X2SPACE, Y1SPACE, and Y2SPACE.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE
- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default: GRAPHPERCENT (unless overridden by a coordinate draw space, such as X1SPACE)

Restriction: For the SG PANEL and SG SCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

See: For more information about these values, see [“Drawing Space Values” on page 562](#).

LAYER= “BACK” | “FRONT”

specifies whether the annotation is drawn in front of (FRONT) or behind (BACK) the graph.

Default: FRONT

LINECOLOR= “color” | “style-attribute”

specifies a line color for the annotation. You can specify a color or a style element attribute. Here is an example of a style attribute:

GraphData2:Color

You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: COLOR attribute in the graph’s current style

LINEPATTERN= “line-pattern”

specifies a line pattern for the annotation. You can reference SAS patterns by number or by name. For a list of line patterns, see [“Line Patterns” on page 663](#).

Default: LineStyle attribute in the graph’s current style

LINETHICKNESS= *n*

specifies the thickness of the line. You cannot specify the unit of measure. The default unit of pixels is always used.

Default: LineThickness attribute in the graph’s current style

SCALE= *numeric-value*


specifies a scale factor for the arrowheads. Specify a positive number.

Default: determined by the procedure


SHAPE= BARBED | CLOSED | FILLED | OPEN

specifies the shape of the arrowheads. Specify one of the following:


BARBED

a solid triangle with an indent at the base. 


CLOSED

an outline of a triangle. 

FILLED

a solid triangle. 

OPEN

a triangle that resembles the letter "V". 

Default: OPEN

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the annotation. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default: 0.0

X1SPACE= “draw-space-value”

specifies the drawing space of the annotation’s first X coordinate.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL

- DATAVALUE
- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default: GRAPHPERCENT

Restriction: For the SG PANEL and SG SCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

Note: If you do not also specify the X2SPACE= value, then the second X coordinate uses the default drawing space.

See: For more information about these values, see [“Drawing Space Values” on page 562](#).

X2SPACE= “draw-space-value”

specifies the drawing space of the annotation’s second X coordinate.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE
- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default: GRAPHPERCENT

Restriction: For the SG PANEL and SG SCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

Note: If you do not also specify the X1SPACE= value, then the first X coordinate uses the default drawing space.

See: For more information about these values, see [“Drawing Space Values” on page 562](#).

XAXIS= “X” | “X2”

specifies which X axis to use for data space annotations.

Default: X

XC1= “text-string”

specifies the character X coordinate of an annotation.

Default: none

XC2= “text-string”

specifies the second character X coordinate of an annotation.

Default: none

Y1SPACE= “*draw-space-value*”

specifies the drawing space of the annotation’s second Y coordinate.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE
- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default: GRAPHPERCENT

Restriction: For the SGPanel and SGSCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

Note: If you do not also specify the Y2SPACE= value, then the second Y coordinate uses the default drawing space.

See: For more information about these values, see [“Drawing Space Values” on page 562](#).

Y2SPACE= “*draw-space-value*”

specifies the drawing space of the annotation’s second Y coordinate.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE
- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default: GRAPHPERCENT

Restriction: For the SGPanel and SGSCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

Note: If you do not also specify the Y1SPACE= value, then the first Y coordinate uses the default drawing space.

See: For more information about these values, see [“Drawing Space Values” on page 562](#).

YAXIS= “Y” | “Y2”

specifies which Y axis to use for data space annotations.

Default: Y**YC1= “text-string”**

specifies the character Y coordinate of an annotation.

Default: none**YC2= “text-string”**

specifies the second character Y coordinate of an annotation.

Default: none

IMAGE Function

Specifies a graphic file to use for an image annotation. If no (X1, Y1) coordinates are provided, the image appears in the center of the graph.

Syntax

FUNCTION= “IMAGE”;

Required Variable

IMAGE= “file-name”

specifies the image file to be displayed in the graphics output.

Include the complete path and filename. The syntax of external file specifications varies across operating environments.

If you do not specify the full path, then the procedure looks for the file in the SAS current working directory. The SAS current directory is the same directory in which you start your SAS session. If you are running SAS with the windowing environment in the Windows operating system, then the current directory is displayed in the status bar at the bottom of the main SAS window.

For a list of image formats that can be used, see “Supported File Types for Output Destinations” in Chapter 6 of *SAS Output Delivery System: User's Guide*.

Optional Variables

ANCHOR= “TOPLEFT” | “TOP” | “TOPRIGHT” | “RIGHT” |**“BOTTOMRIGHT” | “BOTTOM” | “BOTTOMLEFT” | “LEFT” | “CENTER”**

specifies the anchor position of the annotation. This point is placed on the specified X1 and Y1 positions.

Default: CENTER**BORDER= “TRUE” | “FALSE”**

turns the border on and off.

Default: FALSE**DISCRETEOFFSET= numeric-value**

specifies an amount to offset the annotation from a discrete value in data space. Specify a value from -0.5 (left offset) to +0.5 (right offset).

Default: 0.0 (no offset)

DRAWSPACE= “draw-space-value”

specifies the drawing space and units for the annotation. DRAWSPACE can be used rather than specify X1SPACE and Y1SPACE individually.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE
- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default: GRAPHPERCENT (unless overridden by a coordinate draw space, such as X1SPACE)

Restriction: For the SGPanel and SGScatter procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

See: For more information about these values, see [“Drawing Space Values” on page 562](#).

HEIGHT= *numeric-value*

specifies the height of the annotation. Specify a positive number greater than zero. You can use the HEIGHTUNIT variable to specify the unit of measurement.

Default: Determined by the system

HEIGHTUNIT= “DATA” | “PERCENT” | “PIXEL”

specifies the dimension unit to use for the HEIGHT value.

Default: PERCENT

IMAGESCALE= “FIT” | “FITHEIGHT” | “FITWIDTH” | “TILE”

specifies how the image is scaled within the width and height. You can specify the WIDTH and HEIGHT variables.

Default: FITWIDTH or FITHEIGHT if no size or if one size (either the width or the height) is specified (to preserve aspect)

FIT if width and height are both specified

LAYER= “BACK” | “FRONT”

specifies whether the annotation is drawn in front of (FRONT) or behind (BACK) the graph.

Default: FRONT

LINECOLOR= “color” | “style-attribute”

specifies a color of the border around the image, if displayed. You can specify a color or a style element attribute. Here is an example of a style attribute:

GraphData2:Color

You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: COLOR attribute in the graph’s current style

LINEPATTERN= “*line-pattern*”

specifies a line pattern of the border around the image, if displayed. You can reference SAS patterns by number or by name. For a list of line patterns, see [“Line Patterns” on page 663](#).

Default: LineStyle attribute in the graph’s current style

LINETHICKNESS= *n*

specifies the thickness of the border around the image, if displayed. You cannot specify the unit of measure. The default unit of pixels is always used.

Default: LineThickness attribute in the graph’s current style

ROTATE= *degrees*

rotates the annotation. Specify a number between 0 and 360.

Default: 0

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the annotation. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default: 0.0

WIDTH= *numeric-value*

specifies the width of the annotation. Specify a positive number greater than zero. You can use the WIDTHUNIT variable to specify the unit of measurement.

Default: Determined by the system

WIDTHUNIT= “DATA” | “PERCENT” | “PIXEL”

specifies the dimension unit to use for the WIDTH value.

Default: PERCENT

X1= *numeric-value*

specifies the X coordinate of the annotation.

Default: none

X1SPACE= “*draw-space-value*”

specifies the drawing space of the annotation’s X coordinate.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE
- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default: GRAPHPERCENT

Restriction: For the SG PANEL and SG SCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

See: For more information about these values, see [“Drawing Space Values” on page 562](#).

XAXIS= “X” | “X2”

specifies which X axis to use for data space annotations.

Default: X

XC1= “text-string”

specifies the character X coordinate of an annotation.

Default: none

Y1= *numeric-value*

specifies the Y coordinate of the annotation.

Default: none

Y1SPACE= “*draw-space-value*”

specifies the drawing space of the annotation’s Y coordinate.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE
- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default: GRAPHPERCENT

Restriction: For the SGPanel and SGSCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

See: For more information about these values, see [“Drawing Space Values” on page 562](#).

YAXIS= “Y” | “Y2”

specifies which Y axis to use for data space annotations.

Default: Y

YC1= “text-string”

specifies the character Y coordinate of an annotation.

Default: none

LINE Function

Draws a line annotation from the specified (X1, Y1) coordinates to the specified (X2,Y2) coordinates.

Syntax

FUNCTION= “LINE”;

Required Variable**X1= *numeric-value***

specifies the first X coordinate of the annotation.

Default: none**X2= *numeric-value***

specifies the second X coordinate of the annotation.

Default: none**Y1= *numeric-value***

specifies the first Y coordinate of the annotation.

Default: none**Y2= *numeric-value***

specifies the second Y coordinate of the annotation.

Default: none**Optional Variables****DISCRETEOFFSET= *numeric-value***

specifies an amount to offset the annotation from a discrete value in data space. Specify a value from -0.5 (left offset) to +0.5 (right offset).

Default: 0.0 (no offset)**DRAWSPACE= “*draw-space-value*”**

specifies the drawing space and units for the annotation. DRAWSPACE can be used rather than specify individual values for X1SPACE, X2SPACE, Y1SPACE, and Y2SPACE.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE
- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default: GRAPHPERCENT (unless overridden by a coordinate draw space, such as X1SPACE)**Restriction:** For the SG PANEL and SG SCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.**See:** For more information about these values, see [“Drawing Space Values” on page 562](#).**LAYER= “BACK” | “FRONT”**

specifies whether the annotation is drawn in front of (FRONT) or behind (BACK) the graph.

Default: FRONT

LINECOLOR= “color” | “style-attribute”

specifies a line color for the annotation. You can specify a color or a style element attribute. Here is an example of a style attribute:

GraphData2:Color

You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: COLOR attribute in the graph’s current style

LINEPATTERN= “line-pattern”

specifies a line pattern for the annotation. You can reference SAS patterns by number or by name. For a list of line patterns, see [“Line Patterns” on page 663](#).

Default: LineStyle attribute in the graph’s current style

LINETHICKNESS= *n*

specifies the thickness of the line. You cannot specify the unit of measure. The default unit of pixels is always used.

Default: LineThickness attribute in the graph’s current style

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the annotation. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default: 0.0

X1SPACE= “draw-space-value”

specifies the drawing space of the annotation’s first X coordinate.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE
- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default: GRAPHPERCENT

Restriction: For the SG PANEL and SG SCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

Note: If you do not also specify the X2SPACE= value, then the second X coordinate uses the default drawing space.

See: For more information about these values, see [“Drawing Space Values” on page 562](#).

X2SPACE= “draw-space-value”

specifies the drawing space of the annotation’s second X coordinate.

draw-space-value can be one of the following:

- DATAPERCENT

- DATAPIXEL
- DATAVALUE
- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default: GRAPHPERCENT

Restriction: For the SG PANEL and SG SCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

Note: If you do not also specify the X1SPACE= value, then the first X coordinate uses the default drawing space.

See: For more information about these values, see [“Drawing Space Values” on page 562](#).

XAXIS= “X” | “X2”

specifies which X axis to use for data space annotations.

Default: X

XC1= “text-string”

specifies the character X coordinate of an annotation.

Default: none

XC2= “text-string”

specifies the second character X coordinate of an annotation.

Default: none

Y1SPACE= “draw-space-value”

specifies the drawing space of the annotation’s second Y coordinate.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE
- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default: GRAPHPERCENT

Restriction: For the SG PANEL and SG SCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

Note: If you do not also specify the Y2SPACE= value, then the second Y coordinate uses the default drawing space.

See: For more information about these values, see [“Drawing Space Values” on page 562](#).

Y2SPACE= “draw-space-value”

specifies the drawing space of the annotation’s second Y coordinate.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE
- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default: GRAPHPERCENT

Restriction: For the SGPanel and SGScatter procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

Note: If you do not also specify the Y1SPACE= value, then the first Y coordinate uses the default drawing space.

See: For more information about these values, see [“Drawing Space Values” on page 562](#).

YAXIS= “Y” | “Y2”

specifies which Y axis to use for data space annotations.

Default: Y

YC1= “text-string”

specifies the character Y coordinate of an annotation.

Default: none

YC2= “text-string”

specifies the second character Y coordinate of an annotation.

Default: none

OVAL Function

Draws an oval annotation around the specified (X1, Y1) coordinates using the specified height and width.

Tip: An easy way to draw a circle is to specify PIXEL for the HEIGHTUNIT and WIDTHUNIT variables and then provide the same value for the HEIGHT and WIDTH.

Syntax

FUNCTION= “OVAL”;

Required Variables

HEIGHT= *numeric-value*

specifies the height of the annotation. Specify a positive number greater than zero. You can use the HEIGHTUNIT variable to specify the unit of measurement.

Default: none

WIDTH= *numeric-value*

specifies the width of the annotation. Specify a positive number greater than zero. You can use the WIDTHUNIT variable to specify the unit of measurement.

Default: none

X1= *numeric-value*

specifies the X coordinate of the annotation.

Default: none

Y1= *numeric-value*

specifies the Y coordinate of the annotation.

Default: none

Optional Variables

DISCRETEOFFSET= *numeric-value*

specifies an amount to offset the annotation from a discrete value in data space. Specify a value from -0.5 (left offset) to +0.5 (right offset).

Default: 0.0 (no offset)

DISPLAY= “ALL” | “FILLED” | “OUTLINE”

specifies the fill and outline properties. You can specify that the annotation be filled (FILLED), have an outline (OUTLINE), or both (ALL).

Default: OUTLINE

DRAWSPACE= “*draw-space-value*”

specifies the drawing space and units for the annotation. DRAWSPACE can be used rather than specify X1SPACE and Y1SPACE individually.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE
- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default: GRAPHPERCENT (unless overridden by a coordinate draw space, such as X1SPACE)

Restriction: For the SGPanel and SGScatter procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

See: For more information about these values, see [“Drawing Space Values” on page 562](#).

FILLCOLOR= “color” | “style-attribute”

specifies a fill color for the annotation. You can specify a color or a style element attribute. Here is an example of a style attribute: GraphData2:Color

You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: COLOR attribute in the graph’s current style

Interaction: This option takes effect only if the DISPLAY option is specified as FILLED or ALL.

FILLTRANSPARENCY= *numeric-value*

specifies the transparency for the fill portion of an annotation. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default: 0.0

HEIGHTUNIT= “DATA” | “PERCENT” | “PIXEL”

specifies the dimension unit to use for the HEIGHT value.

Default: PERCENT

LAYER= “BACK” | “FRONT”

specifies whether the annotation is drawn in front of (FRONT) or behind (BACK) the graph.

Default: FRONT

LINECOLOR= “color” | “style-attribute”

specifies a line color for the annotation. You can specify a color or a style element attribute. Here is an example of a style attribute:

GraphData2:Color

You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: COLOR attribute in the graph’s current style

LINEPATTERN= “*line-pattern*”

specifies a line pattern for the annotation. You can reference SAS patterns by number or by name. For a list of line patterns, see [“Line Patterns” on page 663](#).

Default: LineStyle attribute in the graph’s current style

LINETHICKNESS= *n*

specifies the thickness of the line. You cannot specify the unit of measure. The default unit of pixels is always used.

Default: LineThickness attribute in the graph’s current style

ROTATE= *degrees*

rotates the annotation. Specify a number between 0 and 360.

Default: 0

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the annotation. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default: 0.0

WIDTHUNIT= “DATA” | “PERCENT” | “PIXEL”

specifies the dimension unit to use for the WIDTH value.

Default: PERCENT

X1SPACE= “*draw-space-value*”

specifies the drawing space of the annotation’s X coordinate.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE
- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default: GRAPHPERCENT

Restriction: For the SGPanel and SGSCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

See: For more information about these values, see “Drawing Space Values” on page 562.

XAXIS= “X” | “X2”

specifies which X axis to use for data space annotations.

Default: X

XC1= “text-string”

specifies the character X coordinate of an annotation.

Default: none

Y1SPACE= “*draw-space-value*”

specifies the drawing space of the annotation’s Y coordinate.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE
- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default: GRAPHPERCENT

Restriction: For the SGPanel and SGScatter procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

See: For more information about these values, see [“Drawing Space Values”](#) on page 562.

YAXIS= “Y” | “Y2”

specifies which Y axis to use for data space annotations.

Default: Y

YC1= “text-string”

specifies the character Y coordinate of an annotation.

Default: none

POLYCONT Function

Continues drawing a polygon that was begun with the POLYGON function, or a line that was begun with the POLYLINE function. POLYCONT specifies each successive point in the polygon or polyline.

Syntax

FUNCTION= “POLYCONT”;

Required Variables

X1= *numeric-value*

specifies the X coordinate of the annotation.

Default: none

Y1= *numeric-value*

specifies the Y coordinate of the annotation.

Default: none

Optional Variables

X1SPACE= “*draw-space-value*”

specifies the drawing space of the annotation’s X coordinate.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE
- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default: GRAPHPERCENT

Restriction: For the SG PANEL and SG SCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

See: For more information about these values, see [“Drawing Space Values” on page 562](#).

XC1= “text-string”

specifies the character X coordinate of an annotation.

Default: none

Y1SPACE= “draw-space-value”

specifies the drawing space of the annotation’s Y coordinate.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE
- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default: GRAPHPERCENT

Restriction: For the SG PANEL and SG SCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

See: For more information about these values, see [“Drawing Space Values” on page 562](#).

YC1= “text-string”

specifies the character Y coordinate of an annotation.

Default: none

POLYGON Function

Specifies the beginning point of a polygon, which is a connected series of lines that form a closed shape. This function is used in conjunction with two or more POLYCONT functions to define vertex points for the polygon. Associated variables can define the fill pattern and color, as well as the line type that outlines the polygon.

Requirement: The SG annotation data set must also specify at least two instances of the POLYCONT function immediately after the POLYGON function.

Syntax

FUNCTION= “POLYGON”;

Required Variable**X1= *numeric-value***

specifies the X coordinate of the annotation.

Default: none**Y1= *numeric-value***

specifies the Y coordinate of the annotation.

Default: none**Optional Variables****DISCRETEOFFSET= *numeric-value***

specifies an amount to offset the annotation from a discrete value in data space.

Specify a value from -0.5 (left offset) to +0.5 (right offset).

Default: 0.0 (no offset)**DISPLAY= “ALL” | “FILLED” | “OUTLINE”**

specifies the fill and outline properties. You can specify that the annotation be filled (FILLED), have an outline (OUTLINE), or both (ALL).

Default: OUTLINE**DRAWSPACE= “*draw-space-value*”**

specifies the drawing space and units for the annotation. DRAWSPACE can be used rather than specify X1SPACE and Y1SPACE individually.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE
- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default: GRAPHPERCENT (unless overridden by a coordinate draw space, such as X1SPACE)**Restriction:** For the SGPanel and SGScatter procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.**See:** For more information about these values, see [“Drawing Space Values” on page 562](#).**FILLCOLOR= “color” | “*style-attribute*”**

specifies a fill color for the annotation. You can specify a color or a style element attribute. Here is an example of a style attribute: GraphData2:Color

You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.**Default:** COLOR attribute in the graph’s current style

Interaction: This option takes effect only if the DISPLAY option is specified as FILLED or ALL.

FILLTRANSPARENCY= *numeric-value*

specifies the transparency for the fill portion of an annotation. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default: 0.0

LAYER= “BACK” | “FRONT”

specifies whether the annotation is drawn in front of (FRONT) or behind (BACK) the graph.

Default: FRONT

LINECOLOR= “color” | “style-attribute”

specifies a line color for the annotation. You can specify a color or a style element attribute. Here is an example of a style attribute:

GraphData2:Color

You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: COLOR attribute in the graph’s current style

LINEPATTERN= “line-pattern”

specifies a line pattern for the annotation. You can reference SAS patterns by number or by name. For a list of line patterns, see [“Line Patterns” on page 663](#).

Default: LineStyle attribute in the graph’s current style

LINETHICKNESS= *n*

specifies the thickness of the line. You cannot specify the unit of measure. The default unit of pixels is always used.

Default: LineThickness attribute in the graph’s current style

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the annotation. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default: 0.0

X1SPACE= “draw-space-value”

specifies the drawing space of the annotation’s X coordinate.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE
- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default: GRAPHPERCENT

Restriction: For the SGPanel and SGScatter procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

See: For more information about these values, see [“Drawing Space Values” on page 562](#).

XAXIS= “X” | “X2”

specifies which X axis to use for data space annotations.

Default: X

XC1= “text-string”

specifies the character X coordinate of an annotation.

Default: none

Y1SPACE= “draw-space-value”

specifies the drawing space of the annotation’s Y coordinate.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE
- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default: GRAPHPERCENT

Restriction: For the SGPanel and SGScatter procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

See: For more information about these values, see [“Drawing Space Values” on page 562](#).

YAXIS= “Y” | “Y2”

specifies which Y axis to use for data space annotations.

Default: Y

YC1= “text-string”

specifies the character Y coordinate of an annotation.

Default: none

POLYLINE Function

Specifies the beginning point of a polyline, which is a connected series of line segments. This function is used in conjunction with one or more POLYCONT functions to draw a series of connected straight lines.

Requirement: The SG annotation data set must also specify at least one instance of the POLYCONT function immediately after the POLYLINE function.

Note: The POLYLINE object is similar to a POLYGON, except that the POLYLINE object does not need to be a closed shape.

Syntax

FUNCTION= "POLYLINE";

Required Variables

X1= *numeric-value*

specifies the X coordinate of the annotation.

Default: none

Y1= *numeric-value*

specifies the Y coordinate of the annotation.

Default: none

Optional Variables

DISCRETEOFFSET= *numeric-value*

specifies an amount to offset the annotation from a discrete value in data space.

Specify a value from -0.5 (left offset) to +0.5 (right offset).

Default: 0.0 (no offset)

DRAWSPACE= "*draw-space-value*"

specifies the drawing space and units for the annotation. DRAWSPACE can be used rather than specify X1SPACE and Y1SPACE individually.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE
- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default: GRAPHPERCENT (unless overridden by a coordinate draw space, such as X1SPACE)

Restriction: For the SG PANEL and SG SCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

See: For more information about these values, see ["Drawing Space Values" on page 562](#).

LAYER= "BACK" | "FRONT"

specifies whether the annotation is drawn in front of (FRONT) or behind (BACK) the graph.

Default: FRONT

LINECOLOR= “color” | “style-attribute”

specifies a line color for the annotation. You can specify a color or a style element attribute. Here is an example of a style attribute:

GraphData2:Color

You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: COLOR attribute in the graph’s current style

LINEPATTERN= “line-pattern”

specifies a line pattern for the annotation. You can reference SAS patterns by number or by name. For a list of line patterns, see “Line Patterns” on page 663.

Default: LineStyle attribute in the graph’s current style

LINETHICKNESS= *n*

specifies the thickness of the line. You cannot specify the unit of measure. The default unit of pixels is always used.

Default: LineThickness attribute in the graph’s current style

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the annotation. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default: 0.0

X1SPACE= “draw-space-value”

specifies the drawing space of the annotation’s X coordinate.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE
- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default: GRAPHPERCENT

Restriction: For the SG PANEL and SG SCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

See: For more information about these values, see “Drawing Space Values” on page 562.

XAXIS= “X” | “X2”

specifies which X axis to use for data space annotations.

Default: X

XC1= “text-string”

specifies the character X coordinate of an annotation.

Default: none

Y1SPACE= “draw-space-value”

specifies the drawing space of the annotation’s Y coordinate.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE
- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default: GRAPHPERCENT

Restriction: For the SG PANEL and SG SCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

See: For more information about these values, see [“Drawing Space Values” on page 562](#).

YAXIS= “Y” | “Y2”

specifies which Y axis to use for data space annotations.

Default: Y

YC1= “text-string”

specifies the character Y coordinate of an annotation.

Default: none

RECTANGLE Function

Draws a rectangle annotation around the specified (X1, Y1) coordinates using the specified height and width. By default, the (X1, Y1) coordinates constitute the center of the rectangle, though you can change this behavior using the ANCHOR option.

Tip: An easy way to draw a square is to specify PIXEL for the HEIGHTUNIT and WIDTHUNIT variables and then provide the same value for the HEIGHT and WIDTH.

Syntax

FUNCTION= “RECTANGLE”;

Required Variables**HEIGHT= numeric-value**

specifies the height of the annotation. Specify a positive number greater than zero. You can use the HEIGHTUNIT variable to specify the unit of measurement.

Default: none

WIDTH= *numeric-value*

specifies the width of the annotation. Specify a positive number greater than zero. You can use the WIDTHUNIT variable to specify the unit of measurement.

Default: none

X1= *numeric-value*

specifies the X coordinate of the annotation.

Default: none

Y1= *numeric-value*

specifies the Y coordinate of the annotation.

Default: none

Optional Variables**ANCHOR= “TOPLEFT” | “TOP” | “TOPRIGHT” | “RIGHT” | “BOTTOMRIGHT” | “BOTTOM” | “BOTTOMLEFT” | “LEFT” | “CENTER”**

specifies the anchor position of the annotation. This point is placed on the specified X1 and Y1 positions.

Default: CENTER

CORNERRADIUS= *numeric-value*

specifies the roundness of the corners of rectangles. Specify a value from 0.0 (completely rectangular) to 1.0 (oval).

Default: 0.0

DISCRETEOFFSET= *numeric-value*

specifies an amount to offset the annotation from a discrete value in data space. Specify a value from -0.5 (left offset) to +0.5 (right offset).

Default: 0.0 (no offset)

DISPLAY= “ALL” | “FILLED” | “OUTLINE”

specifies the fill and outline properties. You can specify that the annotation be filled (FILLED), have an outline (OUTLINE), or both (ALL).

Default: OUTLINE

DRAWSPACE= “*draw-space-value*”

specifies the drawing space and units for the annotation. DRAWSPACE can be used rather than specify X1SPACE and Y1SPACE individually.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE
- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default: GRAPHPERCENT (unless overridden by a coordinate draw space, such as X1SPACE)

Restriction: For the SG PANEL and SG SCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

See: For more information about these values, see [“Drawing Space Values” on page 562](#).

FILLCOLOR= “color” | “style-attribute”

specifies a fill color for the annotation. You can specify a color or a style element attribute. Here is an example of a style attribute: GraphData2:Color

You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: COLOR attribute in the graph’s current style

Interaction: This option takes effect only if the DISPLAY option is specified as FILLED or ALL.

FILLTRANSPARENCY= numeric-value

specifies the transparency for the fill portion of an annotation. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default: 0.0

HEIGHTUNIT= “DATA” | “PERCENT” | “PIXEL”

specifies the dimension unit to use for the HEIGHT value.

Default: PERCENT

LAYER= “BACK” | “FRONT”

specifies whether the annotation is drawn in front of (FRONT) or behind (BACK) the graph.

Default: FRONT

LINECOLOR= “color” | “style-attribute”

specifies a line color for the annotation. You can specify a color or a style element attribute. Here is an example of a style attribute:

GraphData2:Color

You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: COLOR attribute in the graph’s current style

LINEPATTERN= “line-pattern”

specifies a line pattern for the annotation. You can reference SAS patterns by number or by name. For a list of line patterns, see [“Line Patterns” on page 663](#).

Default: LineStyle attribute in the graph’s current style

LINETHICKNESS= n

specifies the thickness of the line. You cannot specify the unit of measure. The default unit of pixels is always used.

Default: LineThickness attribute in the graph’s current style

ROTATE= degrees

rotates the annotation. Specify a number between 0 and 360.

Default: 0

TRANSPARENCY= numeric-value

specifies the degree of transparency for the annotation. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default: 0.0

WIDTHUNIT= “DATA” | “PERCENT” | “PIXEL”

specifies the dimension unit to use for the WIDTH value.

Default: PERCENT

X1SPACE= “*draw-space-value*”

specifies the drawing space of the annotation’s X coordinate.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE
- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default: GRAPHPERCENT

Restriction: For the SGPanel and SGSCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

See: For more information about these values, see “Drawing Space Values” on page 562.

XAXIS= “X” | “X2”

specifies which X axis to use for data space annotations.

Default: X

XC1= “text-string”

specifies the character X coordinate of an annotation.

Default: none

Y1SPACE= “*draw-space-value*”

specifies the drawing space of the annotation’s Y coordinate.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE
- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default: GRAPHPERCENT

Restriction: For the SG PANEL and SG SCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

See: For more information about these values, see [“Drawing Space Values”](#) on page 562.

YAXIS= “Y” | “Y2”

specifies which Y axis to use for data space annotations.

Default: Y

YC1= “text-string”

specifies the character Y coordinate of an annotation.

Default: none

TEXT Function

Places text in the graph output. Associated variables can control the color, size, font, base angle, and rotation of the characters displayed. If you do not supply the (X1, Y1) coordinates, the text is placed in the center of the graph.

Syntax

FUNCTION= “TEXT”;

Required Variable

LABEL= “text-string”

specifies the text label. The *text-string* can contain superscripts, subscripts, and Unicode characters by using ODS escapement notation. Here is a simple example:

```
data anno;
  input function $ label $7-40;
  datalines;
  text This label is rich(*ESC*){sup '2'}
  ;
run;
proc sgplot data=sashelp.class sganno=anno;
  scatter x=weight y=height;
run;
```

Default: none

Optional Variables

ANCHOR= “TOPLEFT” | “TOP” | “TOPRIGHT” | “RIGHT” |

“BOTTOMRIGHT” | “BOTTOM” | “BOTTOMLEFT” | “LEFT” | “CENTER”

specifies the anchor position of the annotation. This point is placed on the specified X1 and Y1 positions.

Default: CENTER

BORDER= “TRUE” | “FALSE”

turns the border on and off.

Default: FALSE

DISCRETEOFFSET= *numeric-value*

specifies an amount to offset the annotation from a discrete value in data space. Specify a value from -0.5 (left offset) to +0.5 (right offset).

Default: 0.0 (no offset)

DRAWSPACE= “*draw-space-value*”

specifies the drawing space and units for the annotation. DRAWSPACE can be used rather than specify X1SPACE and Y1SPACE individually.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE
- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default: GRAPHPERCENT (unless overridden by a coordinate draw space, such as X1SPACE)

Restriction: For the SGPanel and SGScatter procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

See: For more information about these values, see [“Drawing Space Values” on page 562](#).

FILLCOLOR= “color” | “*style-attribute*”

specifies the background color for the text annotation. You can specify a color or a style element attribute. Here is an example of a style attribute: GraphData2:Color

You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: COLOR attribute in the graph’s current style

FILLTRANSPARENCY= *numeric-value*

specifies the transparency for the fill portion of an annotation. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default: 0.0

JUSTIFY= “CENTER” | “LEFT” | “RIGHT”

specifies the text justification.

