

Paper 7-26

Presentation Graphics Using the ANNOTATE Facility

Joseph L. McCormick, Trilogy Consulting, Denver, CO

ABSTRACT

Since its introduction, users have sought ways to create specialized graphic output with the SAS/GRAPH product. Creative manipulation of data can often produce the desired picture, but the Annotate facility permits you to customize any SAS/GRAPH procedure output to enhance your graphic presentation. This presentation will demonstrate how to program the Annotate facility using simple examples, and it will show the many benefits of using Annotate in a production environment. This tutorial will show actual SAS code and provide many examples of possible Annotate applications that can turn your information graphics into presentation graphics.

INTRODUCTION

The Annotate facility lets you customize SAS/GRAPH procedure output using variables and values in non-graphic SAS data sets. The facility is capable of defining polygons, labeling specific points, placing any text on the page, and so on. It adds to existing procedures the capability of producing custom graphic output in a DATA step.

This paper is intended to be an overview of the Annotate facility and its capabilities. Complete documentation is contained in the Volume 1 of the SAS/GRAPH User's Guide.

THE PROBLEM

Since the introduction of the SAS/GRAPH product in 1980, thousands of organizations worldwide have been producing business graphics with the SAS System. The standard graphics procedures satisfy the great majority of users' requirements for information graphics, but more and more people want to make modifications to obtain a specific look. The turn to presentation graphics requires a system that will let you place additional graphics on a given plot or chart.

Until the Annotate facility, customization of graphs was tedious at best. Clever DATA steps were created, or many attempts made until the proper "move" and "draw" combinations were attained. These methods gave a wide range of customization, but one small change in the data could necessitate several more attempts. The more complex the application, the less likely creative programming would obtain the desired look.

THE ANNOTATE FACILITY

The Annotate facility is a command driven graphics customization tool. The commands are stored as observations in a regular SAS data set. Each observation is a complete command, and special variables are used in these Annotate data sets to convey information. This data set is then specified in the ANNOTATE= option in a graphics procedure, such as,

```
PROC GCHART DATA = sample.data
  ANNOTATE = company. logo;
  VBAR var1 / ANNOTATE = chart.detail1 options;
  BLOCK var2 / ANNOTATE = chart.detail2 options;
```

The immediate uses of the Annotate facility center on accurate data dependent positioning of text labels. Its many other uses include:

- Placement of names and areas on maps
- Highlight data minimum, maximum or inflection points
- Draw lines between points
- Define polygons for added information
- Create special presentation graphics

Because Annotate commands are specified in a SAS data set, the facility is flexible enough that an entire graphics procedure may be simulated with DATA step coding mechanisms. Using clever SAS code lets you create custom graphs in a production environment without a programmer changing any code. This also eliminates using PC graphics products for some projects.

ANNOTATE VARIABLES

The Annotate facility is directed by a series of observations and variables contained in a SAS data set. Observations are the commands, and the variables convey the information in the command. These special variables have reserved names that indicate their use. The following is a discussion of some of these variables.

FUNCTION

This character variable indicates which Annotate feature you want to perform. Below is a list of some of the available Annotate functions and a short description of each.

MOVE - Permits movement to a specific point on the graphics page without drawing a line. MOVE is most often used to prepare for a DRAW function.

DRAW - Draws a line on the graphics page. You have control over color, style, and thickness of the drawn line.

LABEL - Places text on the graphics page. You can specify the color, size, font, base angle, rotation and position of the characters displayed.

BAR - Makes a bar or rectangle on the graphics page. You can control the color, pattern of fill, size, line style and thickness of the edge lines.

PIE - Draws pie slices on the graphics page. You can control the color, fill pattern, arc angle, radius, and edge lines of the slice to be drawn.

Additional basic and advanced functions are available for designing polygons, including comments in your Annotate data sets, placing special symbols on your graph, and drawing frames round specified display areas.

X and Y

X and Y specify the coordinates on the graph where the FUNCTION will be applied. The X variable, or horizontal coordinate, and the Y variable, or vertical coordinate, can be used to translate/transform input values to the appropriate screen locations.

XSYS and YSYS

These are character variables that define the area and coordinate system used by the X and Y variables to display the Annotate information. The available coordinate systems can be classified as absolute or relative to determine location. The coordinate units can be classified as data percent, data value, screen percent, screen value, window percent, or window value.

The screen systems use the entire graphics area to determine location. Users of the NOTE statement are already familiar with the screen value (4) system. The display area is divided into grid cells depending on the HPOS and VPOS values to determine screen size. The screen percentage (5) system

divides the area into a more constant 0 to 100 field. The screen percent system helps guarantee independence of Annotate generated graphics.

