ENHANCING THE VISUAL IMPACT AND CLARITY OF GRAPHICS OUTPUT USING THE CUSTOMIZING POWER OF THE ANNOTATE= DATA SET

Karen Lacy Helsel

The SAS/GRAPH* ANNOTATE= data set is the most powerful and flexible tool available to SAS/GRAPH programmers for customizing their graphics output. Through the use of this feature, graphics output can be made more readable as well as more explanatory in nature. The ANNOTATE= data set also allows presentation in the exact format desired for an application when the default format may be inappropriate or inadequate to fully describe the data in the manner desired.

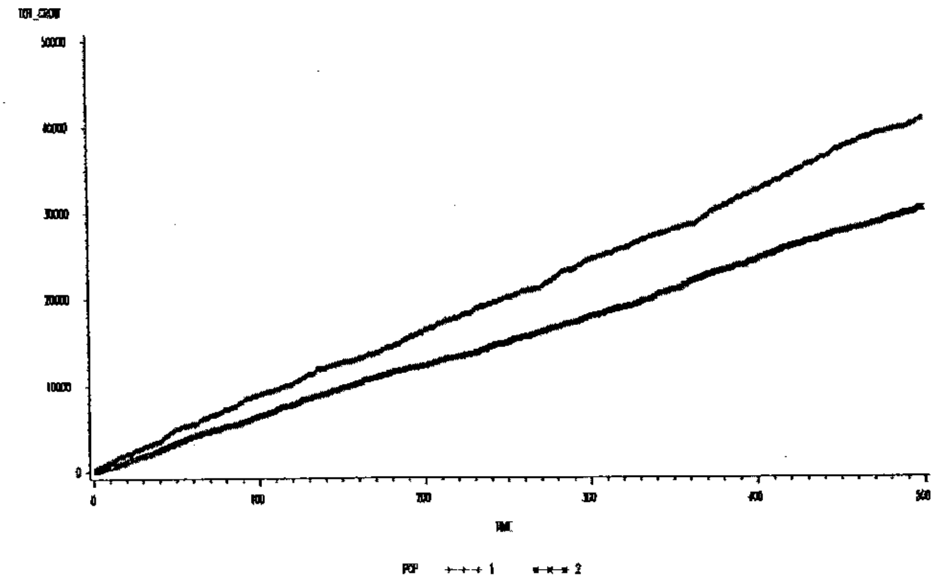
The examples presented herein are designed to introduce the programmer to the basic features of the ANNOTATE= data set. They also offer a sampling of its diverse capabilities within the context of the enhancement of the default output of certain procedures, namely PROC GPLOT and PROC GMAP, although the techniques presented can be applied to any SAS/GRAPH procedure. All of these applications use the actual data values to determine details such as the placement of labels, and the text they contain, so they are data driven and therefore completely portable.

The first application is fairly straightforward. Figure 1a shows the default output produced by PROC GPLOT of the rate of growth over time, separated into two distinct populations. Each individual observation is marked by a symbol, either a '+' or an 'X', depending upon the population to which it belongs. It is obvious from this example that with any significant number of observations it becomes virtually impossible to determine which symbol identified in the legend refers to which line.

One option fully available without accessing an ANNOTATE= data set is the suppression of the individual symbols, thus plotting only the lines themselves. However, a more complicated example where there are numerous lines of similar line patterns renders this option less than successful. Also, if one is constrained to a monochrome environment, such as publications, offering different colors is not a viable option.

The graphics output of Figure 1b is only one of the many possible solutions to this problem using the Annotate= data set. Plotted here are lines with differing line patterns which have been enhanced through the addition of descriptive labels at their endpoints. Labels such as the ones placed on this simple example become extremely valuable tools to help refer back to the legend, or to enhance the information provided in the legend. The code to create Figure 1b is presented on the following page:

Figure 1a DEFAULT PROC GPLOT OUTPUT, OMITTING ALL CUSTOMIZING OPTIONS



```
DATA LABELS;
  SET FINAL;
  BY POP DESCENDING TIME;
  IF FIRST.POP THEN DO;
    FUNCTION = 'LABEL';
    XSYS='2';  /* ABSOLUTE DATA VALUES */
    YSYS='2';  /* ABSOLUTE DATA VALUES */
    STYLE='TRIPLEX';
    SIZE=1.5;
    X=TIME;     /* FINAL VALUE OF X-AXIS VAR. */
    Y=TOT_GROW;/* FINAL VALUE OF Y-AXIS VAR. */
    IF POP=1 THEN DO;
      POSITION = '2'; /* CENTERED ABOVE */
      TEXT = 'POP # 1';
    END;
    ELSE IF POP=2 THEN DO;
      POSITION = '8'; /* CENTERED BELOW */
    TEXT = 'POP # 2';
    END;
    OUTPUT;
  END;

  PROC GPLOT DATA=FINAL;
    PLOT TOT_GROW*TIME=POP / ANNOTATE=LABELS;
    SYMBOL1 L=2 I=JOIN C=BLACK;
    SYMBOL2 L=1 I=JOIN C=BLACK;
    TITLE;
    LABEL TOT_GROW = 'AGGREGATED GROWTH';
    FORMAT POP POPFMT. ;
```

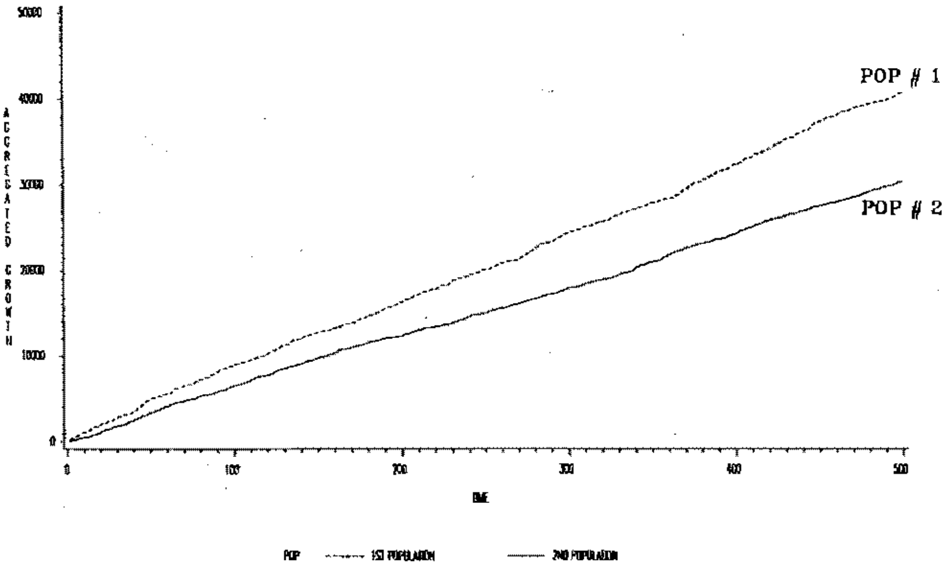
Unlike the example presented in the SAS/GRAPH manual, version 5 (page 150, example 9), coding for this ANNOTATE= data set does not require prior knowledge of the data values within the data set. Here, the data has been previously sorted by two of the three variables that will ultimately be plotted, POPULATION and descending TIME. Thus, keying on the "FIRST." values captures the "final" observation within each population, instead of relying upon a data dependent condition to be identified. Once these observations are identified, they are used to create the observations to be output to the eventual ANNOTATE= data set.

The X value and the Y value come directly from the variables TIME and TOT_GROW contained within the "FIRST." observation. These define exactly the coordinate location to place the information, so XSYS and YSYS direct the SAS software to interpret these values as absolute data values. Other interpretations available for these variables are a percentage of the data, or a percentage of the window space. In this case these are obviously inappropriate choices -- the TIME variable in this example has a maximum value of 500, and 500 interpreted as a data percent would place the information at 500% of the final observation. Likewise, 500% of the available window space is impossible to access and so is an illegal argument.

The value of the function variable is an obvious choice. To place text at certain coordinates choose 'LABEL'. Finally the TEXT variable is assigned a value, and this is placed above or below the previously defined location with the value assigned to the POSITION variable.

The final PROC GPLOT syntax is straightforward, with only the ANNOTATE= data set requiring any prior coding before being invoked.

Figure 1b PLACING IDENTIFYING LABELS AT KEY POINTS WITHIN PROC GPLOT OUTPUT



The next example deals with the following situation. The ultimate objective is to convey certain statistical information about the data that is not immediately available from a basic plot of the raw data. There are several options. One can place this statistical information as numbers contained in the title, if it is hard coded after an initial pass of the data, or it could be passed to the title as a macro variable reference. Another choice leads to a graphic solution, which presents the information in an immediately digestible format, as seen in Figure 2a.

