Abstract

A powerful feature of the annotate facility is the capacity of using the system of coordinates generated by a graphics procedure to provide custom enhancements to the procedure output. This paper explores the use of the DATA step and the macro facility to add data-dependent, user-defined modifications to graphics output from the GSLIDE, GPLOT, and GMAP procedures.

General Capabilities of the Annotate Facility

SAS/GRAPH® software provides a powerful system of procedures for producing business graphics in a production environment. The graphics user can display information stored in a SAS® data set by selecting a SAS/GRAPH procedure and providing appropriate statements and options. A fixed range of modifications are possible for each procedure when selected statements and options are used.

To produce graphics output that is outside the range of modification possible with a procedure, or to produce graphics output without procedural constraints, SAS/GRAPH software provides the annotate facility.

The annotate facility gives you a set of commands that can be used to direct the actions of a graphics device. The commands are contained in a SAS data set and allow you to create pictures that can be overlayed with the output of a graphics procedure, or produce pictures independent of other graphics procedures.

The data sets used by the annotate facility are the same as those used by procedures and read by SET, MERGE, and UPDATE statements. Because they are created in the DATA step, you have the SAS language, the function library, and the macro facility at your disposal when generating the graphics commands.

Description of the Tutorial

This tutorial explores some of the ways that you can use the annotate facility to enhance the output of SAS/GRAPH procedures. Specific examples will illustrate the techniques for placing data-dependent text on a page of graphics output, generating a customized area fill for a line plot, and placing information on a map.

Strategies for developing code that addresses each of the techniques will also be presented.

Text Applications

Text is used in nearly every graphics display to clarify the content and meaning of the picture. Text is also used alone to disseminate information or to prepare the viewer for the next picture.

TITLE, NOTE and FOOTNOTE statements can be used to position text strings if the text strings have fixed values and occupy fixed positions on the page. To accommodate variable text strings in variable positions with TITLE, NOTE, or FOOTNOTE statements requires the use of the macro facility. A much easier alternative is provided by the annotate facility.

A Strategy for Annotating Text Strings

Any strategy for using the annotate facility requires identifying the fixed and variable components of the desired image. When dealing with text strings, use the following guidelines:

Determine fixed value text strings. Titles and footnotes that describe the display fall in this category.

Position variable value text strings. This is really a two-step process inside the DATA step used to produce the annotate= data set.

1. Set up variables the values of which will be the text strings.
2. Set up variables the values of which will be the X and Y positions of the text strings.

Locate the source data set. Typically, the information to be displayed is part of a data set, or a data set has been created containing the appropriate information.

Produce the ANNOTATE= data set. After gathering all pertinent information in the form of variables, you can begin to assign values to annotate.
variables. You need to produce one observation for each text string with appropriate values for the X, Y, and TEXT variables. Those are the variables that the annotate facility will use when actually drawing the text.

Display the ANNOTATE= data set with the GSLIDE procedure. You rely on the GSLIDE procedure for display because you have fixed value/position text strings in titles or footnotes.

Development of the DATA Step

To illustrate the creation of an ANNOTATE= data set we will produce a slide showing summarized information for six different sales regions of a company. We can assume this will be used as a presentation graphic and include appropriate titles and footnotes.

The first step then involves segregating the information that will go into titles and footnotes.

GOPTIONS NOTEXT82;
TITLE H=10 PCT F=XSWISS C=WHITE 'Regional Sales Figures';
FOOTNOTE H=3 peT F=SIMPJ.JEX C=WHITE J=L 'Fiscal Year 1985';

The second step involves determining the number of lines of text required to display the variable information. In our example we already know that we have six regions, so we could write a data step with that in mind. To generalize the code that we generate, let's pre-process the data set containing the information to be displayed, then use that information to construct the ANNOTATE= data set.

