

## Easy-to-use Macros to Create a Line Plot with Error Bars Including Functions of Automated Jittering and Adjusting Axis Ranges

Andrew Miskell, GlaxoSmithKline, Research Triangle Park, NC

Peixin Sun, GlaxoSmithKline, Research Triangle Park, NC

Yufei Du, GlaxoSmithKline, Research Triangle Park, NC

### ABSTRACT

This paper presents three macros to create a line plot with error bars that includes automatic jittering and adjusting axis ranges within the plot. The purpose of these three macros is 1) to create a mean or median line plot with error bar of standard error, standard deviation, or range that has individual panels for each value of a parameter and also pages by a different parameter; 2) to automatically jitter the x-axis variable based on the number of lines in the graph; 3) to dynamically define axis ranges that are identical within each page-by parameter yet vary across page-by parameters with options to incorporate up to two reference lines in the range. The jittering and the axis range macros are stand alone macros and can be used apart from the first macro as well.

SAS version 9.1 is used for these macros. Windows platform needed and only tested on version 9.1 and above. Basic SAS skills with understanding of macro calls needed.

### INTRODUCTION

It is often necessary to create multi-page and multi-panel line plots with error bars for reporting needs. Within these line plots, there are multiple parameters, dosing regimens, days, and event times. Each parameter and treatment needs to have a separate panel and a separate line within each panel for each day. Error bars are also needed for each data point in the graph.

The GPLOT procedure is useful in creating multi-panel and multi-page line-plots, however there are limitations to the procedure. For instance in a multi-panel plot, the y-axis is not standardized across the panels (e.g., the user is forced to either have one range across all panels or free-scale all panels). Also, if there are multiple groups within one x-axis point, separating each group by jittering x-axis is often needed for better visibility.

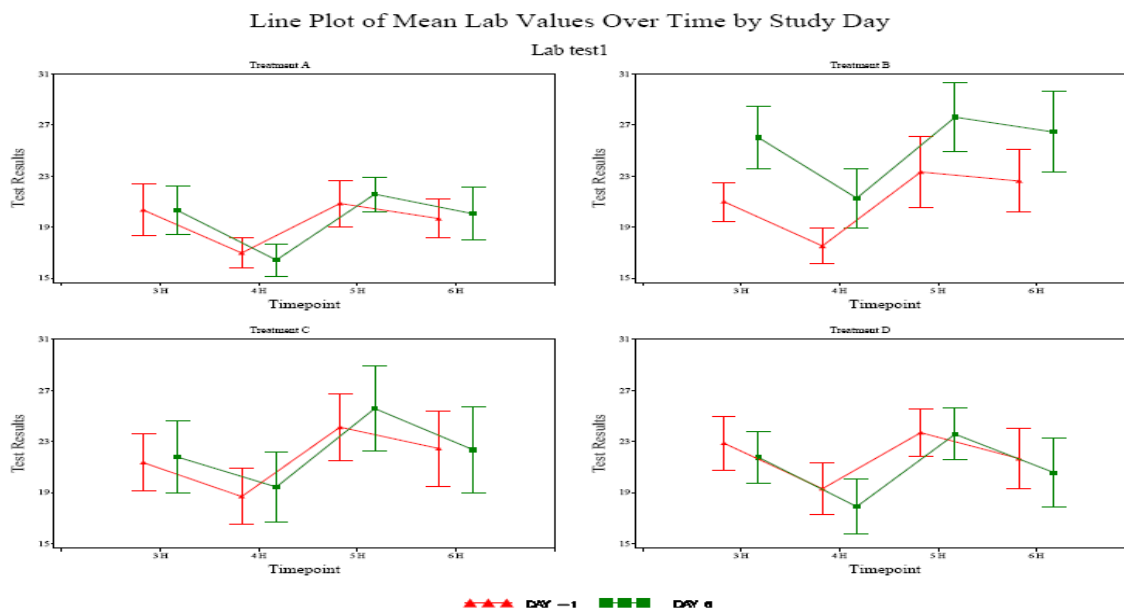
Three macros have been developed to meet these graph criteria. One macro creates a multi-page and multi-panel mean and median line plots with error bars. This macro is able to produce error bar of standard error, standard deviation or full ranges. It calls two macros: one to distinguish different groups on same x-axis points (jittering macro) and another to standardize axis ranges across multiple panels within a single parameter yet automatically update for each new parameter (axis range macro). The macro is also able to plot 1, 2, or 4 graphs (in a 2x2 matrix) per page.

The jittering macro automatically jitters the x-axis variable based on the number of lines in the graph and the smallest increment between any two x-axis points. The axis range macro has the options to incorporate up to two reference lines in the range. The jitter and the axis range macros are stand alone and can be used with any other graphs as well.

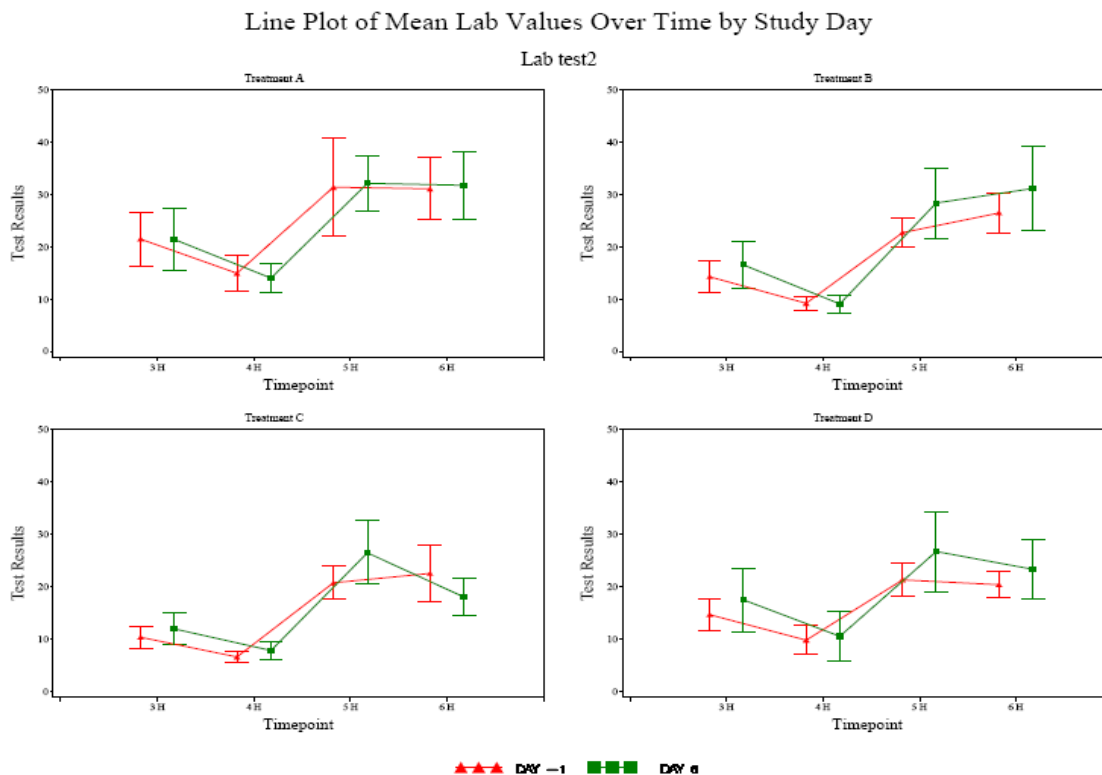
Figure 1 is a sample graph (2 pages in length) created using these stand alone macros. It is a multi-page and multi-panel mean line plot with bars representing the standard error. In this graph, the x-axis is planned time point of each lab test. The y-axis displays the lab test results. Each line represents different profile day (Day -1 vs. Day 6) and each panel displays a different treatment (A, B, C, or D). A new page is started for each new lab test. Apart from the concise and clear layout of the entire graph for different time points, days, treatments and lab tests, there exist two unique features. The first is for each line, the points are jittered to distinguish them on same time points. The second feature is each individual lab test has a single Y-axis range for all panels that is independent from other lab tests.

Figure 1 Line Plot of Mean Lab Values Over Time by Study Day

Page 1



Page 2



(Figure 1 includes 4 treatments and 2 lab tests (page 1 vs. page 2) with 4 time points (3,4,5 and 6 hours after treatment) in two different days (Day -1 vs. Day 6). The graph pages by lab tests and panels by treatments. The y-axis ranges are standardized for each lab test and the mean points are jittered for different days at the same the time point. )

## METHODOLOGY

Figures 1 above utilizes three macros working together. The LinePlotMeanMedian macro is called to create the line plot with error bar. Within this macro, the jittering macro is called to distinguish different days on same time point and the axis range macro is called to standardize y-axis ranges.