For example, on a terminal with HPOS=80 and VPOS=32, a coordinate x=40 in screen value is equivalent to x=50 in screen percentage units and produces a location in the center of the display area. If we now move to a device with HPOS=160 and VPOS=80, the x=40 screen value system "changes position" while the screen percentage system maintains its position in the center of the screen.

Window systems and screen systems are different in only one aspect. The screen systems ignore the area occupied by TITLE and FOOTNOTE statements and define the entire screen area. Window systems recognize the area occupied by TITLE and FOOTNOTE statements and restrict their activity to the area enclosed by the TITLES and FOOTNOTES. If no TITLE or FOOTNOTE statements are in use, the window and screen systems are identical.

The data systems use actual data values, or a percentage of the data area, to specify their coordinates. The data system graphic area is the area that is bordered by the horizontal and vertical axes (in PROC GMAP, the range of map coordinates) specified in the procedure step. These systems offer the greatest flexibility to produce customized graphics that are data dependent and device independent.

Other Variables

Additional variables can be supplied to enhance most functions. Color, line weight, line style, font, or pattern are some of the descriptive variables that can be used.

DEMONSTRATION

To illustrate the use of the Annotate facility suppose you want to draw a box and put the text "SUGI 26" in it. You decide to use the MOVE and DRAW functions for the box and the LABEL function for the text inside the box. The following code will produce the output in Figure 1:

```
DATA BOX;
  LENGTH FUNCTION COLOR STYLE $ 8;
* Draw the box;
  COLOR='RED'; XSYS='3'; YSYS='3';
  FUNCTION='MOVE'; X=30; Y=30; OUTPUT;
  FUNCTION='DRAW'; X=70; Y=30; OUTPUT;
  FUNCTION='DRAW'; X=70; Y=70; OUTPUT;
  FUNCTION='DRAW'; X=30; Y=70; OUTPUT;
  FUNCTION='DRAW'; X=30; Y=30; OUTPUT;
* Put text inside the box;
  COLOR='BLUE'; TEXT='SUGI 26'; STYLE='SWISS';
  SIZE=3; FUNCTION='LABEL'; X=50; Y=50; OUTPUT;
PROC GSLIDE FRAME ANNOTATE=BOX; RUN;
```



Figure 1

In the next example, we want to identify each line on the plot by some other means than the legend. We can use the Annotate facility to place the city name on the first point of its line. The following code will produce Figure 2:

```
DATA TEMPS;
* Read monthly average temperatures;
  LENGTH CITY $ 10;
  INPUT MONTH CITY $ HIGH;
  DATALINES;
*** input data lines go here ***
PROC SORT; BY CITY MONTH;

DATA LABELS; SET TEMPS; BY CITY;
* Create labels for each city name;
  LENGTH TEXT $ 10 FUNCTION STYLE $ 8;
  IF FIRST.CITY THEN DO;
    FUNCTION='LABEL';
    STYLE='SWISS';
    XSYS='2'; YSYS='2';
    TEXT=CITY;
    X=MONTH; Y=TEMP;
    IF CITY='Phoenix'
      THEN DO; POSITION='E';
        COLOR='GREEN'; END;
      ELSE DO; POSITION='2';
        COLOR='RED '; END;
    OUTPUT;
  END;
RUN;

PROC GPLOT DATA=TEMPS;
* Plot the temperature data with city labels;
  PLOT HIGH*MONTH=CITY / ANNOTATE=LABELS
  NOLEGEND HAXIS=0 TO 13
  VAXIS=40 TO 110 BY 10;
  FORMAT MONTH ID.;
  SYMBOL1 C=RED V=NONE I=JOIN LINE=1;
  SYMBOL2 C=GREEN V=NONE I=JOIN LINE=1;
  TITLE1 F=SWISS C=BLUE
  'Comparison of Average High Monthly Temperatures';
RUN;
```

