PharmaSUG 2012 – Paper CC13

Techniques for Improvising the Standard Error Bar Graph and Axis Values Completely Through SAS Annotation

Sunil Kumar Ganeshna, PharmaNet/i3, Pune, India
Venkateswara Rao, PharmaNet/i3, Pune, India

ABSTRACT

This paper introduces the multiple challenges and several useful techniques for enhancing SAS/GRAPH® output when creating plots with standard error bars and symbols of the least square means/mean and enhancing multiple x-axis values with uneven intervals. The macro presented in this paper utilizes the techniques of data manipulation, SAS annotation, markers and fonts, plot statement, axis statement, symbol and legend statements; this single macro call generates an annotated dataset that is utilized to create an Error bar graph with multiple axis values.

INTRODUCTION

There are several methods for drawing standard error bars onto a mean plot. The most common method is to use the options INTERPOL=STDJT or INTERPOL=HILOJT in the SYMBOL statement, and each has its own drawbacks when used for displaying the mean data values with symbols. If the VALUE and HEIGHT options are applied in the SYMBOL statement, you are unable to annotate the mean values with symbols, but all of the individual data values are plotted along the standard error bars. In order to overcome the drawbacks with the above-mentioned options, we created mean plots displaying standard error bars and symbols for the means by annotation. For displaying the multiple x-axis values, we can use the TICK option in the AXIS statement, but still this may cause a problem for uneven intervals.

Challenges involved in this task:

1) Creating error bars with special symbols like inverted triangles, or triangles that are fully filled with color.
2) Jittering the axis values to avoid overlapping of the treatments.
3) Creating two different line types for a single treatment between the visits, showing the differentiation of the visits or switching of the treatments.
4) Creating multiple x-axis values with uneven intervals with their corresponding text.
5) Avoiding unwanted tick marks and placing them only at the respective positions.
6) Creating a y-axis range with uniform intervals for multiple outputs.

TECHNIQUES

A macro was created to resolve all of the above challenges.

1) This macro creates two annotated datasets. One annotated dataset is used to create all of the error bars for each treatment with respect to each visit using different symbol fonts.
2) This data set is manipulated to create two different line types for a specified treatment, varying between the visits to differentiate visits of a particular treatment. (Check the output for black and blue colored lines how the line types are differing for each of these treatments).
3) Special symbols were used rather than a simple hyphen to differentiate the upper and lower limits.(Check for Black and Red color symbols in the output in Appendix 1
4) The second annotated data set is created to capture the x-axis interval values at multiple levels i.e., each row represents different levels, with a text description at the beginning of the row. For example, the first row describes the month, the second row displays its corresponding visit, and the third row displays the week (see the graph x-axis values in Appendix 1).
5) One more macro is involved to calculate the y-axis range with uniform intervals. This macro takes care of the uniform distribution of the y-axis values. and is useful if we are creating multiple outputs from a single program, rather than calculating the minimum and maximum value for each plot every time.

CONCLUSION

SAS ANNOTATION is a powerful and productive tool in SAS/GRAPH and should be fully examined and exploited. In this paper, we have shown how to use the annotation facility to create a macro that will resolve all the above challenges and create the error bar graph.

The advantages of this macro are

1) It addresses all the above challenges while making the code intuitive.
2) A single macro call can generate multiple graphs that take care of all of the necessary requirements without much effort.
3) Portions of the macro can also be used as pieces of code, if required. For example, annotation is only needed for defining the axis intervals- it's not required for the error bars.
4) This macro can be used as a reference for other graphs, and also for creating annotation datasets.

A complete description of the macro is found in Appendix 2, and each parameter is defined in Appendix 2, using the graph as an example

REFERENCES

SAS/GRAPH Software: version 9.1.3, SAS® Institute Inc., Cary NC

SAS/Graph help online documentation: Version 9, SAS® Institute Inc,, Cary NC.
"Tired of Defining Axis Scale for SAS Graphs? A Solution with an Automatic Optimizing Approach".
Don (Dongguang) Li, NCIC-CTG at Queen's University, Kingston, ON, Canada. PharmaSUG 2003 conference proceedings.

