Enhancing SAS/GRAPH® Output with the Annotate Facility

Steven First
Systems Seminar Consultants
Madison, WI 53716 (608) 222-7081

Abstract

The SAS/GRAPH Annotate facility allows the user to enhance graphics created with SAS/GRAPH software. This paper presents the basic concepts of the Annotate facility and introduces the audience to text placement, symbol placement, line drawing, polygon construction, map labeling, custom legends, and line types. Also covered are SAS Institute provided annotation macros and advanced functions to allow complex labeling and freehand graphics.

Introduction

When SAS/GRAPH Version 5 software was released, the new Annotate facility provided a comprehensive solution to some long requested missing capabilities with the product. Annotation is any text, symbol, line, or polygon that is overlaid on an existing graphic image. There are at least three or four different levels of annotation required.

First a simple method was needed to place annotation anywhere on a graphic image. An example of this type of annotation might be to place a company logo in the same place on all graphs regardless of the titles and footnotes a user chooses.

A second need was to place annotation in the plot "window". This would allow the user to code as many titles and footnotes as needed again, but have the annotation appear below the titles and above the footnotes.

A third need was to somehow tie the annotation to the data axes. An example of this is to place text that "floats" above the bars of a chart. This is the most complex type of annotation because the data values might change from run to run and the annotation must move with the graph.

Finally it would be useful to be able to generate annotation from another source. An example of this might be to read a text file, and generate text charts using SAS/GRAPH fonts and colors.

The Annotate facility does all of the above and much more. It uses a rather batch oriented approach and does not offer features like interactive cursor draw at this time.

Ways to modify SAS/GRAPH output

Without annotation we can enhance graphic output by:

- Changing PROC options or additional statements
- Changing the data
- Using titles, footnotes, notes
- PROC GREPLAY can place multiple pictures on a page

The Annotate facility allows the user to:
- Place text anywhere on the screen
- Draw lines or other figures
- Define polygons to fill and color
- Create custom "free hand" graphics
- Place text or images on maps

The annotate commands are stored in a SAS dataset using special variables. This dataset can be specified on the ANNOTATE= parameter for PROCs GANNO, GCHART, GCONTOUR, G3D, GMAP, GPLOT, and GSLIDE. When an annotate dataset is specified the PROC will draw the graphic as usual, and then overlay the annotation on the graph. PROC GANNO produces no graphics by itself and is intended to produce only annotation.

The following program will create a simple annotate dataset and then draw the image shown in figure 1. Note that the variables must be named exactly as shown. Many other variables can be added, but if not present a default value will be used.

```
DATA ANNOOS1;
INPUT FUNCTION $ X Y SIZE TEXT $20. ;
DATALINES;
MOVE 20 10 ;
DRAW 20 30 3 ;
DRAW 80 30 3 ;
DRAW 80 10 3 ;
LABEL 50 25 3 A simple box
;
PROC GANNO ANNOTATE=ANNOOS1;RUN;
```

Figure 1 A simple box
Figures 2 through 6 show various capabilities of the annotate facility.

### Figure 2
Various polygons

![Various polygons](image1)

### Figure 3
Map annotation

![Map annotation](image2)

### Figure 4
Map annotation

![Map annotation](image3)

### Figure 5
Regional Sales

<table>
<thead>
<tr>
<th>Region</th>
<th>Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>North</td>
<td>$244,678</td>
</tr>
<tr>
<td>South</td>
<td>$532,123</td>
</tr>
<tr>
<td>East</td>
<td>$564,343</td>
</tr>
<tr>
<td>West</td>
<td>$167,343</td>
</tr>
<tr>
<td>Alaska</td>
<td>$233,434</td>
</tr>
<tr>
<td>Hawaii</td>
<td>$171,232</td>
</tr>
</tbody>
</table>

### Figure 6
Freehand graphics

![Freehand graphics](image4)

**Annotation sequence**

The following steps are required to annotate a graph:

1. Identify the components of the image.
2. Establish a frame or reference.
3. Digitize components, functions, attributes.
4. Store the above as variables in a SAS datasets.
5. Run a SAS/GRAPH PROC with ANNOTATE option.