The audience has an immediate grasp of the implications of the statistics and how they relate to the data distribution without having to scan any text or mentally visualize its placement.

A portion of the code to generate Figure 2a is presented below:

```
PROC MEANS NOPRINT DATA=INFILE.INDATA;
  FREQ NUMW_VAL;
  VAR TEST_VAL;
  OUTPUT OUT=MEAN_DAT
         MEAN=MEAN_VAL
         STD=STDV_VAL;

PROC MEANS NOPRINT DATA=INFILE.INDATA;
  VAR NUMW_VAL;
  OUTPUT OUT=MAX_DAT
         MAX=NUMW_VAL;

DATA ANNOSTAT;
  MERGE MEAN_DAT MAX_DAT;
  XSYS = '2';/* ABSOLUTE DATA VALUES */
  SIZE=1.5;
  POSITION = '2'; /* CENTERED 1 CELL ABOVE */
  FUNCTION = 'MOVE';
  X = MEAN_VAL - STDV_VAL;
  YSYS = 'I';/* PERCENT OF THE DATA RANGE */
  Y = 0;
  OUTPUT;     /* MOVES TO -I STD DEV ON X-AXIS */
  FUNCTION = 'DRAW';
  YSYS = '2';/* ABSOLUTE DATA VALUES */
  Y = MAX_NUM + 2;
  OUTPUT;     /* DRAWS A VERTICAL LINE */
  FUNCTION = 'LABEL';
  TEXT = '-1 STD. DEV.';
  OUTPUT;     /* PLACES LABEL ABOVE LINE */

 PROC GPLOT DATA=INFILE.INDATA GOUT=OUT;
  AXISI LABEL=('DIST. OF VARIABLE OF INTEREST'
              F=TRIPLEX
              H=I.5)
  ORDER=0 TO I00 BY 1
  MINOR=1;
  AXIS2 LABEL=('POPULATION COUNT'
              F=TRIPLEX
              H=1.5
              ANGLE=90
              ROTATE=0);
  PLOT NUMW_VAL*TEST_VAL / HAXIS=AXISI
                           VAXIS=AXIS2
                           ANNOTATE=ANNOSTAT;
  SYMBOL1 I=JOIN;
  TITLE;
```

This data has been collapsed to contain one observation per value of the variable "TEST_VAL", with another variable (NUMW_VAL) containing the number of observations in the original data that had the TEST_VAL value. (This data set will produce the default portion of the graph). Since the data is presented in this manner, two separate PROC MEANS procedures are required - one weighted by the raw count to determine the mean of the original data, and one on the variable holding the raw count itself to find the maximum value that will appear as a data point on the Y-axis. The output is then directed to WORK data sets instead of being allowed to print, thus providing access to create the ANNOTATE= data set.

Next, a one-to-one merge of the single records output above from each of the PROC MEANS procedures is accomplished. One begins by defining the variable values that remain constant for each observation produced, such as XSYS, STYLE, SIZE and POSITION. Not all of these variables are needed for each of the functions that we will choose, so those observations will simply ignore this as extraneous information once it is passed to PROC GPLOT as an ANNOTATE= data set.

The first observation that is output moves to a point on the X-axis that is computed from the statistics generated above. The YSYS value of '1' tells the SAS software to interpret the Y value as a percentage of the Y-axis, so here 0% is identically the X-axis line itself. Since this data was randomly generated, the Y variable can take on any possible value. The scale on the Y-axis is also allowed to default within the PROC GPLOT, so a Y-axis value of 0 may not actually exist. This is the reasoning behind the use of a percentage of the Y-axis to identify the X-axis line, instead of a Y data value of 0.