### MEAN/MEDIAN LINE PLOT WITH ERROR BAR MACRO (%LinePlotMeanMedian)

This macro can create multiple pages (e.g. page by parameter) and multiple panels (e.g. panel by treatment) within a page that include a mean or median line plot with error bars of standard error, standard deviation or full ranges.

The macro first of all performs parameter validation to provide feedback to the user if a parameter is used in the wrong way. This validation includes checking for proper variable types, correct number of variables in each parameter, and invalid values passed.

Where necessary, the macro will parse parameters of multiple words to be used later in the macro code. If either a code-decode pair or a format is passed, the macro will read that into a format to be used to create the graph.

The macro will calculate the line points (either Mean or Median) as well as the error bars (either Standard Error, Standard Deviation, or Range of Data).

PROC GREPLAY is next called to set up the graph page. A different GREPLAY is called based on if the macro is creating 1, 2, or 4 graphs per page. There is one element of the GREPLAY template for each panel on a page plus an extra element for titles, footers, and legend. For example, if 4 graphs per page are required, then the GREPLAY would have 5 elements (one per panel and one for the titles, footers, and legend).

The macro will count how many lines exist in the data. If multiple lines exist, the jittering macro will be called. Following this call, the new variable created for the jittered values will be renamed back to the original x-axis variable.

If there is a variable to have multiple panels per lab test, then the macro will iterate through each lab test, calling the axis range macro once per lab test to assign a single y-axis range each time. Following the call to define the axis range, the axis statement is used to set up the y-axis.

The annotate dataset is set up to draw the error bars as well as annotate the label of the treatment grouping to the top of the panel if necessary.

The macro will then count how many panels are in each lab test and call PROC GPLOT for each separate panel. Once the macro has iterated through all the panels for a treatment, it will count how many panels are left that will fit on the page. For instance, if the parameter dictates that there shall be 4 graphs per page and there are 5 panels within a lab test, then the first page would be filled and the second page would have one panel with three blank ones. For each blank panel needed, the macro will call PROC GSLIDE to produce a blank graph for that panel. This will enable the next lab test to start a new page.

Once the macro has iterated through each panel within each lab test, PROC GREPLAY within an ODS PDF statement is called to produce the final graph. The iteration will go through in order to place each panel on the page, including the blank panels.

At the end of the macro, the graph catalog will be cleared to prevent a new run of the macro from using old graphs.

### FUNCTIONALITY OF %LinePlotMeanMedian

- Plots line of mean or median with error bar (top or bottom or both)
- Plots multiple pages with 1, 2, or 4 panels on each page.
- Each graph panel handles up to 10 different lines
- Page, panel and group variables are defined by users

### MACRO PARAMETERS OF %LinePlotMeanMedian

The macro contains 23 parameters, 7 of which are required and the remaining are optional.

For several parameters (xvar, zvar, pagevar) there exists two different ways to format a variable, if desired. The first way is to specify a code-decode pair. The code variable will determine how the data is sorted and plotted while the decode variable will show up on the graph. The same functionality can be obtained by specifying one variable in the parameter and then using the formats parameter to specify a format.

The parameters details are as follows:

Easy-to-use Macros to Create a Line Plot with Error Bars Including Functions of Automated Jittering and Adjusting Axis Ranges, continued

Parameter	Description	Default/Notes
libname	Area where dataset is located. This must be in the format of a Windows page (i.e. C:\SAS)	Optional, If a work dataset, then leave blank
dsn	Dataset name with no library preceding it and no file extension (i.e. LABTEST and not LABTEST.sas7bdat)	<b>Required</b>
xvar	One or two variables If two variables, then must be a code/decode pair	<b>Required</b> , if this needs to be formatted, put the format in the format parameter below.
yvar	one variable	<b>Required</b>
zvar	One or two variables If two variables, then must be a code/decode pair	<b>Required</b> , line by variable
panelvar	One variable	Optional, Panel by variable within a value of page-by variable
pagevar	One or two variables If two variables, then must be a code/decode pair	Optional, Page by variable, sorted before trellis variable
meanmed	Specify Mean or Median for statistic of graph line points	Use value 'mean' or 'median', Required, not case sensitive
bartype	Specify SD, SE, or Range for statistic of bar height (range implies minimum and maximum)	Use value 'sd' or 'se', 'range' Required, not case sensitive
formats	Formats to go in the proc gplot statement	Optional
title	Title of Graph	Optional
xaxislabel	x-axis label	Optional
yaxislabel	y-axis label	Optional
xmin	x-axis minimum	Optional, If this parameter is populated, then xmax must also be populated. Warning - be sure to include extra room for jittering below the minimum x value in your data
xmax	x-axis maximum	Optional, If this parameter is populated, then xmin must also be populated. Warning - be sure to include extra room for jittering above the maximum x value in your data
xint	x-axis interval	Optional, If this parameter is populated, then both xmin and xmax need to be populated
ymin	y-axis minimum	Optional, If this parameter is populated, then ymax must also be populated.
ymax	y-axis maximum	Optional, If this parameter is populated, then ymin must also be populated.
yint	y-axis interval	Optional, If this parameter is populated, then both ymin and ymax need to be populated
graphspage	Number of graphs per page	<b>Required</b> , Value MUST be 1, 2, or 4
topbaronly	Tell macro only produce top bar	Optional: If Y then only the top bar will be displayed. If N then top and bottom bars will be displayed. This parameter does not apply

		if bartype=range.
yaxisfreeyn	Tell macro if y-axis should be free scaled	Optional: If Y or <blank> then each panel will be free scaled. If N then each page-by-grouping will have identical y-axes. If N and no page by-groupings then every panel will have identical y-axes. NOTE: If ymin and ymax are populated then yaxisfreeyn will be overwritten and y-axis will default to those values for every panel.
outputpathfilename	Path and file name of output file	<b>Required</b> , must be a PDF extension

### SAMPLE CALL OF %LinePlotMeanMedian

```
%LinePlotMeanMedian(libname=W:\USER\YD\PharmaSUG\final,
                    dsn=paperdata1,
                    xvar=time,
                    yvar=value,
                    zvar=day,
                    pagevar=param,
                    panelvar=treat,
                    meanmed=mean,
                    bartype=se,
                    formats=,
                    title=Lab Test Results,
                    xaxislabel=Timepoint,
                    yaxislabel=Test Results,
                    xmin=,
                    xmax=,
                    xint=,
                    ymin=,
                    ymax=,
                    yint=,
                    graphspage=2,
                    topbaronly=N,
                    yaxisfreeyn=N,
outputpathfilename=W:\USER\YD\PharmaSUG\final\profilesmean_twopage.pdf);
```

### JITTERING MACRO (%jitterx)

This macro automatically jitters the x-axis variable based on the number of lines in the graph and the smallest increment between any two x-axis points.

The macro will determine the smallest interval between any two x-axis points. This distance will be assigned the values of 'jitmin'. The number of lines (i.e. treatments) will then be counted and assigned to a local macro variable called 'trtcnt'.

if there are no treatments or only 1 treatment, then no jittering will be done as none is needed.

The next step if jittering is required is to assign the range of jittering. The goal is to jitter the points enough to be distinguished but not enough to overlap with a preceding or succeeding x-axis value. To do this, the maximum range of jittering is determined. If there are between 2 and 4 lines to be jittered, then the jittering range will be set to 35% of the jitmin variable from above. If 5 or more lines need to be jittered, then the jittering range will be 50% of the value of jitmin. This range will be centered around the original x-axis value.

As an example, if the smallest increment between any two points is 1 and there are 5 treatments, then the jittering range would be set to .5. if the initial x-axis value is 2, the jittering area would be centered around that point so jittered values would exist from 1.75 to 2.25.

After the jittering area is determined, each treatment is evenly spaced throughout that range. In the example above with 5 treatments and a jittering area from 1.75 to 2.25, the first treatment would be assigned to 1.75, the second to 1.875, the third to 2.00, the fourth to 2.125, and the fifth to 2.25. The 'jitterx' variable will be created to hold these new jittered values.

The initial dataset is passed out of the macro in the same state as it entered except with the added jitterx variable.

Please note that in most cases, there are three local macros declared in this macro – trtcnt, jitrng, jitinc. However, due to the functionality of the LinePlotMeanMedian macro above, the jitinc macro was made into a global macro variable for the singular purpose of this specific graph macro. It is recommended as part of good programming practice to ensure jitinc is also defined as a local macro variable for most purposes.