ACKNOWLEDGMENTS

Thanks to Vijay Moolaveesala and Nancy Brucken for their suggestions on this paper.

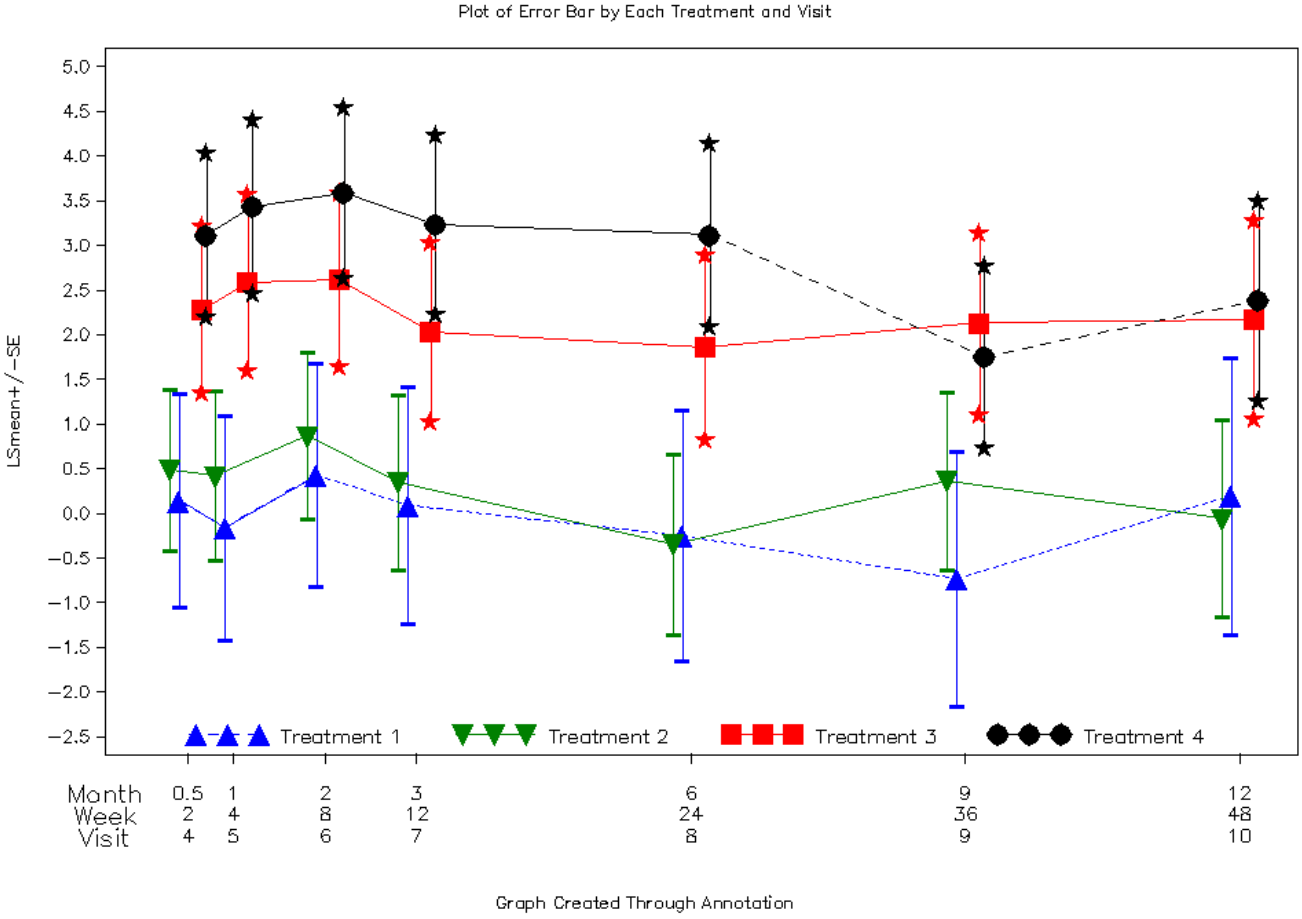
CONTACT INFORMATION

Please send your comments, questions, and inquiries to:

Name: Sunil Kumar Ganeshna,
Address: PharmaNet/i3, Commerzone,
6th Floor, Building No 4,
Yerwada, Pune, India.
E-mail: skumar2@pharmanet-i3.com
sunilganeshna@gmail.com

Name: Venkateswara Rao Ch.,M.Sc (Statistics)
Address: PharmaNet/i3, Commerzone,
6th Floor, Building No 4,
Yerwada, Pune, India.
E-mail: verao@pharmanet-i3.com
venkibhu@aol.com

APPENDIX 1


Plot of Error Bar by Each Treatment and Visit
LSmean+/-SE
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
Treatment 1
Treatment 2
Treatment 3
Treatment 4
Month 0.5 1 2 3 6 9 12
Week 2 4 8 12 24 36 48
Visit 4 5 6 7 8 9 10
Graph Created Through Annotation

APPENDIX 2

MACRO PARAMETERS

Name	Type	Default	Description and Valid Values
Indata	Required		Name of the input dataset The input dataset must contain the following variables Grouping variable – for example, treatment. 1) X-axis – variable that contains the values plotted on the x- axis. 2) Estimate – variable that contains the point estimates (for example mean is the point estimate if the y-axis in the graph is mean+/-std, Lsmeans is the point estimate if the y-axis is Lsmeans+/-se based on the requirements). 3) Upper – upper limit of the estimates. 4) Lower – lower limit of the estimates. These 5 variables are mandatory for running this macro.
Group_var	Optional	Null	Name of the grouping variable in the input data set. This should be numeric.
X_axis	Required		Name of the x-axis variable in the input data set. This should be numeric.
Gap_bars	Required		To avoid the overlapping of bars of distinct levels of classification variable at a value of X. Specify the margins separated by ",". If the classification variable contains 4 levels, then the gap_bars should be "0.1, 0.2,0.3,0.4". The values 0.1, 0.2, 0.3 have to be adjusted based on the range of the x-axis. The number of values specified in "gap_bars" should be equal to the number of levels in the grouping variable.
Estimate	Required	Numeric Value	Name of the variable containing the values of point estimates for e.g. mean, lsmeans.
Upper	Required	Numeric Value	Name of the variable containing the values of upper limits of the estimates.
Lower	Required	Numeric Value	Name of the variable containing the values of lower limits of the estimates.
Col_sym	Required	black	Name of the colors to be displayed for the symbols in the graph.
Sym_est	Required		Specifies the plot symbols for the data points. For example, if there are 4 treatments, each treatment has to be represented by a separate symbol. If Treatment A is represented by an inverted triangle, the symbol D displays that inverted triangle in the specified font.
Font_sym_est	Required		Name of the font to be used for the symbols that we specified in the sym_est parameter. For example, the MARKER font was used to display inverted triangles for Treatment A.

			The following site lists many fonts and markers that can be used in the SYMBOL statement: http://support.sas.com/documentation/cdl/en/graphref/63022/HTML/default/viewer.htm#font-font-lists.htm
line_type	Required	1	Sometimes, the lines in the graph may be dashed lines with different spaces and sizes. This parameter is used to specify the type of line for every category in the group variable. Default is "1" (a straight line).
Sym_lim	Required		This parameter is used to define the symbols for both the upper and lower limits. For example if – (hyphen) is used, then the corresponding symbol for this is "J".
Font_sym_limits	Required		Name of the fonts to be used for the symbols that we specified in the sym_lim parameter. Since we have defined "J" as the respective symbol, the corresponding font for this is "MUSIC".
Gp_visit_line	Optional	Null	Required if we want to differentiate the line type for a specified group_var (ex: treatment) from a particular visit to the last visit. For ex: for treatment 2, we want a dashed line from visit 3 to the end, so this should be specified as 2 3 2 (treatment visit line type). If this needs to cover multiple treatments, then the delimiter is a comma .If we want to differentiate the line type for two treatments then they should be separated by comma like 2 3 2, 3 5 20 (for treatment 3 the line type should change from visit 5 to end.)
Rows_xaxis	Required		Sometimes the x-axis value labels require multiple lines. For example, the x-axis value labels may contain both visit and month. In this case, the value of rows_xaxis is 3.
Labels_rows_xaxis	Required		Text to be printed for the x-axis values in multiple rows. See the graph for x-axis values and their corresponding left side text.
Formats	Required		Format to be applied to the x-axis values.
X_axis_label	Optional		Label for the x-axis.
Zero_yaxis	Required	no	Sometimes a reference line corresponding to the "0" on y-axis is needed for graphs of change from baseline. The valid values for this parameter are "yes" and "no".
Pos_x_values	Required		The x-axis values are displayed with the annotation facility. If the placement of the legend changes, then the position values of the x-axis will change automatically. The user has to determine the values, and then manually check to get the proper alignment of x-axis values.

