ABSTRACT

There are a number of ways within the Annotate Facility to describe the location, or display coordinates, that will be used by a function. The Annotate Facility accomplishes the task through reference systems. These reference systems permit the specification of display coordinates in a number of different measurement systems. For example, the location may be expressed as percentages of the screen area, or as actual data values. We will be addressing the actual data value specifications and how they relate to particular SASiGRAPH® procedures.

INTRODUCTION

The data system is special in the Annotate framework. It permits us to describe locations in terms of the data displayed by the procedure. A properly constructed Annotate application which uses the data system is extremely flexible. Because the annotation reflects the current data, as our data changes, so may the annotation.

With the data systems, we are released from the requirements of having to know so much about the exact display locations of axes, footnotes and other graph elements. We can then concentrate on the application, rather than becoming overwhelmed by the display awareness required with the screen system.

Each of the SASiGRAPH procedures has a particular way of dealing with our input data. We will be looking at how the procedure processing affects the way we use the data systems in our annotation. And how we can tailor our application based on the procedure we choose.

GPLOT PROCEDURE

Description of system reference

For the GPLOT procedure, the Annotate data system is the graphics area defined along the axis lines from the first major tic mark to the last major tic mark inclusive. No data values exist outside this area.

Highlighting the data area

DATA xy ;
  input x y @@ ;
cards;
  1 2 1 3 3 4 2 5 5
RUN ;
DATA range ;
  RETAIN xsys ysys '2' style 'S' color 'YELLOW' ;
  x = . ; y = . ;
  function = 'FRAME' ; output ;
RUN ;
SYMBOL I=SPLINE C=WHITE V=STAR ;
PATTERN V=SOLID C=GREEN ;
TITLE 'PROG GPLOT DATA SYSTEM AREA' ;
PROC GPLOT DATA=xy ;
  PLOT y * x / ANNOTATE=range AREAS=1 ;
RUN ;

Annotate variables

There are some generalizations we can make about the Annotate variables and their relationship to the procedure variables. These rules apply to all Annotate / procedure variable relationships.

- The Annotate X variable is always a numeric type and always refers to horizontal, i.e. left-to-right, positions
- The Annotate XC variable is always a character type and always refers to horizontal positions
- The Annotate Y variable is always a numeric type and always refers to vertical, i.e. top-to-bottom, positions
- The Annotate YC variable is always a character type and always refers to vertical positions

GPLOT required variables

The GPLOT procedure is capable of plotting both numeric and character valued data. From the rules listed above, we can conclude the following about the required Annotate variables for PROC GPLOT:
If the horizontal plot variable is a character variable, we must have the XC variable in our ANNOTATE= data set. Otherwise, we must have the X variable in our ANNOTATE= data set.

If the vertical plot variable is a character variable, we must have the YC variable in our ANNOTATE= data set. Otherwise, we must have the Y variable in our ANNOTATE= data set.

Appropriate diagnostic messages will be issued by the Annotate Facility if these requirements are not satisfied in our ANNOTATE= data set. The actual procedure relationship rules are parallel to the variable requirements. We should be careful here to associate Annotate variables with the locations to which they refer, not particular plot variable names.

If the horizontal plot variable is a character variable, we must use the XC variable to place data system annotation. Otherwise, we must use the X variable.

If the vertical plot variable is a character variable, we must use the YC variable to place data system annotation. Otherwise, we must use the Y variable.

Appropriate diagnostic messages will be issued by the Annotate Facility if an attempt is made to annotate a character valued axis with a numeric data system reference and vice versa.

From Table 2.1, we can see that the same variables are used in the ANNOTATE= data set when annotating on the PLOT2 statement as on the PLOT statement. This is because separate ANNOTATE= data sets may be used on the PLOT and PLOT2 statements. All data system references for the PLOT2 statement are made relative to the right-hand vertical axis definition of PROC GPILOT.