### FUNCTIONALITY OF %jitterx

- Jitters the x-axis variable based on the number of lines in the graph and the smallest increment between any two x-axis points automatically
- 

### MACRO PARAMETERS OF % jitterx

The macro contains 4 parameters, all of which are required. The parameters details are as follows:

Parameter	Description	Default/Notes
dsn	Input dataset name (must be a one or two-level SAS dataset name (i.e. either JITTERDSET or MYAREA.JITTERDSET). A folder path will not work.	<b>Required</b>
outdsn	Output dataset name (must be a one or two-level SAS dataset name (i.e. either JITTERDSET or MYAREA.JITTERDSET). A folder path will not work.	<b>Required</b>
xvar	X variable	<b>Required</b> , must be numeric
trtvar	Line by variable	<b>Required</b> , can be numeric or character

### SAMPLE CALL OF % jitterx

```
%jitterx(dsn=vsanal,
         outdsn=vsanal1,
         xvar=visitnum,
         trtvar=trtcd);
```

### AXIS RANGE MACRO

This macro standardizes axis ranges across multiple panels yet automatically adjusts the range for each new parameter. There are also options to include up to two reference lines in the range. This will ensure if there are reference lines outside of the data range that they get included as well.

The macro will incorporate the low and the high values specified into the y-variable to be included in the axis range. The macro will then calculate the range from the minimum value to the maximum value (inclusive of the Y-variable, low reference line, and the high reference line).

This range will be slotted into an exponent of 10 based around a multiple of 1.7 for the purposes of rounding the axis range to an appropriate number. For instance, if the axis range falls between the values of 1.7 (inclusive) and 17, the round value will be assigned a value of 10 (10<sup>1</sup>=10). If the range is between 17 and 170, the assigned round value will be 100 (10<sup>2</sup>=100). This will work for small numbers (i.e. ranges significantly less than 1) as well.

Once the round value is assigned, the minimum of the data range will be rounded down to the closest value and the maximum value will be round up to the closest value.

For instance, if the minimum axis value is 38 and the maximum axis value is 144, then the axis range is 106. This would fall in to the round value of 100 from the formula above. Therefore the minimum and maximum would be rounded to the nearest 10 with the axis range included. Therefore, the axis would span from 30 to 150 (an axis range of 120).

After the axis range is determined, the tick mark intervals are calculated. This is also done with care to ensure logical

and aesthetic numbers. The final axis range is divided by 10 raised to the round value (i.e.  $120/10^1$ ) in this example. This will give an integer value. Based on this integer, the axis range will be broken into even intervals. In this case, the integer is 12 so there will be intervals of  $2 \times 10^2$  (i.e. 20). So the tick marks will be 30, 50, 70, 90, 110, 130, 150.

The code is set up to always give reasonable intervals. If the range had been 150 or 180 rather than 120, then the interval length would have been 30.

The macro will output the axis minimum, axis maximum, and axis interval length in three global variables (axisymin, axisymax, and axisyint respectively). These global variables can then be used outside the macro to utilize in an axis statement (i.e. `order=(&axisymin to &axisymax by &axisyint)`).

### FUNCTIONALITY OF %axisrange

- This macro takes a numeric variable from an input dataset and calculates valid axis ranges and outputs them in the form of global macro variables axisymin (for the low end of the axis), axisymax (for the high end of the axis), and axisyint (for the interval length between each tick mark).
- The user can optionally have a low range and/or high range variable or a constant that will ensure that the range encompasses it.
- The macro chooses an axis range and interval length for each parameter that is both close to the actual range of data and is rounded to a reasonable number.

### MACRO PARAMETERS OF % axisrange

The macro contains 4 parameters, 2 of which are required. The parameters details are as follows:

Parameter	Description	Default/Notes
dsn	Input dataset name (must be a one or two-level SAS dataset name (i.e. either JITTERDSET or MYAREA.JITTERDSET). A folder path will not work.	<b>Required</b>
yvar	Y variable	<b>Required</b> , must be numeric
low	Normal Range Low variable or numeric constant to be included in range interval or blank	Optional. If blank, macro will automatically choose the best values
hi	Normal Range High variable or numeric constant to be included in range interval or blank	Optional. If blank, macro will automatically choose the best values

### SAMPLE CALL OF %axisrange

```
%axisrange(dsn=lab2,yvar=lbstresn);
```

### CONCLUSION

As the examples above show, these three macros are easy to use and are equipped with lots of options and flexibility. The macros provide an easy and efficient way to create a complicated yet commonly used mean/median line plot. Also, the jittering and axis range macros can be used separately as stand-alone tools or they can be combined with other graph macros or programs as well. These two macros make distinguishing line-by variables on the same point on the x-axis and presenting the y-axis ranges in reasonable and aesthetic intervals as simple as inputting several arguments in macro call. These three macros have been used in clinical trial reporting and are enabled to be flexible and robust.

### REFERENCES

SAS Institute Inc., SAS 9.1.3 Documentation, Cary, NC: SAS Institute Inc.

## ACKNOWLEDGMENTS

Authors would like to thank Denise Shortino and David Collins for reviewing the manuscript and providing very helpful suggestions and comments.

## RECOMMENDED READING

- SAS® Online Help
- The Little SAS® Book a primer

## CONTACT INFORMATION

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

Name: Andrew Miskell  
Enterprise: GlaxoSmithKline, Inc.  
Address: 5 Moore Drive  
City, State ZIP: Research Triangle Park, NC 27709  
Work Phone: 919-483-7035  
Fax: 919-483-8948  
E-mail: [andy.t.miskell@gsk.com](mailto:andy.t.miskell@gsk.com)

Name: Peixin Sun  
Enterprise: GlaxoSmithKline, Inc.  
Address: 5 Moore Drive  
City, State ZIP: Research Triangle Park, NC 27709  
Work Phone: 919-483-2686  
Fax: 919-483-8948  
E-mail: [peixin.x.sun@gsk.com](mailto:peixin.x.sun@gsk.com)

Name: Yufei Du  
Enterprise: GlaxoSmithKline, Inc.  
Address: 5 Moore Drive  
City, State ZIP: Research Triangle Park, NC 27709  
Work Phone: 919-483-8680  
Fax: 919-483-8948  
E-mail: [yufei.a.du@gsk.com](mailto:yufei.a.du@gsk.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.



## APPENDIX

### SAS CODE OF %LinePlotMeanMedian

```

/*****
|
| Macro Name:      LinePlotMeanMedian
|
| Macro Version:   1.1
|
| SAS Version:    9.1.3
|
| Created By:     Andy Miskell, Peixin Sun, Yufei Du
|
| Date:          September 3, 2010
|
| Macro Purpose:  This macro will read in a SAS dataset and create a line plot of mean or median
|                  values with error bars showing either Standard Deviation, Standard Error,
|                  or Range of the data.  Optionally, line-by, panel-by, and page-by variables
|                  may also be specified.  The macro will also output either 1, 2, or 4 graphs
|                  per page.
|
| Macro Design:   Procedure Style
|
|*****/

%include "U:\PharmaSUG\jitterx.sas";
%include "U:\PharmaSUG\axisrange.sas";

%macro LinePlotMeanMedian(libname=, /* Optional: area where dataset is located. If a work
dataset,then leave blank */
                        dsn=, /* Required: dataset name with no library preceding it */
                        xvar=, /* Required: One or two variables (code/decode pair) - if
this needs to be formatted,put the format in the format
parameter below */
                        yvar=, /* Required: one variable */
                        zvar=, /* Required: one or two variables for profile day
(code/decode pair).Line-by variable */
                        panelvar=, /* Optional: one variable. Panel by treatments */
                        pagevar=, /* Optional: one or two variables for paging
(code/decode pair).Page-By-Groupings - sorted before
trellis variable */
                        meanmed=, /*Specify Mean or Median for graph line points*/
                        bartype=, /* Specify SD, SE, or Range for statistic of bar height
(range implies minimum and maximum */
                        formats=, /* Optional:formats to go in the proc gplot statement*/
                        title=, /* Optional: Title of Graph */
                        axislabel=, /* Optional: x-axis label */
                        yaxislabel=, /* Optional: y-axis label */
                        xmin=, /* Optional: x-axis minimum.If this parameter is
populated,then xmax must also be populated
Warning - be sure to include extra room for jittering
below the minimum x value in your data */
                        xmax=, /* Optional: x-axis maximum. If this parameter is
populated, then xmin must also be populated
Warning - be sure to include extra room for jittering