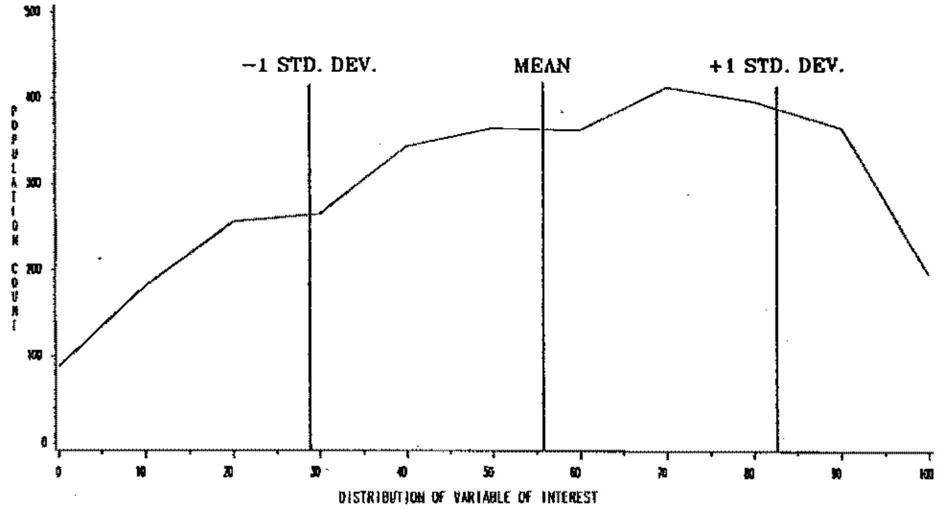
The next observation output to the eventual ANNOTATE= data set draws a line starting from the position on the X-axis just moved to, and ending up at 2 above the maximum Y value that we found. Here the Y value is interpreted as a strict data value (YSYS = '2'), since the Y value being keyed upon will be generated by the plot.

Finally, the LABEL function seen before is used to place text centered above the coordinates from the previous observation. All of the other variables used in the first example by population where the function was also LABEL have been defined elsewhere in this data step prior to this point, so the only remaining variable needing definition is the TEXT value. For this application, the actual statistical numbers may be more desirable than labels. Simply define the TEXT variable to hold a character conversion of the numeric statistical numbers - possibly the result of the SAS PUT function. Again, no prior knowledge of the data is required because nothing is hard coded as a value that must exist in the data.

This code is reproduced almost line for line within the same data step to create observations that generate the identifying lines at the mean, and the plus 1 standard deviation value on the X-axis. The only changes are in the TEXT variable, and in the calculation of the X value.

In this example as before, the final PROC GPLOT syntax is deceptively straightforward. All of the bells and whistles are coded before arriving at this point and are simply invoked by using the ANNOTATE= option that specifies the data set created above.

Figure 2a GRAPHICALLY DISPLAYING POPULATION STATISTICS ON PROC GPLOT OUTPUT



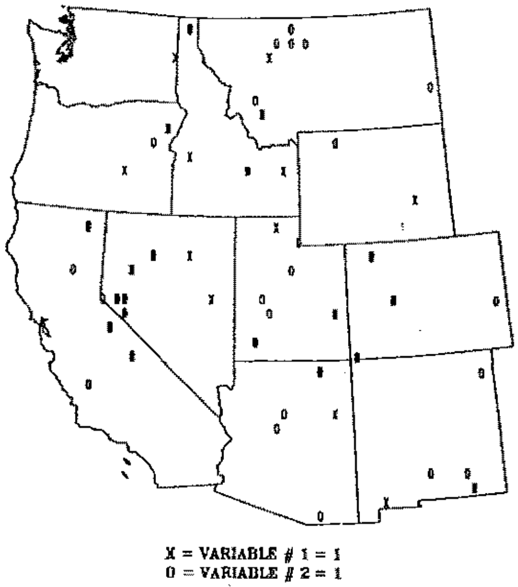
The final example is more advanced and is rather specialized in the output it produces (see Figure 3a), although the concept of plotting multiple variables contained within the data set as different symbols on the graphics output is not specialized in the least. The default output is inappropriate and actually quite misleading in this case. Within PROC GMAP, the ID variables available to determine the level of detail are state, county, zip code, city, etc. The data for this example contain only the state, latitude and longitude as geographic identifiers, so the default output choice is a state level plot. This results in all of the states having the same pattern completely filling their borders, since all of the states have at least one observation in the response data set. It is therefore necessary to key on the latitude and longitude variables available to place information within PROC GMAP.