All components images must be defined on a cartesian coordinate system as shown in figure 7.
Annotation variables

Special Annotate variables can be grouped into five categories:

1. To establish a frame of reference.
2. To set a function or identify a component.
3. To position an image.
4. Enhance the image display.
5. Position text.

Each of the above categories will control a different part of annotation. The dataset can contain more than the special variables in which case they will be ignored. Some examples and values of some of the variables are:

<table>
<thead>
<tr>
<th>Category</th>
<th>Variable</th>
<th>Type</th>
<th>Length</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>XSYS</td>
<td>char</td>
<td>1</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>2</td>
<td>YSYS</td>
<td>char</td>
<td>1</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>3</td>
<td>FUNCTION</td>
<td>char</td>
<td>1</td>
<td>&quot;LABEL&quot;</td>
</tr>
<tr>
<td>4</td>
<td>X</td>
<td>num</td>
<td>8</td>
<td>XLAST</td>
</tr>
<tr>
<td>5</td>
<td>Y</td>
<td>num</td>
<td>8</td>
<td>YLAST</td>
</tr>
<tr>
<td>6</td>
<td>SIZE</td>
<td>num</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>STYLE</td>
<td>char</td>
<td>8</td>
<td>&quot;NONE&quot;</td>
</tr>
<tr>
<td>8</td>
<td>COLOR</td>
<td>char</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>LINE</td>
<td>num</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>POSITION</td>
<td>char</td>
<td>1</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>11</td>
<td>TEXT</td>
<td>char</td>
<td>200</td>
<td>&quot;&quot;</td>
</tr>
</tbody>
</table>

It is critical that the variables are created with the correct attributes. It is a very good idea to use PROC CONTENTS and PROC PRINT to display the annotate dataset at least while debugging because the annotate facility is very fussy about variable attributes and there is no point trying to annotate graphics if the underlying dataset is incorrect.

The program below uses default annotation to display the text shown in figure 8.

```
DATA DEFAULTA;
  INPUT X Y TEXT $ 50.;
DATALINES;
50 30 Label is the default function
50 20 with text centered about X and Y
```

Frames of reference

The Annotate facility can address three areas of the screen; the entire screen, the window, or the same data coordinate system that the PROC uses. All of the frames can be referenced by percentage or by screen cells, and can be referred to in a relative or absolute manner. Figure 9 shows the three frames of reference.

```
DATA DEFAULTA;
  INPUT X Y TEXT $ 50.;
DATALINES;
50 30 Label is the default function
50 20 with text centered about X and Y
```
Using '5' as a value in XSYS and YSYS tells the annotate facility to use a percentage of the window area and allows annotation between titles and footnotes. The following program created figure 10.

```
DATA WINDOW;
   INPUT XSYS $ 2 YSYS $ 4 X Y TEXT $ 50.;
DATALINES;
   5 5 50 99 This line is directly under titles
   5 5 50 50 and this is the center of the window
   5 5 50 02 How about just above footnotes?
;
PROC GCHART ANNOTATE=WINDOW;
   TITLE C=RED F=XSIS 'Annotation in ';
   FOOTNOTE C=RED F=XSIS 'the window';RUN;
```

Figure 10  Percentage of windows

A value of '3' in XSYS and YSYS tells the Annotate facility to use a percentage of the screen frame of reference. This allows the user to use annotation instead of using footnotes and titles. The user can easily locate any point on the screen by specifying an X and Y value somewhere between 1 and 100%. The following program produced the graph in figure 11.

```
DATA MORTGAGE;
   INPUT RATE YEAR;
DATALINES;
   12.0 1978
   14.1 1980
   15.0 1982
   13.8 1984
   9.7 1986
;
PROC GCHART DATA=MORTGAGE;
   PLOT RATE * YEAR/ VAXIS=0 TO 20 BY 2
   ANNOTATE=MORTG;
   SYMBOL 1=JOIN V=NONE;
   TITLE C=RED F=XSIS '%' ;
   FOOTNOTE C=RED F=XSIS 'the window';RUN;
```

