Paper SAS520-2017

Easy Polar Graphs with SG Procedures

Prashant Hebbar and Sanjay Matange, SAS Institute Inc., Cary, NC

ABSTRACT

Some data is best visualized in a polar orientation, particularly when the data is directional or cyclical. Although the SG procedures and Graph Template Language (GTL) do not directly support polar coordinates, they are quite capable of drawing such graphs with a little bit of data processing. We demonstrate how to convert your data from polar coordinates to Cartesian coordinates and use the power of SG procedures to create graphs that retain the polar nature of your data.

Stop going around in circles: let us show you the way out with SG procedures!

INTRODUCTION

The SG procedures debuted with ODS Graphics in SAS® 9.2. They offer a simple interface to using the Graph Template Language (GTL) that underlies the ODS Graphics system. The SGPLOT procedure lets you create single-celled scatter plots, series plots, box plots and more in a quick and simple manner.

Currently, GTL uses the Cartesian coordinate system (X, Y) and does not directly support the polar coordinates (r, theta). Here, we demonstrate that we can draw graphs using data in polar coordinates after pre-processing the data to transform polar coordinates into Cartesian coordinates.

We also illustrate some additional steps for creating an axis-like circular grid for radial coordinate reference as well as radial lines for angular coordinate reference.

These techniques will help you draw polar graphs with the SGPLOT procedure with ease! Note that some of the examples shown here require the *fourth maintenance release* for SAS® 9.4.

See the Resources section later in this paper for links to the PDF file of this paper and the programs used here.

Transforming from Polar to Cartesian Coordinates

If you have a data point A with a radial distance r and an angle θ, the equivalent Cartesian coordinates are given by the well-known trigonometric equations:

```
x = r * cos(θ)
y = r * sin(θ)
```

as shown in Figure 1.

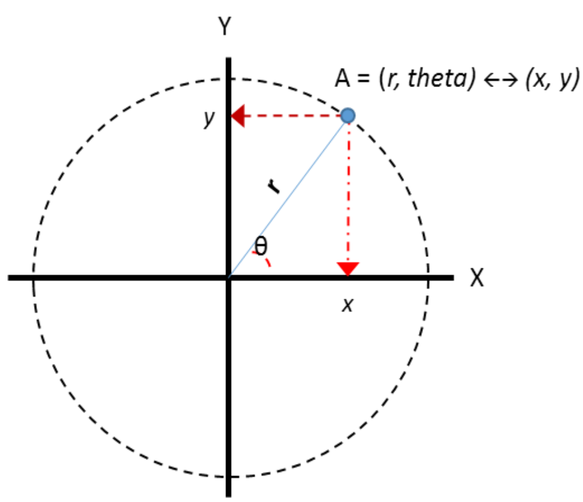

Figure 1. Polar and Cartesian Coordinates

We use this transform in all the examples that follow.

SCATTER PLOT

Let us consider plotting the temperatures for Orlando, FL over several years as a scatter plot. An SGPLOT example is shown in Figure 2:

