

Enhancements to SAS/GRAPH® Software in SAS 9.2

Himesh Patel, SAS Institute Inc., Cary, NC

ABSTRACT

This paper covers the key functionalities that have been added to SAS/GRAPH® 9.2. Highlights include integration of styles with current SAS/GRAPH procedures; new SAS/GRAPH statistical graphics procedures such as SGPlot, SGScatter, and SGPanel; Graph Template Language (GTL) that can be rendered using the SGRender procedure; GKPI and GTILE procedures; Network Visualization Workshop; new device drivers such as Scalable Vector Graphics (SVG); a new set of TrueType fonts that are shipped as part of SAS; support for more than 256 colors for the devices that support them; several new map data sets; and much more. In addition, integration of Output Delivery System (ODS) with graph procedures has been improved to automatically pick the best device and size based on the output destination.

INTRODUCTION

There are some major changes in the way defaults are set for graphics output. With SAS 9.2, ODS styles are used with almost all device types. This paper highlights the major new functionality and features with SAS/GRAPH 9.2 :

- The new SAS/GRAPH statistical graphics suite provides a new set of procedures, a new language, and a graph editor specifically designed for creating and editing statistical graphics.
- All SAS/GRAPH procedures now support ODS styles with almost all devices.
- New fonts and improved font rendering
- SAS/GRAPH now provides Image transparency and True Color support, which allows over 16 million colors in a single image.
- Support for multiple open ODS destinations
- The new GKPI procedure generates several key performance indicators.
- The new GTILE procedure generates tile charts.
- The new GINSIDE procedure determines which polygon in a map data set contains the geographic coordinates in your input data set.
- The new GEOCODE procedure allows you to estimate the latitude/longitude location of a given address.
- The new Network Visualization Workshop enables you to visualize and investigate the patterns and relationships hidden in network data (node-link data).
- All procedures now support graphics output filenames up to 256 characters long.
- Many procedures have significant enhancements and new options.
- The new Scalable Vector Graphics device driver.
- Several new map data sets, as well as new feature data sets, have been added to the MAPS library. Several existing map data sets have been updated.
- Changes in SAS/GRAPH documentation
- Reverting to a Pre-Version SAS 9.2 Appearance

SAS/GRAPH STATISTICAL GRAPHICS

ODS Statistical Graphics (ODS Graphics) is a major new piece of functionality for creating statistical graphics. It is available in a number of SAS software products, including SAS/STAT® software, SAS/ETS® software, and SAS/QC® software. Many statistical procedures have been enabled to use this functionality, and these procedures now produce graphs as automatically as they produce tables. In addition, the new statistical graphics (SG) family of SAS/GRAPH procedures uses this functionality to produce plots for exploratory data analysis and customized statistical displays. ODS Graphics includes the new SAS/GRAPH statistical graphics suite. This suite provides the following new features:

SAS/GRAPH STATISTICAL GRAPHICS PROCEDURES

The new 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 new procedures include the SGSCATTER, SGPLOT, and SGPANEL procedures. In addition, the SGRENDER procedure provides a SAS procedure interface to execute graph templates that are created with the new Graph Template Language.

The SGSCATTER procedure allows you to generate panels of scatter plots and matrices. The procedure has three action statements, one, and only one, of which must be specified with the procedure:

- PLOT creates a panel of independent scatter plots. The procedure also supports overlay of Loess, Regression, or Spline fit lines as well as confidence or prediction ellipses. Axes always appear on each plot.
- COMPARE creates a rectangular (MxN) scatter-plot matrix that compares one set of variables against another set of variables, Loess, Regression, or penalized B-spline fit lines as well as confidence or prediction ellipses can be overlaid on each plot. Axes always appear external to the plots.
- MATRIX creates a symmetric (NxN) scatter-plot matrix from a list of variables. Confidence or prediction ellipses can be overlaid on the plots. By default, plot variables are labeled in the matrix diagonal, but you can use DIAGONAL option to add plots in the matrix diagonal. Axes always appear external to the plots.

Figure 1 shows how you can use the matrix statement with the diagonal option to display a histogram with kernel density plot.

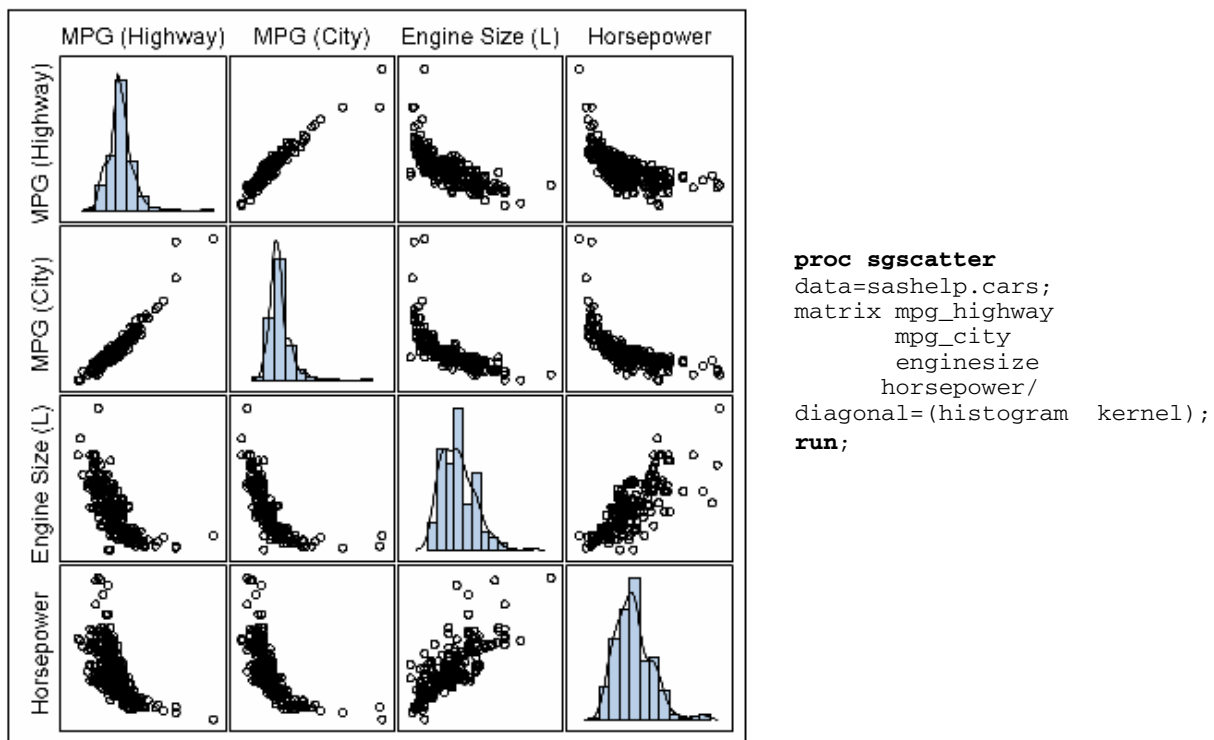
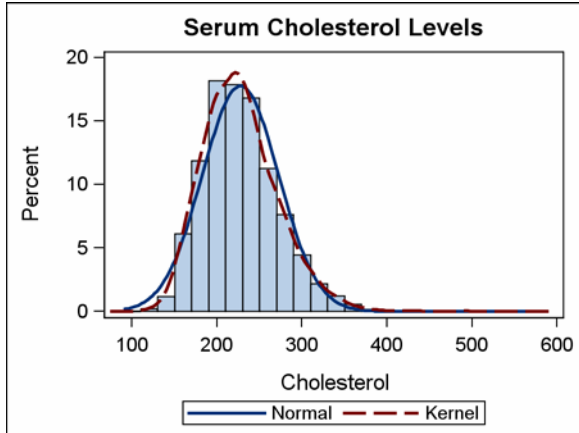


Figure 1: Scatter-plot matrix

The SGPLOT procedure allows you to generate single-celled graphs with overlay capabilities. This plot consists of multiple plot elements inside an overlay container with a common set of axes. Various plot types are supported including scatter plots, box plots, loess fits, histograms, and more. The SGPLOT procedure also supports other plot elements, such as reference lines, legends, and insets. Figure 2 shows how you can overlay each statement to generate a more complex plot.



```

title "Serum Cholesterol Levels";
proc sgplot data=sashelp.heart;
  histogram cholesterol;
  density cholesterol;
  density cholesterol / type=kernel;
run;

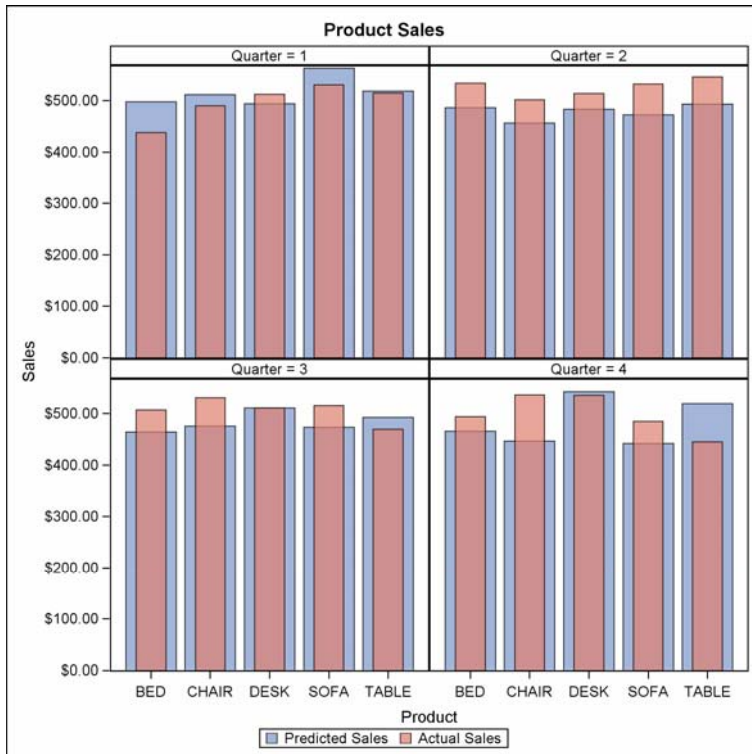
```

Figure 2: Histogram with normal and kernel density overlays

The SG PANEL procedure creates paneled plots with multiple classification variables. The defined plot composite is drawn for each crossing of the classification variables, showing the subset of the data in each cell. The procedure also supports easy paging of the plot in the case of a large number of levels for the class variables. As with SG PLOT, the SG PANEL procedure allows you to combine a variety of plot and chart types into more elaborate overlays. The plot types supported by SG PANEL are listed below:

- Basic plot (scatter, series, step, band, and needle)
- Fits and Confidence (loess, regression, penalized B-spline)
- Distribution (horizontal and vertical box plots, histograms, normal curves, and kernel-density estimates)
- Categorization (dot plots, horizontal and vertical bar charts, horizontal and vertical line charts)

Figure 3 shows an example of generating a graph of *panelby quarter* with an overlay of two vertical bars with a transparency to show actual versus predicted values.



```

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;

```

Figure 3: Panel by quarter of predicted sales versus actual sales

GRAPH TEMPLATE LANGUAGE (GTL)

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 GTL to either modify these templates or to create your own customized graphs. GTL also enables you to create sophisticated analytical graphics that are not available from traditional SAS/GRAPH procedure statements.

