A Generalized Procedure to Create SAS®/Graph Error Bar Plots
Sanjiv Ramalingam, Consultant, Octagon Research Solutions, Inc.

ABSTRACT
Different methodologies exist to create error bar related plots. Procedures exist to use the dataset by itself to create error bars either for standard deviation +/- mean (INTERPOLATION=STD1JT), standard errors +/- mean (INTERPOLATION=STD1TJM) or min and max values but it becomes cumbersome to generate figures by just using the PROC PLOT procedure as one has to go through numerous options available to find the right option. This paper discusses a methodology to plot the difference between the means with the standard error as an example.

INTRODUCTION
A typical scenario involving error bars is to obtain standard errors. If the error bars are obtained as the standard errors of not one treatment (as related to the pharmaceutical industry) but as the difference of two different treatments it is always advantageous to have a general process. In this process, the PROC MEANS/UNIVARIATE procedure can be used to calculate the statistics and a separate data step can be used to calculate the derived statistics such as the standard error for two different treatments. Then one generalized plot procedure can be used to create the error bar plot.

METHODOLOGY
CASE 1
The concept is based on the assumption that all values required to create the graph are already available in the form of a dataset. Let the first objective be to create a standard error plot for the difference between two different treatments by visit. The parameters required are the means for each of the treatments (the difference of the means is then computed). The standard error of the difference of the means is computed using the formula:

\[ \sigma_{\text{Diff}} = \sqrt{\left(\frac{\sigma_{\text{T1}}^2}{n_{\text{T1}}}\right) + \left(\frac{\sigma_{\text{T2}}^2}{n_{\text{T2}}}\right)} \]

where,
\[ \sigma_{\text{Diff}} = \text{standard error of the difference} \]
\[ \sigma_{\text{Trt1}}^2 = \text{variance of first treatment (square of the standard deviation)} \]
\[ \sigma_{\text{Trt2}}^2 = \text{variance of second treatment} \]

\[ n_{\text{Trt1}} = \text{number of subjects in the first treatment} \]
\[ n_{\text{Trt2}} = \text{number of subjects in the second treatment} \]

After the means for each of the treatments have been computed, the following can be computed using the sample code statements namely, the difference of the means (\( d_{\text{mean}} \)), the standard error (\( SE \)) of the difference of the means (\( d_{\text{se_m}} \)), the maximum (\( \text{maxm;mean+SE} \)) and minimum (\( \text{minm;mean-SE} \)).

\[
\begin{align*}
  d_{\text{mean}} &= \text{trt1}_\text{- trt2}_\text{mean}; \\
  d_{\text{se_m}} &= \sqrt{ ( (se_{\text{trt1}}^*se_{\text{trt1}})/n_{\text{trt1}} ) + ( (se_{\text{trt2}}^*se_{\text{trt2}})/n_{\text{trt2}} ) } ; \\
  \text{maxm} &= d_{\text{mean}} + d_{\text{se_m}}; \\
  \text{minm} &= d_{\text{mean}} - d_{\text{se_m}} ; \\
\end{align*}
\]

The output dataset created namely, DS1 would then have a structure like the one below.

<table>
<thead>
<tr>
<th>VISIT</th>
<th>D_MEAN</th>
<th>MAXM</th>
<th>MINM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Week 1</td>
<td>xxx</td>
<td>xxx</td>
<td>xxx</td>
</tr>
<tr>
<td>Week 2</td>
<td>xxx</td>
<td>xxx</td>
<td>xxx</td>
</tr>
<tr>
<td>Week 3</td>
<td>xxx</td>
<td>xxx</td>
<td>xxx</td>
</tr>
<tr>
<td>........</td>
<td>.......</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The statistical parameters are now available. Since the error bars basically are two points with a common reference point, the HILOCTJ option in SAS/Graph can be used. However to use the HILOCTJ option which basically draws a solid vertical line that connects the minimum and maximum Y values for each X value, the dataset should have individual records for each of the
maxm and minm variables. Hence a PROC TRANPOSE can be done on the variables maxm and minm in the dataset DS1:

```
proc transpose data=ds1 out=ds2;
by visit ;
var d_mean minm maxm;
run;
```