**Simple applications**

The following examples illustrate the variable requirements when annotating with PROC GPILOT. All the preliminary examples will use the sample data set shown below. It contains a numeric student identification number, a character name field and a numeric test score. A letter grade is also assigned, based on a simple linear scale.

<table>
<thead>
<tr>
<th>student</th>
<th>name</th>
<th>score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allan</td>
<td>98</td>
<td></td>
</tr>
<tr>
<td>Johnny</td>
<td>95</td>
<td></td>
</tr>
<tr>
<td>Kathy</td>
<td>76</td>
<td></td>
</tr>
<tr>
<td>Lisette</td>
<td>80</td>
<td></td>
</tr>
<tr>
<td>Mary</td>
<td>92</td>
<td></td>
</tr>
<tr>
<td>Stephen</td>
<td>65</td>
<td></td>
</tr>
<tr>
<td>Teresa</td>
<td>81</td>
<td></td>
</tr>
<tr>
<td>William</td>
<td>70</td>
<td></td>
</tr>
</tbody>
</table>

The first example requires numeric placements for both the horizontal and vertical locations. We must use the X and Y locator variables for the data dependent placement. By specifying POSITION '2', the letter grade will be placed above the marker symbol for each student's performance.

```plaintext
/* NUMERIC data value annotation */
data grades;
    length function $ 8;
    retain xsys ysys position '2' function 'LABEL';
    keep x y xsys ysys position function text;
    set scores;
    y = score;
    x = stu_id;
    text = grade;
    run;
```

**BUBBLE and BUBBLE2 statements**

Annotating within the data value system with a BUBBLE or BUBBLE2 statement is not fully supported at this time. There is no data communication between the Annotate Facility and the GPILOT procedure in relation to the size of the bubble produced. You can use the data system to position yourself at the center of the appropriate bubble, but any additional Annotate placement calculations must be made within your application.
The next example switches from numeric student codes to character names on the horizontal axis. Numeric placements are still required for the vertical locations, but character placements are now needed for the horizontal locations. We must use the XC and Y locator variables for the data dependent placement.

```/* NUMERIC/CHARACTER data value annotation */ /* place letter grade above student score */ DATA grades ; LENGTH function $ 8 ; RETAIN xsys ysys position '2' function 'LABEL' ; KEEP xc yc xsys ysys position function text ; SET scores ; y = score ; xc = student ; text = grade ; RUN ;
```

SYMBOL I=NONE V=DOT C=WHITE ;

PROC G PLOT DATA=scores ;
AXIS1 ORDER=(40 to 100 by 20) OFFSET=(10,10) ;
AXIS2 OFFSET=(10,10) ;
PLOT score * student / ANNOTATE=grades
VAXIS=AXIS1 HAXIS=AXIS2 FRAME ;
RUN ;

This example displays numeric scores above the letter grade marker for each student. Character placements are now needed for both horizontal and vertical locations. We must use the XC and YC locator variables for the data dependent placement.

```/* CHARACTER/CHARACTER data value annotation */ /* place numeric score above student grade */ DATA grades ; LENGTH function $ 8 ; RETAIN xsys ysys position '2' function 'LABEL' ; KEEP xc yc xsys ysys position function text ; SET scores ; yc = grade ; xc = student ; text = put(score,f3.) ; RUN ;
```

SYMBOL I=NONE V=STAR C=WHITE ;

PROC G PLOT DATA=scores ;
AXIS1 ORDER=('F' 'D' 'C' 'B' 'A') OFFSET=(10,10) ;
AXIS2 OFFSET=(10,10) ;
PLOT grade * student / ANNOTATE=grades
VAXIS=AXIS1 HAXIS=AXIS2 FRAME ;
RUN ;

