PharmaSUG 2011 – Paper CC04

Plotting an Error/Line Plot and Bar Graph in a Single Plot with Dual Y-Axis Scales

Sanjiv Ramalingam, Octagon Research Solutions Inc.

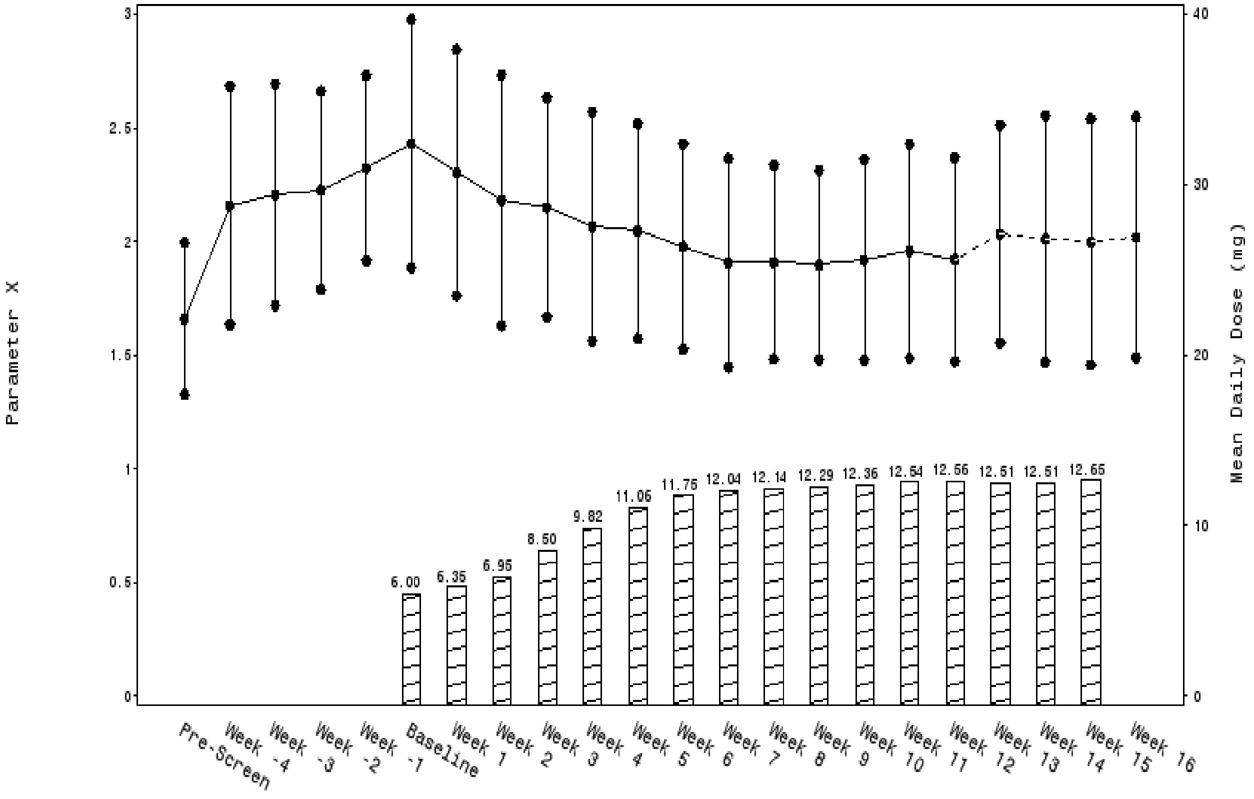
ABSTRACT

In analyzing clinical trial data, it is sometimes ideal to visualize the changes in the efficacy parameter with respect to change in dosing over time. This facilitates to observe the relationship between the two parameters and to also provide dosing recommendations, i.e. dose selection and choice of dose interval. A bar graph representing the dose and an error bar to represent the change in the efficacy parameter would clearly illustrate the relationship between the two parameters over time. The paper combines PROC GPLOT and the ANNOTATE feature to implement the above graph. A very simple and robust method is described to draw the bars and the pattern within every bar in a few steps.

INTRODUCTION

A bar graph to illustrate the distribution of dose over time and an error bar plot to represent the change in parameter of interest is plotted on the same graph. The graph below illustrates the change in mean values of the laboratory parameter X with the dosing regimen. The graph illustrates how effective the drug (dosing) was as evident in the response of the lab parameter of interest.

One feature of the graph includes two different Y-scales ; the scale on the left axis represents the mean change in value of the parameter of interest as an error bar plot and the scale on the right axis represents the dosing regimen over time. The horizontal axis is common to the two plots. The values for each value of the bar are displayed on top of the bar and stripped lines are created to shade each bar. The technique discussed can be used to create other patterns.



METHODOLOGY

There are two parts to creating the desired graph. The first part involves creating the error bar plot and the space required to draw the bar graph. The second part involves drawing the bar graph using the annotate feature.