Comparison of Average High Monthly Temperatures

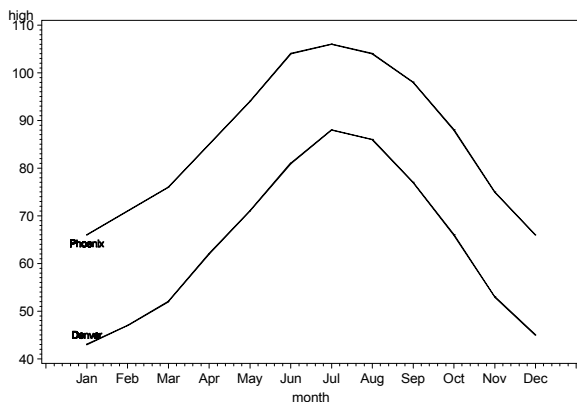


Figure 2

Labeling Important Data Points

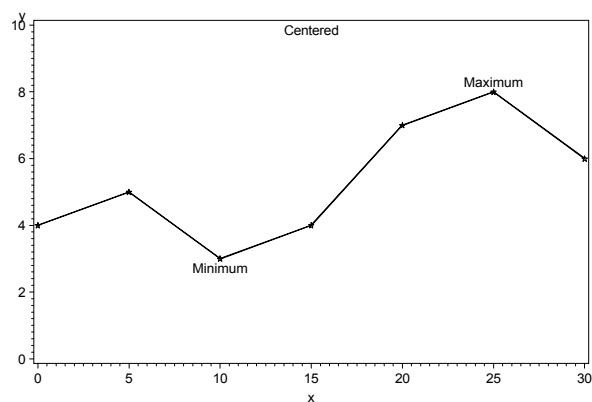


Figure 3

In the next example we want to automatically highlight the maximum and minimum points on a plot. SAS will determine which are the extreme points no matter how often the data changes. The output is reproduced as Figure 3.

```
DATA POINTS;
* Read data values;
  INPUT X Y @@;
  DATALINES;
0 4 5 5 10 3 15 4 20 7 25 8 30 6
RUN;

PROC MEANS NOPRINT;
* Calculate minimum and maximum;
  VAR Y;
  OUTPUT OUT=STAT N=NOBS
  MIN=YMIN MAX=YMAX;
RUN;

DATA DESCRIPT;
* Create Annotate data set;
  LENGTH TEXT $ 8;
  RETAIN NOBS YMIN YMAX;
  IF _N_ =1 THEN SET STAT;
  SET POINTS;
  IF _N_ < NOBS THEN DO;
    XSYS='2'; YSYS='2';
    SELECT( Y );
    WHEN (YMIN) DO;
      POSITION='8'; TEXT='Minimum';
    OUTPUT; END;
    WHEN (YMAX) DO;
      POSITION='2'; TEXT='Maximum';
    OUTPUT; END;
    OTHERWISE;
  END;
  ELSE DO;
    XSYS='1'; YSYS='1'; X=50;
    Y=100; TEXT='Centered';
  OUTPUT;
  END;
RUN;

PROC GPLOT DATA=POINTS;
* Plot data points and highlight special points;
  PLOT Y * X / ANNOTATE=DESCRIPT
  HAXIS=0 TO 30 BY 5
  VAXIS=0 TO 10 BY 2;
  SYMBOL I=JOIN V='+';
  TITLE "Labeling Important Data Points";
RUN;
```

In the next example a bar chart and line plot will appear on the same axis with the specific type, bar or line, depending on the date. The output is reproduced as Figure 4.

```
DATA SALES;
* Read sales data points;
  INPUT MONTH DATE7. SALES;
  DATALINES;
*** data lines go here ***
RUN;

DATA BARS LINES; SET SALES;
* Determine type of data based on date;
  * IF MONTH <= TODAY()
  THEN OUTPUT BARS;
  IF MONTH <= '01AUG00'D
  THEN OUTPUT BARS;
  ELSE OUTPUT LINES;

DATA ANNOBARS; SET BARS;
* Create data label for bars;
  LENGTH COLOR FUNTION STYLE $ 8
  TEXT $ 3;
  WHEN='B'; XSYS='2'; YSYS='2';
  FUNCTION='MOVE'; X=MONTH-12;
  Y=0; OUTPUT;
  FUNCTION='BAR'; X=MONTH+12;
  Y=SALES; COLOR='RED';
  STYLE='SIMPLEX'; OUTPUT;
  FUNCTION='LABEL'; X=MONTH;
  TEXT=SALES; POSITION='2';
  OUTPUT;

PROC GPLOT DATA=LINES
  ANNOTATE=ANNOBARS;
* Plot past data in bars and forecasts with lines;
  AXIS1 LABEL=(F=SWISS H=1 'Sales $000')
  VALUE=(F=SWISS H=1 C=GREEN)
  ORDER=0 TO 1000 BY 250;
  AXIS2 LABEL=(F=SWISS H=1 'Month')
  VALUE=(F=SIMPLEX H=1 C=RED)
  ORDER='01DEC99'D '01JAN00'D
  '01FEB00'D '01MAR00'D '01APR00'D
  '01MAY00'D '01JUN00'D '01JUL00'D
  '01AUG00'D '01SEP00'D '01OCT00'D
  '01NOV00'D '01DEC00'D;
  SYMBOL I=JOIN V='+' C=WHITE;
  FORMAT MONTH WORDDATE3.;
```

```

PLOT SALES * MONTH / FRAME
  VAXIS=AXIS1 HAXIS=AXIS2;
TITLE1 F=SWISS C=BLUE 'Acme Tissue Company';
TITLE2 F=SWISS H=1.5 C=BLUE
  '2001 Sales Figures in $000';
RUN;