ODS graphics templates are defined with PROC TEMPLATE using GTL. The GTL includes control statements that control program flow; layout statements that specify the arrangement of graph features; plot statements that request specific plot types (such as histograms and scatter plots); and text and legend statements that specify titles, footnotes, legends, and other text-based graph elements. Typically, the GTL templates are rendered using the SGRENDER procedure, which specifies a data source that contains appropriate data values and the template to use for rendering the graph.

Figure 4 shows an example of a stock chart that was generated with a single column lattice layout with a common date axis. Several plots were overlaid to generate the top and bottom graphs in the lattice.

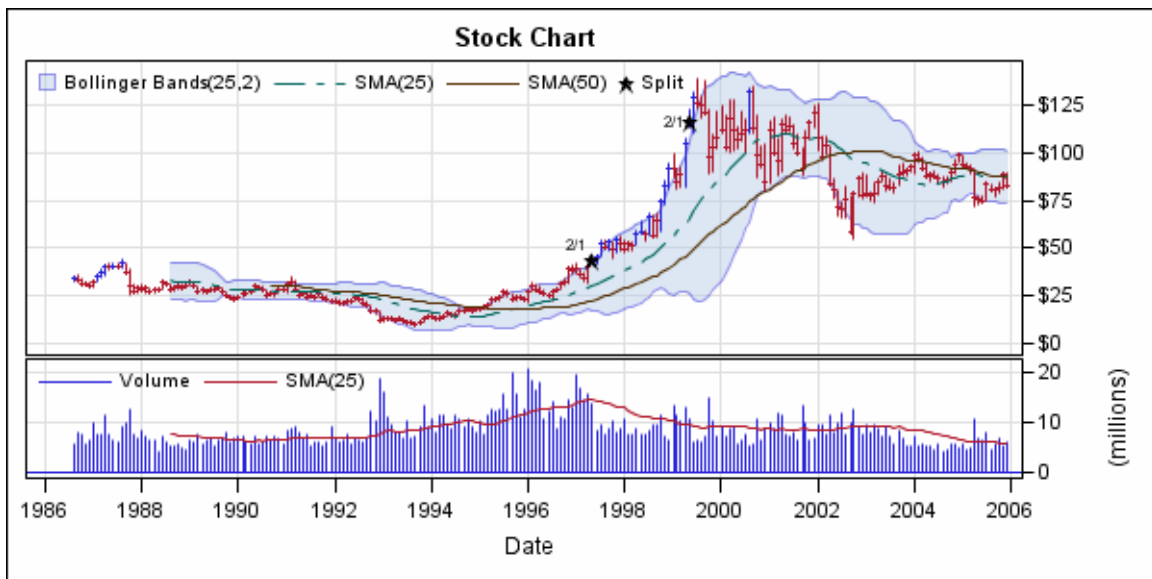


Figure 4: Stock chart for IBM from 1986-2006

```
proc template;
define statgraph stockplot;
begingraph;
Entrytitle "Stock Chart";
layout lattice / columns=1 columndatarange=union rowweights=(0.7 0.3);
columnaxes;
columnaxis / offsetmin=0.02 griddisplay=on;
endcolumnaxes;
layout overlay / cycleattrs=true yaxisopts=(griddisplay=on label=" "
display=(line) displaysecondary=all);
bandplot x=date limitupper=bolupper limitlower=follower / datatransparency=0.5
display=(fill outline) outlineattrs=(pattern=solid)
fillattrs=graphconfidence name="Bollinger"
legendlabel="Bollinger Bands (25,2)";
vectorplot xorigin=date y=low yorigin=high x=date / arrowheads=false
group=cindex index=cindex
lineattrs=(pattern=solid thickness=1px) shaftprotected=true;
scatterplot x=date y=close / group=cindex index=cindex
markerattrs=(symbol=plus size=3);
scatterplot x=date y=close / freq=freq markerattrs=(symbol=starfilled size=7)
datalabel=split name="split" legendlabel="Split"
markerattrs=graphdatadefault;
seriesplot x=date y=move25avg / lineattrs=(thickness=1px) legendlabel="SMA (25)"
name="day25";
```

```

seriesplot x=date y=move50avg / lineattrs=(pattern=solid thickness=1px)
      legendlabel="SMA(50)" name="day50";
discretelegend "boilinger" "day25" "day50" "split" / across=4 border=off
      valign=top halign=left location=inside;
endlayout;

layout overlay / yaxisopts=(griddisplay=on display=(line) displaysecondary=all)
      xaxisopts=(griddisplay=on cycleattrs=true ;
needleplot x=date y=volume / index=vindex name="volume" legendlabel="Volume"
      lineattrs=(pattern=solid thickness=1px);
seriesplot x=date y=vmove25avg / name="vmoveavg" legendlabel="SMA(25)"
      lineattrs=(pattern=solid thickness=1px);
discretelegend "volume" "vmoveavg" / location=inside border=off halign=left
valign=top;
endlayout;
endlayout;
endgraph;
end;
run;
ods listing image_dpi=100;
ods graphics / reset width=600 px height=300 px imagename='Stocktest';
proc sgrender data=moveavg template=stockplot; run;

```

ODS GRAPHICS EDITOR

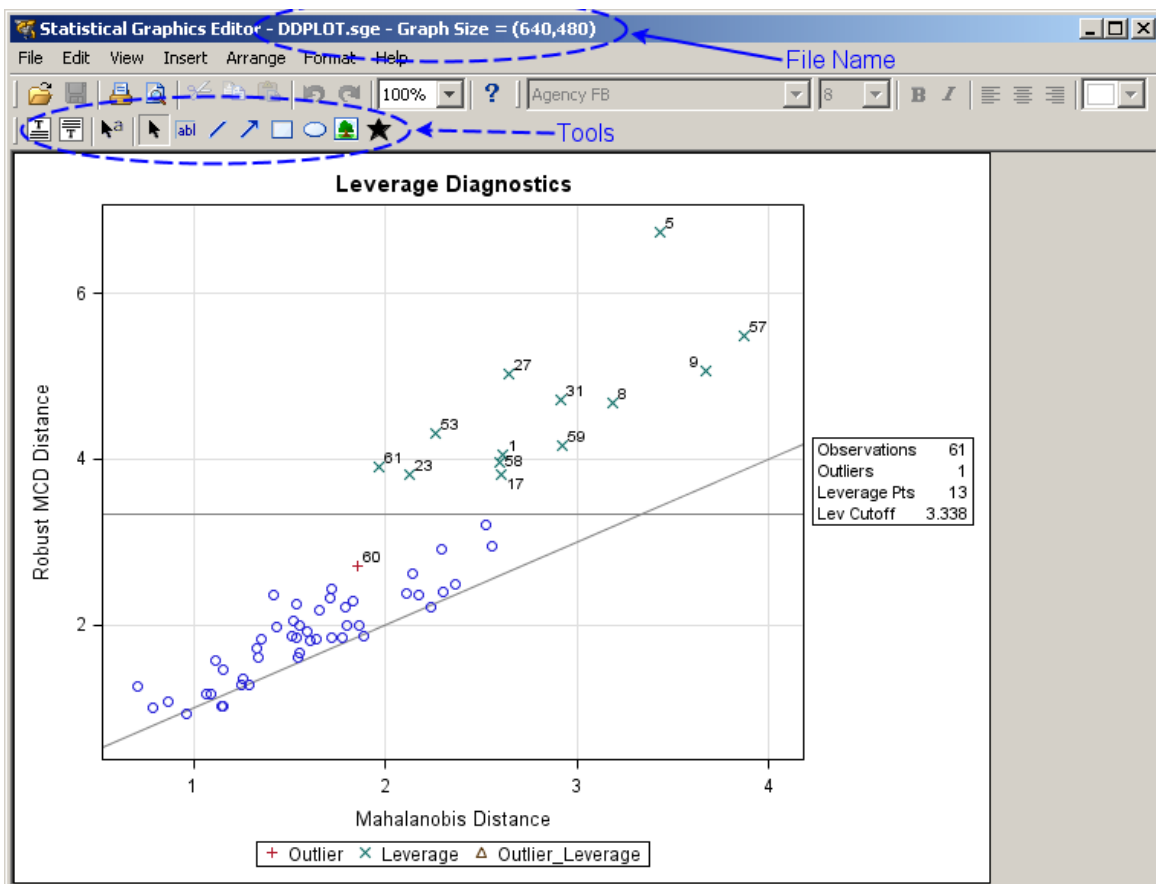


Figure 5: ODS Graphics Editor Window

To customize the ODS graphics output, you typically need to edit the associated template or program, and resubmit it to get the customized output. It can be a tedious task to make minor changes to the graphs. The ODS Graphics editor provides an interactive GUI that makes such changes easy. You can use the GUI to change the graph style and size.

You can edit, add, or delete titles and footnotes; reposition the legends; and customize the visual properties of the plots and axes. Furthermore, you can add free-form annotation to the graph to emphasize the relevant aspects of the results. You can directly copy the customized graph to the system clipboard to include in other documents, or you can save the graph for future use.

To edit a graph that has been generated using ODS graphics, highlight the results window. Then submit **sgedit on** from the command line to enable the creation of editable ODS graphs. ODS graphs that are created after submission of this command can be edited. You can now submit your code and then go to Results Viewer and right-click on the graph node. Select the edit command to bring up the editor. Note that you cannot edit ODS graphs that were created by programs that were submitted before issuing **sgedit on**.

The ODS Graphics Editor is available to SAS users as a free download for use on workstations where SAS is not licensed. When graphical output has to be included in publications and presentations, the analysts or statisticians can make the output available to the publications staff, who can then customize the output using the stand-alone ODS Graphics Editor.

SUPPORT FOR ODS STYLES

By default, all SAS/GRAPH procedures and devices now support ODS styles. All colors, fonts, and additional information are derived from the current style. You can use procedure statement options and SAS/GRAPH GOPTIONS to override individual elements of the style and customize the appearance of any graph. Additionally, the colors used by the styles have been updated to enhance the appearance of your graphics output. By default, ODS styles are active. Figure 6 compares a graph created in SAS 9.1.3 and a graph created in SAS 9.2, both of which were generated using the PNG device driver with default settings.

