SAS/Graph® Annotate Coordinate Systems by Example
Max Cherny, GlaxoSmithKline, Collegeville, PA

ABSTRACT
The Annotate facility is a very powerful tool within SAS/Graph. It is used to create or enhance SAS graphics by placing graphical elements such as lines or symbols anywhere on a screen. However with this power comes a certain degree of complexity which makes learning the Annotate facility a difficult task.

One of the most challenging aspects of the Annotate facility is its use of graphical coordinates. There are 12 different systems which control the placement of a coordinate. A SAS user needs to understand how each system identifies the exact location of a coordinate in order to judiciously use the facility. Some of these systems are complicated, for example the ones which use relative position of a coordinate.

This paper explains the Annotate coordinate systems and provides an easy and reusable example for each of them. These examples can function either as the stand-alone programs or as the starting point for further experimentation with the Annotate coordinate systems. These programs can also serve as the basic general introduction to the Annotate facility.

Additionally, the programs explain POSITION and MIDPOINT Annotate variables. The powerful, but underutilized TXT2CNTL and CNTL2TXT functions used to capture the last position of a graphical element are discussed as well.

INTRODUCTION
SAS/Graph is a very powerful application which allows SAS users to create different types of graphs. Most graphs can be created by using various SAS/Graph procedures and options. However, sometimes these procedures and options may not be sufficient to draw complex graphs. Annotate facility are used to enhance output of SAS/Graph procedures. It can also be used to create entire plots.

The Annotate facility allows plot any graphical elements (i.e. lines, labels, symbols) by manually specifying their coordinates on the drawing area. Coordinates for such elements can be expressed as X and Y if they are numeric and XC and YC if they are characters. Depending on a type of a graphical element, its positions can also be expressed by using MIDPOINT, POSITION or SUBGROUP variables. The coordinate systems can also control the size of a graphic element by using HSYS variable.

Coordinates are translated to a position on the drawing area according to the coordinate systems There are 12 different coordinates systems used in the Annotate facility : ‘1’, ’2’, ’3’, ’4’, ’5’, ’6’, ’7’, ’8’, ’9’, ’A’, ’B’ and ’C’. Coordinates ’1’ to ’6’ are referred to as Absolute and use the exact positions of a graphical element on the drawing area. Coordinates ’7’ to ’C’ are called Relative. They are used to place a graphical element relative to a position of another graphical element.

Coordinates ’1’, ’3’, ’5’, ’7’, ’9’ and ’B’ use percentages of the drawing area and are specified in percentage units. Coordinates ’2’ and ’8’ use axis values. Coordinates ’4’, ’6’, ’A’ and ’C’ use units measured in cell values. Cell values depend on the device and number of rows and columns available in the drawing area. The number of cell and columns depend on the graphic settings for a given graph.

BRIEF OVERVIEW OF THE ANNOTATE FACILITY
The annotate facility uses commands to place graphical elements on the drawing area. These commands are contained in an Annotate data set which is then invoked during the execution of SAS/GRAPH procedures such as GPLOT or GCHART. These commands can be executed by PROC GANNO as well.

Every observation in the Annotate data set indicates which graphical element is drawn, its position and its characteristics. There are many functions and variables available in the Annotate facility. The most common Annotate variables are:

| X and Y  | Indicate position of a graphic element |
| HSYS and YSYS | Indicate coordinate system |
| TEXT  | Text of a label |
| SIZE  | Indicates size of a graphic element |

The most frequently used Annotate functions are:
<table>
<thead>
<tr>
<th>LABEL</th>
<th>Draws text on the drawing area</th>
</tr>
</thead>
<tbody>
<tr>
<td>LINE</td>
<td>Connects two points with a line. Each point has x and y coordinates.</td>
</tr>
<tr>
<td>MOVE</td>
<td>Moves to a specific position</td>
</tr>
</tbody>
</table>

These variables and functions can be used to create or enhance most graphs. They can be specified in a data set by either using explicit statements such as:

```plaintext
function="MOVE"; x=10; y=20;
```

The same function can be coded by using Annotate macros as follows:

```plaintext
%move(10,20);
```

These statements move to a coordinate having values of X as 10 and Y as 20. Note that using Annotate macros requires invoking `%ANNOMAC` statement at least once in the program.