Default: LEFT

LAYER= “BACK” | “FRONT”

specifies whether the annotation is drawn in front of (FRONT) or behind (BACK) the graph.

Default: FRONT

LINECOLOR= “color” | “*style-attribute*”

specifies a line color for the border of the text annotation. You can specify a color or a style element attribute. Here is an example of a style attribute: GraphData2:Color

You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: COLOR attribute in the graph’s current style

Interaction: This option takes effect only if BORDER= “TRUE”.

LINEPATTERN= “*line-pattern*”

specifies a line pattern for the border of the text annotation. You can reference SAS patterns by number or by name. For a list of line patterns, see “Line Patterns” on page 663.

Default: LineStyle attribute in the graph’s current style

Interaction: This option takes effect only if BORDER= “TRUE”.

LINETHICKNESS= *n*

specifies the line thickness of the border of the text annotation. You cannot specify the unit of measure. The default unit of pixels is always used.

Default: LineThickness attribute in the graph’s current style

Interaction: This option takes effect only if BORDER= “TRUE”.

ROTATE= *degrees*

rotates the annotation. Specify a number between 0 and 360.

Default: 0

TEXTCOLOR= “*color*” | “*style-attribute*”

specifies the text color for the annotation. You can specify a color or a style element attribute. Here is an example of a style attribute: GraphData2:Color

You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: COLOR attribute of the GraphAnnoText style element in the current style

TEXTFONT= “*font-family*”

specifies the font family for the annotation. The SAS ODS styles use TrueType system fonts. For more information, see “TrueType Fonts Supplied by SAS” in Chapter 15 of *SAS Language Reference: Concepts*.

Default: FontFamily attribute of the GraphAnnoText style element in the current style.

TEXTSIZE= *n* <*units*>

specifies the font size of the annotation. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: FontSize attribute of the GraphAnnoText style element in the current style.

TEXTSTYLE= “*ITALIC*” | “*NORMAL*”

specifies whether the annotation characters are italic (ITALIC) or normal (NORMAL).

Default: FontStyle attribute of the GraphAnnoText style element in the current style.

TEXTWEIGHT= *BOLD* | *NORMAL*

specifies whether the annotation characters are bold (BOLD) or normal (NORMAL).

Default: FontWeight attribute of the GraphAnnoText style element in the current style.

TRANSPARENCY= *numeric-value*

specifies the degree of transparency for the annotation. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default: 0.0

WIDTH= *numeric-value*

specifies the width of the annotation. Specify a positive number greater than zero. You can use the WIDTHUNIT variable to specify the unit of measurement.

Default: Determined by the system

WIDTHUNIT= “DATA” | “PERCENT” | “PIXEL”

specifies the dimension unit to use for the WIDTH value.

Default: PERCENT

X1= *numeric-value*

specifies the X coordinate of the annotation.

Default: none

X1SPACE= “*draw-space-value*”

specifies the drawing space of the annotation’s X coordinate.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE
- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default: GRAPHPERCENT

Restriction: For the SGPanel and SGScatter procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

See: For more information about these values, see [“Drawing Space Values” on page 562](#).

XAXIS= “X” | “X2”

specifies which X axis to use for data space annotations.

Default: X

XC1= “*text-string*”

specifies the character X coordinate of an annotation.

Default: none

Y1= *numeric-value*

specifies the Y coordinate of the annotation.

Default: none

Y1SPACE= “*draw-space-value*”

specifies the drawing space of the annotation’s Y coordinate.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE
- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default: GRAPHPERCENT

Restriction: For the SG PANEL and SG SCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

See: For more information about these values, see [“Drawing Space Values” on page 562](#).

YAXIS= “Y” | “Y2”

specifies which Y axis to use for data space annotations.

Default: Y

YC1= “text-string”

specifies the character Y coordinate of an annotation.

Default: none

TEXTCONT Function

Continues a text string. You can set the continued text apart from the main text by specifying different text attributes.

Syntax

FUNCTION= “TEXTCONT”;

Required Variable

LABEL= “text-string”

specifies the text label. The *text-string* can contain superscripts, subscripts, and Unicode characters by using ODS escapement notation. Here is a simple example:

```
data anno;
  infile datalines dlm='#';
  length function $8 label $35;
  input function $ label $ textcolor $;
  datalines;
  text # This label is rich(*ESC*){sup '2'} # black
  textcont # and this text is green # green
  ;
run;
```

```
proc sgplot data=sashelp.class sganno=anno;
  scatter x=weight y=height;
run;
```

Default: none

Optional Variables

TEXTCOLOR= “color” | “style-attribute”

specifies the text color for the annotation. You can specify a color or a style element attribute. Here is an example of a style attribute: GraphData2:Color

You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

Default: COLOR attribute of the GraphAnnoText style element in the current style

TEXTFONT= “font-family”

specifies the font family for the annotation. The SAS ODS styles use TrueType system fonts. For more information, see “TrueType Fonts Supplied by SAS” in Chapter 15 of *SAS Language Reference: Concepts*.

Default: FontFamily attribute of the GraphAnnoText style element in the current style.

TEXTSIZE= *n* <units>

specifies the font size of the annotation. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 659.

Default: FontSize attribute of the GraphAnnoText style element in the current style.

TEXTSTYLE= “ITALIC” | “NORMAL”

specifies whether the annotation characters are italic (ITALIC) or normal (NORMAL).

Default: FontStyle attribute of the GraphAnnoText style element in the current style.

TEXTWEIGHT= BOLD | NORMAL

specifies whether the annotation characters are bold (BOLD) or normal (NORMAL).

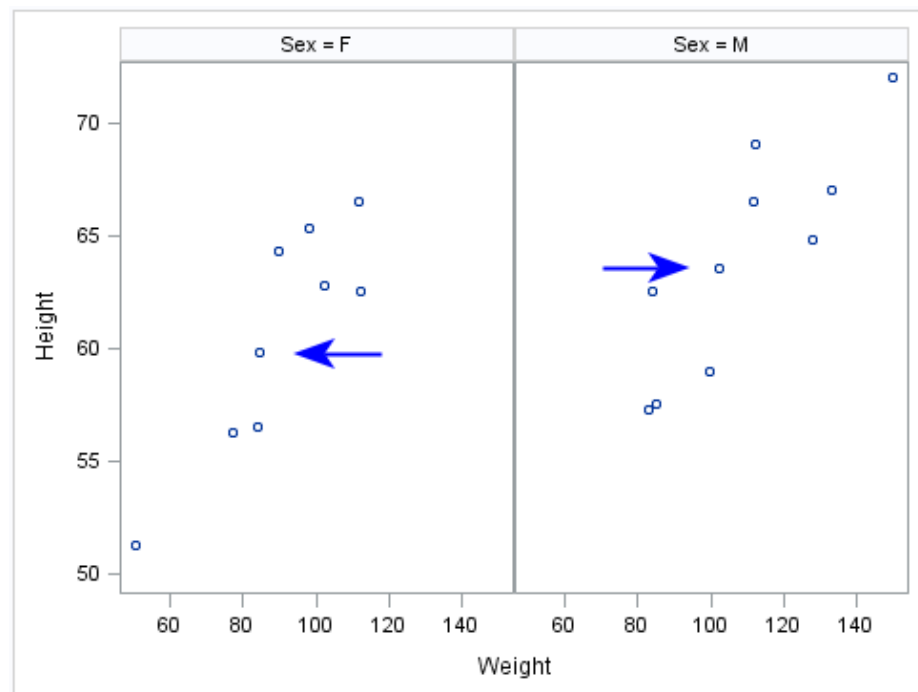
Default: FontWeight attribute of the GraphAnnoText style element in the current style.

Examples

Example 1: Create an Arrow in a Paneled Graph

Features: ARROW

This example uses the SGPANEL procedure to create an arrow near the middle data point for the height variable in each panel.

Output**Program**

```
data annoArrow;
retain drawspace "layoutpercent" linecolor "blue";
input function $ x1 y1 x2 y2 shape $ direction $;
datalines;
arrow 30 50 40 50 barbed in
arrow 65 63 75 63 barbed out
;
run;

proc sgpanel data=sashelp.class sganno=annoArrow;
panelby sex;
scatter x=weight y=height;
run;
```

Program Description**Create an SG annotation data set.**

```
data annoArrow;
retain drawspace "layoutpercent" linecolor "blue";
input function $ x1 y1 x2 y2 shape $ direction $;
datalines;
arrow 30 50 40 50 barbed in
arrow 65 63 75 63 barbed out
;
run;
```

Create the panel with the annotation.

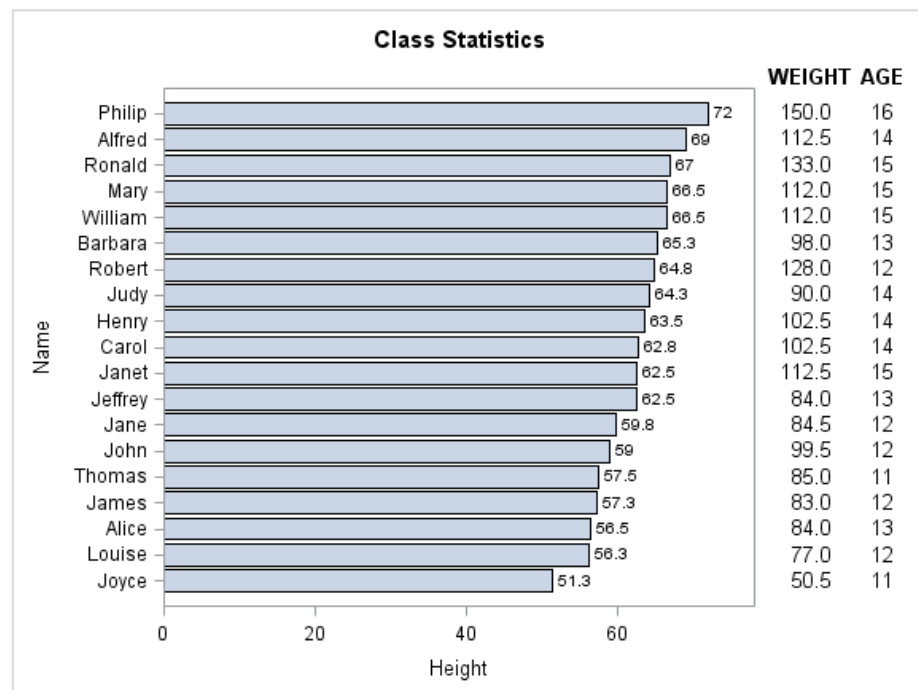
```
proc sgpanel data=sashelp.class sganno=annoArrow;
panelby sex;
scatter x=weight y=height;
run;
```

Example 2: Create a Tabular Text Annotation

Features: TEXT annotation
PAD option

This example uses the SGPLOT procedure to create an axis-aligned table to the right of a horizontal bar chart.

Output



Program

```
data anno;
length ylspace $ 12 label $ 6;
set sashelp.class (keep=name weight);
rename name=yc1;
retain ylspace "datavalue" xlspace "graphpercent" x1 90 function "text"
       textcolor "black" textweight "normal" y1 . anchor "right";
label=put(weight, F5.1);
run;

data temp;
length ylspace $ 12 label $ 6;
set sashelp.class (keep=name age);
rename name=yc1;
retain ylspace "datavalue" xlspace "graphpercent" x1 95 function "text"
       textcolor "black" textweight "normal";
```

```

label=put(age, F2.0);
run;

data headers;
length textweight $ 6;
retain ylspace "graphpercent" xlspace "graphpercent" function "text" y1 90
      textcolor "black" textweight "bold";
x1=87;
label="WEIGHT";
output;
x1=95;
label="AGE";
output;
run;

proc append base=anno (drop=weight) data=temp (drop=age) force nowarn;
run;

proc append base=anno (drop=weight) data=headers force nowarn;
run;

title1 "Class Statistics";
title2 " ";
proc sgplot data=sashelp.class sganno=anno pad=(right=25%);
hbar name / response=height datalabel nostatlabel
      categoryorder=respdesc;
run;
title1;

```

Program Description

Create a data set with the Weight column. Use annotation variables to define attributes for the weight text.

```

data anno;
length ylspace $ 12 label $ 6;
set sashelp.class (keep=name weight);
rename name=yc1;
retain ylspace "datavalue" xlspace "graphpercent" x1 90 function "text"
      textcolor "black" textweight "normal" y1 . anchor "right";
label=put(weight, F5.1);
run;

```

Create a data set with the Age column. Use annotation variables to define attributes for the age text.

```

data temp;
length ylspace $ 12 label $ 6;
set sashelp.class (keep=name age);
rename name=yc1;
retain ylspace "datavalue" xlspace "graphpercent" x1 95 function "text"
      textcolor "black" textweight "normal";
label=put(age, F2.0);
run;

```

Create the positions for the two columns of the table in the graph area.


```

data headers;
length textweight $ 6;
retain y1space "graphpercent" x1space "graphpercent" function "text" y1 90
      textcolor "black" textweight "bold";
x1=87;
label="WEIGHT";
output;
x1=95;
label="AGE";
output;
run;

```

Append the observations of TEMP to ANNO, and then append the observations of HEADERS.

```

proc append base=anno (drop=weight) data=temp (drop=age) force nowarn;
run;

proc append base=anno (drop=weight) data=headers force nowarn;
run;

```

Create the plot with the annotation.

```

title1 "Class Statistics";
title2 " ";
proc sgplot data=sashelp.class sganno=anno pad=(right=25%);
hbar name / response=height datalabel nostatlabel
      categoryorder=respdesc;
run;
title1;

```


Part 4

SG Attribute Maps

Chapter 12

***Using SG Attribute Maps to Control Visual Attributes* 605**

Chapter 12

Using SG Attribute Maps to Control Visual Attributes

Overview of SG Attribute Maps	605
SG Attribute Map Data Sets	606
About the SG Attribute Map Data Set	606
Data Sets That Contain Multiple SG Attribute Maps	608
Reserved SG Attribute Map Variables	608
Modify the Procedure to Use the SG Attribute Map Data Set	610
Example: Create a Plot That Uses a Single SG Attribute Map	611
Example: Combine Multiple SG Attribute Maps in a Graph	612
Example: Create a Panel That Uses an Attribute Map	614

Overview of SG Attribute Maps

The SG attribute map feature provides a mechanism for controlling the visual attributes that are applied to specific data values in your graphs.

For example, if you create a graph that plots items sold in different countries, you can specify the display attributes for the sales data of each country by name. Attribute maps apply only to group data. Attribute maps enable you to ensure that particular visual attributes are applied based on the value of the data instead of the position of the data in the data set.

Two main tasks are required for attribute mapping:

1. Create an SG attribute map data set, which associates data values with particular visual attributes. Each observation defines the attributes for a group value. For more information, see [“SG Attribute Map Data Sets” on page 606](#).
2. Modify the SG procedure and its plot statements to use the data in the SG attribute map. You can use attribute maps in the SGPLOT, SGPANEL, and SGSCATTER procedures (not all plot statements support attribute maps). For more information, see [“Modify the Procedure to Use the SG Attribute Map Data Set” on page 610](#).

SG Attribute Map Data Sets

About the SG Attribute Map Data Set

Attribute map data sets are used in the SG procedures to associate data values with visual attributes.

When you have determined which attributes you want to apply to the group data in a graph, you can create an SG attribute map data set. Each observation defines the attributes for a particular data group. An observation uses reserved variable names for the attribute map identifier (ID), the group value (VALUE), and the attributes.

You create the data set using the same methods that you use to create any SAS data set. The main distinctions are that the SG attribute map data set uses reserved keywords for its variable names, and each observation represents the attributes for a particular data group. The most commonly used method for creating data sets is with a DATA step. For more information about the DATA step, see *SAS Language Reference: Concepts*.

Note: Incorrect data in the attribute map data set can cause the graph to fail. For example, truncated variable values caused by the incorrect variable length being specified results in incorrect data.

Here is an example of an SG attribute map data set called MYATTRMAP. The observations in this data set contain the attribute map identifier (ID), the group value (VALUE), and the attributes (LINECOLOR, FILLCOLOR).

Display 12.1 Listing of the SG Attribute Map Data Set MYATTRMAP

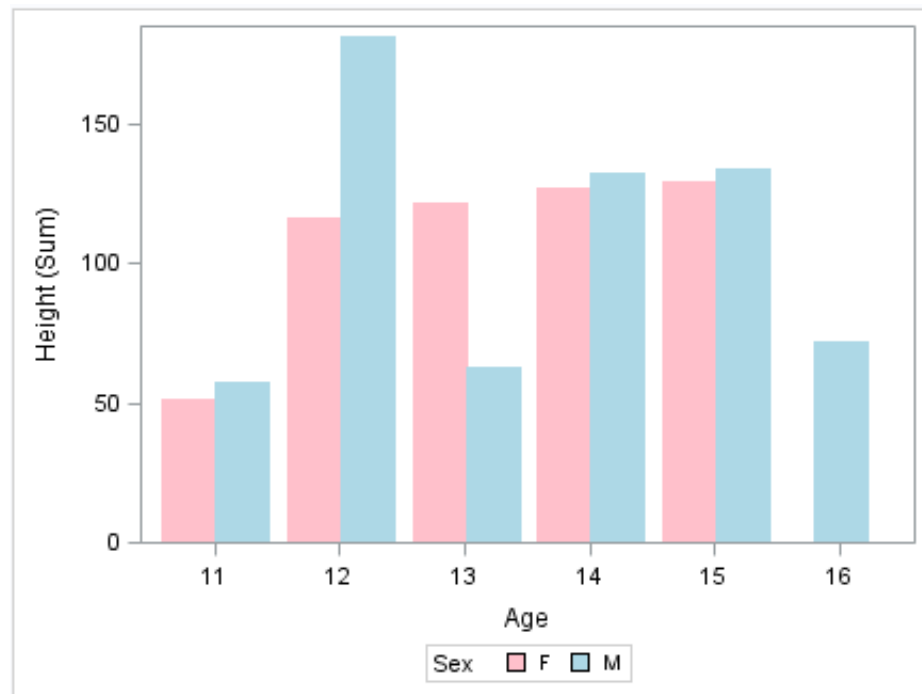
Obs	linecolor	fillcolor	ID	value
1	pink	pink	myid	F
2	lightblue	lightblue	myid	M

The data set has these characteristics:

- All variables listed are CHAR variables.
- The value of the ID variable, MYID in this case, is referenced in one or more plot statements within the procedure. You can use attribute maps in the SGPLOT, SGPANEL, and SGSCATTER procedures.
- The values of the VALUE variable are valid data group values. These values are case sensitive. The data group is assigned in the plot statement with the GROUP= option.
- The values for LINECOLOR= and FILLCOLOR= are valid SAS colors. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

The following output shows a bar chart that uses the MYATTRMAP data set for its pink and blue bar colors. The FILLCOLOR= values determine the color of the bars, and the LINECOLOR= values determine the color of the border edges around the bars.

The chart was produced with the SGPLOT procedure. The name of the data set is referenced in the procedure statement.

Display 12.2 Example Output That Uses an SG Attribute Map

This DATA step creates the SG attribute map data set MYATTRMAP. The ID values for the attribute map are MYID.

```
data myattrmap;
  length linecolor $ 9 fillcolor $ 9;
  input ID $ value $ linecolor $ fillcolor $;
  datalines;
  myid F pink pink
  myid M lightblue lightblue
  ;
run;
```

This procedure generates the graph. The SGPLOT statement references the SASHELP.CLASS data set and the ATTRMAP attribute map data set. The plot statement references the attribute map ID (MYID). The options that reference the attribute map (data set and ID) are highlighted.

```
proc sgplot data=sashelp.class dattrmap=myattrmap;
  vbar age / response=height group=sex groupdisplay=cluster attrid=myid;
run;
```

For more information about the highlighted options that are used in the procedure, see [“Modify the Procedure to Use the SG Attribute Map Data Set” on page 610](#).

You use a standard syntax to specify colors, line thickness, line patterns, and marker symbols. For more information about this syntax, see the following topics:

- “Specifying Colors in SAS/GRAPH Programs” in Chapter 14 of *SAS/GRAPH: Reference*
- [“Units of Measurement” on page 659](#)
- [“Line Patterns” on page 663](#)
- [“Marker Symbols” on page 661](#)

Data Sets That Contain Multiple SG Attribute Maps

An SG attribute map data set can contain more than one attribute map. This capability enables you to apply different attribute maps to different group variables in a graph.

When an SG attribute data set contains multiple attribute maps:

- The ID variable has more than one value.
- The VALUE variable has different values that correspond to different data groups.

In the procedure that generates the graph, the plot statements can specify different attribute map ID values.

Note: Plots that specify different ID values must also specify different groups. A group variable can be associated with only one attribute map ID. If a group is associated with more than one attribute map ID value, the graph produces incorrect attribute mapping and a warning is written to the SAS log.

For an example, see [“Example: Combine Multiple SG Attribute Maps in a Graph” on page 612](#).

Reserved SG Attribute Map Variables

About the Reserved SG Attribute Map Variables

When an SG attribute map data set is processed, the SG procedure looks at the values of specific variables in the SG attribute map data set. The procedure uses these values to associate visual attributes with group data. Variables in the SG attribute map data set have predefined names. In each observation, the procedure looks only for variables with those names. Other variables can be present, but they are ignored.

Variables That Have Style Values

For some variables, you can specify a style element or a style attribute as the value.

Here are examples:

- GraphData2 is an example of a style element.
- GraphData2:Color is an example of a style attribute.

For more information about style elements, see [“Style Elements for Use with ODS Graphics” on page 640](#).

For a table of the style elements and attributes that you can use with ODS Statistical Graphics, see Appendix 5, “Style Elements Affecting Template-Based Graphics,” in *SAS Output Delivery System: User's Guide*. This table contains each style element, the portion of the graph that it affects, and the default attribute values.

Required Reserved Variables

The ID and VALUE variables are required for every SG attribute map data set. If they are not found, a warning is written to the SAS log, and the data set is ignored. The remaining attribute variables are used as applicable to the plot type.

Descriptions of the Reserved Variables

The following list describes each reserved variable.

Note: All variables are of type CHAR, including variables that might contain numeric values.

FILLCOLOR= “color” | “style-attribute”

specifies the fill color. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

FILLSTYLE= “style-element”

specifies the style element for fill attributes. If you specify the style element, you can also specify the FILLCOLOR variable to override the color.

ID= “text-string”

(required) specifies the ID of the attribute map. This value is referenced from the ATTRID option on one or more plot statements.

Though every observation must have an ID value, different observations can have different values. You might use different values in the data set if you want plots to have different attributes.

LINECOLOR= “color” | “style-attribute”

specifies the color of the line. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

LINEPATTERN= “line-pattern” | “style-attribute”

specifies a line pattern for lines and outlines. You can reference SAS patterns by number or by name. For a list of line patterns, see “Line Patterns” on page 663.

LINESTYLE= “style-element”

specifies the style element for line attributes. If you specify the style element, you can also specify the LINEPATTERN and LINECOLOR variables to override specific appearance attributes.

MARKERCOLOR= “color” | “style-attribute”

specifies the color for the markers. You can specify colors using the same color schemes that are supported by SAS/GRAPH software. For more information, see “Color-Naming Schemes” in Chapter 14 of *SAS/GRAPH: Reference*.

MARKERSTYLE= “style-element”

specifies the style element for marker attributes. If you specify the style element, you can also specify the MARKERSYMBOL and MARKERCOLOR variables to override specific appearance attributes.

MARKERSYMBOL= “symbol-name” | “style-attribute”

specifies the symbol for the markers. For a list of valid marker symbols, see “Marker Symbols” on page 661.

VALUE= “text-string”

(required) specifies the group data value that is assigned to the attributes. The value must be valid for the data group that is assigned in the plot statement with the GROUP= option. You can provide different values in the data set if your plots specify different group variables.

Note: The value is case sensitive. Each instance of the value must have the exact same capitalization.

Modify the Procedure to Use the SG Attribute Map Data Set

After you have created an SG attribute map data set, you can modify the SG procedure and its plot statements to reference the SG attribute map data. You can use attribute maps in the SGPLOT, SGPANEL, and SGSCATTER procedures.

To reference attribute map data in a procedure:

1. Add the DATTRMAP= option to the procedure statement and specify the name of the SG attribute map data set.
2. For each plot statement that maps attributes, perform these steps:
 - Specify the GROUP= option. The group's values must correspond to the values in the VALUE variable in the attribute map data set.
 - Specify the ATTRID= option. This option specifies the value of the ID variable in the SG attribute map data set.

If the SG attribute map data set contains more than one ID value, then the plot statements can specify different ATTRID values. However, plots that specify different ATTRID values must also specify different groups. A group variable can be associated with only one attribute map ID. For more information, see [“Data Sets That Contain Multiple SG Attribute Maps” on page 608](#).

If the ATTRID value that you specify does not match a value for the ID variable in the attribute map data set, a warning is written to the SAS log and the ATTRID is ignored.

Attribute mapping is supported in the PLOT statement of the SGSCATTER procedure.

In the SGPLOT and SGPANEL procedures, any plot statement that supports GROUP variables supports attribute mapping. In these procedures, all plot statements support attribute mapping except the following:

Table 12.1 SGPANEL and SGPLOT Statements That Do Not Support Attribute Mapping

SGPANEL	SGPLOT
DENSITY	DENSITY
HISTOGRAM	ELLIPSE
KEYLEGEND	HISTOGRAM
REFLINE	INSET
	KEYLEGEND
	REFLINE

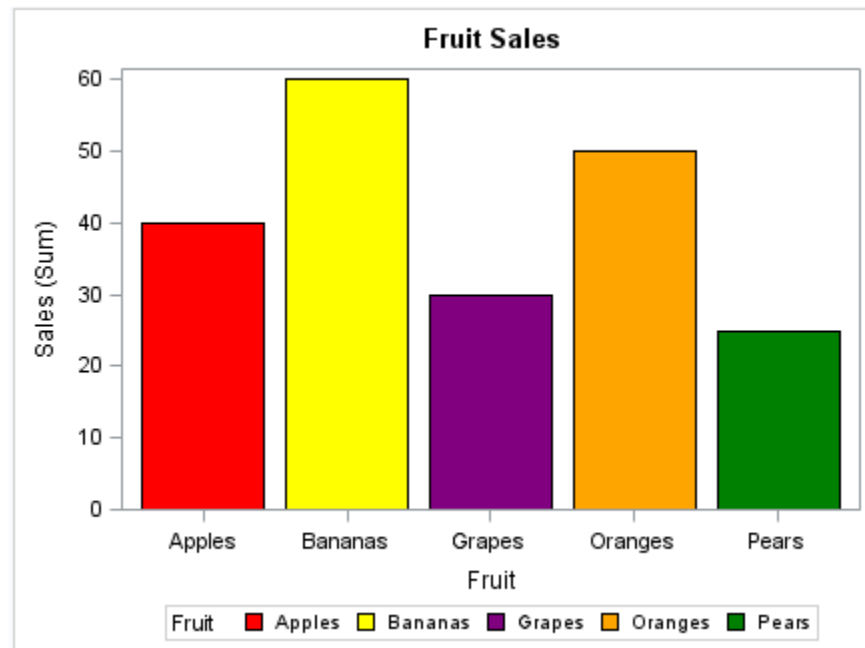
Here is an example of an SGPLOT procedure that references the SG attribute map data set shown in the example [Display 12.1 on page 606](#). The options that reference the attribute map (data set and ID) are highlighted.

```
proc sgplot data=sashelp.class dattrmap=myattrmap;
vbar age / response=height group=sex groupdisplay=cluster attrid=myid;
run;
```

Example: Create a Plot That Uses a Single SG Attribute Map

This example creates a simple bar chart that uses an SG attribute map.

Display 12.3 Bar Chart That Uses an SG Attribute Map



Here is the SAS code that creates the input data set, the attribute map data set, and the SGPLOT procedure output.

This DATA step creates the data that is used for the graph. The FRUIT variable is used in the procedure to group the data.

```
data fruit_sales;
input Fruit $ Sales;
datalines;
Apples 40
Pears 25
Oranges 50
Grapes 30
Bananas 60
run;
```

This DATA step creates the SG attribute map data set. The data set contains one attribute map, which is specified by the ID variable. All ID values for the attribute map are FRUIT.

```
data attrmapfruit;
retain linecolor "black";
input id $ value $ fillcolor $;
datalines;
fruit Apples red
fruit Pears green
fruit Oranges orange
fruit Grapes purple
fruit Bananas yellow
;
run;
```

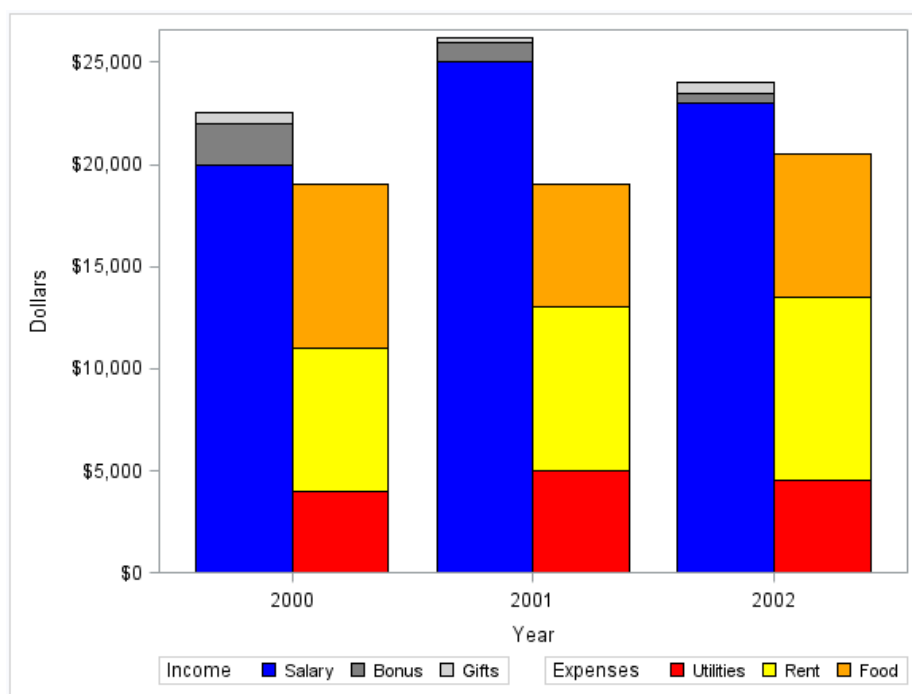
This PROC step generates the graph. The SGPLOT statement references the FRUIT_SALES data set and the ATTRMAPFRUIT attribute map data set. The plot statement references the attribute map ID (FRUIT).

```
title "Fruit Sales";
proc sgplot data=fruit_sales dattrmap=attrmapfruit;
vbar Fruit / response=Sales group=Fruit attrid=fruit;
run;
title;
```

Example: Combine Multiple SG Attribute Maps in a Graph

This example uses an SG attribute map data set that contains two attribute maps. Here are the noteworthy characteristics of the attribute map data set and the procedure:

- The attribute map data set has these features:
 - The ID variable has two values.
 - The VALUE variable has different values that correspond to two different data groups.
- In the procedure, the plot statements specify different ATTRID= values in order to use the different attribute maps. The plot statements also specify different GROUP= values. (A group variable can be associated with only one attribute map ID. For more information, see [“Data Sets That Contain Multiple SG Attribute Maps” on page 608.](#))

Display 12.4 Output That Uses Two SG Attribute Maps

Here is the SAS code that creates the attribute map data set and uses the SGPLOT procedure to generate the graph.