Usage Notes:

Typical Macro Call(s) and Description:

This macro will also create the following internal macro variables:

max_y --- Maximum value on y-Axis
min --- Minimum value on y-Axis
interval --- Difference between major tick marks on the y-Axis.

max_x --- maximum value on x-Axis
min_x --- minimum value on x-Axis

Use these parameters in the corresponding axis statements.

```
%macro Error_Graph(indata=,
                   group_var=,
                   x_axis=,
                   gap_bars=,
                   estimate=,
                   upper=,
                   lower=,
                   col_sym=,
                   sym_est=,
                   font_sym_est=,
                    line_type=,
                    gp_visit_line=,
                    sym_lim=,
                    font_sym_lim=,
                    rows_xaxis=,
                    labels_rows_xaxis=,
                    formats=,
                    xaxis_label=,
                    zero_yaxis=,
                    pos_x_values=);

      %if %sysfunc(exist(anno)) %then %do;
            /*Dropping the ANNO dataset to avoid the problems on appending the
            datasets again and again*/
            PROC SQL;
                  DROP TABLE anno;
            QUIT;
      %end;

      %let height_symbol=2.5;%let height_line=0.3;

      %if &group_var ne %then %do;

            /* Creating a dataset with distinct categories of Group Variable */
            PROC SORT DATA=&indata(KEEP=&group_var) OUT=group nodupkey;
                  BY &group_var;
            RUN;

            /*Assigning the distinct categories in Group variable to macro variable to
            create one Annotation dataset for each category and also Number of Distinct
            Categories*/
            DATA group;
                  SET group;
                  CALL SYMPUT (CATT("group",_n_),PUT(&group_var,best.));
                  CALL SYMPUT("no_groups",PUT(_n_,best.));
            Run;

      %end;

      %else %do;
```

```
        /*Fixing the values of the macro variables "no_groups" and "group1" when the
        Grouping variable is missing or null*/
        %let no_groups=1;
        %let group1=1;

        DATA &indata;
               Set &indata;
               group_var=1;
        RUN;

%end;

/*Creating annotation dataset for every single category in the grouping variable*/
%Do i = 1 %to &no_groups;

        DATA _null_;
               /*Assigning the gap between bars, color of symbol, symbol and font for
               a symbol, Symbl and font for both the limits*/
               CALL SYMPUT("gap",SCAN("&gap_bars",&i,","));
               CALL SYMPUT("color", SCAN("&col_sym",&i,","));
               call symput("sym_estimate", SCAN("&sym_est",&i,","));
               CALL SYMPUT("font_sym_estimate", SCAN("&font_sym_est",&i,","));
               CALL SYMPUT("sym_limits", SCAN("&sym_lim",&i,","));
               CALL SYMPUT("font_sym_limits", SCAN("&font_sym_lim",&i,","));
               CALL SYMPUT("type", SCAN("&line_type",&i,","));
        RUN;

        %if &color= %then %let color=black;
        %if &type= %then %let type=1;
        /*Creating a symbol statement for every category in the grouping variable*/
        %if &group_var ne %then %do;
               SYMBOL&i VALUE=&sym_estimate FONT=&font_sym_estimate LINE=&type
               CV=&color INTERPOL=join HEIGHT=0.01;
        %end;
        %else %do;
               SYMBOL&i VALUE=&sym_estimate FONT=&font_sym_estimate CV=&color
