USING ADVANCED ANNOTATE FUNCTIONS TO DISPLAY WIND DISTRIBUTIONS

Ronald L. Coleman, U. S. Air Force ETAC

Abstract

Sometimes there isn't a procedure to draw that special graph you need. So you turn to Annotate data sets with all those functions. But wait, you still can't draw that graph. What next?

Give up?

Draw it by hand?

Use the advanced functions for Annotate data sets.

Here is a graph that draws itself! Rectangles that aren't limited to vertical and horizontal. Variable scaling to handle large distributions. A full legend to explain the graph and still room for titles and footnotes.

This special graph is a wind rose, depicting the frequency that winds occur from each of 16 compass points. This type of graph is used extensively by design engineers when structures are planned. The graph is created by a macro so any user with raw data can use it with a one-line macro call.

Background

Graphical wind roses are used by design engineers to determine wind patterns. The wind rose, drawn below, incorporates direction and percent frequency of occurrence by grouped wind speeds. The wind patterns help determine runway placement, building design, snow fence placement, and runway repair schedules.

Looking at one leaf, each rectangle is a different size because it represents the frequency of a wind speed group from the given direction. Therefore, the dimensions of each rectangle are determined by the input data frequency. Computing the X and Y coordinates for the opposing corners is very difficult.

Problems arose when trying to build the leaves of the wind rose as rectangles stacked end to end. Rectangles are relatively easy to draw using the ANNOTATE BAR function, but our needs were beyond this function because each rectangle varies in size, direction, and pattern.

The ANNOTATE BAR function easily creates horizontal and vertical rectangles, but we need the rectangles to lie along the 16 points of the compass. This eliminates the BAR function from consideration. The rectangles also need varying fill patterns to distinguish between the speed categories.

The POLY and POLYCONT functions proved to be the solution to our problem. The corners of each rectangle could be computed and then aligned in any orientation. A POLY option also fills the polygon and eliminates our fill pattern problem.

Technique

Each observation in the input data set contains the wind direction grouped into 16 compass points, wind speed grouped into the appropriate categories, month, and the percent frequency for the category. The data set is sorted by month, wind direction, and wind speed.

Macros are used to issue all ANNOTATE functions. The SAS Institute supplied macro ANNOMAC loads these macros into memory for use by the calling program. If you are not familiar with macro processing this is a good time to show you what these powerful tools can do for you. The Institute did not include a macro for the PIECNTR function, so here is the code needed.

\%MACRO PIECNTR(x1,y1,siz);
  FUNCTION = "PIECNTR ";
  X = &x1;
  Y = &y1;
  SIZE = &siz;
  OUTPUT;
\%MEND PIECNTR;

February
The macro is called with a leading percent sign:

```
%PIECNTR(X,Y,SIZE);
```

As you can see these macros reduce the amount of code necessary in your programs, one line instead of five!

The ANNOTATE environment is initialized with XSYS, YSYS, and ZSYS set to '5' using the SYSTEM macro. This allows the functions to operate within the graphics output portion of the screen allowing TITLE and FOOTNOTE statements. All distances are measured in percentage of the graphics screen.

The wind rose begins by drawing the center circle with a radius of 5 (percent of the screen). Each observation causes one rectangle or leaf to be drawn. For the first observation of a direction, (DIRECT), the distance from the center of the circle for the rectangle origin point, (DIST), is set as the radius of the circle. Each rectangle is then constructed by a macro called WINDLEAF.

```
WINOLEAF is passed the rectangle origin offset (DIST), direction (DIRECT), width of the rectangle (WIDTH) which is based on the wind speed group, the fill pattern (EMPTY or SOUD), and the percent frequency for this wind speed group (PERCENT). Because the POLY and POLYCONT functions don't allow you to compute new coordinates until the polygon is finished, the corner points of the rectangle are computed by the PIECNTR and PIEXY functions and then stored in the stack.
```

The circle center points are saved with the PUSH function. Then we move to the starting point of the rectangle using the PIECNTR and PIEXY functions. From the starting point the first corner is reached by moving to the left, (DIRECT - 90), one half the width of the rectangle, (WIDTH / 2). PIECNTR sets the current position as the center of a circle of a given radius, usually either the width of the rectangle or the length as given by PERCENT. PIEXY calculates the coordinates along the circle in a given direction from the center. Using the ANNOTATE macros the commands are:

```
%PIECNTR(.,.,&WIDTH / 2);
%PIEXY(&DIRECT -90.1);
%PUSH;
```

The coordinates for the first point are then pushed onto the stack for later use.

Each remaining corners around the rectangle are then computed and saved with this code:

```
%PIECNTR(.,.,&PERCENT);
%PIEXY(&DIRECT,1);
%PUSH;

%PIECNTR(.,.,&WIDTH);
%PIEXY(&DIRECT + 90.1);
%PUSH;

%PIECNTR(.,.,&PERCENT);
%PIEXY(&DIRECT + 180.1);
```

Now we start the polygon, the coordinates from the last corner are still in XLAST and YLAST so we don't need to do anything special. The POLY macro call looks like this:

```
%POLY(.,.,&COLOR,&STYLE,1);
```

The next corner is popped off the stack and we continue to build the rectangle with POLYCONT:

```
%POP;
%POLYCONT(-999,-999,*)
```

Notice that in PIECNTR and POLY we used a period '.' to use the current settings of XLAST and YLAST but POLYCONT requires -999 for this same function.

The last two corners are popped and drawn the same way which completes the polygon since the POLY function will always close the polygon with a line from the last point to the first.

```
WINDLEAF does some clean-up by popping the original circle center coordinates off the stack and recomputing the DIST for the next rectangle to start.
```

**Summary**

Although I haven't explained all the details involved in creating the wind rose, I have shown you how each leaf is created. Each wind rose is built using the WINDLEAF macro many times.

The ANNOTATE macros really made this project much simpler by decreasing the amount of code I needed to write. I learned the importance of and the incredible power of SAS Macro processing.

**Author:**

Ronald L. Coleman  
USAFETAC  
Scott AFB, IL 62225-5438  
(618) 256-5323

*SAS is the registered trademark of the SAS Institute, Inc., Cary, NC.*