**SAS/GRAPH DISPLAY AREAS**

There are three types of display areas used by the Annotate facility: Data Area, Procedure Output Area and Graphics Output Area. The Data Area is the drawing area of the plot itself. Figure 1 displays plot of change from baseline in Body Mass Index (BMI) over time. The shaded area represents the Data Area.

![Data Area](image1.png)

**Figure 1: Data Area**

The area which includes plot itself, labels and legend is referred to as the Procedure Output Area and is represented on Figure 2 as the shaded area.

![Procedure Output Area](image2.png)

**Figure 2: Procedure Output Area**

The entire drawing area is called the Graphics Output area. It includes the Data and the Procedure Output Areas as well as areas reserved for footnotes and titles. This is why Figure 3 is completely shaded.
Figure 3: Graphics Output Area

Note that the Graphics Output Area is not necessarily the same area as the entire screen. Many graphic devices contain margins. Figure 4 displays the Graphics Output Area with a left margin. The left margin was created on the “Win” device using the HORIGIN=0.2in option.

Figure 4: Graphics Output Area with Left Margin

For most graphs the horizontal dimension of the Procedure Output Area will be the same as of the Graphics Output Area, unless a vertical title or a footnote is drawn. The relationship between these 3 display areas can be seen on the Figure 5.

Figure 5: All Graphics Areas
COORDINATE SYSTEM ‘1’

This coordinate system is one of the easiest to use. It refers to the Data Area and uses percentages of the axis within the Data Area to establish position of a graphics element.

The following example illustrates the use of Coordinate System ‘1’. The data set below represents change from baseline in BMI over several months. This data set will be used in many examples in this paper.

```sas
data bmi_data;
    length Drug $1 ;
    input Drug $1 Month 3 Mean 4-9 n 11-14 std 15-19 high 20-24 low 25-30 ;
datalines;
A 1 0 200 2.6
B 1 0 200 2.8
A 2 -12.5 170 2.6 -9.8 -15.1
B 2 -4.08 170 2.7 -1.2 -6.87
A 3 -19.4 140 2.5 -16.9 -22.0
B 3 -4.15 140 2.7 -1.4 -6.8
A 4 -22.5 130 2.4 -20.1 -25.07
B 4 -0.15 130 2.7 2.62 -2.95 ;
run;
```

The code below creates a plot of BMI over time shown on Figure 1:

```sas
axis1 order=(-40 to 10 by 5) value=(f="Times New Roman" h=1.5);
axis2 minor=none offset=(5,5) value=(f="Times New Roman" h=1.5) ;
symbol1 i=j c=black v=square ;
symbol2 i=j c=black v=dot;
proc gplot data= bmi_data ;
plot mean*month=Drug / vaxis=axis1 haxis=axis2 anno=anno ;
run;
```

Let’s assume we need to draw the label, “Change in BMI”, within the Data Area in the center of the graph and below the top horizontal axis. Additionally, the label must be in the same location regardless of the scale of the graph. The following code creates an Annotate data set containing commands to draw such a text:

```sas
%annomac; /* invoke annotate macros */
data anno; /* create annotate data set */
    %system(1,1);
    %label(50, 80, "Change in BMI", , 0, 0 , 2, 'Times New Roman', 6);
run;
```

<table>
<thead>
<tr>
<th>XSYS</th>
<th>YSYS</th>
<th>HSYS</th>
<th>X</th>
<th>Y</th>
<th>ANGLE</th>
<th>ROTATE</th>
<th>SIZE</th>
<th>STYLE</th>
<th>TEXT</th>
<th>POSITION</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>50</td>
<td>80</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td></td>
<td>'Times New Roman'</td>
<td>Change in BMI</td>
<td>6</td>
<td>LABEL</td>
</tr>
</tbody>
</table>