Shown below is the output from PROC CONTENTS for the input data set in our example,

CONTENTS OF SAS MEMBER WORK.SALES
NUMBER OF OBSERVATIONS: 5645  NUMBER OF VARIABLES: 7

Alphabetic List of Variables and Attributes

<table>
<thead>
<tr>
<th>Variable Type</th>
<th>Length</th>
<th>Position</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 AMOUNT</td>
<td>8</td>
<td>8</td>
<td>DOLLAR12.2</td>
</tr>
<tr>
<td>2 CUSTID</td>
<td>CHAR</td>
<td>8</td>
<td>12</td>
</tr>
<tr>
<td>3 DATE</td>
<td>NUM</td>
<td>8</td>
<td>33 MMDDYY8.</td>
</tr>
<tr>
<td>4 INVNO</td>
<td>NUM</td>
<td>8</td>
<td>65</td>
</tr>
<tr>
<td>5 REGION</td>
<td>CHAR</td>
<td>16</td>
<td>41</td>
</tr>
<tr>
<td>6 SITE</td>
<td>NUM</td>
<td>8</td>
<td>57</td>
</tr>
<tr>
<td>7 STAWARE</td>
<td>CHAR</td>
<td>15</td>
<td>20</td>
</tr>
</tbody>
</table>

The following code can be used to pre-process the data set:

DATA REGNUM;
SET SALES END=EOF;
BY REGION;
IF FIRST. REGION THEN R+1;
IF EOF THEN OUTPUT;
KEEP R;

Since our input data set is not summarized, we must summarize it prior to producing the slide. The summary data set is the one we will read to actually create the ANNOTATE= data set. The SUMMARY procedure will produce an appropriate data set when invoked in the following manner:

PROC SUMMARY DATA=SALES;
CLASS REGION;
VAR AMOUNT;
OUTPUT OUT=REGSALES SUM=REGSUM;
RUN;

We now have all the necessary information to produce the slide. The following DATA step puts everything together.

DATA GSLIDE;
IF _N=1 THEN SET REGNUM;
SET REGSALES;
IF REGION=' ' THEN DELETE;
RETAIN XSYS YSYS HSYS '3' Y 90;
LENGTH COLOR STYLE $ 8 TEXT $ 30;
YDECR=70/R;
SIZE=YDECR*.5;
STYLE='XSWISS';
COLOR='CYAN';
POSITION='6';
X=15;
Y=Y-YDECR;
TEXT=REGION; OUTPUT;
STYLE='XSWISSU';
COLOR='GREEN';
POSITION='4';
X=85;
TEXT=PUT(REGSUM,DOLLAR12.);
OUTPUT;
RUN;

Displaying the ANNOTATE= Data Set

Once the data set has been created, it is a simply a matter of displaying it on a device with the appropriate titles and footnotes.

PROC GSLIDE ANNOTATE=GSLIDE;
RUN;

Figure 1 Annotating Variable Value and Position Text Strings

Regional Sales Figures

Atlanta $3,436,700
Chicago $8,586,220
Dallas $1,466,760
Denver $5,455,500
New York $3,089,660
Seattle $12,419,540

Fiscal Year 1985

979
Plotting Applications

Line plots or two-dimensional scatter diagrams frequently need additional text within the axis boundaries to clarify the values of points or peaks in the plot. Using the annotate facility's LABEL function in the same frame of reference as the plot itself provides that capability.

Another enhancement to line plots that can be provided by the annotate facility is the use of custom interpolation techniques. If you wanted a single line on a plot that was solid in some areas and dashed in others, the SYMBOL statement could not provide a method of interpolation. The same would hold true if you wanted to use color to highlight different parts of the line.

The annotate facility gives you the opportunity to generate custom interpolations that vary in response to the data that is being plotted.

A Strategy for Generating Custom Interpolations

As an example let's assume that a company monitors actual production figures for a manufactured product and compares those figures to fixed, predetermined quotas. The data are stored in the PLOTDATA data set.

The first 10 observations of the PLOTDATA data set are shown below.