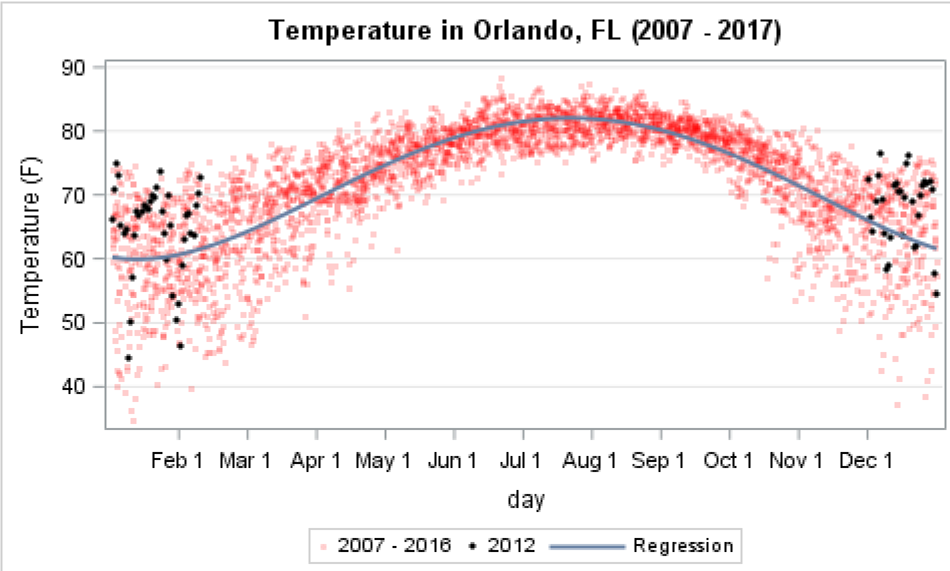

Figure 2. Scatter Plot of Orlando FL Temperatures

The data set contains temperatures from 2007 to 2017. The observations for 2017 have been plotted with black symbols. The blue line represents a regression line for the 2007 to 2016 data. Since temperature tends to be cyclical with time, this data set is a good candidate for plotting as a polar graph with temperature as the radial coordinate and time as the angle coordinate. Let us tackle this in parts.

Data Preparation

After the data was downloaded from an online site, we fed it into an SGPLOT procedure call with a scatter statement and a regression statement, and captured the resulting output data set (named SG_COMPUTED). This data set contains the raw variable TEMP1 for year 2007 to 2016 temperatures, TEMP2 for year 2017 temperatures, and a DAY variable (mapped to [0, 360]). In addition, the regression computed by SGPLOT has been stored into PREDICT and REG_DAY variables.

DAY	TEMP1	TEMP2	REG_DAY	PREDICT
0.984	70.6	.	0.984	60.223
1.967	65	.	2.779	60.127
...	...			
54	.	68	.	.

Table 1. Data Captured from SGPLOT

Now we can use the transform discussed in "Transforming from Polar to Cartesian Coordinates" to convert the data into Cartesian coordinates:

```
/*--Transform temp and fit values in to polar coordinates--*/
%let minR=0;    /* Ideally, floor(min temp/20)*20 for R grids of 20 */
%let maxR = 100; /* Ideally, ceil(max temp/20)*20 for R grids of 20 */
data polar;
```

```
    set sg_computed ; /* captured from sgplot run */
    keep x1 y1 x2 y2 xp yp;
    theta=%deg2rad(day);
    reg_theta=%deg2rad(reg_day);
    xCos = cos(theta);
    ySin = sin(theta);

    /* Compute equivalent Cartesian coordinates for polar plot of temp */
    x1=(temp1 - &minR) * xCos;
    y1=(temp1 - &minR) * ySin;
    x2=(temp2 - &minR) * xCos;
    y2=(temp2 - &minR) * ySin;

    /* Compute equivalent Cartesian coordinates for polar plot of predict */
    xp=(predict - &minR) * cos(reg_theta);
    yp=(predict - &minR) * sin(reg_theta);

  run;
```

This gives us variables X1 and Y1 for the historical observations, and X2 and Y2 for the year 2017 observations. Variables XP and YP give us the regression points.

X1	Y1	X2	Y2	XP	YP
70.589	1.211	.	.	60.215	1.033
64.961	2.231	.	.	60.056	2.914
...	...				
.	.	39.874	55.081	.	.