```

Easy-to-use Macros to Create a Line Plot with Error Bars Including Functions of Automated Jittering and Adjusting Axis Ranges, continued

```

        above the maximum x value in your data */
xint=, /*Optional: x-axis interval.If this parameter is
        populated, then both xmin and xmax need to be populated*/
ymin=, /* Optional:y-axis minimum.If this parameter is
        populated, then ymax must also be populated */
ymax=, /* Optional: y-axis maximum.If this parameter is
        populated, then ymin must also be populated */
yint=, /* Optional: y-axis interval.If this parameter is
        populated,then both ymin and ymax need to be populated*/
graphspage=, /* Required: Number of graphs per page. Value MUST be
        1, 2, or 4 */
topbaronly=N, /* Optional: If Y then only the top bar will be
        displayed. If N then top and bottom bars will be
        displayed.This parameter does not apply if
        bartype=range.*/
yaxisfreeyn=N, /* Optional: If Y or <blank> then each panel will be
        free scaled. If N then each by-grouping will have
        identical y-axes. If N and no by-groupings then every
        panel will have identical y-axes. Note: If ymin and
        ymax are populated then yaxisfreeyn will be
        overwritten and y-axis will default to those values for
        every panel.*/
outputpathfilename= /* Required: Path and file name of output file (must be a
        PDF extension ) */
);

/* Notes:
   Only handles up to 10 different profile day lines distinctively. After that, symbols will
   start repeating.
   y-axis label will be rotated incorrectly if run in SAS 9.2. There is no way to get it to
   run the same in 9.1 and 9.2.
*/

options spool;
%let meanmed=%upcase(&meanmed);
%let bartype=%upcase(&bartype);

%let title = %nrbrquote(&title.);
%let xaxislabel = %nrbrquote(&xaxislabel.);
%let yaxislabel = %nrbrquote(&yaxislabel.);
%let outputpathfilename = %nrbrquote(&outputpathfilename.);

title ' ';

/* Read in data */
%if &libname^= %then %do;
    libname topbot "&libname" access=readonly;

    data work.pdtopbot (keep=&xvar &yvar &panelvar &zvar &pagevar);
        set topbot.&dsn;
    run;
%end;
%else %do;
    data work.pdtopbot (keep=&xvar &yvar &panelvar &zvar &pagevar);
        set work.&dsn;
    run;
%end;

```

Easy-to-use Macros to Create a Line Plot with Error Bars Including Functions of Automated Jittering and Adjusting Axis Ranges, continued

```
%local trtttext;
%let trtttext= ;

/* Parse out macro parameters for x-axis, treatment groupings, and profile day variables */
%local xvarcount zvarcount trtvarcount firsttrt lasttrt firstx lastx firstz lastz firstby lastby;

%if %scan(&xvar,2,' ')^= %then %do;
    %let xvarcount=2;
%end;
%else %do;
    %let xvarcount=1;
%end;
%if %scan(&zvar,2,' ')^= %then %do;
    %let zvarcount=2;
%end;
%else %do;
    %let zvarcount=1;
%end;
%if %scan(&panelvar,2,' ')^= %then %do;
    %let trtvarcount=2;
%end;
%else %if %scan(&panelvar,1,' ')^= %then %do;
    %let trtvarcount=1;
%end;
%else %do;
    %let trtvarcount=0;
%end;
%if %scan(&pagevar,2,' ')^= %then %do;
    %let byvarcount=2;
%end;
%else %do;
    %let byvarcount=1;
%end;

%if &xvarcount=1 %then %do;
    %let firstx=&xvar;
    %let lastx=&xvar;
%end;
%if &zvarcount=1 %then %do;
    %let firstz=&zvar;
    %let lastz=&zvar;
%end;
%if &trtvarcount=1 %then %do;
    %let firsttrt=&panelvar;
    %let lasttrt=&panelvar;
%end;
%if &byvarcount=1 %then %do;
    %let firstby=&pagevar;
    %let lastby=&pagevar;
%end;
%if &xvarcount=2 %then %do;
    %let firstx=%scan(&xvar,1,' ');
    %let lastx=%scan(&xvar,2,' ');
%end;
%if &zvarcount=2 %then %do;
    %let firstz=%scan(&zvar,1,' ');
```

Easy-to-use Macros to Create a Line Plot with Error Bars Including Functions of Automated Jittering and Adjusting Axis Ranges, continued

```
%let lastz=%scan(&zvar,2,' ');
%end;
%if &trtvarcount=2 %then %do;
  %let firsttrt=%scan(&panelvar,1,' ');
  %let lasttrt=%scan(&panelvar,2,' ');
%end;
%if &byvarcount=2 %then %do;
  %let firstby=%scan(&pagevar,1,' ');
  %let lastby=%scan(&pagevar,2,' ');
%end;

/* Find the statistics of the data for each by-variable, profile day variable, timepoint, and z-
variable (i.e. treatment) */
proc sort data=pdtopbot;
  by &pagevar &panelvar &xvar &zvar;
run;

proc univariate data=pdtopbot noprint;
  by &pagevar &panelvar &xvar &zvar;
  var &yvar;
  output out=pdtopbot mean=mean std=sd n=num min=min max=max stdmean=se median=median;
run;

data pdtopbot (keep=&pagevar &panelvar &xvar &zvar &yvar statindex);
  set pdtopbot;
  by &pagevar &panelvar &xvar &zvar;
  length barvar meanmed $ 10;
  barvar=trim(left("&bartype"));
  meanmed=trim(left("&meanmed"));
  if meanmed='MEAN' then startvalue=mean;

  if meanmed='MEDIAN' then startvalue=median;
  if barvar='SE' then do;
    hivalue=startvalue+se;
    lovalue=startvalue-se;
  end;
  if barvar='SD' then do;
    hivalue=startvalue+sd;
    lovalue=startvalue-sd;
  end;
  if barvar='RANGE' then do;
    hivalue=max;
    lovalue=min;
  end;
  statindex=1;
  &yvar=startvalue;
  output;
  statindex=2;
  &yvar=hivalue;
  output;
  %if &topbaronly^=Y %then %do;
    statindex=3;
    &yvar=lovalue;
    output;
  %end;
  statindex=4;
  &yvar=startvalue;
```

Easy-to-use Macros to Create a Line Plot with Error Bars Including Functions of Automated Jittering and Adjusting Axis Ranges, continued

```

output;
run;

/* Enter format parameter values into dataset and other macro parameters */
%if &formats^= %then %do;
  %local forvar1 format1;
  /* Scan formats parameter. Find first format */
  %if %scan(&formats,2,' ')^= %then %do;
    %let forvar1=;
    %let format1=;
    /* Determine variable name and format name of first format */
    %let forvar1=%scan(&formats,1,' ');
    %let format1=%scan(&formats,2,' ');

    /* Create decoded variable of first format named formvariable1 */
    data pdtopbot;
      set pdtopbot;
      length formvariable1 $256;
      formvariable1=put(&forvar1,?? &format1.);
    run;

    /* Determine if first format is meant for profile day value, x-axis, or treatment grouping
    */
    /* For whichever one it is meant for, replace the effected macro parameters with new
    decoded variable */
    %if %index(&panelvar,&forvar1)^=0 and &firsttrt=&forvar1 %then %do;
      %let trtvarcount=2;
      %let lasttrt=formvariable1;
      %let panelvar=&firsttrt formvariable1;
    %end;
    %if %index(&xvar,&forvar1)^=0 and &firstx=&forvar1 %then %do;
      %let xvarcount=2;
      %let lastx=formvariable1;
      %let xvar=&firstx formvariable1;
    %end;
    %if %index(&zvar,&forvar1)^=0 and &firstz=&forvar1 %then %do;
      %let zvarcount=2;
      %let lastz=formvariable1;
      %let zvar=&firstz formvariable1;
    %end;
    %if %index(&pagevar,&forvar1)^=0 and &firstby=&forvar1 %then %do;
      %let byvarcount=2;
      %let lastby=formvariable1;
      %let pagevar=&firstby formvariable1;
    %end;

  %end;

  /* Scan formats parameter. Find second format */
  %if %scan(&formats,4,' ')^= %then %do;
    %let forvar1=;
    %let format1=;
    %let forvar1=%scan(&formats,3,' ');
    %let format1=%scan(&formats,4,' ');

    /* Create decoded variable of second format named formvariable2 */
    data pdtopbot;