This DATA step creates the data that is used for the graph. The STATE and the TYPE variables are used in the procedure to group the data.

```
data finances;
format income dollar8. expense dollar8.;
length expensetype $ 9;
input Year incometype $ income expensetype $ expense;
datalines;
2000 Salary 20000 Utilities 4000
2000 Bonus 2000 Rent 7000
2000 Gifts 500 Food 8000
2001 Salary 25000 Utilities 5000
2001 Bonus 1000 Rent 8000
2001 Gifts 200 Food 6000
2002 Salary 23000 Utilities 4500
2002 Bonus 500 Rent 9000
2002 Gifts 500 Food 7000
;
run;
```

This DATA step creates the SG attribute map data set. The data set contains two attribute maps. Each attribute map is specified by an ID value: INCOME or EXPENSE.

```
data attrmap;
length value $ 9 fillcolor $ 9;
retain linecolor "black";
input id $ value $ fillcolor $;
datalines;
income Salary blue
income Bonus gray
```

```

income  Gifts      lightgray
expense Utilities red
expense Rent       yellow
expense Food       orange
;
run;

```

This PROC step generates the graph. The SGPLOT statement references the FINANCES data set and the ATTRMAP attribute map data set. Each plot statement references an attribute map ID. The plot statements also specify different GROUP= values.

```

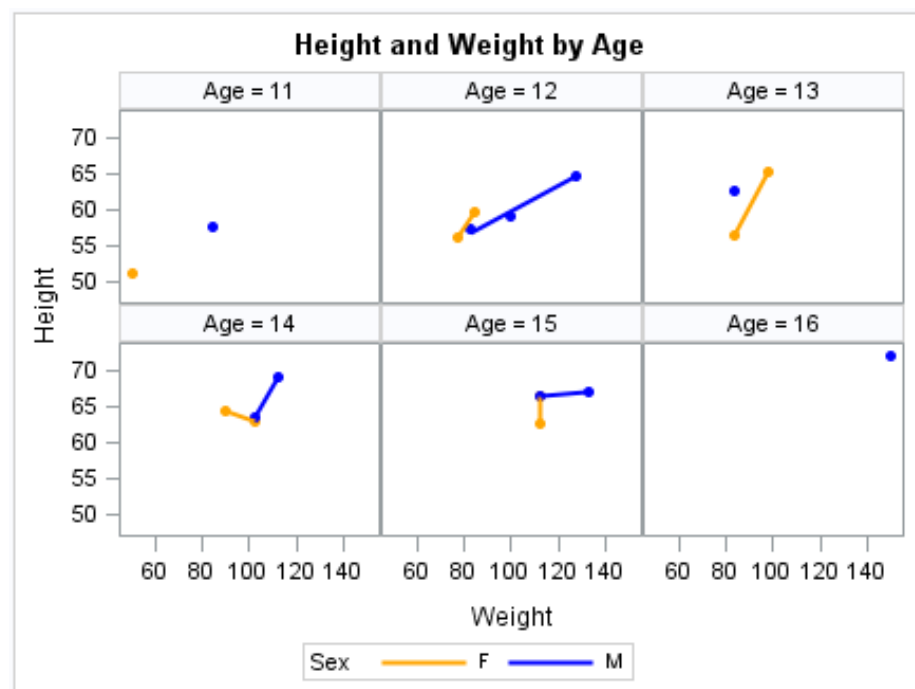
proc sgplot data=finances dattrmap=attrmap;
  yaxis label="Dollars";
  vbarparm category=year response=income / group=incometype attrid=income
    barwidth=0.4 groupdisplay=stack
    discreteoffset=-0.2 name="income";
  vbarparm category=year response=expense / group=expensetype attrid=expense
    barwidth=0.4 groupdisplay=stack
    discreteoffset=0.2 name="expense";
  keylegend "income" / position=bottomleft title="Income";
  keylegend "expense" / position=bottomright title="Expenses";
run;

```

Example: Create a Panel That Uses an Attribute Map

This example uses the SGPANEL procedure to generate SG attribute map output.

Display 12.5 Panel That Uses an SG Attribute Map



Here is the SAS code that creates the attribute map data set and runs the SG PANEL procedure.

This DATA step creates the SG attribute map data set. The data set contains one attribute map, which is specified by the ID variable. All ID values for the attribute map are MYID.

```
data attrmap;
  retain ID "myid";
  retain markersymbol "circlefilled";
  input value $ markercolor $ linecolor $ linepattern $;
  datalines;
F  orange  orange  solid
M  blue    blue    solid
;
run;
```

This PROC step generates the graph. The SG PANEL statement references the SASHELP.CLASS data set and the ATTRMAP attribute map data set. The plot statement references the attribute map ID (MYID).

```
title "Height and Weight by Age";
proc sgpanel data=sashelp.class dattrmap=attrmap;
  panelby age;
  reg x=weight y=height / group=sex attrid=myid;
run;
title;
```


Part 5

Customizing ODS Graphics

Chapter 13

Controlling the Appearance of Your Graphs 619

Chapter 14

Managing Your Graphics with ODS 649

Chapter 13

Controlling the Appearance of Your Graphs

Overview	619
Understanding Styles	620
About Styles and Style Elements	620
About the Default Styles	621
Recommended Styles	621
Viewing a Style Template	622
Examples of Styles	623
Specifying Styles	624
Specify a Style for an ODS Destination	624
Change the Default Style for the HTML Destination	625
Changing a Default Style in the SAS Registry	625
Using Procedure Options to Control Graph Appearance	625
Overview of the Appearance Options	625
Specifying a Style Element	626
Specifying Hardcoded Values	627
Specifying a Style Element with Hardcoded Values	629
Output for Grouped versus Non-Grouped Data	630
Non-Grouped Data	630
Grouped Data	633
Modifying Style Templates	635
Using Fill Patterns to Distinguish Grouped Bar Charts	636
Style Elements for Use with ODS Graphics	640

Overview

Along with table and page attributes, ODS styles contain a collection of graphical attributes such as color, marker shape, line pattern, fonts, and so on. Many carefully designed styles are shipped with SAS that enhance the visual impact of the graphics.

In addition to creating visually appealing graphics, many key elements of effective graphics are built into the various elements of the styles, such as the following:

- provide maximum contrast between backgrounds and data
- provide for ease of discrimination among different groups
- provide equal emphasis for data with equal importance

The ODS styles are used to produce professional looking graphics without the need for further modification. However, you have several options for modifying the appearance of your graphs if you choose to do so.

You can customize your graphical output at three levels.

The following table shows the three levels of customization.

Table 13.1 Levels of Customization

Method	Description	Level of Complexity
Change the style. See “Specifying Styles” on page 624 .	Specify a style with the STYLE= option on the ODS destination statement to change the appearance of the entire graph. Requires no further modification. The active style affects all SAS output and remains in effect until the ODS style or destination is changed.	low
Use Appearance options. See “Using Procedure Options to Control Graph Appearance” on page 625 .	Specify an appearance option in the procedure to change various aspects of your graph. Appearance options override the corresponding defaults from the current style. <i>Note:</i> Not all appearance attributes can be specified in this way. Appearance options affect only the procedure in which they are specified.	low
Modify individual style elements. See “Modifying Style Templates” on page 635 .	Specify or change style attributes in order to modify a style element. This requires the use of PROC TEMPLATE style statements. The active style affects all SAS output and remains in effect until the ODS style or destination is changed.	high

Understanding Styles

About Styles and Style Elements

ODS styles are produced from compiled STYLE templates written in PROC TEMPLATE style syntax. An ODS style template is a collection of style elements that provides specific visual attributes for your SAS output.

The style elements of a style are designed to ensure the goals of effective graphics. Each style element is a named collection of style attributes such as color, marker symbol, line style, font face, as well as many others. Each graphical element of a plot, such as a marker, a bar, a line or a title, derives its visual attributes from a specific style element from the active style.

Changing the style for an ODS destination is the easiest way to change a graph's appearance. Changing the current style requires only the use of the `STYLE=` option on an ODS destination statement. For more information, see [“Specify a Style for an ODS Destination” on page 624](#).

Note: The style that a destination uses is applied to tabular output as well as graphical output.

About the Default Styles

Every ODS output destination has a default style associated with it. These default styles are different for each destination. Therefore, your output might look different depending on which destination you use. For example, the default style for the PRINTER destination is “Printer” while the default style for the RTF destination is “RTF”.

Note: Starting with SAS 9.3, the default style for the HTML destination is HTMLBlue.

For a table that lists the default styles for ODS destinations, see “Working with Styles ” in Chapter 13 of *SAS Output Delivery System: User's Guide*.

You can display a list of the available styles by submitting the following PROC TEMPLATE statements:

```
proc template;  
  list styles;  
run;
```

You can change the default style for a destination by modifying the SAS Preferences or the SAS Registry.

See Also

- [“Change the Default Style for the HTML Destination” on page 625](#)
- [“Changing a Default Style in the SAS Registry” on page 625](#)

Recommended Styles

SAS ships a set of styles that have been designed by GUI experts to address the needs of different situations, while ensuring the principles of effective graphics.

The following is a subset of the styles shipped with SAS that are particularly suited for statistical graphics:

Table 13.2 Recommended Styles

Desired Output	Recommended Styles	Comments
Full color	HTMLBLUE*	white background, optimized for HTML output
	DEFAULT	gray background, optimized for HTML output
	ANALYSIS	yellow background
	STATISTICAL	white background, colored fills
	LISTING	white background, optimized for color format on white paper
	PRINTER	optimized for PS and PDF output
Gray scale	JOURNAL	interior filled areas are gray scale
	JOURNAL2**	interior filled areas are gray scale
Black and white	JOURNAL2**	interior filled areas have no color

* HTMLBlue is the default style for the ODS HTML destination.

** Journal2 and Journal3 by default render grouped bars with fill patterns. For more information, see [“Using Fill Patterns to Distinguish Grouped Bar Charts”](#) on page 636.

Viewing a Style Template

You can view the styles that SAS provides by using the command line or the SAS Windowing Environment.

To use the SAS Windowing Environment, follow these steps:

1. In the Results window, select the **Results** folder. Right-click and select **Templates** to open the Templates window.
2. Double-click **Sashelp.Tmplmst** to view the contents of that directory.
3. Double-click **styles** to view the contents of that directory.
4. Double-click the style definition that you want to view. For example, the HTMLBlue style definition is the template store for HTML output. Similarly, the RTF style definition is the template store for RTF output.

To use the command line, follow these steps:

1. To view the Templates window, submit this command in the command line:
ods templates

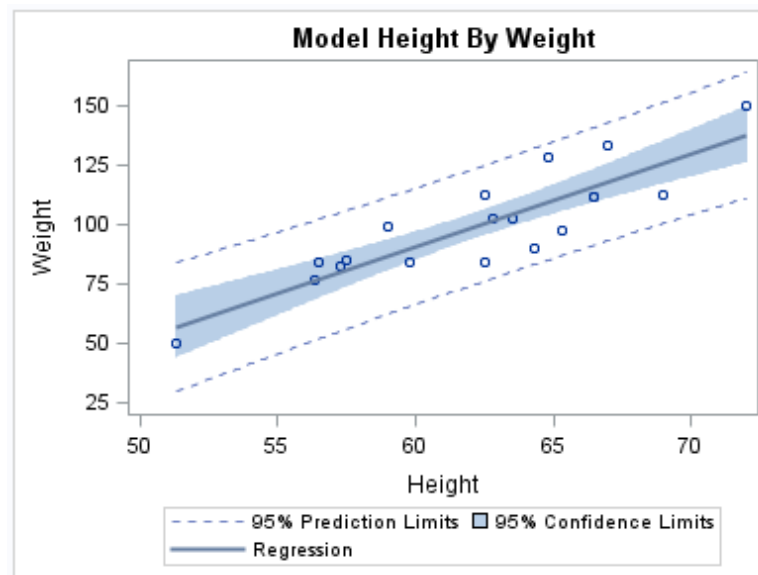
The Templates window contains the item stores **Sasuser.Templat** and **Sashelp.Tmplmst**.

2. Double-click **Sashelp.Tmplmst** to expand the list of directories where ODS templates are stored.
3. To view the style definitions that SAS provides, double-click the Styles item store.
4. Right-click the style definition that you want to view and select **Open**. The style definition is displayed in the Template Browser window.

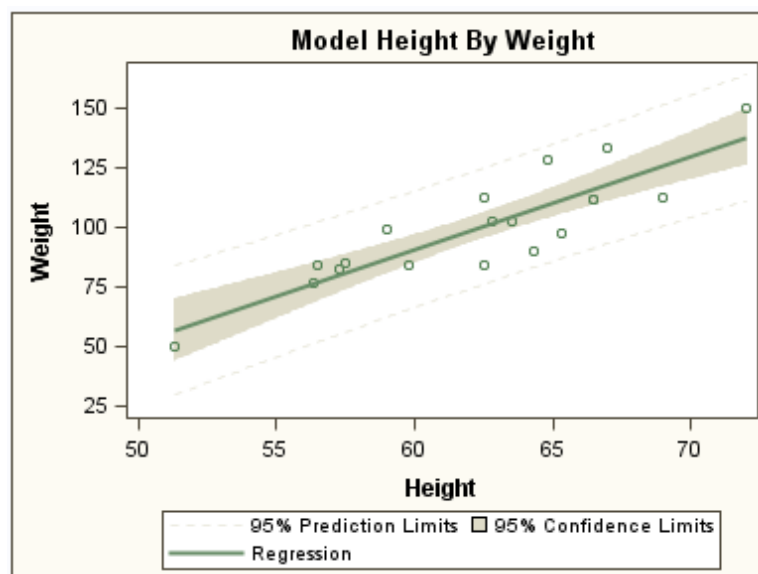
Examples of Styles

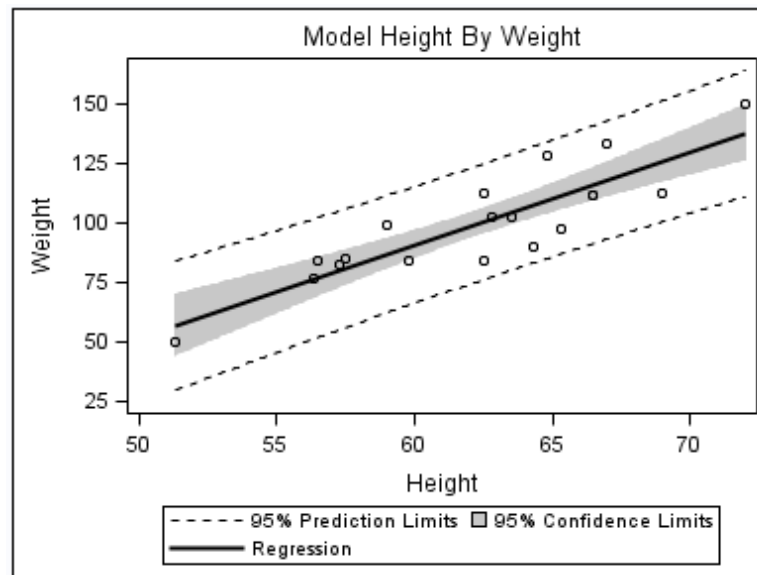
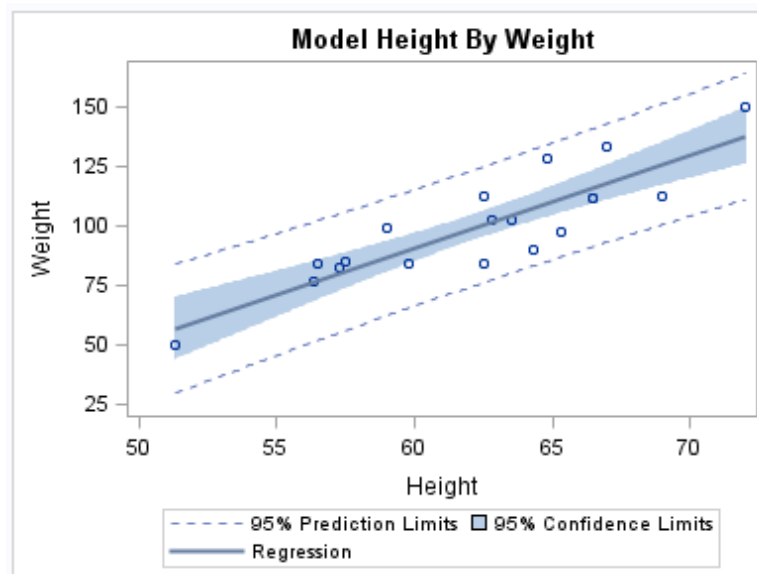
SAS provides styles that are recommended for use with statistical graphics. Each of the following graphs was created using the HTML destination with a different style.

Display 13.1 HTML Output Using the HTMLBlue Style



Display 13.2 HTML Output Using the Analysis Style



Display 13.3 HTML Output Using the Journal Style**Display 13.4** HTML Output Using the Statistical Style

Specifying Styles

Specify a Style for an ODS Destination

Each ODS destination has a default style that is set by SAS. By specifying `STYLE=style-definition` in your ODS destination statement, you can create an entirely different appearance for your graphs.

For example, you can specify that ODS apply the Styles.Journal style to all HTML output with one of the following statements:

```
ods html style=styles.journal;
ods html style=journal;
```

This style is applied to all output for that destination until you change the style or start a new SAS session.

Note: In Windows and UNIX operating environments, when the ODS Graphics procedures are executed in the SAS Windowing environment, HTML is the default destination. If you close this destination and do not open another destination, then no destinations are open. (In batch mode, the default destination is LISTING.)

Change the Default Style for the HTML Destination

The default style for the HTML destination is HTMLBlue.

To change the default style:

1. Select **Tools** ⇒ **Options** ⇒ **Preferences**.
2. Select the **Results** tab.
3. Select a new default style from the **Style** list box.
4. Click **OK**.

Changing a Default Style in the SAS Registry

By default, the SAS registry is configured to apply a default style to the output for each ODS destination. To permanently change the default style for a destination, you can change the setting of **Selected Style** in the SAS registry.

For more information about ODS destination settings in the SAS registry, see “Changing ODS Destination Default Settings” in Chapter 3 of *SAS Output Delivery System: User's Guide*.

Using Procedure Options to Control Graph Appearance

Overview of the Appearance Options

ODS styles enable you to control the overall appearance of the graphs. The ODS Graphics procedures also enable you to control the appearance of particular graphics elements in a graph. Graphics elements include lines, bars, markers, text, and so on.

Many ODS Graphics procedure statements have options and suboptions that control the appearance of different parts of a graph. Default visual attributes of various graphics elements are derived from the specific style elements of the active style. By using appearance options in your procedure statements, you can change the appearance of one or more aspects of your graph without changing the overall style.

For example, the following statement specifies that the mean value confidence limits (for a regression plot) use the GraphConfidence2 style element instead of the default GraphConfidence style element:

```
reg x=height y=weight / clmattrs=GraphConfidence2;
```

You can specify values for appearance options by using three different methods:

- specify a style element.

```
density height / lineattrs=graphfit2;
```

Note: This is the preferred method.

See [“Specifying a Style Element” on page 626](#).

- specify hardcoded values.

```
density height / lineattrs=(pattern=dashdashdot color=red);
```

This method overrides one or more attributes of the style element. See [“Specifying Hardcoded Values” on page 627](#).

- specify a style element, but override one or more attributes with hardcoded values. This method combines the first two methods.

```
density height / lineattrs=graphfit2 (pattern=dashdashdot);
```

See [“Specifying a Style Element with Hardcoded Values” on page 629](#).

The appearance options and their values are specific to each statement. For complete documentation about the statements and their options, see the syntax section for the appropriate procedure and statement.

See Also

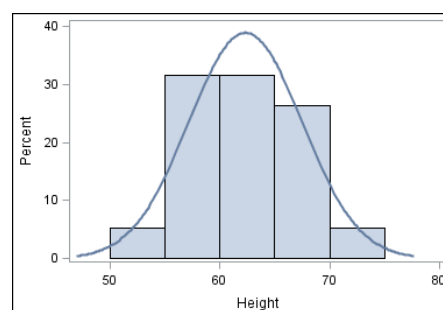
[“About Styles and Style Elements” on page 620](#)

Specifying a Style Element

You can change the visual attributes that are used for a graphics element by specifying a particular style element for any appearance option. Graphics elements include lines, bars, markers, text, and so on.

For example, the graph shown here is a histogram with a normal density curve. By default, the visual attributes of the density curve are specified by the style attributes within the style element GraphFit.

Table 13.3 Histogram with Density Curve That Uses the Default Style Element



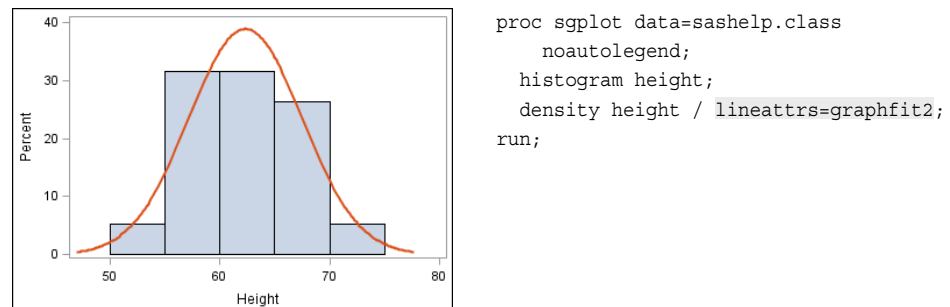
```
ods graphics / width=4in;
proc sgplot data=sashelp.class
  noautolegend;
  histogram height;
  density height;
run;
```

You can use the `LINEATTRS=` option in the `DENSITY` statement to change the style element that is used for the line attributes. For example, when you are using the `HTMLBlue` style, specifying the `GraphFit2` style element changes the density curve line to an orange color.

```
density height / lineattrs=graphfit2;
```

The full code is below.

Table 13.4 Histogram with Density Curve That Specifies a Style Element



Specifying a style element is the preferred way to change the appearance of a graphics element. Visual attributes that are referenced by the style element are chosen to provide consistency and appropriate emphasis based on display principles for statistical graphics. If you later change the style for the graph, the graphics element is compatible with the new style.

See Also

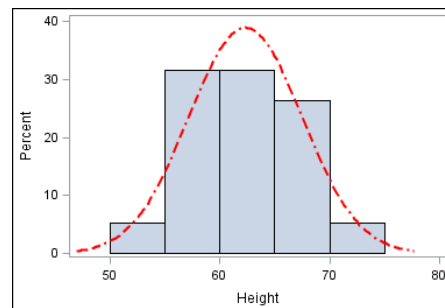
- [“About Styles and Style Elements” on page 620](#)
- [“Style Elements for Use with ODS Graphics” on page 640](#)

Specifying Hardcoded Values

In some cases it is important to have a specific visual effect, but it is not necessary for the plot to be reused with different styles. In such a case, you can specify hardcoded values for the visual attributes in the appearance option.

For example, you could use the following statement to set the density curve attributes:

```
density height / lineattrs=(pattern=dashdashdot color=red);
```

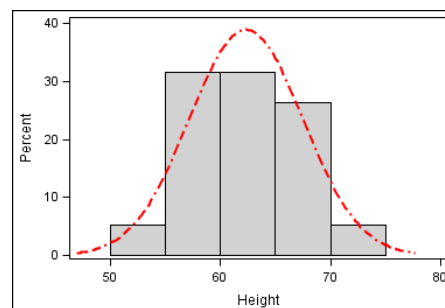
Table 13.5 Using Hardcoded Values

```
ods graphics / width=4in;
proc sgplot data=sashelp.class
    noautolegend;
    histogram height;
    density height /
        lineattrs=(pattern=dashdashdot color=red);
run;
```

The result is a red density curve with a DashDashDot pattern.

In general, it is recommended that you specify the style element rather than hardcoded values. Style elements are designed to derive all of the relevant visual attributes from the applied style. When you specify a hardcoded value for the line color, as in this example, the color is no longer derived from the specified style element. If you later change the style that is applied to the graph, the hardcoded attribute might conflict with the new style. Also, a hardcoded color could be exactly the same color as the wall color of another style, and become indistinguishable from the wall.

For example, if you change the overall style to Journal, which only uses gray-scale colors, the color for the curve is still red. This result might not be desirable.

Table 13.6 Hardcoded Attributes When the Journal Style Is Applied to the Graph

```
ods html style=journal;
proc sgplot data=sashelp.class
    noautolegend;
    histogram height;
    density height /
        lineattrs=(pattern=dashdashdot color=red);
run;
ods html style=htmlblue;
```

You use a standard syntax to hardcode colors, line thickness, line patterns, and marker symbols.

For more information about this syntax, see the following topics:

- “Specifying Colors in SAS/GRAPH Programs” in Chapter 14 of *SAS/GRAPH: Reference*
- “Units of Measurement” on page 659
- “Line Patterns” on page 663
- “Marker Symbols” on page 661

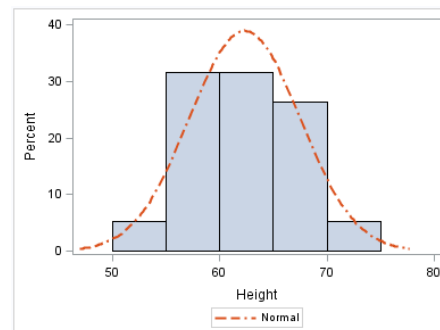
Specifying a Style Element with Hardcoded Values

You can specify a style element, but override one or more attributes of that style element with hardcoded values.

This example modifies a density curve. The example specifies the GraphFit2 style element for line attributes, but also specifies a particular line pattern with a hardcoded value.

```
density height / lineattrs=graphfit2 (pattern=dashdashdot);
```

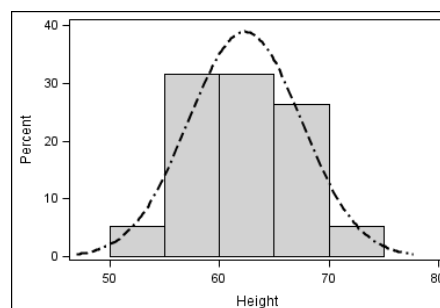
Table 13.7 Using a Hardcoded Value with a Style Element, Default Style



```
ods graphics / width=4in
  noautolegend;
proc sgplot data=sashelp.class;
  histogram height;
  density height / lineattrs=graphfit2
    (pattern=dashdashdot);
run;
```

If you change the overall style to Journal, the color for this curve changes as well. However, the pattern remains DashDashDot because it was hardcoded.

Table 13.8 Using a Hardcoded Value with a Style Element, Journal Style



```
ods html style=journal
  noautolegend;
proc sgplot data=sashelp.class;
  histogram height;
  density height / lineattrs=graphfit2
    (pattern=dashdashdot);
run;
ods html style=htmlblue;
```

Always be careful when you hardcode a value. When you specify a hardcoded value for the line pattern, as in this example, the pattern is no longer derived from the specified style element. If you later change the style that is applied to the graph, the hardcoded attribute might conflict with the new style.

Output for Grouped versus Non-Grouped Data

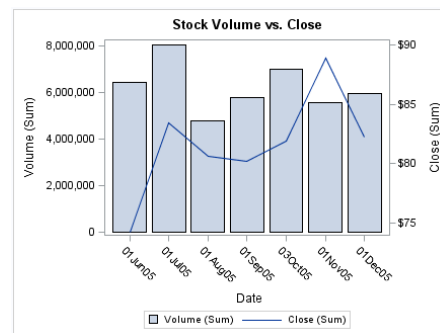
Non-Grouped Data

The Default Appearance of Non-Grouped Output

For the SGPLOT and SGPANEL procedures, many plots use the same style element, GraphDataDefault, for graphics elements such as lines, bars, and markers. These plots have the same default appearance.

In the graph that is produced by the following code, the bar chart and line chart use visual attributes from the GraphDataDefault element. The visual attributes for the line (and markers if specified) are designed to coordinate with the bar fill areas.

Table 13.9 Default Appearance of Non-Grouped Bars and Lines

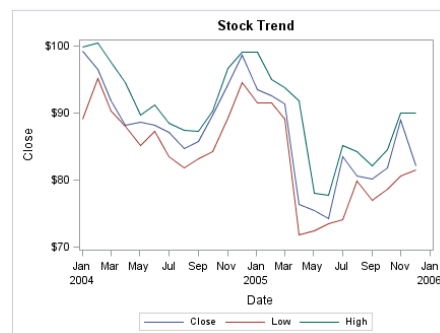


```
ods graphics on / width=4.5in;
proc sgplot data=sashelp.stocks
  (where=(date >= "01jun2005"d
    and stock = "IBM"));
  title "Stock Volume vs. Close";
  vbar date / response=volume;
  vline date / response=close y2axis;
run;
title;
```

However, when identical plot types are overlaid, the procedures instead use style elements named GraphData1 to GraphData*n* (where $n=12$ for most styles).

In the following graph, the series lines have different colors and are easy to distinguish. The plots use the GraphData1, GraphData2, and GraphData3 style elements.

Table 13.10 Default Appearance When Three Series Plots Are Overlaid



```
ods graphics on / width=4.5in;
proc sgplot data=sashelp.stocks
  (where=(date >= "01jan2004"d
    and stock = "IBM"));
  title "Stock Trend";
  series x=date y=close;
  series x=date y=low;
  series x=date y=high;
run;
title;
```

The GraphData*n* style elements use different marker symbols and fill colors to ensure that the plots differ in appearance. All line and marker colors are of different hues but with the same brightness, which means that all 12 colors can be distinguished but none

stands out more than another. Fill colors are based on the same hue but have less saturation, making them similar but more muted than the corresponding contrast colors.