The resulting dataset will be of the structure:

<table>
<thead>
<tr>
<th>VISIT</th>
<th><em>NAME</em></th>
<th>COL1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Week 1</td>
<td>MINM</td>
<td>xxx</td>
</tr>
<tr>
<td>Week 1</td>
<td>D_MEAN</td>
<td>xxx</td>
</tr>
<tr>
<td>Week 1</td>
<td>MAXM</td>
<td>xxx</td>
</tr>
<tr>
<td>Week 2</td>
<td>MINM</td>
<td>xxx</td>
</tr>
<tr>
<td>Week 2</td>
<td>D_MEAN</td>
<td>xxx</td>
</tr>
<tr>
<td>Week 2</td>
<td>MAXM</td>
<td>xxx</td>
</tr>
<tr>
<td>.........</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

To automatically assign the range for the Y-axis and also the number of divisions within each unit the dataset can be sorted by the col1 variable. The first and last values would then be the minimum and the maximum values. The difference between the maximum and the minimum divided by 10 (assuming that 10 divisions are preferred within each unit) rounded to the nearest preferred number of decimal places will yield the number of subdivisions (referred by the variable orderby). All the above parameters (minimum(mino), maximum(maxo) and orderby) would be the inputs to the order option in the AXIS statement. Alternatively, the standard deviation can be used to determine the number of subdivisions within each unit. Ensure that the dataset is sorted back to its original order by visit. The above-described procedure can be implemented in code as shown below.
PROC SORT DATA=DS2;
    BY COL1;
    RUN;
/* Create macro variables that serve as parameters in the order statement for vertical axis */
DATA DS2;
    RETAIN MINTEMP;
    SET DS2 END=EOF;
    IF _N_ EQ 1 THEN MINTEMP=COL1;
    IF EOF THEN DO;
        MAXTEMP=COL1;
        ORDERBY_=ROUND(((MAXTEMP - MINTEMP)/10),0.1);
        IF ORDERBY_ EQ 0 THEN
            ORDERBY_=ROUND(((MAXTEMP-MINTEMP)/10),0.01);
        TEMP1=ROUND(MAXTEMP+ORDERBY_,0.1);
        IF MINTEMP GT 0 THEN
            TEMP2=ROUND(MINTEMP-ORDERBY_,0.01);
        ELSE TEMP2=MINTEMP-ORDERBY_; 
    IF TEMP1 GT TEMP2 THEN DO;
        CALL SYMPUT('MINO', PUT(TEMP2, BEST12.));
        CALL SYMPUT ('MAXO', PUT(TEMP1, BEST12.));
    END;
    ELSE DO;
        CALL SYMPUT('MINO', PUT(TEMP1, BEST12.));
        CALL SYMPUT ('MAXO', PUT(TEMP2, BEST12.));
    END;
    CALL SYMPUT ('ORDERBY', PUT(ORDERBY_, BEST12.));
    END;
RUN;
PROC SORT DATA=DS2;
    BY VISIT;
    RUN;
The plot procedure can now be implemented. Keeping in mind the objective of obtaining a generalized procedure to create error bar related graphs, less importance has been given to the plethora of options that come along with the GOPTIONS, SYMBOL, AXIS and other statements.

symbol1 v=star c=black i=hiloctj mode=include;
proc gplot data=ds2;
/* Symbol1 represents the symbol used to represent the data points
mode=include; specifies SAS to use all the values in the dataset
v=symbol to be used; I represents the interpolation method and hiloctj=
connects the lower and upper error bar with a straight line and also
connects each of the means with a straight line; c=color */
/*axis 1 refers to the vertical axis */
axis1 label=(angle=90 'Parameter')
order=(&mino. to &maxo. by &orderby.) width=1 offset=(0,0);
/*axis 2 refers to the horizontal axis */
axis2 label=( 'Visits');
/* plot values.
vref basically draws a reference line at Y value=0,color is black and
lvref=20 draws a dotted line. The different patterns available can be
obtained from the statgraph subsection of the support section in SAS.
*/
plot col1*visit/ vref=0 cvref=black lvref=20
haxis=axis2
vaxis=axis1
;
run;
quit;
The graph should be similar to Figure 1.