Advanced application

The final example might be a good 'first look' at test performance. We use PROC UNIVARIATE to calculate display percentiles based on raw test scores and build an ANNOTATE= data set to display the resulting appraisals. A multi-colored background highlights the percentiles and reference lines mark the mode and extreme scores. PROC GPLOT provides the scatter plot of the actual test scores.

```/* Calculate percentiles and statistics */ /* some appropriate raw data */ DATA scores ;
DO student = 1 TO 100 ;
   score = 72 + floor(11 * RANNOR(7)) ;
OUTPUT ;
END ;
RUN ;
```

```/* ANNOTATE the display stats */ DATA grades ;
LENGTH function style color $ 8 ;
RETAIN when 'B' style 'SOLID' ;
KEEP x y xsys ysys function ;
SET stats ;
/* below 25th */
COLOR='RED';
xsys='1'; ysys='1'; x=0; y=0;
function='move'; output;
xsys='1'; ysys='2'; x=100; y=q1;
function='bar'; output;
/* 25th - 50th */
COLOR='ORANGE';
xsys='1'; ysys='2'; x=0; y=q1;
function='move'; output;
xsys='1'; ysys='2'; x=100; y=median;
function='bar'; output;
/* 50th - 75th */
COLOR='YELLOW';
xsys='1'; ysys='2'; x=0; y=median;
function='move'; output;
xsys='1'; ysys='2'; x=100; y=q3;
function='bar'; output;
/* 75th - 95th */
COLOR='GREEN';
xsys='1'; ysys='2'; x=0; y=q3;
function='move'; output;
xsys='1'; ysys='2'; x=100; y=p5;
function='bar'; output;
/* 95th - 99th */
COLOR='BLUE';
xsys='1'; ysys='2'; x=0; y=p5;
function='move'; output;
xsys='1'; ysys='2'; x=100; y=p9;
function='bar'; output;
```

### 1334
/* 99th */
color='PURPLE';
xs ysys='1'; ysys='2'; x=0; y=99;
function='move'; output;
xs ysys='1'; ysys='1'; x=100; y=100;
function='bar'; output;

/* pseudo-reference lines */
color='WHITE'; line=32; when='A';
xs ysys='1'; ysys='2'; x=0; y=min;
function='move'; output;
xs ysys='1'; ysys='2'; x=100; y=min;
function='draw'; output;
xs ysys='1'; ysys='2'; x=0; y=max;
function='move'; output;
xs ysys='1'; ysys='2'; x=100; y=max;
function='draw'; output;
RUN;

SYM B O L I =N O N E V =D OT C =W H I T E ;

PROC GPLOT DATA=scores;
AXIS1 offset=(10,10);
AXIS2 offset=(10,10);
PLOT score * student / ANNOTATE=grades
VAXIS=AXIS1 HAXIS=AXIS2
FRAME;
RUN;

GMAP PROCEDURE

Description of system reference

For the GMAP procedure, the Annotate data system is the graphics area defined by the map data minimum and maximum values. This area is less distinct than that of the GPlot procedure because GMAP uses no axes. The Annotate FRAME function will clarify the range extents.

Highlighting the data area

DATA map;
SET MAPS.us;
IF ( state EQ STFIPS('FL') ) ;
RUN;

DATA range;
RETAIN xs ysys '2' style 'S' color 'YELLOW';
x = . ; y = . ;
function = 'FRAME'; output;
RUN;

TITLE 'PROC GMAP DATA SYSTEM AREA' ;
PATTERN V=SOLID C=GREEN ;
PROC GMAP DATA=map MAP=map ;

ID state ;
CHORO state / nolegend ANNOTATE=range ;
RUN ;

GMAP required variables

As with the GPlot procedure, there are some generalizations we can make about the Annotate variables and their relationship to the procedure variables. But, because GMAP deals only with numeric data, these rules can be greatly simplified.

The horizontal map variable is always a numeric variable, so, we must have the X variable in our ANNOTATE= data set.

The vertical map variable is also always a numeric variable, so, we must have the Y variable in our ANNOTATE= data set.

