SESUG-304-2018

A Map is Just a Graph without Axes

Nat Wooding, J. Sargeant Reynolds Community College, Richmond, VA

ABSTRACT

SAS' ® PROC GMAP can produce a variety of maps of varying complexity but to go beyond the basic capabilities of GMAP, it is necessary to use the ANNOTATE Facility in order to add additional information such as symbols in specific places. Furthermore, there are times that the desired map is simply a sketch of geographically related measurements that need to be displayed in a simulated, not to scale, map. A map is simply a collection of coordinates that are plotted but for which no X/Y axis system is typically shown (although items such as road atlases or military maps or charts may have a grid and axes to help locate specific points of reference). By remembering this, one can sometimes create an embellished map using PROC GPLOT without having to create an ANNOTATE data set. Furthermore, by using GPLOT with the axes, one can locate invalid map coordinates in user created map files. Finally, an example of creating a plotted outline map with dots showing environmental variables using ANNOTATE and PROC GPLOT is offered. ANNOTATE is used in the latter case since it was necessary to dynamically scale the dots that represent the location and magnitude of the plotted values.

PLOTTING ENVIRONMENTAL SAMPLES

SAS' PROC GMAP can produce a variety of choropleth or filled outline maps. However, to show information such as a distribution map of measurements of environmental parameters such as levels of metals, one needs to produce an outline of an area embellished with symbols that show the collection locations and colors that indicate the measured magnitude. Traditionally I have resorted to SAS' ANNOTATE Facility to add the symbols and colors. While this can readily be done with GMAP and ANNOTATE, one still has to learn enough about GMAP and ANNOTATE in order to accomplish the task. However, by using PROC GPLOT, suppressing the axes, crafting Symbol statements properly, and treating our additional plotting points as if they were values of some other "line", one can create a simple and informative map with simpler code.

Some years ago, I was presented with mercury data measured in fish tissue that a colleague had found on a Virginia Department of Environmental Quality (DEQ) website. The current source of the data is https://www.deq.virginia.gov/Programs/Water/WaterQualityInformationTMDLs/WaterQualityMonitoring/FishTissueMonitoring/FishTissueResults.aspx

(Caveat: I was able to use data from this site in 2013 but the original link no longer works and I have not verified that the mercury data are still posted.)

My goal was to produce a map of the Commonwealth showing the collection locations and the maximum measurement at each location.

Skipping over details such as extracting the data from a rather complex Microsoft Excel® spreadsheet, my first tasks involved manipulating and reformatting the data. The first issue was that the data were stored as character strings and values below a certain level of detection (the lowest value that could be measured) had a prepended "<" symbol. The "less than" symbols were removed using the compress function and then an input function read a numeric variable from the remaining character string. Then, the maximum value of multiple measurements at each set of x/y coordinates was found with PROC MEANS.

```
DATA Merc;
SET …
Mercury = INPUT( COMPRESS(cmercury,'<') , 8. );* create a numeric value from the character
variable;
DROP Cmercury;
RUN;

PROC  SORT;
     BY lat long;
RUN;

PROC MEANS NOPRINT DATA = Merc ;
     OUTPUT OUT = Merc   MAX = mercury;
     VAR mercury;
```

```
     BY  lat long;
RUN;
```

Now, since there are numerous levels of mercury readings, we need to scale them in order to have a reasonable number of plot symbols. We create a format and then use this format to create arbitrary scaled values which will be stored as text values. While we are working on these numbers, the coordinates which were in degrees/minutes/seconds are converted to radians in order to be compatible with the coordinates used in the SAS Maps Data sets. The latter step would, of course, be needed when using PROC GMAP with a projected map data set.

```
FORMAT;
  VALUE levels
         .         = ' '
           0 -.29 = '<0.29'
  .29001 -  0.49  = '0.20 to 0.49'
  .49001 -  1.90  = '0.49 to 1.90'
;

DATA PLOT ;
     SET Merc ;
     Level       =  put( mercury , levels. ); * Note that the mercury level is now a
character string;

/* convert degrees to radians */
     X= ( ATAN(1)  /  45  )  *  ( -1 * long );* longitudes are recorded in negative degrees;
     Y= ( ATAN(1)  /  45  )  *  lat;
RUN;
```