By default, the attributes of various elements of the graph are derived from specific style elements within the style unless you explicitly override them with procedure or global statement options in the SAS/GRAPH program. For example, you can use the CTITLE= and CTEXT= options on the GOPTIONS statement to change the color of the text in all of your graphs, or you can use the SYMBOL statement to specify colors for markers. Instead of specifying global options, which affect all of your SAS/GRAPH output, you can specify options on specific action statements that will affect only the output produced by that statement. Goptions and/or procedure options that control colors and fonts override any graph style defaults. The relationship between graph options and style attributes is documented under each procedure. For backward compatibility, see the section “Reverting to a SAS 9.1.3. or Earlier Appearance of SAS/GRAPH.” Here are some of the major differences that you will notice:

- Before SAS 9.2, the default font was non-scalable. In SAS 9.2 the default font is Albany AMT (which is a TrueType font and is shipped with SAS). For more information on fonts, please see Fonts and Font Rendering.
- In addition to default font change, the font-rendering technique in SAS 9.2 was changed from Host-font rendering to Freetype-font rendering for all devices except display type devices. Freetype-font rendering now allows you to render font size that is the same as the Microsoft font size.
- Before SAS 9.2, the default device for html destination was GIF. In SAS 9.2, the default device is PNG.
- Starting with SAS 9.2 you do not need to specify a device when you use ODS destinations. ODS assigns a default device that is appropriate for the open destination. You can modify the default device by specifying a specific device using GOPTIONS, or by changing it in the SAS registry.

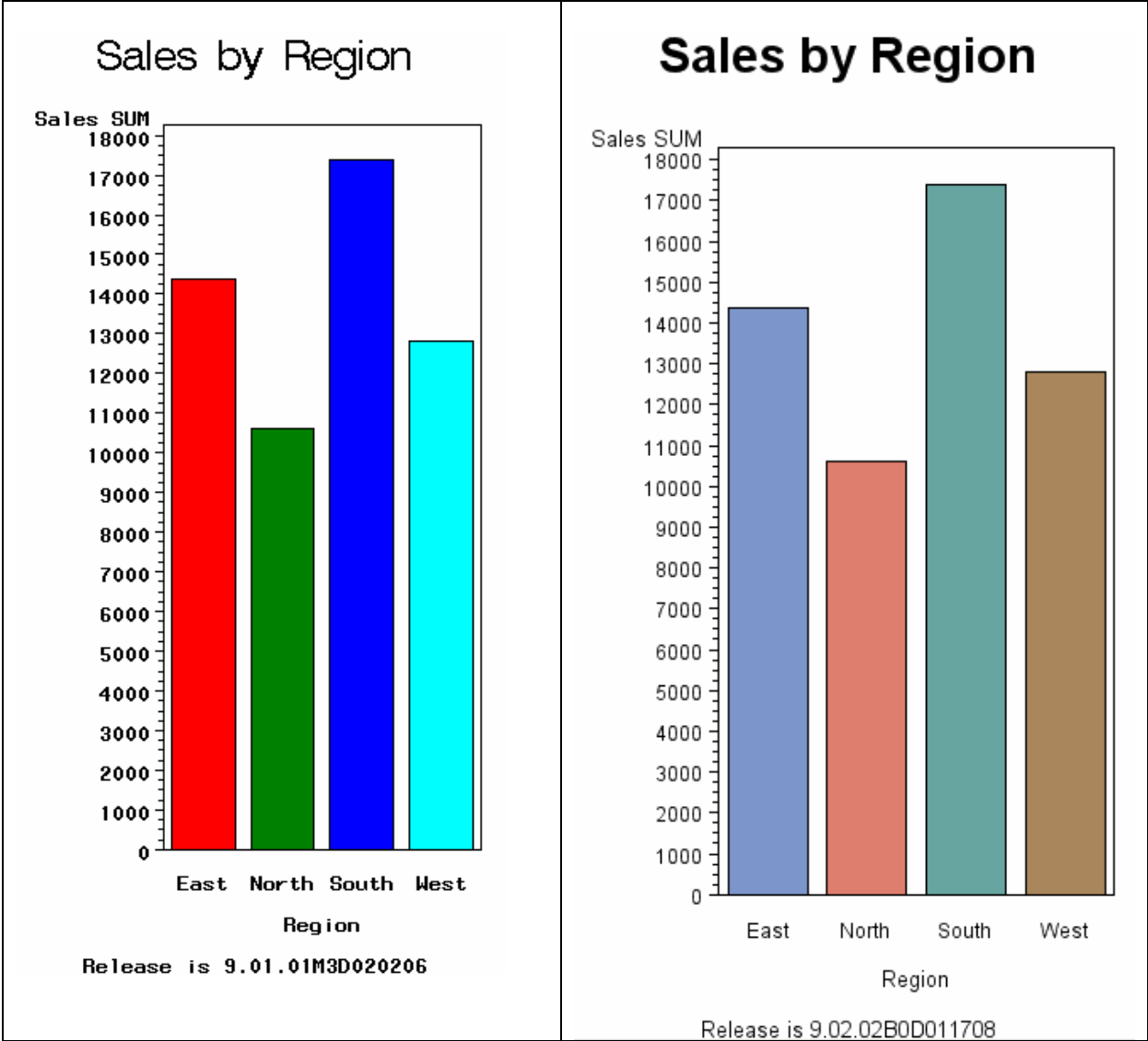
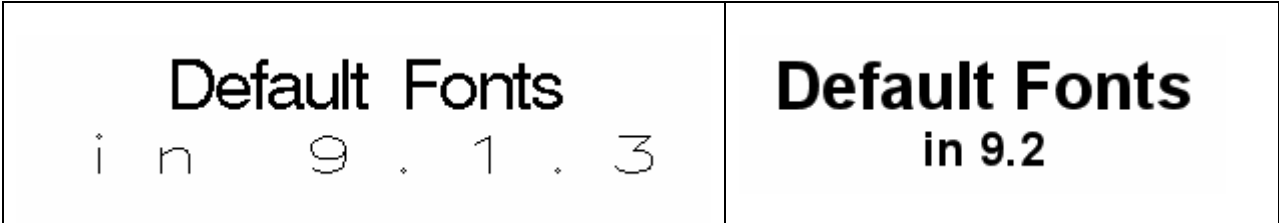


Figure 6: Comparison of default output between SAS 9.1.3 and SAS 9.2

NEW FONTS AND IMPROVED FONT RENDERING

The following fonts are now obsolete: DAVID, NHIRA, NKATA. Some of the characters in the Hebrew font are mapped differently to the Roman character set. Fonts are now rendered using the FreeType engine with most commonly used devices, such as image-based devices (gif, png, and so on), postscript, pdf, pcl, and svg. This new font rendering might result in fonts appearing larger than they did in previous versions of SAS/GRAPH. Below is the output from SAS 9.1.3 and SAS 9.2.



Many new TrueType fonts have been added. SAS/GRAPH now uses Albany AMT as the default font. Default simfont has also been changed to Albany AMT. The following fonts are now installed as part of Base SAS®:

Albany AMT*	Thorndale Duospace WT SC	GungsuhChe
Cumberland AMT*	Thorndale Duospace WT TC	Dotum
Thorndale AMT*	Arial Symbol	DotumChe
Symbol MT*	Times New Roman Symbol	Gulim
Monotype Sorts	MS PMincho	GulimChe
Monotype Sans WT J	MS Mincho	NSimSun
Monotype Sans WT K	MS PGothic	SimHei
Monotype Sans WT SC	MS UI Gothic	SimSun
Monotype Sans WT TC	Batang	PMingLiU
Thorndale Duospace WT J	BatangChe	MingLiU
Thorndale Duospace WT K	Gungsuh	HeiT

* Albany AMT, Cumberland AMT, Thorndale AMT, and Symbol MT are font families. Normal, bold, italic, and bold italic versions of these fonts are provided.

IMAGE TRANSPARENCY AND TRUE COLOR SUPPORT

IMAGE TRANSPARENCY

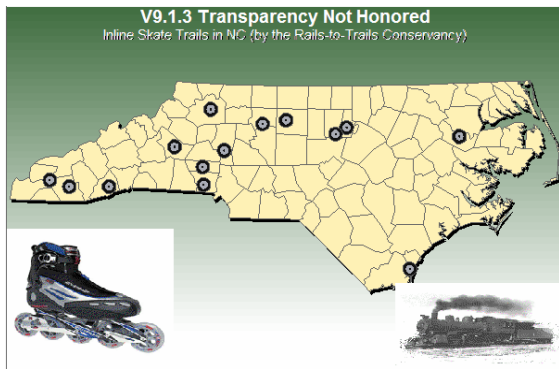


Figure 7.1

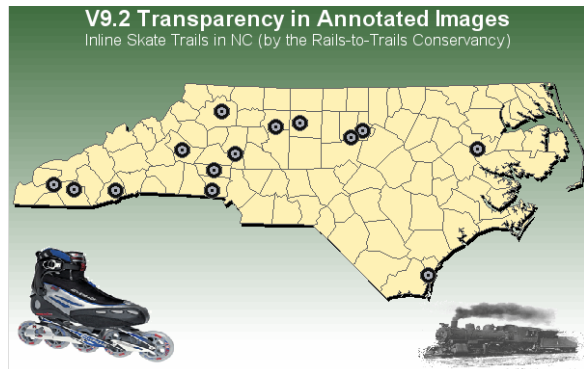


Figure 7.2

Before SAS 9.2, transparency worked only when you used an image that was generated using the GIF device driver with GOPTIONS TRANSPARENCY. If the overlaid image was not generated with SAS/GRAPH, then the transparent background was not honored. In SAS 9.2, you can overlay any image (graph or a picture) with a transparent background on a graph. Figure 7.1 was generated using SAS 9.1.3. Notice how the background is covering up the graph area. Figure 7.2 was generated using SAS 9.2. Now the picture blends into the graph area, and you don't see the white background of the images.

TRUE COLOR SUPPORT

Before SAS 9.2, SAS/GRAPH device drivers were limited to 256 colors. In SAS 9.2, SAS/GRAPH supports all of the colors that a device or graphics format is capable of supporting. For example, the GIF graphics format has a limit of 256 colors, so the SAS/GRAPH GIF device driver would never support more than 256 colors. With SAS/GRAPH 9.2,

this 256-color limit has been lifted for those devices and graphics formats that support more than 256 colors. Export device drivers (such as PNG and SVG) and display device drivers (such as WIN and XCOLOR) now support millions of colors. Figure 8.1 and 8.2 compare the output using the SAS 9.1.3 PNG device driver and the SAS 9.2 PNG device driver.