Line patterns can also vary depending on the style that is used for the graph. Some styles vary line patterns so that elements in color plots can be distinguished even when the plot is sent to a black-and-white printer. Line patterns do not vary for the HTMLBlue style or any style that is defined with the ATTRPRIORITY="Color" option.

Note: If you have a bar-line overlay with multiple lines, the lines cycle through GraphData*n*, whereas the bar remains GraphDataDefault.

Other plots use more specialized style elements for their appearance. For example, the line attributes for density, loess, PBSpline, and regression curves are determined by the GraphFit style element.

Here are some interactions for plots with specialized style elements:

- If you have a histogram with multiple density curves, the curve style pattern changes with the number of curves. For one curve, GraphFit is used. For two curves, GraphFit and GraphFit2 are used. For three or more, all curves switch over to GraphData*n*. The histogram uses GraphDataDefault.
- If you have multiple fit plots (loess, regression, or PBSpline), the behavior is like the density curves. For one fit, GraphFit is used. For two fits, GraphFit and GraphFit2 are used. For three or more, all fits switch to GraphData*n*. (This behavior also applies to the SGSCATTER procedure when multiple fits are specified.)
- If you have multiple ellipses overlaid on another plot (such as a scatter), the ellipses cycle through GraphData*n*, whereas the plot uses GraphDataDefault.

In general, if all of the plots have the same statement name (all scatter, all series, and so on), then the plots automatically cycle.

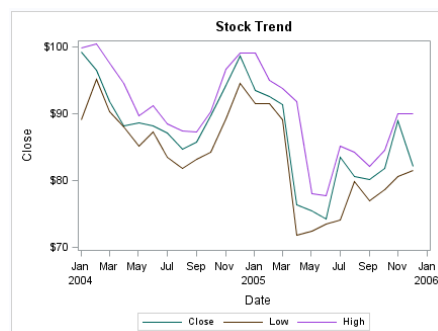
The CYCLEATTRS | NOCYCLEATTRS options in the SGPLOT and SGPANEL procedure statements can be used to manually control whether the plots have unique attributes. CYCLEATTRS forces cycling in situations where cycling is not automatic. The NOCYCLEATTRS option forces cycling to be off. These options are described in the next section.

Changing the Appearance of Non-Grouped Output

You can use plot options to specify particular GraphData*n* style elements for your plots in order to achieve a different appearance.

In the following graph, the style elements GraphData3, GraphData4, and GraphData5 are used to change the default appearance of the series lines in the graph.

Table 13.11 Overlaid Series Plots with Particular GraphData*n* Elements Specified



```
ods graphics on / width=4.5in;
proc sgplot data=sashelp.stocks
  (where=(date >= "01jan2004"d
    and stock = "IBM"));
  title "Stock Trend";
  series x=date y=close
    / lineattrs= GraphData3;
  series x=date y=low
    / lineattrs= GraphData4;
  series x=date y=high
    / lineattrs= GraphData5;
run;
title;
```

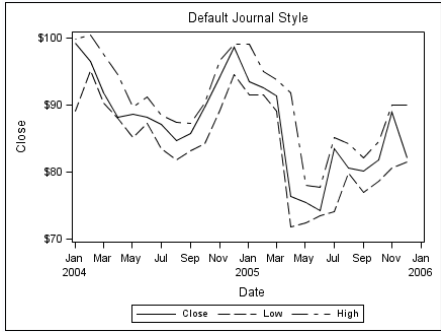
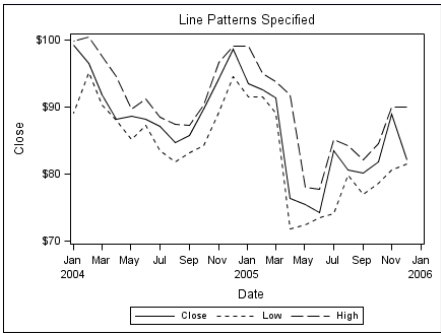
Note: To achieve a different appearance for the series lines, you can use any style element that has line properties. However, the `GraphData` style elements are ideally suited for this purpose. The `GraphData` style elements have been carefully constructed with different hues but with the same brightness, which means that all 12 colors can be distinguished but none stands out more than another.

For more information, see [“Specifying a Style Element” on page 626](#).

Sometimes, you might want to control one or more attributes of a style element. You can use plot options to override, or hardcode, an attribute in the style element.

The following two graphs show the same plots before and after the `PATTERN=` option is used. The `PATTERN=` option is used with two series lines to force a different line pattern.

Table 13.12 The Effect of Specifying a Hardcoded Line Pattern

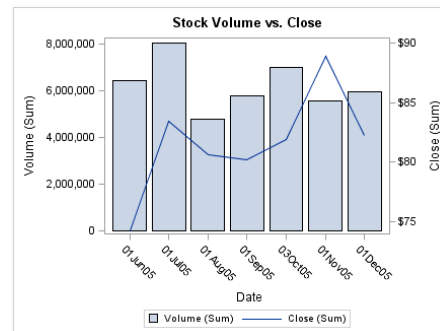
	<pre>ods html style=journal; proc sgplot data=sashelp.stocks (where=(date >= "01jan2004"d and stock = "IBM")); title "Default Journal Style"; series x=date y=close; series x=date y=low; series x=date y=high; run; title;</pre>
	<pre>proc sgplot data=sashelp.stocks (where=(date >= "01jan2004"d and stock = "IBM")); title "Line Patterns Specified"; series x=date y=close; series x=date y=low / lineattrs= (pattern=2); series x=date y=high / lineattrs= (pattern=4); run; title;</pre> <p style="text-align: right;">ods html style=htmlblue;</p>

For more information, see [“Specifying Hardcoded Values” on page 627](#) and [“Specifying a Style Element with Hardcoded Values” on page 629](#).

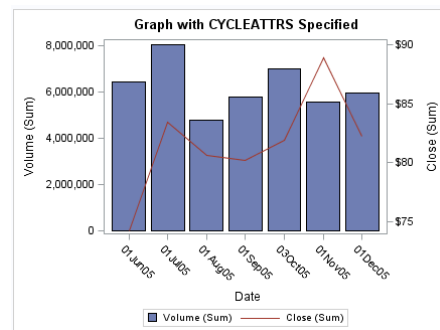
You can also use the `CYCLEATTRS` | `NOCYCLEATTRS` options in the `SGPLOT` and `SGPANEL` procedure statements to control whether the plots have unique attributes. `CYCLEATTRS` forces cycling in situations where cycling is not automatic. If the plots do not have unique attributes by default, then you can specify the `CYCLEATTRS` option to force unique attributes for each plot in the graph. The `NOCYCLEATTRS` option prevents the procedure from assigning unique attributes.

The following two graphs show the same plots before and after the CYCLEATTRS option is used.

Table 13.13 The Effect of Specifying the CYCLEATTRS Option



```
ods graphics on / width=4.5in;
proc sgplot data=sashelp.stocks;
  where(date >= "01jun2005"d and
        stock = "IBM");
  title "Stock Volume vs. Close";
  vbar date / response=volume;
  vline date / response=close y2axis;
run;
title;
```



```
proc sgplot data=sashelp.stocks
  CYCLEATTRS ;
  where(date >= "01jun2005"d and
        stock = "IBM");
  title "Graph with CYCLEATTRS Specified";
  vbar date / response=volume;
  vline date / response=close y2axis;
run;
title;
```

In the second graph, the procedure retrieves the attributes from GraphData1 and GraphData2. The first graph retrieves all of its data attributes from GraphDataDefault.

Grouped Data

The Appearance of Grouped Output

The GROUP= option is used to plot data when a classification or grouping variable is available. By default, this option automatically uses the GraphData n style elements for the presentation of each unique group value. In general, you cannot specify the line or marker properties for specific group values directly, as you can for non-grouped data.

Consider the following option specification for a line: LINEATTRS= (PATTERN=2). When this line option is used, the line pattern is set as a style override. However, this override applies equally to all group values. The same is true for overrides of color or line thickness. This means that you can set one or more fixed appearance attributes for all group values, but you cannot directly assign unique visual properties to an individual group value. For example, you cannot specify LINEATTRS= (PATTERN=2 4) because this statement is not valid.

You can, however, use attribute maps to specify particular visual attributes for group values of the data. Attribute mapping is explained in the following section.

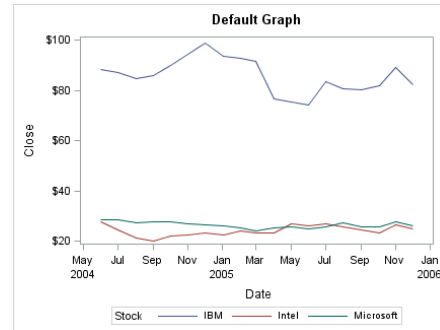
Making the Appearance of Grouped Data Independent of Data Order

When unique group values are gathered, they are internally recorded in the order in which they appear in the data. They are not subsequently sorted. This means that if an

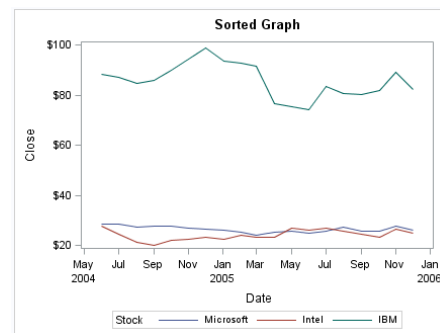
input data source is modified, sorted, or filtered, the order of the group values and their associations with `GraphData` might change.

The following two graphs show the same plots before and after the data is sorted.

Table 13.14 The Effect That Sorting Has on Grouped Plot Attributes



```
ods graphics on / width=4.5in;
proc sgplot data=stocks;
  where date >= "01jun2004"d;
  title "Default Graph";
  series x=date y=close / group= stock;
run;
title;
```



```
proc sort data=sashelp.stocks out=stocks;
  by descending stock;
run;

proc sgplot data=stocks;
  where date >= "01jun2004"d;
  title "Sorted Graph";
  series x=date y=close / group= stock;
run;
title;
```

Because the position of the data in the sorted data set has changed, the sorted graph uses different `GraphData` style elements for the group values.

In many cases, this might not be a problem if you do not care which line pattern, marker symbols, or colors are associated with particular group values. However, in some cases you might care. For example, if you create many plots grouped by gender, you might want a consistent set of visual properties for females and males across plots, regardless of the input data order.

SG attribute maps enable you to ensure that particular visual attributes are applied based on the `GROUP=` value of the data instead of the position of the data in the data set.

Two main tasks are required for attribute mapping. They are summarized here:

1. Create an SG attribute map data set, which associates data values with particular visual attributes. Each observation defines the attributes for a group value.
2. Modify the SG procedure and its plot statements to use the data in the SG attribute map. You can use attribute maps in the `SGPLOT`, `SGPANEL`, and `SGSCATTER` procedures.

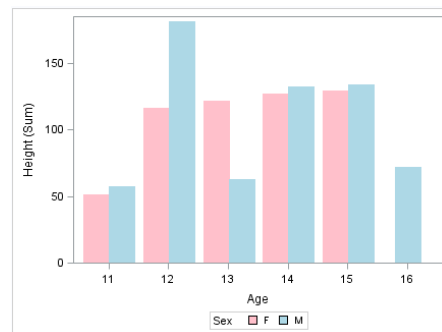
Attribute mapping is supported in the `PLOT` statement of the `SGSCATTER` procedure. In the `SGPLOT` and `SGPANEL` procedures, any plot statement that supports `GROUP` variables supports attribute mapping. For more information, see [Table 12.1 on page 610](#).

Here is an example of an SG attribute map data set called MYATTRMAP. The observations in this data set contain the group value (VALUE) and the attributes (LINECOLOR, FILLCOLOR).

Display 13.5 Listing of the SG Attribute Map Data Set MYATTRMAP

Obs	linecolor	fillcolor	ID	value
1	pink	pink	myid	F
2	lightblue	lightblue	myid	M

The following output shows a bar chart that uses the MYATTRMAP data set for its pink and blue bar colors. The FILLCOLOR= values determine the color of the bars, and the LINECOLOR= values determine the color of the border edges around the bars.



```
data myattrmap;
length linecolor $ 9 fillcolor $ 9;
input ID $ value $ linecolor $ fillcolor $;
datalines;
myid F pink pink
myid M lightblue lightblue
;
run;

proc sgplot data=sashelp.class
  dattrmap=myattrmap;
  vbar age / response=height group=sex
    groupdisplay=cluster attrid=myid;
run;
```

You use a standard syntax to specify colors, line thickness, line patterns, and marker symbols. For more information about this syntax, see the following topics:

- “Specifying Colors in SAS/GRAPH Programs” in Chapter 14 of *SAS/GRAPH: Reference*
- “Units of Measurement” on page 659
- “Line Patterns” on page 663
- “Marker Symbols” on page 661

For complete information about attribute maps, see [Chapter 12, “Using SG Attribute Maps to Control Visual Attributes,”](#) on page 605.

Modifying Style Templates

Style templates are created and modified with the TEMPLATE procedure. You can use the DEFINE STYLE statement to create your own style from scratch or from an existing style. When you create styles from existing styles, you can modify the individual style elements.

For example, the following program shows the style element `GraphDataDefault`, as defined in the `Default` style:

```
proc template;
  define style Styles.Default;

  ... more style elements ...

  class GraphDataDefault /
    endcolor = GraphColors("gramp3cend")
    neutralcolor = GraphColors("gramp3cneutral")
    startcolor = GraphColors("gramp3cstart")
    markersize = 7px
    markersymbol = "circle"
    linethickness = 1px
    linestyle = 1
    contrastcolor = GraphColors("gcdata")
    color = GraphColors("gdata");
```

You can use the `DEFINE STYLE` statement to create a new style from the `Default` style and modify the `GraphDataDefault` style element.

The following program creates the new style `MyStyleDefault`, which inherits all of its style elements and style attributes from the `Default` style, and customizes the `GraphDataDefault` style element:

```
proc template;
  define style MyStyleDefault;
  parent=Styles.Default;
  style GraphDataDefault from GraphDataDefault /
    markersize = 10px
    markersymbol = "square";
  end;
run;
```

For complete documentation on using `PROC TEMPLATE` to modify and create styles, see the *SAS Output Delivery System: User's Guide*.

Using Fill Patterns to Distinguish Grouped Bar Charts

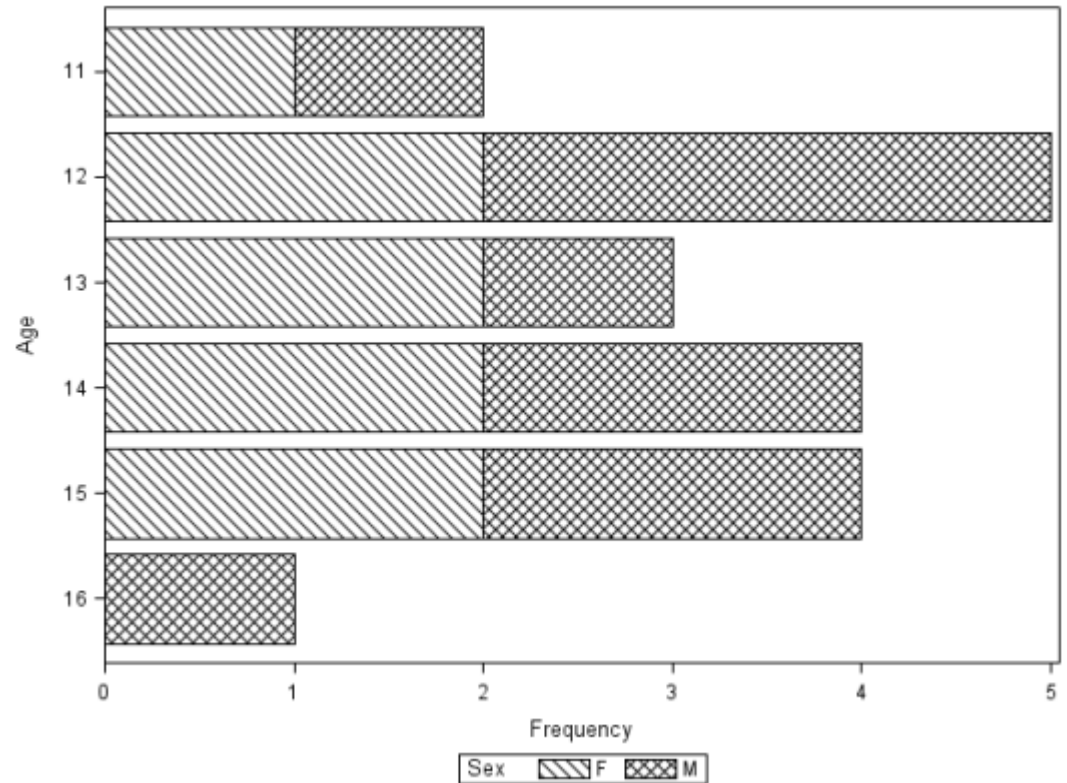
Some SAS styles display fill patterns for grouped bars. Fill patterns help distinguish between different categories when you are producing or printing black and white output.

The fill patterns appear for grouped data in the following types of plots, whether generated using the `SGPLOT` or the `SGPANEL` procedure:

- `HBAR` and `VBAR`
- `HBARPARM` and `VBARPARM`

Here is example output that was generated using the JOURNAL2 style.

Display 13.6 Fill Pattern for Grouped Bars



The following code was used to produce the example output.

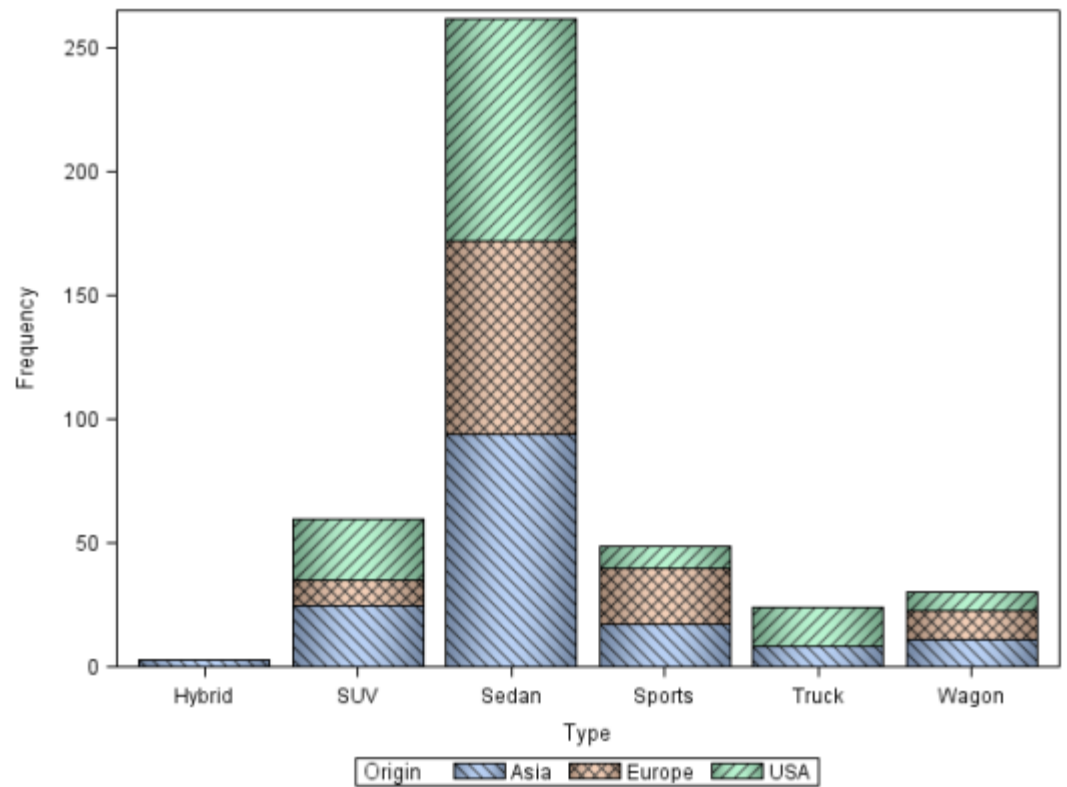
```
ods listing style=journal2;
proc sgplot data=sashelp.class;
hbar age / group= sex;
run;
ods listing close;
```

Here are the SAS styles that by default render grouped bars with fill patterns:

- JOURNAL2
- JOURNAL3 (uses gray and the fill pattern)
- MONOCHROMEPRINTER

These styles use the GRAPHBAR style element for the fill pattern. If you customize a style in the GTL, then you can add the GRAPHBAR style element to your custom style.

The following example shows a graph that uses a custom style. The style, based on the Journal2 style (which uses the GRAPHBAR element), adds color style elements. The result combines patterns, colors, and skins.

Display 13.7 Fill Pattern with Color

The following code was used to produce the output.

```
proc template;
  define style Styles.MyJournal2;
    parent = styles.journal2;
    style GraphColors from GraphColors /
      'gdata'=cxffffff
      'gdata1'=cxffffff
      'gdata2'=cxc0c0c0
      'gdata3'=cxe0e0e0
      'gdata4'=cxa0a0a0
      'gdata5'=cx909090
      'gdata6'=cxffffff
      'gdata7'=cxc0c0c0
      'gdata8'=cxe0e0e0
      'gdata9'=cxa0a0a0
      'gdata10'=cx909090
      'gdata11'=cxffffff
      'gdata12'=cxc0c0c0
      'gdata13'=cxe0e0e0
      'gdata14'=cxa0a0a0
      'gdata15'=cx909090;
    style GraphHistogram from GraphComponent /
      displayopts = "outline";
    style GraphEllipse from GraphComponent /
      displayopts = "outline";
    style GraphBand from GraphComponent /
      displayopts = "outline";
    style GraphBox from GraphComponent /
```

```

        displayopts = "caps median mean outliers"
        connect = "mean"
        capstyle = "serif";
    style GraphBar from GraphComponent /
        displayopts = "outline fillpattern";
    style GraphData1 from GraphData1 /
        fillpattern = "L1";
    style GraphData2 from GraphData2 /
        fillpattern = "X1";
    style GraphData3 from GraphData3 /
        fillpattern = "R1";
    style GraphData4 from GraphData4 /
        fillpattern = "L2";
    style GraphData5 from GraphData5 /
        fillpattern = "X2";
    style GraphData6 from GraphData6 /
        fillpattern = "R2";
    style GraphData7 from GraphData7 /
        fillpattern = "L3";
    style GraphData8 from GraphData8 /
        fillpattern = "X3";
    style GraphData9 from GraphData9 /
        fillpattern = "R3";
    style GraphData10 from GraphData10 /
        fillpattern = "L4";
    style GraphData11 from GraphData11 /
        fillpattern = "X4";
    style GraphData12 from GraphData12 /
        fillpattern = "R4";
    style GraphData13 from GraphComponent /
        fillpattern = "L5";
    style GraphData14 from GraphComponent /
        fillpattern = "X5";
    style GraphData15 from GraphComponent /
        fillpattern = "R5";
end;

define style Styles.MyJournal3;
    parent = styles.MyJournal2;
    style GraphBar from GraphComponent /
        displayopts = "fill outline fillpattern";
    style GraphColors from GraphColors /
        'gdata'=cxxxxfff
        'gdata1'=cxa0c0f0
        'gdata2'=cxf0c0a0
        'gdata3'=cxa0f0c0;
end;
run;

ods listing style=MyJournal3;
ods graphics / reset imagename='MyBarPatterns';
proc sgplot data=sashelp.cars;
vbar type / group= origin dataskin= pressed;
run;
ods listing close;

```

Style Elements for Use with ODS Graphics

Style elements affect ODS graphics and can be specified by GTL appearance options or used in styles. Attribute values can be changed with the `TEMPLATE` procedure.

In the ODS Graphics procedures, you can specify style elements to override the default appearance of particular graphics elements, such as data labels. Certain style elements were created to be used with specific plots. For example, the style element `GraphFit2` is best used to modify secondary fit lines. The style element `GraphConfidence2` was created to modify secondary confidence bands. However, many plots provide the option to override the style element with a different style element, or to specify particular attributes for a style element.

Here are examples that specify a style element and an attribute of a style element:

- `GraphData2` is an example of a style element.
- `GraphData2:Color` is an example of a style attribute.

The tables below list each style element, the portion of the graph that it affects, and the applicable attributes.

Table 13.15 Graph Style Elements: General Graph Appearance

Style Element	Portion of Graph Affected	Recognized Attributes
Graph	Graph size and outer border appearance	OutputWidth OutputHeight BorderColor BorderWidth CellPadding CellSpacing
GraphAnnoLine	Annotation lines	ContrastColor LineStyle LineThickness
GraphAnnoShape	Annotation closed shapes such as circles, and squares	Color ContrastColor LineThickness LineStyle Transparency
GraphAnnoText	Annotation text	Font or <i>font-attributes</i> [*] Color MarkerSize MarkerSymbol

Style Element	Portion of Graph Affected	Recognized Attributes
GraphAxisLines	X, Y and Z axis lines	ContrastColor LineStyle LineThickness TickDisplay
GraphBackground	Background of the graph	Color Transparency
GraphBorderLines	Border around graph wall, legend border, borders to complete axis frame	ContrastColor LineThickness LineStyle
GraphDataText	Text font and color for point and line labels	Font or <i>font-attributes</i> * Color
GraphFootnoteText	Text font and color for footnote(s)	Font or <i>font-attributes</i> * Color
GraphLabelText	Text font and color for axis labels and legend titles	Font or <i>font-attributes</i> * Color
GraphOutlines	Outline properties for fill areas such as bars, pie slices, box plots, ellipses, and histograms	Color ContrastColor LineStyle LineThickness
GraphReference	Horizontal and vertical reference lines and drop lines	ContrastColor LineStyle LineThickness
GraphTitleText	Text font and color for title(s)	Font or <i>font-attributes</i> * Color
GraphUnicodeText	Text font for unicode values	Font or <i>font-attributes</i> * Color

Style Element	Portion of Graph Affected	Recognized Attributes
GraphValueText	Text font and color for axis tick values and legend values	Font or <i>font-attributes</i> * Color

* *Font-attributes* can be one of the following: FONTFAMILY=, FONTSIZE=, FONTSTYLE=, FONTWEIGHT=.

Table 13.16 Style Elements Affecting Graphical Data Representation

Style Element	Portion of Graph Affected	Recognized Attributes
GraphBoxMean	Marker for mean	ContrastColor MarkerSize MarkerSymbol
GraphBoxMedian	Line for median	ContrastColor LineStyle LineThickness
GraphBoxWhisker	Box whiskers and serifs	ContrastColor LineStyle LineThickness
GraphConfidence	Primary confidence lines and bands, colors for bands and lines	ContrastColor Color MarkerSize MarkerSymbol LineStyle LineThickness
GraphConfidence2	Secondary confidence lines and bands, color for bands, and contrast color for lines	ContrastColor Color MarkerSize MarkerSymbol LineStyle LineThickness
GraphConnectLine	Line for connecting boxes or bars	ContrastColor LineStyle LineThickness
GraphCutLine	Cutline attributes for a dendogram	Color LineStyle

Style Element	Portion of Graph Affected	Recognized Attributes
GraphDataDefault	Primitives related to non-grouped data items, colors for filled areas, markers, and lines	Color ContrastColor MarkerSymbol MarkerSize LineStyle LineThickness StartColor NeutralColor EndColor
GraphError	Error line or error bar fill, ContrastColor for lines, Color for bar fill	ContrastColor Color LineStyle Transparency
GraphFit	Primary fit lines such as a normal density curve	ContrastColor Color MarkerSize MarkerSymbol LineStyle LineThickness
GraphFit2	Secondary fit lines such as a kernel density curve	ContrastColor Color MarkerSize MarkerSymbol LineStyle LineThickness
GraphFinal	Final data for the waterfall chart. Color applies to filled areas.	Color ContrastColor LineStyle LineThickness MarkerSize MarkerSymbol TextColor

Style Element	Portion of Graph Affected	Recognized Attributes
GraphInitial	Initial data for the waterfall chart. Color applies to filled areas.	Color ContrastColor LineStyle LineThickness MarkerSize MarkerSymbol TextColor
GraphMissing	Properties for graph items representing missing values	ContrastColor Color MarkerSymbol MarkerSize LineStyle LineThickness Transparency
GraphOther	Other data for the graph. Color applies to filled areas.	Color ContrastColor LineStyle LineThickness MarkerSize MarkerSymbol TextColor
GraphOverflow	Overflow data for the graph. Color applies to filled areas. ContrastColor applies to markers and lines.	Color ContrastColor LineStyle LineThickness MarkerSize MarkerSymbol TextColor
GraphOutlier	Outlier data for the graph	ContrastColor Color MarkerSize MarkerSymbol LineStyle LineThickness

Style Element	Portion of Graph Affected	Recognized Attributes
GraphPrediction	Prediction lines	ContrastColor Color LineStyle LineThickness MarkerSize MarkerSymbol
GraphPredictionLimits	Fills for prediction limits	ContrastColor Color MarkerSize MarkerSymbol
GraphUnderflow	Underflow data for the graph. Color applies to filled areas. ContrastColor applies to markers and lines.	Color ContrastColor LineStyle LineThickness MarkerSize MarkerSymbol TextColor
GraphSelection	For interactive graphs, visual properties of selected item. Color for selected fill area, ContrastColor for selected marker or line.	ContrastColor Color MarkerSymbol MarkerSize LineStyle LineThickness
ThreeColorAltRamp	Line contours, markers, and data labels with segmented range color response	StartColor NeutralColor EndColor
ThreeColorRamp	Gradient contours, surfaces, markers, and data labels with continuous color response	StartColor NeutralColor EndColor
TwoColorAltRamp	Line contours, markers, and data labels with segmented range color response	StartColor EndColor
TwoColorRamp	Gradient contours, surfaces, markers, and data labels with continuous color response	StartColor EndColor

Table 13.17 Graphical Style Elements: Data Related (Grouped)

Style Element	Portion of Graph Affected	Recognized Attributes
GraphData1	Primitives related to 1st grouped data items. Color applies to filled areas. ContrastColor applies to markers and lines.	Color ContrastColor MarkerSymbol LineStyle
GraphData2	Primitives related to 2nd grouped data items	Color ContrastColor MarkerSymbol LineStyle
GraphData3	Primitives related to 3rd grouped data items	Color ContrastColor MarkerSymbol LineStyle
GraphData4	Primitives related to 4th grouped data items	Color ContrastColor MarkerSymbol LineStyle
GraphData5	Primitives related to 5th grouped data items	Color ContrastColor MarkerSymbol LineStyle
GraphData6	Primitives related to 6th grouped data items	Color ContrastColor MarkerSymbol LineStyle
GraphData7	Primitives related to 7th grouped data items	Color ContrastColor MarkerSymbol LineStyle
GraphData8	Primitives related to 8th grouped data items	Color ContrastColor LineStyle
GraphData9	Primitives related to 9th grouped data items	Color ContrastColor LineStyle

Style Element	Portion of Graph Affected	Recognized Attributes
GraphData10	Primitives related to 10th grouped data items	Color ContrastColor LineStyle
GraphData11	Primitives related to 11th grouped data items	Color ContrastColor LineStyle
GraphData12	Primitives related to 12th grouped data items	Color ContrastColor

Table 13.18 Display Style Elements

Style Element	Portion of Graph Affected	Recognized Attributes	Possible Values
GraphAltBlock	Alternate fill color for block plots	Color	GraphColors("gablock")
GraphBand	Display options for confidence bands	DisplayOpts	"Fill "
GraphBar	Display options for bar charts	DisplayOpts	"Fill outline"
GraphBox	Display options for box plots	DisplayOpts CapStyle Connect	"Fill caps mean Median outliers " "Serif" "Mean"
GraphBlock	Fill color for block plots	Color	GraphColors("gblock")
GraphEllipse	Display options for confidence ellipses	DisplayOpts	"Outline"
GraphHistogram	Display options for histograms	DisplayOpts	"Fill outline"

Chapter 14

Managing Your Graphics with ODS

Introduction	649
Specifying a Destination	649
Using the ODS GRAPHICS Statement	651
Introduction	651
Controlling the Size of Your Graph	652
Adding Data Tips and Other Features	654
Resetting ODS GRAPHICS Options	654
Accessing Graphs as Separate Files	654

Introduction

The Output Delivery System (ODS) manages all output created by procedures and enables you to display the output in a variety of forms, such as HTML, PDF, and RTF. SAS ODS Graphics procedures and many SAS Analytical procedures use ODS Graphics for creation of their graphs. The ODS destination statements and the ODS GRAPHICS statement provide options for control of many relevant features.

