

CHAPTER

1

Introduction to SAS/GRAPH Software

<i>Overview</i>	4
<i>Components of SAS/GRAPH Software</i>	4
<i>Device-Based Graphics and Template-Based Graphics</i>	6
<i>Graph Types</i>	6
<i>Charts</i>	7
<i>Block charts</i>	7
<i>Horizontal bar charts</i>	7
<i>Vertical bar charts</i>	8
<i>Pie charts, Detailed pie charts, 3D pie charts, and Donut charts</i>	8
<i>Star charts</i>	9
<i>Bar-line Charts</i>	10
<i>Area Bar Charts</i>	10
<i>Tile Charts</i>	11
<i>Radar Charts</i>	12
<i>Two-Dimensional Plots</i>	12
<i>Two-dimensional scatter plots</i>	12
<i>Simple line plots</i>	13
<i>Regression plots</i>	13
<i>High-low plots</i>	14
<i>Bubble plots</i>	15
<i>Three-Dimensional Plots</i>	15
<i>Surface plots</i>	15
<i>Scatter plots</i>	16
<i>Contour plots</i>	16
<i>Maps</i>	17
<i>Block maps</i>	17
<i>Choropleth maps</i>	18
<i>Prism maps</i>	18
<i>Surface maps</i>	19
<i>KPI Charts</i>	20
<i>Creating Text Slide and Presentation Graphics</i>	20
<i>Text Slides</i>	20
<i>Combining Output into One Slide</i>	21
<i>Enhancing Graphics Output (graphs and text slides)</i>	22
<i>SAS/GRAPH Statements</i>	22
<i>The Annotate Facility</i>	22
<i>Creating Custom Graphics</i>	22
<i>The DATA Step Graphics Interface</i>	22
<i>Graph-N-Go</i>	23
<i>About this Document</i>	23
<i>Audience</i>	23

Prerequisites	23
Conventions Used in This Document	24
Syntax Conventions	24
Conventions for Examples and Output	26
Information You Should Know	27
Support Personnel	27
Sample Programs	27
Map Data Sets	29
Annotate Macros Data Set	29

Overview

SAS/GRAPH is the data visualization and presentation (graphics) component of the SAS System. As such, SAS/GRAPH:

- organizes the presentation of your data and visually represents the relationship between data values as two- and three-dimensional graphs, including charts, plots, and maps.
- enhances the appearance of your output by allowing you to select text fonts, colors, patterns, and line styles, and control the size and position of many graphics elements.
- creates presentation graphics. SAS/GRAPH can create text slides, display several graphs at one time, combine graphs and text in one display, and create automated presentations.
- generates a variety of graphics output that you can display on your screen or in a Web browser, store in catalogs, review, or send to a hard copy graphics output device such as a laser printer, plotter, or slide camera.
- provides utility procedures and statements to manage the output.

This chapter describes the graphs that are produced by SAS/GRAPH and explains some of the parts and features of SAS/GRAPH programs.

Components of SAS/GRAPH Software

There are several components to SAS/GRAPH software.

Device-based SAS/GRAPH procedures

enable you to create a variety of graphs, including bar charts, pie charts, scatter plots, surface plots, contour plots, a variety of maps, and much more. The device-based SAS/GRAPH procedures include the GAREABAR, GCHART, GPLOT, GMAP, GBARLINE, GKPI, GCONTOUR, and G3D procedures, as well as others. These procedures use device drivers to generate output. SAS/GRAPH device drivers enable you to send output directly to your output device as well as create output in a variety of formats such as PNG files and interactive ActiveX controls or Java applets. This document, *SAS/GRAPH: Reference*, describes the device-based SAS/GRAPH procedures and how to use devices. See also “Device-Based Graphics and Template-Based Graphics” on page 6.

The Annotate Facility

enables you to generate a special data set of graphics commands from which you can produce graphics output. This data set is referred to as an Annotate data set. You can use it to generate custom graphics or to enhance graphics output from

many device-based SAS/GRAPH procedures, including GCHART, GPLOT, GMAP, GBARLINE, GCONTOUR, and G3D, as well as others. For more information, see Chapter 29, “Using Annotate Data Sets,” on page 643.

Network Visualization (NV) Workshop

enables you to visualize and investigate the patterns and relationships hidden in network data (node-link data). Some common applications that use network data include supply chains, communication networks, Web sites, database schema, and software module dependencies. NV Workshop is designed for visualizing large networks. Using a combination of data tables, statistical graphs, and network graphs, NV Workshop enables you to extract information that would otherwise remain hidden. Help is available from the menu within the product. To start NV Workshop, select **Start ▶ Programs ▶ SAS ▶ SAS GRAPH NV Workshop 2.1**. For additional information, see *SAS/GRAPH: Network Visualization Workshop User’s Guide*.

SAS/GRAPH statistical graphics suite

is part of ODS Statistical Graphics (referred to as ODS Graphics for short). ODS Graphics is functionality for creating statistical graphics that is available in a number of SAS software products, including SAS/STAT, SAS/ETS, SAS/QC, and SAS/GRAPH. The SAS/GRAPH statistical graphics suite provides the following features:

SAS/GRAPH statistical graphics procedures

provide a simple syntax for creating graphics commonly used in exploratory data analysis and for creating customized statistical displays. These procedures include the SGPANEL, SGPLOT, and SGSCATTER procedures. In addition, the SGRENDER procedure provides a SAS procedure interface to create graphs using the Graph Template Language. These procedures are template-based procedures; they do not use devices like the device-based SAS/GRAPH procedures. For more information, see “Device-Based Graphics and Template-Based Graphics” on page 6 and *SAS/GRAPH: Statistical Graphics Procedures Guide*.

Graph Template Language (GTL)

is the underlying language for the default templates that are provided by SAS for procedures that use ODS Statistical Graphics. You can use the GTL either to modify these templates or to create your own customized graphs. Templates written with the GTL are built with the TEMPLATE procedure. For more information about Graph Template Language, see *SAS/GRAPH: Graph Template Language User’s Guide* and *SAS/GRAPH: Graph Template Language Reference*.

ODS Graphics Editor

is an interactive editor that enables you to edit and enhance graphs that are produced by procedures that use ODS Graphics. You can use the ODS Graphics Editor to modify the existing elements of a graph such as titles and labels, or to add features such as text annotation for data points. For more information, see *SAS/GRAPH: ODS Graphics Editor User’s Guide*.

ODS Graphics Designer

provides a point-and-click interface for creating ODS Graphics. Using the ODS Graphics Designer does not require knowledge of ODS templates or the Graph Template Language. With the ODS Graphics Designer, you can easily create multi-cell graphs, classification panels, scatter plot matrices, and more. You can save your output as an image file or as an ODS Graphics Designer file (SGD file) that you can edit later. For more information, see the

SAS Help and Documentation. Select **SAS Products** ► **SAS/GRAPH** ► **SAS/GRAPH Windows** ► **ODS Graphics Designer**.

For additional information on the ODS Statistical Graphics functionality, see *SAS Output Delivery System: User's Guide* and *SAS/STAT User's Guide*.

Device-Based Graphics and Template-Based Graphics

SAS/GRAPH produces graphics using two very distinct systems. SAS/GRAPH can produce output using a device-based system or using a template-based system. The traditional system for producing graphics output that most users are familiar with is the device-based system.

device-based graphics

are SAS/GRAPH output that is generated by a default or user-specified device (DEVICE= option). Device drivers supplied by SAS are stored in the SAS/GRAPH catalog. Examples of device drivers are GIF, PNG, ACTIVEX, SVG, and SASPRTC. Common SAS/GRAPH procedures that produce device-based graphics include the GCHART, GPLOT, GMAP, GBARLINE, GCONTOUR, and G3D procedures. Most procedures that produce device-based graphics also produce GRSEG catalog entries in addition to any image files that are produced. For device-based graphics, you can use the GOPTIONS statement to control the graphical environment.

template-based graphics

are SAS/GRAPH output that is produced from a compiled ODS template of type STATGRAPH. Templates supplied by SAS are stored in SAS/GRAPH. Device drivers and most global statements (such as SYMBOL, PATTERN, AXIS, and LEGEND) have no effect on template-based graphics. The SAS/GRAPH procedures that produce template-based graphics are the SGPLOT, SGPANEL, SGSCATTER, and SGRENDER procedures. Many SAS/STAT, SAS/ETS, and SAS/QC procedures also produce template-based graphics when you specify the ODS GRAPHICS ON statement. (Template-based graphics are frequently referred to as *ODS graphics*.) 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.