Table 2. Transformed Data

Drawing the Polar Plot

We can now create the polar plot version of the graph in Figure 2 using the transformed data:

```
  proc sgplot data=polarFinal aspect=1.0;
    format deg deg. ;
    scatter x=x1 y=y1 / markerattrs=(symbol=circleFilled size=3 color=red)
          name='a' legendlabel='2007 - 2016' transparency=0.8; /*historical*/
    scatter x=x2 y=y2 / markerattrs=(symbol=diamondFilled size=4 color=black)
          name='b' legendlabel='2017'; /* latest */
    series x=xp y=yp/ lineattrs=graphfit name='c' legendLabel='Regression';

    /* Polar axis components */
    vector x=thX y=thY / xOrigin=0 yOrigin=0 noArrowHeads
          lineAttrs=graphgridlines; /* Radial lines for theta "axis" */
    text x=thLabelX y=thLabelY text=deg / textAttrs=(size=10); /* labels */

    /* NOTE: EllipseParm requires SAS 9.4 M4 */
    ellipseParm semimajor=semi semiminor=semi /
          lineAttrs=graphGridlines;  /* Circles for r "axis" */
    text x=rLabelX y=rLAbelY text=semi / textAttrs=(size=10); /* labels */

    discreteLegend 'a' 'b' 'c';
```

```
    %let comOffset=0.05; /* keep data area square */
    xaxis display=none offsetMin=&comOffset. offsetMax=&comOffset.;
    yaxis display=none offsetMin=&comOffset. offsetMax=&comOffset.;
  run;
```

The result is shown in Figure 3. Polar Plot of Temperature

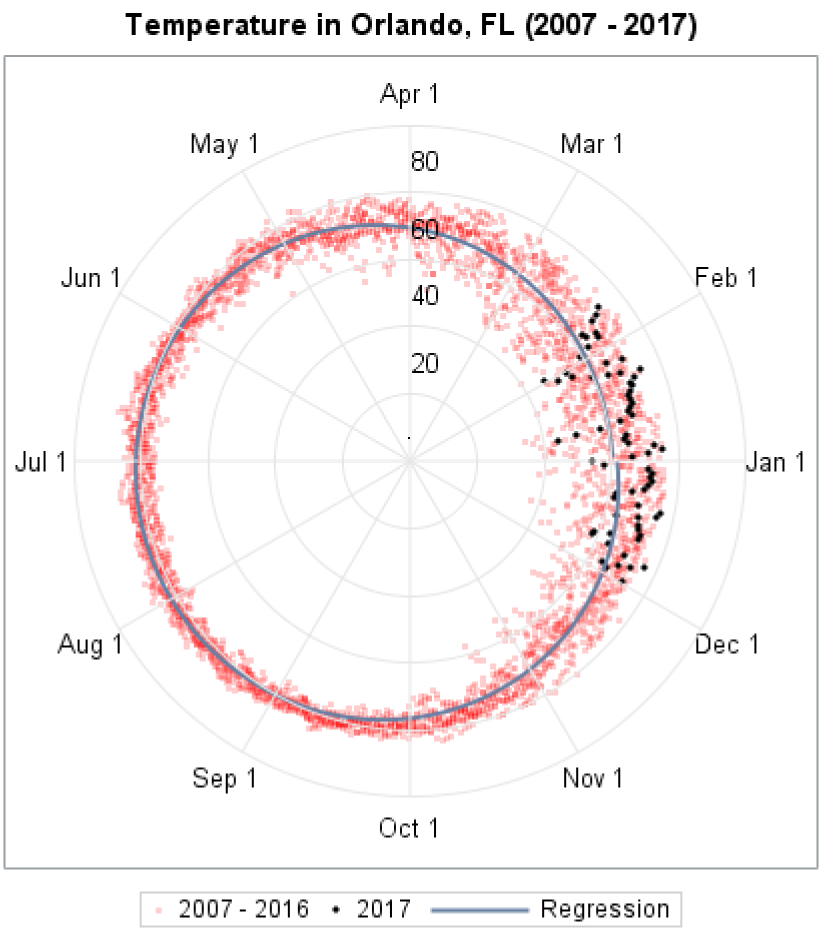

Figure 3. Polar Plot of Temperature

With this graph we can clearly see how the temperature changes with time. It is easy to see that the winter temperatures have more variation compared to the summer months. In the X-Y plot, the winter months were disjointed due to the axis starting at January and ending with December. This graph was previously published in the Graphically Speaking blog post “Simpler is better” at http://blogs.sas.com/content/graphicallyspeaking/2012/04/09/simpler-is-better/

Polar Axes

We have used radial lines at regular intervals to provide a reference for the theta coordinate. The interval was judiciously chosen to provide 12 sectors and implemented using the VECTOR statement and labeled with the TEXT statement.