Appropriate diagnostic messages will be issued by the Annotate Facility if these requirements are not satisfied in our ANNOTATE= data set.

The actual procedure relationship rules are as straightforward as the variable requirements. We should be careful here to associate Annotate variables with the locations to which they refer, not particular map variable names.

Since the horizontal map variable is a numeric variable, we must use the X variable to place data system annotation.

The vertical map variable is also a numeric variable, so we must use the Y variable to place data system annotation.

Appropriate diagnostic messages will be issued by the Annotate Facility if an attempt is made to annotate a numeric value with a character data system reference and vice versa.

All Annotate coordinate placement is based on the MAP= data set. The DATA= data set does not become involved in processing ANNOTATE= observations. Since there is no data communication between the Annotate Facility and the GMAP procedure, we can access only map areas with annotation. This makes it impossible to locate parts of maps which have been relocated or positioned by values contained in the DATA= data set. Data system support is therefore limited to the CHORO statement.
Table 3.1 The possible combinations of chart variable types and the appropriate Annotate variables for PROC GMAP.

Simple application

Most times we use map annotation to locate specific cities and place their names on the generated map. The USCITY data set contains map coordinates for many major US cities, as well as some additional demographic information. Because they have already been projected, the X and Y variables contained in USCITY may be used directly when annotating the US map. (Latitude and Longitude values are also supplied for use with unprojected map data.)

```
/* +-----------------------------------+ */
| Put Orlando on the map. | +-----------------------------------+ */

LIBNAME MAPS 'your map data library';
DATA city;
LENGTH function style color $ a;
RETAIN xsys ysys '2' when 'A' color 'WHITE' when='A';
SET maps.USCITY;
IF state = STFIPS('FL');
IF UPCASE(city) = 'ORLANDO';
THEN DO;
/* Orange makes the '0' */
xsys='2'; ysys='2';
function='pie'; color='orange';
angle=0; rotate=360; size=1.5;
style='solid'; OUTPUT ;
/* 'lando' text string */
xsys='4'; ysys='4';
function='pixxy'; angle=330;
size=1.4; OUTPUT ;
x=.; y=.;
function='cntl2txt'; OUTPUT ;
position='0'; style='BRUSH';
size=2; angle=0; rotate=0;
text=substr(city,2);
function='label'; color='white';
OUTPUT ;
/* leaf */
function='pixxy'; angle=90;
size=0.75; OUTPUT ;
function='cntl2txt'; OUTPUT ;
position='0'; style='WEATHER';
size=2; angle=350; rotate=0;
text='E'; function='label';
color='green'; OUTPUT ;
END;
ELSE DELETE ; /* only use ORLANDO data */
RUN;
```

```
/* +-----------------------------------+ */
| Produce the annotated map. | +-----------------------------------+ */

PATTERN C=green V=empty ;
PROC GMAP DATA=map MAP=map
ID state ;
CHORD state / NOLEGEND ANNOTATE=city ;
RUN;
```