Note: Some default ODS output settings have changed for SAS 9.3. For more information, see [“What's New in SAS ODS Graphics Procedures 9.3” on page vii](#).

The following sections describe techniques for managing this graphical output:

- [“Specifying a Destination” on page 649](#)
- [“Using the ODS GRAPHICS Statement” on page 651](#)

Specifying a Destination

For creation of ODS graphs, a valid ODS destination must be active. In Windows and UNIX operating environments, when the ODS Graphics procedures are executed in the SAS Windowing environment, HTML is the default destination. The default destination for batch mode is LISTING.

CAUTION:

Take care when closing the HTML destination. If you close this destination and do not open another destination, then no destinations are open.

You can use ODS destination statement options to specify where you want your output to be displayed and where you want your image files to be stored.

These options are specified on ODS destination statements, using the following form:

ODS destination <option(s);>

The ODS HTML statement below specifies the following:

- the name of the output file
- the folder where images are stored
- the style to be used

For the complete program, see [Example Code 14.1 on page 651](#) .

```
ods html file="BoxPlot-Body.html"
  gpath="C:\myfiles\images" style=journal;

... SAS statements ...
```

The output is written to the file **BoxPlot-Body.html**. Images created by the output are stored in the folder "**C:\myfiles\images**". The **STYLE=** option changes the style from **HTMLBlue.Styles** to **Journal.Styles**. For more information about changing the appearance of your graphics, see [“Controlling the Appearance of Your Graphs” on page 619](#) .

If you do not specify a filename for your output, then SAS provides a default file that is determined by the ODS destination. You can check the SAS log to verify the name of the file in which your output is saved.

When the procedures are executed in the SAS Windowing environment, if you do not specify a directory for the file, the file is saved in the directory that corresponds to your SAS Work library. The default directory for batch mode is the SAS current directory.

TIP You can verify the location of the current SAS Work library by opening the Libraries folder in the SAS Explorer window. Right-click **Work** and select **Properties** from the pop-up menu.

For a table that lists ODS destinations and the formatted output that results from each destination, see “Understanding ODS Destinations” in Chapter 3 of *SAS Output Delivery System: User's Guide*.

Graphs rendered to the LISTING destination are not automatically displayed in the output window. To view these graphs, go to the Results window and select the desired graph to view.

You can also specify some appearance options for the graphics output such as image DPI, width, height, and style. For more information, see [“Using the ODS GRAPHICS Statement” on page 651](#).

See Also

- “Understanding ODS Destinations” in Chapter 3 of *SAS Output Delivery System: User's Guide*
- Chapter 6, “Dictionary of ODS Language Statements,” in *SAS Output Delivery System: User's Guide*

Using the ODS GRAPHICS Statement

Introduction

You can use the ODS GRAPHICS statement options to control many aspects of your graphics. The settings that you specify remain in effect for all graphics until you change or reset these settings with another ODS GRAPHICS statement.

When you use the GTL language or SAS ODS Graphics procedures in the SAS Windowing environment, ODS GRAPHICS is ON by default. In batch mode, ODS GRAPHICS is OFF by default.

You can use the ODS GRAPHICS statement to control many features, including the following:

- the size of the image
- the type and name of the image created
- whether features such as data tips, scaling, and anti-aliasing are used

The basic syntax for the ODS GRAPHICS statement is as follows:

ODS GRAPHICS < OFF | ON> </ option(s)>;

Note: For the complete statement syntax, see “ODS GRAPHICS Statement” in *SAS Output Delivery System: User's Guide*.

The following program uses the ODS GRAPHICS statement to illustrate several techniques for controlling your graphics.

These techniques are discussed in the following sections:

- [“Controlling the Size of Your Graph” on page 652](#)
- [“Adding Data Tips and Other Features” on page 654](#)
- [“Accessing Graphs as Separate Files” on page 654](#)

Note: The following sections provide only summary information. For the complete information, see “ODS GRAPHICS Statement” in *SAS Output Delivery System: User's Guide*.

Example Code 14.1 Controlling Graphics with the ODS GRAPHICS Statement

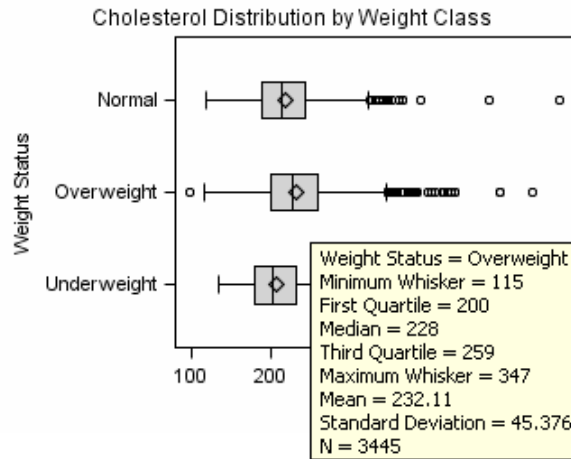
```
ods graphics on /
    width=3.25in
    outputfmt=gif
    imagemap=on
    imagename="MyBoxplot"
    border=off;

ods html file="Boxplot-Body.html" style=journal
gpath="C:\myfiles\images";

proc sgplot data=sashelp.heart;
    title "Cholesterol Distribution by Weight Class";
    hbox cholesterol / category=weight_status;
run;
```

```
ods html style=htmlblue;
ods graphics on / reset=all;
```

Display 14.1 3.25 Inch Graph with Data Tips and No Border



Controlling the Size of Your Graph

Overview

The output size of a graph is determined by the following:

- For SGRENDER output, the design size for the graph is specified in the STATGRAPH template.

The default sizes defined in the registry are:

- Default design width=640px
- Default design height=480px
- Default aspect ratio is 4/3

For SGPLOT, SGPANEL, or SGSCATTER output, the preceding default sizes are used.

- DPI settings for the output destinations are specified by the IMAGE_DPI= or DPI= options
- Settings for width and height are specified by the WIDTH= option and the HEIGHT= option for the ODS GRAPHICS statement.

When modifying the size of your graph, it is important to take into consideration the image scaling and aspect ratio, as well as the width and height of the graph. For example, rendering a graph to the LISTING destination at the default design size and default DPI of 100, produces a graph that is 640 by 480 pixels. If the same graph is rendered to the RTF destination, which has a default DPI of 200, the graph size is closer to 1280 by 960 pixels. The graph is scaled using 96 DPI as a baseline ($640 * 200 / 96$). In both cases, when this image is embedded into an RTF document, its size in the document is 6.4 in. by 4.8 in, since DPI data is included in the image. The aspect ratio of the graph is retained.

Graph Scaling for DPI

When the DPI of a graph is changed due to a change in output destination or user specification, the graph is scaled using a baseline DPI of 96 pixels. All graphical elements such as marker size, line thickness, font sizes, and gutters are also scaled linearly.

Specifying Output Size with the ODS GRAPHICS statement

You can control the output size of a graph by specifying the WIDTH= or HEIGHT= options in the ODS GRAPHICS statement. In the ODS GRAPHICS statement below, taken from [Example Code 14.1 on page 651](#), the WIDTH= option changes the width of the graph to 3.25 inches:

```
ods graphics on /
  width=3.25in
  outputfmt=gif
  imagemap=on
  imagename="MyBoxplot"
  border=off;
```

Retaining Aspect Ratio

Although you can use the ODS GRAPHICS statement to specify both the width and height, it is highly recommended that only one of these options be specified at a time. Using one option at a time ensures that the design aspect ratio of the graph is maintained.

When only width is specified, SAS uses the design aspect ratio of the graph to compute the appropriate height. Retaining the design aspect ratio of the graph is important in many cases as the graph might have a specific layout. For example, a plot that has multiple columns, or that has a statistics table on the side needs a wide aspect ratio. Changing the aspect ratio for this plot by specifying both width and height might produce unpredictable results.

Graph Scaling For User Specified Image Size

When you use the WIDTH= option in the ODS GRAPHICS statement, the output size of the graph is computed based on the new size and the active DPI, using a baseline DPI of 96. All graphical elements such as marker size, line thickness, font size, and gutters are scaled using a non-linear scale. This is done to prevent rapid shrinking or growth of the fonts and markers.

Scaling of graph elements for change in graph size can be disabled by the ODS GRAPHICS statement options NOSCALE or SCALE=OFF. By default, scaling is on. Specifying NOSCALE or SCALE=OFF prevents the scaling of the graph elements. If you want to shrink your graph and keep the default sizes for fonts, use the NOSCALE or SCALE=OFF option in the ODS GRAPHICS statement.

Tip For Embedding Graph Images in Documents

It is often useful to produce a graph that fits in one column of a two column page. In this case, the column size might be 3.25 inches wide. You can place a default output image, which is 640 by 480 pixels, in this space, but then all graphical elements scale down. This results in small markers and unreadable small fonts.

In this case, it is useful to render the graph to the exact size required, and specify the correct DPI. In the situation above, you would specify a width of 3.25 inches and use the IMAGE_DPI= destination statement option to specify a DPI of 200. Then, the graph produced can be placed in the 3.25" wide column and the fonts and markers are readable.

Note: To get exactly the specified font sizes, use the NOSCALE option in the ODS GRAPHICS statement.

Adding Data Tips and Other Features

With ODS GRAPHICS statement options, you can control different features of your graphs, such as borders and data tips. In the ODS GRAPHICS statement below, taken from [Example Code 14.1 on page 651](#), the BORDER=OFF option specifies that there be no border around the graph. The IMAGEMAP=ON option enables the generation of data tips and any drill downs:

```
ods graphics on /
  width=3.25in
  outputfmt=gif
  imagemap=on
  imagename="MyBoxplot"
  border=off;
```

When viewing HTML output, data tips appear when you move a mouse over certain features of the graph. Data tips and URL drill down functionality are available only for the HTML destination. You can also specify the maximum number of distinct mouse over areas allowed before data tips are disabled by using the TOOLTIPMAX= option. There are many other ODS GRAPHICS options that you can use to control other aspects of your graphics. For more information, see “ODS GRAPHICS Statement” in *SAS Output Delivery System: User's Guide*.

Resetting ODS GRAPHICS Options

You can specify the RESET option to change the values for these options back to their defaults. In [Example Code 14.1 on page 651](#), the second ODS GRAPHICS statement uses the RESET=ALL option to set all options back to their defaults for a new graph:

```
ods graphics on / reset=all;
```

You can also reset ODS GRAPHICS options individually.

Accessing Graphs as Separate Files

ODS enables you to change the ODS destination where you are sending your output to result in different output formats such as RTF or PDF. There might also be times when you want to access your graphs as individual image files.

The default image file type is determined by the ODS destination. You can use the OUTPUTFMT= option to specify a different image file format. The type of image allowed is dependent on the destination that you have specified.

You can also specify names for your graphics image files and the directory in which you want to save them. The IMAGENAME= option specifies the base image filename and the GPATH= option specifies a directory for your images.

In the ODS GRAPHICS statement below, taken from [Example Code 14.1 on page 651](#), the OUTPUTFMT= option specifies that the file type is a GIF, and the IMAGENAME= option specifies that the name of the image is “**MyBoxplot**”:

```
ods graphics on /
  width=3.25in
  outputfmt=gif
  imagemap=on
```

```

imagename="MyBoxplot"
border=off;

```

The resulting filename for the image created is **MyBoxplot.GIF**. If there is more than one image, they are named **MyBoxplot1.GIF**, **MyBoxplot2.GIF**, and so on.

The following ODS GRAPHICS statement specifies the RESET=ALL option, which changes the image filename and format to the defaults:

```
ods graphics on / reset=all;
```

You can use the GPATH= option to specify a directory for saving your graphics image files. For example, in [Example Code 14.1 on page 651](#) the GPATH= option on both of the ODS HTML statements specifies that all images be saved in the directory **C:\myfiles\images**.

See Also

- “Supported File Types for Output Destinations” in Chapter 6 of *SAS Output Delivery System: User's Guide*
- “Description of Supported File Types” in Chapter 6 of *SAS Output Delivery System: User's Guide*

See Also

“ODS GRAPHICS Statement” in *SAS Output Delivery System: User's Guide*

Part 6

Appendix

<i>Appendix 1</i>	
Units of Measurement	659
<i>Appendix 2</i>	
Marker Symbols	661
<i>Appendix 3</i>	
Line Patterns	663
<i>Appendix 4</i>	
ODS Graphics Software	665
<i>Appendix 5</i>	
Comparisons with the SAS/GRAPH Procedures	667

Appendix 1

Units of Measurement

Some options such as the LINEATTRS= option enable you specify the unit of measurement as part of the value.

The following table contains the units that are available:

Table A1.1 *Measurement Units*

Unit	Description
CM	centimeters
IN	inches
MM	millimeters
PCT or %	percentage
PT	point size, calculated at 100 dots per inch
PX	pixels

Appendix 2

Marker Symbols

The MARKERATTRS= option in some of the plot statements enables you to specify the marker symbol that is used to represent your data.

The following list shows the marker symbols that you can use:

Figure A2.1 List of Marker Symbols

↓ ArrowDown	▽ HomeDown	~ Tilde	× X	■ SquareFilled
* Asterisk	⊥ Ibeam	△ Triangle	Y Y	★ StarFilled
○ Circle	+ Plus	▽ TriangleDown	Z Z	▲ TriangleFilled
◇ Diamond	□ Square	◁ TriangleLeft	● CircleFilled	▼ TriangleDownFilled
> GreaterThan	☆ Star	▷ TriangleRight	◆ DiamondFilled	◀ TriangleLeftFilled
# Hash	⊥ Tack	∪ Union	▼ HomeDownFilled	▶ TriangleRightFilled

Appendix 3

Line Patterns

The LINEATTRS= option in some plot statements enables you to specify the line pattern that is used for the lines in your plot.

The following list shows the line patterns that you can use:

Figure A3.1 List of Line Patterns

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

Appendix 4

ODS Graphics Software

SAS ODS Graphics is an extension of the SAS Output Delivery System (ODS). ODS manages all output created by procedures and enables you to display the output in a variety of forms, such as HTML, PDF, and others.

For graphical output, ODS Graphics use the Graph Template Language (GTL) syntax, which provides the power and flexibility to create many complex graphs. The GTL is a comprehensive language for defining statistical graphics.

The ODS Graphics system contains the following software:

SAS ODS Graphics procedures

provide a concise syntax for creating effective statistical graphs. The procedures provide a traditional SAS procedure interface for the most commonly used features of the SAS Graph Template Language.

ODS GRAPHICS statement

adds graphics capabilities to many SAS analytical and Base procedures. The statement enables or disables ODS graphics processing and sets graphics environment options. By default, the ODS GRAPHICS statement is set to ON for SAS procedures that support ODS Graphics when the procedures are executed in the SAS windowing environment on Windows and UNIX operating systems. In batch mode, the ODS GRAPHICS statement is set to OFF by default.

You can use the ODS GRAPHICS statement options to control many aspects of your graphics. For example, you can use options to specify the size and format of your output images. The settings that you specify remain in effect for all graphics until you change or reset these settings with another ODS GRAPHICS statement.

SAS Graph Template Language (GTL)

provides a comprehensive language for creating statistical graphics. You can use the Graph Template Language to create customized layouts and graphs that are beyond the scope of the ODS Graphics procedures.

SAS ODS Graphics Editor

enables you to edit and enhance graphs that are produced by the ODS Graphics procedures or by the Graph Template Language.

The ODS graphics editor is an interactive editor that enables you to modify the elements of a graph or to add new features, such as titles, arrows, and text boxes.

SAS ODS Graphics Designer

enables you to create and design custom graphs. The ODS Graphics Designer provides a graphical user interface for designing graphs easily without having to know the details of the Graph Template Language.

See Also

- “ODS GRAPHICS Statement” in *SAS Output Delivery System: User's Guide*
- *SAS Graph Template Language: User's Guide*
- *SAS Graph Template Language: Reference*
- *SAS ODS Graphics Editor: User's Guide*
- *SAS ODS Graphics Designer: User's Guide*

Appendix 5

Comparisons with the SAS/GRAPH Procedures

SAS/GRAPH Output versus ODS Graphics	667
Differences between the ODS Graphics Procedures and SAS/GRAPH Procedures	668

SAS/GRAPH Output versus ODS Graphics

SAS produces graphics using two very distinct systems. SAS/GRAPH produces graphics using a device-based system. Base SAS produces graphics through the Output Delivery System (ODS) using a template-based system.

device-based graphics (SAS/GRAPH output)

output that is produced by SAS/GRAPH, which uses devices to generate output. Devices determine the type of output. Examples of device drivers are GIF, PNG, JPEG, ACTIVEX, SVG, and SASPRTC. Device drivers supplied by SAS are stored in the SASHELP.DEVICES catalog. Most procedures that produce device-based graphics also produce GRSEG catalog entries in addition to any image files, vector files, or displayed output that are produced. SAS/GRAPH procedures that produce device-based graphics and GRSEG catalog entries include the GCHART, GPLOT, GMAP, GBARLINE, GCONTOUR, and G3D procedures. The device-based procedures that do not produce GRSEG catalog entries are the GAREABAR, GKPI, and GTILE procedures. For device-based graphics, you can use the GOPTIONS statement to control the graphical environment. For example, you can specify which device is used to generate SAS/GRAPH output by specifying the DEVICE= option in the GOPTIONS statement. Information about device-based graphics is in this document, *SAS/GRAPH: Reference*.

template-based graphics (ODS Graphics)

output that is produced from a compiled ODS template of type STATGRAPH. Templates supplied by SAS are stored in SASHELP.TMPLMST. Device drivers and most SAS/GRAPH global statements (such as AXIS, LEGEND, PATTERN, and SYMBOL) have no effect on template-based graphics. The Base SAS procedures that produce template-based graphics are the SGPLOT, SGPANEL, SGSCATTER, SGDESIGN, and SGRENDER procedures. Many SAS/STAT, SAS/ETS, and SAS/QC procedures also produce template-based graphics automatically by default. Template-based graphics are always produced as image files and never as GRSEG catalog entries. For template-based graphics, you must use the ODS GRAPHICS statement to control the graphical environment. For example, you can specify the type of image file (GIF, PNG, JPEG, SVG, and so on) that is produced by specifying the OUTPUTFMT= option in the ODS GRAPHICS statement. Template-based

graphics are referred to as ODS Graphics. In addition to this *SAS ODS Graphics: Procedures Guide*, information about ODS Graphics is in *SAS ODS Graphics: Getting Started with Business and Statistical Graphics*, *SAS Graph Template Language: User's Guide*, *SAS Graph Template Language: Reference*, *SAS ODS Graphics Designer: User's Guide*, and *SAS ODS Graphics Editor: User's Guide*.

Differences between the ODS Graphics Procedures and SAS/GRAPH Procedures

SAS produces graphics using two very distinct systems. SAS/GRAPH produces graphics using a device-based system. The Graph Template Language and ODS Graphics produce graphics through the Output Delivery System (ODS) using a template-based system.

Though the ODS Graphics procedures do not require SAS/GRAPH, you might have SAS/GRAPH installed. For more information about SAS/GRAPH, see *SAS/GRAPH: Reference*.

The following table lists some of the differences between the ODS Graphics procedures and SAS/GRAPH:

Table A5.1 Differences between the ODS Graphics Procedures and SAS/GRAPH Procedures

ODS Graphics Procedures	SAS/GRAPH Procedures
Creates image files. GRSEGs and device drivers are not used. You can select your image format by using the OUTPUTFMT= option in the ODS GRAPHICS statement.	Most SAS/GRAPH procedures produce a GRSEG entry in a SAS catalog. Other output formats, such as an image or metagraphics file, can be created by selecting an appropriate device driver.
User control over visual properties is set with statements or options within the procedure.	Properties for text, markers, and lines can be set with global statements such as GOPTIONS, AXIS, LEGEND, PATTERN, SYMBOL, and NOTE.
The plot type is determined by the plot statement only.	For some graphs, the plot type is determined by global options. For example, the INTERPOL= option in the SYMBOL statement might determine whether a graph is a scatter plot or a box plot.
The size, format, and name of output images can be controlled with the HEIGHT=, WIDTH=, OUTPUTFMT=, and IMAGENAME= options in the ODS GRAPHICS statement. The ODS GRAPHICS statement is similar in purpose to the GOPTIONS statement, but it is used with the ODS Graphics procedures only.	The size and format of graphical output is controlled with options such as the HSIZE=, VSIZE=, and DEVICE= options in the GOPTIONS statement.

ODS Graphics Procedures	SAS/GRAPH Procedures
All of the ODS destinations are supported. For the LISTING destination, an image node is created for the graph in the Results tree.	All of the ODS destinations are supported. For the LISTING destination, a GRSEG node is created in the Results tree and the image appears in the graph window.
Some options for the TITLE and FOOTNOTE statements are not supported. See “TITLE and FOOTNOTE Statements” on page 58 .	All options for the TITLE and FOOTNOTE statements are supported.
Only system fonts are supported.	SAS/GRAPH supports SAS/GRAPH fonts (such as SWISSB), device-resident fonts, and system fonts (such as Arial).
Marker symbols can be selected only from a predefined set of named marker symbols. The named marker symbols are different from the named marker symbols in SAS/GRAPH.	Marker symbols can be either created from fonts or selected from a predefined set of named marker symbols.
Area fills can use solid colors only. Transparent fills are supported.	Area fills can use either solid colors or patterns such as crosshatching.
Scaling of fonts and markers is on by default. This means that the sizes of fonts and markers are adjusted as appropriate to the size of your graph. You can disable scaling by using the NOSCALE option in the ODS GRAPHICS statement.	Scaling of fonts and markers is not supported.
RUN-group processing is not supported.	Some procedures support RUN-group processing.
An SG Annotation facility is supported. The ODS Graphics procedures do not support the SAS/GRAPH Annotate facility. You can also use the ODS Graphics Editor to annotate your graphs.	The SAS/GRAPH Annotate facility is supported.

Glossary

axis

a line that represents the midpoints (for a discrete axis) or the scale (for a continuous or interval axis) for graphing variable or data values. An axis typically consists of an axis line with tick marks, tick values (or midpoint values), and a label.

cell

in ODS graphics, a distinct rectangular subregion of a graph that can contain plots, text, or legends. For example, a graph created with the SGPLOT procedure contains only one cell, whereas a graph created with the SGPANEL procedure contains more than one cell.

classification panel

a multi-cell graph in which the cell data is driven by the values of one or more classification variables. The number of the cells is determined by the unique values of the classification variables. Each cell of the panel has the same types of plots.

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.

dynamic variable

a variable defined in a template with the DYNAMIC statement that can be initialized at template run time.

global statement

a SAS statement that you can specify anywhere in a SAS program.

Graph Template Language

an extension to the Output Delivery System (ODS) that enables users to create sophisticated analytical graphs. Short form: GTL.

GTL

See Graph Template Language.

ODS

See Output Delivery System.

ODS Graphics

an extension to ODS that is used to create analytical graphs using the Graph Template Language.

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, other formats, and SAS data sets. Short form: ODS.

panel

a graph with multiple cells.

plot

a visual representation of data such as a scatter plot, a series line, or a histogram. Multiple plots can be overlaid in a cell to create a graph.

SGD file

an ODS Graphics Designer file. Users can open this file in the designer and change the graph. Users can also render the graph to an ODS destination by using the SGDESIGN procedure.

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, a style element might specify the color and font properties of title text or other text in a table or graph.

system font

a font that can be used by any SAS procedure and by other software such as Microsoft Word. These fonts include TrueType and Type1 fonts. Examples of system fonts include Albany AMT, Monotype Sorts, and Arial.

Index

Special Characters

- #BYLINE option
 - TITLE and FOOTNOTE statements 58
- #BYVAL option
 - TITLE and FOOTNOTE statements 58
- #BYVAR option
 - TITLE and FOOTNOTE statements 58

A

- ACROSS= option
 - KEYLEGEND statement (SGPANEL) 158
 - KEYLEGEND statement (SGPLOT) 361
- ALPHA= option
 - DOT statement (SGPANEL) 106
 - DOT statement (SGPLOT) 300
 - ELLIPSE statement (SGPLOT) 307
 - HBAR statement (SGPANEL) 113
 - HBAR statement (SGPLOT) 311
 - HLINE statement (SGPANEL) 149
 - HLINE statement (SGPLOT) 348
 - LOESS statement (SGPANEL) 167
 - LOESS statement (SGPLOT) 373
 - PBSPLINE statement (SGPANEL) 178
 - PBSPLINE statement (SGPLOT) 384
 - REG statement (SGPANEL) 187
 - REG statement (SGPLOT) 394
 - VBAR statement (SGPANEL) 214
 - VBAR statement (SGPLOT) 422
 - VLINE statement (SGPANEL) 244
 - VLINE statement (SGPLOT) 454
- ALTERNATE option
 - COLAXIS statement (SGPANEL) 253
 - ROWAXIS statement (SGPANEL) 261
- annotation
 - about 557
 - ARROW function 565
 - controlling the drawing space 561
 - data sets 558
 - drawing space variables 562
 - examples 597, 599
 - functions 560
 - IMAGE function 570
 - LINE function 573
 - OVAL function 577
 - PAD= option 561
 - POLYCONT function 581
 - POLYGON function 582
 - POLYLINE function 585
 - RECTANGLE function 588
 - SGANNO= option 561
 - specifying in a procedure 561
 - TEXT function 592
 - TEXTCONT function 596
 - variables 559
- appearance of graphs
 - See also* ODS styles
 - procedure options 625
- ARROW function (annotation) 565
- ARROWDIRECTION= option
 - VECTOR statement (SGPANEL) 239
 - VECTOR statement (SGPLOT) 449
- ARROWHEADSHAPE= option
 - VECTOR statement (SGPANEL) 239
 - VECTOR statement (SGPLOT) 449
- aspect ratio 653
- attribute maps
 - about 605
 - ATTRID= option 610
 - data sets 606
 - DATTRMAP= option 610
 - examples 611, 612, 614
 - reserved variables 608
 - specifying in a procedure 610
- ATTRID= option 610
- axes
 - discrete 82, 278
 - linear 82, 278
 - logarithmic 82, 278

- paneled scatter plot with shared axes
 - 523, 529
- plot axes 82, 278
- text box inside plot axes 356
- time axes 82, 279
- axis options 251, 259
- axis statements 16
- AXIS= option
 - REFLINE statement (SGPANEL) 184
 - REFLINE statement (SGPLOT) 391
- B**
- B-spline curves
 - See also [penalized B-spline plots](#)
 - fitted penalized (SGPANEL) 176
 - fitted penalized (SGPLOT) 383
- BAND statement
 - about band plots 20
 - example 20, 504
 - SGPANEL procedure 90
 - SGPLOT procedure 284
- bands
 - about band plots 20
 - example 20, 504
 - highlighting plots (SGPANEL) 90
 - highlighting plots (SGPLOT) 284
- bar charts
 - about 42
 - example 42, 78, 270, 276, 509
 - fill patterns 636
 - horizontal (SGPANEL) 111
 - horizontal (SGPLOT) 309
 - parameterized horizontal (SGPANEL) 120
 - parameterized horizontal (SGPLOT) 317
 - parameterized vertical (SGPANEL) 220
 - parameterized vertical (SGPLOT) 429
 - vertical (SGPANEL) 212
 - vertical (SGPLOT) 420
- bar-line charts 451
 - example 509
- BARWIDTH= option
 - HBAR statement (SGPANEL) 113
 - HBAR statement (SGPLOT) 311
 - HBARPARM statement (SGPANEL) 122
 - HBARPARM statement (SGPLOT) 319
 - VBAR statement (SGPANEL) 214
 - VBAR statement (SGPLOT) 423
 - VBARPARM statement (SGPANEL) 222
 - VBARPARM statement (SGPLOT) 431
- WATERFALL statement (SGPLOT) 463
- BASELINE= option
 - NEEDLE statement (SGPANEL) 172
 - NEEDLE statement (SGPLOT) 379
- basic plots
 - about 19
 - compatible plot types (SGPANEL) 81
 - compatible plot types (SGPLOT) 277
- BCOLOR= option
 - TITLE and FOOTNOTE statements 58
- BINSTART= option
 - HISTOGRAM statement (SGPANEL) 146
 - HISTOGRAM statement (SGPLOT) 344
- BINWIDTH= option
 - HISTOGRAM statement (SGPANEL) 146
 - HISTOGRAM statement (SGPLOT) 344
- BOLD option
 - TITLE and FOOTNOTE statements 58
- BORDER option
 - INSET statement (SGPLOT) 357
 - KEYLEGEND statement (SGPANEL) 158
 - KEYLEGEND statement (SGPLOT) 361
 - PANELBY statement (SGPANEL) 88
- borders
 - controlling with ODS GRAPHICS statement 654
- BOUNDARY= option
 - HISTOGRAM statement (SGPANEL) 146
 - HISTOGRAM statement (SGPLOT) 344
- box plots
 - about 37, 334
 - example 37, 78, 276, 508
 - horizontal (SGPANEL) 126
 - horizontal (SGPLOT) 324
 - vertical (SGPANEL) 227
 - vertical (SGPLOT) 436
- BOX= option
 - TITLE and FOOTNOTE statements 58
- BOXWIDTH= option
 - HBOX statement (SGPANEL) 128
 - HBOX statement (SGPLOT) 326
 - VBOX statement (SGPANEL) 229
 - VBOX statement (SGPLOT) 438
- BRADIUSMAX= option
 - BUBBLE statement (SGPANEL) 97