In order to use this dataset, the anno option must be used in the procedure call:

```sas
proc gplot data= bmi_data ;
    plot mean*month=Drug / vaxis=axis1 haxis=axis2 anno=anno ;
run;
```

The graphical output is shown on the Figure 6.
Figure 6: Example of the Coordinate System ‘1’

Figure 7 displays the same plot in the Figure 6 adding a grid representing the percentages of the Data Area. This graphic demonstrates how using 40% of the horizontal axis and 80% of the vertical axis creates a text label in that position. The text is centered and left aligned by setting the POSITION variable to 6. The POSITION variable controls the exact position of text.

Figure 7: Example of the Coordinate System ‘1’

The use of coordinate system ‘1’ guarantees that the position of the graphical element will remain the same regardless of the data and the format of the graph. The position of the text does not depend on the scale of the plot. As illustrated on the Figure 8, the label will still appear in the same position even if the BMI data is changed.
This may be a problem if the text overlaps with the plot. Therefore the position of the label must be carefully chosen.

**POSITION VARIABLE**

Note that the text string in the Figure 6 begins exactly at the coordinate that coincides with 40% and 80% of the x and y axes, respectively. The exact position of the text string and its alignment can be controlled by the POSITION annotation variable. There are 16 possible values of POSITION represented on the Figure 9. The small circle represents a point defined by X and Y variables. The default position is 5 (center).

```
Text string: Position 2  Text string: Position 3  Text string: Position 1
Text string: Position 5  Text string: Position 6  Text string: Position 4
Text string: Position 8  Text string: Position 9  Text string: Position 7
Text string: Position A Text string: Position C
Text string: Position D Text string: Position F
```

**Example of Position 0**

Position 0 retains the last position of a text string. The text can then be continued at the position. Figure 1 demonstrates the effect of changing the POSITION variable to 5. The value of 5 places the middle of the text string exactly at 40% of the axis.
Figure 10: Effect of POSITION Value on Text String

COORDINATE SYSTEM ‘2’
This coordinate system is the most frequently used coordinate system. It uses the values of the axis. Usually the values of the axis would be based on the plotted data. SAS generates an error if the requested coordinate of a graphical element is outside the axis.
Figure 11 provides an example of using the coordinate system ‘2’ in order to add whiskers to the plot. Top and bottom whiskers represent low and high values, respectively.

Figure 11: Example of the Coordinate System ‘2’
The following code will create the annotate data containing positions of the lines constructing the whiskers. The low and high values must exist prior to creating the Annotate dataset.
data anno;
   set bmi_data;
   if month ne 1; /* do not use create whiskers for the baseline month */
   %system(2,2); /* use Coordinate system '2' */
   /* Draw a line connecting high and low values for each month */
   %line(month,high,month,low,black,1,1);
   /* Draw a small horizontal line on top of each high value
      This is done by connecting a point a little bit to the left of high
      value to a point a little bit to the right of the high value. This
      little distance is 0.04 */
   %line(month-0.04,high,month+0.04,high,black,1,1);
   /* Draw a small horizontal line below each low value */
   %line(month-0.04,low,month+0.04,low,black,1,1);
run;

The GPLOT procedure statement remains the same:
proc gplot data= bmi_data ;
   plot mean*month=Drug / vaxis=axis1 haxis=axis2 anno=anno ;
run;

Figure 12: Example of the Coordinate System ‘2’

The lines are drawn by connecting two points. As shown on the Figure 12, the horizontal coordinates correspond to the MONTH variable and Y coordinates correspond to High and Low, respectively. The %LINE statement connects these points.