```

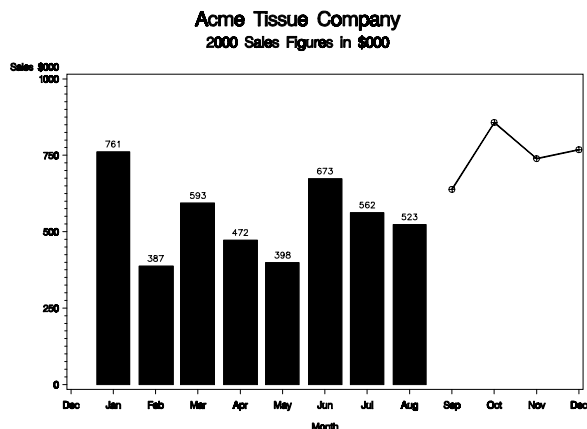


Figure 4

ANNOTATE MACROS

The Annotate facility can be used very effectively with the SAS macro language. The macros can be called with appropriate parameters, which lets the user avoid many individual assignment statements. The following is a list of Annotate macros and their parameter lists. The source for these macros is included in the SAS sample library that was distributed with the software.

The ANNOMAC macro compiles the standard Annotate macros and makes them available for use. You must submit the ANNOMAC macro (syntax is %ANNOMAC;) before you can use the Annotate macros.

```

%BAR( X1, Y1, X2, Y2, COLOR, LINE, PATTERN );
%CIRCLE( X, Y, RADIUS, COLOR );
%CNLT2TXT;
%COMMENT( TXT );
%DCLANNO;
%DRAW( X, Y, COLOR, LINE, WIDTH );
%DRAW2TXT( COLOR, LINE, WIDTH );
%FRAME( COLOR, LINE, WIDTH, STYLE );
%LABEL( X, Y, TXT, COLOR, ANGLE,
  ROTATE, HEIGHT, FONT, POS );
%LINE( X1, Y1, X2, Y2, COLOR, LINE, WIDTH );
%MOVE( X, Y );
%PIEXY( ANGLE, MULTIPLIER );
%POLY( X, Y, COLOR, PATTERN, LINE );
%POLYCONT( X, Y, COLOR );
%POP;
%PUSH;
%RECT( X1, Y1, X2, Y2, COLOR, LINE, WIDTH );
%SCALE( PTX, PTY, XMIN, YMIN, XMAX,
  YMAX, VXMIN, VYMIN, VXMAX, VYMAX );
%SCALET( PTX, PTY, XMIN, YMIN, XMAX,
  YMAX, VXMIN, VYMIN, VXMAX, VYMAX );
%SEQUENCE( WHEN );
%SLICE( X, Y, ANGLE, ROTATION,
  RADIUS, COLOR, PATTERN, LINE );
%SWAP;
%SYSTEM( XSYS, YSYS, HSYS );
%TXT2CNLT;

```

EXAMPLE USING ANNOTATE MACROS

This next example uses the Annotate macros that come with the SAS/GRAPH software. When looking at the following SAS program statements, try to imagine how many DATA step statements it would take to replace the macro statements.

This example assumes the %ANNOMAC statement is used to make the Annotate macros available. The output is reproduced as Figure 5.

```

* Specify the last month of data to include;
%LET ENDDATE=01NOV00;

```

```

DATA RATE;
*** DATA step to create sales data ***
*** Variables include region, datein, grp, rate ***
RUN;

```

```

PROC SORT DATA=RATE; BY REGION DATEIN;
RUN;

```

```

DATA ANNO; SET RATE;
  BY REGION;
* Set common Annotate variables;
%DCLANNO;
LENGTH TEXT $ 50;
%SYSTEM(3,3,4);

```

```

IF FIRST.REGION THEN DO;
* Write a title and footnote;
%LABEL(50,97,'Monthly Sales Comparison by
  Region',BLACK,0,0,2,SWISSB,5);