We need the map of Virginia so we use one of the SAS maps data sets and combine it with our mercury data. We assign it a value of the variable level in order to later associate it with a SYMBOL statement.

```
DATA Virginia ;
       SET MAPS.STATES ( WHERE ( STATE = 51 ) IN = In1) Plot ;
       LENGTH level $ 15;
       IF In1 THEN LEVEL='Map';
RUN;
```

The data need to be projected so apply PROC GPROJECT.

```
PROC SORT DATA = Virginia;
     BY level  ;
RUN ;

PROC GPROJECT DATA = plot OUT = plot ASIS ;
     ID level ;
RUN;
```

All that we have done so far in terms of manipulating the data would have been necessary if we were to use PROC GMAP. In addition, we would have had to create an ANNOTATE data set with label statements for the ANNOTATE facility to use to create and place the plot symbols for the various levels of mercury. We are now going to simply plot the x/y pairs and the color will show the level of mercury or the outline of Virginia. In my original work, I used an ActiveX driver and html but I am omitting those details for simplicity. Note that the plot statement includes the NOAXES option.

```
PROC GPLOT  ;
    PLOT y*x = LEVEL / NOAXES LEGEND  ;
    LABEL level = 'Levels of Mercury in Fish Tissue (ppm)';
    SYMBOL1  I = NONE  C = CYAN  V = DOT H = .5;
    SYMBOL2  C = RED   V = DOT   H = .5   ;
    SYMBOL3  C = CYAN  V = DOT   H = .5;
    SYMBOL4  C = BLACK V = POINT H = .5  ;
RUN ;
QUIT;
```

The resulting graph is shown in Figure 1.