COORDINATE SYSTEM ‘3’
While the coordinate systems ‘1’ and ‘2’ reference coordinates within the Data Area, the coordinate system ‘3’ can plot graphical elements by using the percentage of the entire drawing area. This means that coordinates will never change. This system should be used for graphs where graphical elements are required to be always placed in the same position. Since the entire drawing area is used, the coordinates may lie within a plot as well as outside the plot.

Figure 13 shows a bar chart representing frequencies of patients in the BMI dataset at each month. The month is displayed to the left of the plot and the total number of patients is displayed to the right of the bar using default PROC GCHART option.
Figure 13: Bar Chart Using Default Options

This figure was generated by the following code:

```plaintext
legend1 position=(inside right) across=1 mode=protect;
axis2 label=none origin=(7pct);
axis1 label=none ;
proc gchart data=bmi_data ;
  hbar month/discrete subgroup=drug sumvar =n width=3
  raxis=axis1 maxis=axis2 sum legend=legend1;
run;
quit;
```

Figure 14 displays the same bar chart, however the Annotate facility was used to display the total number of patients to the left of the plot. The label "SUM" is suppressed and "n" is drawn above the total numbers of patients. The horizontal position of the total number of patients will be always at 10% of the entire drawing area. Such approach will be necessary if multiple graphs are needed to be generated and consistency in appearance among the graphs is required.

Figure 14: Enhanced Bar Chart Using Coordinate System ‘3’

```plaintext
/* calculate total number of patients at each month */
proc means data=bmi_data sum noprint;
  var n;
  by month;
  output out=sum sum=sum;
run;

data anno;
  set sum;
  %system(3,2); /* use coordinate system '3' for horizontal position */
  when="a";
```
midpoint=month; /* Use MIDPOINT variable to reference the month*/
x=10; /* position text at 10% of the x axis */
text=cat(sum); /* Create text label representing number of patient*/
output;

/* Use coordinate system 1 to display n at the 90% of vertical axis*/
%system(3,1);
y=95;
text="n"; /* Display the word n*/
output;
run;

legend1 position=(inside right) across=1 mode=protect;
axis2 label=none origin=(7pct) offset=(,8pct);;
axis1 =none offset=(5pct) ;
proc gchart data=bmi_data anno=anno;
  hbar month/discrete subgroup=drug sumvar =n w
  raxis=axis1 maxis=axis2 sum legend=legend1
  annotate=anno nostats vref=0.1;
run;
quit;

The number of patients is displayed in the area before the bars in Figure 14. This area is created by using the OFFSET option. The OFFSET option specifies the distance from the first and last major tick marks or bars to the ends of the axis line. Setting the value of the WHEN variable to ‘a’ draws the text after the main plot is drawn. Note the use of MIDPOINT variable in the program. This MIDPOINT variable references the position of bars representing months.

Figure 15: Details of the Enhanced Bar Chart Using Coordinate System ‘3’

It is important to notice that this program is setting coordinate system to ‘1’ only for the horizontal axis. The Y coordinates are derived from using the MIDPOINT variable. A separate coordinate system is required to specify each coordinate of a graphical element. These coordinate systems do not have to be the same. The only disadvantage of this method is that the total number of patients must be calculated prior to using the Annotate facility, while the SUM label in Figure 13 is generated automatically by PROC GCHART.

COORDINATE SYSTEM ‘4’

This system does not use percentages or axis values. Instead, absolute values of cells are used to position a coordinate within the entire drawing area. External dimensions of the graph and the number of rows and columns define the characteristics of the cells. Figure 16 displays the relationship between the cells and the graphic
properties. There is not a relationship between the number of cells in a graph and the actual screen size. The number of cells in the graphics output area is determined by the device driver that is used to generate the graph, and can be modified by using the HPOS= and/or VPOS= options on a GOPTIONS statement. Therefore any coordinate defined in cell units will depend directly on graphic properties of the device.