V9.1.3 256 Colors

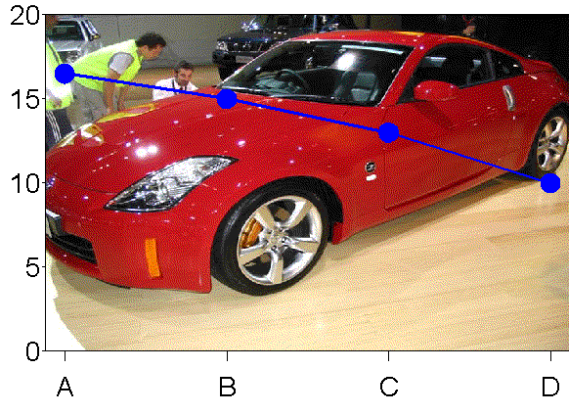


Figure 8.1

V9.2 >256 Colors

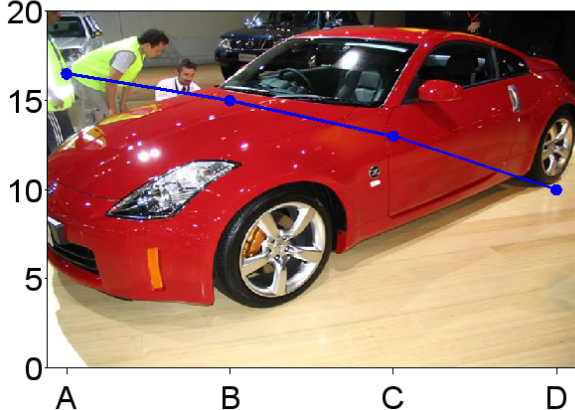


Figure 8.2

Figure 8.1 shows the output from SAS 9.1.3. Note that the background image is dithered due to color limitations. Figure 8.2 was generated using the same image as the background using SAS 9.2. Color mapping is also enhanced in SAS 9.2. Before SAS 9.2, when SAS/GRAPH procedure ran out of colors or encountered unknown colors, it mapped to the second color (first is the background color) in the color list. In SAS 9.2 color mapping is done based on RGB values and is mapped to the closest color that is available.

SUPPORT FOR MULTIPLE OPEN ODS DESTINATIONS

If you have multiple ODS destinations open, then SAS/GRAPH selects the appropriate device for each destination automatically. In addition, each graph uses the ODS style associated with each destination. You do not need to specify a device or style to get optimal results. For example, if you do not specify a device, then SAS/GRAPH automatically selects the PNG device for the HTML destination, if it is open, and the SASEMF device for the RTF destination.

Also, if you have multiple ODS destinations open and you are using a device other than the Java or ActiveX devices (ACTIVEX, JAVA, ACTXIMG, or JAVAIMG), a different GRSEG is created for each open destination. The GRSEGS for the first destination are stored in WORK.GSEG. The GRSEGS for any other open destinations are stored in catalogs named according to the destinations, for example, WORK.HTML.

GKPI PROCEDURE

The GKPI procedure creates graphical Key Performance Indicator (KPI) charts. KPIs are metrics that help a business monitor performance and measure progress toward specific goals. The procedure produces five KPI chart types: slider, bullet graph, dial, speedometer, and traffic light.

The GKPI procedure produces a two or three-dimensional KPI chart that is based on a series of segment boundaries and an actual KPI value that you specify. If you specify a target value, the KPI chart also displays the target value. The procedure uses a set of default colors for the KPI chart, but you can specify your own colors. Note the only device driver supported by the GKPI procedure is JAVAIMG. To use output from the GKPI procedure in a dashboard generated with the GREPLAY procedure, you must first create a GRSEG containing the GKPI procedure output. You can use the IBACK="gkpimage.png" option on the GOPTIONS statement with the GSLIDE or GANNO procedures to generate the GRSEG. Alternatively, you can use "ODS tagsets.htmlpanel" to include output from PROC GKPI without using the GREPLAY procedure.

Slider KPI charts display a bar divided into segments according to the boundary values that you specify. The actual value of the KPI is indicated with a triangle pointer on the top (for horizontal slider) or the left (for a vertical slider). This

actual value indicator is the same color as the segment that contains the actual KPI value. The target value, if it is specified, is displayed as a smaller triangle on the bottom (or right side) of the slider.

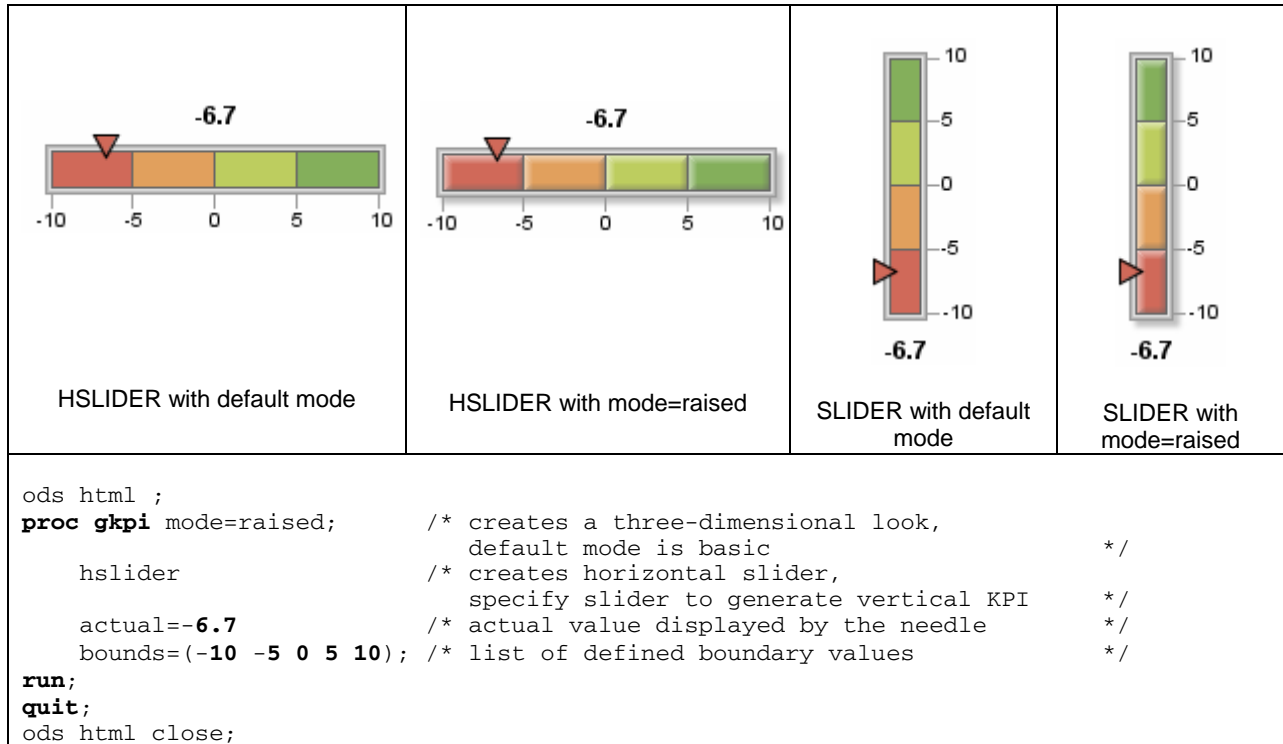
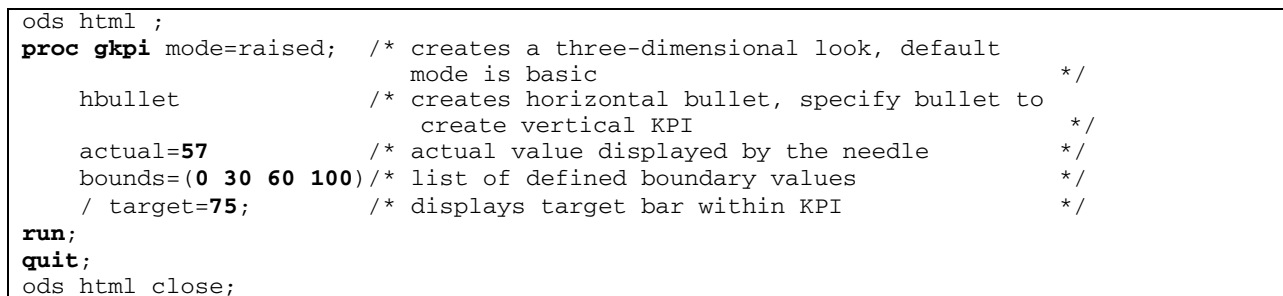


Figure 9: SLIDER KPI showing actual value of -6.7

Bullet KPI Charts display a bar divided into segments according to the boundary values that you specify. The actual value of the KPI is indicated with a black line, or bullet, down the center of the graph. The target value, if it is specified, is displayed as a vertical line (in a horizontal bullet graph) or a horizontal line (in a vertical bullet graph) across the graph.



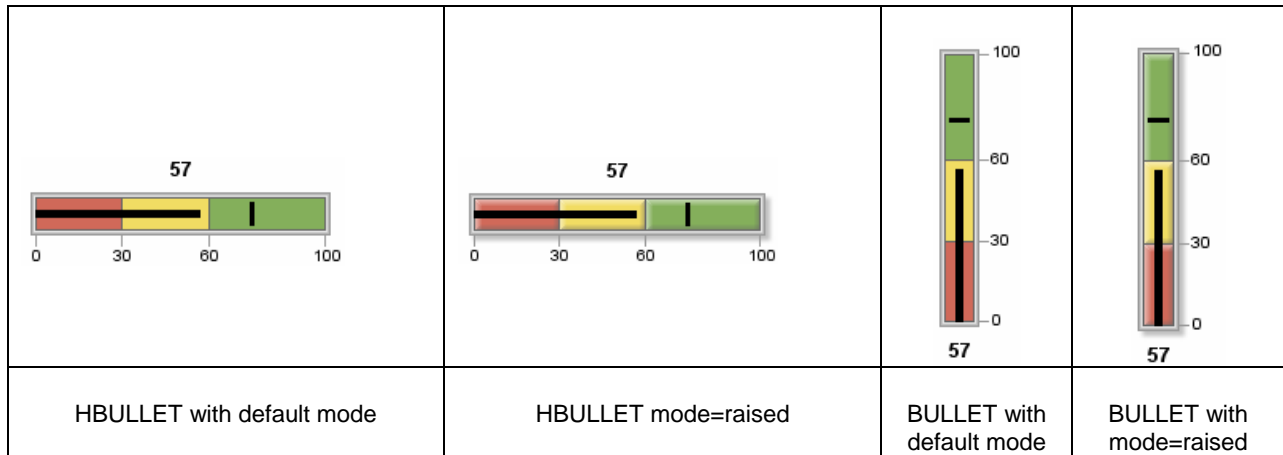


Figure 10: BULLET KPI, showing target and actual value of 57

Dial KPI Charts display a dial divided into segments according to the boundary values. The actual value of the KPI is indicated with a large, white, triangle pointer. The target value, if it is specified, is displayed as a small, black triangle. The center of the dial is the same color as the segment that contains the actual KPI value.