<table>
<thead>
<tr>
<th>OBS</th>
<th>X</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8000</td>
<td>490.28</td>
</tr>
<tr>
<td>2</td>
<td>8007</td>
<td>255.18</td>
</tr>
<tr>
<td>3</td>
<td>8014</td>
<td>1845.89</td>
</tr>
<tr>
<td>4</td>
<td>8021</td>
<td>647.26</td>
</tr>
<tr>
<td>5</td>
<td>8028</td>
<td>593.12</td>
</tr>
<tr>
<td>6</td>
<td>8035</td>
<td>1530.35</td>
</tr>
<tr>
<td>7</td>
<td>8042</td>
<td>1688.41</td>
</tr>
<tr>
<td>8</td>
<td>8049</td>
<td>1521.97</td>
</tr>
<tr>
<td>9</td>
<td>8056</td>
<td>1205.82</td>
</tr>
<tr>
<td>10</td>
<td>8063</td>
<td>617.75</td>
</tr>
</tbody>
</table>

The picture we want to produce is a line plot showing the distribution of production volume over time. Fill in the area above the quota (and below the plot line) with solid green, and the area below the quota (and above the plot line) with solid red.

We can use the following strategy to create the picture:

1. **Determine the Quota Value**
   - The quota value can be a numeric constant, DATA step variable, macro variable, or macro parameter. To make the quota value accessible to procedures and the DATA step, we assign its value to a macro variable with a %LET statement.
   ```
   %LET Q=1000;
   ```

2. **Determine the Polygon Points**
   - Since we are processing a data set that will be used by the GPLOT procedure, we can assume that the data are in ascending order by date. The following strategy will assist in the development of the DATA step:
     1. Compare the current point with the previous point to see if the quota line has been crossed.
     2. If the quota line has been crossed, generate a point on the quota line that is on the line between the two data points.

   ```
   DATA FIRST(KEEP=X Y);
   HX=X; HY=Y;
   SET PLOTDATA;
   RETAIN HX X HY Y;
   IF HY THEN DO;
     SLOPE=(HY-Y)/(HX-X);
     INTER=Y-SLOPE*X;
   END;
   ```

3. **Determine the Quota Value**
   - This is the easy part since someone else has already determined the quota. For our purposes we simply have to make the quota value accessible to the DATA step.

4. **Determine points that make up the polygons**
   - In order to fill in an area with color we must identify the area to the annotate facility as a polygon, then process the plot data set to determine the points of the polygons.

   ```
   ;
   ```

Select colors for the polygons.

We choose green above the quota line, red below the quota line.

Generate the plot with PROC GPLOT.

Annotate the plot with the polygons.