```

```

* Draw vertical and horizontal axes;
%MOVE(10,90);
%DRAW(10,10,BLACK,1,1);
%DRAW(85,10,BLACK,1,1);

* Draw solid and dotted reference lines;
%MOVE(22.5,90);
%DRAW(22.5,10,BLACK,2,1);
%MOVE(35,90);
%DRAW(35,10,BLACK,1,1);
%MOVE(47.5,90);
%DRAW(47.5,10,BLACK,2,1);
%MOVE(60,90);
%DRAW(60,10,BLACK,1,1);
%MOVE(72.5,90);
%DRAW(72.5,10,BLACK,2,1);
%MOVE(85,90);
%DRAW(85,10,BLACK,1,1);

```

```

* Write region name;
%LABEL(50,93,REGION,BLUE,0,0,1.75,SWISSB,5);

```

```

* Write horizontal axis label;
%LABEL(47.5,4,'Percent of Sales',
  BLACK,0,0,1.5,SWISSB,5);

```

```

* Create a legend;
%LABEL(88,50,'Gadgets',BLACK,0,0,1.2,SWISS,6);
%BAR(85,48,87,51,H0AAAAFF,0,SOLID);
%LABEL(88,55,'Widgets',BLACK,0,0,1.2,SWISS,6);
%BAR(85,56,87,53,H096A0FF,0,SOLID);

```

```

* Write tick marks and labels for horizontal axis;
%LABEL(10,10,'0%',BLACK,0,0,1.2,SWISS,8);
%LABEL(22.5,10,'2.5%',BLACK,0,0,1.2,SWISS,8);
%LABEL(35,10,'5%',BLACK,0,0,1.2,SWISS,8);
%LABEL(47.5,10,'7.5%',BLACK,0,0,1.2,SWISS,8);

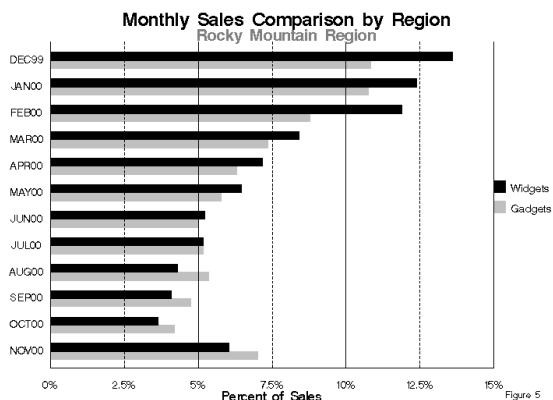
```

```
%LABEL(60,10,'10%',BLACK,0,0,1.2,SWISS,8);
%LABEL(72.5,10,'12.5%',BLACK,0,0,1.2,SWISS,8);
%LABEL(85,10,'15%',BLACK,0,0,1.2,SWISS,8);
```

```
* Write monthly labels on vertical axis;
DO I=1 TO 12;
  DATE=INTNX('MONTH',"&ENDDATE"D,1-1);
  MONTH=PUT(DATE,MONYY5.);
  YPOS=15-((1-I)*6.5);
  %LABEL(6,YPOS,MONTH,BLACK,0,0,1.2,SWISS,5);
END;
END;

* Calculate location of month on vertical axis and
create bars for the two products;
INCR=INTCK('MONTH',"&ENDDATE"D,DATEIN);
YPOS=14-(INCR*6.5);
XPOS=((RATE/.15)*75)+10;
IF GRP="Widgets"
  THEN DO;
  %BAR(10,YPOS,XPOS,YPOS+2,H096A0FF,0,SOLID);
END;
ELSE DO;
  %BAR(10,YPOS,XPOS,YPOS-2,H0AAAAFF,0,SOLID);
END;
END;
RUN;
```

```
* Create graph with PROC GANNO;
PROC GANNO ANNOTATE=ANNO NAME=REGION;
RUN;
QUIT;
```



SUMMARY

The Annotate facility is a flexible means for adding descriptive annotation to SAS/GRAPH procedure output. Using the power of the DATA step and the SAS macro language, it can be extended to perform a wide range of graphics customization. For further information on the Annotate facility and complete documentation, please refer to Volume 1 of the SAS/GRAPH User's Guide. Also check the SAS/GRAPH procedures GANNO and GSLIDE in Volume 2 of the SAS/GRAPH User's Guide.

CONTACT INFORMATION

Joe McCormick
Trilogy Consulting
Denver, CO
mccormick452@qwest.net

Joe is a long-time SAS user. He has used every major version of SAS, including SAS72. Joe has worked for companies using SAS in medical and pharmaceutical research, database marketing and reporting. Joe also provided technical and marketing support for several years working for SAS Institute in Sydney, Australia. He is now employed as a Senior Technical Consultant with Trilogy Consulting in Denver, Colorado.

SAS and SAS/GRAPH are registered trademarks of SAS Institute Inc. in the USA and other countries.