```

Easy-to-use Macros to Create a Line Plot with Error Bars Including Functions of Automated Jittering and Adjusting Axis Ranges, continued

```

    set pdtopbot;
    length formvariable2 $256;
    formvariable2=put(&forvar1,?? &format1.);
run;

/* Determine if second format is meant for by-value, x-axis, or treatment grouping */
/* For whichever one it is meant for, replace the effected macro parameters with new
   decoded variable */
%if %index(&panelvar,&forvar1)^=0 and &firsttrt=&forvar1 %then %do;
    %let trtvarcount=2;
    %let lasttrt=formvariable2;
    %let panelvar=&firsttrt formvariable2;
%end;
%if %index(&xvar,&forvar1)^=0 and &firstx=&forvar1 %then %do;
    %let xvarcount=2;
    %let lastx=formvariable2;
    %let xvar=&firstx formvariable2;
%end;
%if %index(&zvar,&forvar1)^=0 and &firstz=&forvar1 %then %do;
    %let zvarcount=2;
    %let lastz=formvariable2;
    %let zvar=&firstz formvariable2;
%end;
%if %index(&pagevar,&forvar1)^=0 and &firstby=&forvar1 %then %do;
    %let byvarcount=2;
    %let lastby=formvariable2;
    %let pagevar=&firstby formvariable2;
%end;

%end;

/* Scan formats parameter. Find third format */
%if %scan(&formats,6,' ')^= %then %do;
    %let forvar1=;
    %let format1=;
    %let forvar1=%scan(&formats,5,' ');
    %let format1=%scan(&formats,6,' ');

/* Create decoded variable of third format named formvariable3 */
data pdtopbot;
    set pdtopbot;
    length formvariable3 $256;
    formvariable3=put(&forvar1,?? &format1.);
run;

/* Determine if third format is meant for by-value, x-axis, or treatment grouping */
/* For whichever one it is meant for, replace the effected macro parameters with new
   decoded variable */
%if %index(&panelvar,&forvar1)^=0 and &firsttrt=&forvar1 %then %do;
    %let trtvarcount=2;
    %let lasttrt=formvariable3;
    %let panelvar=&firsttrt formvariable3;
%end;
%if %index(&xvar,&forvar1)^=0 and &firstx=&forvar1 %then %do;
    %let xvarcount=2;
    %let lastx=formvariable3;
    %let xvar=&firstx formvariable3;

```

Easy-to-use Macros to Create a Line Plot with Error Bars Including Functions of Automated Jittering and Adjusting Axis Ranges, continued

```
%end;
%if %index(&zvar,&forvar1)^=0 and &firstz=&forvar1 %then %do;
  %let zvarcount=2;
  %let lastz=formvariable3;
  %let zvar=&firstz formvariable3;
%end;
%if %index(&pagevar,&forvar1)^=0 and &firstby=&forvar1 %then %do;
  %let byvarcount=2;
  %let lastby=formvariable3;
  %let pagevar=&firstby formvariable3;
%end;

%end;

/* Scan formats parameter. Find fourth format */
%if %scan(&formats,6,' ')^= %then %do;
  %let forvar1=;
  %let format1=;
  %let forvar1=%scan(&formats,7,' ');
  %let format1=%scan(&formats,8,' ');

  /* Create decoded variable of third format named formvariable3 */
  data pdtopbot;
    set pdtopbot;
    length formvariable4 $256;
    formvariable4=put(&forvar1,?? &format1.);
run;

  /* Determine if third format is meant for by-value, x-axis, or treatment grouping */
  /* For whichever one it is meant for, replace the effected macro parameters with new
  decoded variable */
%if %index(&panelvar,&forvar1)^=0 and &firsttrt=&forvar1 %then %do;
  %let trtvarcount=2;
  %let lasttrt=formvariable4;
  %let panelvar=&firsttrt formvariable4;
%end;
%if %index(&xvar,&forvar1)^=0 and &firstx=&forvar1 %then %do;
  %let xvarcount=2;
  %let lastx=formvariable4;
  %let xvar=&firstx formvariable4;
%end;
%if %index(&zvar,&forvar1)^=0 and &firstz=&forvar1 %then %do;
  %let zvarcount=2;
  %let lastz=formvariable4;
  %let zvar=&firstz formvariable4;
%end;
%if %index(&pagevar,&forvar1)^=0 and &firstby=&forvar1 %then %do;
  %let byvarcount=2;
  %let lastby=formvariable4;
  %let pagevar=&firstby formvariable4;
%end;

%end;

%end;
/* End Formats part */
```

Easy-to-use Macros to Create a Line Plot with Error Bars Including Functions of Automated Jittering and Adjusting Axis Ranges, continued

```
/* Read profile day variables into a format zvariable. */
%if &zvar^= %then %do;
  proc sort data=pdtopbot out=zvartest (keep=&zvar) nodupkey;
    by &zvar;
  run;

%local leglabl;

/* Get count of number of profile days */
data zvartest (drop=legvar);
  set zvartest;
  length legvar $ 256;
  legvar=label(&firstz);
  call symput('leglabl',legvar);
  retain dayvar;
  if _n_=1 then dayvar=0;
  dayvar+1;
run;

data zvartest;
  set zvartest;
  label dayvar="&leglabl";
run;

/* Create dataset for creating a format. Codes are index of count of profile days.
Decodes are the profile day names. */
data fmtnam2;
  set zvartest;
  length label $256;
  start=dayvar;
  fmtname='dayvariable';
  type='N';
  label=&lastz;
  label=left(label);
  codetype='TEXT';
  output;
  if _n_=1 then do;
    hlo='o';
    label=' ';
    output;
  end;
run;

/* Read into dayvariable. format */
proc format cntlin=fmtnam2;
run;

/* Merge treatment indexing back into main dataset */
proc sort data=pdtopbot out=pdtopbot;
  by &zvar;
run;

data pdtopbot;
  merge zvartest pdtopbot;
  by &zvar;
run;
```



Easy-to-use Macros to Create a Line Plot with Error Bars Including Functions of Automated Jittering and Adjusting Axis Ranges, continued

```
/* Create macro variable &daycnt for number of treatment groupings */
proc univariate data=zvartest noprint;
  var dayvar;
  output out=cntdayvar max=cntr;
run;

%local daycnt;

data cntdayvar;
  set cntdayvar;
  call symput('daycnt',put(cntr,?? 4.));
run;

%end;

/* Read treatment variables into a format zvariable. */
%local trtcnt treatvara;
%let trtcnt=0;
%let treatvara= ;

%if &panelvar^= %then %do;
  %let treatvara=treatvar;

proc sort data=pdtopbot out=pdlabtest (keep=&panelvar) nodupkey;
  by &panelvar;
run;

/* Get count of number of by-groupings in data and index them. */
data pdlabtest;
  set pdlabtest;
  retain treatvar;
  if _n_=1 then treatvar=0;
  treatvar+1;
run;

/* Create dataset for creating a format. Codes are index of count of by-groupings.
Decodes are the by-groupings. */
data fmtnam1;
  set pdlabtest;
  length label $256;
  start=treatvar;
  fmtname='treatvariable';
  type='N';
  label=&lasttrt;
  codetype='TEXT';
  output;
  if _n_=1 then do;
    hlo='o';
    label=' ';
    output;
  end;
run;

/* Read into treatvariable. format */
proc format cntlin=fmtnam1;
run;
```

Easy-to-use Macros to Create a Line Plot with Error Bars Including Functions of Automated Jittering and Adjusting Axis Ranges, continued

```
/* Merge treatment grouping indexing back into main dataset */
proc sort data=pdtopbot out=pdtopbot;
  by &panelvar;
run;

data pdtopbot;
  merge pdlabtest pdtopbot;
  by &panelvar;
run;

/* Create macro variable &trtcnt for number of treatment groupings */
proc univariate data=pdlabtest noprint;
  var treatvar;
  output out=cntbyvar max=cntr;
run;

data cntbyvar;
  set cntbyvar;
  call symput('trtcnt',put(cntr,?? 4.));
run;

%end;

/* Read by variables into a format byvariable. */
%if &pagevar^= %then %do;
  proc sort data=pdtopbot out=byvartest (keep=&pagevar) nodupkey;
    by &pagevar;
  run;

%local leglabl;

/* Get count of number of profile days */
data byvartest (drop=legvar);
  set byvartest;
  length legvar $ 256;
  legvar=label(&firstby);
  call symput('leglabl',legvar);
  retain dayvar;
  if _n_=1 then dayvar=0;
  dayvar+1;
run;

data byvartest;
  set byvartest;
  label dayvar="&leglabl";
run;

/* Create dataset for creating a format. Codes are index of count of profile days.
Decodes are the profile day names. */
data fmtnam2;
  set byvartest;
  length label $256;
  start=dayvar;
  fmtname='byvariable';
  type='N';
  label=&lastby;
  label=left(label);
```