The *SAS/GRAPH: Reference* contains information about device-based graphics only. For information about template-based graphics, see *SAS/GRAPH: Statistical Graphics Procedures Guide* and *SAS/GRAPH: Graph Template Language Reference*.

Graph Types

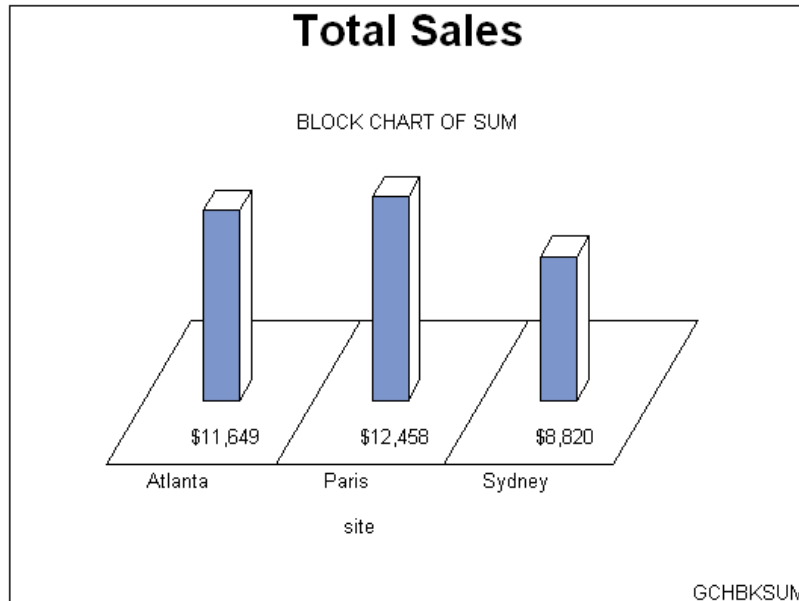
SAS/GRAPH produces many kinds of charts, plots, and maps in both two- and three-dimensional versions. In addition to helping you understand the variety of graphs that are available to you, these descriptions also help you choose the correct type of graph for your data and point you to the appropriate chapter.

Charts

SAS/GRAPH uses the GCHART procedure to produce charts that graphically represent the value of a statistic for one or more variables in a SAS data set. See Chapter 36, “The GCHART Procedure,” on page 989 for a complete description.

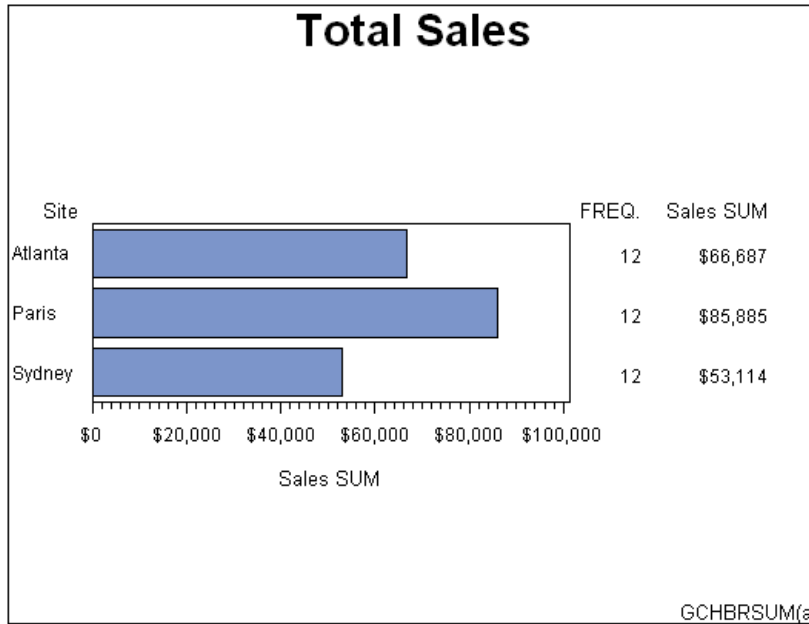
Block charts

Block charts use three-dimensional blocks to graphically represent values of statistics. Block charts are useful for emphasizing relative magnitudes and differences among data values.



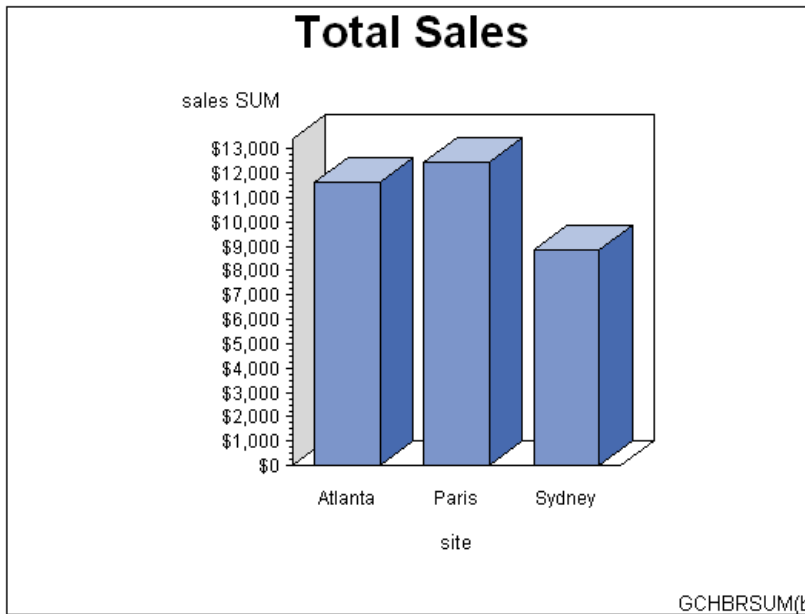
Horizontal bar charts

Horizontal bar charts use horizontal bars to represent statistics based on the values of one or more variables. Horizontal bar charts can generate a table of chart statistics and are useful for displaying exact magnitudes and emphasizing differences.



Vertical bar charts

Vertical bar charts use vertical bars to represent statistics based on the values of one or more variables. Vertical bar charts, which generate only one statistic, are useful for displaying exact magnitudes and emphasizing differences.



Pie charts, Detailed pie charts, 3D pie charts, and Donut charts

Pie charts, detailed pie charts, 3-D pie charts, and Donut charts use the angle of pie slices to graphically represent the value of a statistic for a data range. Pie charts are useful for examining how the values of a variable contribute to the whole and for comparing the values of several variables.