Advanced application

Other times we may use data systems values to set up for a more complicated annotation. We build on the example above to create a logo on the map of Florida. In order to place our logo directly over Orlando, we use the X and Y values from USCITY for the center of an orange. We then continue by adding on the additional pieces, text and a leaf, based on this initial location.

```
/* +-----------------------------------+ */
| Generate an Orlando logo. | +-----------------------------------+ */

DATA city;
LENGTH function style color $ a;
RETAIN ;
SET maps.USCITY;
```

```
IF state = STFIPS('FL') ;
IF UPCASE(city) = 'ORLANDO'
THEN DO;
/* Orange makes the '0' */
xsys='2'; ysys='2';
function='pie'; color='orange';
angle=0; rotate=360; size=1.5;
style='solid'; OUTPUT ;
/* 'lando' text string */
xsys='4'; ysys='4';
function='pixxy'; angle=330;
size=1.4; OUTPUT ;
x=.; y=.;
function='cntl2txt'; OUTPUT ;
position='0'; style='BRUSH';
size=2; angle=0; rotate=0;
text=substr(city,2);
function='label'; color='white';
OUTPUT ;
/* leaf */
function='pixxy'; angle=90;
size=0.75; OUTPUT ;
function='cntl2txt'; OUTPUT ;
position='0'; style='WEATHER';
size=2; angle=350; rotate=0;
text='E'; function='label';
color='green'; OUTPUT ;
END;
ELSE DELETE ; /* only use ORLANDO data */
RUN;
```

```
/* +-----------------------------------+ */
| Produce the annotated map. | +-----------------------------------+ */

PATTERN C=green V=empty ;
PROC GMAP DATA=map MAP=map
ID state ;
CHORD state / NOLEGEND ANNOTATE=city ;
RUN;
```
• Added observations should have a unique value for the PROC GMAP ID variable(s)
• Added observations should be in the same general coordinate system as the existing map
• Adding a single observation per unique ID is sufficient to stretch a particular dimension and not cause the drawing of an unwanted area

/* +------------------------------------------------------*/
| Produce a drop shadow for the US map. |
+------------------------------------------------------*/

LIBNAME MAPS 'your map data library';

DATA shadow;
LENGTH function style color $ 8;
RETAIN color 'BLACK' style 'S' xsys ysys '2';
SET MAPS.us;
BY state segment;
if (state NE stfips('DC')) ;
if (state NE stfips('AK')) and state NE stfips('HI')) ;
IF (FIRST.state OR FIRST.segment )
THEN function = 'POLY';
ELSE function = 'POLYCON';
x = x + 0.01 ; /* shift EAST */
y = y - 0.01 ; /* push SOUTH */
OUTPUT;
RUN;

/* +------------------------------------------------------*/
| Generate a special map with controlled data range. |
+------------------------------------------------------*/

DATA map;
SET HAPS.us END=eof;
if state NE stfips('DC')
if state NE stfips('AK')
state NE stfips('HI')) ;
OUTPUT;

if (eof)
THEN DO;
/* dummy boundaries */
state=-1 ; x=-0.40 ; y=-0.30 ; OUTPUT ;
state=-2 ; x=0.40 ; y=0.30 ; OUTPUT ;
END;
RUN;

PROC GMAP DATA=map MAP=map ;
ID state ;
CHORD state / ANNOTATE=shadow
NOLEGEND COUTLINE=greybb;
RUN;

GCHART PROCEDURE

Because of its number of different chart types and processing options, PROC GCHART is probably the most complicated procedure we will encounter when annotating.

Description of system reference

As with the GPlot procedure, the Annotate data system for PROC GCHART is the graphics area defined along the axis lines from the first major tic mark to the last major tic mark inclusive. No data values exist outside this area.

Highlighting the data area

DATA xy;
RETAIN response 0;
DO midpoint=1 to 5;
response = response + 1;
OUTPUT;
END;
RUN;

DATA range;
RETAIN xsys ysys '2' style 'S' color 'YELLOW';
x = . ; y = . ;
function = 'FRAME'; output ;
RUN;

PATTERN V=SOLID G=GREEN ;
TITLE 'PROC GCHART DATA SYSTEM AREA' ;
PROC GCHART DATA=xy;
VBAR midpoint / DISCRETE SUMVAR=response
ANNOTATE=range ;
RUN;

Since data value annotation is currently supported only on the HBAR and VBAR statements, we will focus our attention on bar charts. GCHART produces two basic styles of bar charts. The VBAR statement produces bars oriented in the vertical direction, the HBAR statement orients its bars horizontally.