The following data step will accommodate our needs.

```
END;
ELSE DO; OUTPUT; RETURN; END;
HOLDX=X; HOLDY=Y;
IF (HY>=&Q AND Y<&Q) OR
(HY<&Q AND Y>=&Q) THEN DO;
X=(&Q-INTER)/SLOPE;
Y=&Q;
OUTPUT;
X=HOLDX; Y=HOLDY;
OUTPUT;
END;
ELSE OUTPUT;
RUN;

The first 10 observations in the FIRST data set are shown below.

<table>
<thead>
<tr>
<th>OBS</th>
<th>X</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8000.00</td>
<td>490.28</td>
</tr>
<tr>
<td>2</td>
<td>8007.00</td>
<td>255.18</td>
</tr>
<tr>
<td>3</td>
<td>8010.28</td>
<td>1000.00</td>
</tr>
<tr>
<td>4</td>
<td>8014.00</td>
<td>1845.89</td>
</tr>
<tr>
<td>5</td>
<td>8018.94</td>
<td>1000.00</td>
</tr>
<tr>
<td>6</td>
<td>8021.00</td>
<td>647.26</td>
</tr>
<tr>
<td>7</td>
<td>8028.00</td>
<td>593.12</td>
</tr>
<tr>
<td>8</td>
<td>8031.04</td>
<td>1000.00</td>
</tr>
<tr>
<td>9</td>
<td>8035.00</td>
<td>1530.35</td>
</tr>
<tr>
<td>10</td>
<td>8042.00</td>
<td>1688.41</td>
</tr>
</tbody>
</table>

Associate the Points with the Polygons

Now that we have all the individual points that we need to describe the polygons, we can create the polygons with the following rules in mind:

1. All polygons begin and end with points on the quota line.
2. The first point and the last point must be on the quota line.

The following DATA step assembles the points into polygon boundaries.

DATA SECOND(KEEP=X Y POLY);
SET FIRST END=EOF;
IF _tL=l THEN DO;
POLY+l;
IF Y=&Q THEN DO;
HY=Y; Y=&Q; OUTPUT;
Y=HY; OUTPUT;
END;
ELSE OUTPUT; RETURN;
END;
ELSE IF EOF THEN DO;
OUTPUT; POLY+l; OUTPUT;
ELSE OUTPUT; END;
RUN;

The first 10 observations in the SECOND data set are shown below.

<table>
<thead>
<tr>
<th>OBS</th>
<th>X</th>
<th>Y</th>
<th>POLY</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8000.00</td>
<td>1000.00</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>8000.00</td>
<td>490.28</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>8007.00</td>
<td>255.18</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>8010.28</td>
<td>1000.00</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>8010.28</td>
<td>1000.00</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>8014.00</td>
<td>1845.89</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>8018.94</td>
<td>1000.00</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>8018.94</td>
<td>1000.00</td>
<td>3</td>
</tr>
<tr>
<td>9</td>
<td>8021.00</td>
<td>647.26</td>
<td>3</td>
</tr>
<tr>
<td>10</td>
<td>8026.00</td>
<td>593.12</td>
<td>3</td>
</tr>
</tbody>
</table>

Select Colors for the Polygons

Color selection involves comparing the average Y value of a polygon with the quota value. If the average is greater than the quota, we assign it the color green; otherwise the polygon is assigned the color red.

PROC MEANS NOPRINT;
VAR Y; BY POLY;
OUTPUT OUT=POLYBAR MEAN=YB;
RUN;
DATA ANNPLOT(KEEP=FUNCT X Y COLOR STYLE XSYS YSYS HSYS);
SET SECOND;
LENGTH COLOR FUNCTION STYLE $ 8;
RETAIN XSYS YSYS HSYS '2' STYLE 'SOLID';
IF FIRST. POLY THEN DO;
FUNCTION=' POLY' ;
SET POLYBAR;
END;
ELSE FUNCTION='POLYCONT';
IF YB>&Q THEN COLOR= 'GREEN';
ELSE COLOR=' RED' ;
RUN;

The first 10 observations of the ANNPLOT data set are shown below.

<table>
<thead>
<tr>
<th>OBS</th>
<th>X</th>
<th>Y</th>
<th>COLOR</th>
<th>FUNCTION</th>
<th>STYLE</th>
<th>XSYS</th>
<th>YSYS</th>
<th>HSYS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8000.00</td>
<td>1000.00</td>
<td>RED</td>
<td>POLY</td>
<td>SOLID</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>8000.00</td>
<td>490.28</td>
<td>RED</td>
<td>POLYCONT</td>
<td>SOLID</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>8007.00</td>
<td>255.18</td>
<td>RED</td>
<td>POLY</td>
<td>SOLID</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>8010.28</td>
<td>1000.00</td>
<td>RED</td>
<td>POLY</td>