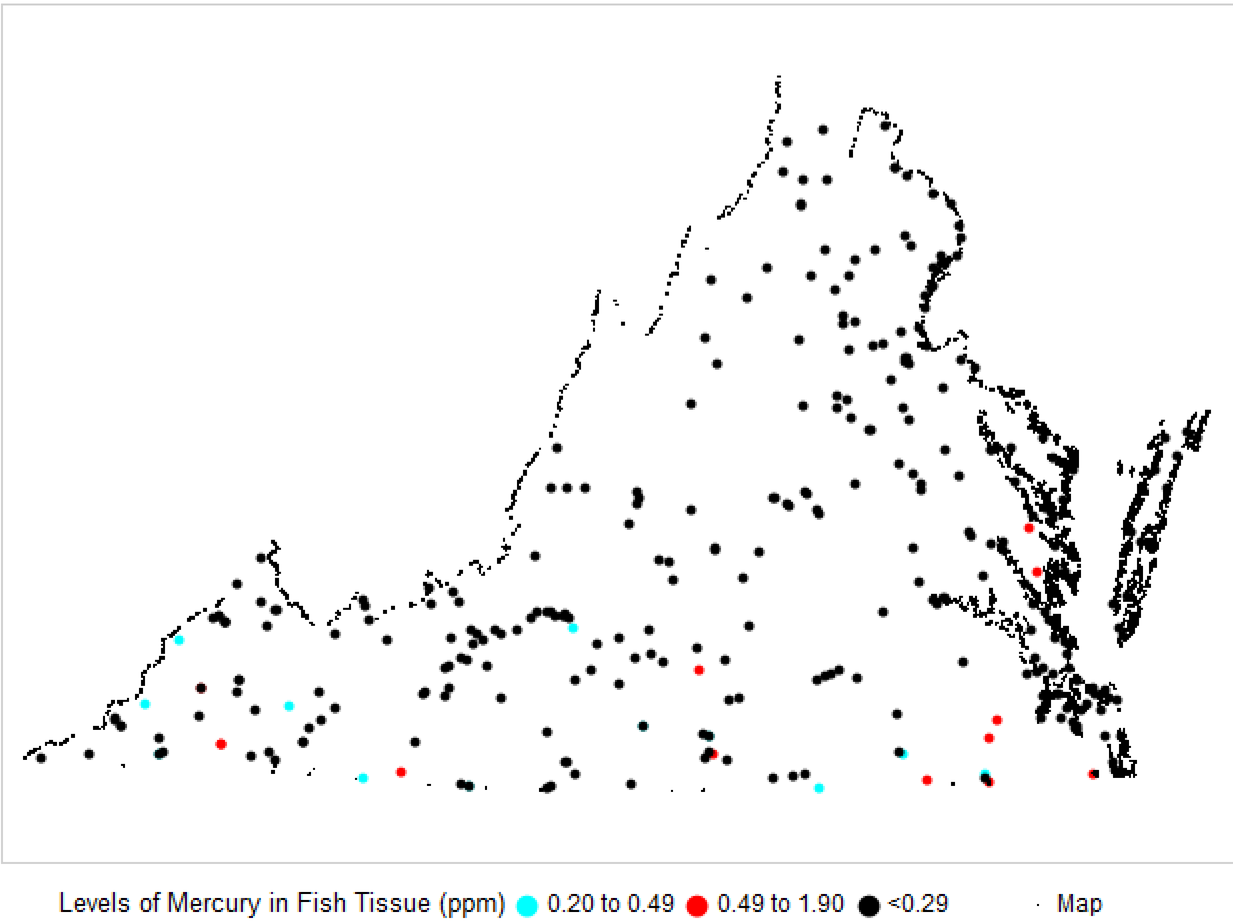

Figure 1. Levels of Mercury Measured in Fish Tissue

This approach eliminates the need to learn ANNOTATE commands as well as the nuances of PROC GMAP and furthermore, the plotting statement is very simple. However, the method is not as useful if data are being plotted in a number of regions whose boundaries are straight lines since the locations of the regions are not always clear due to the lack of intermediate points between the corners of the regions. Consider, for example, the long straight borders between Pennsylvania and New York, West Virginia, and Maryland.

CLEANING DIGITIZED MAP DATA

For a second example of plotting map data, we will consider a situation in which a colleague needed to show the locations of thick patches of a nuisance aquatic weed which grew in shallow water along a lake's shore and caused problems for swimmers and boaters. To assess the extent of the situation, he flew over the lake in a helicopter and marked the weed patches on a map. (This was in the days before global positioning systems became common) and he then used a digitizer to read the coordinates of the boundaries of the weed patches. The digitizing operation sometimes produced erroneous points so the data needed to be cleansed before the final maps could be produced. Three different approaches to identifying the erroneous points will be discussed.

Using PROC GPLOT with Axes
Here, we actually want to see the axes so that the X/Y coordinates of bad points can be found.
The code to produce this is the simplest of PROC GPLOT statements.

```
PROC GPLOT DATA = Weed1;
PLOT Y*X;
SYMBOL1 I = JOIN V = SQUARE;
RUN;
QUIT;
```

