Creating Graphs with Your Choice of Axes Using Annotate
Abraham T Pulavarti, ICON Clinical Research, North Wales, PA

ABSTRACT
In Clinical Research, Programmers and Statisticians often have to produce a variety of graphs and in many instances the standard output from the SAS/GRAPH procedures must be altered and customized to suit individual requirements. One such requirement is to re-label and customize the axes. By default SAS/GRAPH procedures uniformly distribute or equally space the values on the axes. This built-in functionality needs to be changed in situations where the axes are to be drawn to scale, where, for example, the visits in a clinical trial are not conducted at equal intervals. This paper discusses in detail the different ways to overcome this problem of customizing the axes to individual needs and also presents simple solutions to modify the labels on the axes. This paper will show the graphic options available to address the issue and also put forth an efficient solution using the Annotate Facility. The basic concepts of the Annotate Facility are introduced by presenting the important principles of Coordinate systems, units for Coordinate values and built-in Annotate macros. This paper gives simple examples on how to use these powerful tools in generating high quality graphs.

INTRODUCTION
This paper introduces the fundamentals of the Annotate Facility and then discusses in detail the methods involved in customizing the axes. This paper is intended for Intermediate level SAS Programmers. Much of the material is taken from the SAS/GRAPH Reference Manual. At the end, there are references to past SAS user group presentations which are an excellent resource to get an overall exposure to the Annotate Facility. This entire paper uses the ‘trial’ dataset given below and puts forth six methods to customize the X-axis, where the first two methods use the graphic options available in the AXIS statements and the other four methods use the Annotate Facility.

CREATING THE ‘TRIAL’ DATASET:
This sample dataset presents a hypothetical subject who is part of a clinical trial to test a new drug to evaluate subjects with Hypercholesterolemia. The efficacy variable, ‘Total Cholesterol’ is collected at Week 1, 2, 5, 7, 10 and 12. The investigator wants to look at the patient profile to understand how the drug has worked. A figure has been requested which has the ‘Total Cholesterol’ over time where time is drawn to scale. Since the weeks are not equidistant, the X-axis has to be drawn to scale so that the distance from Week 2 to Week 1 is not the same as the distance from Week 5 to week 2.

DATA trial;
  input week tc;
datalines;
1 310
2 295
5 260
7 225
10 240
12 250
;
RUN;

Let’s take a moment to look at the standard output from SAS/GRAPH without using any techniques mentioned in this paper. The code used to create this plot is shown below.

goptions ftext='Arial' htext=2 gunit=pct;
symbol v=dot I=join h=3;
axis1 order=(210 to 320 by 10) label=(angle=90 'Total Cholesterol (mg/dL)')
  minor=(n=1) offset=(2,2);
axis2 order=(1 to 2 by 1 5 to 7 by 2 10 to 12 by 2) minor=none label=('Weeks')
  offset=(2,2);
title height=3 'Standard Output Without Manipulation';
PROC GPLOT data=trial;
  plot tc*week/vaxis=axis1 haxis=axis2;
RUN; QUIT;
The above figure provides an illustration where Weeks 1, 2, 5, 7, 10 and 12 are evenly spaced along the x-axes. This representation of the data as shown in the graph above is misleading since the actual time points are not at equal intervals. The distance between Week 2 and Week 1 should not be the same as the distance from Week 5 to Week2. To address this problem the following methods can be used.

METHOD ONE:
This method makes use of the 'VALUE' option in the AXIS statement.

```plaintext
value=('1' '2' '' '' '5' '' '7' '' '' '10' '' '12');
```

Setting the 'value' option in this way, forces blank x-axis tick-marks into the positions where no study visit actually took place, thereby distributing the rest of the x-axis tick-marks according to their relative values. Usage is shown in context below (in bold).

```plaintext
options ftext='Arial' htext=2 gunit=pct;
symbol v=dot I=join h=3;
axis1 order=(210 to 320 by 10) label=(angle=90 'Total Cholesterol (mg/dL)')
  minor=(n=1) offset=(2,2);
axis2 order=(1 to 12 by 1) minor=none label=('Weeks') offset=(2,2)
  value=('1' '2' '' '' '5' '' '7' '' '' '10' '' '12');
title height=3 'Method One – Using Value Option';
PROC GPLOT data=trial;
  plot tc*week/vaxis=axis1 haxis=axis2;
RUN; QUIT;
```
Method One - Using Value Option

![Graph showing total cholesterol levels over weeks](image)

**METHOD TWO:**
This method uses the ‘TICK’ and the ‘VALUE’ options in the AXIS statement as shown below in bold.

```sas
options ftext='Arial' htext=3 gunit=pct;
symbol v=dot I=join h=2;
axis1 order=(210 to 320 by 10) label=(angle=90 'Total Cholesterol (mg/dL)')
          minor=(n=1) offset=(2,2);
axis2 order=(1 to 12 by 1) minor=none label=('Weeks') offset=(2,2)
          value=(tick=3 '' tick=4 '' tick=6 '' tick=8 '' tick=9 '' tick=11 '');
title height=3 'Same as Method One';
PROC GPLOT data=trial;
   plot tc*week/vaxis=axis1 haxis=axis2;
RUN;
quit;
```

This method gives the same output as Figure 1 above. Both of the above methods have successfully created a figure where the Weeks on the X-axis are not equidistant to each other. But for someone who finds unlabeled tick marks annoying, and who wants a graph where the tick marks are shown only for Weeks where visits were actually scheduled then we need to make use of the great Annotate Facility.