               HEIGHT=0.01;
        %end;

        /*Creating macro variables for the required variables */

        DATA _null_;
               SET &indata;
               CALL SYMPUT(cats("original_x",_n_),put(&x_axis,best.));

               CALL SYMPUT(CATS("x",_n_),PUT(&x_axis+&gap,best.));
               CALL SYMPUT(CATS("est",_n_), PUT(&estimate,best.));
               CALL SYMPUT(CATS("upper_limit",_n_), PUT(&upper,best.));
               CALL SYMPUT(CATS("lower_limit",_n_), PUT(&lower,best.));
               CALL SYMPUT("no_visits", PUT(_n_,best.));
               WHERE &group_var=&&group&i;
        Run;

                /*Creating annotated dataset for drawing the error bars with special
        symbols and for connecting each error bar*/

        DATA anno_%sysfunc(left(&&group&i));
               FORMAT x 10.5;
               LENGTH function $20. style $20. text $20. color $10.;
               RETAIN xsys ysys "2" hsys "3" color "&color";;
               %if &group_var ne %then %do;
                      group_var=&&group&i;
               %end;
```

```
        %else %do;
                Group_var=1;
        %end;
        %do j = 1 %to &no_visits;
                %if &j < &no_visits %then %do;

data="estimate_line";original_x=&&original_x&j;x=&&x&j;y=&&est&j;function="MO
VE";size=&height_line;line=&type; OUTPUT;
                        %let k=%eval(&j+1);

data="estimate_line";original_x=&&original_x&k;x=&&x&k;y=&&est&k;function="DR
AW"; OUTPUT;size=&height_line;line=&type; OUTPUT;

data="estimate";original_x=&&original_x&j;x=&&x&j;y=&&est&j;function="SYMBOL"
;text="&sym_estimate";size=&height_symbol;style="&font_sym_estimate";when="A"
; OUTPUT;

data="limits_join";original_x=&&original_x&j;x=&&x&j;y=&&lower_limit&j;
function="MOVE";size=&height_line;line=1; OUTPUT;

data="limits_join";original_x=&&original_x&j;x=&&x&j;y=&&upper_limit&j;
function="DRAW";size=&height_line;line=1; OUTPUT;

data="lower";original_x=&&original_x&j;x=&&x&j;y=&&lower_limit&j;function="SY
MBOL";size=&height_symbol;text="&sym_limits";style="&font_sym_limits";when="A
"; OUTPUT;

data="upper";original_x=&&original_x&j;x=&&x&j;y=&&upper_limit&j;function="SY
MBOL";size=&height_symbol;text="&sym_limits";style="&font_sym_limits";when="A
"; OUTPUT;
                %end;
                %else %do;

data="limits_join";original_x=&&original_x&j;x=&&x&j;y=&&lower_limit&j;
function="MOVE"; size=&height_line;line=1; OUTPUT;

data="limits_join";original_x=&&original_x&j;x=&&x&j;y=&&upper_limit&j;
function="DRAW"; size=&height_line;line=1; OUTPUT;


data="estimate";original_x=&&original_x&j;x=&&x&j;y=&&est&j;function="SYMBOL"
;text="&sym_estimate";style="&font_sym_estimate";when="A";
size=&height_symbol; OUTPUT;

data="lower";original_x=&&original_x&j;x=&&x&j;y=&&lower_limit&j;function="SY
MBOL";text="&sym_limits";style="&font_sym_limits";when="A";
size=&height_symbol; OUTPUT;

data="upper";original_x=&&original_x&j;x=&&x&j;y=&&upper_limit&j;function="SY
MBOL";  text="&sym_limits";style="&font_sym_limits";when="A";
size=&height_symbol; OUTPUT;
                %end;
        %end;
run;

%if  %sysfunc(exist(Anno)) %then %do;
/*Appending the annotation datasets of every category in group variable to
the dataset ANNO*/
        PROC APPEND DATA=anno_%sysfunc(LEFT(&&group&i)) BASE=Anno;
        RUN;
%end;

%else %do;
```

```
                DATA ANNO;
                        SET anno_%sysfunc(LEFT(&&group&i));