Figure 1.1 Detailed Pie Chart

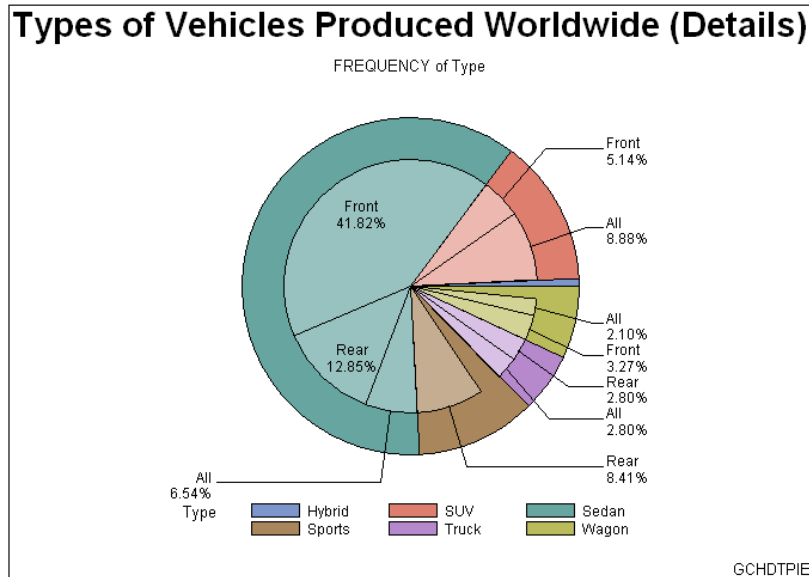
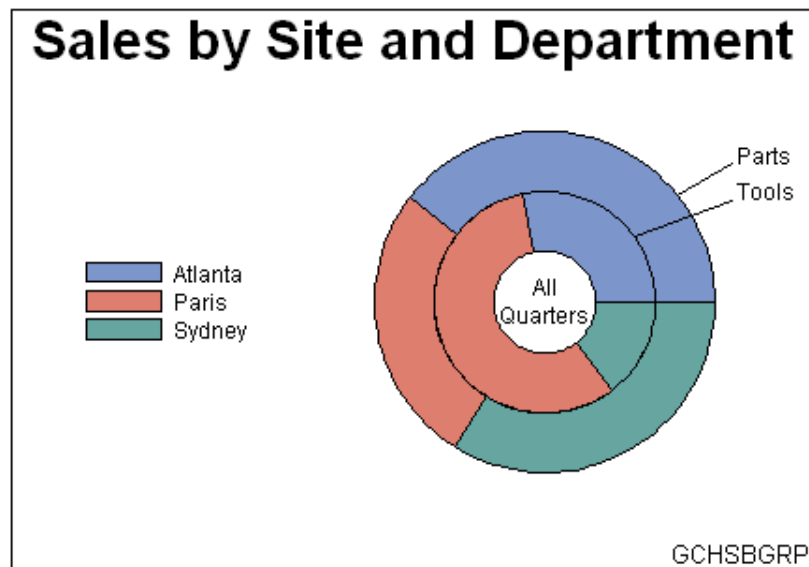
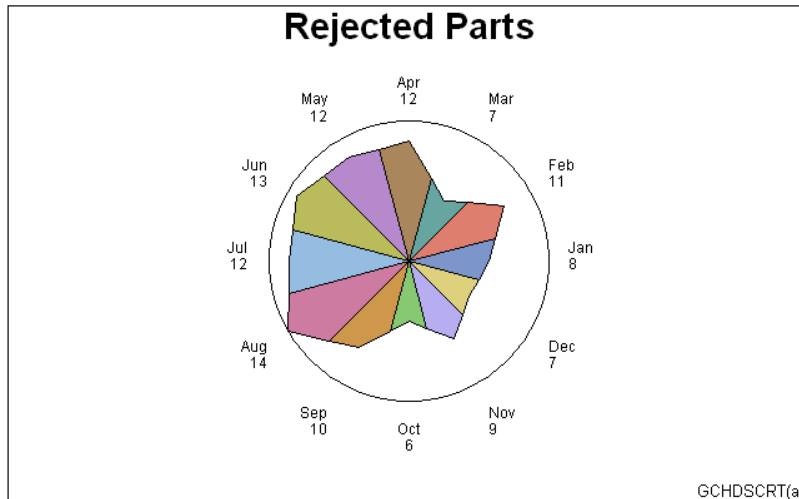


Figure 1.2 Donut Chart



Star charts

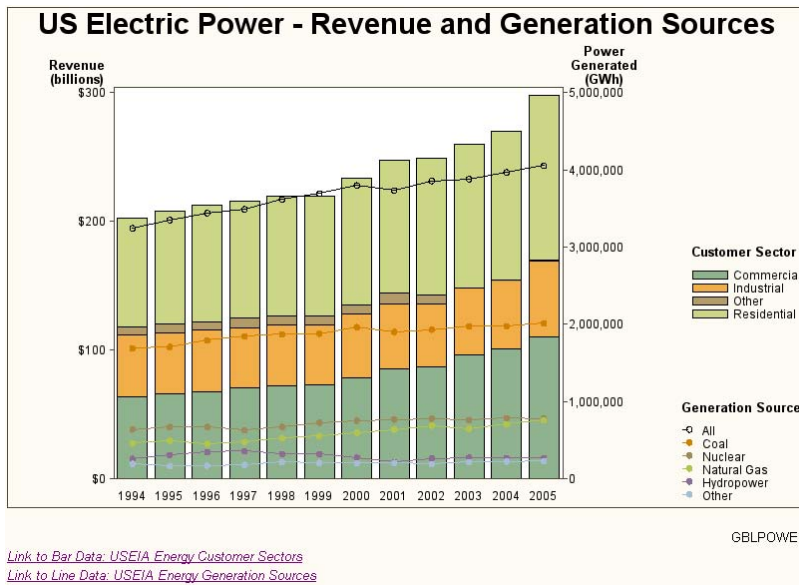
Star charts use the length of spines to graphically represent the value of a statistic for a data range. Star charts are useful for analyzing where data are out of balance.



Bar-line Charts

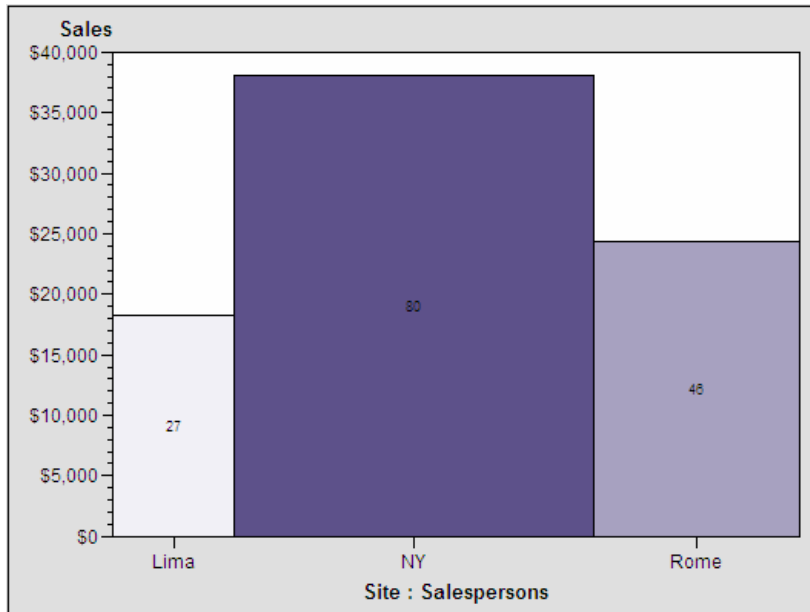
The GBARLINE procedure produces vertical bar charts with plot overlays. These charts graphically represent the value of a statistic calculated for one or more variables in an input SAS data set. The charted variables can be either numeric or character.

See Chapter 35, “The GBARLINE Procedure,” on page 947 for a complete description.



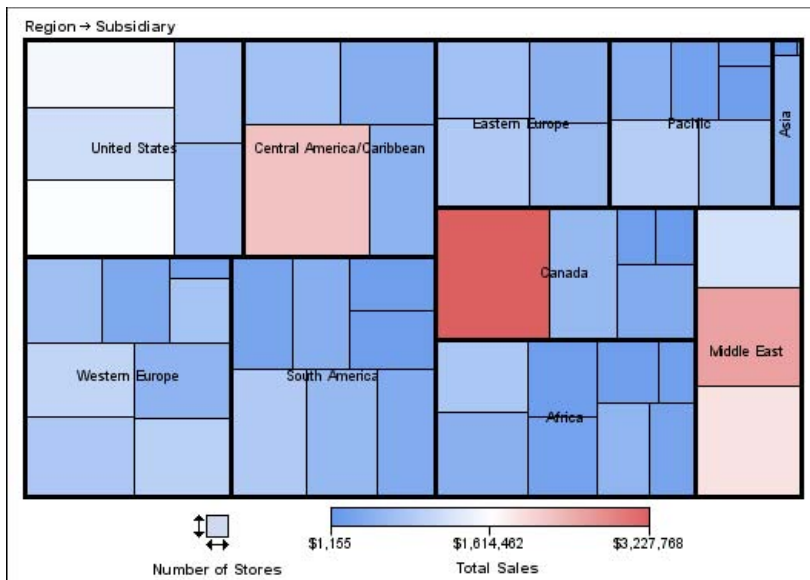
Area Bar Charts

The GAREABAR procedure produces area bar charts that show the magnitudes of *two* variables for each category of data. For example, the following area bar chart shows the sales total for each of three geographical sites. The width of each bar indicates the number of sales persons at each site. In a bar chart such as the chart shown in “Vertical bar charts” on page 8, the width is the same for each bar. In an area bar chart, the width and height of each bar is determined by the value of variables. See Chapter 34, “The GAREABAR Procedure,” on page 931 for a complete description.