Figure 11  Percentage of screen

A value of '2' in XSYS and YSYS selects the data frame of reference. This allows the user to refer to the same data area that the procedure will be using. The example shown in figure 12 is a PROC GCHART which has Y axis values (RATE) between 0 and 20, and X axis values (YEAR). The program below moves those same values to X and Y in the annotate dataset which allows the user to position text directly over the point on the plot. This particular example uses one dataset for the plotting and another for the annotation, but it could be easily modified to generate the annotation from the plot data using SET instead. Note also that PROC GCHART and PROC GCHART can also plot character variables. If this were the case, the Annotate facility requires that the character variables X and Y be used instead of X and Y.

```
DATA MORTGAGE;
   INPUT XSYS $ 2 YSYS $ 4 X Y TEXT $ 50.;
DATALINES;
   3 3 50 96 Annotated mortgage plot
   3 3 25 05 This example uses two frames of reference
   2 2 1978 12 (1978,12)
   2 2 1982 15 (1982,15)
   2 2 1984 13.8 (1984,13.8)
   2 2 1986 9.7 (1986,9.7)
;
PROC GCHART DATA=MORTGAGE;
   PLOT RATE * YEAR/ VAXIS=0 TO 20 BY 2
   ANNOTATE=MORTG;
   SYMBOL 1=JOIN V=NONE;
   TITLE C=RED F=XSIS 'Annotation can be used for titles'
   FOOTNOTE C=RED F=XSIS 'the window';RUN;
```
Additional annotation variables

The plot above centers the text directly over the plot points and since the plot line and axis also are drawn at those points, the annotation is cluttered. We need a method to move the annotation slightly away from the lines and the axis. In addition we may need to place other images such as lines, bars etc on the graph. The Annotate facility allows several other values of the FUNCTION variable to do the placing as shown in the table below. There are more advanced functions as well that are not listed here.

Values for FUNCTION:

- **LABEL** Position text at X,Y coordinates
- **MOVE** Move with the pen up
- **DRAW** Move with the pen down (draws a line)
- **BAR** Construct and fill a rectangle
- **FRAME** Frame the frame of reference
- **PIC** Draw and fill pie slices
- **POINT** Place a single point at X,Y
- **POLY** The beginning of a polygon to fill
- **POLYCONT** Successive points in a polygon
- **SYMBOL** Place a symbol on graph

The LABEL FUNCTION depends on several other variables to help place the text and also specify other attributes such a font, color, angle, etc. Below is listed some of the associated variables and their usage.

- **POSITION** Values '0' to 'f' (default = '5')
- **SIZE** Any positive number (default = 1)
- **ANGLE** -90 to 90 (default = 0)
- **STYLE** Any valid font (default = 'NONE')
- **COLOR** Any valid color (default = first in color list)
- **TEXT** Any text string (default = ' ') or PL

The POSITION variable is especially important to align a block of text many different ways around a point. The following list shows the different options for text placement and should help us solve the placement problem in the last graph.

- **'1'** One cell above, right aligned
- **'2'** One cell above, centered
- **'3'** One cell above, left aligned
- **'4'** Centered, right aligned
- **'5'** Centered, centered

By using the POSITION, SIZE, and STYLE variables we can enhance the GPLOT job from earlier. The label marked (1978,12) uses a POSITION value of 5 which aligns the left edge of the text one cell below the X and Y value. Now the label is positioned in available space instead of overlaid on the axis as before. Using a $ character from the GREEK font also plots the $ character over the points on the line. The program below produces figure 13.