                RUN;
        %end;
%end;

        PROC SORT DATA=&indata(KEEP=&x_axis) OUT=xaxis(RENAME=(&x_axis=x)) NODUPKEY;
         BY &x_axis;
        RUN;

        DATA anno_axis1;
         LENGTH text $100. data $20.;
         RETAIN hsys "1" size 2.5;
         SET xaxis(KEEP=x in=a) xaxis(KEEP=x in=b) end=eof;
         if a then do;

        data="tick";xsys="2";ysys="1";text="I";y=0;position="5";style="simplex";
        OUTPUT;
        end;
        else if b then do;
                %do k=1 %to &rows_xaxis;

        format=CATT(SCAN("&formats",&k,","),".");data="x_value";size=3;xsys="2";ysys=
        "5";text=left(putn(x,format)); y=&pos_x_values-
        2.5*&k;position="5";style="simplex"; OUTPUT;
                %end;
        end;
        if eof then do;
                %do k=1 %to &rows_xaxis;
        data="x_value_label";size=3;xsys="1";ysys="5";text=SCAN("&labels_rows_xaxis",
        &k,",") ;x=0;y=&pos_x_values-2*&k;position="5";style="simplex"; OUTPUT;
                %end;

        data="Label";size=3;xsys="2";ysys="5";text="&xaxis_label";x=50;xsys="1";y=&po
        s_x_values-2.5*(&k+1);position="5";style="simplex"; OUTPUT;
        end;
 RUN;

DATA anno;
        SET anno anno_axis1;
RUN;

%global max_y;

/*Finding the maximum value for the Y-axis to decide the number of intervals to be
shown in the Y-axis*/
PROC SQL NOPRINT;
        SELECT MIN(y),MAX(y),MIN(group_var) INTO :min_y,:max_y,:useless FROM anno
        WHERE group_var >=0;
QUIT;
%put &min_y;

%MACRO scale (max);
        %if &max=0 %then %do;
                %let max=5;
        %end;
        DATA _null;
                unit=&max/10;
                grade=FLOOR(LOG10(unit));
                sunit=unit/(10**grade);
                if sunit<SQRT(2)
                then interval=10**grade*1;
                else if sunit<SQRT(10)
```

```
            then interval=10**grade*2;
            else if sunit<SQRT(50)
            then interval=10**grade*5;
            else interval=10**grade*10;
            maxscale=CEIL(&max/interval)*interval;
            %global maxscale interval;
            CALL SYMPUT ("maxscale",LEFT(PUT(maxscale,best.)));
            CALL SYMPUT ("interval",LEFT(PUT(interval,best.)));
      run;
%MEND scale;
%scale(&max_y);

%put &maxscale;
%put &interval;

DATA e1;
      do l = -&maxscale to &maxscale by &interval;
            l1=lag(l);
            y=&min_y;
            OUTPUT;
      end;
RUN;

DATA _null_;
      SET e1;
      If "&zero_yaxis"="yes" then do;
            if y <0 then do;
                  if l1<y;
                  CALL SYMPUTX("min",l1,"g");
            end;
            else do;
                  if l1=0;
                  CALL SYMPUTX("min",l1,"g");
            end;
      end;
      else do;
            if l1<y;
            CALL SYMPUTX("min",l1,"g");
      end;
Run;

PROC SORT DATA=anno OUT=plot_data;
      BY group_var x;
      WHERE data eq "estimate" and group_var ne .;
RUN;

DATA _NULL_;
      CALL SYMPUT("commas",COUNT("&gp_visit_line",",")+1);
RUN;

DATA plot_data;
      SET plot_data;
      %if &gp_visit_line ne %then %do;
            %do l= 1 %to &commas;
                  %let word=%sysfunc(SCAN(&gp_visit_line,&l,","));
                  %let one=%sysfunc(SCAN(&word,1));
                  %let two=%sysfunc(SCAN(&word,2));
                  if group_var = &one and original_x>&two then DELETE;
            %end;
      %end;
RUN;
```

```
        DATA anno;
                SET anno;