If we generalize, then one axis represents the independent variable, or midpoint, the other axis the dependent variable, or response. In a VBAR chart, the midpoint is displayed along the horizontal axis, the response along the vertical. In an HBAR chart, the locations of the midpoint and response are exchanged.
Annotate variables

There are some generalizations we can make about the Annotate variables and their relationship to the GCHART procedure variables.

- The Annotate X variable is always a numeric type and always refers to horizontal, i.e. left-to-right, positions.
- The Annotate XC variable is always a character type and always refers to horizontal positions.
- The Annotate Y variable is always a numeric type and always refers to vertical, i.e. top-to-bottom, positions.
- The Annotate YC variable is always a character type and always refers to vertical positions.
- The Annotate MIDPOINT variable may be either numeric or character type and refers to a midpoint location.
- The Annotate GROUP variable may be either numeric or character type and refers to a group location.
- The Annotate SUBGROUP variable may be either numeric or character type and refers to a subgroup location.

GCHART required variables

The GCHART procedure is capable of plotting both numeric and character valued data. From the rules listed above, we conclude the following about the required Annotate variables for PROC GCHART:

- If the horizontal plot variable is a character variable, we may have the XC variable in our ANNOTATE= data set. If the horizontal plot variable is a numeric variable, we may have the X variable in our ANNOTATE= data set.
- If the vertical plot variable is a character variable, we may have the YC variable in our ANNOTATE= data set. If the vertical plot variable is a numeric variable, we may have the Y variable in our ANNOTATE= data set.

Appropriate diagnostic messages will be issued by the Annotate Facility if these requirements are not satisfied in our ANNOTATE= data set.

Simple applications

Suppose we want to place the word 'TWO' above the second of three vertical bars. The Annotate data system and LABEL function will be used.

```
DATA CHART ;
  INPUT midpoint response @@ ;
  CARDS ;
  1 2 3 3 5
RUN ;
```

The chart is successfully annotated, as shown in Figure 4.1, with the following code.

```
DATA BARTOP ;
  LENGTH FUNCTION $ 8 ;
  RETAIN XSYS YSYS 'z' FUNCTION 'LABEL' TEXT 'TWO' X::.:Z /
  Y=3; /* (response) location */
OUTPUT RUN ;
```

Suppose that we now want to change to an HBAR to display our data. We supply the appropriate HBAR statement to PROC GCHART, and produce the output shown in Figure 4.2.
Why isn't the LABEL in the correct position? Remember, the Annotate X variable always represents the horizontal positions on the chart, and the Annotate Y variable always represents the vertical positions on the chart. By changing to an HBAR, we have changed the orientation of the bars by exchanging the X and Y. As a result, the Annotate variables do not properly correspond to their intended chart positions. We can correct the Annotate data by swapping the X and Y variables.

DATA BAREND;
  SET BARTOP
  RENAME ( X= y y=X )
RUN;

PROC GCHART DATA=CHART;
  HBAR midpoint / DISCRETE SUMVAR=response
  ANNOTATE=BAREND;
RUN;

We can now successfully annotate HBAR and VBAR charts by swapping the Annotate X and Y variables depending on the desired chart type. This method works adequately. But it is not really necessary to produce one version of ANNOTATE= data for an HBAR and a second version for a VBAR. Annotate has special variables for this purpose named MIDPOINT, GROUP and SUBGROUP which are designed to more closely represent the data being annotated. (See the special considerations section for release notes on these variables.)

DATA ANNOBAR;
  LENGTH FUNCTION $ 8 ;
  RETAIN XSYS YSYS '2' FUNCTION 'LABEL' TEXT 'TWO';
  MIDPOINT=2 ; /* VBAR - (midpoint) location */
  Y=3 ; /* - (response) location */
  OUTPUT
RUN;

PROC GCHART DATA=CHART;
  VBAR midpoint / DISCRETE SUMVAR=response
  ANNOTATE=ANNOBAR;
RUN;

The second half of the table illustrates a particular feature of the MIDPOINT variable and simple (no group or subgroup) bar charts. Specifying only the midpoint variable is sufficient to position us at the top of the requested bar!