The axis reference for the r coordinate was implemented using the ELLIPSEPARM statement to draw circles at intervals of 20.

Equal X and Y Drawing Scale

It is important to retain equal data to drawing scale in both axis when drawing polar plots. Since SGPLOT does not support an equated layout similar to GTL's LAYOUT OVERLAYEQUATED, we need to approximate that effect with the following:

- Set the aspect ratio to 1.0 using the ASPECT= option in the SGPLOT statement. This ensures a square wall area.
- Set the min and max offsets for both axes to the same value. Automatic offsets computed by the rendering system may not be equal for both axes. Choose the smallest value that will retain all the labels in the graph.

These measures help to keep the rendering of circles from getting skewed into ovals due to different X and Y data ranges. The full program for this graph (and a GTL implementation) can be found in the program code file.

SERIES PLOT

The previous example has already shown a series plot for regression values being drawn in polar coordinates, but there is a caveat when using sparse data.

The expectation of a "segment" between two points with the same r values but different angles in a polar graph is an arc of constant radius r. In Figure 4, the blue arc is what you expect in a polar graph. However, our render system is based on the Cartesian system, and a segment between two points is drawn as a straight line, as shown by the red dotted line.

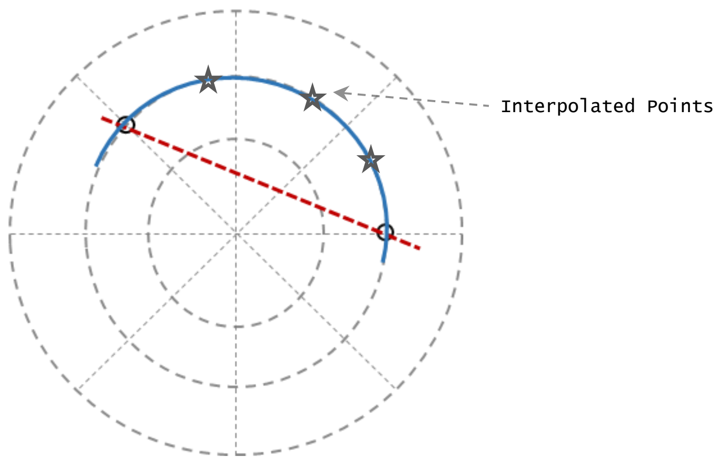

Figure 4. Series Plot Interpolation as an X-Y Graph and Polar Graph

To work around this issue for series plots in our approach, you have to add intermediate points between two data points if they are not close enough, as indicated by the star symbols in Figure 4.

These points can be calculated using linear interpolation in the original (r, theta) data space and then transformed into the equivalent (x, y) coordinates. In addition, the SMOOTHCONNECT option in the SERIES statement helps smooth the segment.

NEEDLE PLOT

A needle plot in polar coordinates is useful to accentuate the vector direction (relative to the origin) of the points being plotted. The needle statement in the SGPLOT procedure will only draw needles to the X axis baseline.

Instead, you can use the vector plot to render a needle plot in polar coordinates. If you want to show markers at the end point of the vector, you can overlay a scatter plot on the needle plot.

Astigmatism Vector Example

Figure 5 shows an example of a graph used in studying astigmatism changes with eye surgery. Note that the data has been approximately re-created for illustration.