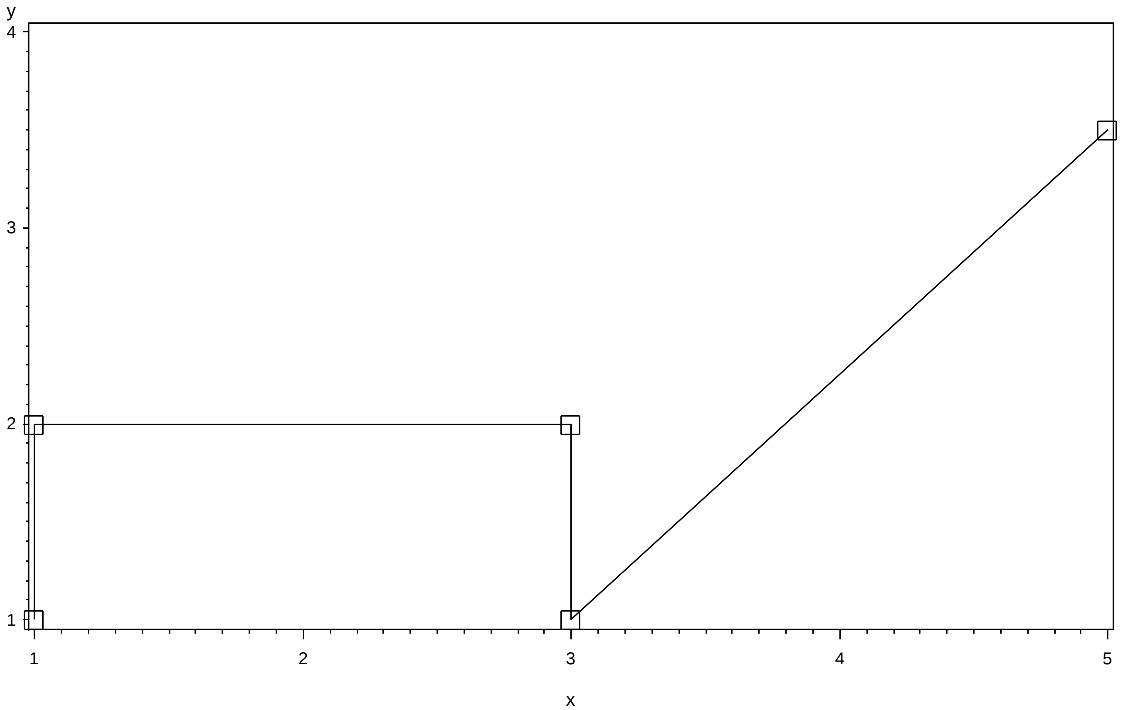

Figure 2. A Simple Set of Map Coordinates With One Bad Point

Once this map is printed, it is a simple task to use a draftsman's right triangle to locate the bad X and Y coordinates on the appropriate axes so that they may be found in the data set and deleted.

For those who have access to SAS/Insight ®, you could display the points as a scatter plot and click on bad points in order to see the particular set of coordinates flagged. Assuming that our coordinates are in a SAS data set, the steps involved in finding the coordinates are:

1) Click in sequence the tabs Solutions/Analysis/Interactive Data Analysis
2) In the box that opens, select the library where the coordinates are stored and then click on the name of the SAS data set and select the Open button.
3) A table view window showing the data will now appear. Click on the heading of one of the columns of coordinates and drag the cursor onto the other column so both are highlighted.
4) Click the Analyze tab and select Scatter Plot from the drop down menu.
5) In the plot window that now appears, click on the bad point(s). The particular observation(s) will now have the observation number highlighted.