- BUBBLE statement (SGPLOT) 292
- BRADIUSMIN= option
 - BUBBLE statement (SGPANEL) 97
 - BUBBLE statement (SGPLOT) 292
- BREAK option
 - HLINE statement (SGPANEL) 150
 - HLINE statement (SGPLOT) 348
 - SERIES statement (SGPANEL) 199
 - SERIES statement (SGPLOT) 407
 - STEP statement (SGPANEL) 206
 - STEP statement (SGPLOT) 415
 - VLINE statement (SGPANEL) 244
 - VLINE statement (SGPLOT) 454
- BSPACE= option
 - TITLE and FOOTNOTE statements 58
- bubble plots
 - about bubble plots 21
 - example 21
 - SGPANEL procedure 96
 - SGPLOT procedure 290
- BUBBLE statement
 - about bubble plots 21
 - example 21
 - SGPANEL procedure 96
 - SGPLOT procedure 290
- BY groups
 - creating graphs for each BY group 52
- BY lines 53
 - substituting values in a text string 62
- BY statement 52
 - with SGPLOT procedure 53
 - with TITLE and FOOTNOTE statements 53
- BY-group processing
 - data preparation for 53
- C**
- CAPSHAPE= option
 - HBOX statement (SGPANEL) 128
 - HBOX statement (SGPLOT) 326
 - VBOX statement (SGPANEL) 229
 - VBOX statement (SGPLOT) 438
- categorization plots
 - about 42
 - compatible plot types (SGPANEL) 81
 - compatible plot types (SGPLOT) 277
- CATEGORY= option
 - HBOX statement (SGPANEL) 129
 - HBOX statement (SGPLOT) 326
 - VBOX statement (SGPANEL) 229
 - VBOX statement (SGPLOT) 439
- CATEGORYORDER option
 - DOT statement (SGPLOT) 301
- CATEGORYORDER= option
 - DOT statement (SGPANEL) 106
- HBAR statement (SGPANEL) 114
- HBAR statement (SGPLOT) 311
- HLINE statement (SGPANEL) 150
- HLINE statement (SGPLOT) 348
- VBAR statement (SGPANEL) 214
- VBAR statement (SGPLOT) 423
- VLINE statement (SGPANEL) 244
- VLINE statement (SGPLOT) 454
- cell 5
 - creating multi-cell graphs 7
 - creating paneled scatter plots 8
 - creating single-cell graphs 6
- charts
 - example, bar charts 270
 - example, bar-line charts 509
 - example, high-low charts 510
 - example, line charts 272
 - example, vertical bar charts 78
 - high-low charts 137, 335
 - horizontal bar charts (SGPANEL) 111
 - horizontal bar charts (SGPLOT) 309
 - horizontal line (SGPANEL) 148
 - horizontal line (SGPLOT) 346
 - parameterized bar charts (SGPANEL) 120
 - parameterized bar charts (SGPLOT) 317, 429
 - parameterized bar charts (SGPANEL) 220
 - vertical bar charts (SGPANEL) 212
 - vertical bar charts (SGPLOT) 420
 - vertical line (SGPANEL) 242
 - vertical line (SGPLOT) 451
 - waterfall charts 462
- classification variables
 - for panels 87
- CLI option
 - PBSPLINE statement (SGPANEL) 178
 - PBSPLINE statement (SGPLOT) 385
 - REG statement (SGPANEL) 187
 - REG statement (SGPLOT) 394
- CLIATTRS= option
 - PBSPLINE statement (SGPANEL) 178
 - PBSPLINE statement (SGPLOT) 385
 - REG statement (SGPANEL) 187
 - REG statement (SGPLOT) 394
- CLIP option
 - ELLIPSE statement (SGPLOT) 307
- CLIP= option
 - LINEPARM statement (SGPANEL) 162
 - LINEPARM statement (SGPLOT) 368
- CLM option
 - LOESS statement (SGPANEL) 167
 - LOESS statement (SGPLOT) 373
 - PBSPLINE statement (SGPANEL) 179

- PBSPLINE statement (SGPLOT) 386
- REG statement (SGPANEL) 188
- REG statement (SGPLOT) 395
- CLMATTRS= option
 - LOESS statement (SGPANEL) 167
 - LOESS statement (SGPLOT) 373
 - PBSPLINE statement (SGPANEL) 179
 - PBSPLINE statement (SGPLOT) 386
 - REG statement (SGPANEL) 188
 - REG statement (SGPLOT) 395
- CLMTRANSPARENCY= option
 - LOESS statement (SGPANEL) 168
 - LOESS statement (SGPLOT) 374
 - PBSPLINE statement (SGPANEL) 180
 - PBSPLINE statement (SGPLOT) 387
 - REG statement (SGPANEL) 189
 - REG statement (SGPLOT) 396
- clustering
 - example 499
- CLUSTERWIDTH= option
 - DOT statement (SGPANEL) 106
 - DOT statement (SGPLOT) 301
 - HBAR statement (SGPANEL) 114
 - HBAR statement (SGPLOT) 312
 - HBARPARM statement (SGPANEL) 122
 - HBARPARM statement (SGPLOT) 319
 - HBOX statement (SGPANEL) 129
 - HBOX statement (SGPLOT) 326
 - HLINE statement (SGPANEL) 150, 349
 - NEEDLE statement (SGPANEL) 172
 - NEEDLE statement (SGPLOT) 379
 - SCATTER statement (SGPANEL) 193
 - SCATTER statement (SGPLOT) 401
 - SERIES statement (SGPANEL) 200
 - SERIES statement (SGPLOT) 408
 - STEP statement (SGPANEL) 206
 - STEP statement (SGPLOT) 415
 - VBAR statement (SGPANEL) 215
 - VBAR statement (SGPLOT) 423
 - VBARPARM statement (SGPANEL) 222
 - VBARPARM statement (SGPLOT) 431
 - VBOX statement (SGPANEL) 229
 - VBOX statement (SGPLOT) 439
 - VLINE statement (SGPANEL) 244
 - VLINE statement (SGPLOT) 454
- COLAXIS statement
 - SGPANEL procedure 251
- COLHEADERPOS= option
 - PANELBY statement (SGPANEL) 88
- COLOR= option
 - TITLE and FOOTNOTE statements 58
- VBAR statement (SGPLOT) 426
- COLOR= suboption
 - BAND statement (SGPANEL) 94
 - BAND statement (SGPLOT) 288
 - ELLIPSE statement (SGPLOT) 307
 - HBAR statement (SGPANEL) 116
 - HBAR statement (SGPLOT) 314
 - HISTOGRAM statement (SGPANEL) 146
 - HISTOGRAM statement (SGPLOT) 344
 - VBAR statement (SGPANEL) 217
- COLORGROUP= option
 - WATERFALL statement (SGPLOT) 464
- COLUMNS= option
 - PANELBY statement (SGPANEL) 88
 - PLOT statement (SGSCATTER) 541
- combining plots
 - example 500
- comparative scatter plots
 - with grouped data 551
 - with regression fits and confidence ellipses 552
 - with shared axes 529
- COMPARE statement
 - SGSCATTER procedure 526, 529
- compatible plot types
 - SGPANEL 81
 - SGPLOT 277
- components of a graph 5
- confidence bands
 - example, adding to regression plots 502
- confidence ellipse 306, 552
 - about ellipse plots 33
- CONNECT= option
 - HBOX statement (SGPANEL) 129
 - HBOX statement (SGPLOT) 326
 - VBOX statement (SGPANEL) 229
 - VBOX statement (SGPLOT) 439
- CONNECTATTRS= option
 - HBOX statement (SGPANEL) 129
 - HBOX statement (SGPLOT) 327
 - VBOX statement (SGPANEL) 229
 - VBOX statement (SGPLOT) 439
- CONTENT option
 - PROC SGDESIGN statement 71
- CURVELABEL= option
 - HLINE statement (SGPANEL) 150
 - HLINE statement (SGPLOT) 349
 - LINEPARM statement (SGPANEL) 163
 - LINEPARM statement (SGPLOT) 368
 - LOESS statement (SGPANEL) 168
 - LOESS statement (SGPLOT) 374
 - PBSPLINE statement (SGPANEL) 180

- PBSPLINE statement (SGPLOT) 387
 - REG statement (SGPANEL) 189
 - REG statement (SGPLOT) 397
 - SERIES statement (SGPANEL) 200
 - SERIES statement (SGPLOT) 408
 - STEP statement (SGPANEL) 206
 - STEP statement (SGPLOT) 415
 - VLINE statement (SGPANEL) 245
 - VLINE statement (SGPLOT) 455
 - CURVELABELATTRS= option
 - BAND statement (SGPANEL) 92
 - BAND statement (SGPLOT) 286
 - HLINE statement (SGPANEL) 151
 - HLINE statement (SGPLOT) 349
 - LINEPARM statement (SGPANEL) 163
 - LINEPARM statement (SGPLOT) 368
 - SERIES statement (SGPANEL) 200
 - SERIES statement (SGPLOT) 408
 - STEP statement (SGPANEL) 206
 - STEP statement (SGPLOT) 415
 - VLINE statement (SGPANEL) 245
 - VLINE statement (SGPLOT) 455
 - CURVELABELLOC= option
 - BAND statement (SGPLOT) 287
 - HLINE statement (SGPLOT) 350
 - LINEPARM statement (SGPLOT) 369
 - LOESS statement (SGPLOT) 374
 - PBSPLINE statement (SGPLOT) 387
 - REG statement (SGPLOT) 397
 - SERIES statement (SGPLOT) 409
 - STEP statement (SGPLOT) 416
 - VLINE statement (SGPLOT) 456
 - CURVELABELLOWER= option
 - BAND statement (SGPANEL) 93
 - BAND statement (SGPLOT) 287
 - CURVELABELPOS= option
 - BAND statement (SGPANEL) 93
 - BAND statement (SGPLOT) 287
 - HLINE statement (SGPANEL) 151
 - HLINE statement (SGPLOT) 350
 - LINEPARM statement (SGPANEL) 164
 - LINEPARM statement (SGPLOT) 369
 - LOESS statement (SGPANEL) 168
 - LOESS statement (SGPLOT) 375
 - PBSPLINE statement (SGPANEL) 180
 - PBSPLINE statement (SGPLOT) 387
 - REG statement (SGPANEL) 189
 - REG statement (SGPLOT) 397
 - SERIES statement (SGPANEL) 201
 - SERIES statement (SGPLOT) 409
 - STEP statement (SGPANEL) 207
 - STEP statement (SGPLOT) 416
 - VLINE statement (SGPANEL) 246
 - VLINE statement (SGPLOT) 456
 - CURVELABELUPPER= option
 - BAND statement (SGPANEL) 93
 - BAND statement (SGPLOT) 287
 - customizing graphics output 619
 - levels of customization 620
 - CYCLEATTRS option
 - PROC SGPANEL statement 85
 - PROC SGPLOT statement 281
- D**
- data preparation
 - for BY-group processing 53
 - data tips 654
 - DATA= option
 - PROC SGDESIGN statement 71
 - PROC SGPANEL statement 85
 - PROC SGPLOT statement 282
 - PROC SGRENDER statement 514
 - PROC SGSCATTER statement 528
 - DATALABEL= option
 - BUBBLE statement (SGPANEL) 98
 - BUBBLE statement (SGPLOT) 292
 - COMPARE statement (SGSCATTER) 530
 - DOT statement (SGPANEL) 107
 - DOT statement (SGPLOT) 301
 - HBAR statement (SGPANEL) 114
 - HBAR statement (SGPLOT) 312
 - HBARPARM statement (SGPANEL) 122
 - HBARPARM statement (SGPLOT) 320
 - HBOX statement (SGPANEL) 130
 - HBOX statement (SGPLOT) 327
 - HLINE statement (SGPANEL) 152
 - HLINE statement (SGPLOT) 350
 - LOESS statement (SGPANEL) 169
 - LOESS statement (SGPLOT) 375
 - MATRIX statement (SGSCATTER) 538
 - NEEDLE statement (SGPANEL) 172
 - NEEDLE statement (SGPLOT) 379
 - PBSPLINE statement (SGPANEL) 181
 - PBSPLINE statement (SGPLOT) 388
 - PLOT statement (SGSCATTER) 541
 - REG statement (SGPANEL) 190
 - REG statement (SGPLOT) 397
 - SCATTER statement (SGPANEL) 193
 - SCATTER statement (SGPLOT) 401
 - SERIES statement (SGPANEL) 201
 - SERIES statement (SGPLOT) 409
 - STEP statement (SGPANEL) 208
 - STEP statement (SGPLOT) 416
 - VBAR statement (SGPANEL) 215
 - VBAR statement (SGPLOT) 424

- VBARPARAM statement (SGPANEL) 222
- VBARPARAM statement (SGPLOT) 432
- VBOX statement (SGPANEL) 230
- VBOX statement (SGPLOT) 440
- VECTOR statement (SGPANEL) 239
- VECTOR statement (SGPLOT) 449
- VLINE statement (SGPANEL) 246
- VLINE statement (SGPLOT) 456
- WATERFALL statement (SGPLOT) 464
- DATALABELATTRS= option
 - BUBBLE statement (SGPANEL) 98
 - BUBBLE statement (SGPLOT) 292
 - DOT statement (SGPANEL) 107
 - DOT statement (SGPLOT) 301
 - HBAR statement (SGPANEL) 114
 - HBAR statement (SGPLOT) 312
 - HBARPARAM statement (SGPANEL) 122
 - HBARPARAM statement (SGPLOT) 320
 - HBOX statement (SGPANEL) 130
 - HBOX statement (SGPLOT) 327
 - HLINE statement (SGPANEL) 152
 - HLINE statement (SGPLOT) 350
 - NEEDLE statement (SGPANEL) 173
 - NEEDLE statement (SGPLOT) 379
 - SCATTER statement (SGPANEL) 194
 - SCATTER statement (SGPLOT) 401
 - SERIES statement (SGPANEL) 201
 - SERIES statement (SGPLOT) 409
 - STEP statement (SGPANEL) 208
 - STEP statement (SGPLOT) 416
 - VBAR statement (SGPANEL) 215
 - VBAR statement (SGPLOT) 424
 - VBARPARAM statement (SGPANEL) 223
 - VBARPARAM statement (SGPLOT) 432
 - VBOX statement (SGPLOT) 440
 - VECTOR statement (SGPANEL) 239
 - VECTOR statement (SGPLOT) 449
 - VLINE statement (SGPANEL) 246
 - VLINE statement (SGPLOT) 456
 - WATERFALL statement (SGPLOT) 464
- DATALABELPOS= option
 - VBAR statement (SGPLOT) 424
 - VBARPARAM statement (SGPLOT) 433
 - VLINE statement (SGPLOT) 457
- DATAPERCENT (annotation space) 562
- DATAPIXEL (annotation space) 562
- DATASKIN= option
 - HBAR statement (SGPANEL) 115
 - HBAR statement (SGPLOT) 313
 - HBARPARAM statement (SGPANEL) 123
 - HBARPARAM statement (SGPLOT) 321, 322
 - VBAR statement (SGPANEL) 216
 - VBAR statement (SGPLOT) 425
 - VBARPARAM statement (SGPANEL) 223
 - VBARPARAM statement (SGPLOT) 433
 - WATERFALL statement (SGPLOT) 465
- DATAVALUE (annotation space) 562
- DATTRMAP= option 610
- DEFINE STYLE statement
 - TEMPLATE procedure 635
- DEGREE= option
 - LOESS statement (SGPANEL) 169
 - LOESS statement (SGPLOT) 375
 - PBSPLINE statement (SGPANEL) 181
 - PBSPLINE statement (SGPLOT) 388
 - REG statement (SGPANEL) 190
 - REG statement (SGPLOT) 397
- density curves
 - See [density plots](#)
- density plots
 - about 39
 - example 7, 39, 78, 507
 - kernel density function 104
 - kernel density weight functions 104
 - normal density function 103
 - panel of graph cells 267
 - scatter plot matrix with 523
 - SGPANEL procedure 101
 - SGPLOT procedure 295
- DENSITY statement
 - about density plots 39
 - example 7, 39, 78, 267, 507
 - SGPANEL procedure 101
 - SGPLOT procedure 295
- DESCENDING option
 - BY statement 52
- DESCRIPTION= option
 - PROC SGDESIGN statement 72
 - PROC SGPANEL statement 86
 - PROC SGPLOT statement 282
 - PROC SGSCATTER statement 528
- descriptive labels
 - associating with variables 57
- destinations, ODS
 - See [ODS destinations](#)
- DIAGONAL= option
 - MATRIX statement (SGSCATTER) 538

- discrete axes 82, 278
- DISCRETEOFFSET= option
 - BAND statement (SGPANEL) 93
 - BAND statement (SGPLOT) 287
 - DOT statement (SGPANEL) 108
 - DOT statement (SGPLOT) 302
 - HBAR statement (SGPANEL) 116
 - HBAR statement (SGPLOT) 314
 - HBARPARAM statement (SGPANEL) 124
 - HBARPARAM statement (SGPLOT) 321
 - HBOX statement (SGPANEL) 131
 - HBOX statement (SGPLOT) 328
 - HLINE statement (SGPANEL) 153
 - HLINE statement (SGPLOT) 351
 - NEEDLE statement (SGPANEL) 173
 - NEEDLE statement (SGPLOT) 380
 - REFLINE statement (SGPANEL) 184
 - REFLINE statement (SGPLOT) 391
 - SCATTER statement (SGPANEL) 194
 - SCATTER statement (SGPLOT) 402
 - SERIES statement (SGPANEL) 202
 - SERIES statement (SGPLOT) 410
 - STEP statement (SGPANEL) 209
 - STEP statement (SGPLOT) 417
 - VBAR statement (SGPANEL) 216
 - VBAR statement (SGPLOT) 426
 - VBARPARAM statement (SGPANEL) 224
 - VBARPARAM statement (SGPLOT) 434
 - VBOX statement (SGPANEL) 231
 - VBOX statement (SGPLOT) 441
 - VLINE statement (SGPANEL) 247
 - VLINE statement (SGPLOT) 457
- DISCRETEORDER= option
 - COLAXIS statement (SGPANEL) 253
 - ROWAXIS statement (SGPANEL) 261
 - X2AXIS statement (SGPLOT) 477
 - XAXIS statement (SGPLOT) 469
 - Y2AXIS statement (SGPLOT) 492
 - YAXIS statement (SGPLOT) 484
- DISPLAY= option
 - X2AXIS statement (SGPLOT) 477
 - XAXIS statement (SGPLOT) 469
 - Y2AXIS statement (SGPLOT) 492
 - YAXIS statement (SGPLOT) 485
- distribution plots
 - about 37
 - compatible plot types (SGPANEL) 81
 - compatible plot types (SGPLOT) 277
- documents
 - embedding images in 653
- dot plots
 - about 46
 - example 46, 506
 - SGPANEL procedure 104
 - SGPLOT procedure 299
- DOT statement
 - about dot plots 46
 - example 46, 506
 - SGPANEL procedure 104
 - SGPLOT procedure 299
- DOWN= option
 - KEYLEGEND statement (SGPANEL) 158
 - KEYLEGEND statement (SGPLOT) 361
- DPI scaling 653
- drill down 654
- DYNAMIC statement
 - SGDESIGN procedure 72
 - SGDESIGN procedure example 74
 - SGRENDER procedure 515
- dynamic variables
 - in statgraph template code 515
 - setting values in statgraph templates 518
 - SGDESIGN procedure 72
 - SGDESIGN procedure concepts 69
 - SGDESIGN procedure example 74
- E**
- ELLIPSE option
 - COMPARE statement (SGSCATTER) 531
 - MATRIX statement (SGSCATTER) 538
 - PLOT statement (SGSCATTER) 542
- ellipse plots
 - about 33
 - example 33, 276
 - example, adding to scatter plots 503
- ELLIPSE statement
 - about ellipse plots 33
 - example 33, 276, 503
 - SGPLOT procedure 306
- embedding images in documents 653
- ERRORBARATTRS= option
 - SCATTER statement (SGPANEL) 195
 - SCATTER statement (SGPLOT) 402
 - STEP statement (SGPANEL) 209
 - STEP statement (SGPLOT) 417
- EXTREME option
 - HBOX statement (SGPANEL) 131
 - HBOX statement (SGPLOT) 328
 - VBOX statement (SGPANEL) 231
 - VBOX statement (SGPLOT) 441

F

FILL option

BAND statement (SGPANEL) 93
 BAND statement (SGPLOT) 287
 BUBBLE statement (SGPANEL) 99
 BUBBLE statement (SGPLOT) 293
 ELLIPSE statement (SGPLOT) 307
 HBAR statement (SGPANEL) 116
 HBAR statement (SGPLOT) 314
 HBARPARAM statement (SGPANEL) 124
 HBARPARAM statement (SGPLOT) 321
 HBOX statement (SGPANEL) 131
 HBOX statement (SGPLOT) 329
 HISTOGRAM statement (SGPANEL) 146
 HISTOGRAM statement (SGPLOT) 344
 VBAR statement (SGPANEL) 217
 VBAR statement (SGPLOT) 426
 VBARPARAM statement (SGPANEL) 224
 VBARPARAM statement (SGPLOT) 434
 VBOX statement (SGPANEL) 231
 VBOX statement (SGPLOT) 441
 WATERFALL statement (SGPLOT) 465

fill patterns 636

FILLATTRS= option

BAND statement (SGPANEL) 94
 BAND statement (SGPLOT) 288
 BUBBLE statement (SGPANEL) 99
 BUBBLE statement (SGPLOT) 293
 ELLIPSE statement (SGPLOT) 307
 HBAR statement (SGPANEL) 116
 HBAR statement (SGPLOT) 314
 HBARPARAM statement (SGPANEL) 124
 HBOX statement (SGPANEL) 131
 HBOX statement (SGPLOT) 329
 HISTOGRAM statement (SGPANEL) 146
 HISTOGRAM statement (SGPLOT) 344
 VBAR statement (SGPANEL) 217
 VBAR statement (SGPLOT) 426
 VBARPARAM statement (SGPANEL) 224
 VBARPARAM statement (SGPLOT) 434
 VBOX statement (SGPANEL) 231
 VBOX statement (SGPLOT) 441
 WATERFALL statement (SGPLOT) 465

FINALBARATTRS= option

WATERFALL statement (SGPLOT) 466

FINALBARTICKVALUE= option

WATERFALL statement (SGPLOT) 466

fit and confidence plots

about 32
 compatible plot types (SGPANEL) 81
 compatible plot types (SGPLOT) 277

FITPOLICY= option

COLAXIS statement (SGPANEL) 253
 X2AXIS statement (SGPLOT) 477
 XAXIS statement (SGPLOT) 469

fitted loess plots

about 34
 SGPANEL procedure 165
 SGPLOT procedure 371

fitted penalized B-spline curves

See also [penalized B-spline plots](#)
 SGPANEL procedure 176
 SGPLOT procedure 383

fitted regression curves

See also [regression plots](#)
 SGPANEL procedure 185
 SGPLOT procedure 392

FONT= option

TITLE and FOOTNOTE statements 58

FOOTNOTE statement 61

BY statement with 53
 measurement units 63
 substituting BY line values in a text string 62

footnotes 61

FORMAT statement 54

formats

associating with variables 54

FREQ= option

DENSITY statement (SGPANEL) 102
 DENSITY statement (SGPLOT) 296
 DOT statement (SGPANEL) 108
 DOT statement (SGPLOT) 302
 ELLIPSE statement (SGPLOT) 308
 HBAR statement (SGPANEL) 116
 HBAR statement (SGPLOT) 314
 HBOX statement (SGPANEL) 131
 HBOX statement (SGPLOT) 329
 HISTOGRAM statement (SGPANEL) 147
 HISTOGRAM statement (SGPLOT) 345
 HLINE statement (SGPANEL) 153
 HLINE statement (SGPLOT) 351
 PBSPLINE statement (SGPANEL) 181
 PBSPLINE statement (SGPLOT) 388
 REG statement (SGPANEL) 190

- REG statement (SGPLOT) 397
 - SCATTER statement (SGPANEL) 195
 - SCATTER statement (SGPLOT) 403
 - VBAR statement (SGPANEL) 217
 - VBAR statement (SGPLOT) 426
 - VBOX statement (SGPANEL) 232
 - VBOX statement (SGPLOT) 441
 - VLINE statement (SGPANEL) 247
 - VLINE statement (SGPLOT) 458
- G**
- global statements 16
 - global statements, overview 51
 - graph appearance
 - See also* ODS styles
 - procedure options 625
 - graph components 5
 - Graph Template Language (GTL) 3, 513, 665
 - See also* SGRENDER procedure
 - creating statgraph templates 517
 - modifying styles 640
 - SGPLOT (TMPLOUT= option) 283
 - GRAPHICS 651
 - graphics output
 - customizing 619
 - from templates 513
 - settings for 58
 - GRAPHPERCENT (annotation space) 562
 - GRAPHPIXEL (annotation space) 562
 - graphs
 - accessing as separate image files 654
 - creating, for each BY group 52
 - output size 653
 - scaling for DPI 653
 - scaling for user-specified image size 653
 - size of 652
 - GRID option
 - COLAXIS statement (SGPANEL) 254
 - COMPARE statement (SGSCATTER) 531
 - PLOT statement (SGSCATTER) 542
 - ROWAXIS statement (SGPANEL) 261
 - X2AXIS statement (SGPLOT) 478
 - XAXIS statement (SGPLOT) 470
 - Y2AXIS statement (SGPLOT) 492
 - YAXIS statement (SGPLOT) 485
 - GROUP= option
 - BAND statement (SGPANEL) 94
 - BAND statement (SGPLOT) 288
 - BUBBLE statement (SGPANEL) 99
 - BUBBLE statement (SGPLOT) 293
 - COMPARE statement (SGSCATTER) 531
 - DOT statement (SGPANEL) 108
 - DOT statement (SGPLOT) 303
 - HBAR statement (SGPANEL) 117
 - HBAR statement (SGPLOT) 314
 - HBOX statement (SGPANEL) 131
 - HBOX statement (SGPLOT) 329
 - HLINE statement (SGPANEL) 153
 - HLINE statement (SGPLOT) 352
 - LINEPARM statement (SGPANEL) 164
 - LINEPARM statement (SGPLOT) 370
 - LOESS statement (SGPANEL) 169
 - LOESS statement (SGPLOT) 375
 - MATRIX statement (SGSCATTER) 538
 - NEEDLE statement (SGPANEL) 174
 - PBSPLINE statement (SGPANEL) 181
 - PBSPLINE statement (SGPLOT) 388
 - PLOT statement (SGSCATTER) 542
 - REG statement (SGPANEL) 190
 - REG statement (SGPLOT) 397
 - SCATTER statement (SGPANEL) 195
 - SERIES statement (SGPANEL) 202
 - STEP statement (SGPANEL) 209
 - STEP statement (SGPLOT) 418
 - VBAR statement (SGPANEL) 217
 - VBARPARM statement (SGPANEL) 225
 - VBARPARM statement (SGPLOT) 434
 - VECTOR statement (SGPANEL) 240
 - VECTOR statement (SGPLOT) 450
 - VLINE statement (SGPANEL) 247
 - VLINE statement (SGPLOT) 458
 - GROUPDISPLAY= option
 - DOT statement (SGPANEL) 108
 - DOT statement (SGPLOT) 303
 - HBAR statement (SGPANEL) 117
 - HBAR statement (SGPLOT) 314
 - HBOX statement (SGPANEL) 131
 - HBOX statement (SGPLOT) 329
 - HLINE statement (SGPANEL) 153
 - HLINE statement (SGPLOT) 352
 - NEEDLE statement (SGPANEL) 174
 - NEEDLE statement (SGPLOT) 380
 - SCATTER statement (SGPANEL) 195
 - SCATTER statement (SGPLOT) 403
 - SERIES statement (SGPANEL) 202
 - SERIES statement (SGPLOT) 410
 - STEP statement (SGPANEL) 209
 - STEP statement (SGPLOT) 418

- VBAR statement (SGPANEL) 217
 - VBAR statement (SGPLOT) 426
 - VBOX statement (SGPANEL) 232
 - VBOX statement (SGPLOT) 441
 - VLINE statement (SGPANEL) 247
 - VLINE statement (SGPLOT) 458
 - grouping
 - example 498
 - example, SGSCATTER 548, 551
 - GROUPORDER= option
 - DOT statement (SGPANEL) 108
 - DOT statement (SGPLOT) 303
 - HBAR statement (SGPANEL) 117
 - HBAR statement (SGPLOT) 314
 - HBARPARAM statement (SGPANEL) 124
 - HBARPARAM statement (SGPLOT) 322
 - HBOX statement (SGPANEL) 132
 - HBOX statement (SGPLOT) 329
 - HLINE statement (SGPANEL) 153
 - HLINE statement (SGPLOT) 352
 - NEEDLE statement (SGPANEL) 174
 - NEEDLE statement (SGPLOT) 381
 - SCATTER statement (SGPANEL) 195
 - SCATTER statement (SGPLOT) 403
 - SERIES statement (SGPANEL) 202
 - SERIES statement (SGPLOT) 411
 - STEP statement (SGPANEL) 209
 - STEP statement (SGPLOT) 418
 - VBAR statement (SGPANEL) 217
 - VBAR statement (SGPLOT) 426
 - VBARPARAM statement (SGPANEL) 225
 - VBARPARAM statement (SGPLOT) 434
 - VBOX statement (SGPANEL) 232
 - VBOX statement (SGPLOT) 442
 - VLINE statement (SGPANEL) 247
 - VLINE statement (SGPLOT) 458
 - GTL
 - See [Graph Template Language \(GTL\)](#)
- H**
- hardcoded values 627
 - HBAR statement
 - about bar charts 42
 - example 42, 270
 - SGPANEL procedure 111
 - SGPLOT procedure 309
 - HBARPARAM statement
 - about bar charts 42
 - example 42
 - SGPANEL procedure 120
 - SGPLOT procedure 317
 - HBOX statement
 - about box plots 37
 - example 37, 78, 276, 508
 - SGPANEL procedure 126
 - SGPLOT procedure 324, 334
 - HEIGHT= option
 - TITLE and FOOTNOTE statements 58
 - high-low charts
 - about high-low charts 22
 - example 22, 510
 - SGPANEL procedure 137
 - SGPLOT procedure 335
 - highlighting plots 90, 284
 - example 504
 - HIGHLOW statement
 - about high-low charts 22
 - example 22, 510
 - SGPANEL procedure 137
 - SGPLOT procedure 335
 - HISTOGRAM statement
 - about histograms 40
 - example 40, 78, 267, 276, 507
 - SGPANEL procedure 145
 - SGPLOT procedure 343
 - histograms
 - about 40
 - example 40, 78, 267, 276, 507
 - scatter plot matrix with 523
 - SGPANEL procedure 145
 - SGPLOT procedure 343
 - HLINE statement
 - about line charts 47
 - example 47
 - SGPANEL procedure 148
 - SGPLOT procedure 346
 - horizontal bar charts
 - about bar charts 42
 - example 42, 270
 - parameterized (SGPANEL) 120
 - parameterized (SGPLOT) 317
 - SGPANEL procedure 111
 - SGPLOT procedure 309
 - horizontal bar-line charts
 - example 509
 - horizontal box plots
 - about 37, 334
 - example 37, 78, 276, 508
 - SGPANEL procedure 126
 - SGPLOT procedure 324
 - horizontal line
 - SGPANEL procedure 161
 - SGPLOT procedure 366
 - horizontal line charts
 - about 47
 - example 47
 - SGPANEL procedure 148