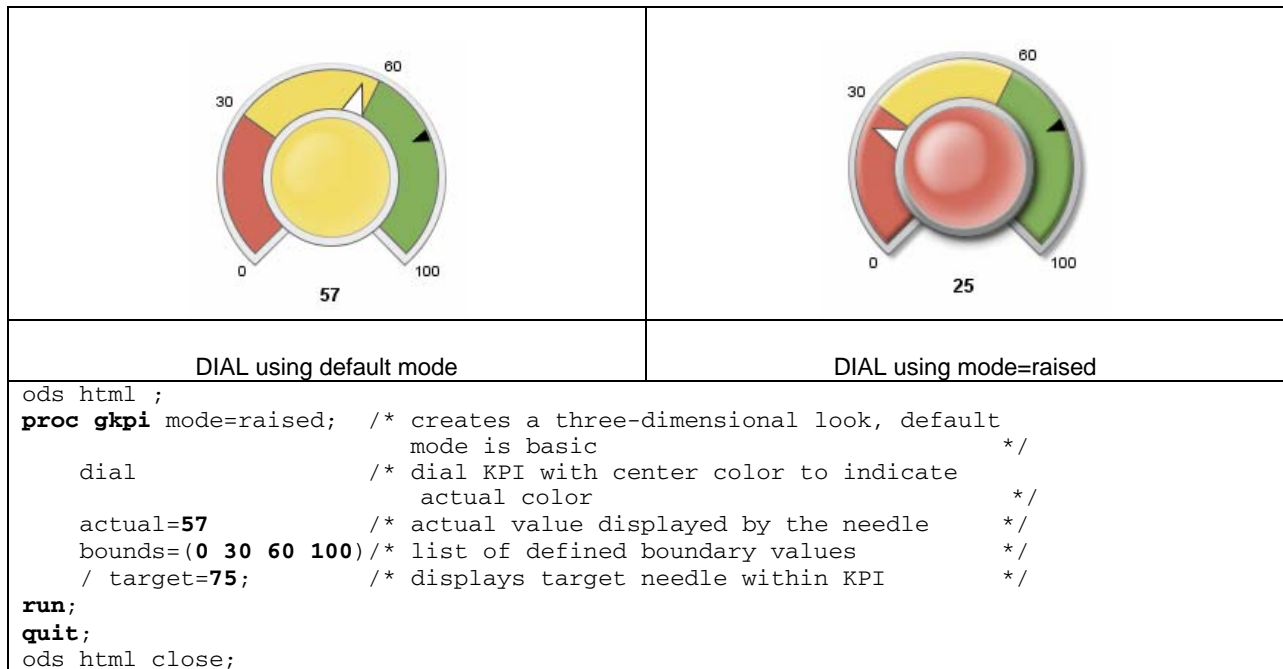
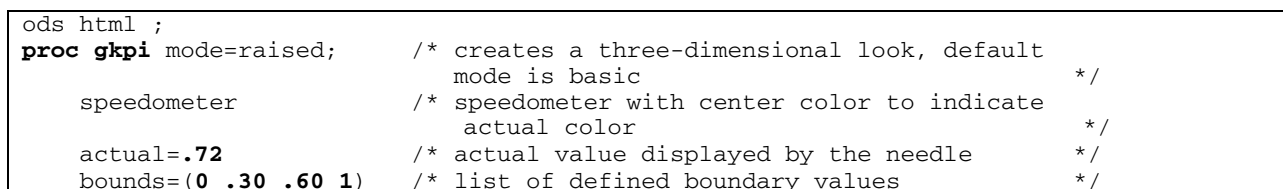


Figure 11: DIAL KPI showing target and actual values

Speedometer KPI Charts display a speedometer with the tick marks evenly spaced around the dial and colored segments that correspond to the segment boundaries that you specify. Speedometers can be displayed as a full speedometer, as a half speedometer, or as a quarter speedometer. The actual value of the KPI is indicated by a long pointer. The target value, if it is specified, is displayed as a small, black triangle.



```

/ target=.85          /* displays target needle within KPI          */
  type=half          /* valid values are full(default)/half/quarter*/
  format="percent8.0"; /* format range and actual values in percent */
quit;
ODS HTML CLOSE;

```

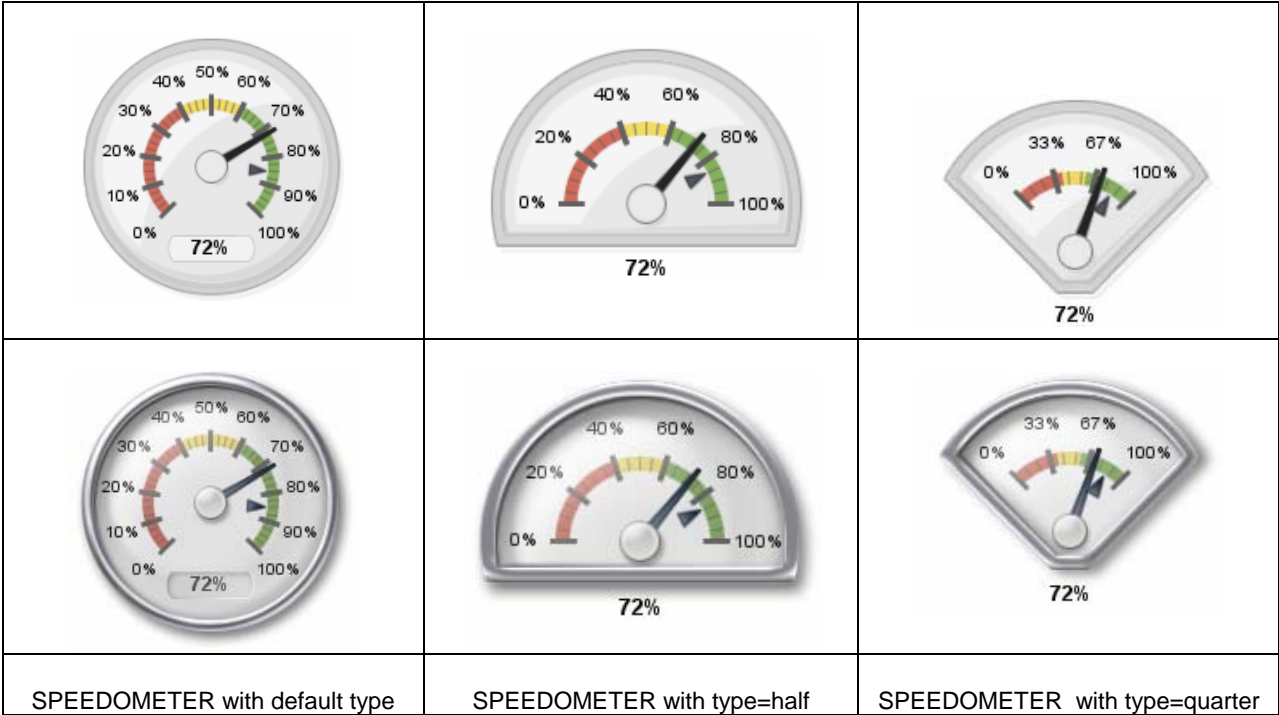
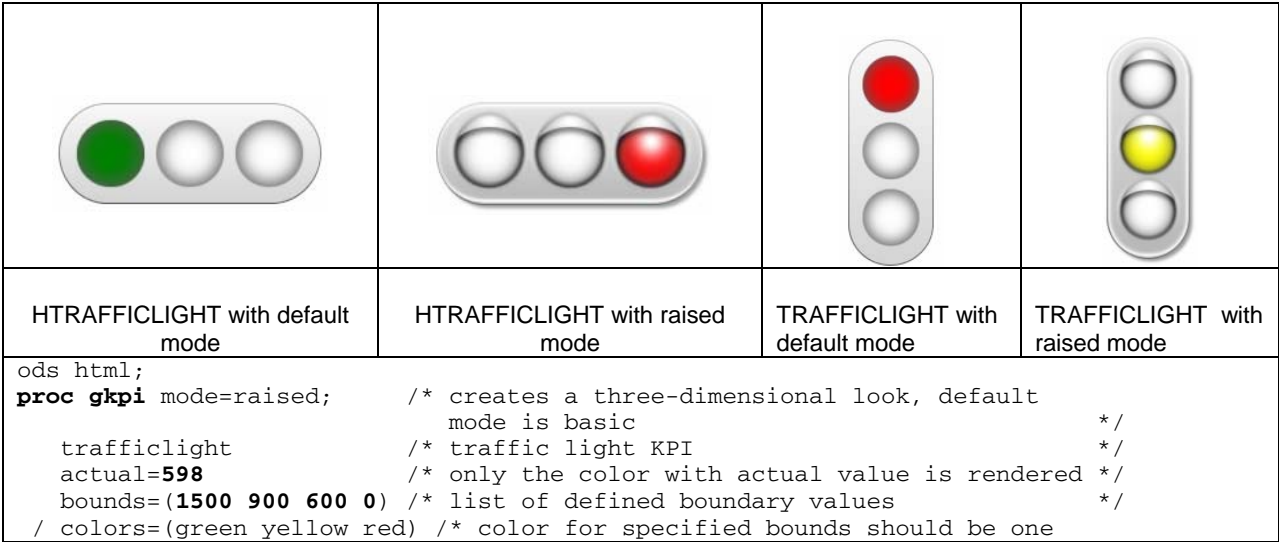


Figure 12: Speedometer displaying values using percent format

Traffic Light KPI charts display a traffic light that contains one light for each segment. The segment that contains the actual value is displayed in color. The remaining segments are gray. In other words, only one "light" is "turned on" at a time. Traffic lights do not display target values.



```

noavalue;                               Less than bounds                               */
quit;                                    /* do not display actual value                    */
ods html close;

```

Figure 13: Traffic light using different actual values to show the overall status

DASHBOARDS USING KPI

Dashboards provide an easy view of all your metrics. These metrics can be collected from various different applications or data. KPI is one of the key components of the dashboard that gives you a quick view of your company's status. At a single glance, you can decide either to do nothing or to drill further into the information to find out why it's below par.

Figure 14 is an overview of revenue income from different sources for a make-shift magazine company. This example demonstrates how the output from KPIs can be used to display a dashboard. Because the procedure generates png files only, you have to use the annotate IMAGE function with proc gslide to create a grseg. The GREPLAY procedure was used to do layout and display the output from gslide.