<td>SOLID</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>8010.28</td>
<td>1000.00</td>
<td>GREEN</td>
<td>POLY</td>
<td>SOLID</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>8014.00</td>
<td>1845.89</td>
<td>GREEN</td>
<td>POLY</td>
<td>SOLID</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>8018.94</td>
<td>1000.00</td>
<td>GREEN</td>
<td>POLY</td>
<td>SOLID</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>8018.94</td>
<td>1000.00</td>
<td>RED</td>
<td>POLY</td>
<td>SOLID</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>9</td>
<td>8021.00</td>
<td>647.26</td>
<td>RED</td>
<td>POLYCONT</td>
<td>SOLID</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>8026.00</td>
<td>593.12</td>
<td>RED</td>
<td>POLYCONT</td>
<td>SOLID</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

Generate the Plot

Finally, we can invoke the GPLOT procedure to create the plot. The GPLOT procedure provides the plotting environment in which we display the ANNOTATE= data set created in the last DATA step.

OPTIONS NOTEXT82;
PROC GPLOT DATA=PLOTDATA;
PLOT y*x
ANNOTATE=ANNOPLOT
VREF:::&.Q
HAXIS=8000 TO 9000 BY QTR;
FORMAT x TIQ.;
SYMBOL V=NOt-.'E;
TITLE F=XSWISS H=5 PCT C=WHITE 'Anno'tat'ing 'the GPLOT Procedure';
RUN;
Mapping Applications

Enhancements for mapping applications made possible by the annotate facility include:
- placing unit names on map surfaces
- drawing lines between points on a map
- placing bar or pie charts on a map surface
- plotting contour lines on a map surface.

This tutorial develops a strategy and shows how to draw data dependent bar charts on the surface of a map.

A Strategy for Placing Bar Charts on a Map

The example for this section involves displaying sales information for six regional sales centers. We want to show the difference between two years at each center, representing each year as a bar on the map located geographically around the sales center. Each sales center should be labeled with the city name and a star. The bars should have the actual dollar amounts that they represent placed on top of the bar.

Developing an ANNOTATE= data set for a mapping application involves using the coordinate system provided by a particular map. Our strategy for placing bar charts on a map is to initialize the position of the bar with a map coordinate, then draw the bar and position text by moving relative to the initial position. This will require the use of the relative frames of reference provided by the annotate facility.

The following strategy can be used when positioning bars on a map:

Determine the coordinate system of the map.
- This step simply requires choosing a map data set on which to display the bar charts.

Create an ANNOTATE= data set.
- Use the map coordinates and an input data set to create the ANNOTATE= data set. The size of the bars should vary according to a variable in the input data set.

Display the map with the GMAP procedure.

Display the bars on the map with the annotate facility.

Determine the Coordinate System of the Map

The map that is chosen will determine the coordinates used to initialize the bars and accompanying text strings when constructing the ANNOTATE= data set. We have a choice between using a projected map or an unprojected map.

If we use an unprojected map the coordinates we use for the annotation will be in degrees or radians latitude and longitude. To ensure proper positioning of the annotation we must concatenate the annotated coordinates with the map data set and project both sets of data as a unit.

Using a map that is already projected simplifies matters considerably because we can place the annotation directly on the map. The only prerequisite is knowledge of the coordinate system of the projected map.