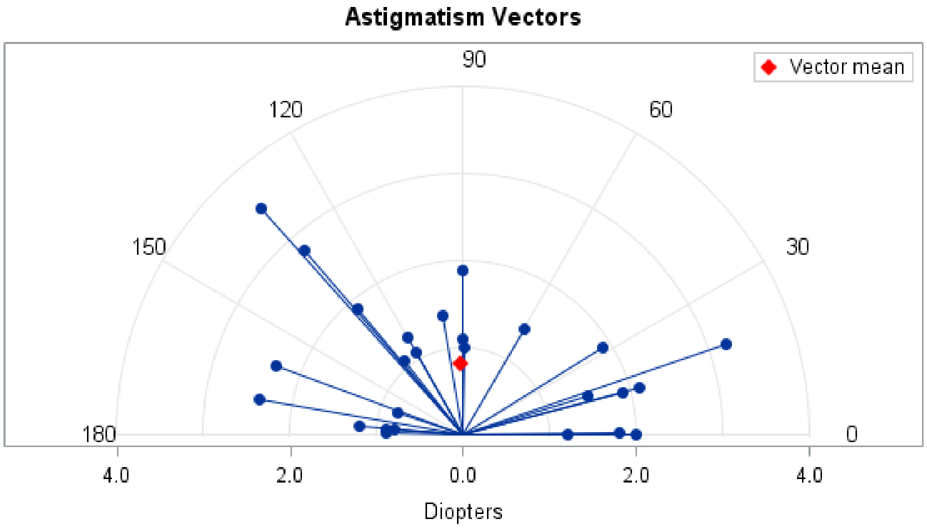

Figure 5. Astigmatism Vectors

In addition to the raw observations, we have also shown the vector mean of all the observations as a red diamond using another scatter plot.

Since this graph typically uses data with an angle range of 0 to 180 degrees (semi-circle), we can repurpose the standard X axis for our r axis. Note that the tick values show only the magnitudes and not the sign. This was done with a user-defined format to drop the negative sign. In addition, we have used an aspect ratio of 0.5 for this graph.

Once the polar coordinates have been transformed to Cartesian values as shown previously, the graph can be drawn as follows:

```
proc sgplot data=polarFinal aspect=0.5;
  xaxis label="Diopters" tickValueFormat=absValue.;
  yaxis display=none min=0 offsetMin=0.03;

  /* Polar axis and grid components */
  vector x=thX y=thY / xorigin=0 yorigin=0
          noarrowHeads lineattrs=graphGridlines;
  text x=thLabelX y=thLabelY text=deg / textAttrs=(size=10);
  /* Needs SAS 9.4 M4 */
  ellipseParm semimajor=semi semiminor=semi / clip
          lineAttrs=graphGridlines;

  /* Main plot components */
  vector x=x y=y / xorigin=0 yorigin=0 noarrowHeads;
  scatter x=x y=y / name='a' markerAttrs=(symbol=circleFilled);
  scatter x=xM y=yM / name='m' markerAttrs=(symbol=diamondFilled color=red)
                      legendLabel="Vector mean"; /* Mean marker */

  discreteLegend 'm' / location=inside position=topRight;
run;
```

Note that we have set the CLIP flag in the ELLIPSEPARM statement to allow the resulting circles to be clipped to the upper half. This prevents the circles from being drawn in the lower half where there are no observations present. The details of data preparation are available in the program code file.

BAR CHART

What would a bar chart look like in polar coordinates? One treatment is shown in Figure 6, taken from Paper 267-2012 (See References):