PART I

If it was required to create two error bar plots one for each parameter or variable of interest it is relatively easy and is achieved using two PLOT statements in a PROC GPLOT.

As the first step, a previously used methodology will be used to create two error bar plots for the two different parameters of interest. This methodology has been discussed extensively by the author previously[1].

In brief the data set to be used for graph creation with the name "SRC1" will have the structure below at every visit.

VISIT	_NAME_	PARAM1	PARAM2
1	mean	XX	XX
1	min	XX	XX
1	max	XX	XX
2	mean	XX	XX
….			

Two PLOT statements within the PROC GPLOT are used to plot the error bars for each of the parameter. Lets assume that the mean values of PARAM2 will be presented as a histogram (bar chart). In this scenario a PLOT2 statement will be used to plot the second error plot, the only difference being that the color used to plot the lines will be made white thus making the plot invisible. We shall than use this space to create the bar chart. Using the PLOT2 statement does serve a purpose. It creates the second vertical axis and the framework required so that the annotate feature can be used to create the bar chart.

Note that the color used in the SYMBOL2 statement is white. The ORDER option in defining the axis for the second parameter is used to determine the area that will be used to draw the bars.

```
Proc gplot data=src1;
      symbol1  v=dot c=black I=hiloctj mode=include;
      symbol2 v=star height=1 i=join  C = white mode=include;
      axis2 label=(angle=90 h=1.2 'Param X') order=(0 to 3 by 0.5)
      minor=none;
      axis1   label=none   order=(0   to   23   by   1)   minor=none   major=none
      color=black value=none origin=(12 pct,22 pct);
      axis3 label=(angle=90 h=1.2 'Param Y')order=(0 to 40 by 10) minor=none;
      plot param1*visitnum/haxis=axis1 vaxis=axis2   ;
      plot2 param2*visitnum/anno=annof vaxis=axis3  ;
run;
quit;
```

A graph with an error bar plot minus the bar chart will be obtained.

PART II

DRAWING THE BARS AT EACH VISIT

The data set, SRC1 will be used as the source data set to create the annotate data set. Since only the mean values are required, the SRC1 data set is subset to only hold the mean values and will be referred to as the SRC2 data set. The variables required are the visit number (will be renamed to Xn and represents the time points on the x-axis) ; the mean value (param2) and will be renamed to Yn.

Only a combination of MOVE and DRAW statements is required to draw the bar at each visit. Since only the data area will be used XSYS and YSYS will be set to 2.

To draw the bar at every visit, the following pattern will be used. Each of the steps described below will be output.

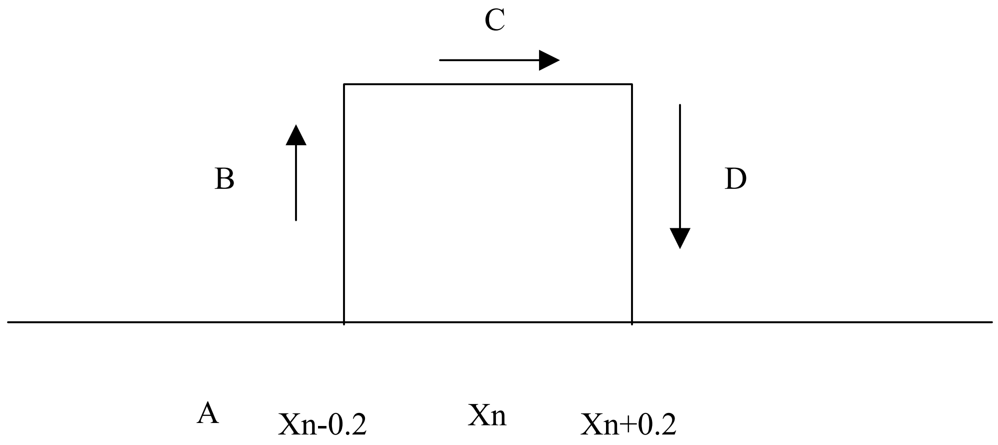
A. Assuming that each bar will be of 0.4 units width (with an x-axis that is 1 unit spaced apart) with Xn as the center of every time point, the 'pointer' is moved to 0.2 units before the time point (Xn-0.2) using the MOVE statement. The Y value would be –0.4 to accommodate for the shift in origin. Ideally it would be 0.

B. From the horizontal value, X=Xn-2, a vertical line will be drawn to the required height (PARAM2), Y=Yn, using the DRAW statement.

C. A horizontal line will now be drawn from X=Xn-2 and y=Yn to X=Xn+0.2 and Y=Yn using the DRAW statement.

D. The bar is completed by drawing line back to the X-axis, X=Xn+0.2 and Yn=-0.4 using the DRAW statement.

Flow of statements when drawing each bar is illustrated below.