Tile Charts

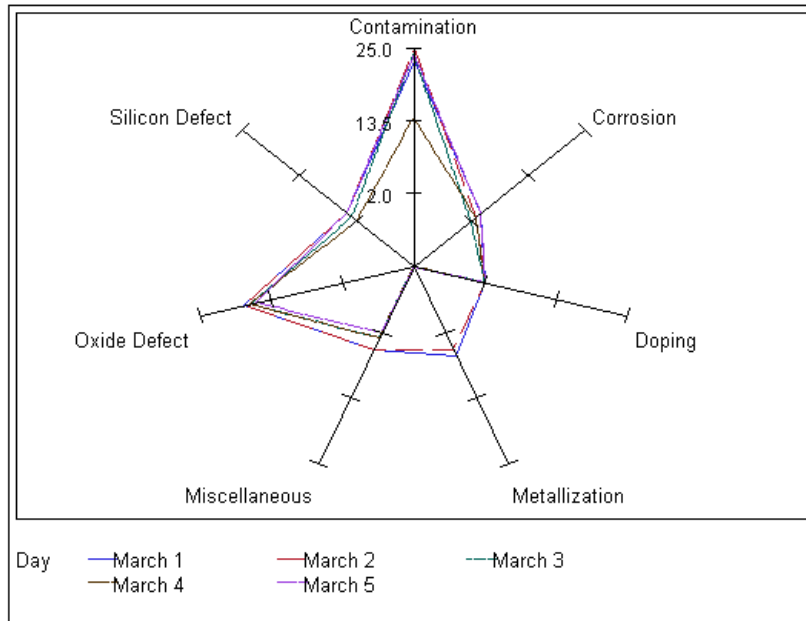
The GTILE procedure produces charts that tile charts, which consist of a rectangle or square divided into tiles. The sizes of the individual tiles represent the value of the size variable. You can also specify a color variable, so that the colors of the individual tiles represent the magnitude of the color variable. Tile charts are useful for determining the relative magnitude of categories of data or the contribution of a category toward the whole.



Radar Charts

The GRADAR procedure produces radar charts that show the relative frequency of data measures. On a radar chart, the chart statistics are displayed along spokes that radiate from the center of the chart. The charts are often stacked on top of one another with reference circles, thus giving them the look of a radar screen. Radar charts are frequently called star charts and are often used in quality control or market research problems.

See Chapter 47, “The GRADAR Procedure,” on page 1409 for a complete description.



Two-Dimensional Plots

SAS/GRAPH uses the GPLOT procedure to produce two-dimensional graphs that plot one or more dependent variables against an independent variable within a set of coordinate axes. GPLOT can display the data points as individual symbols (as in a scatter plot), or use interpolation methods specified by the SYMBOL statement to join the points, request spline interpolation or regression analysis, produce various high-low plots, or generate several other types of plots.

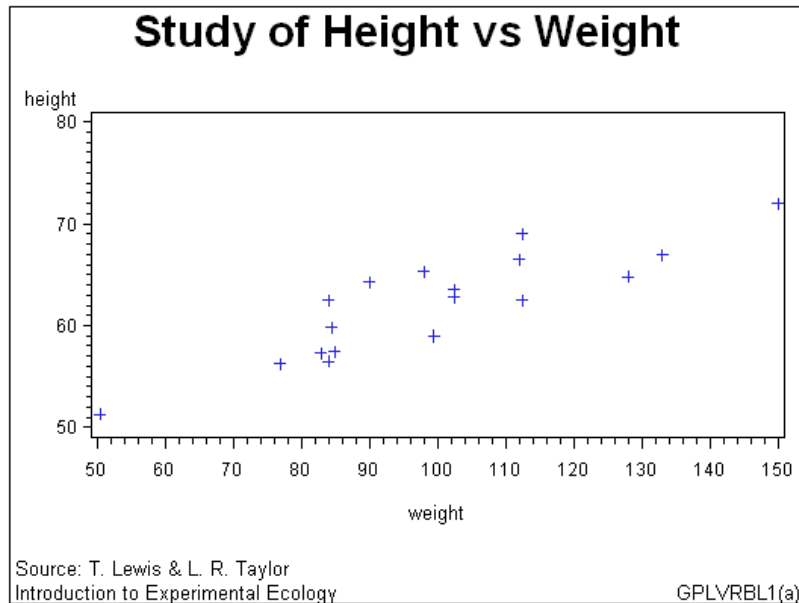
GPLOT can also display data as bubble plots in which circles of different sizes represent the values of a third variable.

Plots are useful for demonstrating the relationship between two or more variables and frequently compare trends or data values or depict movements of data values over time.

See Chapter 45, “The GPLOT Procedure,” on page 1315 for a complete description.

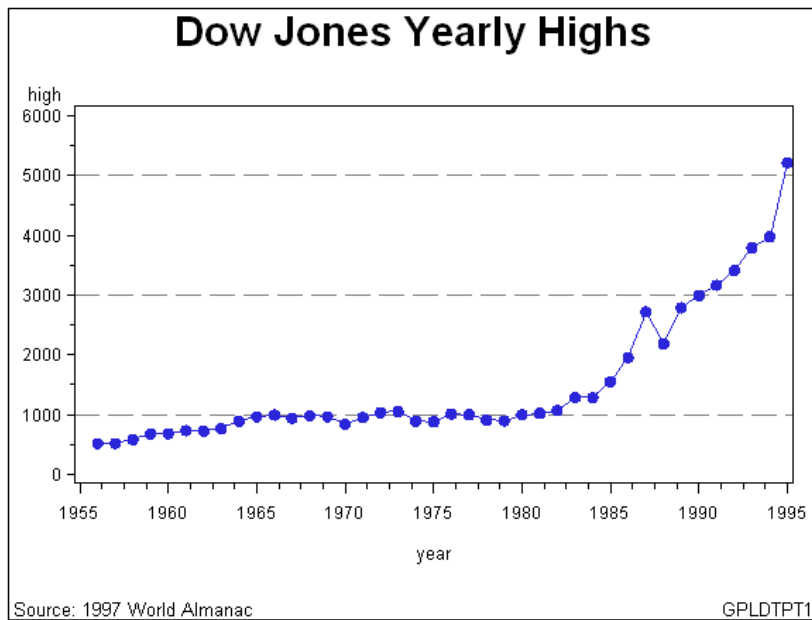
Two-dimensional scatter plots

Two-dimensional 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.



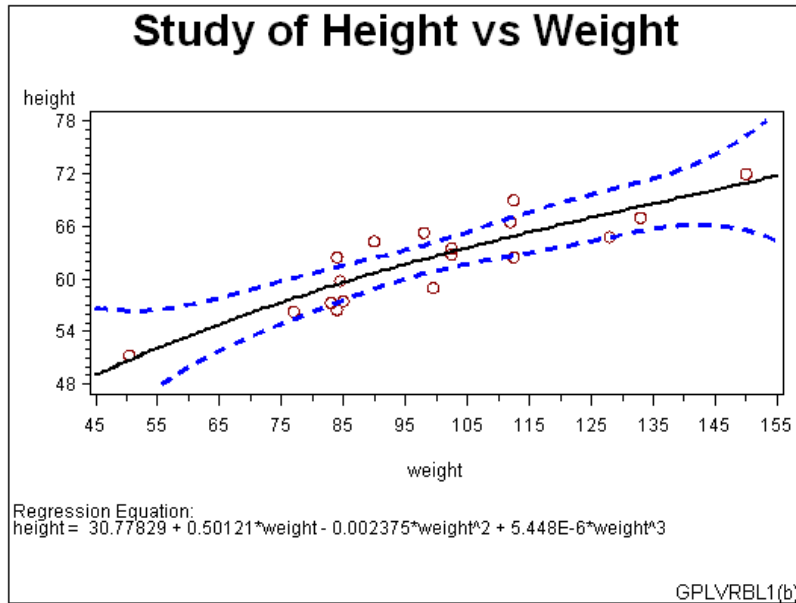
Simple line plots

Simple line plots show the relationship of one variable to another, often as movements or trends in the data over a period of time. Typically, each variable value on the horizontal axis has only one corresponding value on the vertical axis. The line connecting data points can be smoothed using a variety of interpolation methods, including the Lagrange and the cubic spline interpolation methods.