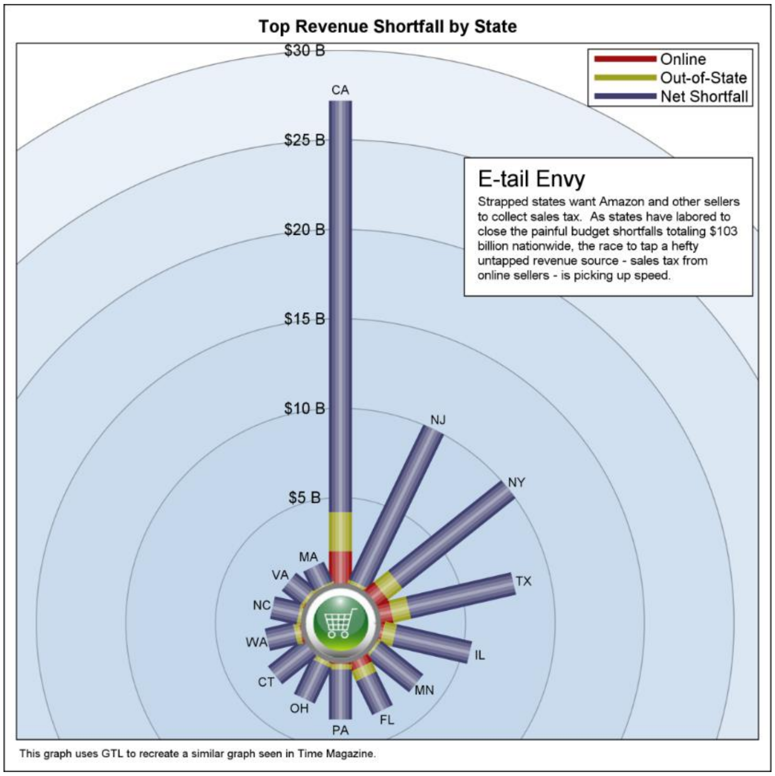

Figure 6. Polar Bar Chart - Using a Vector Plot

Here the bars have constant width along the r axis – it is really a thick vector plot using GTL, but it can be re-created with SGPLOT as well. You can find the link to the code for this graph in the References section.

Wind Rose Example

Another way to graph a bar chart in polar coordinates is to project the bar elements as if they were a rectangle in Cartesian space. Let us consider the stacked group bar chart shown in Figure 7. The graph shows the concentration of black carbon with wind direction.

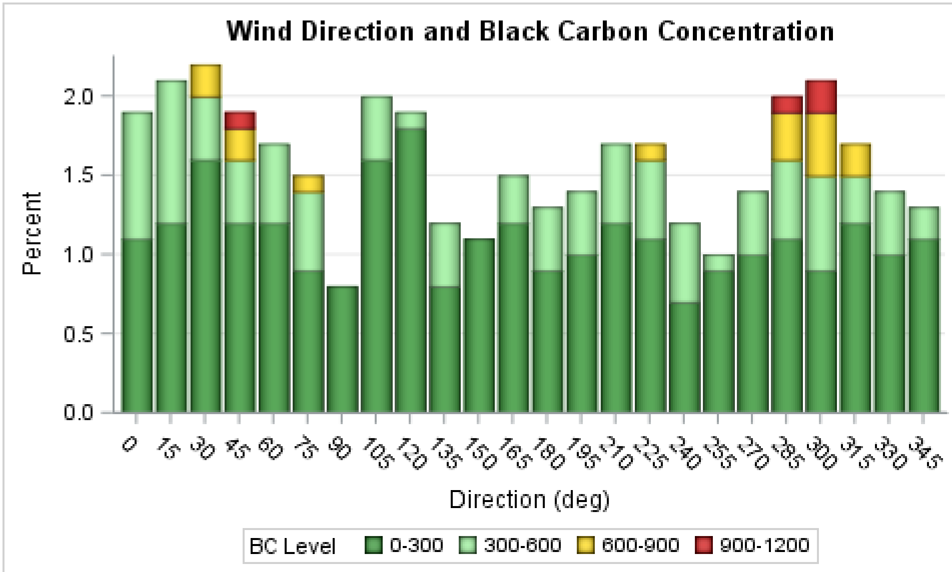

Figure 7. Cartesian Bar Chart

To draw this as a polar graph, you can think of each vertex of the bar element being transformed into polar coordinates as shown in Figure 8.