**USING THE ANNOTATE FACILITY**
The Annotate facility allows the user to customize and modify graphic output. Through the use of the Annotate facility, one can greatly extend the powerful capabilities of SAS/GRAPH. The Annotate facility generates a dataset of graphic commands from which one can produce graphics output.

To use the Annotate Facility, first determine what you want to draw and the coordinates specifying where you want it
positioned on the graphics output. Then build an Annotate data set composed of specifically named variables. Each observation is interpreted as a graphics primitive (which draws a graphic element) or programming functions. Each observation contains variables that specify what graphic element to draw, where to draw it, and how to draw it. To produce the graphics output, submit a call to a SAS/GRAPH procedure, including the ANNOTATE= option in the appropriate statement.

```sas
PROC GPLOT data=trial;
   plot tc*week/vaxis=axis1 haxis=axis2 annotate=ann;
```

Each observation in an Annotate dataset represents a command to draw a graphics element or perform an action. The values assigned to the Annotate variables in each observation determine what is done and how it is done. Annotate variables have predefined names. In each observation, the Annotate facility looks only for those names. Other variables can be present, but they are ignored.

Conceptually, there are three types of variables.

1. Function Variables: Tell the Annotate facility what to do – these specify what graphic elements to draw (graphics primitive) or what action to take (programming function)
2. Positioning Variables: Tell the Annotate facility where to do it – these specify the coordinates at which to draw the graphics element.
3. Attribute Variables: Tell the Annotate facility how to do it – these specify the attributes of the graphics primitive. (For example, color, size, line style, text font, etc).

### What to Draw: Graphics Primitives
Graphics primitives draw graphic elements. In the Annotate facility, the FUNCTION variable determines the graphics element that is drawn or the programming function that is performed. Examples include `Function='Move'` or `Draw'` or `Bar' or 'Point' or 'Pie' etc.

### Where to Draw: Coordinates
Coordinates specify where to put graphic elements. The X, Y and Z variables are used for numeric coordinates. XC and YC are used for character coordinates. Coordinates are interpreted in the context of a specific coordinate system in order to identify a precise location in the graphics output.

### Coordinate Systems
A Coordinate system determines how coordinates are interpreted. The coordinate system to be used for each axis is specified using `XSYS`, `YSYS`, and `ZSYS` variables (for X, Y and Z respectively).

### Internal Coordinates
The Annotate facility maintains two pairs of internal coordinates. These are the coordinates of the last graphics element drawn or the coordinates from the last move (XLAST, YLAST) and the coordinates of the last text element drawn (XLSST, YLSST). These internal coordinates can provide a convenient reference point for further Annotate commands.

A set of Annotate macros is provided in the SAS sample library. You can use macro calls in a DATA step to create observations in an Annotate data set (use of these macros can greatly reduce the number of statements required to generate an Annotate dataset). You can also use Annotate macros and explicit variable assignments together in the same DATA step. The general form of a call to an Annotate macro is as follows:

```sas
%macro-name (parameters);```

The Annotate macros used in this paper are as follows:

- **Annotate Macros**:
  - `%LABEL(x, y, text, color, angle, rotate, size, style, position)`: Draws text
  - `%MOVE(x, y)`: Moves to a point
  - `%DRAW(x, y, color, line, size)`: Draws line

The most important concept to understand when creating an Annotate dataset is the Coordinate system. Below is a figure showing the different graphics areas along with their respective Coordinate systems. Master this figure and you have mastered Annotate Facility.
Each coordinate system refers to one of the three drawing areas: Data area, graphics output area and procedure output area. Coordinates are measured from a different origin for each area and they also have different limits. The units in the above figure are based on data values (range of values expressed along the axes), cells (range of values depending on the type of area, the device and the number of rows and columns set by the graphic options) and percentages (percentage of the total area available). As mentioned previously, coordinate systems are selected by means of the ‘XSYS’, ‘YSYS’, and ‘ZSYS’ variables; specifically, these variables are set to one of the values 1-9, A, B, C shown in Figure A. This will be illustrated in next example.

The placement of the coordinate can be absolute or relative. Absolute coordinates name the exact location for a graphics element in the graphics output. Relative coordinates name the location with respect to another graphics element in the output.

In total, there are 12 possibilities with the areas and the Coordinate systems.

(3 drawing areas) x (each with 2 Unit types) x (placement as Relative or Absolute) = 12.

Let’s start applying the Annotate principles we have covered so far. Below are four simple yet powerful methods to customize the Axis.

METHOD THREE
In this example we first create an Annotate dataset with the variables needed and then use this dataset in the PLOT statement in the PROC GPLOT. Please note that we will use the OFFSET= (0) option in this first method on both the AXIS statements. The reason we use this option is to simplify the concepts of Annotate Facility. Before using any built in Annotate Macros, the program should run %annomac. This invokes the macro call library and enables the program to access the Annotate macros. The simplest and most intuitive of the Coordinate systems is ‘2’. This refers to the exact values on both the axes defined by the ORDER option. The %move macro moves the pointer to the (x, y) position. %draw macro draws the tick mark at the point required. To draw a tick mark we use ‘B’ coordinate system because the line has to be outside the ‘Data’ area as shown in Figure A. The x and y coordinates in the %draw are same across observations since they are drawn in reference to the last observation which is given by %move.