**Figure 13** Using POSITION, SIZE, STYLE

**Line drawing**

MOVE and DRAW as FUNCTION values along with the COLOR, SIZE, and LINE variables can be used to draw lines and in effect create custom interpolation types. In the below program part of the lines are drawn in a dashed pattern and the other parts solidly producing figure 14.
DATA MORTGAGE;
INPUT XSYS YSYS $ 4 FUNCTION $ X Y;
POSITION $ 21 STYLE $ SIZE LINE TEXT $40.;
DATALINES;
3 3 LABEL 50 96 5 WSX11S 2 Annotated mortgage plot
3 3 LABEL 50 5 WSX11S 1 Using the draw function
2 2 MOVE 1978 9.1...
2 2 DRAW 1980 13...
2 2 DRAW 1982 14...
2 2 DRAW 1984 13...
2 2 DRAW 1986 8.7...
2 PROC GPLOT DATA=MORTGAGE;
PLOT RATE * YEAR;
AXIS1=1978 TO 1996 BY 2
AXIS=0 TO 20 BY 2
ANNOTATE=MORTD;
SYMBOL I=JOIN V=NONE;
TITLE1 '';
FOOTNOTE '';
RUN;

Figure 14 Drawing lines

Plot labeling

Another variation is to let GPLOT do the line drawing, but to use annotation to place the text MORTGAGE and PRIME near the lines to act as a legend as in figure 15.

DATA MORTGAGE;
INPUT RATE YEAR $ 5 $;
DATALINES;
12.6 1978 MORTGAGE
14.1 1978 PRIME
13 1980 MORTGAGE
14 1980 PRIME
13.8 1982 MORTGAGE
13 1982 PRIME
9.7 1986 MORTGAGE
8.7 1986 PRIME

DATA LABELS;
XSYS='2'; YSYS='2'; POSITION='F';
STYLE='SIMPLEX'; Y='12'; X='17';
TEXT='MORTGAGE'; COLOR='BLUE';
OUTPUT;
Y='9.1'; TEXT='PRIME'; COLOR='RED';
OUTPUT;
PROC GPLOT DATA=MORTGAGE;
PLOT RATE * YEAR=TYPE/OVERLAY NOLEGEND
AXIS=1978 TO 1996 BY 2
AXIS=0 TO 20 BY 2
ANNOTATE=LABELS;
SYMBOL1 I=JOIN V=NONE C=RED;
SYMBOL2 I=JOIN V=NONE C=BLUE;
PATTERN1 V=SOLID C=RED;
PATTERN2 V=SOLID C=BLUE;
TITLE1 F=XSUIS C=BLUE 'Mortgage'
C=BLACK and 'PRIME Rate'; RUN;

Figure 15 Plot labeling

Custom legends

Another use of the Annotate facility is to build custom legends in available space. The BAR or PIE value for FUNCTION can draw rectangles or circles, and the LABEL FUNCTION can add the needed text as shown in figure 16.

DATA LEGENDS;
LENGTH TEXT $16 FUNCTION STYLE $ 8;
WHEN;'A';
INPUT XSYS YSYS $ 4 FUNCTION $ X Y;
POSITION $ 35; SIZE LINE TEXT $40.;
DATALINES;
2 2 LABEL 1979 19 GREEN SIMPLEX 2 LEGEND
2 2 LABEL 1979 18 BLACK SIMPLEX 1 MORTGAGE RATE
2 2 LABEL 1979 17 BLACK SIMPLEX 1 PRIME RATE
2 2 MOVE 1979 16...
A A BAR -1 BLUE SOLID 0...
A A BAR -1 RED SOLID 0...
PROC GPLOT DATA=LENGEND;
PLOT RATE * YEAR=TYPE/OVERLAY
AXIS=1978 TO 1996 BY 2
AXIS=0 TO 20 BY 2
ANNOTATE=LEGEND;
SYMBOL I=JOIN V=NONE C=RED;
SYMBOL2 I=JOIN V=NONE C=BLUE;
PATTERN1 V=SOLID C=RED;
PATTERN2 V=SOLID C=BLUE;
TITLE1 F=XSUIS C=BLUE 'Mortgage'
C=BLACK and 'PRIME Rate'; RUN;