**Rows, Columns, and Cells in the Graphics Output Area**

![Diagram showing rows, columns, and cells in the graphics output area](image)

**Figure 16: Rows, columns and cells (SAS Help and Documentation)**

**Figure 17: Coordinate System ‘4’. Default HPOS for WIN Device**

For example, the WIN device only allows specification of up to 100 cells. These 100 cells do not cover the entire screen as can be seen on Figure 17 where the grid at the bottom corresponds to cell units. 100 cells cover about two thirds of the screen. In order to cover the entire area of the plot the value of HPOS option must be modified. Figure 18 demonstrates how changing the HPOS option to 101 extends the cell units to cover the entire plot area.
Figure 18. Coordinate System ‘4’. WIN device where HPOS=101

The cell units on the specific device can be determined by experimentation. Figure 19 demonstrates a line which was drawn under the "Change in BMI" label using coordinates defined in cell units.

```markdown
data anno;
/* use coordinate system ‘4’ for horizontal coordinates */
/* use coordinate system ‘1’ for vertical coordinates */
%system(4,1);
/* draw a line connecting a point at x=42 cells/y=75% with a point at x=60 cells/y=75% */
%line(42,75,60,75,,1,2);
run;
```
Since the coordinate system using cell units depends on the size options and is affected by the device driver, the author does not recommend this system.

**COORDINATE SYSTEM ‘5’**

The coordinate system ‘5’ uses percentages of the available drawing area in the Procedure Output Area. The horizontal coordinates will usually be the same as the horizontal coordinates in the system ‘3’ (the entire screen). The coordinates will be different in some cases, for examples, if titles or footnotes are placed to the left or right of a plot.

The line plot in Figure 20 has been enhanced by adding the number of subjects at each month. This was done by placing the number at the 10% from the bottom within the Procedure Output Area.

![Figure 20: Placing Number of Subjects at Each Month Using the Coordinate System ‘5’](image)

Note the X coordinate is using the MONTH variable in the coordinate system ‘2’. Even though the coordinate system ‘2’ is referencing the Data Area, its positions actually extend beyond the Data Area and into the entire Procedure Output Area.

```sas
data anno;
  set bmi_data;
  %system(2,5);
  text="n=" || cat(n); /* create text variable for number of subjects */
  %label(month,12, text, black, 0, 0 ,1 ,'' , 5);
run;
```

The PROC GPLOT statement is the same as in the coordinate system ‘1’ example. Figure 21 explains how the text label is drawn. The shaded area represents the Procedure Output Area and the positions of the MONTH variable extends beyond the Data Area.
The coordinate systems ‘5’ and ‘3’ are very similar to each other. The key difference between the systems is that coordinates in system ‘5’ do not extend to the area reserved for titles and footnotes.

**COORDINATE SYSTEM ‘6’**
The coordinate system ‘6’ is very similar to the system ‘4’ since it references positions of graphical elements in cell units. However coordinates in system ‘6’ do not extend to the area reserved for titles and footnotes. The author does not recommend using the system.

**COORDINATE SYSTEM ‘7’**
The coordinate system ‘7’ creates a graphical element relative to a position of another graphical element. This coordinate system is more complex to use than the previously described systems. However, it allows one to write more efficient and portable programs.

Figure 22 is again using the plot of BMI over time. However this time the BMI label contains its units (kg/m$^2$). The complexity here is in drawing the superscript. The superscript can be created by a number of different techniques. One approach is to use a special font or a symbol to display 2 as superscript. Another approach is to ‘hardcode’ the coordinates of 2. However these approaches require one to indicate the exact position of number 2. Instead the coordinate system ‘7’ places “2” relative to the end of the unit text as superscript. The relative distance is expressed as the percentage of the Data Area.
The advantage of this approach is that the position of 2 is not fixed to a specific coordinate. Instead, it depends only on the position of the previous text string.

**COORDINATE SYSTEM ‘8’**

This system is very similar to the previous system, however instead of percentages, the axis values within the Data Area are used to reference the coordinates of graphical elements.

**COORDINATE SYSTEM ‘9’**

This system uses relative position of a graphical element within the entire drawing area.