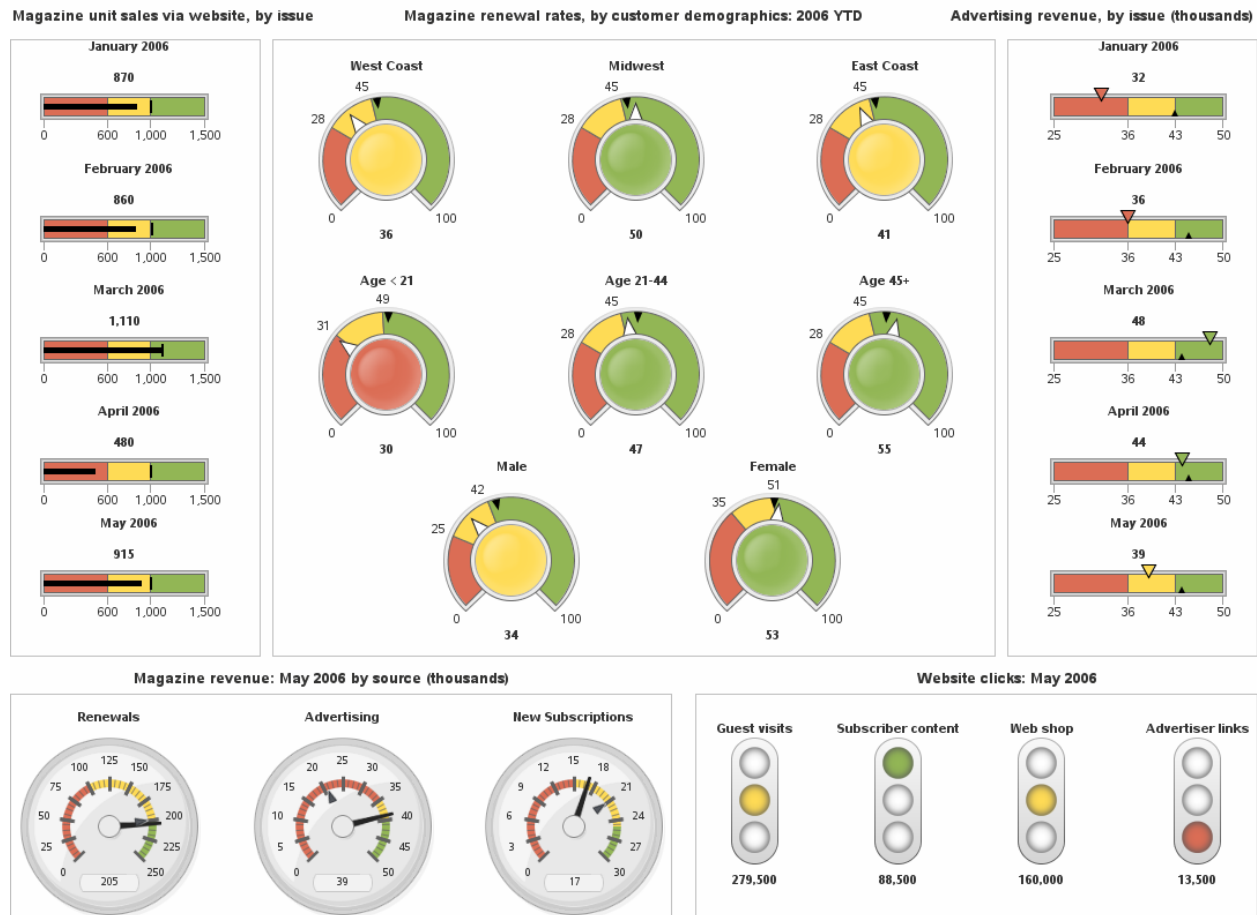


Figure 14: Magazine revenues for 2006

GTILE PROCEDURE

The GTILE procedure produces charts that consist of a rectangle or square that is divided into tile-shaped segments. These charts graphically represent the relative sizes of tiles to one another, and to the whole. The TILEBY keyword is followed by any number of numeric or character variables. A separate tile as well as a separate level is created for

each of these variables. Each level contains a subset of tiles, whose size represents the sum of the layout variable's values for that category. You can assign an additional numeric variable as the color variable using the COLORVAR= option.

A three-color gradient legend provides a key to the value of the colors plotted on the chart. The legend also provides the name of the variable that is used to color the tiles. The legend displays the variable's minimum, maximum, and midpoint value.

By providing multiple TILEBY variables and specifying the DEVICE=Java option or the DEVICE=ActiveX option, the GTILE procedure creates interactive charts. These charts enable you to display subsets (or levels) of your data.

The GTILE procedure provides three different layouts to visualize your data: FLOW, TILE, and TOGGLE.

When you use the flow layout, the data order is honored. Sorting your data can enhance the readability of this layout. This chart is read from left to right and from top to bottom. The flow layout can give the appearance of several aligned rows, depending on the amount of data. This layout is particularly useful for subsetting large amounts of data. Figure 15 shows an example of flow layout after subsetting data to compare usage that is based on 1 or 8 CPU.

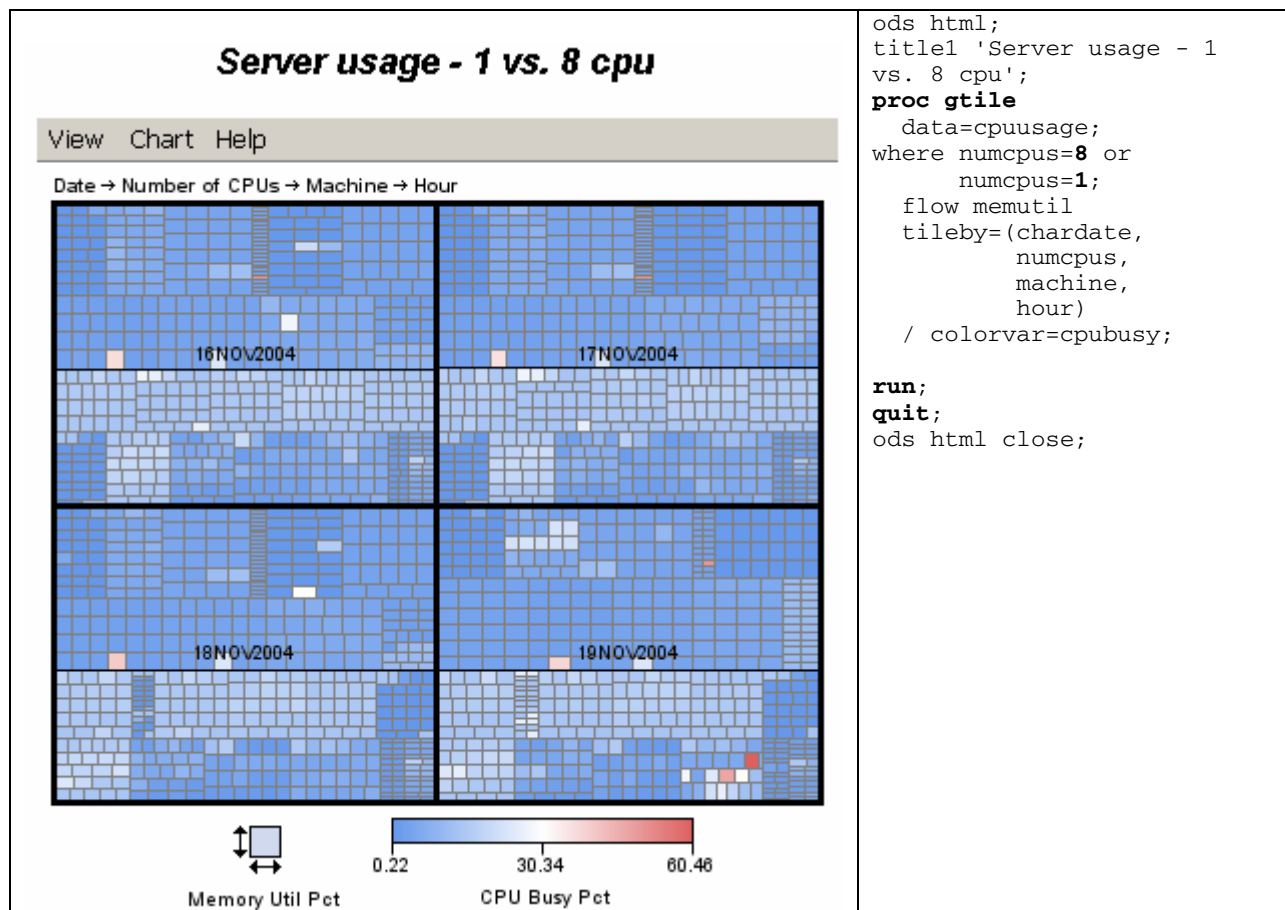


Figure 15: Server usage

When you use the tile layout, data order is not honored. With this layout each tile appears as close to a square as possible. The tiles are laid out starting with the largest value in the lower left corner. The smallest values can typically be found in the top right corner. This layout works well when comparing data extremes. Figure 16 is using a tile layout; as you can see, Middle East has larger square indicating the most returned products and higher sales, while Asia is in the top right with the least amount of returns and lower sales.

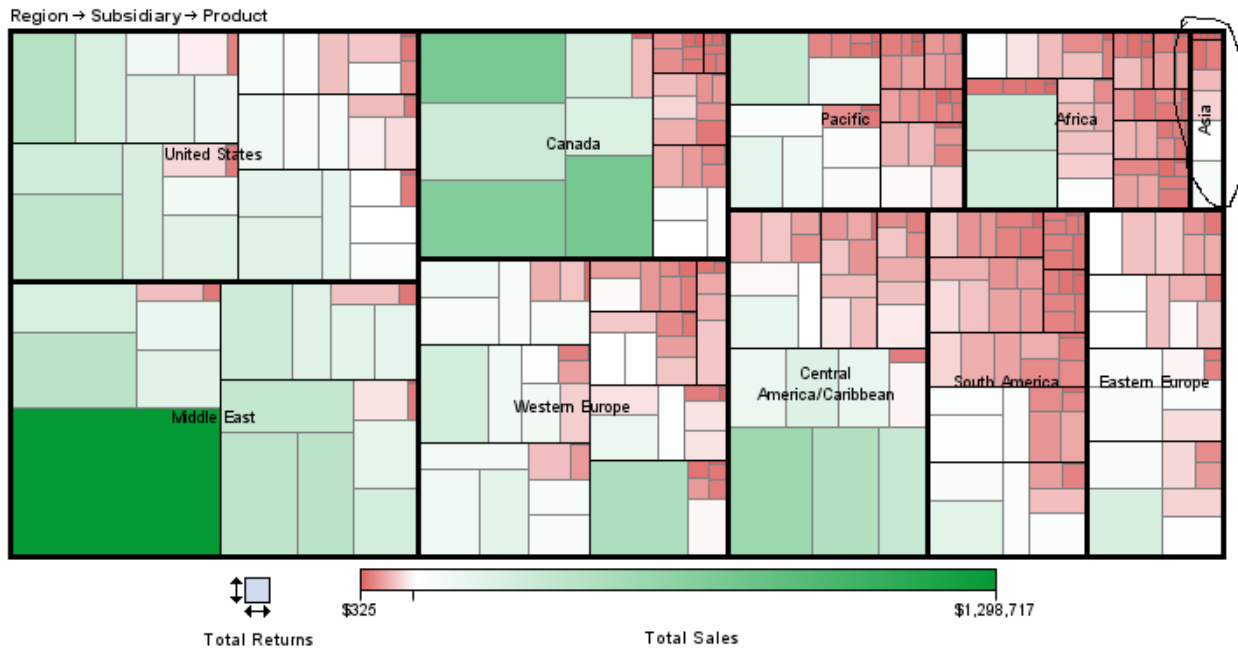


Figure 16: Tile layout

When you use the toggle layout, data order is honored. Sorting your data can enhance the readability of this layout. This layout is read from left to right. Each time you change levels, the data is reorganized. If the chart is displayed in a one-row layout, changing levels toggles the layout to a one-column display. If the chart is displayed in one-column layout, changing levels toggles the layout to a one-row display. This presentation works well for smaller quantities of data.