Figure 16 A custom legend
Bar chart annotation

Another common application is to place annotation at the end of a bar chart. This is easy to do by again using the data frame of reference. The position variable can place the text above the bar as shown, or if desired position could be used to place text within the bar. Again the data frame of reference allows the annotation to "float" on the end of the bars, and the program could be easily written such that the annotation will automatically move if different data is input. The Annotate facility doesn't support any "clipping" so unless the text has a contrasting color, it may not show very well and depending on the device it may smear if drawn before the bar is filled. The WHEN variable can be set to 'A' if the annotation should take place after the chart is draw, and a value of 'B' will draw annotation before the chart drawing. The following program produces the output shown in figure 17.

```
DATA MORTBAR;
  INPUT XSYS $ 2. YSYS $ 4. X  Y  POSITJON $ 16  STYLE $  SIZE  TEXT $ 50. ;
  DATALINES;
  2. 2. 1978 12.0 2. SIMPLEX 1 12.0
  2. 2. 1980 14.1 2. SIMPLEX 1 14.1
  2. 2. 1982 15.0 2. SIMPLEX 1 15.0
  2. 2. 1984 13.8 2. SIMPLEX 1 13.8
  2. 2. 1986 9.7 2. SIMPLEX 1 9.7;
  PROC GCHART DATA=MORTGAGE;
  SUMVAR:RATE
  SPACE=4 WIDTH=5
  AXIS=D TO 100 BY 2
  ANNOTATE:MORTBAR
  DES='MORTGAGE VBAR ';
  PATTERN1 C=RED V=SOLID;
  TITLE1 C=RED F=XSWISS 'Home mortgage rates';
  TITLE2 C=BLACK F=SIMPLEX '(Annual percentage rates)';
  RUN;
```

```
<table>
<thead>
<tr>
<th>YEAR</th>
<th>RATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1978</td>
<td>12.0</td>
</tr>
<tr>
<td>1980</td>
<td>14.1</td>
</tr>
<tr>
<td>1982</td>
<td>15.0</td>
</tr>
<tr>
<td>1984</td>
<td>13.8</td>
</tr>
<tr>
<td>1986</td>
<td>9.7</td>
</tr>
</tbody>
</table>
```

Figure 17  Annotated bar chart

Text applications

A good application of the Annotate facility is to build text charts. The user has all the text features available and is able to build attractive word charts easily. This program uses a frame of reference that specifies '3' or percentage of screen. Note that the program also has a variable called HSYS which is also set to '3'. This variable is used to set the size of the text string as a percentage of screen. The program uses a TITLE statement, but then assumes 70% of the screen is left for the remaining text lines. The Y variable starts at 90 and X starts at 15. After a text line is output, a value called YDECOR is used to move down the graph. YDECOR is defined as 70 percent of the screen divided by the number of lines to display (6). Each line has a size of half the YDECOR value and finally the dollar value is placed at horizontal position 85. Even though this application uses hard coded positions, systems can be easily created to generate word charts from input text files and use all the SAS/GRAPH text features. The program below generated figure 18.

```
DATA SALES;
  INPUT REGION $ 1-6 SALES;
  LENGTH TEXT $ 30;
  RETAIN XSYS YSYS HSYS '3' Y 90;
  YDECOR=70/6;
  SIZE=YDECOR*.5;
  STYLE='XSWISSU';
  COLOR=' BLACK ';
  POSITION='6';
  X=15; Y=Y-YDECOR;
  ANNOTATE=SALES;
  TITLE H=5 F=XS\WISS C=RED 'Regional Sales';
  RUN;
```

```
<table>
<thead>
<tr>
<th>REGION</th>
<th>SALES</th>
</tr>
</thead>
<tbody>
<tr>
<td>North</td>
<td>$244,678</td>
</tr>
<tr>
<td>South</td>
<td>$532,123</td>
</tr>
<tr>
<td>East</td>
<td>$564,343</td>
</tr>
<tr>
<td>West</td>
<td>$167,343</td>
</tr>
<tr>
<td>Alaska</td>
<td>$233,434</td>
</tr>
<tr>
<td>Hawaii</td>
<td>$171,232</td>
</tr>
</tbody>
</table>
```