The SAS code below was used to generate this output.

```
DATA ALL;
  SET OURLOCAT(IN=OURDAT)
      MAPS.STATES(IN=SASDAT);
  IF PUT(STATE,STFMT.) = 'WEST COAST';
  IF OURDAT THEN FLAG=1;
  IF SASDAT THEN FLAG=2;

PROC GPROJECT DATA=ALL OUT=PROJECT;
  ID STATE;

DATA PROJOUR PROJSAS;
  SET PROJECT;
  IF FLAG=1 THEN OUTPUT PROJOUR;
  ELSE OUTPUT PROJSAS;

DATA ANNOMAP;
  SET PROJOUR;
  FUNCTION = 'LABEL';
  XSYS = '2';  /* ABSOLUTE DATA VALUES */
  YSYS = '2';  /* ABSOLUTE DATA VALUES */
  SIZE = 1;
  POSITION = '5';/* CENTERED ON THE COORDINATE */
  IF VAR_1 = '1' THEN DO;
    TEXT = 'X';  /* OUTPUTS 'X' IF VAR. OF */
    OUTPUT;      /* INTEREST # 1 IS TRUE   */
  END;
  IF VAR_2 = '1' THEN DO;
    TEXT = 'O';  /* OUTPUTS 'O' IF VAR. OF */
    OUTPUT;      /* INTEREST # 2 IS TRUE   */
  END;

PROC GMAP DATA=PROJSAS
          MAP=PROJSAS
          ALL
          GOUT=OUT;
  CHORO STATE / NOLEGEND
                CEMPTY=BLACK
                COUTLINE=BLACK
                ANNOTATE=ANNOMAP;
  ID STATE;
  PATTERN1 V=E;
  TITLE;
  FOOTNOTE;
```

Figure 3a

PLOTTING MULTIPLE VARIABLES AS INDIVIDUAL POINTS
ON AN EMPTY PROC GMAP OUTLINE



First the data is projected with the SAS data set that will be used to generate the outlines - the SAS STATES data set. This projection scales the latitude and longitude values contained in the variables X and Y to X and Y values that are comparable to the scale of the SAS STATES data set. Without this step, the data points and the states' outlines have no relationship - half the data could end up in the Pacific Ocean! The data is then extracted from the projected combined data, and is used to create the familiar Annotate= variables in the next data step.

The first section again defines variables that remain unchanged for each observation produced. (This is a convention I use, not one that is dictated by the fact that an ANNOTATE= data set is being created.) Again the function value of LABEL is chosen, although SYMBOL would work just as well, due to the fact that the eventual text output consists of a single character. One thing to note, with the SYMBOL function, one can choose from special symbols defined in the V= section of the PATTERN options used in basic SAS/GRAPH output, such as diamonds, spades, and hearts.

Since the projected data values are what is plotted by the ANNOTATE= data set in this example, the XSYS and YSYS values, as well as the POSITION variable, force the text to be placed at these exact X and Y coordinate values contained in the X and Y values from the projected data.

Now the variables of interest are keyed upon. In this example, there are 2 flags per observation, either of which (or both, or neither) could be set per observation. If the first variable is "true", an 'X' is output, and likewise if the second flag is "true", an '0' is output. Thus, one incoming observation could generate multiple ANNOTATE= observations if both variables of interest meet the criteria. This appears on the output as both generated symbols occupying the same space on the graph, so be careful and choose the symbols wisely!

This time the final PROC GMAP syntax differs a bit from the default. The procedure is invoked, and the SAS STATES data set that was subset previously is passed as the MAP= data set. Instead of passing the projected data as the response DATA= data set, again pass the subset STATES data set. This means that the default portion of the procedure is not processing any of the data at all. Since the pattern statement specifies "empty" , this generates the outline of the states of interest, and the procedure then enhances this with the information passed in the ANNOTATE= option.

The output shows two pieces of information for each of the original observations, either by the presence of a symbol or, as equally informative in certain cases, by the absence of a symbol.

In these three examples, a very few basic variables and even fewer function values have been used to produce highly informative output that enhances the default of graphics procedures. In addition, the ANNOTATE= data set has been created from the data itself, from information about the raw data, and has even replaced the raw data, in some sense, within the graphics procedure.

We have explored, within the limits of this paper, basic and slightly more advanced applications of the Annotate= data set and its power as a customizing tool. These concepts can be easily applied to many varied situations, and should be evidence of the flexibility, potential power, and the relative ease of use, which are characteristic of the Annotate= data set.

Acknowledgements

The author would like to thank William E. Helsel for his invaluable assistance in preparing this paper, Luanne Reeves for her contribution in developing the abstract, and IMS, Inc. for their role in its final appearance in these proceedings.

The author may be contacted at the following address:

Corporate Cost Management
9039 Shady Grove Court
Gaithersburg, MD 20877

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

References

SAS Institute Inc. SAS User's Guide: Basics, Version 5 Edition. Cary, NC: SAS Institute Inc., 1985.

SAS Institute Inc. SAS/GRAPH User's Guide, Version 5 Edition. Cary, NC: SAS Institute Inc., 1985.