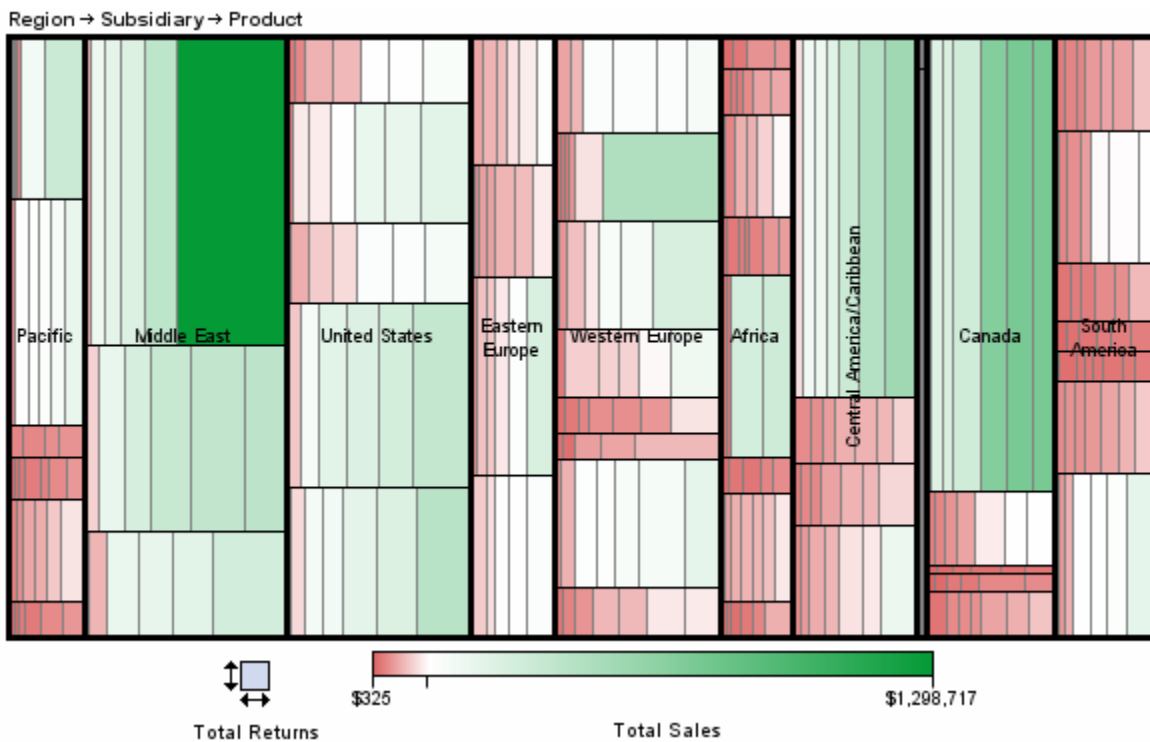


Figure 17: Toggle layout

GINSIDE PROCEDURE

The GINSIDE procedure compares a data set of X and Y coordinates to a map data set containing map polygons. The procedure determines whether the X and Y coordinates for each point fall inside or outside the map polygons. If the point falls inside of a polygon, then the ID variable is set to the ID value of that polygon. For example, if a map contains states, then the ID variable of the output data set is set to the state that contains the point. The GINSIDE procedure can be used with the SAS/GRAPH map data sets, and the results can then be used to annotate onto a map generated using the GMAP procedure.

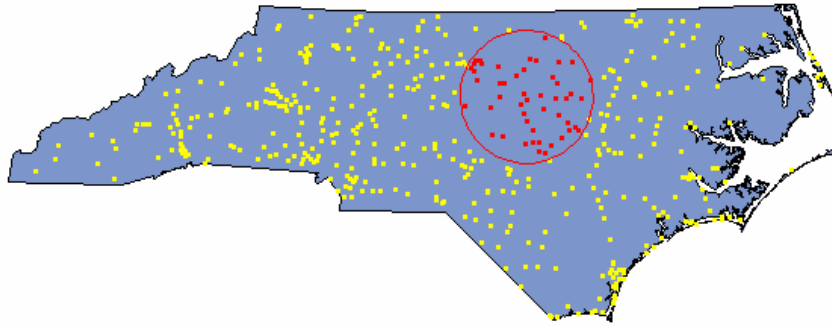


Figure 18: Customers within 40-mile radius

GEOCODE PROCEDURE

Geocoding is the process of adding geographic coordinates (latitude and longitude values) to an address. The coordinates typically represent the center of a ZIP code, a city, an address, or any geographic region. After geocoding, you can use the coordinates to display a point on a map or to calculate distances.

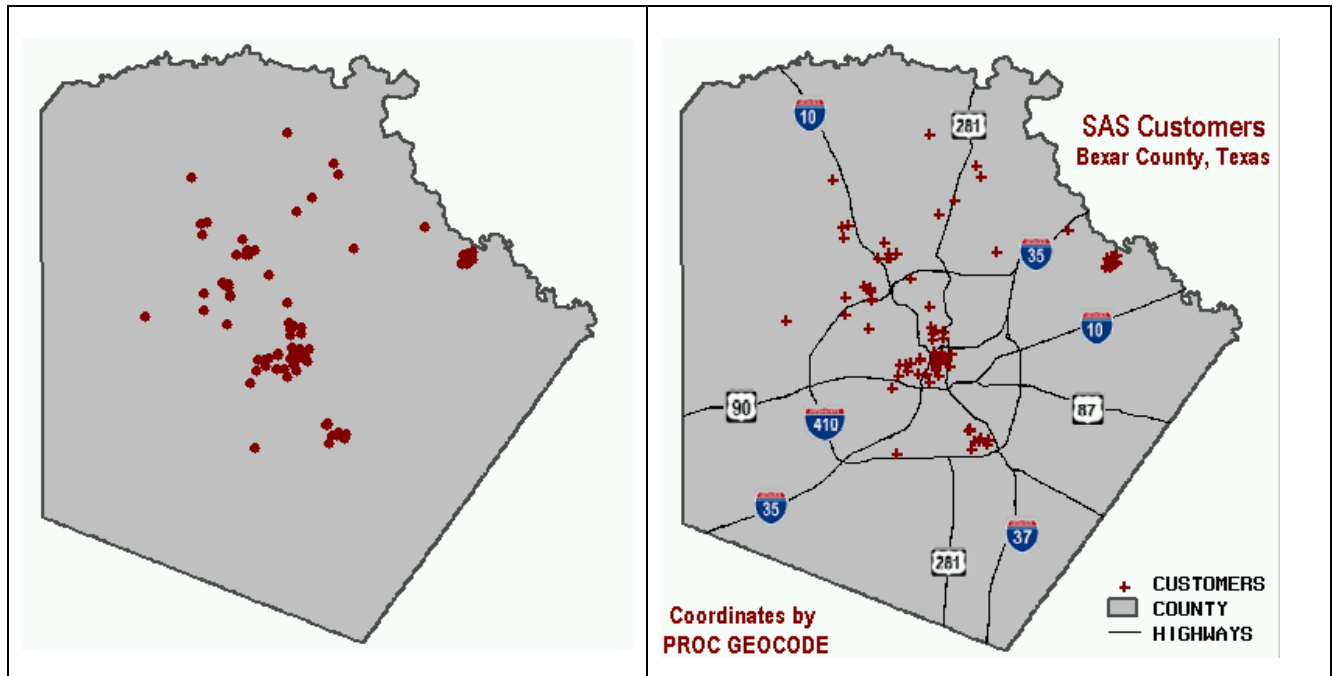


Figure 19.1: SAS customers in Bexar County

Figure 19.2

Figure 19.1 was generated by taking output from PROC GEOCODE and displaying those points using annotate with PROC GMAP. Figure 19.2 was generated using output from PROC GEOCODE and displaying that as a layer using SAS/GIS®.

NETWORK VISUALIZATION WORKSHOP

SAS/GRAPH Network Visualization Workshop is an interactive, graphics-oriented application for visualizing and investigating data. The application uses visualization techniques that enable you to detect patterns and extract information that might be concealed, often in very large quantities of data. SAS/GRAPH Network Visualization Workshop is particularly useful for examining network data, that is, data that is structured into nodes and links that connect the nodes.

STEPS FOR VISUALIZING NETWORK DATA

SAS/GRAPH Network Visualization Workshop assumes that the network to be investigated is described by two data sets:

- One data set contains information about the nodes in the network.
- A second data set contains information about the links that connect the nodes.

If only link data is available, you can generate a simple node data set from the link data.

There are multiple ways to investigate network data, and each approach can vary from what is described here. In general, a typical session might consist of the following steps:

1. Load node and link data into SAS/GRAPH Network Visualization Workshop.
2. Use the Edit Data Attributes dialog box to identify the FROM and TO variables in the link data set and the NODE variable in the node data set. When you set a data attribute, you can also specify which variables to use for link colors as well as for node colors, shapes, and labels.
3. Create one or more statistical graphs to see standard relational views of node and link data separately. Each statistical graph is based on node or link data.
4. Create one or more network graphs to see the network of nodes and links. You can choose from several layout patterns when generating network graphs because you might not be aware initially of any underlying structure in the data, it can be helpful to try multiple layout patterns to determine which are the most useful for the network you are investigating. After creating the graph, you can later change its layout.
5. Explore and investigate the data in the following ways:
 - Explore and manipulate data in the data tables, for example, by sorting the data.
 - Use a combination of statistical graphics, network graphs, and data tables to selectively view and filter the data. To filter the data, select individual observations or groups of observations in a data table or in a graph. The observations are automatically selected in all graphs and data tables. You can also use local selection mode to graphically subset data at multiple levels.
 - Use the tools available to manipulate the graphs. For example, you can zoom in on a portion of a graph or apply a magnifying lens to a network graph. You can also show the labels for particular nodes in a network graph.
6. You might want to change the appearance and behaviors associated with a graph by changing the graph's properties. For example, the properties of a statistical graph determine features such as the graph's style and colors, the orientation and appearance of the graph's axes, whether and where the graph displays a legend, whether and where a graph displays titles, and more.
7. Save the data as a project so that you can easily reopen the data tables and associated graphs.

Credit-card Fraud Example

This example describes how to use SAS/GRAPH Network Visualization Workshop to investigate patterns of credit-card fraud. The data that is used here represents a sampling from a large credit-card transaction database. All transactions included here, both valid and fraudulent, relate to customers with at least one fraudulent credit-card transaction. A visualization of this data is shown in Figure 20. The blue nodes in the network correspond to merchants, and the red nodes correspond to customers. Each link represents a transaction between a customer and a merchant. The green links correspond to valid transactions, whereas the red links correspond to fraudulent transactions.