Figure 23: Example of the Coordinate System ‘9’

Figure 23 displays a simple forest plot graph. The “favor” text and the subsequent arrows are drawn using the coordinate system ‘9’.

data forest_plot;
  length Group $8 ;
  input Group 1-8 hr 9-12 low 13-17 high 18-20 groupcd 22-24;
  datalines;
  Group 1 0.98 0.6 1.4 1
  Group 2 1.1 0.3 1.5 2
  Group 3 0.9 0.2 1.1 3;
  run;

  /* format subgroups*/
  proc format;
  value group
    1="Subgroup A"
  run;
2="Subgroup B"
3="Subgroup C"
run;
symbol i=none v=dot;
axis1 order=(-0.5 to 3 by 0.5) length=50pct label=none;
axis2 label=none offset=(10pct,10pct) minor=none length=50pct;

/* Start ANOMAC macros*/
%annomac;

/* draw confidence level lines using coordinate system 2*/
data ci_lines;
%system(2,2);
set forest_plot;
%line(low,groupcd,high,groupcd,black,1,1);
run;
data hr_arrows;
length text style $30;
%system(2,1);
/* Use system 2 for the horiz. coordinate and 1 for vertical */
%move(0,0); /* Move to coordinate 0,0 */
%system(2,9);/* Use system 2 for the horiz. coordinate and 9 for vertical */
%cntl2txt; /* capture the current x and y coordinates */
/* Move to position 1.1 on X axis and down by 10% of the drawing area */
/* draw Favours Study Drug at that position */
/* Use position C to right align the text string */
%label(1.1,-10,"Favours Study Drug",black,0,0,1.7,'Times New Roman',C);
%txt2cntl; /* capture the x and y coordinates of the label*/
/* Use coordinate system 2 for the horizontal coordinate and 9 for vertical */
%system(9,9);
%move(2,1); /* Move 2% right and up by 1% of the drawing area */
%system(1,9);
/* Draw a line to 99% of the data area (the end of x-axis). Keep the same Y coordinate */
%draw(99,0,black,1,1);
%cntl2txt; /* Capture the x and y coordinates of the end of the line */
%system(9,9); /* Use coordinate system 9 */
/* Draw string B at the end of the line. The use of Marker font will display the string as arrow end*/
/* Use position + to place arrow end at the center of the line*/
%label(0,0, "B",black,0,0,0.7,'Marker',+);
/* The code to display the left arrows is omitted but it can be done in the similar fashion. */
run;

/* combine two annotate datasets together */
data anno;
set ci_lines
hr_arrows;
run;

proc gplot data=forest_plot; /* create plot */
plot groupcd*hr /haxis=axis1 vaxis=axis2 href=1 lhref=2 anno=anno nolegend;
format groupcd group.;
run;
quit;
Figure 24: Coordinate System ‘9’

Note the use of the %CNTL2TXT and %TXT2CNTL functions in the program. They are used to capture the last coordinates of the non-text and text graphical elements, respectively. A new graphical element can then be drawn relative to that position. The same plot can be also created by simply specifying the exact positions of graphical elements on the graph. However if there are changes to either the text of the “Favor” string or the horizontal axis those positions will not work correctly. The code in the example will automatically adjust the coordinates for any axis or text strings.

COORDINATE SYSTEMS ‘A’ AND ‘C’

These systems use relative positions of graphical elements expressed in cell units within the Procedure Output Area and the Graphics Output Area, respectively. The use of these systems should be avoided.

COORDINATE SYSTEM ‘B’

This system is similar to the system ‘9’ and uses relative position of a graphical element as percentages of the Procedure Output Area.

CONTROLLING SIZE OF A GRAPHICAL ELEMENT

Size of a graphical element can be controlled by setting HSYS Annotate variable to one of the coordinate systems.