![Graph](image)

**Figure 1.** An example of a standard error of the difference of the means between two treatments.

**CASE 2**

The same technique can be used to create a standard error bar for comparing different treatments against two parameters. Let there be two treatments of interest (A and B) and let the parameters to be compared be ‘X’ and ‘Y’. We now want to compare the change from baseline (CHG) for these two parameters side by side for both the treatments (A and B). After the means for each of the parameters within each of the treatments are obtained using the PROC MEANS procedure, the output dataset is then transposed twice with the transpose variables being meanA, minA(meanA-standard errorA) and max(meanA+standard errorA). The two transposed datasets are then merged. The merged dataset (DS2) should be of the structure as described below.
The error bars can then be plotted using PROC GPLOT.

The data looks like this:

<table>
<thead>
<tr>
<th></th>
<th>Week 1</th>
<th>Week 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>MEAN_X</td>
<td>MEAN_X</td>
</tr>
<tr>
<td>A</td>
<td>MIN_X</td>
<td>MIN_X</td>
</tr>
<tr>
<td>A</td>
<td>MAX_X</td>
<td>MAX_X</td>
</tr>
<tr>
<td>A</td>
<td>MEAN_Y</td>
<td>MEAN_Y</td>
</tr>
<tr>
<td>A</td>
<td>MIN_Y</td>
<td>MIN_Y</td>
</tr>
<tr>
<td>A</td>
<td>MAX_Y</td>
<td>MAX_Y</td>
</tr>
</tbody>
</table>

The two legend statements are used one for each of the plot statements for each of the parameters X and Y. The down option is used so that each of the legend values appear one below the other. Down=2 allocates two rows. */

```/*  Two legend statements are used one for each of the plot statements for each of the parameters X and Y. The down option is used so that each of the legend values appear one below the other. Down=2 allocates two rows. */
legend1 label=(justify=left 'Treatment:') down=2 position=(top center outside) value=(tick=1 justify=left
"A; X" tick=2 justify=left "B; X");
legend2 label=(justify=left ' ') down=2 position=(top center outside) value=( tick=1 justify=left
"A; Y" tick=2 justify=left "B; Y");```
The graph output should be similar in structure to Figure 2.

Figure 2. An example of mean and standard error between two treatments (A and B) for two different parameters (X and Y)
It can be noticed that the spacing between intervals along the x-axis is not equal and reflects true intervals. This can be achieved using proc format and using the picture format for visits that are not represented.

```
proc format;
picture trtoth
  4=" "
  6=" "
;
value visit
  1="Week 1"
  2="Week 2"
  3="Week 3"
  5="Week 5"
other=[trtoth.]
;
run;
```

Also there is a slight shift between the error bars for each of the parameters. This is done so that the error bars within each parameter do not overlap and can be visualized better and is achieved by adding a small numerical value (0.04 etc) to the visit (numeric attribute) for one of the treatment arms[1].

**CONCLUSION**

A methodology has been shown that can easily create error bar plots and similar graphs. The above code can be made into a macro as the most essential things required are a statistical procedure, a transpose procedure and a very simple plot procedure.

**ACKNOWLEDGEMENTS**

I would like to thank my colleagues at Octagon Research Solutions for their encouragement and support especially my director Terek Peterson, my project manager Brian Shilling, Katie Mao for
independently implementing the technique, Frank Canale for his suggestions and to Allan Glaser for critiquing the paper. I would also like to thank MCS GLOBAL Inc for their support.

COPYRIGHT

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the United States of America and other countries. ® indicates USA registration.

Other brand or product names are registered trademarks or trademarks of their respective companies.

REFERENCE


Author contact

Sanjiv Ramalingam
Consultant (MCS GLOBAL Inc.)
Octagon Research Solutions
Wayne, PA-19087
r.sanjiv@yahoo.com