SGPLOT procedure 346

TITLE and FOOTNOTE statements 58

I

image files
 accessing graphs as separate files 654
 IMAGE function (annotation) 570
 image size
 scaling for user-specified 653
 images
 embedding in documents 653
 INITIALBARATTRS= option
 WATERFALL statement (SGPLOT) 466
 INITIALBARTICKVALUE= option
 WATERFALL statement (SGPLOT) 466
 INITIALBARVALUE= option
 WATERFALL statement (SGPLOT) 466
 INSET statement
 about text insets 31
 example 31
 SGPLOT procedure 356
 INTEGER option
 COLAXIS statement (SGPANEL) 254
 ROWAXIS statement (SGPANEL) 261
 X2AXIS statement (SGPLOT) 478
 XAXIS statement (SGPLOT) 470
 Y2AXIS statement (SGPLOT) 493
 YAXIS statement (SGPLOT) 485
 INTERPOLATION= option
 LOESS statement (SGPANEL) 169
 LOESS statement (SGPLOT) 375
 INTERVAL= option
 COLAXIS statement (SGPANEL) 254
 ROWAXIS statement (SGPANEL) 262
 X2AXIS statement (SGPLOT) 478
 XAXIS statement (SGPLOT) 470
 Y2AXIS statement (SGPLOT) 493
 YAXIS statement (SGPLOT) 485
 intra-quartile range (IQR)
 horizontal box plots 137, 335
 vertical box plots 237, 447
 ITALIC option
 TITLE and FOOTNOTE statements 58

J

JOIN option
 COMPARE statement (SGSCATTER) 531
 PLOT statement (SGSCATTER) 542
 JUSTIFY= option
 STEP statement (SGPANEL) 210
 STEP statement (SGPLOT) 418

K

kernel density curves
 See also density plots
 example 7
 kernel density function 104
 kernel density weight functions 104
 KEYLEGEND statement
 example 503, 504, 507
 SGPANEL procedure 157
 SGPLOT procedure 360

L

LABEL option
 REFLINE statement (SGPANEL) 184
 REFLINE statement (SGPLOT) 391
 LABEL= option
 COLAXIS statement (SGPANEL) 254
 ROWAXIS statement (SGPANEL) 262
 X2AXIS statement (SGPLOT) 478
 XAXIS statement (SGPLOT) 470
 Y2AXIS statement (SGPLOT) 493
 YAXIS statement (SGPLOT) 486
 LABELALIGN= option
 INSET statement (SGPLOT) 357
 LABELATTRS= option
 COLAXIS statement (SGPANEL) 254
 ROWAXIS statement (SGPANEL) 262
 VALUEATTRS statement (SGPLOT) 496
 X2AXIS statement (SGPLOT) 478
 XAXIS statement (SGPLOT) 470
 Y2AXIS statement (SGPLOT) 493
 YAXIS statement (SGPLOT) 486
 LABELFAR option
 HBOX statement (SGPANEL) 132
 HBOX statement (SGPLOT) 330
 VBOX statement (SGPANEL) 232
 VBOX statement (SGPLOT) 442
 LABELLOC= option
 REFLINE statement (SGPLOT) 391
 LABELPOS= option
 REFLINE statement (SGPANEL) 184
 REFLINE statement (SGPLOT) 391
 labels
 associating with variables 57
 language elements
 PROC steps 14
 used by programs 13
 lattice layout style 80
 example 78
 layout types, panels 79, 87
 LAYOUT= option

- PANELBY statement (SGPANEL) 88
- LAYOUTPERCENT (annotation space) 562
- LAYOUTPIXEL (annotation space) 562
- legend statements
 - SGPANEL procedure 157
 - SGPLOT procedure 360
- LEGEND= option
 - COMPARE statement (SGSCATTER) 532
 - MATRIX statement (SGSCATTER) 538
 - PLOT statement (SGSCATTER) 543
- LEGENDLABEL= option
 - BAND statement (SGPANEL) 94
 - BAND statement (SGPLOT) 288
 - BUBBLE statement (SGPANEL) 99
 - BUBBLE statement (SGPLOT) 293
 - DENSITY statement (SGPANEL) 102
 - DENSITY statement (SGPLOT) 296
 - DOT statement (SGPANEL) 108
 - DOT statement (SGPLOT) 303
 - ELLIPSE statement (SGPLOT) 308
 - HBAR statement (SGPANEL) 117
 - HBAR statement (SGPLOT) 315
 - HBARPARM statement (SGPANEL) 125
 - HBARPARM statement (SGPLOT) 322
 - HBOX statement (SGPANEL) 132
 - HBOX statement (SGPLOT) 330
 - HISTOGRAM statement (SGPANEL) 147
 - HISTOGRAM statement (SGPLOT) 345
 - HLINE statement (SGPANEL) 154
 - HLINE statement (SGPLOT) 352
 - LINEPARM statement (SGPANEL) 164
 - LINEPARM statement (SGPLOT) 370
 - LOESS statement (SGPANEL) 169
 - LOESS statement (SGPLOT) 375
 - NEEDLE statement (SGPANEL) 174
 - NEEDLE statement (SGPLOT) 381
 - PBSPLINE statement (SGPANEL) 181
 - PBSPLINE statement (SGPLOT) 388
 - REFLINE statement (SGPANEL) 184
 - REFLINE statement (SGPLOT) 391
 - REG statement (SGPANEL) 190
 - REG statement (SGPLOT) 398
 - SCATTER statement (SGPANEL) 196
 - SCATTER statement (SGPLOT) 403
 - SERIES statement (SGPANEL) 203
 - SERIES statement (SGPLOT) 411
 - STEP statement (SGPANEL) 210
 - STEP statement (SGPLOT) 418
 - VBAR statement (SGPANEL) 218
 - VBAR statement (SGPLOT) 427
 - VBARPARM statement (SGPANEL) 225
 - VBARPARM statement (SGPLOT) 434
 - VBOX statement (SGPANEL) 232
 - VBOX statement (SGPLOT) 442
 - VECTOR statement (SGPANEL) 240
 - VECTOR statement (SGPLOT) 450
 - VLINE statement (SGPANEL) 248
 - VLINE statement (SGPLOT) 458
- legends
 - about 15
 - example 503, 504, 507
 - SGPANEL procedure 82, 157
 - SGPLOT procedure 279, 360
 - SGSCATTER procedure 527
- LIBNAME= option
 - PROC SGDESIGN statement 72
- LIMITATTRS= option
 - DOT statement (SGPANEL) 109
 - DOT statement (SGPLOT) 303
 - HBAR statement (SGPANEL) 117
 - HBAR statement (SGPLOT) 315
 - HBARPARM statement (SGPANEL) 125
 - HBARPARM statement (SGPLOT) 322
 - HLINE statement (SGPANEL) 154
 - HLINE statement (SGPLOT) 352
 - VBAR statement (SGPANEL) 218
 - VBAR statement (SGPLOT) 427
 - VBARPARM statement (SGPANEL) 225
 - VBARPARM statement (SGPLOT) 435
 - VLINE statement (SGPANEL) 248
 - VLINE statement (SGPLOT) 458
- LIMITLOWER= option
 - HBARPARM statement (SGPANEL) 125
 - HBARPARM statement (SGPLOT) 323
 - VBARPARM statement (SGPANEL) 225
 - VBARPARM statement (SGPLOT) 435
- LIMITS= option
 - DOT statement (SGPANEL) 109
 - DOT statement (SGPLOT) 304
 - HBAR statement (SGPANEL) 118
 - HBAR statement (SGPLOT) 315
 - HLINE statement (SGPANEL) 154
 - HLINE statement (SGPLOT) 353
 - VBAR statement (SGPANEL) 218

- VBAR statement (SGPLOT) 427
- VLINE statement (SGPANEL) 248
- VLINE statement (SGPLOT) 459
- LIMITSTAT= option
 - DOT statement (SGPANEL) 109
 - DOT statement (SGPLOT) 304
 - HBAR statement (SGPANEL) 118
 - HBAR statement (SGPLOT) 316
 - HLINE statement (SGPANEL) 154
 - HLINE statement (SGPLOT) 353
 - VBAR statement (SGPANEL) 219
 - VBAR statement (SGPLOT) 428
 - VLINE statement (SGPANEL) 248
 - VLINE statement (SGPLOT) 459
- LIMITUPPER= option
 - HBARPARAM statement (SGPANEL) 125
 - HBARPARAM statement (SGPLOT) 323
 - VBARPARAM statement (SGPANEL) 226
 - VBARPARAM statement (SGPLOT) 435
- line charts
 - about 47
 - example 47, 272, 509
 - horizontal (SGPANEL) 148
 - horizontal (SGPLOT) 346
 - vertical (SGPANEL) 242
 - vertical (SGPLOT) 451
- LINE function (annotation) 573
- line patterns 663
- line plots
 - See [line charts](#)
- linear axes 82, 278
- LINEATTRS= option
 - BAND statement (SGPANEL) 94
 - BAND statement (SGPLOT) 288
 - BUBBLE statement (SGPANEL) 99
 - BUBBLE statement (SGPLOT) 294
 - DENSITY statement (SGPANEL) 102
 - DENSITY statement (SGPLOT) 296
 - ELLIPSE statement (SGPLOT) 308
 - HBOX statement (SGPANEL) 132
 - HBOX statement (SGPLOT) 330
 - HLINE statement (SGPANEL) 155
 - HLINE statement (SGPLOT) 353
 - LINEPANEL statement (SGPANEL) 164
 - LINEPANEL statement (SGPLOT) 370
 - LOESS statement (SGPANEL) 169
 - LOESS statement (SGPLOT) 375
 - NEEDLE statement (SGPANEL) 174
 - NEEDLE statement (SGPLOT) 381
 - PBSPLINE statement (SGPANEL) 181
 - PBSPLINE statement (SGPLOT) 388
 - REFLINE statement (SGPANEL) 184
 - REFLINE statement (SGPLOT) 391
 - REG statement (SGPANEL) 190
 - REG statement (SGPLOT) 398
 - SERIES statement (SGPANEL) 203
 - SERIES statement (SGPLOT) 411
 - STEP statement (SGPANEL) 210
 - STEP statement (SGPLOT) 419
 - VBOX statement (SGPANEL) 232
 - VBOX statement (SGPLOT) 442
 - VECTOR statement (SGPANEL) 240
 - VECTOR statement (SGPLOT) 450
 - VLINE statement (SGPANEL) 249
 - VLINE statement (SGPLOT) 459
- LINEPANEL statement
 - about parameterized lines 24
 - example 24
 - SGPANEL procedure 161
 - SGPLOT procedure 366
- lines
 - about parameterized lines 24
 - about reference lines 23
 - parameterized (SGPANEL) 161
 - parameterized (SGPLOT) 366
 - reference (SGPANEL) 183
 - reference (SGPLOT) 390
- LOCATION= option
 - KEYLEGEND statement (SGPLOT) 361
- loess curves
 - See [loess plots](#)
- LOESS option
 - COMPARE statement (SGSCATTER) 532
 - PLOT statement (SGSCATTER) 544
- loess plots
 - about 34
 - example 34, 78
 - SGPANEL procedure 165
 - SGPLOT procedure 371
- LOESS statement
 - about loess plots 34
 - example 34, 78
 - SGPANEL procedure 165
 - SGPLOT procedure 371
- logarithmic axes 82, 278
- LOGBASE= option
 - COLAXIS statement (SGPANEL) 255
 - ROWAXIS statement (SGPANEL) 263
 - X2AXIS statement (SGPLOT) 479
 - XAXIS statement (SGPLOT) 471
 - Y2AXIS statement (SGPLOT) 494
 - YAXIS statement (SGPLOT) 486
- LOGSTYLE= option
 - COLAXIS statement (SGPANEL) 255
 - ROWAXIS statement (SGPANEL) 263

X2AXIS statement (SGPLOT) 479
 XAXIS statement (SGPLOT) 471
 Y2AXIS statement (SGPLOT) 494
 YAXIS statement (SGPLOT) 487
 LOWER= argument
 BAND statement (SGPANEL) 92
 BAND statement (SGPLOT) 285
 LSPACE= option
 TITLE and FOOTNOTE statements 58

M

marker symbols 661
 MARKERATTRS= option
 COMPARE statement (SGSCATTER) 534
 DOT statement (SGPANEL) 110
 DOT statement (SGPLOT) 304
 HLINE statement (SGPANEL) 155
 HLINE statement (SGPLOT) 354
 LOESS statement (SGPANEL) 170
 LOESS statement (SGPLOT) 376
 MATRIX statement (SGSCATTER) 539
 NEEDLE statement (SGPANEL) 175
 NEEDLE statement (SGPLOT) 381
 PBSPLINE statement (SGPANEL) 182
 PBSPLINE statement (SGPLOT) 389
 PLOT statement (SGSCATTER) 545
 REG statement (SGPANEL) 191
 REG statement (SGPLOT) 398
 SCATTER statement (SGPANEL) 196
 SCATTER statement (SGPLOT) 404
 SERIES statement (SGPANEL) 203
 SERIES statement (SGPLOT) 411
 STEP statement (SGPANEL) 211
 STEP statement (SGPLOT) 419
 VLINE statement (SGPANEL) 249
 VLINE statement (SGPLOT) 460
 MARKERCHAR= option
 SCATTER statement (SGPANEL) 196
 SCATTER statement (SGPLOT) 404
 MARKERCHARATTRS= option
 SCATTER statement (SGPANEL) 196
 SCATTER statement (SGPLOT) 404
 MARKERS option
 HLINE statement (SGPANEL) 156
 HLINE statement (SGPLOT) 354
 NEEDLE statement (SGPANEL) 175
 NEEDLE statement (SGPLOT) 382
 SERIES statement (SGPANEL) 204
 SERIES statement (SGPLOT) 412
 STEP statement (SGPANEL) 211
 STEP statement (SGPLOT) 420
 VLINE statement (SGPANEL) 250
 VLINE statement (SGPLOT) 461

MATRIX statement
 SGSCATTER procedure 526, 537
 MAX= option
 COLAXIS statement (SGPANEL) 256
 ROWAXIS statement (SGPANEL) 264
 X2AXIS statement (SGPLOT) 480
 XAXIS statement (SGPLOT) 472
 Y2AXIS statement (SGPLOT) 495
 YAXIS statement (SGPLOT) 487
 MAXPOINTS= option
 LOESS statement (SGPANEL) 170
 LOESS statement (SGPLOT) 377
 PBSPLINE statement (SGPANEL) 182
 PBSPLINE statement (SGPLOT) 389
 REG statement (SGPANEL) 191
 REG statement (SGPLOT) 399
 MEANATTRS= option
 HBOX statement (SGPANEL) 133
 HBOX statement (SGPLOT) 331
 VBOX statement (SGPANEL) 233
 VBOX statement (SGPLOT) 443
 measurement units 659
 TITLE and FOOTNOTE statements 63
 MEDIANATTRS= option
 HBOX statement (SGPANEL) 134
 HBOX statement (SGPLOT) 331
 VBOX statement (SGPANEL) 234
 VBOX statement (SGPLOT) 443
 MIN= option
 COLAXIS statement (SGPANEL) 256
 ROWAXIS statement (SGPANEL) 264
 X2AXIS statement (SGPLOT) 480
 XAXIS statement (SGPLOT) 472
 Y2AXIS statement (SGPLOT) 495
 YAXIS statement (SGPLOT) 487
 MINOR option
 COLAXIS statement (SGPANEL) 256
 ROWAXIS statement (SGPANEL) 264
 X2AXIS statement (SGPLOT) 480
 XAXIS statement (SGPLOT) 472
 Y2AXIS statement (SGPLOT) 495
 YAXIS statement (SGPLOT) 488
 MISSING option
 DOT statement (SGPANEL) 110
 DOT statement (SGPLOT) 305
 HBAR statement (SGPANEL) 118
 HBAR statement (SGPLOT) 316
 HBARPARM statement (SGPANEL) 126
 HBARPARM statement (SGPLOT) 323
 HBOX statement (SGPANEL) 134
 HBOX statement (SGPLOT) 332
 HLINE statement (SGPANEL) 156
 HLINE statement (SGPLOT) 354
 PANELBY statement (SGPANEL) 88

- VBAR statement (SGPANEL) 219
 - VBAR statement (SGPLOT) 428
 - VBARPARAM statement (SGPANEL) 226
 - VBARPARAM statement (SGPLOT) 435
 - VBOX statement (SGPANEL) 235
 - VBOX statement (SGPLOT) 444
 - VLINE statement (SGPANEL) 250
 - VLINE statement (SGPLOT) 461
 - WATERFALL statement (SGPLOT) 466
 - MODELNAME= option
 - BAND statement (SGPANEL) 95
 - BAND statement (SGPLOT) 289
 - multi-cell graphs 7
 - creating paneled scatter plots 8
- N**
- NAME= option
 - BAND statement (SGPANEL) 95
 - BAND statement (SGPLOT) 289
 - BUBBLE statement (SGPANEL) 100
 - BUBBLE statement (SGPLOT) 294
 - DENSITY statement (SGPANEL) 102
 - DENSITY statement (SGPLOT) 297
 - DOT statement (SGPANEL) 110
 - DOT statement (SGPLOT) 305
 - ELLIPSE statement (SGPLOT) 308
 - HBAR statement (SGPANEL) 119
 - HBAR statement (SGPLOT) 316
 - HBARPARAM statement (SGPANEL) 126
 - HBARPARAM statement (SGPLOT) 323
 - HBOX statement (SGPANEL) 134
 - HBOX statement (SGPLOT) 332
 - HISTOGRAM statement (SGPANEL) 147
 - HISTOGRAM statement (SGPLOT) 345
 - HLINE statement (SGPANEL) 156
 - HLINE statement (SGPLOT) 355
 - LINEPARAM statement (SGPANEL) 165
 - LINEPARAM statement (SGPLOT) 371
 - LOESS statement (SGPANEL) 170
 - LOESS statement (SGPLOT) 377
 - NEEDLE statement (SGPANEL) 175
 - NEEDLE statement (SGPLOT) 382
 - PBSPLINE statement (SGPANEL) 182
 - PBSPLINE statement (SGPLOT) 389
 - REFLINE statement (SGPANEL) 185
 - REFLINE statement (SGPLOT) 392
 - REG statement (SGPANEL) 191
 - REG statement (SGPLOT) 399
 - SCATTER statement (SGPANEL) 197
 - SCATTER statement (SGPLOT) 405
 - SERIES statement (SGPANEL) 203
 - SERIES statement (SGPLOT) 412
 - STEP statement (SGPANEL) 211
 - STEP statement (SGPLOT) 420
 - VBAR statement (SGPANEL) 219
 - VBAR statement (SGPLOT) 428
 - VBARPARAM statement (SGPANEL) 226
 - VBARPARAM statement (SGPLOT) 435
 - VBOX statement (SGPANEL) 235
 - VBOX statement (SGPLOT) 444
 - VECTOR statement (SGPANEL) 241
 - VECTOR statement (SGPLOT) 451
 - VLINE statement (SGPANEL) 250
 - VLINE statement (SGPLOT) 461
 - WATERFALL statement (SGPLOT) 466
 - NBINS= option
 - HISTOGRAM statement (SGPANEL) 147
 - HISTOGRAM statement (SGPLOT) 345
 - needle plots
 - about needle plots 26
 - example 26
 - SGPANEL procedure 171
 - SGPLOT procedure 377
 - NEEDLE statement
 - about needle plots 26
 - example 26
 - SGPANEL procedure 171
 - SGPLOT procedure 377
 - NKNOTS= option
 - PBSPLINE statement (SGPANEL) 182
 - PBSPLINE statement (SGPLOT) 389
 - NOARROWHEADS option
 - VECTOR statement (SGPANEL) 241
 - VECTOR statement (SGPLOT) 451
 - NOAUTOLEGEND option
 - PROC SGPANEL statement 86
 - PROC SGPLOT statement 282
 - NOBORDER option
 - INSET statement (SGPLOT) 357
 - KEYLEGEND statement (SGPANEL) 158
 - KEYLEGEND statement (SGPLOT) 361
 - PANELBY statement (SGPANEL) 88
 - NOCAPS option
 - HBOX statement (SGPANEL) 134
 - HBOX statement (SGPLOT) 332
 - VBOX statement (SGPANEL) 235

- VBOX statement (SGPLOT) 444
- NOCLIP option
 - REFLINE statement (SGPANEL) 185
 - REFLINE statement (SGPLOT) 392
- NOCYCLEATTRS option
 - PROC SGPANEL statement 85
- NOEXTEND option
 - BAND statement (SGPANEL) 95
 - BAND statement (SGPLOT) 289
 - LINEPARM statement (SGPLOT) 371
- NOEXTEND= option
 - LINEPARM statement (SGPANEL) 165
- NOFILL option
 - BAND statement (SGPANEL) 93
 - BAND statement (SGPLOT) 287
 - BUBBLE statement (SGPANEL) 99
 - BUBBLE statement (SGPLOT) 293
 - ELLIPSE statement (SGPLOT) 307
 - HBAR statement (SGPANEL) 116
 - HBAR statement (SGPLOT) 314
 - HBARPARM statement (SGPANEL) 124
 - HBARPARM statement (SGPLOT) 321
 - HBOX statement (SGPANEL) 131
 - HBOX statement (SGPLOT) 329
 - HISTOGRAM statement (SGPANEL) 146
 - HISTOGRAM statement (SGPLOT) 344
 - VBAR statement (SGPANEL) 217
 - VBAR statement (SGPLOT) 426
 - VBARPARM statement (SGPANEL) 224
 - VBARPARM statement (SGPLOT) 434
 - VBOX statement (SGPANEL) 231
 - VBOX statement (SGPLOT) 441
 - WATERFALL statement (SGPLOT) 465
- NOLEGCLI option
 - PBSPLINE statement (SGPANEL) 182
 - PBSPLINE statement (SGPLOT) 389
 - REG statement (SGPANEL) 191
 - REG statement (SGPLOT) 399
- NOLEGCLM option
 - LOESS statement (SGPANEL) 170
 - LOESS statement (SGPLOT) 377
 - PBSPLINE statement (SGPANEL) 182
 - PBSPLINE statement (SGPLOT) 389
 - REG statement (SGPANEL) 191
 - REG statement (SGPLOT) 399
- NOLEGEND option
 - COMPARE statement (SGSCATTER) 534
- MATRIX statement (SGSCATTER) 539
- PLOT statement (SGSCATTER) 545
- NOLEGFIT option
 - LOESS statement (SGPANEL) 170
 - LOESS statement (SGPLOT) 377
 - PBSPLINE statement (SGPANEL) 182
 - PBSPLINE statement (SGPLOT) 390
 - REG statement (SGPANEL) 191
 - REG statement (SGPLOT) 399
- NOMARKERS option
 - LOESS statement (SGPANEL) 170
 - LOESS statement (SGPLOT) 377
 - PBSPLINE statement (SGPANEL) 183
 - PBSPLINE statement (SGPLOT) 390
 - REG statement (SGPANEL) 192
 - REG statement (SGPLOT) 399
- NOMEAN option
 - HBOX statement (SGPANEL) 134
 - HBOX statement (SGPLOT) 332
 - VBOX statement (SGPANEL) 235
 - VBOX statement (SGPLOT) 444
- NOMEDIAN option
 - HBOX statement (SGPANEL) 134
 - HBOX statement (SGPLOT) 332
 - VBOX statement (SGPANEL) 235
 - VBOX statement (SGPLOT) 444
- NOMISSINGGROUP option
 - BAND statement (SGPANEL) 95
 - BAND statement (SGPLOT) 289
 - BUBBLE statement (SGPANEL) 100
 - BUBBLE statement (SGPLOT) 294
 - LINEPARM statement (SGPANEL) 165
 - LINEPARM statement (SGPLOT) 371
 - NEEDLE statement (SGPANEL) 176
 - NEEDLE statement (SGPLOT) 382
 - SCATTER statement (SGPANEL) 197
 - SCATTER statement (SGPLOT) 405
 - SERIES statement (SGPANEL) 204
 - SERIES statement (SGPLOT) 412
 - STEP statement (SGPANEL) 211
 - STEP statement (SGPLOT) 420
 - VECTOR statement (SGPANEL) 241
 - VECTOR statement (SGPLOT) 451
- NOOUTLIERS option
 - HBOX statement (SGPANEL) 135
 - HBOX statement (SGPLOT) 332
 - VBOX statement (SGPANEL) 235
 - VBOX statement (SGPLOT) 444
- NOOUTLINE option
 - BAND statement (SGPANEL) 95
 - BAND statement (SGPLOT) 289
 - ELLIPSE statement (SGPLOT) 308
 - HBAR statement (SGPANEL) 119
 - HBAR statement (SGPLOT) 316

- HISTOGRAM statement (SGPANEL) 147
 - HISTOGRAM statement (SGPLOT) 345
 - VBAR statement (SGPANEL) 219
 - VBAR statement (SGPLOT) 428
 - WATERFALL statement (SGPLOT) 466
 - NOOUTLINE= option
 - BUBBLE statement (SGPANEL) 100
 - BUBBLE statement (SGPLOT) 295
 - HBARPARM statement (SGPANEL) 126
 - HBARPARM statement (SGPLOT) 323
 - VBARPARM statement (SGPANEL) 226
 - VBARPARM statement (SGPLOT) 436
 - normal density curves 78
 - normal density function 103
 - NOSTATLABEL option
 - DOT statement (SGPANEL) 110
 - DOT statement (SGPLOT) 305
 - HBAR statement (SGPANEL) 119
 - HBAR statement (SGPLOT) 316
 - HLINE statement (SGPANEL) 156
 - HLINE statement (SGPLOT) 355
 - VBAR statement (SGPANEL) 219
 - VBAR statement (SGPLOT) 428
 - VLINE statement (SGPANEL) 250
 - VLINE statement (SGPLOT) 461
 - NOTCHES option
 - HBOX statement (SGPANEL) 135
 - HBOX statement (SGPLOT) 332
 - VBOX statement (SGPANEL) 235
 - VBOX statement (SGPLOT) 444
 - NOTIMESPLIT option
 - COLAXIS statement (SGPANEL) 256
 - ROWAXIS statement (SGPANEL) 264
 - X2AXIS statement (SGPLOT) 480
 - XAXIS statement (SGPLOT) 472
 - Y2AXIS statement (SGPLOT) 495
 - YAXIS statement (SGPLOT) 488
 - NOTSORTED option
 - BY statement 52
 - NOVARNAME option
 - PANELBY statement (SGPANEL) 88
 - numeric variables
 - distribution of values (SGPANEL) 101
 - distribution of values (SGPLOT) 295
 - NUMSTD= option
 - DOT statement (SGPANEL) 110
 - DOT statement (SGPLOT) 305
 - HBAR statement (SGPANEL) 119
 - HBAR statement (SGPLOT) 316
 - HLINE statement (SGPANEL) 156
 - HLINE statement (SGPLOT) 355
 - VBAR statement (SGPANEL) 219
 - VBAR statement (SGPLOT) 428
 - VLINE statement (SGPANEL) 250
 - VLINE statement (SGPLOT) 461
- O**
- OBJECT= option
 - PROC SGRENDER statement 514
 - OBJECTLABEL= option
 - PROC SGRENDER statement 515
 - ODS (Output Delivery System)
 - managing graphs with 649
 - ODS graphics procedures and 665
 - ODS destination statements
 - changing current graph style 624
 - STYLE= option 624
 - ODS destinations
 - about 11
 - specifying 649
 - ODS Graphics Editor 665
 - ODS graphics procedures 3, 665
 - appearance option values 625
 - compared with SAS/GRAPH procedures 668
 - controlling graph appearance with 625
 - global statements 51
 - hardcoded values for graph appearance 627
 - ODS and 665
 - style element references 626
 - ODS GRAPHICS statement 58, 651, 665
 - accessing graphs as separate image files 654
 - adding data tips or borders 654
 - graph size 652
 - resetting options 654
 - syntax 651
 - ODS statements 17
 - destinations 11
 - GRAPHICS 17
 - styles 12
 - ODS styles 619, 620
 - about 12
 - appearance option values 625
 - changing current style 624
 - changing default style 625
 - changing default style, SAS Registry 625
 - examples 623
 - hardcoded values 627
 - modifying templates 635
 - ODS graphics procedures options 625
 - recommended 621