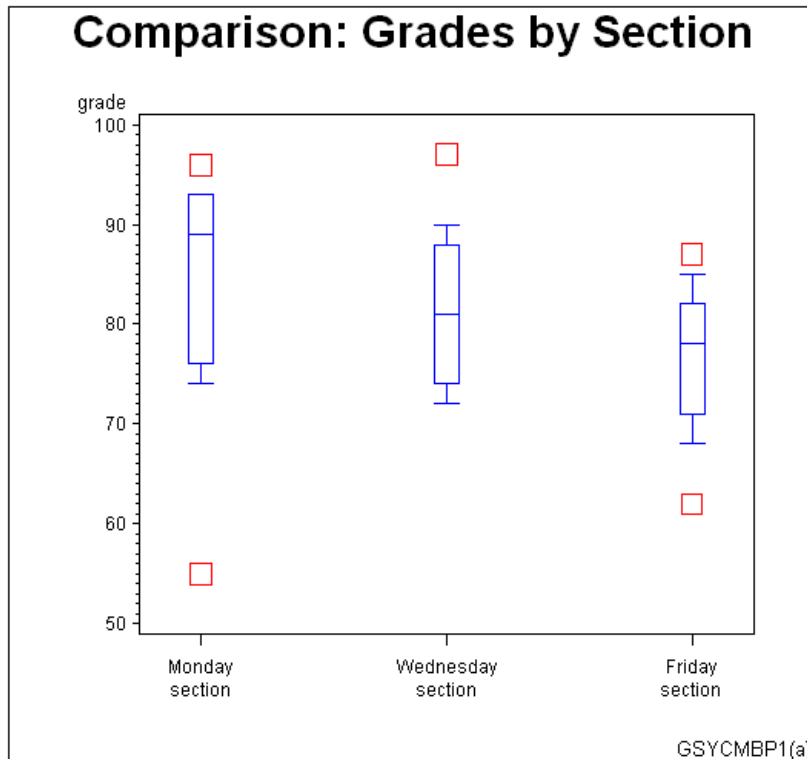
Regression plots

Regression plots specify that the plot is a regression analysis. You can specify one of three types of regression equation – linear, quadratic, or cubic, and you can choose to display confidence limits for mean predicted values or individual predicted values.



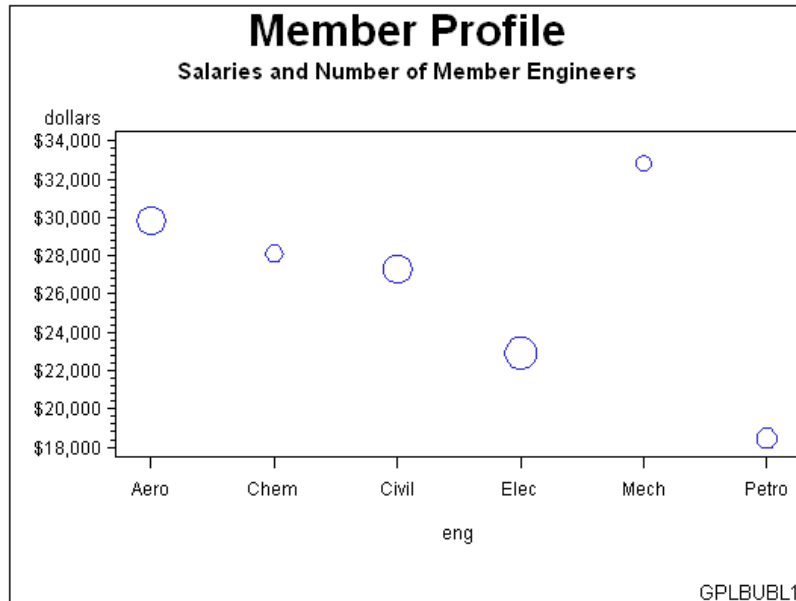
High-low plots

High-low plots 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. High-low plots include box, needle, and stock market plots.



Bubble plots

Bubble plots show the relative magnitude of one variable in relation to two other variables. 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.



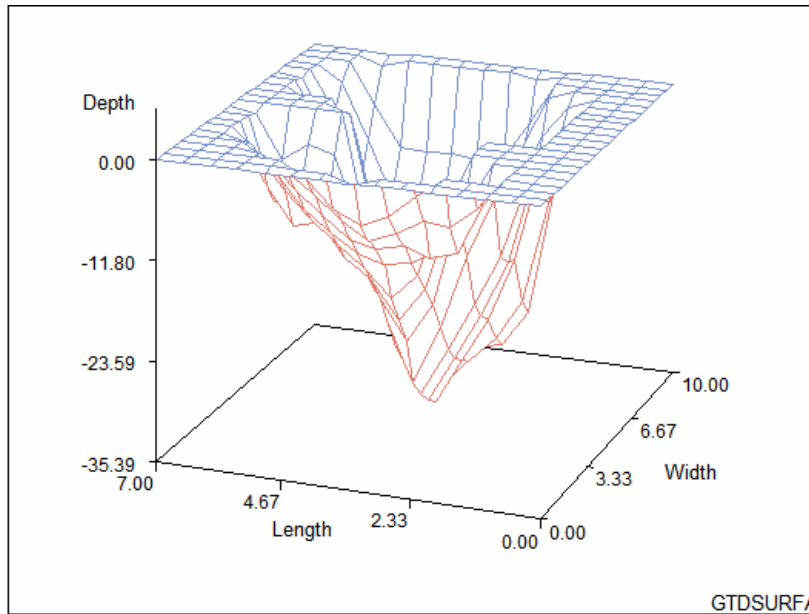
Three-Dimensional Plots

SAS/GRAPH uses the G3D procedure to produce three-dimensional surface and scatter plots that examine the relationship among three variables. Variable values are plotted on a set of three coordinate axes.

See Chapter 53, “The G3D Procedure,” on page 1533 for a complete description.

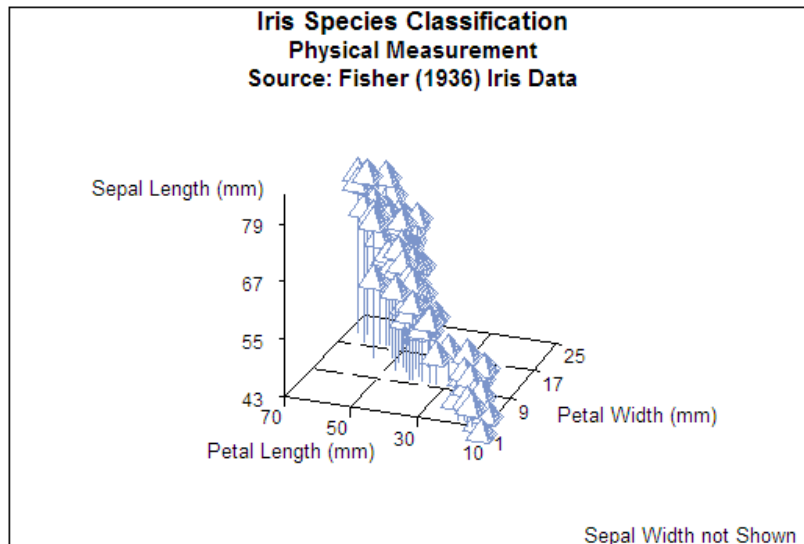
Surface plots

Surface plots are three-dimensional plots that display the relationship of three variables as a continuous surface. Surface plots examine the three-dimensional shape of data.



Scatter plots

Scatter plots enable you to examine three-dimensional data points instead of surfaces and to classify your data using size, color, shape, or a combination of these features.

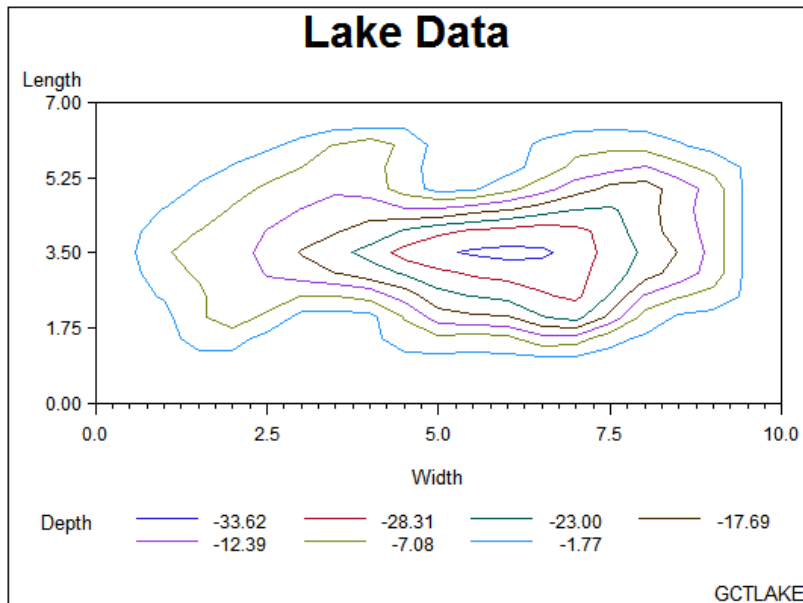


Contour plots

SAS/GRAPH uses the GCONTOUR procedure to examine three-dimensional data in two dimensions. Lines or areas in a contour plot represent levels of magnitude (z) corresponding to a position on a plane (x,y).

See Chapter 37, “The GCONTOUR Procedure,” on page 1095 for a complete description.

Contour plots are two-dimensional plots that show three-dimensional relationships. These plots use contour lines or patterns to represent levels of magnitude of a contour variable plotted on the horizontal and vertical axes.