In our example we will use the projected US map supplied with SAS/GRAPH software. The USCITY data set, also supplied with SAS/GRAPH, provides the coordinates for major U.S. cities located on the US map. For the input data set we will use the SALES data set containing yearly sales summaries for the six regional sales centers of a company.

The SALES data set is shown below:

<table>
<thead>
<tr>
<th>OBS</th>
<th>CITY</th>
<th>YEAR</th>
<th>SALES</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Atlanta</td>
<td>1984</td>
<td>114400</td>
</tr>
<tr>
<td>2</td>
<td>Atlanta</td>
<td>1985</td>
<td>137620</td>
</tr>
<tr>
<td>3</td>
<td>Chicago</td>
<td>1984</td>
<td>2888070</td>
</tr>
<tr>
<td>4</td>
<td>Chicago</td>
<td>1985</td>
<td>3567100</td>
</tr>
<tr>
<td>5</td>
<td>Dallas</td>
<td>1984</td>
<td>455960</td>
</tr>
<tr>
<td>6</td>
<td>Dallas</td>
<td>1985</td>
<td>6093350</td>
</tr>
<tr>
<td>7</td>
<td>Denver</td>
<td>1984</td>
<td>1954140</td>
</tr>
<tr>
<td>8</td>
<td>Denver</td>
<td>1985</td>
<td>2223900</td>
</tr>
<tr>
<td>9</td>
<td>New York</td>
<td>1984</td>
<td>1058620</td>
</tr>
<tr>
<td>10</td>
<td>New York</td>
<td>1985</td>
<td>1294010</td>
</tr>
<tr>
<td>11</td>
<td>Seattle</td>
<td>1984</td>
<td>4246120</td>
</tr>
<tr>
<td>12</td>
<td>Seattle</td>
<td>1985</td>
<td>5167130</td>
</tr>
</tbody>
</table>

Create the ANNOTATE= Data Set

The first step is to calculate the overall sales total for all sales regions and years. This figure will be needed in the final DATA step that
actually determines the size of the bars. We can use the MEANS procedure to perform the calculation and place the value in a macro variable using the DATA step.

PROC MEANS DATA=SALES NOPRINT; VAR SALES; OUTPUT OUT=TOT SUM=TOT; DATA _NULL_; SET TOT; CALL SYMPUT(SALES,TOT); RUN;

The next step is to take the observations from the USCITY data set that correspond with the sales centers and create the CITIES data set with two observations for each city. This data set will provide the labels for the cities and the star at each city location. It will also provide the initialization point for the bars in the next data step.

Notice that the northernmost city, Seattle, has its label positioned above the star. This is to allow the bars to be placed below the star and not run off the top of the map.

DATA CITIES; SET USCITY; IF CITY='New York' OR CITY='Chicago' OR CITY='Atlanta' OR CITY='Seattle' OR CITY='Denver' OR CITY='Dallas'; WHEN='A'; RETAIN XSYS YSYS '2' HSYS '1' COLOR 'WHITE' FUNCTION 'LABEL'; STYLE='SPECIAL'; SIZE=7; TEXT='I'; OUTPUT; STYLE='DUPLEX'; SIZE=4; TEXT=CITY ; IF CITY='Seattle' THEN POSITION='2'; ELSE POSITION='1'; OUTPUT; KEEP FUNCTION CITY X Y POSITION STYLE COLOR TEXT SIZE XSYS YSYS HSYS; PROC SORT DATA=CITIES; BY CITY; RUN;

The CITIES data set is shown below.

<table>
<thead>
<tr>
<th>OBS</th>
<th>CITY</th>
<th>X</th>
<th>Y</th>
<th>POSITION</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Atlanta</td>
<td>0.16521</td>
<td>-0.08151</td>
<td>6</td>
<td>LABEL</td>
</tr>
<tr>
<td>2</td>
<td>Atlanta</td>
<td>0.16521</td>
<td>-0.08151</td>
<td>6</td>
<td>LABEL</td>
</tr>
<tr>
<td>3</td>
<td>Chicago</td>
<td>0.16626</td>
<td>0.08754</td>
<td>6</td>
<td>LABEL</td>
</tr>
<tr>
<td>4</td>
<td>Chicago</td>
<td>0.16626</td>
<td>0.08754</td>
<td>6</td>
<td>LABEL</td>
</tr>
<tr>
<td>5</td>
<td>Dallas</td>
<td>0.01156</td>
<td>-0.78536</td>
<td>8</td>
<td>LABEL</td>
</tr>
<tr>
<td>6</td>
<td>Dallas</td>
<td>0.01156</td>
<td>-0.78536</td>
<td>8</td>
<td>LABEL</td>
</tr>
<tr>
<td>7</td>
<td>Denver</td>
<td>-0.31591</td>
<td>0.45958</td>
<td>8</td>
<td>LABEL</td>
</tr>
<tr>
<td>8</td>
<td>Denver</td>
<td>-0.31591</td>
<td>0.45958</td>
<td>8</td>
<td>LABEL</td>
</tr>
<tr>
<td>9</td>
<td>New York</td>
<td>0.28527</td>
<td>0.09419</td>
<td>8</td>
<td>LABEL</td>
</tr>
<tr>
<td>10</td>
<td>New York</td>