                %if &gp_visit_line ne %then %do;
                        %do l= 1 %to &commas;
                                %let word=%sysfunc(SCAN(&gp_visit_line,&l,","));
                                %let one=%sysfunc(SCAN(&word,1));
                                %let line=%sysfunc(SCAN(&word,3));
                                %let two=%sysfunc(SCAN(&word,2));
                                if group_var = &one and original_x=&two and data="estimate_line"
                and funtion="MOVE" then line=&line;
                                else if group_var = &one and original_x>&two and
                data="estimate_line" then line=&line;
                        %end;
                %end;
        RUN;

        %if &group_var = %then %do;
                DATA anno;
                        SET anno;
                        WHERE data ne "estimate_line";
                RUN;
        %end;

        %global min_x max_x;

        PROC SQL NOPRINT;
                SELECT MIN(x),MAX(x),MAX(group_var) INTO :min_x,:max_x,:useless FROM anno
                WHERE group_var>=0;
        QUIT;

%mend Error_Graph;


Sample macro call and steps to pass the proc gplot .


%Error_Graph(indata=indata,
                group_var=trtsort,
                x_axis=plmosrnd,
                gap_bars=%str(-0.1,-0.2,0.15,0.2),
                estimate=estimate,
                upper=upper,
                lower=lower,
                col_sym=%str(blue,green,red,black),
                sym_est=%str(C,D,U,W),
                font_sym_est= %str(marker,marker,marker,marker),
                line_type=%str(1,1,1,1),
                gp_visit_line=%str(1 2 2,7 6 20),
                sym_lim=%str(J,J,V,V),
                font_sym_lim=%str(music,music,marker,marker),
                rows_xaxis=3,
                labels_rows_xaxis=%str(Month,Week,Visit),
                formats=%str(best,wek,visit),
                xaxis_label=,
                zero_yaxis=no,
                pos_x_values=14);

*Origin can be changed according to user's requirement;

   GOPTIONS RESET=all;
   GOPTIONS RESET=goptions DEVICE = png CBACK = white CTEXT = black noborder
   TARGET=png XMAX=10 in YMAX=7.5 in FTEXT = simplex FTITLE = simplex HTEXT = 2.1
   VSIZE=7in HSIZE=10 in Htitle=2 COLORS = (black) GSFLEN=80 GUNIT = pct
```

```
display GSFMODE = replace ;

SYMBOL1  COLOR =blue   HEIGHT=0.0001  FONT="marker" V=C WIDTH=0.5 I=j lINE=20;
SYMBOL2 COLOR = green  HEIGHT=0.0001  FONT="marker" v=D WIDTH=0.5 I=j  ;
SYMBOL3  COLOR =red    HEIGHT=0.0001  FONT="marker" v=U WIDTH=0.5  I=j ;
SYMBOL4 COLOR =black HEIGHT=0.0001 WIDTH=0.5 FONT="marker" V=W I=j ;


AXIS1 MAJOR=none VALUE=none MINOR=none   ORDER=(&min_x to &max_x BY %sysevalf(&max_x-
&min_x)) LABEL=(angle=90 j=c "               ") OFFSET=(5,3);
AXIS2 MINOR=none ORDER=(&min to %sysevalf(&max_y+&interval) BY &interval)
LABEL=(angle=90 rotate=360 "LSmean+/-SE") OFFSET=(2,2);

LEGEND1 MODE = protect  ACROSS = 4 NOFRAME  VALUE= (h=2.4 justify = left)
        POSITION = (bottom center inside) SHAPE = symbol(6,2.5) OFFSET =(0,0) LABEL =
none ;

TITLE1 "Plot of Error Bar by Each Treatment and Visit";
FOOTNOTE1 "Graph Created Through Annotation";

OPTIONS ORIENTATION =landscape PS=64 ;
ODS listing close;
ODS rtf FILE="&pharmasug/rptnm.rtf";
PROC GPLOT DATA=plot_data;
   PLOT y*x=group_var / HAXIS=axis1  ANNOTATE=anno VAXIS=axis2 LEGEND=legend1;
   FORMAT GROUP_VAR TRT. ;
   RUN;
QUIT;

ODS rtf close;
ODS listing;
```