Figure 18  Text applications

Map annotation

Another good annotation application is to place text or other annotation on maps. SAS/GRAPH provides several maps, and also a dataset called USCITY containing the names of major cities in the US. This dataset is already projected into X and Y coordinates.
and can be easily used to annotate any US map if it also is projected. The following example creates an annotate dataset containing the major cities in Wisconsin. It then creates a small map of Wisconsin from the SASMAPS.US dataset which includes the entire country. A simple PROC GMAP with the ANNOTATE option is all that is needed to draw the map shown in figure 19.

```sas
DATA WICTITIES;
LENGTH TEXT $ 40;
LENGTH COLOR FUNCTION STYLE $ 8;
SET SASHAPS.USCITY;
IF STATE=STFKPS('WI') ;
XSYS='2' ; YSYS='2';
TEXT=CITY:
FUNCTION='LABEL' ;
POSITION='5';
WHEN='A' ;
COLOR='RED' ;
STYLE='SIMPLEX' ;
OUTPUT;
RUN;
DATA WIMAP;
SET SASHAPS.US;
IF STATE = SHFKPS('WI');
RUN;
PROC GMAP DATA=WIMAP MAP;
ID STATE;
CHORD STATE/NOLEGEND ANNOTATE=WICTITIES;
TITLE C='COLOR' F=XSWISS 'Wisconsin Cities';
FOOTNOTE 'Source: Projected States dataset';
PATTERN1 COLOR=BLACK V=EMPTY;
RUN;
```

Figure 19 Projected annotation

Annotating unprojected maps is a much more difficult task. The SAS/GRAPH unprojected maps provide much more detail and generally look better. The key to the following program running successfully is that the annotation and the map must be projected at the same time. This means that the two datasets must be combined, projected, split apart, and then PROC GMAP can draw the final map. The coordinates themselves need to be transformed before projection using a rather obscure formula dug out of the SAS/GRAPH User's Guide. The following program produced figure 20.

```sas
DATA WIUNPROJ;
SET SASHAPS.STATES;
IF STATE=55;
RUN;
DATA ALL;
SET WIUNPROJ IN=QCITY);
IF ONEXIT THEN
00;
D2K= ATAN(1)
145;
X=LONG * D2R;
Y=LAT * D2R;
END;
RUN;
PROC GPROJECT OAT=A ALL OUT=ALLPROJ;
ID STATE;
RUN;
DATA WIPROJM WIUNPROJ;
SET ALLPROJ;
IF FROMMAP='YES' THEN OUTPUT WIUNPROJ;
ELSE OUTPUT WIUNPROJ;
RUN;
PROC GMAP DATA=WIPROJM MAP=WIPROJM;
ID STATE;
CHORD STATE/NOLEGEND ANNOTATE=WITITIES2;
TITLE C='COLOR' F=XSWISS 'Wisconsin Cities';
FOOTNOTE 'Source: Projected States dataset';
PATTERN1 COLOR=BLACK V=EMPTY;
RUN;
```

Figure 20 Unprojected annotation

SAS/GRAPH comes with an extensive library of annotate macros that can be used to simplify setting up annotation datasets. They are normally stored in the SAS sample source library in a member called ANNOMAC. If they are not in a library that is included in the SAS autocall library list, the ANNOMAC member must be %INCLUDEd before they will be available for use. They are rather lengthy so you may want to turn off SOURCE2 before including them. It should be noted that these macros should be used only in a OAT A step. A list of the macros and their usage is described below.