<table>
<thead>
<tr>
<th>Hsys</th>
<th>size</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td><img src="example1.png" alt="Example" /></td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td><img src="example2.png" alt="Example" /></td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td><img src="example3.png" alt="Example" /></td>
</tr>
</tbody>
</table>

LIMITATIONS OF COORDINATE SYSTEMS

While various coordinate systems allow placing the coordinates of graphical elements in almost any area of the drawing area, there are some limitations. For example, there is not a specific coordinate system that will automatically place annotation between bars on a horizontal bar chart. However, some areas of a bar can be dynamically referenced by using MIDPOINT and POSITION variables.

CONCLUSION

The Annotate facility offers a variety of ways to reference coordinates of a graphical element. Using a coordinate system appropriate for a given task can make SAS programs more efficient and portable. The most popular coordinate system is ‘2’ because it uses values of axis and thus allows referencing the plotted data. Most programs will use this system for at least one of the two required coordinates. The coordinates systems ‘3’ and ‘5’ can be used
to reference positions outside of the Data Area. The coordinate systems ‘4’, ‘6’, ‘A’ and ‘C’ all use cell units and should be avoided since the positions will change based on the size options and device drivers. Coordinates systems ‘7’, ‘8’, ‘9’ and ‘B’ use relative positions of graphical elements. The use of such systems is relatively complex. However, use of such systems is recommended in a graph in which the coordinates of graphical elements can change every time the plotted data is changed. If properly used, the relative coordinate systems can allow for automatic placement of the graphical elements in a graph.

<table>
<thead>
<tr>
<th>Type of Coordinates</th>
<th>Area</th>
<th>Units</th>
<th>Range</th>
<th>System Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Absolute</td>
<td>Data</td>
<td>%</td>
<td>0-100% of axis</td>
<td>1’ *</td>
</tr>
<tr>
<td></td>
<td>Data</td>
<td>values</td>
<td>minimum to maximum of axis</td>
<td>2’ *</td>
</tr>
<tr>
<td></td>
<td>graphics output area</td>
<td>%</td>
<td>0-100% of graphics output area</td>
<td>3’</td>
</tr>
<tr>
<td></td>
<td>graphics output area</td>
<td>cells</td>
<td>0 to limit of graphics output area</td>
<td>4’</td>
</tr>
<tr>
<td></td>
<td>procedure output area</td>
<td>%</td>
<td>0-100% of procedure output area</td>
<td>5’</td>
</tr>
<tr>
<td></td>
<td>procedure output area</td>
<td>cells</td>
<td>0 to limit of procedure output area</td>
<td>6’</td>
</tr>
<tr>
<td>Relative</td>
<td>data</td>
<td>%</td>
<td>0-100% of axis</td>
<td>7’ *</td>
</tr>
<tr>
<td></td>
<td>data</td>
<td>values</td>
<td>minimum to maximum of axis</td>
<td>8’ *</td>
</tr>
<tr>
<td></td>
<td>graphics output area</td>
<td>%</td>
<td>0-100% of graphics output area</td>
<td>9’</td>
</tr>
<tr>
<td></td>
<td>graphics output area</td>
<td>cells</td>
<td>0 to limit of graphics output area</td>
<td>A’</td>
</tr>
<tr>
<td></td>
<td>procedure output area</td>
<td>%</td>
<td>0-100% of procedure output area</td>
<td>B’</td>
</tr>
<tr>
<td></td>
<td>procedure output area</td>
<td>cells</td>
<td>0 to limit of procedure output area</td>
<td>C’</td>
</tr>
</tbody>
</table>

Coordinate System Values (SAS Help and Documentation)

In addition to the coordinate systems, the placement of graphical elements can be done by POSITION, SIZE and MIDPOINT variables.

ACKNOWLEDGMENTS

Author would like to thank Greg Cicconetti, Ph.D. for his help with this paper.

REFERENCES


CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:
Max Cherny
GlaxoSmithKline
Email: chernym@yahoo.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. © indicates USA registration. Other brand and product names are trademarks of their respective companies.