Easy-to-use Macros to Create a Line Plot with Error Bars Including Functions of Automated Jittering and Adjusting Axis Ranges, continued

```
codetype='TEXT';
output;
if _n_=1 then do;
  hlo='o';
  label=' ';
  output;
end;
run;

/* Read into dayvariable. format */
proc format cntlin=fmtnam2;
run;

/* Merge treatment indexing back into main dataset */
proc sort data=pdtopbot out=pdtopbot;
  by &pagevar;
run;

data pdtopbot;
  merge byvartest pdtopbot;
  by &pagevar;
run;

/* Create macro variable &daycnt for number of treatment groupings */
proc univariate data=byvartest noprint;
  var dayvar;
  output out=cntdayvar max=cntr;
run;

%local bycnt;

data cntdayvar;
  set cntdayvar;
  call symput('bycnt',put(cntr,?? 4.));
run;

%end;

/* Read x variables into a format xvariable. */
%if &xvar^= %then %do;
  proc sort data=pdtopbot out=xvartest (keep=&xvar) nodupkey;
    by &xvar;
  run;

  /* Find out if x variable is character or numeric */
  proc contents data=xvartest noprint out=xdset;
  run;

  %local xvartype;

  data xdset;
    set xdset;
    length tempvar1 $ 256;
    tempvar1="&firstx";
    if upcase(name)=upcase(trim(tempvar1)) then do;
      call symput('xvartype',put(type,?? 1.));
    end;
end;
```

Easy-to-use Macros to Create a Line Plot with Error Bars Including Functions of Automated Jittering and Adjusting Axis Ranges, continued

```
run;

/* If x variable is character */
%if &xvartype=2 %then %do;
  data xvaritest;
    set xvaritest;
    retain newxvar;
    if _n_=1 then newxvar=0;
    newxvar+1;
run;

/* Create dataset for creating a format. Codes are index of count of profile days.
Decodes are the profile day names. */
data fmtnam3;
  set xvaritest;
  length label $256;
  start=newxvar;
  fmtname='xvariable';
  type='N';
  label=&lastx;
  codetype='TEXT';
  output;
  if _n_=1 then do;
    hlo='o';
    label=' ';
    output;
  end;
run;

/* Read into xvariable. format */
proc format cntlin=fmtnam3;
run;

/* Merge treatment indexing back into main dataset */
proc sort data=pdtopbot out=pdtopbot;
  by &xvar;
run;

data pdtopbot;
  merge xvaritest pdtopbot;
  by &xvar;
run;

%end;

/* If x variable is numeric */
%if &xvartype=1 %then %do;
  data xvaritest;
    set xvaritest;
    newxvar=&firstx;
run;

/* Create dataset for creating a format. Codes are index of count of profile days.
Decodes are the profile day names. */
data fmtnam3;
  set xvaritest;
  length label $256;
```

Easy-to-use Macros to Create a Line Plot with Error Bars Including Functions of Automated Jittering and Adjusting Axis Ranges, continued

```
start=newxvar;
fmtname='xvariable';
type='N';
label=&lastx;
codetype='TEXT';
output;
run;

/* Read into charxvariable. format */
proc format cntlin=fmtnam3;
run;

/* Merge treatment indexing back into main dataset */
proc sort data=pdtopbot out=pdtopbot;
  by &xvar;
run;

data pdtopbot;
  merge xvarrest pdtopbot;
  by &xvar;
run;

%end;

%end;

data workpdmean;
  set pdtopbot;
run;

proc sort data=workpdmean out=workpdmean;
  by &pagevar &panelvar &xvar &zvar statindex;
run;

/* Set up graphing options */
goptions reset=all;
goptions goutmode=replace rotate=landscape; * hsize=10.49 vsize=7.99;
options orientation=landscape ls=108 ps=65;
title ' ';
/* Create header and title information */
proc gslide gout=grph_cat name="tbox";
  title1 j=c h=1.05 font=swiss " ";
run;
quit;

goptions goutmode=append;

/* Set up graph template based on number of graphs desired per page */
%if "&graphspage"="4" %then %do;
  proc greplay tc=grph_cat nofs ;
    tdef newtemp
      /*Top left*/
      1/llx=0 lly=50 ulx=0 uly=93
        urx=49 ury=93 lrx=49 lry=50
      /*Top right*/
      2/llx=51 lly=50 ulx=51 uly=93
        urx=100 ury=93 lrx=100 lry=50
```

Easy-to-use Macros to Create a Line Plot with Error Bars Including Functions of Automated Jittering and Adjusting Axis Ranges, continued

```

/*Bottom left*/
3/llx=0   lly=6   ulx=0   uly=49
   urx=49  ury=49  lrx=49  lry=6
/*Bottom right*/
4/llx=51   lly=6   ulx=51   uly=49
   urx=100  ury=49  lrx=100  lry=6

5/ llx=0   lly=0   ulx=0   uly=100
   urx=100  ury=100 lrx=100 lry=0;
run;
%end;
%if "&graphspage"="2" %then %do;
proc greplay tc=grph_cat nofs ;
tdef newtemp
/*Top*/
1/llx=0   lly=50  ulx=0   uly=93
   urx=100  ury=93  lrx=100 lry=50
/*Bottom*/
2/llx=0   lly=6   ulx=0   uly=49
   urx=100  ury=49  lrx=100 lry=6

3/ llx=0   lly=0   ulx=0   uly=100
   urx=100  ury=100 lrx=100 lry=0;
run;
%end;
%if "&graphspage"="1" %then %do;
proc greplay tc=grph_cat nofs ;
tdef newtemp
1/llx=0   lly=3   ulx=0   uly=93
   urx=100  ury=93  lrx=100 lry=3

2/ llx=0   lly=0   ulx=0   uly=100
   urx=100  ury=100 lrx=100 lry=0;
run;
%end;

/* Set up x and y axes as well as legend and symbol statements */
axis1
  %if "%unquote(&xaxislabel)"^=" " %then %do;
/*      label=(h=1.0 cm angle=90 rotate=0 font="Times New Roman" "%unquote(&yaxislabel)") */
      label=(h=1.0 cm angle=0 rotate=0 font="Times New Roman" "%unquote(&xaxislabel)")
  %end;
  %else %do;
      label=(" ")
  %end;
  %if &xmin^= and &xint= %then %do;
      order=(&xmin to &xmax)
  %end;
  %if &xmin^= and &xint^= %then %do;
      order=(&xmin to &xmax by &xint)
  %end;
  offset=(0.15 in,0 in)
  minor=none
  value=(rotate=0 h=.6 cm f="Times New Roman")
  ;

```

Easy-to-use Macros to Create a Line Plot with Error Bars Including Functions of Automated Jittering and Adjusting Axis Ranges, continued

```
legend1 position=(bottom center outside) label=(h=.4 cm " " font="Times New Roman") value=(h=.3
cm font="swiss")
      shape=symbol(5,1);

/* If there are profile days... */
%if &zvar^= %then %do;
  /* Find all treatments in data */
  proc sort data=workpdmean out=zvarnodup (keep=&zvar dayvar) nodupkey;
    by &zvar dayvar;
  run;

  /*Find all timepoints and treatment variables in data */
  proc sort data=workpdmean out=xvarnodup (keep=&xvar &panelvar &treatvara newxvar) nodupkey;
    by &xvar &panelvar &treatvara newxvar;
  run;

  /*Create cartesian product for each timepoint, by-variable, and treatment */
  proc sql;
    create table fullproduct as
    select zvarnodup.*
           ,xvarnodup.*

    from zvarnodup
           ,xvarnodup;
  quit;

  /* The next section of code merges the full cartesian product above with the
     data for both the top and bottom graphs to ensure both graphs have all possible treatments
     on them. The y-variable will be missing if it is not already on the dataset so
     the graph will not change although this will ensure SAS takes into account the treatment
     when assigning symbols.
  */
  proc sort data=workpdmean;
    by &xvar &panelvar &treatvara newxvar dayvar &zvar;
  run;

  proc sort data=fullproduct;
    by &xvar &panelvar &treatvara newxvar dayvar &zvar;
  run;

  data workpdmean;
    merge fullproduct (in=a) workpdmean (in=b);
    by &xvar &panelvar &treatvara newxvar dayvar &zvar;
    if a or b;
  run;

  /* Count treatments and assign symbols */
  proc sort data=workpdmean out=pdtemp nodupkey;
    by &zvar;
  run;

  proc contents data=pdtemp noprint out=combdset;
  run;

  %local cntpday m ;
  /* Macro variable cnttrt will hold the total treatments for the study */
```