<td>0.28527</td>
<td>0.09419</td>
<td>8</td>
<td>LABEL</td>
</tr>
<tr>
<td>11</td>
<td>Seattle</td>
<td>-0.31591</td>
<td>0.22430</td>
<td>2</td>
<td>LABEL</td>
</tr>
<tr>
<td>12</td>
<td>Seattle</td>
<td>-0.31591</td>
<td>0.22430</td>
<td>2</td>
<td>LABEL</td>
</tr>
</tbody>
</table>

The last step is the creation of the BARS data set. We must merge the CITIES data set with the SALES data set. The CITIES data set provides the initialization point for the bars; the SALES data set provides the size of the bars and the value of the bars.

A key factor in the creation of the BARS data set is the capability to initialize the bar position in an absolute frame of reference and to draw the bar in a relative frame of reference.

Create the ANNOTATE= Data Set

DATA BARS; MERGE CITIES SALES; BY CITY; WHEN='A'; IF FIRST.CITY THEN DO; FUNCTION='MOVE'; OUTPUT; FUNCTION='LABEL'; OUTPUT; XSYS='9'; YSYS='9'; HSYS='9'; TEXT=LEFT(PUT(SALES/1000000,5.2) || 'M'); OUTPUT; END; ELSE DO; FUNCTION='NOVE'; OUTPUT; FUNCTION='LABEL'; OUTPUT; XSYS='9'; YSYS='9'; HSYS='9'; TEXT=LEFT(PUT(SALES/1000000,5.2) || 'M'); OUTPUT; END; KEEP FUNCTION CITY X Y POSITION STYLE COLOR TEXT SIZE XSYS YSYS HSYS; PROC SORT DATA=BARS; BY CITY; RUN;
Create the ANNOTATE= Data Set

The first 12 observations of the BARS data set are shown below.

<table>
<thead>
<tr>
<th>OBS</th>
<th>CITY</th>
<th>X</th>
<th>Y</th>
<th>POSITION</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Atlanta</td>
<td>0.1652</td>
<td>-0.05152</td>
<td>MOVE</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Atlanta</td>
<td>0.1652</td>
<td>-0.05152</td>
<td>MOVE</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Atlanta</td>
<td>-1.0000</td>
<td>1.00000</td>
<td>MOVE</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Atlanta</td>
<td>-1.0000</td>
<td>0.42237</td>
<td>BAR</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Atlanta</td>
<td>0.0000</td>
<td>0.00000</td>
<td>3</td>
<td>LABEL</td>
</tr>
<tr>
<td>6</td>
<td>Atlanta</td>
<td>0.1652</td>
<td>-0.05152</td>
<td>MOVE</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Atlanta</td>
<td>0.1652</td>
<td>-0.05152</td>
<td>MOVE</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Atlanta</td>
<td>1.0000</td>
<td>1.00000</td>
<td>MOVE</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Atlanta</td>
<td>0.0000</td>
<td>0.00000</td>
<td>3</td>
<td>LABEL</td>
</tr>
<tr>
<td>10</td>
<td>Atlanta</td>
<td>0.1652</td>
<td>-0.05152</td>
<td>MOVE</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Atlanta</td>
<td>0.0000</td>
<td>0.00000</td>
<td>3</td>
<td>LABEL</td>
</tr>
</tbody>
</table>

OBS STYLE COLOR TEXT XSYS YSYS HSYS SIZE
1 SPECIAL WHITE I 2 2 1 7
2 SPECIAL WHITE I 2 2 1 7
3 SPECIAL WHITE I 2 2 1 7
4 SOLID GREEN I 7 7 7 7
5 SOLID GREEN I 7 7 7 7
6 DUPLEX WHITE 0.11M 7 7 7 7
7 DUPLEX WHITE Atlanta 2 2 1 4
8 DUPLEX WHITE Atlanta 2 2 1 4
9 DUPLEX WHITE Atlanta 2 2 1 4
10 SOLID RED Atlanta 7 7 7 4
11 SOLID RED Atlanta 7 7 7 4
12 DUPLEX WHITE 0.11M 7 7 7 3

To avoid clutter on the map we will use the GREMOVE procedure to excise all internal state boundaries from the US map. After we place the bars and text on the map, the state boundaries will not provide significant information to the viewer of the picture.

DATA US;
SET MAPS.US;
COUNTRY=C;
IF STATE=STFIPS('AK') OR STATE=STFIPS('HI') THEN DELETE;
PROC GREMOVE DATA=US OUTUS;
ID STATE;
BY COUNTRY;
RUN;

Display the Map and Annotated Bar Charts

The last step is also the simplest to accomplish. Invoke the GMAP procedure with the appropriate statements and options, including the ANNOTATE= option.

PROC GMAP DATA=US MAPUS;
ID COUNTRY;
CHORD COUNTRY / ANNOTATE=BARS NOLEGEND OUTLINE=CYAN;
PATTERN=V4;
RUN;

The resulting picture is shown below.

Figure 3 US Map with Annotated Labels and Bar Charts