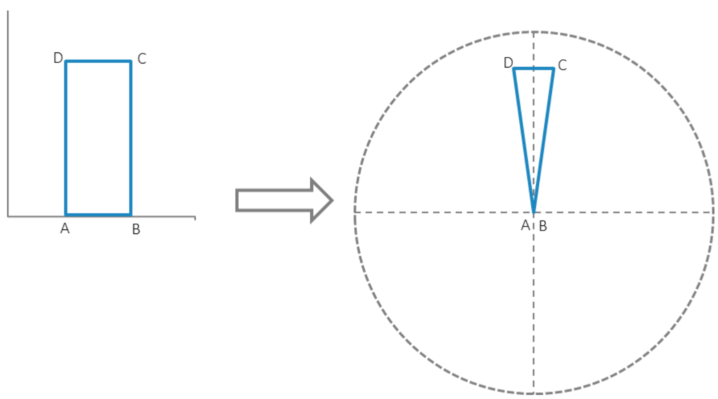

Figure 8. Bar Element Transformation

We choose the angular width of the polar bar element by equally dividing the circle into the number of categories in the data. We then slightly reduce that width so that we get some gap between the elements.

As the lower end of the bar element tends towards r = 0, it collapses into a single point at the origin. You can use the POLYGON statement to draw the transformed “bar” for the polar space. We used this approach to draw a Wind Rose graph as shown in Figure 9.