Easy-to-use Macros to Create a Line Plot with Error Bars Including Functions of Automated Jittering and Adjusting Axis Ranges, continued

```

data combdset;
  set combdset;
  call symput('cntpday',nobs);
run;

symbol;
symbol1 i=join h=1.5 /*v=dot*/;
%do m=1 %to &cntpday;
  %if &m=1 %then %do;
    symbol1 i=join h=1.5 /*value=triangle*/ color=red;
  %end;
  %else %if &m=2 %then %do;
    symbol2 /*value=U*/ f=marker line=1 i=join color=green;
  %end;
  %else %if &m=3 %then %do;
    symbol3 /*value=square*/ line=1 i=join color=blue;
  %end;
  %else %if &m=4 %then %do;
    symbol4 /*value=A*/ f=marker line=1 i=join color=black;
  %end;
  %else %if &m=5 %then %do;
    symbol5 /*value=V*/ f=marker line=1 i=join color=orange;
  %end;
  %else %if &m=6 %then %do;
    symbol6 /*value=Y*/ f=marker line=1 i=join color=brown;
  %end;
  %else %if &m=7 %then %do;
    symbol7 /*value=C*/ f=marker line=1 i=join color=cyan;
  %end;
  %else %if &m=8 %then %do;
    symbol8 /*value=N*/ f=marker line=1 i=join color=magenta;
  %end;
  %else %if &m=9 %then %do;
    symbol9 /*value=O*/ f=marker line=1 i=join color=purple;
  %end;
  %else %if &m=10 %then %do;
    symbol10 /*value=P*/ f=marker line=1 i=join color=yellow;
  %end;
  %else %do;
    symbol&m /*value=M*/ f=marker line=1 i=join color=blue;
  %end;
%end;
%end;
%else %do;
  symbol1 i=join /*v=dot*/ h=1.5;
%end;

/* Jitter x values */
%jitterx(dsn=workpdmean,
         outdsn=workpdmean1,
         xvar=newxvar,
         trtvar=&firstz);

data workpdmean;
  set workpdmean1;
  newxvar=jitterx;
run;

```



Easy-to-use Macros to Create a Line Plot with Error Bars Including Functions of Automated Jittering and Adjusting Axis Ranges, continued

```
/* If treatment variable parameter is specified ... */
%if &panelvar^= %then %do;

%local i subjtrt pagecnt;

/* Initialize page count to 0 */
%let pagecnt=0;
%let pagecnt=%eval(&pagecnt+1);

/* Allow program to run the same whether or not there are by-variables */
%if &pagevar= %then %do;

%let bycnt=1;
%let pagevar=byvar_dummyvariable;
%let firstby=byvar_dummyvariable;

data workpdmean;
  set workpdmean;
  byvar_dummyvariable=1;
run;

%end;

proc sort data=workpdmean;
  by &pagevar;
run;

data workpdmean;
  set workpdmean;
  by &pagevar;
  retain bycntvar;
  if _n_=1 then bycntvar=0;
  if first.&firstby then bycntvar+1;
run;

%do a=1 %to &bycnt;
  %if &a^=1 %then %let pagecnt=%eval(&pagecnt+1);

  data workpdmeanby;
    set workpdmean;
    if bycntvar=&a;
  run;

  %axisrange(dsn=workpdmeanby,
             yvar=&yvar);

axis2
  %if "%unquote(&yaxislabel)"^= " " %then %do;
    label=(h=1.0 cm angle=90 rotate=0 font="Times New Roman" "%unquote(&yaxislabel)");
  %end;
  %else %do;
    label=(h=0.5 cm angle=90 rotate=0 " ");
  %end;
  %if &ymin^= and &yint= %then %do;
    order=(&ymin to &ymax)
  %end;
```

Easy-to-use Macros to Create a Line Plot with Error Bars Including Functions of Automated Jittering and Adjusting Axis Ranges, continued

```

    %else %if &ymin^= and &yint^= %then %do;
        order=(&ymin to &ymax by &yint)
    %end;
    %else %if &yaxisfreeyn=N %then %do;
        order=(&axisymin to &axisymax by &axisyint)
    %end;
    offset=(0.15 in,0 in)
    minor=none
    value=(h=0.65 cm font="Times New Roman")
    ;

/* Initialize number of graphs on current page to 0 */
%local modfour;
%let modfour=0;
/* For each treatment, iterate separately */
%do i=1 %to &trtcnt;
/* Increase page count by 1 for new page */

    /* Subset for only current treatment */
    data thistrt;
        set workpdmeanby;
        if treatvar=&i;
    run;

    /* Increase the graph on current page count by 1 */
    %let modfour=%eval(&modfour+1);
    /* Subset for only current subject */
    data thissubj;
        set thistrt;
    run;

    /* If the old page is full, increase page count by 1 and reset graphs on current page to 1
*/
    %if "&graphspage"="4" %then %do;
        %if "&modfour"="5" %then %do;
            %let pagecnt=%eval(&pagecnt+1);
            %let modfour=1;
        %end;
    %end;
    %if "&graphspage"="2" %then %do;
        %if "&modfour"="3" %then %do;
            %let pagecnt=%eval(&pagecnt+1);
            %let modfour=1;
        %end;
    %end;
    %if "&graphspage"="1" %then %do;
        %if "&modfour"="2" %then %do;
            %let pagecnt=%eval(&pagecnt+1);
            %let modfour=1;
        %end;
    %end;

    proc sort data=thissubj;
        by dayvar newxvar statindex;
    run;

```

## Easy-to-use Macros to Create a Line Plot with Error Bars Including Functions of Automated Jittering and Adjusting Axis Ranges, continued

```
/* Set up annotate dataset. Must add treatment titles, error bars and caps, and symbols
for the mean/median */
data anno;
  set thissubj;
  format newxvar xvariable. dayvar dayvariable. treatvar treatvariable.;
  length text $ 200 color $ 10 function $ 8 style $ 20;
  style="Times New Roman";
  if _n_ = 1 then do;
    function='label';
    xsys='3'; ysys='3';
    x=50;
    y=96; text=trim(left(put(treatvar,?? treatvariable.))); size=2; output;
  end;
  jitinc=&jitinc;
  xmin="&xmin";
  xmax="&xmax";
  xsys='2';
  ysys='2';
  xmin1=input(xmin,?? best.);
  xmax1=input(xmax,?? best.);
  if xmin1 ne . and xmax1 ne . and (newxvar lt xmin1 or newxvar gt xmax1) then delete;
  if dayvar=1 then do;
    color='red';
    text='C';
  end;
  else if dayvar=2 then do;
    color='green';
    text='U';
  end;
  else if dayvar=3 then do;
    color='blue';
    text='U';
  end;
  else if dayvar=4 then do;
    color='black';
    text='A';
  end;
  else if dayvar=5 then do;
    color='orange';
    text='V';
  end;
  else if dayvar=6 then do;
    color='brown';
    text='Y';
  end;
  else if dayvar=7 then do;
    color='cyan';
    text='D';
  end;
  else if dayvar=8 then do;
    color='magenta';
    text='N';
  end;
  else if dayvar=9 then do;
    color='purple';
    text='O';
  end;
end;
```

Easy-to-use Macros to Create a Line Plot with Error Bars Including Functions of Automated Jittering and Adjusting Axis Ranges, continued

```
else if dayvar=10 then do;
  color='yellow';
  text='P';
end;
else do;
  color='black';
  text='M';
end;
if statindex=1 then do;
  function='symbol';
  x=newxvar;
  y=&yvar;
  size=1;
  if dayvar ne 3 then style='marker ';
  else style='markere';
  if y ne . then output;
end;
if statindex=1 then do;
  function='move';
  size=1;
  line=1;
  x=newxvar;
  y=&yvar;
  output;
end;
if statindex in (2, 3) then do;
  function='draw';
  size=1;
  line=1;
  x=newxvar;
  y=&yvar;
  output;
  x=newxvar-(jitinc/3);
  output;
  x=newxvar+(jitinc/3);
  output;
  x=newxvar;
  output;
end;
if statindex=4 then do;
  function='draw';
  size=1;
  line=1;
  x=newxvar;
  y=&yvar;
  output;
end;
run;

/* Set up treatment text to go in header */
%if &panelvar^= %then %do;
  data trtdset;
  set thissubj;
  length tempvar $ 256;
  tempvar=trim(left(label(&lasttrt)))||'='||trim(left(put(treatvar,?? treatvariable)));
  call symput('trttext',tempvar);
run;
```