- %CLAMNO Set length and data type of variables
- %COMMENT Insert a comment observation
- %DIAGSEQ Draw before or after other graphics proc
- %SYSTEM Define frame of reference
- %DRAW Move with pen up
- %DRAW Move with pen down
- %LINE Draw a line between two points
- %LABEL Add text
- %RECT Draw an empty rectangle
- %RECT Draw a filled rectangle
- %DRAW Draw a circle
- %ELICE Draw a line slice
- %EPXY Draw an empty coordinate
- %EPXY Draw a filled coordinate
- %LABEL Draw a label on previous slice
- %POLY Draw a polygon
- %POLYCONT Continue a polygon
- %SCALE Scale coordinates
- %PUSH Push text and control onto LIFO stack
- %POP Pop text and control onto LIFO stack
- %TEXT Exchange text and control coordinates
- %TEXT2CNTL Assign text coordinate to control coordinate
Miscellaneous applications

An application of the annotate macros might be to generate a grid of the cells for a graphics device. The program below created figure 21 for a device that has 40 rows (VPOS) and 100 columns (HPOS). It could be easily modified to any HPOS or VPOS, or it might use another frame of reference such as percentage of screen and draw lines every 10% or so. If the grid is drawn on transparency material it can be overlaid over any graph and can be used to help place annotation.

```
DATA GRID;
  %DCLANNO;
  DO y=0 TO 40;
    XLINE(0, y, 100, y, BLACK, 1, 1);
  END;
  DO x=0 TO 100;
    XLINE(x, 0, 100, x, BLACK, 1, 1);
  END;
RUN;
PROC GCHART DATA=GRID;
  TITLE;
RUN;
```

Figure 21  A generated grid

Another option may be to use a similar grid and have the annotate facility overlay it on a chart. By drawing a grid with lines every 10% and overlaying it on the chart, the user can note the locations of needed annotation and enter it in an annotate dataset which can be used to produce figure 22.

```
DATA ANOTEX5;
  %DCLANNO;
  XSYS='3'; YSYS='3'; HSYS='4';
  IF _N_ = 1 THEN
    DO Y=0 TO 100 BY 10;
      XLINE(0, y, 100, y, BLACK, 1, 1);
    END;
    DO X=0 TO 100 BY 10;
      XLINE(x, 0, 100, x, BLACK, 1, 1);
    END;
  END;
RUN;
PROC GCHART DATA=ANOTEX5;
  TITLE 'STANDARD GRID DATASET';
RUN;
```

After looking at the previous graph an annotate dataset can be built and the final graph drawn as in figure 23. This technique is quick if only one graph needs to be annotated, but if the data vary from run to run, it may be better to use the data frame of reference as described earlier.

```
DATA ANOTEX6;
  XSYS='3'; YSYS='3'; HSYS='4';
  INPUT FUNCTION $ X Y TEXT $ 14-16 STYLE $ COLOR $ SIZE POSITION $ ;
  DATALINES;
  LABEL 43 65 622 SIHPLEX BLACK;
  LABEL 51 72 708 SIHPLEX BLACK;
  LABEL 59 71 699 SIHPLEX BLACK;
  LABEL 67 80 804 SIHPLEX BLACK;
RUN;
PROC GCHART DATA=ANOTEX6;
  TITLE 'ANNOTATE DATASET 6';
RUN;
```

Figure 22  Bar chart with grid
Freehand graphics

Freehand graphics can be drawn by following the following steps:

1. Include all points in all polygons.
2. Use MOVE, DRAW to draw lines.
3. Use LABEL for drawing text.

This can be a very time consuming task if done manually, but if the polygons and text are already in a machine readable form it can be automated.

The dataset below defines four different polygons and the text on the graph. It actually consists of several hundred points, but all were not shown because of space. The final graph is shown in figure 24.

DATA CYCLONE;
PROC GSLIDE ANNOTATE=CYCLONE;
TITLE F=SWISS C=RED 'TORNADO WATCH';FOOTNOTE;RUN;

SAS and SAS/GRAPH are registered trademarks of SAS Institute Inc.

212