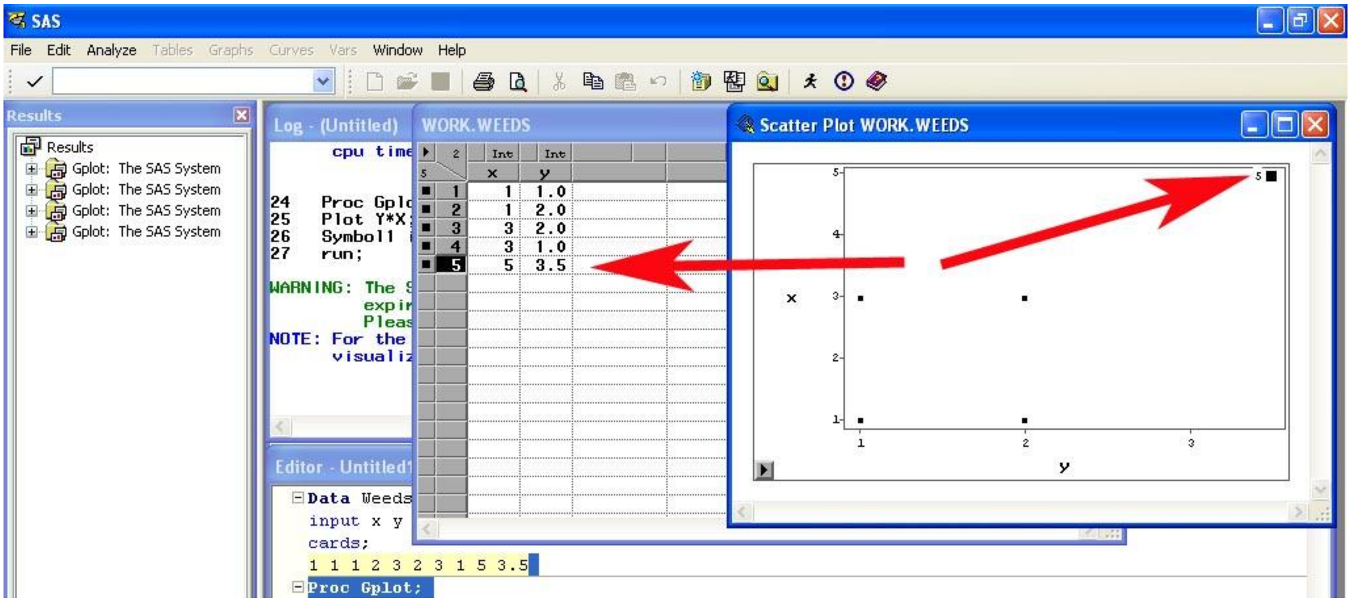

Figure 3. Using SAS/Insight to Locate Bad Coordinates

A third method would be to use tool tips so that if you hovered over a questionable point in a plot, the coordinates would appear. In the interest of space and time, I will not show this method.

CREATING A SCHEMATIC MAP

In the final case, the organization that I was supporting was conducting environmental sampling in a small river in an area where there were a number of former strip mines. Streams that drained the areas of these mines often had high concentrations of dissolved ions and we needed to show the values of various parameters in the river as well as in the tributaries before they entered the river. The river was about 10 miles long and in many cases the measures were taken a mile or more apart so a map drawn to scale and of a readable size would have been quite unwieldy. Hence, a simple not-to-scale sketch map was in order. Basically, we needed a plot of the points with the X axis representing distance and the Y position indicating the location in the river where the reading was taken (left bank, middle, or right bank). We wanted to use a “Dot Plot” to show the relative magnitude of the measurements and also the numeric values. For the example here, we are working with the parameter Conductivity which is the inverse of electrical resistance: the higher the conductivity, the easier electricity flows through water due to the greater the concentration of dissolved ions.

Figure 4 shows a sample plot.