When you need to interpolate or smooth data values that are used by the G3D and GCONTOUR procedures, use the G3GRID procedure. The G3GRID procedure does not produce graphics output but processes existing data sets to create data sets that the G3D or GCONTOUR procedure can use to produce three-dimensional surface or contour plots. See Chapter 54, “The G3GRID Procedure,” on page 1563 for a complete description.

Maps

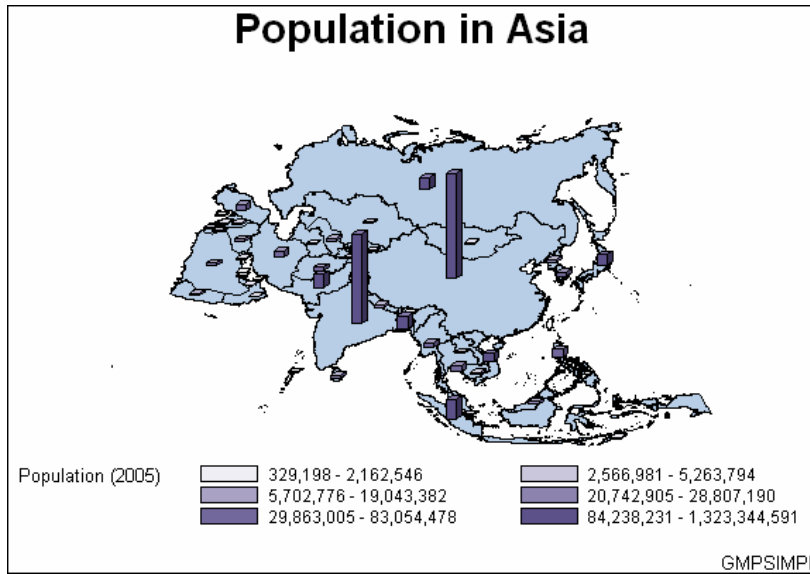
SAS/GRAPH uses the GMAP procedure to produce two- and three-dimensional maps that can show an area or represent values of response variables for subareas.

SAS/GRAPH includes data sets to produce geographic maps. In addition, you can create your own map data sets.

See Chapter 43, “The GMAP Procedure,” on page 1229 for a complete description.

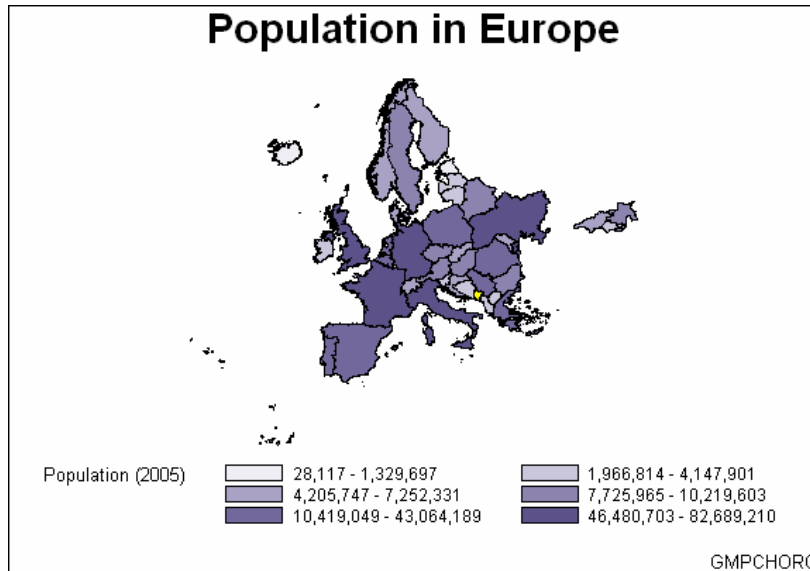
Block maps

Block maps are three-dimensional maps that represent data values as blocks of varying height rising from the middle of the map areas.



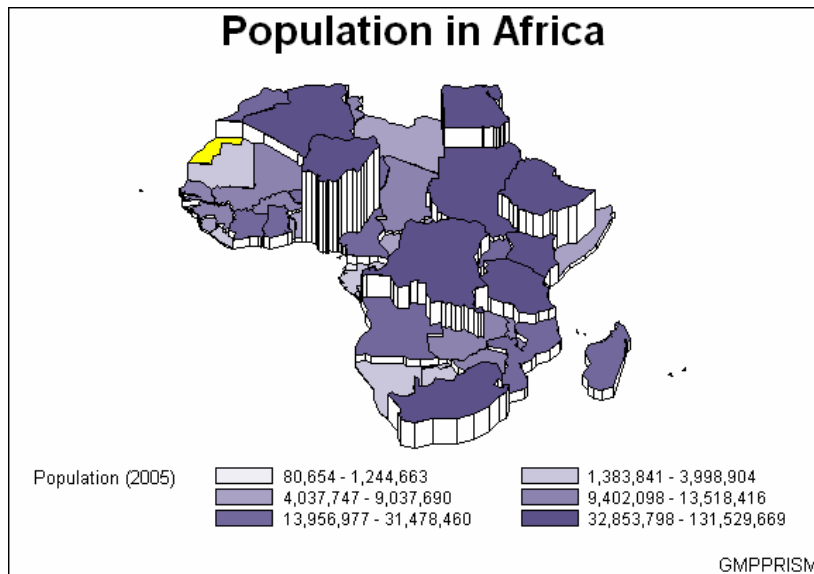
Choropleth maps

Choropleth maps are two-dimensional maps that display data values by filling map areas with combinations of patterns and color that represent the data values.



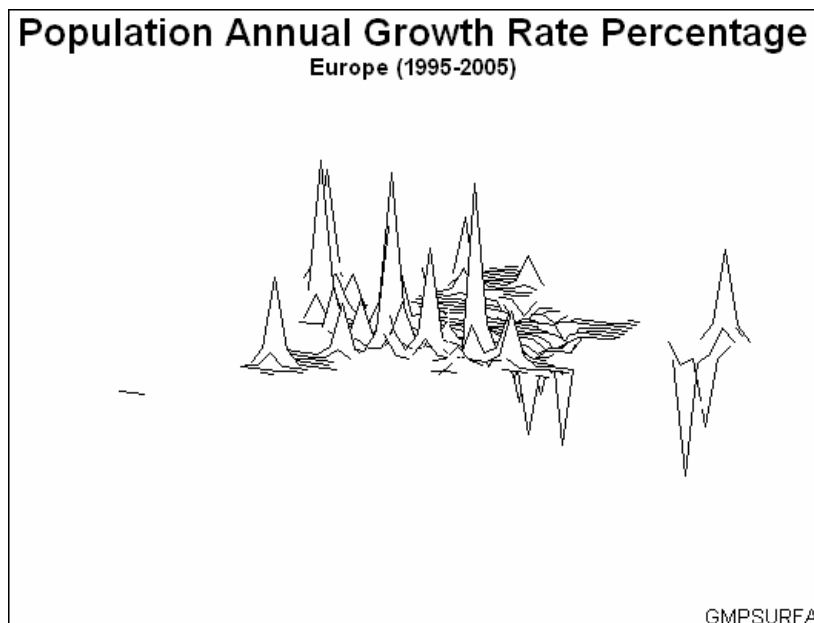
Prism maps

Prism maps are three-dimensional maps that display data by raising the map areas and filling them with combinations of patterns and colors.



Surface maps

Surface maps are three-dimensional maps that represent data values as spikes of varying heights.



SAS/GRAPH also provides several utility procedures for handling map data.

The GPROJECT procedure lets you choose how geographic maps are projected. This is particularly important for large areas because producing a map of any large area on the Earth involves distorting some areas in the process of projecting the spherical surface of the Earth onto a flat plane. You can use the procedure to select the projection method that least distorts your map.