```plaintext
%annomac;
DATA ann;
  xsys='2'; ysys='2'; %move(1,210);
  xsys='B'; ysys='B'; %draw(0,-0.5,black,1,1);
  xsys='2'; ysys='2'; %move(2,210);
  xsys='B'; ysys='B'; %draw(0,-0.5,black,1,1);
  xsys='2'; ysys='2'; %move(5,210);
  xsys='B'; ysys='B'; %draw(0,-0.5,black,1,1);
  xsys='2'; ysys='2'; %move(7,210);
  xsys='B'; ysys='B'; %draw(0,-0.5,black,1,1);
```
%macro custom(point);
xsys='2'; ysys='2'; %move(&point.,210);
xsys='B'; ysys='B'; %draw(0,-0.5,black,1,1);
%mend custom;
DATA ann;
%custom(1); %custom(2); %custom(5);
%custom(7); %custom(10); %custom(12);
RUN;

Method Three - with OFFSET set to Zero
METHOD FOUR
In the above Figure, the program utilized OFFSET= (0). By default PROC GPLOT adds some space to offset the first and last major tick marks from the origin and the end of axis. So, the code given in method three will not work if OFFSET= (0) is not explicitly stated. In the example shown below, the code has been modified to allow the OFFSET option to not be defined. The change is highlighted in bold.

```plaintext
%annomac;
%macro custom(point);
   xsys='5'; ysys='5'; %move(&point.,8.7);
   xsys='B'; ysys='B'; %draw(0,-0.5,black,1,1);
%mend custom;
DATA ann;
   %custom(7.8); %custom(15.8); %custom(40.6);
   %custom(57.1); %custom(81.8); %custom(98.3);
RUN;
gooptions ftext='Arial' htext=3 gunit=pct;
symbol v=dot I=join h=2;
axis1 order=(210 to 320 by 10) label=(angle=90 'Total Cholesterol (mg/dL)')
   minor=(n=1);
axis2 order=(1 to 12 by 1) minor=none major=none label=('Weeks')
   value=(tick=3 '' tick=4 '' tick=6 '' tick=8 '' tick=9 '' tick=11 '');
title height=3 'Method Four – with Default OFFSET';
PROC GPLOT data=trial;
   plot tc*week/vaxis=axis1 haxis=axis2 annotate=ann;
RUN;
quit;
```

Method Four - with Default OFFSET

![Graph showing total cholesterol levels over weeks with custom annotations.](image-url)
METHOD FIVE:
In situations where the OFFSET is specifically defined with a certain value like OFFSET=(2, 2) then the program has to change a little bit to accommodate the change in the axis. The change is highlighted in bold.