DATA ANNOBAR;
  LENGTH FUNCTION $ 8 ;
  RETAIN XSYS YSYS '2' FUNCTION 'LABEL' TEXT 'TWO';
  MIDPOINT=2 ; /* VBAR - (midpoint) location */
  /* - no vertical location */
  OUTPUT
RUN;

PROC GCHART DATA=CHART;
  VBAR midpoint / DISCRETE SUMVAR=response
  ANNOTATE=ANNOBAR;
RUN;

You should note this behavior is only active in 'simple' bar charts. There is a similar defaulting behavior pertaining to subgroups when used in a bar chart.

Table 4.2 The possible combinations of chart variable types and the appropriate Annotate variables for PROC GCHART.

Table 4.3 The possible combinations of chart variable types and the appropriate Annotate variables for PROC GCHART.

Table 4.4 The possible combinations of chart variable types and the appropriate Annotate variables for PROC GCHART.

In order to understand how Annotate performs this association, let us consider the following GCHART examples and their outputs (Figures 4.6 and 4.7)

DATA CHART1;
  INPUT midpoint response @@;
  CARDS;
  1 2 2 3 3 5
RUN;

PROC GCHART DATA=CHART1;
  VBAR midpoint / DISCRETE SUMVAR=response;
RUN;
DATA CHART2;
RETAIN subgroup 1;
  INPUT midpoint response @@
  CARDS;
  1 2 3 3 5
RUN;

PROC GCHART DATA=CHART2;
  VBAR midpoint / DISCRETE SUMVAR=response
    SUBGROUP=subgroup NOLEGEND
RUN;

The two outputs are essentially the same. One sample explicitly uses the SUBGROUP= option, the other does not. A clever choice of data to illustrate the following anomaly. Annotate internally considers GCHART as having a subgroup variable, even if the statement involved is not explicitly using the SUBGROUP= option.

If SUBGROUP= is not used in the statement, Annotate assigns each midpoint a dummy index of one (as was done in the CHART2 example above). Thus, SUBGROUP=1 will always be a valid bar position for a midpoint value in the subject chart; i.e. it will position us at the top of the appropriate midpoint bar. In the case of no vertical specification at all, Annotate simply assumed the dummy index of one and continued. This permits us to use the Annotate SUBGROUP as the response variable for either chart type, with or without the SUBGROUP= option being used. If a SUBGROUP= option is used, we specify an appropriate value from our input data. Otherwise, we can specify SUBGROUP=1 and proceed with annotation.

By using the special MIDPOINT and SUBGROUP variables, the burden of locating the display location is placed entirely upon the Annotate Facility and is therefore a generic/portable method for annotating either form of bar chart.

Notice that MIDPOINT, GROUP and SUBGROUP variables may be either numeric or character type. The chart variable type in the GCHART statement will dictate which is appropriate. The Annotate Facility will issue diagnostic messages should a type mismatch occur.

Table 4.5 The possible combinations of chart variable types and the appropriate Annotate variables for PROC GCHART:

<table>
<thead>
<tr>
<th>MIDPOINT</th>
<th>GROUP</th>
<th>SUBGROUP</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHARACTER</td>
<td>CHARACTER</td>
<td>CHARACTER</td>
</tr>
<tr>
<td>CHARACTER</td>
<td>CHARACTER</td>
<td>CHARACTER</td>
</tr>
<tr>
<td>CHARACTER</td>
<td>CHARACTER</td>
<td>CHARACTER</td>
</tr>
<tr>
<td>CHARACTER</td>
<td>CHARACTER</td>
<td>CHARACTER</td>
</tr>
</tbody>
</table>

Annotating with groups

A simple rule for the use of groups with GCHART must be added here for completeness.

If you use a GROUP= variable in GCHART and wish to annotate, you must use the Annotate GROUP variable.