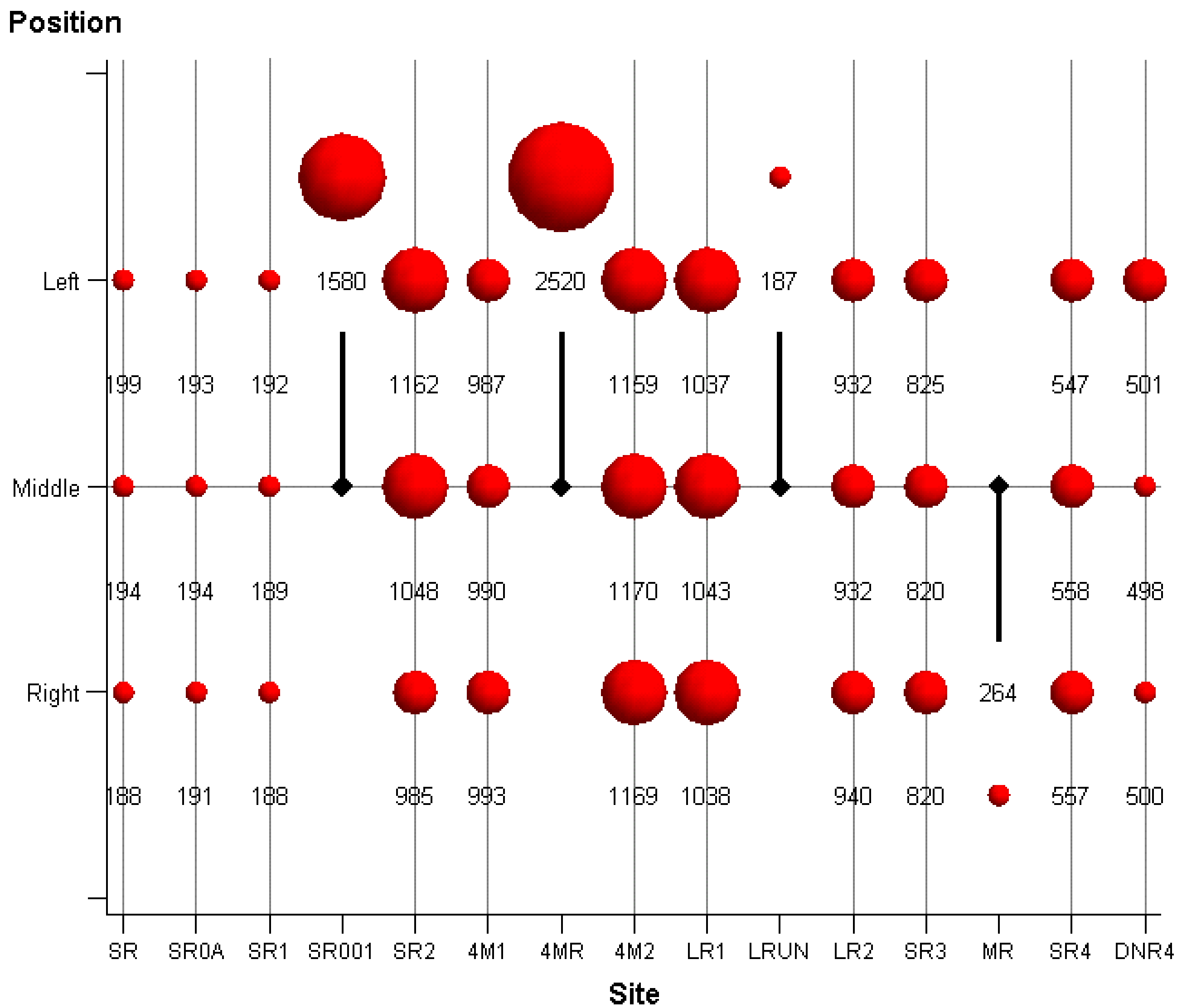

Figure 4. Dot Plot Showing Conductivity Measurements of an Acid Mine Drainage Impacted River. The Heavy Vertical Lines Indicate Streams Entering the River. The Size of the Dots Indicates the Relative Value of the Conductivity Reading.

The following code has been simplified for publication. Please contact me if you have any questions concerning it.

The code for producing this plot started with steps that used IF statements to assign Y values according to the position in the river of the measurements. Also, the site name was used to create an X value called order.