- style element references 626
- style elements for ODS statistical graphics 640
- style specification 624
- viewing templates 622
- OFFSETMAX= option
 - COLAXIS statement (SGPANEL) 257
 - ROWAXIS statement (SGPANEL) 264
 - X2AXIS statement (SGPLOT) 480
 - XAXIS statement (SGPLOT) 473
 - Y2AXIS statement (SGPLOT) 495
 - YAXIS statement (SGPLOT) 488
- OFFSETMIN= option
 - COLAXIS statement (SGPANEL) 257
 - ROWAXIS statement (SGPANEL) 265
 - X2AXIS statement (SGPLOT) 481
 - XAXIS statement (SGPLOT) 473
 - Y2AXIS statement (SGPLOT) 496
 - YAXIS statement (SGPLOT) 488
- ONEPANEL option
 - PANELBY statement (SGPANEL) 89
- OUTLIERATTRS= option
 - HBOX statement (SGPANEL) 135
 - HBOX statement (SGPLOT) 332
 - VBOX statement (SGPANEL) 235
 - VBOX statement (SGPLOT) 445
- OUTLINE option
 - BAND statement (SGPANEL) 95
 - BAND statement (SGPLOT) 289
 - ELLIPSE statement (SGPLOT) 308
 - HBAR statement (SGPANEL) 119
 - HBAR statement (SGPLOT) 316
 - HISTOGRAM statement (SGPANEL) 147
 - HISTOGRAM statement (SGPLOT) 345
 - VBAR statement (SGPANEL) 219
 - VBAR statement (SGPLOT) 428
 - WATERFALL statement (SGPLOT) 466
- OUTLINE= option
 - BUBBLE statement (SGPANEL) 100
 - BUBBLE statement (SGPLOT) 295
 - HBARPARM statement (SGPANEL) 126
 - HBARPARM statement (SGPLOT) 323
 - VBARPARM statement (SGPANEL) 226
 - VBARPARM statement (SGPLOT) 436
- output
 - settings for 58
 - size of 653
- OVAL function (annotation) 577
- P**
 - PAD= option 561
 - panel layout style 79
 - PANELBY statement
 - SGPANEL procedure 87
 - panels 78
 - See also SGPANEL procedure
 - classification variables for 87
 - creating 7, 79
 - creating paneled scatter plots 8
 - dot plots 104
 - example 78
 - example, bar charts 270
 - example, density plots 267
 - example, histograms 267
 - example, line charts 272
 - example, regression curves 269
 - histograms 145
 - layout styles 79, 87
 - layouts 78
 - legends 82
 - line charts, horizontal 148
 - line charts, vertical 242
 - parameterized vertical bar charts 220
 - scatter plots 523
 - SGSCATTER procedure 525
 - vertical bar charts 212
 - vertical box plots 227
 - parameterized lines
 - about parameterized lines 24
 - example 24
 - SGPANEL procedure 161
 - SGPLOT procedure 366
 - patterns
 - line patterns 663
 - PBSPLINE option
 - COMPARE statement (SGSCATTER) 534
 - PLOT statement (SGSCATTER) 545
 - PBSPLINE statement
 - about penalized B-spline plots 35
 - example 35
 - SGPANEL procedure 176
 - SGPLOT procedure 383
 - penalized B-spline curves
 - See penalized B-spline plots
 - penalized B-spline plots
 - about 35
 - example 35
 - SGPANEL procedure 176
 - SGPLOT procedure 383
 - PERCENTILE= option
 - HBOX statement (SGPANEL) 135
 - HBOX statement (SGPLOT) 333
 - VBOX statement (SGPANEL) 236
 - VBOX statement (SGPLOT) 445

- plot area 5
- plot axes 82, 251, 259, 278
 - adding text box inside of 356
- PLOT statement
 - SGSCATTER procedure 525, 540, 549
- plot statements
 - about 15
 - SGSCATTER procedure 525, 540, 549
- plot type compatibility
 - SGPANEL 81
 - SGPLOT 277
- plots
 - See also scatter plots
 - See also SGPANEL procedure
 - See also SGPLOT procedure
 - adding confidence or prediction ellipse to 306
 - bubble 96, 290
 - compatible plot types (SGPANEL) 81
 - compatible plot types (SGPLOT) 277
 - dot plots (SGPANEL) 104
 - dot plots (SGPLOT) 299
 - example, adding statistical limits to a dot plot 506
 - example, bar-line charts 509
 - example, combining histograms with density plots 507
 - example, ellipse and scatter plots 503
 - example, histograms and density plot 267
 - example, horizontal box plots 508
 - example, line charts 272
 - example, plotting three series 500
 - example, regression plots 502
 - example, scatter plot with fit and confidence 504
 - examples (SGPANEL) 78
 - examples (SGPLOT) 276
 - fit and confidence plots 81
 - highlighting with bands 90, 284
 - horizontal box plots 334
 - horizontal box plots (SGPANEL) 126
 - horizontal box plots (SGPLOT) 324
 - horizontal line (SGPANEL) 148
 - horizontal line (SGPLOT) 346
 - legends on 157, 360
 - line patterns 663
 - marker symbols 661
 - needle plots 171, 377
 - parameterized lines (SGPANEL) 161
 - parameterized lines (SGPLOT) 366
 - reference lines (SGPANEL) 183
 - reference lines (SGPLOT) 390
 - series plots 198, 406
 - step plots 204, 413
 - style attributes 83, 279
 - vector plots 238, 447
 - vertical box plots (SGPANEL) 227
 - vertical box plots (SGPLOT) 436
 - vertical line (SGPANEL) 242
 - vertical line (SGPLOT) 451
- POLYCONT function (annotation) 581
- POLYGON function (annotation) 582
- POLYLINE function (annotation) 585
- POSITION= option
 - INSET statement (SGPLOT) 357
 - KEYLEGEND statement (SGPANEL) 158
 - KEYLEGEND statement (SGPLOT) 361
- prediction
 - example, adding to regression plots 502
- prediction ellipse 306
 - about ellipse plots 33
 - example, adding to scatter plots 503
 - scatter plot matrix with 523
- PROC SGDESIGN statement 70
- PROC SGPANEL statement 85
- PROC SGPLOT statement 281
- PROC SGRENDER statement 514
- PROC SGSCATTER statement 528
- PROC steps 14
 - axis statements 16
 - legend statements 15
 - PANELBY statements 16
 - plot statements 15
 - procedure statements 15
 - statements required for SGSCATTER 16
- procedure statements 15
- procedures
 - axis statements 16
 - legend statements 15
 - PANELBY statements 16
 - plot statements 15
 - PROC steps 14
 - statements 15
 - statements required for SGSCATTER 16
- programs
 - a typical program 13
 - axis statements 16
 - global statements 16
 - language elements used by 13
 - legend statements 15
 - ODS GRAPHICS statements 17
 - ODS statements 17
 - PANELBY statements 16
 - plot statements 15
 - PROC steps 14
 - procedure statements 15
 - SAS statements 16

statements required for SGSCATTER
16

R

RECTANGLE function (annotation) 588

reference lines

about reference lines 23

example 23

SGPANEL procedure 183

SGPLOT procedure 390

REFLINE statement

about reference lines 23

example 23

SGPANEL procedure 183

SGPLOT procedure 390

REFTICKS option

COMPARE statement (SGSCATTER)
535

PLOT statement (SGSCATTER) 547

X2AXIS statement (SGPLOT) 481

XAXIS statement (SGPLOT) 473

Y2AXIS statement (SGPLOT) 496

YAXIS statement (SGPLOT) 488

REFTICKS= option

COLAXIS statement (SGPANEL) 257

ROWAXIS statement (SGPANEL) 265

REG option

COMPARE statement (SGSCATTER)
535

PLOT statement (SGSCATTER) 547

REG statement

about regression plots 36

example 36, 269, 502

SGPANEL procedure 185

SGPLOT procedure 392

regression curves

See [regression plots](#)

regression fits

comparative scatter plots with 552

regression plots

about 36

example 36

example, adding prediction and
confidence bands 502

example, panel 269

SGPANEL procedure 185

SGPLOT procedure 392

RESPONSE= option

DOT statement (SGPANEL) 110

DOT statement (SGPLOT) 305

HBAR statement (SGPANEL) 119

HBAR statement (SGPLOT) 317

HLINE statement (SGPANEL) 156

HLINE statement (SGPLOT) 355

VBAR statement (SGPANEL) 219

VBAR statement (SGPLOT) 428

VLINE statement (SGPANEL) 250

VLINE statement (SGPLOT) 461

REVERSE option

COLAXIS statement (SGPANEL) 257

ROWAXIS statement (SGPANEL) 265

X2AXIS statement (SGPLOT) 481

XAXIS statement (SGPLOT) 473

Y2AXIS statement (SGPLOT) 496

YAXIS statement (SGPLOT) 488

REWEIGHT= option

LOESS statement (SGPANEL) 170

LOESS statement (SGPLOT) 377

ROWAXIS statement

SGPANEL procedure 259

ROWHEADERPOS= option

PANELBY statement (SGPANEL) 89

ROWS= option

PANELBY statement (SGPANEL) 89

PLOT statement (SGSCATTER) 548

S

sample library 12

SAS statements 16

SAS/GRAPH procedures

compared with ODS graphics

procedures 668

SCALE= option

DENSITY statement (SGPANEL) 102

DENSITY statement (SGPLOT) 297

HISTOGRAM statement (SGPANEL)
147

HISTOGRAM statement (SGPLOT)
345

scaling

for DPI 653

for user-specified image size 653

scatter plot matrix 537

diagonal with histograms and density
plots 523

with grouped data 548

with prediction ellipse 523

scatter plots

See also [SGSCATTER procedure](#)

about scatter plots 27

comparative, with grouped data 551

comparative, with regression fits and
confidence ellipse 552

comparative, with shared axes 529

example 27, 498, 499, 503, 504

paneled 523

paneled, with multiple independent
plots 540

paneled, with shared axes 523

SGPANEL procedure 192

- SGPLOT procedure 399
 - with fitted splines 549
- SCATTER statement
 - about scatter plots 27
 - example 27, 498, 499, 503, 504
 - SGPANEL procedure 192
 - SGPLOT procedure 399
- series plots
 - about series plots 29
 - example 29, 276, 500, 504
 - SGPANEL procedure 198
 - SGPLOT procedure 406
- SERIES statement
 - about series plots 29
 - example 29, 276, 500, 504
 - SGPANEL procedure 198
 - SGPLOT procedure 406
- SG annotation
 - See [annotation](#)
- SG attribute maps
 - See [attribute maps](#)
- SGANNO= option 561
- SGD= argument
 - PROC SGDESIGN statement 71
- SGDESIGN procedure 67
 - about 10
 - concepts 68
 - DYNAMIC statement 72
 - dynamic variables 69
 - examples 73, 74
 - interaction with ODS Graphics Designer 68
 - PROC SGDESIGN statement 70
 - syntax 70
- SGE= option
 - PROC SGRENDER statement 515
- SGPANEL procedure 78
 - about 7
 - automatic differentiation of visual attributes 83
 - BAND statement 90
 - BUBBLE statement 96
 - COLAXIS statement 251
 - compatible plot types 81
 - concepts 79
 - DENSITY statement 101
 - DOT statement 104
 - example, bar charts 270
 - example, histograms and density plot 267
 - example, line charts 272
 - example, regression curves 269
 - HBAR statement 111
 - HBARPARM statement 120
 - HBOX statement 126
 - HIGHLOW statement 137
 - HISTOGRAM statement 145
 - HLINE statement 148
 - KEYLEGEND statement 157
 - layout styles 79
 - layouts 78
 - line patterns 663
 - LINEPARM statement 161
 - LOESS statement 165
 - marker symbols 661
 - measurement units 659
 - NEEDLE statement 171
 - panel creation 79
 - panel legends 82
 - PANELBY statement 87
 - PBSPLINE statement 176
 - plot axes 82
 - PROC SGPANEL statement 85
 - REFLINE statement 183
 - REG statement 185
 - ROWAXIS statement 259
 - SCATTER statement 192
 - SERIES statement 198
 - STEP statement 204
 - syntax 83
 - VBAR statement 212
 - VBARPARM statement 220
 - VBOX statement 227
 - VECTOR statement 238
 - VLINE statement 242
- SGPLOT procedure
 - about 6
 - automatic differentiation of visual attributes 279
 - BAND statement 284
 - BUBBLE statement 290
 - BY statement with 53
 - compatible plot types 277
 - concepts 277
 - DENSITY statement 295
 - DOT statement 299
 - ELLIPSE statement 306
 - example, adding statistical limits to a dot plot 506
 - example, bar-line charts 509
 - example, clustering scatter plots 499
 - example, combining histograms with density plots 507
 - example, ellipse and scatter plots 503
 - example, grouping scatter plots 498
 - example, high-low charts 510
 - example, horizontal box plots 508
 - example, plotting three series 500
 - example, regression plots 502
 - example, scatter plot with fit and confidence 504
 - examples 276

- HBAR statement 309
- HBARPARM statement 317
- HBOX statement 324, 334
- HIGHLOW statement 335
- HISTOGRAM statement 343
- HLINE statement 346
- INSET statement 356
- KEYLEGEND statement 360
- legends 279
- line patterns 663
- LINEPARM statement 366
- LOESS statement 371
- marker symbols 661
- measurement units 659
- NEEDLE statement 377
- PBSPLINE statement 383
- plot axes 278
- PROC SGPLOT statement 281
- REFLINE statement 390
- REG statement 392
- SCATTER statement 399
- SERIES statement 406
- STEP statement 413
- syntax 279
- VBAR statement 420
- VBARPARM statement 429
- VBOX statement 436
- VECTOR statement 447
- visual attributes 279
- VLINE statement 451
- WATERFALL statement 462
- X2AXIS statement 475
- XAXIS statement 467
- Y2AXIS statement 491
- YAXIS statement 483
- SGRENDER procedure 513
 - about 9
 - defining dynamic variables 518
 - DYNAMIC statement 515
 - examples 517
 - graphs from GTL templates 517
 - PROC SGRENDER statement 514
 - syntax 513
- SGSCATTER procedure 523
 - about 8
 - COMPARE statement 526, 529
 - concepts 525
 - examples 548
 - legends 527
 - line patterns 663
 - marker symbols 661
 - MATRIX statement 526, 537
 - measurement units 659
 - PLOT statement 525, 540
 - PROC SGSCATTER statement 528
 - statements for creating panels 525
 - syntax 527
 - SHOWBINS option
 - HISTOGRAM statement (SGPLOT) 345
 - single-cell graphs 6
 - size of graphs 652
 - SMOOTH= option
 - LOESS statement (SGPANEL) 171
 - LOESS statement (SGPLOT) 377
 - PBSPLINE statement (SGPANEL) 183
 - PBSPLINE statement (SGPLOT) 390
 - SPACING= option
 - COMPARE statement (SGSCATTER) 537
 - PANELBY statement (SGPANEL) 89
 - PLOT statement (SGSCATTER) 548
 - SPARSE option
 - PANELBY statement (SGPANEL) 89
 - splines
 - scatter plots with fitted splines 549
 - SPREAD option
 - HBOX statement (SGPANEL) 135
 - HBOX statement (SGPLOT) 333
 - VBOX statement (SGPANEL) 236
 - VBOX statement (SGPLOT) 445
 - START= option
 - MATRIX statement (SGSCATTER) 539
 - PANELBY statement (SGPANEL) 89
 - STAT= option
 - DOT statement (SGPANEL) 111
 - DOT statement (SGPLOT) 305
 - HBAR statement (SGPANEL) 119
 - HBAR statement (SGPLOT) 317
 - HLINE statement (SGPANEL) 156
 - HLINE statement (SGPLOT) 355
 - VBAR statement (SGPANEL) 219
 - VBAR statement (SGPLOT) 429
 - VLINE statement (SGPANEL) 250
 - VLINE statement (SGPLOT) 461
 - WATERFALL statement (SGPLOT) 467
- statements
 - global 16
 - ODS 17
 - ODS GRAPHICS 17
 - SAS 16
- statgraph templates 517
 - creating 517
 - dynamic variable values 515, 518
 - generating graphs from 517
- statistical limits
 - example, adding to dot plots 506
- step plots
 - about step plots 30
 - example 30

SGPANEL procedure 204
 SGPLOT procedure 413
 STEP statement
 about step plots 30
 example 30
 SGPANEL procedure 204
 SGPLOT procedure 413
 style attributes
 automatic differentiation of 83, 279
 SGPANEL procedure 83
 SGPLOT procedure 279
 style element references 626
 style elements
 for ODS statistical graphics 640
 STYLE= option
 changing current graph style 624
 styles 620
 See also ODS styles
 changing current style 624
 changing default 625
 changing default, SAS Registry 625
 default 621
 default styles 621
 examples 623
 modifying templates 635
 recommended 621
 viewing templates 622

T

TEMPLATE procedure
 DEFINE STYLE statement 635
 modifying styles 635
 TEMPLATE= argument
 PROC SGRENDER statement 514
 templates
 See [statgraph templates](#)
 text box
 about text insets 31
 adding, inside plot axes 356
 example 31
 TEXT function (annotation) 592
 text strings
 substituting BY line values in 62
 TEXTATTRS= option
 INSET statement (SGPLOT) 358
 TEXTCONT function (annotation) 596
 THRESHOLDMAX= option
 COLAXIS statement (SGPANEL) 257
 ROWAXIS statement (SGPANEL) 265
 X2AXIS statement (SGPLOT) 481
 XAXIS statement (SGPLOT) 473
 Y2AXIS statement (SGPLOT) 496
 YAXIS statement (SGPLOT) 488
 THRESHOLDMIN= option
 COLAXIS statement (SGPANEL) 257

ROWAXIS statement (SGPANEL) 265
 X2AXIS statement (SGPLOT) 481
 XAXIS statement (SGPLOT) 473
 Y2AXIS statement (SGPLOT) 496
 YAXIS statement (SGPLOT) 488
 TICKVALUEFORMAT= option
 COLAXIS statement (SGPANEL) 257
 ROWAXIS statement (SGPANEL) 265
 X2AXIS statement (SGPLOT) 481
 XAXIS statement (SGPLOT) 473
 Y2AXIS statement (SGPLOT) 496
 YAXIS statement (SGPLOT) 488
 time axes 82, 279
 TITLE statement 61
 BY statement with 53
 measurement units 63
 substituting BY line values in a text string 62
 TITLE= option
 INSET statement (SGPLOT) 359
 KEYLEGEND statement (SGPANEL) 158
 KEYLEGEND statement (SGPLOT) 362
 TITLEATTRS= option
 INSET statement (SGPLOT) 359
 KEYLEGEND statement (SGPANEL) 158
 KEYLEGEND statement (SGPLOT) 362
 titles 61
 TMPLOUT= option
 PROC SGPLOT statement 283
 PROC SGSCATTER statement 529
 TRANSPARENCY= option
 BAND statement (SGPANEL) 95
 BAND statement (SGPLOT) 289
 BUBBLE statement (SGPANEL) 100
 BUBBLE statement (SGPLOT) 295
 COMPARE statement (SGSCATTER) 537
 DENSITY statement (SGPANEL) 103
 DENSITY statement (SGPLOT) 297
 DOT statement (SGPANEL) 111
 DOT statement (SGPLOT) 306
 ELLIPSE statement (SGPLOT) 309
 HBAR statement (SGPANEL) 119
 HBAR statement (SGPLOT) 317
 HBARPARM statement (SGPANEL) 126
 HBARPARM statement (SGPLOT) 323
 HBOX statement (SGPANEL) 136
 HBOX statement (SGPLOT) 333
 HISTOGRAM statement (SGPANEL) 147

- HISTOGRAM statement (SGPLOT) 346
- HLINE statement (SGPANEL) 157
- HLINE statement (SGPLOT) 355
- LINEPARM statement (SGPANEL) 165
- LINEPARM statement (SGPLOT) 371
- MATRIX statement (SGSCATTER) 540
- NEEDLE statement (SGPANEL) 176
- NEEDLE statement (SGPLOT) 382
- PLOT statement (SGSCATTER) 548
- REFLINE statement (SGPANEL) 185
- REFLINE statement (SGPLOT) 392
- SCATTER statement (SGPANEL) 197
- SCATTER statement (SGPLOT) 405
- SERIES statement (SGPANEL) 204
- SERIES statement (SGPLOT) 412
- STEP statement (SGPANEL) 211
- STEP statement (SGPLOT) 420
- VBAR statement (SGPANEL) 220
- VBAR statement (SGPLOT) 429
- VBARPARM statement (SGPANEL) 226
- VBARPARM statement (SGPLOT) 436
- VBOX statement (SGPANEL) 236
- VBOX statement (SGPLOT) 445
- VECTOR statement (SGPANEL) 241
- VECTOR statement (SGPLOT) 451
- VLINE statement (SGPANEL) 251
- VLINE statement (SGPLOT) 461
- WATERFALL statement (SGPLOT) 467
- TYPE= option
 - BAND statement (SGPANEL) 95
 - BAND statement (SGPLOT) 289
 - COLAXIS statement (SGPANEL) 257
 - DENSITY statement (SGPANEL) 103
 - DENSITY statement (SGPLOT) 297
 - ELLIPSE statement (SGPLOT) 309
 - ROWAXIS statement (SGPANEL) 265
 - X2AXIS statement (SGPLOT) 481
 - XAXIS statement (SGPLOT) 473
 - Y2AXIS statement (SGPLOT) 496
 - YAXIS statement (SGPLOT) 489
- U**
 - UNIFORM= option
 - PROC SGPLOT statement 283
 - UNISCALE= option
 - PANELBY statement (SGPANEL) 90
 - PLOT statement (SGSCATTER) 548
 - UPPER= argument
 - BAND statement (SGPANEL) 92
 - BAND statement (SGPLOT) 286
 - URL drill down 654
 - URL= option
 - BUBBLE statement (SGPANEL) 100
 - BUBBLE statement (SGPLOT) 295
 - DOT statement (SGPANEL) 111
 - HBAR statement (SGPANEL) 119
 - HBAR statement (SGPLOT) 317
 - HBARPARM statement (SGPANEL) 126
 - HBARPARM statement (SGPLOT) 323
 - NEEDLE statement (SGPANEL) 176
 - NEEDLE statement (SGPLOT) 382
 - SCATTER statement (SGPANEL) 197
 - SCATTER statement (SGPLOT) 157, 251, 306, 355, 405, 461
 - SERIES statement (SGPANEL) 204
 - SERIES statement (SGPLOT) 412
 - STEP statement (SGPANEL) 211
 - STEP statement (SGPLOT) 420
 - VBAR statement (SGPANEL) 220
 - VBAR statement (SGPLOT) 429
 - VBARPARM statement (SGPANEL) 226
 - VBARPARM statement (SGPLOT) 436
 - WATERFALL statement (SGPLOT) 467
 - user-defined formats
 - associating with variables 54
 - user-specified image size 653
- V**
 - VALUEALIGN= option
 - INSET statement (SGPLOT) 360
 - VALUEATTRS= option
 - COLAXIS statement (SGPANEL) 258
 - KEYLEGEND statement (SGPANEL) 159
 - KEYLEGEND statement (SGPLOT) 363
 - ROWAXIS statement (SGPANEL) 266
 - X2AXIS statement (SGPLOT) 481
 - XAXIS statement (SGPLOT) 474
 - YAXIS statement (SGPLOT) 489
 - VALUES= option
 - COLAXIS statement (SGPANEL) 258
 - ROWAXIS statement (SGPANEL) 266
 - X2AXIS statement (SGPLOT) 482
 - XAXIS statement (SGPLOT) 474
 - Y2AXIS statement (SGPLOT) 497
 - YAXIS statement (SGPLOT) 490
 - VALUESHINT option
 - COLAXIS statement (SGPANEL) 259

ROWAXIS statement (SGPANEL) 267
 X2AXIS statement (SGPLOT) 483
 XAXIS statement (SGPLOT) 475
 Y2AXIS statement (SGPLOT) 498
 YAXIS statement (SGPLOT) 490
 variables
 associating descriptive labels with 57
 associating formats with 54
 VBAR statement
 about bar charts 42
 example 42, 78, 276, 509
 SGPANEL procedure 212
 SGPLOT procedure 420
 VBARPARAM statement
 about bar charts 42
 example 42
 SGPANEL procedure 220
 SGPLOT procedure 429
 VBOX statement
 about box plots 37
 example 37
 SGPANEL procedure 227
 SGPLOT procedure 436
 vector plots
 about vector plots 31
 example 31
 SGPANEL procedure 238
 SGPLOT procedure 447
 VECTOR statement
 about vector plots 31
 example 31
 SGPANEL procedure 238
 SGPLOT procedure 447
 vertical bar charts
 about bar charts 42
 example 42, 78, 276
 parameterized (SGPANEL) 220
 parameterized (SGPLOT) 429
 SGPANEL procedure 212
 SGPLOT procedure 420
 vertical bar-line charts
 example 509
 vertical box plots
 about 37
 example 37
 SGPANEL procedure 227
 SGPLOT procedure 436
 vertical line
 SGPANEL procedure 161
 SGPLOT procedure 366
 vertical line charts
 about 47
 example 47, 272
 SGPANEL procedure 242
 SGPLOT procedure 451
 visual attributes

automatic differentiation of 83
 SGPLOT procedure 279
 VLINE statement
 about line charts 47
 example 47, 509
 SGPANEL procedure 242
 SGPLOT procedure 451

W

WALLPERCENT (annotation space) 562
 WALLPIXEL (annotation space) 562
 waterfall charts
 about waterfall charts 49
 example 49
 WATERFALL statement 462
 WATERFALL statement
 about waterfall charts 49
 example 49
 SGPLOT procedure 462
 WEIGHT= option
 DOT statement (SGPANEL) 111
 DOT statement (SGPLOT) 306
 HBAR statement (SGPANEL) 120
 HBAR statement (SGPLOT) 317
 HLINE statement (SGPANEL) 157
 HLINE statement (SGPLOT) 355
 LOESS statement (SGPLOT) 377
 PBSPLINE statement (SGPLOT) 390
 REG statement (SGPLOT) 399
 VBAR statement (SGPANEL) 220
 VBAR statement (SGPLOT) 429
 VLINE statement (SGPANEL) 251
 VLINE statement (SGPLOT) 462
 WHISKERATTRS= option
 HBOX statement (SGPANEL) 136
 HBOX statement (SGPLOT) 333
 VBOX statement (SGPANEL) 236
 VBOX statement (SGPLOT) 445
 whiskers
 horizontal box plots 136, 334
 vertical box plots 237, 446

X

X= argument
 BAND statement (SGPANEL) 92
 BAND statement (SGPLOT) 285
 ELLIPSE statement (SGPLOT) 307
 LOESS statement (SGPANEL) 166
 LOESS statement (SGPLOT) 373
 NEEDLE statement (SGPANEL) 172
 NEEDLE statement (SGPLOT) 379
 PBSPLINE statement (SGPANEL) 177
 PBSPLINE statement (SGPLOT) 384
 REG statement (SGPANEL) 186

- REG statement (SGPLOT) 394
- SCATTER statement (SGPANEL) 193
- SCATTER statement (SGPLOT) 401
- SERIES statement (SGPANEL) 199
- SERIES statement (SGPLOT) 407
- STEP statement (SGPANEL) 206
- STEP statement (SGPLOT) 414
- VECTOR statement (SGPANEL) 239
- VECTOR statement (SGPLOT) 448
- X= option
 - COMPARE statement (SGSCATTER) 530
- X2AXIS option
 - BAND statement (SGPLOT) 289
 - DENSITY statement (SGPLOT) 298
 - ELLIPSE statement (SGPLOT) 309
 - HBOX statement (SGPLOT) 334
 - HISTOGRAM statement (SGPLOT) 346
 - HLINE statement (SGPLOT) 356
 - LOESS statement (SGPLOT) 377
 - NEEDLE statement (SGPLOT) 382
 - PBSPLINE statement (SGPLOT) 390
 - REG statement (SGPLOT) 399
 - SCATTER statement (SGPLOT) 405
 - SERIES statement (SGPLOT) 412
 - STEP statement (SGPLOT) 420
 - VBOX statement (SGPLOT) 446
 - VECTOR statement (SGPLOT) 451
 - VLINE statement (SGPLOT) 462
- X2AXIS statement
 - SGPLOT procedure 475
- X2AXIS= option
 - BUBBLE statement (SGPLOT) 295
 - LINEPARM statement (SGPLOT) 371
 - WATERFALL statement (SGPLOT) 467
- XAXIS statement
 - SGPLOT procedure 467
- XERRORLOWER= option
 - SCATTER statement (SGPANEL) 198
 - SCATTER statement (SGPLOT) 405
- XERRORUPPER= option
 - SCATTER statement (SGPANEL) 198
 - SCATTER statement (SGPLOT) 406
- XORIGIN= option
 - VECTOR statement (SGPANEL) 241
 - VECTOR statement (SGPLOT) 451
- Y**
 - Y= argument
 - ELLIPSE statement (SGPLOT) 307
 - LOESS statement (SGPANEL) 166
 - LOESS statement (SGPLOT) 373
 - NEEDLE statement (SGPANEL) 172
 - NEEDLE statement (SGPLOT) 379
 - PBSPLINE statement (SGPANEL) 177
 - PBSPLINE statement (SGPLOT) 384
 - REG statement (SGPANEL) 186
 - REG statement (SGPLOT) 394
 - SCATTER statement (SGPANEL) 193
 - SCATTER statement (SGPLOT) 401
 - SERIES statement (SGPANEL) 199
 - SERIES statement (SGPLOT) 407
 - STEP statement (SGPANEL) 206
 - STEP statement (SGPLOT) 414
 - VECTOR statement (SGPANEL) 239
 - VECTOR statement (SGPLOT) 448
 - Y= option
 - COMPARE statement (SGSCATTER) 530
 - Y2AXIS option
 - BAND statement (SGPLOT) 289
 - DENSITY statement (SGPLOT) 298
 - ELLIPSE statement (SGPLOT) 309
 - HBOX statement (SGPLOT) 334
 - HISTOGRAM statement (SGPLOT) 346
 - HLINE statement (SGPLOT) 356
 - LOESS statement (SGPLOT) 377
 - NEEDLE statement (SGPLOT) 383
 - PBSPLINE statement (SGPLOT) 390
 - REG statement (SGPLOT) 399
 - SCATTER statement (SGPLOT) 406
 - SERIES statement (SGPLOT) 413
 - STEP statement (SGPLOT) 420
 - VBOX statement (SGPLOT) 446
 - VECTOR statement (SGPLOT) 451
 - VLINE statement (SGPLOT) 462
 - Y2AXIS statement
 - SGPLOT procedure 491
 - Y2AXIS= option
 - BUBBLE statement (SGPLOT) 295
 - LINEPARM statement (SGPLOT) 371
 - WATERFALL statement (SGPLOT) 467
 - YAXIS statement
 - SGPLOT procedure 483
 - YERRORLOWER= option
 - SCATTER statement (SGPANEL) 198
 - SCATTER statement (SGPLOT) 406
 - STEP statement (SGPANEL) 212
 - STEP statement (SGPLOT) 420
 - YERRORUPPER= option
 - SCATTER statement (SGPANEL) 198
 - SCATTER statement (SGPLOT) 406
 - STEP statement (SGPANEL) 212
 - STEP statement (SGPLOT) 420
 - YORIGIN= option
 - VECTOR statement (SGPANEL) 241
 - VECTOR statement (SGPLOT) 451