The code required to draw each bar is shown below.

```
data anno2;
       length function $8;
       set src2;
       retain  line 1 xsys ysys '2' line 1 color 'black' when 'a';
       by xn;
       if first.xn then do;
             x=xn-0.2;
             y=-0.4;
             function='move';
             output;
             x=xn-0.2;
             y=yn;
             function='draw';
             output;
           x=xn+0.2;
             y=yn;
             function='draw';
             output;
             x=xn+0.2;
             y=-0.4;
             function='draw';
             output;
         end;
run;
```

The data set for a single time point (e.g. VISIT=6) will be of the structure shown below.

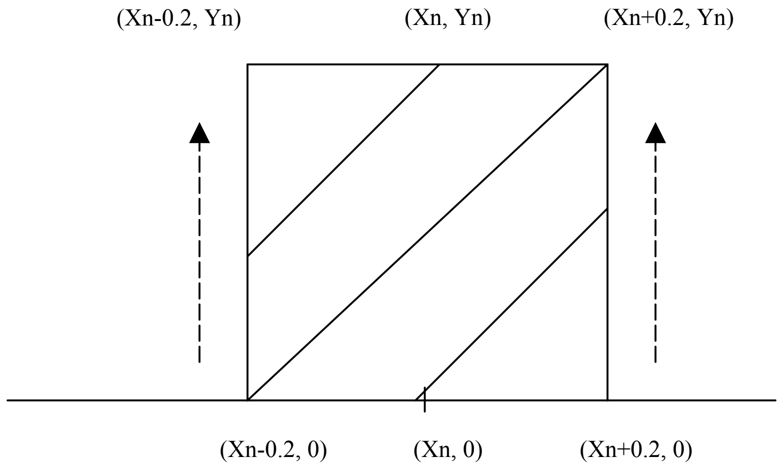
Xn	Yn	FUNCTION	X	Y	XSYS	YSYS	LINE	COLOR	WHEN
6	6	move	5.8	-0.4	2	2	1	black	a
6	6	draw	5.8	6.0	2	2	1	black	a
6	6	draw	6.2	6.0	2	2	1	black	a
6	6	draw	6.2	-0.4	2	2	1	black	a

DRAWING THE PATTERN WITHIN EACH BAR

It is very common to use hatched or cross hatched patterns within bar. A method to draw a single pattern is described below. There are three steps to the whole process.

Slant lines will be drawn between Xn and Yn at regularly spaced intervals. For example, the first line will be drawn from (Xn-0.2,0) to (Xn+0.2,Yn). The next line will be drawn from (Xn-0.2, 0.2) to (Xn, Yn). The take home is that there is a pattern to the way the origin (starting co-ordinates for each line) can be chosen in increments of a constant K as indicated by the dashed arrow in the left side of the figure. Similarly, the end co-ordinates of each line can be chosen in increments of a constant K+B.

In the first step the starting co-ordinates are generated using the FOR loop. Since this is the starting co-ordinate the function used will be, ‘MOVE’ for all statements.



```
data anno2a;
       set src2;
       retain line 1 xsys ysys ‘2’ line 1 color ‘black’ when ‘a’;
       by xn;
       n1=xn-0.2;
       If first.xn then do;
              i=1;
              do i=0 to yn by 0.05;
                     x=n1;
                     y=i;
                     function=’move’;
                     i+1;
                     output;
              end;
       end;
run;
```

The variable , I is used as an index variable to, ‘set’, with the next annotate data set created for all the ending co-ordinates. In the next step, since all the records created will represent the end co-ordinates, the function, DRAW will be used for all the statements.

```
data anno2b;
       set src2;
       retain line 1 xsys ysys ‘2’ line 1 color ‘black’ when ‘a’;
       by xn;
       n1=xn+0.2;
       If first.xn then do;
              i=1;
              do i=0 to yn by 0.05;
                     x=n1;
                     y=i+0.2;
                     function=’draw’;
                     i+1;
                     output;
              end;
       end;
run;
```

In the third and final step both the annotate data sets ANNO2A and ANNO2B are set together by Xn and the index, I, so that the MOVE and DRAW statements occur in the right order. The annotate data sets can then be set together to form the ANNOF data set that is used in PROC GPLOT.

To draw a hatched pattern the same methodology can be applied to draw the other set of slant lines by reversing the origin and end point.

CONCLUSION

A technique to plot an error bar plot and a bar graph on the same graph has been shown. The simple but very powerful statements of the annotate feature have been used to draw the bars and the shading pattern within each bar. Understanding of this simple and powerful feature of SAS® will empower the user to manipulate and customize graphs that may not be otherwise possible to create using a single graphical procedure.

REFERENCE

1. Ramalingam, Sanjiv. A Generalized Procedure to Create SAS/Graph Error Bar Plots. PHARMASUG 2009, Portland, OR.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Please contact the author at:

Sanjiv Ramalingam
Octagon Research Solutions Inc.
585 E Swedesforde Road
Wayne, PA 19087
Email : sramalingam@octagonresearch.com