We then needed to create an ANNOTATE data set that would create the labels that would print our dots and show the locatons of the tributary streams. Formats were used to scale the values of our measurements and a put statement was used to assign the scaled value to a new variable.

```
PROC Format;
INVALUE Cond
       0 -  500 = 1
     501 -  999 = 2
    1000 - 1499 = 3
    1500 - 2000 = 4
    2001 - 2600 = 5;
RUN;

DATA Anno; * THE CONTENTS OF THIS DATA SET WILL CREATE THE DOTS ON THE GRAPH AS WELL AS
             THE LABELS AND ARROWS;
   SET B;

*** first do the dots;
   SIZE = INPUT( Conductivity , cond. );  * This will control the size of our dot;
   X    = order; * Order controls the positions along our X axis;

KEEP WHEN XSYS YSYS POSITION COLOR FUNCTION X Y ORDER SIZE SITE CONDUCTIVITY TEXT STYLE
SURVEY;
LENGTH FUNCTION COLOR TEXT $8;
RETAIN WHEN 'A' XSYS YSYS  '2' POSITION   '5' TEXT   'CIRCLE'  COLOR 'RED' FUNCTION 'SYMBOL';
OUTPUT;

FUNCTION = 'LABEL'; * NOW ADD ANNOTATE STATEMENTS THAT WILL LABEL THE CONDUCTIVITY LEVELS ;
COLOR       = 'BLACK';
TEXT         =  LEFT( PUT( CONDUCTIVITY , $4. ));
OUTPUT;

    * FOR TRIBS, DRAW A LINE FROM THE LOCATION OF THE LABEL TO MIDSTREAM TO INDICATE INFLOW;
IF ORDER IN( 4 , 7 , 10 ,13 ) THEN DO;
IF ORDER=13 THEN DIR =  1; /* MILL RUN (MR) FLOWS FROM RIGHT, REST FROM LEFT */
     ELSE DIR= -1;

FUNCTION='MOVE';* MOVE THE POINTER TO THE POSITION WHERE THE LINE WILL END;
OUTPUT;

SIZE=3;
FUNCTION='DRAW';* DRAW THE LINE;
OUTPUT;

IF SITE =: ‘MR' THEN TEXT='D9'X;*THESE ARE THE ARROWHEADS AT THE MOUTHS OF THE TRIBUTARIES;
     ELSE TEXT = 'DA'X;
FUNCTION= 'LABEL';
STYLE = 'WINGDING';
OUTPUT;
END;
RUN;

PROC GPLOT DATA = B  ;
    PLOT Y*ORDER / VREF=2 HREF=1 2 3  5 6  8 9 11  12    14 15 NOFRAME
         HMINOR   = 0
         VMINOR   = 0
         HAXIS      = 1 TO 15 BY 1
         VAXIS      = 0 TO 4 BY 1
         ANNOTATE = ANNO ;

LABEL Y='POSITION'  ORDER='SITE';
FORMAT ORDER SITES. Y POS. ;
SYMBOL1 V=NONE;
RUN ;
QUIT;
```

The ANNOTATE code shown above is a little complicated and required some tweaking in order to not have labels and dots collide but the resulting plots helped visualize the impact of the acid mine drainage on the water quality in the river.

CONCLUSIONS

A map is simply a plotted set of points, some of which are connected by lines. PROC GMAP is a good tool for producing maps and the ANNOTATE facility can be used to add features to a map. However, one can also produce maps using PROC GPLOT and in some cases, an embellished map can be produced without needing the ANNOTATE facility. In addition, simply plotting raw map coordinates can help clean up a bad map data set. And, sometimes a combination of PROC GPLOT and ANNOTATE can produce a schematic representation of a data set.

REFERENCES

SAS 9.2 Language Reference: Dictionary, Fourth Edition, SAS Institute, Cary, N.C. 2011

SAS/GRAPH 9.3: Reference, Second Edition, SAS Institute, Cary, N.C. 2011 Other editions of these manuals would be equally useful)

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Please feel free to contact the author at:

Nat Wooding
Office of Institutional Effectiveness
J. Sargeant Reynolds Community College
1651 East Parham Road
Richmond, VA 23228

Email: Nwooding@reynolds.edu
Or Nathani@verizon.net
Cell 804-357-6655