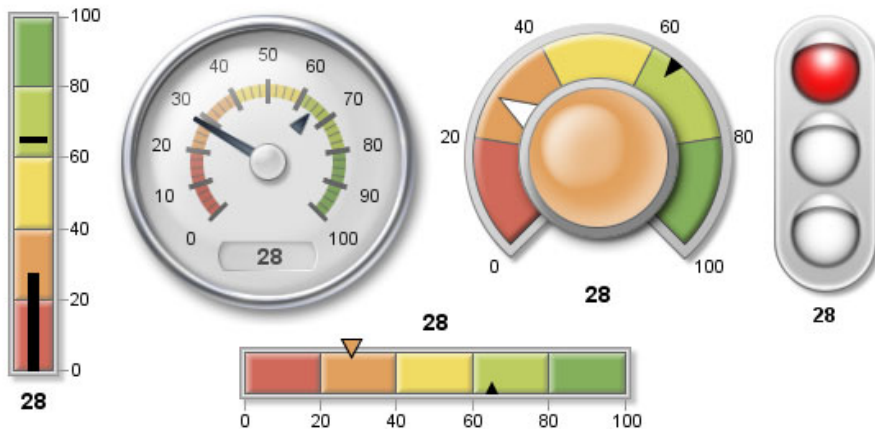
Map areas are constructed of joined data points. Each data point represents an observation in a SAS data set. For large maps, the amount of data can be prohibitively expensive (in terms of computing resources or time to process); the GREDUCE

procedure enables you to reduce the number of points in the data set. The GREMOVE procedure enables you to remove boundary lines within a map.

KPI Charts

The GKPI procedure creates graphical key performance indicator (KPI) charts. KPIs are metrics that help a business monitor its performance and measure its progress toward specific goals. The procedure produces five KPI chart types:

- slider (vertical or horizontal)
- bullet graph (vertical or horizontal)
- radial dial
- speedometer
- traffic light (vertical or horizontal).



Creating Text Slide and Presentation Graphics

You can use SAS/GRAPH to create slide presentations of your graphs. With SAS/GRAPH you can

- create text slides with the GSLIDE procedure
- combine several graphs into one output with the GREPLAY procedure
- automatically or manually replay your graphs and text slides with the GREPLAY procedure.

Text Slides

Use the GSLIDE procedure to create text slides in which you can specify a variety of colors, fonts, sizes, angles, overlays, and other modifications as well as drawing lines and boxes on the output.

See Chapter 51, “The GSLIDE Procedure,” on page 1509 for a complete description.

Text slides display text as graphics output. Text slides can be used as title slides for presentations, or to produce certificates, signs, or other display text.

New Directions

Goals and strategies for the coming year

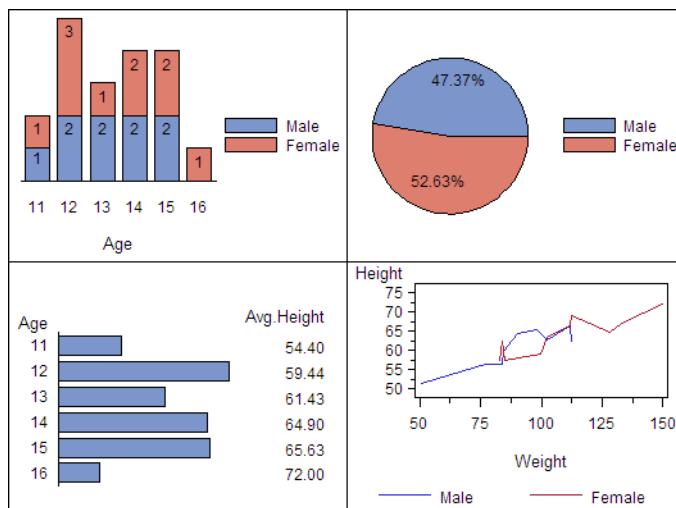
ABC Engineering, Inc
January 1998

Combining Output into One Slide

Use the GREPLAY procedure to combine several graphs into a single output. You can create special effects by overlaying or rotating the graphs at any angle.

Templated graphs display two or more graphs or text slides as one output by replaying stored graphs into a template or framework. Like graphs and text slides, templated graphs can be ordered in groups and stored in catalogs for replay as part of a presentation.

Figure 1.3 Templated graphs



In addition, you can use the GREPLAY procedure to create an automated or user-controlled presentation of graphics output. The GREPLAY procedure enables you to name, arrange, and customize the presentation of graphs that are stored in a catalog.

See Chapter 50, "The GREPLAY Procedure," on page 1465 for a complete description.

Enhancing Graphics Output (graphs and text slides)

SAS/GRAPH Statements

You can also use *global statements* and *graphics options* in SAS/GRAPH programs. With global statements, you can add titles and footnotes and control the appearance of axes, symbols, patterns, and legends. With graphics options, you can control the appearance of graphics elements by specifying default colors, fill patterns, fonts, text height, and so on.

The Annotate Facility

The Annotate facility enables you to program graphics by using certain variables in SAS data sets. It is often used to add text or special elements to the graphics output of other procedures, although it can also be used to construct custom graphics output. Text and graphics can be placed at coordinates derived from input data, as well as coordinates expressed as explicit locations on the display.

Figure 1.4 Annotated graphs



Creating Custom Graphics

The Annotate facility can also be used to generate custom graphics without using any of the SAS/GRAPH graphing procedures.

The DATA Step Graphics Interface

The DATA Step Graphics Interface provides functions and calls that produce graphics output from the DATA step, rather than from a procedure. The functions and calls are similar in form to those specified by the ISO Graphic Kernel Standard (GKS); however, the interface is not an implementation of the GKS. The form is similar enough that many GKS-compliant programs can be converted easily to run as SAS/GRAPH programs.

Graph-N-Go

To generate presentation graphs without writing any SAS/GRAPH code, you can use Graph-N-Go (not available on mainframes). You can start Graph-N-Go in several ways:

- from the menus in any SAS window, select **Solutions ► Reporting ► Graph-N-Go**
- submit either of the following from the SAS command line:

```
gng
```

```
graphngo
```

- use an Explorer window to directly open a GFORM entry. Double-click (or right-click and choose Open) on a GFORM entry to start a Graph-N-Go session using that entry.

Information on using the application is in Graph-N-Go help, which you can access from the application's main window in either of two ways:

- select **Help ► Using This Window**
- press F1 (this might not work in some operating environments).

You can also get help for the application by submitting the following command from the SAS command line:

```
help gng
```

About this Document

This document provides reference information for all facilities, procedures, statements, and options that can be used with SAS/GRAPH. This chapter describes what you need to know to use SAS/GRAPH, and what conventions are used in text and example code. To gain full benefit from using this document, you should familiarize yourself with the information presented in this chapter, and refer to it as needed.

Audience

This document is written for users who are experienced in using the SAS System. You should understand the concepts of programming in the SAS language, and you should have an idea of the tasks you want to perform with SAS/GRAPH.

Prerequisites

The following table summarizes the SAS System concepts that you need to understand in order to use SAS/GRAPH:

To learn how to	Refer to
invoke the SAS System at your site	instructions provided by the on-site SAS support personnel
use Base SAS software	<i>SAS Language Reference: Concepts or SAS Language Reference: Dictionary</i>
use the DATA step to create and manipulate SAS data sets	
use the SAS Text Editor to enter and edit text	

To learn how to	Refer to
allocate SAS libraries and assign librefs create external files and assign filerefs	documentation for using the SAS System under the operating system for the hardware at your site
manipulate SAS data sets using SAS procedures	<i>Base SAS Procedures Guide</i>

Conventions Used in This Document

This section explains the conventions this document uses for text, SAS language syntax, and file and library references. The document uses the following terms in discussing syntax:

keyword	is a literal that is a primary part of the SAS language. (A literal must be spelled exactly as shown, although it can be entered in uppercase or lowercase letters.) Keywords in this document are procedure names, statement names, macro names, routine names, and function names.
argument	is an element that follows a keyword. It is either literal, or it is user-supplied. It has a built-in value (for example, NODISPLAY), or it has a value assigned to it (for example, COLOR= <i>text-color</i>). Arguments that you must use are <i>required arguments</i> . Other arguments are <i>optional arguments</i> , or simply <i>options</i> .
value	is an element that follows an equal sign. It assigns a value to an argument. It might be a literal, or it might be a user-supplied value.
parameter	is a value assigned to an argument that itself takes a value, for example, the COLOR= parameter of the LABEL= option in a LEGEND statement, as shown in the following statement:

```
legend label=(color=blue);
```

Syntax Conventions

Type styles have special meanings when used in the presentation of SAS/GRAPH syntax in this document. The following list explains the style conventions for the syntax sections:

UPPERCASE	identifies SAS keywords such as the names of statements and procedures (for example, PROC GCHART). Uppercase characters also identify arguments and values that are literals (for example, NOLEGEND and LABEL=NONE).
<i>italic</i>	identifies arguments or values that you supply. Items in italic can represent user-supplied values that are either <ul style="list-style-type: none"> □ nonliteral values assigned to an argument (for example, <i>axis-color</i> in COLOR=<i>axis-color</i>) □ nonliteral arguments (for example, VBAR <i>chart-variable</i>. . . ;). <p>In addition, an item in italics can be the generic name for a list of arguments or parameters from which the user can choose (for example, <i>appearance-options</i>).</p>

The following symbols are used to indicate other syntax conventions:

< > (angle brackets)	identify optional arguments. Any argument not enclosed in angle brackets is required.
(vertical bar)	indicates that you can choose one value from a group. Values separated by bars are mutually exclusive.
. . . (ellipsis)	indicates that the argument following the ellipsis can be repeated any number of times (<i>plot-request</i> <. . . <i>plot-request-n</i> >, for example). If the ellipsis and the following argument are enclosed in angle brackets, they are optional. In SAS/GRAPH, an ellipsis also indicates a range from which a value is selected (LINE=1 . . . 46, for example).