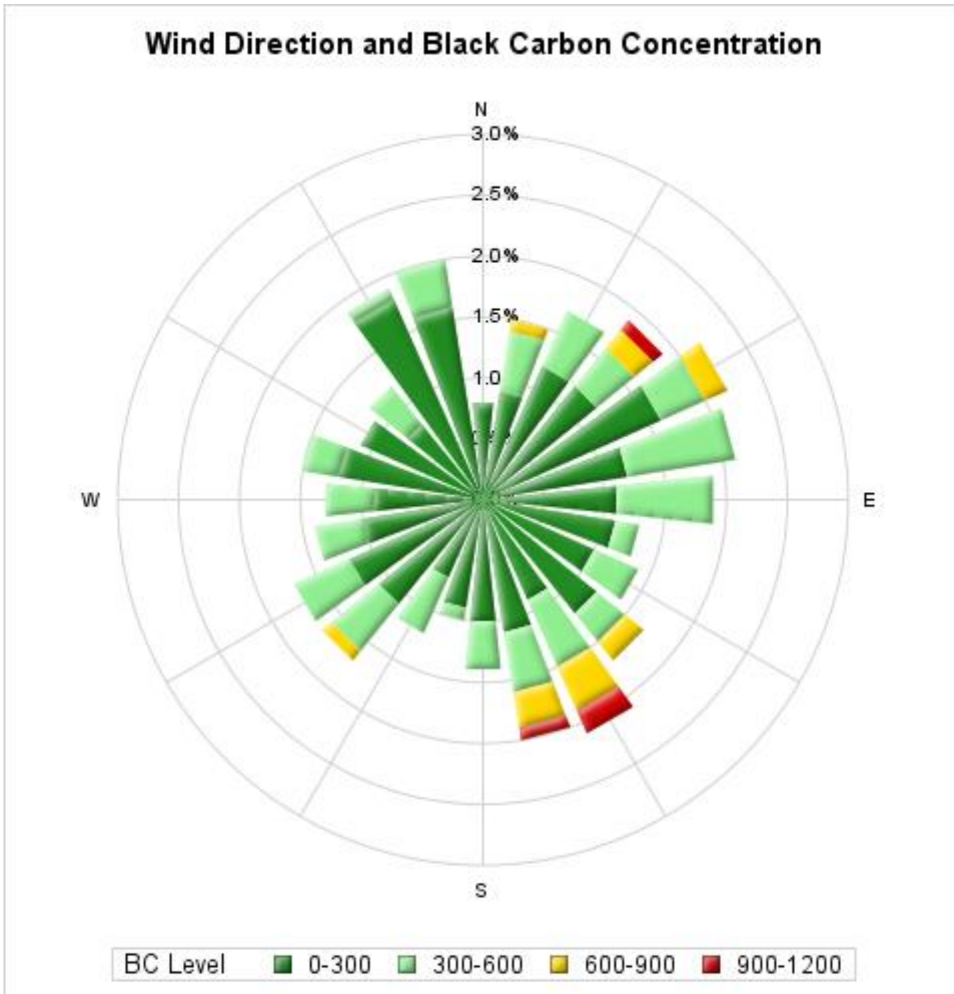

Figure 9. Wind Rose of Black Carbon Concentrations

This technique was also discussed previously in the Graphically Speaking Blog post "Polar Graph – Wind Rose" at http://blogs.sas.com/content/graphicallyspeaking/2016/07/10/polar-graph-wind-rose/.

The main code snippet is shown below:

```
proc sgplot data=WindPolarFinal sgAnno=anno
            aspect=1 noBorder noWall subpixel;
  label concLevel = "BC Level";
  format axisLabel dir. concLevel concRange. ;

  styleAttrs dataColors=(forestgreen lightgreen gold  cxD00000);

  vector x=thX y=thY / lineAttrs=(color=lightGray)
            noArrowHeads; /* Theta axis */
  polygon x=polyX y=polyY id=id / dataSkin=sheen fill noOutline
            group=concLevel name='a'; /* Bar */
  text x=xL y=yL text=axisLabel / textAttrs=(size=7); /* Theta labels */

  %let comOffset=0.02; /* Keep the data area square */
  xaxis display=none offsetMin=&comOffset. offsetMax=&comOffset.;
  yaxis display=none offsetMin=&comOffset. offsetMax=&comOffset.;
run;
```

Axes using Annotations

A variation in this implementation is that we have used annotations to draw the circular grid instead of the ELLIPSEPARM statement that requires the *fourth maintenance release* for SAS 9.4. Using a DATA step, we created the observations for the circular grid using the OVAL function and those for the r labels using the TEXT function. These were saved in a data set named ANNO and then used in the SGPLOT procedure's SGANNO= option. The details of creating the annotation data set are available in the program code file.

You can also create the grids using a SERIES or POLYGON statement if you compute the points with sufficient granularity.

CONCLUSION

We have shown here that you can create polar graphs using the SGPLOT procedure with a few data preparatory steps. You can also add axis-like features using some additional graphical elements. These graphs can also be created in GTL with an equivalent template.

Depending on the nature of your data, a polar graph may be a better visual than the standard Cartesian graph.

RESOURCES

The PDF file of this paper and the SAS code for all the programs are available at the following locations:
Paper: http://support.sas.com/resources/papers/proceedings17/SAS520-2017.pdf
Code: http://support.sas.com/resources/papers/proceedings17/SAS520-2017.zip

REFERENCES

Matange, S., and D. Heath. 2011. *Statistical Graphics Procedures by Example: Effective Graphs Using SAS®*. Cary, NC: SAS Institute Inc.

Matange, Sanjay. 2013. *Getting Started with the Graph Template Language in SAS®: Examples, Tips, and Techniques for Creating Custom Graphs*. Cary, NC: SAS Institute Inc.

Matange, Sanjay. "Graphically Speaking." Available at http://blogs.sas.com/content/graphicallyspeaking . Accessed on February 10, 2017.

Hebbar, Prashant. 2012."Off the Beaten Path: Create Unusual Graphs with GTL" *Proceedings of the SAS Global Forum 2012 Conference*. Available at http://support.sas.com/resources/papers/proceedings12/267-2012.pdf. Code: http://support.sas.com/rnd/papers/sasgf12/Paper267_2012.zip

ACKNOWLEDGMENTS

Thanks to Dan Heath and Lingxiao Li for their helpful comments.

RECOMMENDED READING

- *SAS® 9.4 ODS Graphics: Procedures Guide*

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors:

Prashant Hebbar
SAS Institute, Inc.
prashant.hebbar@sas.com

Sanjay Matange
SAS Institute, Inc.
sanjay.matange@sas.com