```sas
%annomac;
%macro custom(point);
    xsys='5'; ysys='5'; %move(&point, 8.65);
    xsys='B'; ysys='B'; %draw(0,-0.5,black,1,1);
%mend custom;
DATA ann;
    %custom(9.35); %custom(17.3); %custom(41.4);
    %custom(57.6); %custom(81.7); %custom(98);
RUN;
goptions ftext='Arial' htext=3 gunit=pct;
symbol v=dot I=join h=2;
axis1 order=(210 to 320 by 10) label=(angle=90 'Total Cholesterol (mg/dL)')
    minor=(n=1) offset=(2,2);
axis2 order=(1 to 12 by 1) minor=none major=none label=('Weeks') offset=(2,2)
    value=(tick=3 '' tick=4 '' tick=6 '' tick=8 '' tick=9 '' tick=11 '');
title height=3 'Method Five – with OFFSET set to a Value';
PROC GPLOT data=trial;
    plot tc*week/vaxis=axis1 haxis=axis2 annotate=ann;
RUN; QUIT;
```

Method Five - with OFFSET set to a Value
METHOD SIX:
Methods 3, 4 and 5 showed how to put tick marks at the required points on the axis using Annotate Facility and removing the unnecessary labels using the ‘Value’ option in the ‘AXIS’ statement. Going a little further, the Annotate Facility can also label the points on the axis without the use of the ‘Value’ option in the ‘AXIS’ statement. The program to label the axis programmatically is given below.

A new macro ‘%LABEL’ and a new programming function ‘%CNTL2TXT’ are used. The ‘CNTL2TXT’ is a programming function to manipulate system variables to copy (XLAST, YLAST) coordinates to (XLSST, YLSTT) coordinates. To better understand the ‘CNTL2TXT’ function lets get an understanding of the Internal Coordinates.

The Annotate Facility maintains 2 pairs of internal coordinates that are stored in internal variables.
1. Coordinates of the last graphics element drawn or the coordinates from the last move are stored in the variables XLAST and YLAST.
2. Coordinates of the last text drawn are stored in the variables XLSST and YLSTT.

Many functions use these internal coordinates as a starting point, relying on the coordinates that are specified with the function as an ending point. You cannot explicitly assign a value to XLAST, YLAST, XLSST and YLSTT because they are internal variables. However, you can use any of several functions to directly manipulate the values of the internal coordinates. One such function is ‘%CNTL2TXT’.

The parameters of the ‘%LABEL’ macro is given on page 3 of this paper and is used to draw text.

```
%annomac;
%macro custom(point);
  xsys='2'; ysys='2'; %move(&point.,210);
  xsys='B'; ysys='B'; %draw(0,-0.5,black,1,1);
  %cntl2txt; %label(0,0,"&point.",black,0,0,1.5,Simplex,E);
%mend custom;

DATA ann;
  Length text $10;
  %custom(1); %custom(2); %custom(5);
  %custom(7); %custom(10); %custom(12);
RUN;
goptions ftext='Simplex' htext=2 gunit=pct;
symbol v=dot I=join h=2;
axis1 order=(210 to 320 by 10) label=(angle=90 'Total Cholesterol (mg/dL)')
  minor=(n=1) offset=(0);
axis2 order=(1 to 12 by 1) minor=none major=none offset=(0) label=('Weeks')
  value=none;
title height=3 'Method Six';
PROC GPlot data=trial;
  plot tc*week/vaxis=axis1 haxis=axis2 annotate=ann;
RUN; QUIT;
```
The `VALUE=none` option in the above program makes the label of the x-axes 'Weeks' to move up so that it is too close to the points on the axes. To deal with the above problem a little fix is needed as shown below in the 'VALUE' option in the 'AXIS2' statement. The change is highlighted in bold.

```
goptions ftext='Simplex' htext=2 gunit=pct;
symbol v=dot I=join h=2;
axis1 order=(210 to 320 by 10) label=(angle=90 'Total Cholesterol (mg/dL)')
   minor=(n=1) offset=(0);
axis2 order=(1 to 12 by 1) minor=none major=none offset=(0) label=('Weeks')
   value=(tick=1 '' tick=2 '' tick=3 '' tick=4 '' tick=5 '' tick=6 ''
   tick=7 ''
   tick=8 '' tick=9 '' tick=10 '' tick=11 '' tick=12 '');
title height=3 'Method Six – with Value Option';
PROC GPLOT data=trial;
   plot tc*week/vaxis=axis1 haxis=axis2 annotate=ann;
RUN; QUIT;
```
CONCLUSION
This paper gives a thorough introduction on how to customize the axes in SAS/GRAPH output as per specific formatting requirements using the graphic options available and by using Annotate Facility. The Annotate Facility is a very powerful method to customize your graphs and I hope this paper will offer SAS users greater freedom to generate complex graphs.

REFERENCES


ACKNOWLEDGMENTS
I thank God for helping me write this paper.

There are two papers that I extensively referred to, and are very useful to get a clear understanding of the Annotate Facility. They are “Data Driven Annotations: An Introduction to SAS/GRAPH Annotate Facility” by Arthur L Carpenter and “Improving Your Graphics using SAS/GRAPH Annotate Facility” by David Mink and David Pasta.

I would like to thank Brian Fairfield-Carter and Tony Pisegna for their valuable insight and feedback. I express my gratitude to Jackie Lane, William Turner and Monte Jarvis for their encouragement and support in publishing this paper.

CONTACT INFORMATION
Your comments and questions are valued and encouraged. Contact the author at:

Abraham T Pulavarti
ICON Clinical Research
212 Church Road
North Wales PA 19454
Tel: 215.616.3020 / 646.309.0582
Email: pulavartia@iconus.com
abrahamtp@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.