Advanced application

This final example places the sum of all subgroup values at the top of each midpoint bar. Individual values are calculated by a PROC MEANS invocation and match merging data steps. Formats and variable labels are used to enhance the data display.

DATA FOOD;
  DO PARK= 1 TO 3;
    DO LOCATION= 1 TO 3;
      DO ITEM= 1 TO 3;
        NUMBER= FLOOR(5 + RANUNI(98765)*10);
        OUTPUT;
      END;
    END;
  END;
RUN;

PROC FORMAT;
  VALUE STADIUM 1 = 'CODGER'
    2 = 'CRANKY'
    3 = 'CANE';
  VALUE FOOD 1 = 'HOT DOG'
    2 = 'POPCORN';
    3 = 'BEVERAGE';
  VALUE WHERE 1 = '1ST'
    2 = '2ND'
    3 = '3RD';
RUN;

PATTERN1 V=E C=CYAN;
PATTERN2 V=E C=GREEN;
PATTERN3 V=E C=BLUE;

* +--------------------------+
| Create totals for each location |
* +--------------------------+ */

PROC MEANS DATA=FOOD SUM NOPRINT;
  VAR NUMBER;
  BY PARK LOCATION;
  OUTPUT OUT=TOTALS SUM=TOTAL;
RUN;

* +--------------------------+
| Generate the totals |
* +--------------------------+ */

DATA TOTALS;
  MERGE TOTALS FOOD;
  BY PARK LOCATION;
RUN;

DATA TOTALS;
  SET TOTALS;
  IF LOCATION = ITEM;
RUN;
DATA LOCTOTAL;
LENGTH FUNCTION COLOR $ 8;
RETAIN POSITION XSYS YSYS '2' WHEN 'A';
SET TOTALS;
COLOR = 'WHITE';
MIDPOINT = LOCATION;
GROUP = PARK;
SUBGROUP = ITEM;
TEXT = LEFT(PUT(ROUND(TOTAL,1),F2.0));
OUTPUT;
RUN;

PROC GCHART DATA = FOOD;
FORMAT PARK STADIUM.
ITEM FOOD.
LOCATION WHERE . ;
LABEL PARK = 'OO'X ;
LABEL LOCATION = 'OO'X ;
LABEL NUMBER = 'UNITS SOLD';
VBAR LOCATION / MIDPOINTS = 3 2 1
MINOR = 0
SUMVAR = NUMBER
GROUP = PARK
SUBGROUP = LOCATION
CTEXT = GREEN
CAXIS = BLUE
ANNO = LOCTOTAL ;
RUN;

SPECIAL CONSIDERATIONS

Version 5 / Version 6 differences and enhancements

There are some changes and enhancements present in the Version 6 release of the software which may warrant some attention. One change is the addition of the MIDPOINT and SUBGROUP spelling recognition. Versions of the SAS/GRAPH Annotate Facility prior to Version 5.18, named these variables MISPNT and SUBGRP. Later releases support the new and old spellings of these variables, and will use the new over the old to resolve conflicts should both spellings appear on the ANNOTATE data set. You are encouraged to convert to the new spelling as your site software is upgraded.

Critical aspects of BY processing with Annotate

BY processing is supported by the Annotate Facility in releases after Version 5.08. Because there are two different data sets involved in the procedure level handling of the BY processing, one should note the following requirements for Annotate BY processing.

The procedure BY variable(s) must also be in the ANNOTATE data set, and must match the procedure variable processing attributes. These attributes include data type, length, label and format. Mismatches of these attributes will cause the Annotate Facility to return without processing the input data set.

GLOBAL / LOCAL statement specification

We should also note that only the procedure action statements support the BY processing features. The PROC statement invocation is reserved for the addition of annotation regardless of the state of BY processing.

SAS and SAS/GRAPH are registered trademarks of SAS Institute Inc., Cary, NC, USA.