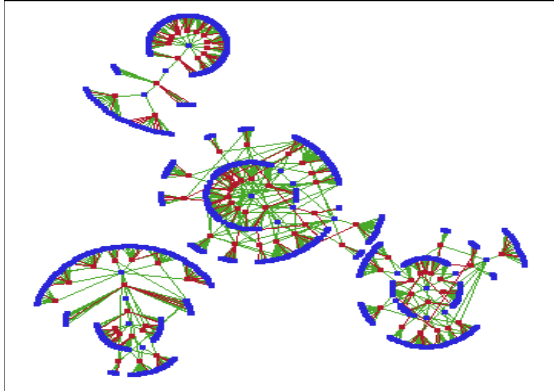


Figure 20

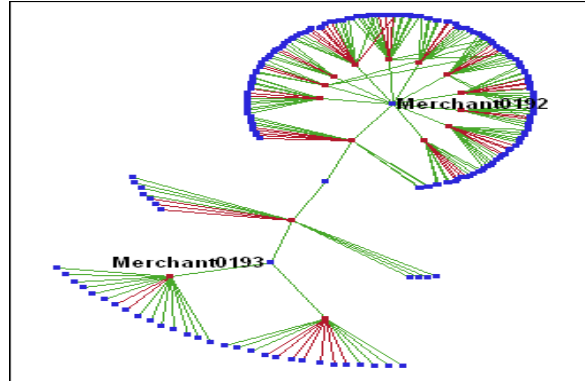


Figure 21

One scenario for credit-card fraud involves employees who work for the merchants in the data sets. In this scenario, the employee steals customers' credit-card numbers and either uses them or sells the numbers. The fraudulent transaction is typically not directly associated with the merchant where the employee works. Instead, the fraudulent transaction is connected to another merchant through a common customer with the original merchant. In a network graph, the fraudulent links (depicted in red here) are typically one link removed from the problem merchant. Therefore, rather than focus on fraudulent transactions, one should look at merchants with a significant number of customers. Keep in mind that all customers in the data set experienced at least one fraudulent transaction. A merchant that has connections with many customers in the transaction network raises suspicions of fraud. Although this approach is not proof of fraud, it identifies specific merchants that had access to customers' credit-card numbers that have fraudulent transactions associated with them. These merchants warrant further investigation. The degree of a node is the number of links having that node as an endpoint. In this example, the degree of a merchant is the number of customers that had a transaction with the merchant. The goal, then, is to identify merchants with a high degree, which is defined here as greater than or equal to three.

The subnetwork in the upper left portion of the network graph contains transactions between merchants and customers that are in Group 1. Figure 21 above magnifies and shows this subnetwork.

The simplicity of this subnetwork enables you to identify the merchants with a high degree by direct observation. (Keep in mind that blue markers represent merchants, and red markers represent customers.) When you apply labels to the merchants with high degree, you can identify these merchants as Merchant0192 and Merchant0193 with a degree of 11 and 3, respectively. For the remaining three clusters in the network graph, the density of the subnetworks makes it difficult to detect all the merchants with high degree using direct observation. You can use a statistical graph and the local selection feature to filter the data in the network graph. Figure 22 shows the result of using a scatter plot with a local selection mode to parse the visualized data. In this graph, the blue nodes that are visible represent all of the merchants with a high degree in groups 2, 3, and 4.

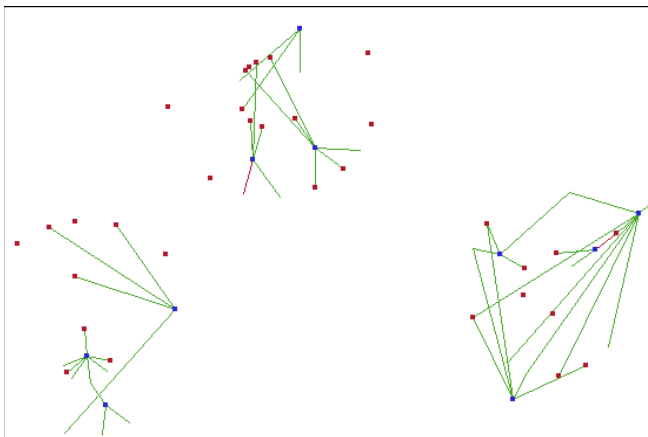


Figure 22

PROCEDURE ENHANCEMENTS AND NEW OPTIONS

Several new options have been added to individual procedures. They are briefly described below. For more detailed information and examples please see online documentation.

PROC GBARLINE now supports multiple PLOT statements with different variables to graph separate lines. Just as with bar charts, the GBARLINE procedure now supports the SUBGROUP option. Legend options have been added to both bar and plot statement. You can use legend statement on either the bar or plot statement.

PROC GMAP now supports the DENSITY option for the maps that have a DENSITY variable. The value that you specify indicates the maximum value that the DENSITY variable can have for the observation you want to display. The AREA statement was also added to control the color for the region part of the block or prism maps. The CDEFAULT option was added to control the default empty fill area.

PROC GRADAR now supports windrose and calendar charts. The windrose chart, named for charts of wind speed and direction, is a type of histogram that is useful when the extreme values of the histogram's midpoint variable are related. Typical applications include histograms involving direction, clock time, or other cyclical values.

The CALENDAR option produces a radar chart that displays 12 equal-sized segments, one for each month of the year (January through December). The color shading of each segment represents the magnitude of the frequency variable.

ANNOTATE now supports the ARROW function. This is similar to how you would draw line segments using the move/draw annotate functions.

With all procedures, the NAME= option now allows for values of up to 256 characters in length. This allows you to create external graph files with long filenames; however, only the first eight characters of the NAME= value are used when creating the name of the GRSEG entry in the SAS/GRAPH catalog.

SCALABLE VECTOR GRAPHICS DEVICE DRIVER (SVG)

SAS 9.2 now supports device of SVG, SVGZ, SVGT, and SVGVIEW. SVG output can be created using ODS LISTING, ODS HTML, or ODS PRINTER. SVG documents display clearly at any size in any viewer or browser that supports SVG. SVG documents are ideal for producing documents to print or to display on a computer monitor, PDA, or cell phone. Because it is a vector graphic, a single SVG document can be transformed to any screen resolution without compromising the clarity of the document. In comparison, output from an image-based device requires using different resolutions to display the image at various screen resolutions and sizes.

The SVG device generates an uncompressed SVG file. You have basic zoom in/out control through right mouse action on the graph.

SVGZ generates compressed SVG output. Some browsers might not be able to uncompress this to display the output. In that case, you will need to use a device of SVG.

SVGT generates a transparent SVG file.

The SVGVIEW device generates a raw SVG file that has buttons at the top of your output. This device is useful with by-group processing when used with ODS LISTING. But if you use ODS PRINTER, all pages in the ODS printer block will be contained in the SVG document. This document will contain paging control buttons and provide an index page with links to each page in the document.

CREATING AND VIEWING SVG DOCUMENTS

You create SVG documents using ODS Listing, HTML, or Printer destinations. SVG documents that are created by the ODS Listing and HTML destinations are useful for creating graphs with SAS/GRAPH. Using the ODS Listing or Printer destination results in a stand-alone SVG file. You can view this file using a browser that supports Scalable Vector Graphics. Some browsers, such as Mozilla Firefox, have built-in support for SVG documents. Using the ODS HTML destination results in an SVG document that is embedded in an HTML document using <embed> tags.

UPDATES TO MAP DATA SETS

Several updates were made to SAS map data sets. For more information on this, review the information at: [HTTP://SUPPORT.SAS.COM/RND/DATAVISUALIZATION/MAPONLINE/HTML/V92MAPS.HTML](http://support.sas.com/rnd/datavisualization/maponline/html/v92maps.html)

CHANGES IN SAS/GRAPH DOCUMENTATION

Information regarding certain features has been removed from the documentation:

- Information about the DS2CSF macro has been removed. The functionality of the DS2CSF macro is available through the new GKPI procedure.
- Information about the META2HTM macro has been removed. As a replacement for the META2HTM macro, use the ODS HTML statement with the JAVAMETA device driver.
- Information about the GPRINT procedure has been removed from the documentation. This procedure is shipped but will be obsolete in the future.

REVERTING TO A SAS 9.1.3. OR EARLIER APPEARANCE OF SAS/GRAPH

Many of the new features in SAS/GRAPH 9.2 change the default appearance of your SAS/GRAPH output. To produce output that looks as if it was produced with versions of SAS/GRAPH before SAS 9.2, perform the following steps:

- Specify the NOGSTYLE system option. This option turns off the use of ODS styles.
- Specify the FONTRENDERING=HOST_PIXELS system option. This option determines how fonts are rendered.
- Specify DEVICE=ZGIF on the GOPTIONS statement when you are sending output to the HTML destination.
- In other cases where your application specifies a device, specify a compatible Z device driver if applicable.

CONCLUSION

Whether you are creating a presentation or performing a data analysis, you will find that in SAS 9.2, SAS/GRAPH has several new features to serve your needs. Using default styles and devices based on the ODS destination will further reduce the amount of coding that you need to generate quality graphs. Support for TrueType fonts makes it easy to generate system-independent graphs. Customized graphs that previously required pages of programming using annotate can now be replaced with fewer lines of code using the Graph Template Language. The new SAS/GRAPH statistical graphics procedure also provides a simple syntax for creating graphics commonly used in exploratory data analysis and for creating customized statistical displays. Several new procedures such as GKPI, GTILE, GEOCODE, and GINSIDE were added to generate data-specific graphs. With all the new functionality and procedures, you can easily generate graphs for your data presentation, publications, or data analysis.

ACKNOWLEDGMENTS

The author would like to acknowledge the contributions of the following colleagues at SAS who have provided valuable feedback and assistance in producing this paper: Dan Heath, Jeff Cartier, Sanjay Matange, Ravi Devarajan, Robert Allison, Jeanne Ferneyhough, Darrell Massengill, Ed Odom, Howard Houston, Rob Dolan, Donna Bravo, Donna Antle, Martin Mincey, and Lelia McConnell. Their experience and work in ODS and SAS/GRAPH provided the foundation for the concepts presented here.

RECOMMENDED READING

Heath, Dan. 2007. "New SAS/GRAPH[®] Procedures for Creating Statistical Graphics in Data Analysis." *Proceedings of SAS Global Forum 2007*. Cary, NC: SAS Institute Inc. Available at <http://support.sas.com/rnd/datavisualization/papers/sgf2007/SGF2007-Proc.pdf>

Heath, Dan. 2008. "Effective Graphics Made Simple Using SAS/GRAPH[®] 'SG' Procedures." *Proceedings of SAS Global Forum 2008*. Cary, NC: SAS Institute Inc.

Matange, Sanjay. 2008. "ODS Graphics Editor." *Proceedings of SAS Global Forum 2008*. Cary, NC: SAS Institute Inc.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Himesh Patel
SAS Institute
R2233 SAS Campus Dr.
Cary, NC 27513
E-mail: Himesh.Patel@sas.com

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 trademarks of their respective companies.