The following examples illustrate the syntax conventions described in this section. These examples contain selected syntax elements, not complete syntax.

```
PROC GANNO ANNOTATE=Annotate-data-set
    <DATASYS>;
```

- PROC GANNO is in uppercase because it is a SAS keyword, the name of a statement. The remaining elements are arguments for the statement.
- ANNOTATE= is not enclosed in angle brackets because it is a required argument. It is in uppercase to indicate that it is a literal and must be spelled as shown.
- *Annotate-data-set* is in italic because it is a value that you must supply; in this case, the value must be a data set name.
- DATASYS is enclosed in angle brackets because it is an optional argument. It is in uppercase to indicate that it is a literal and must be spelled as shown.
- The ending semicolon (;) is required because it is outside the angle brackets for the option.

```
SYMBOL <1 . . . 99>
    <COLOR=symbol-color>
    <MODE=EXCLUDE | INCLUDE>
    <appearance-options>;
```

- SYMBOL is in uppercase because it is a SAS keyword, the name of a statement. The numbers 1 . . . 99 are in angle brackets because they are optional. The ellipsis indicates that you choose one from the range of numbers 1 through 99. The remaining elements are arguments for the statement.
- COLOR= is enclosed in angle brackets because it is an optional argument.
- *Symbol-color* is in italics because it represents a value that you specify.
- MODE= is enclosed in angle brackets because it is an optional argument.
- EXCLUDE and INCLUDE are in uppercase because they are literal values and must be spelled exactly as shown. They are separated by a vertical bar (an OR bar) because you use one or the other but not both.
- *Appearance-options* is in italics because it is a generic name for a list of options that can be used in the SYMBOL statement.

```
HBAR chart-variable< . . . chart-variable-n>
    </ <PATTERNID=BY | GROUP | MIDPOINT | SUBGROUP>
    <statistic-options>>;
```

- *Chart-variable* is italic because it is an argument that you supply. It is required because it is not in angle brackets.
- *Chart-variable-n* is enclosed in angle brackets because additional user-supplied arguments are optional. The ellipsis before the argument indicates that it can be repeated as many times as desired.
- PATTERNID= is a literal option. The values BY, GROUP, MIDPOINT, and SUBGROUP are literal values that are mutually exclusive. You can use only one, and it must be spelled as shown.
- *Statistic-options* is in italics because it is the generic name of a list of options that affect the chart statistics.

When you are using an option, a statement, or a procedure whose syntax shows arguments or values in italics, you must supply the argument or value. When the argument or value is a font, color, or variable name, SAS/GRAPH expects valid font names, color names, and variable names. Consider the following four syntax samples:

FONT=*font*

COLOR=*color*

COLOR=*text-color*

PIE *chart-variable* < . . . *chart-variable-n*>;

- *Font* must be a valid SAS font name. (See Chapter 11, “Specifying Fonts in SAS/GRAPH Programs,” on page 153 for details.)
- *Color* and *text-color* must be valid SAS/GRAPH colors. (See Chapter 12, “SAS/GRAPH Colors and Images,” on page 165 for details.)
- *Chart-variable* must be a valid SAS variable name. (See *SAS Language Reference: Dictionary* for details.)

Conventions for Examples and Output

Most of the chapters in this document include examples that illustrate some of the features of a procedure or its statements. Each example contains

- a description of the highlights of the example
- the program statements that produce the output
- the actual output from the example
- an explanation of the features of the example.

The output that is shown for the examples was generated in a Windows operating environment. If you are using a different operating environment, you might need to make some minor adjustments to the example programs.

In most cases, the output was sent to the Listing destination and generated using the default style and device for that destination. Exceptions are noted in the text.

The dimensions of the graphics output area vary across devices and when using the GRAPH windows. The dimensions can affect aspects of the graphics output – for example, the appearance of axes or the position of graphics elements that use explicit coordinates in units other than percent. You might need to adjust the dimensions of your graphics output area or the size of graphics elements to correct any differences you see. Most of the images of output in this document were generated with a GOPTIONS

statement that specified a size approximately equal 5.5 inches by 4.2 inches, although some images might be larger, if necessary, to accommodate the content of the graph.

```
goptions hsize=5.5in vsize=4.2in;
```

These HSIZE= and VSIZE= settings are not shown in the example code and are not necessary for generating the output, but you might want to use similar settings if your output looks different from the output that is shown in the document.

Most examples specify these options:

RESET=ALL	sets all graphics options to default values and cancels all global statements.
BORDER	draws a border around the graphics output area.

Information You Should Know

This section outlines information you should know before you attempt to run the examples in this document.

Support Personnel

Most sites have personnel available to help users learn to run SAS System. Record the name of your on-site SAS support personnel. Also, record the names of anyone else you regularly turn to for help with running SAS/GRAPH.

Sample Programs

The documentation for each procedure, for global statements, and for features such as the Annotate facility provide examples that demonstrate these features of SAS/GRAPH. You can copy the example code from the help or the OnlineDoc and paste it into the Program Editor in your SAS session.

These same programs are included in the sample library SAS Sample Library. 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 and Documentation. Select **Help ► SAS Help and Documentation**. On the **Contents** tab, select **Learning to Use SAS ► Sample SAS Programs ► SAS/GRAPH ► Samples**.
- In other operating environments, the SAS Sample Library might be installed in your file system. If the SAS Sample Library has been installed at your site, ask your on-site SAS support personnel where it is located.

To access the sample programs through SAS Help and Documentation or through your file system, you must understand the naming convention used for the samples. The naming convention for SAS/GRAPH samples is *Gpcxxxxx*, where *pc* is the product code and *xxxxx* is an abbreviation of the example title. The product code can be a code for a procedure, a statement, or in the case of Java and ActiveX examples, WB for "web graphs." For example, the code for the first example in the GMAP procedure chapter, Example 1 on page 1291, is stored in sample member GMPSIMPL. The sample-library member name is sometimes displayed as a footnote in the output's lower-right corner.

- In the Help system, the sample programs are organized by product. Within each product category, most of the samples are sorted by procedure. Thus, to access the

code for the first example in the GMAP procedure chapter, select **Learning to Use SAS \blacktriangleright SAS/GRAPH \blacktriangleright Samples**, scroll to **GMAP Procedure**, and select **GMPSIMPL—Producing a Simple Block Map**.

- \square In your file system, the files that contain the sample code have filenames that match the sample member names. For example, in a directory-based system, the code for sample member GMPSIMPL is located in a file named GMPSIMPL.SAS.

Note: For Java and ActiveX (web graph) samples, the naming convention is **GWBxxxxx**. \triangle

Table 1.1 Product Codes for SAS/GRAPH Procedures

Procedure	Code
dsgi	DS
ganno	AN
gareabar	AB
gbarline	BL
gchart	CH
gcontour	CT
geocode	GE
gfont	FO
ginside	IN
gkpi	KP
gmap	MP
goptions	OP
gplot	PL
gproject	PJ
gradar	RR
greduce	RD
gremove	RM
greplay	RE
gslide	SL
gtile	TL
g3d	TD
g3grid	TG

Table 1.2 Product Codes for SAS/GRAPH Statements

Statement	Code
axis	AX
by	BY
footnote	FO

Statement	Code
goptions	ON
legend	LG
note	NO
pattern	PN
symbol	SY
title	TI

Map Data Sets

To run the examples that draw maps, you need to know where the map data sets are stored on your system. Depending on your installation, the map data set might automatically be assigned a libref. Ask your on-site SAS support personnel or system administrator where the map data sets are stored for your site.

Annotate Macros Data Set

To run the examples using Annotate macros, you need to know where the Annotate macro data set is stored on your system. Depending on your installation, the Annotate macro data set might automatically be assigned a fileref. Ask your on-site SAS support personnel or system administrator where the Annotate macro data set is stored for your site.