Easy-to-use Macros to Create a Line Plot with Error Bars Including Functions of Automated Jittering and Adjusting Axis Ranges, continued

```

%end;

/* Create dummy data to go in header box so no graph is formed */
data thissubjtemp;
  set thissubj (drop=newxvar);
  newxvar=90;
run;

axis3 order=(-99 to -90 by 1);

/* Create individual subject graph, suppressing the legend */
/* Suppress symbol shapes since they are done in the legend and mins and maxs should not
have shapes */
title ' ';
symbol1 i=join h=1.5 color=red v=none;
symbol2 line=1 i=join color=green v=none;
symbol3 line=1 i=join color=blue v=none;
symbol4 line=1 i=join color=black v=none;
symbol5 line=1 i=join color=orange v=none;
symbol6 line=1 i=join color=brown v=none;
symbol7 v=none line=1 i=join color=cyan;
symbol8 v=none line=1 i=join color=magenta;
symbol9 v=none line=1 i=join color=purple;
symbol10 v=none line=1 i=join color=yellow;
proc gplot data=thissubj gout=grph_cat;
  format newxvar xvariable. dayvar dayvariable. newxvar xvariable.;
  plot &yvar*newxvar=dayvar/ name="Subj&pagecnt.a&modfour" haxis=axis1 vaxis=axis2
anno=anno nolegend;
run;
quit;

/* Add the symbol shapes to the symbol statements from the anno dataset so they appear in
the legend. */
title ' ';
symbol1 i=join h=1.5 value=C f=marker color=red;
symbol2 value=U f=marker line=1 i=join color=green;
symbol3 value=U f=markere line=1 i=join color=blue;
symbol4 value=A f=marker line=1 i=join color=black;
symbol5 value=V f=marker line=1 i=join color=orange;
symbol6 value=Y f=marker line=1 i=join color=brown;
symbol7 value=D f=marker line=1 i=join color=cyan;
symbol8 value=N f=marker line=1 i=join color=magenta;
symbol9 value=O f=marker line=1 i=join color=purple;
symbol10 value=P f=marker line=1 i=join color=yellow;

data annoby;
  set thissubj;
  if _n_=1;
  length text $ 200 color $ 10 function $ 8 style $ 20;
  function='label';
  xsys='3'; ysys='3';
  style="'Times New Roman';
  x=50;
  y=94;
  color='black';
  %if &lastby^= %then %do;
    text=trim(left(put(bycntvar,?? byvariable.)));

```

Easy-to-use Macros to Create a Line Plot with Error Bars Including Functions of Automated Jittering and Adjusting Axis Ranges, continued

```
        %end;
        size=1.5;
        output;
        run;

        /* Create header box with dummy data, title and header information, as well as the legend
*/
proc gplot data=thissubjtemp gout=grph_cat;
    title1 j=c h=2.05 font="Times New Roman" "%unquote(&title)";
    format dayvar dayvariable.;
    plot &yvar*newxvar=dayvar/ name="leg&pagecnt.a&modfour" haxis=axis3 vaxis=axis2 noframe
noaxes anno=annoby
        legend=legend1;
run;
quit;

%end;

/* If all subject in treatment are through, then go through and create empty graphs for
remaining
graphs on the current page */
%if "&modfour" ^= "&graphspage" %then %do;
    %do %until("&modfour" = "&graphspage");

        proc gslide gout=grph_cat name="Subj&pagecnt.a&modfour";
            title1 j=c h=1.05 font=swiss " ";
            run;
            quit;

            %let modfour=%eval(&modfour+1);
        %end;
    %end;

%end; /* by-variable processing */

/* Create output file and replay graphs into file */
ods pdf file="%unquote(&outputpathfilename)";
%local m n;
%do m=1 %to &pagecnt;
    %if "&graphspage" = "4" %then %do;
        proc greplay igout=grph_cat gout=grph_cat tc=grph_cat nofs ;
            template=newtemp ;
            treplay 1:Subj&m.a1 2:Subj&m.a2 3:Subj&m.a3 4:Subj&m.a4 5:leg&m.a1;
            run;
            quit;
        %end;
    %if "&graphspage" = "2" %then %do;
        proc greplay igout=grph_cat gout=grph_cat tc=grph_cat nofs ;
            template=newtemp ;
            treplay 1:Subj&m.a1 2:Subj&m.a2 3:leg&m.a1;
            run;
            quit;
        %end;
    %if "&graphspage" = "1" %then %do;
        proc greplay igout=grph_cat gout=grph_cat tc=grph_cat nofs ;
            template=newtemp ;
            treplay 1:Subj&m.a1 2:leg&m.a1;
```

Easy-to-use Macros to Create a Line Plot with Error Bars Including Functions of Automated Jittering and Adjusting Axis Ranges, continued

```
run;
quit;
%end;
%end;
ods pdf close;
/* Delete graphs from template so nothing is carried over when program is re-run */
%do n=1 %to &pagecnt;
  %if "&graphspage"="4" %then %do;
    proc greplay igout=grph_cat nofs;
      delete Subj&n.a1 Subj&n.a2 Subj&n.a3 Subj&n.a4 leg&n.a&modfour;
    quit;
  %end;
  %if "&graphspage"="2" %then %do;
    proc greplay igout=grph_cat nofs;
      delete Subj&n.a1 Subj&n.a2 leg&n.a&modfour;
    quit;
  %end;
  %if "&graphspage"="1" %then %do;
    proc greplay igout=grph_cat nofs;
      delete Subj&n.a1 eg&n.a&modfour;
    quit;
  %end;
%end;

%end;

/* If treatment variable parameter is not specified ... */
%if &panelvar= %then %do;

  %local i subjtrt pagecnt;

  /* Initialize page count to 0 */
  %let pagecnt=0;
  %let pagecnt=%eval(&pagecnt+1);

  /* Allow program to run the same whether or not there are by-variables */
  %if &pagevar= %then %do;

    %let bycnt=1;
    %let pagevar=byvar_dummyvariable;
    %let firstby=byvar_dummyvariable;

    data workpdmean;
      set workpdmean;
      byvar_dummyvariable=1;
    run;

  %end;

proc sort data=workpdmean;
  by &pagevar;
run;

data workpdmean;
  set workpdmean;
  by &pagevar;
  retain bycntvar;
```

Easy-to-use Macros to Create a Line Plot with Error Bars Including Functions of Automated Jittering and Adjusting Axis Ranges, continued

```

    if _n_=1 then bycntvar=0;
    if first.&firstby then bycntvar+1;
run;

/* Initialize number of graphs on current page to 0 */
%local modfour;
%let modfour=0;

%do a=1 %to &bycnt;
    %if &a^=1 %then %let pagecnt=%eval(&pagecnt+1);

    data workpdmeanby;
        set workpdmean;
        if bycntvar=&a;
    run;

    data thistrt;
        set workpdmeanby;
    run;

    %axisrange(dsn=workpdmeanby,
                yvar=&yvar);

axis2
    %if "%unquote(&yaxislabel)"^=" " %then %do;
        label=(h=1.0 cm angle=90 rotate=0 font="Times New Roman" "%unquote(&yaxislabel) ")
    %end;
    %else %do;
        label=(h=0.5 cm angle=90 rotate=0 " ")
    %end;
    %if &ymin^= and &yint= %then %do;
        order=(&ymin to &ymax)
    %end;
    %else %if &ymin^= and &yint^= %then %do;
        order=(&ymin to &ymax by &yint)
    %end;
    %else %if &yaxisfreeyn=N %then %do;
        order=(&axisymin to &axisymax by &axisyint)
    %end;
    offset=(0.15 in,0 in)
    minor=none
    value=(h=0.65 cm font="Times New Roman")
    ;

/* Increase the graph on current page count by 1 */
%let modfour=%eval(&modfour+1);
/* Subset for only current subject */
data thissubj;
    set thistrt;
run;

/* If the old page is full, increase page count by 1 and reset graphs on current page to 1
*/
%if "&graphspage"="4" %then %do;
    %if "&modfour"="5" %then %do;
        %let pagecnt=%eval(&pagecnt+1);
        %let modfour=1;

```



Easy-to-use Macros to Create a Line Plot with Error Bars Including Functions of Automated Jittering and Adjusting Axis Ranges, continued

```
%end;
%end;
%if "&graphspage"="2" %then %do;
  %if "&modfour"="3" %then %do;
    %let pagecnt=%eval(&pagecnt+1);
    %let modfour=1;
  %end;
%end;
%if "&graphspage"="1" %then %do;
  %if "&modfour"="2" %then %do;
    %let pagecnt=%eval(&pagecnt+1);
    %let modfour=1;
  %end;
%end;

proc sort data=thissubj;
  by dayvar newxvar statindex;
run;

/* Set up annotate dataset. Must add error bars and caps and symbols for the mean/median */
data anno;
  set thissubj;
  format newxvar xvariable. dayvar dayvariable. treatvar treatvariable.;
  length text $ 200 color $ 10 style $ 20;
  jitinc=&jitinc;
  xmin="&xmin";
  xmax="&xmax";
  xsys='2';
  ysys='2';
  xmin1=input(xmin,?? best.);
  xmax1=input(xmax,?? best.);
  if xmin1 ne . and xmax1 ne . and (newxvar lt xmin1 or newxvar gt xmax1) then delete;
  if dayvar=1 then do;
    color='red';
    text='C';
  end;
  else if dayvar=2 then do;
    color='green';
    text='U';
  end;
  else if dayvar=3 then do;
    color='blue';
    text='U';
  end;
  else if dayvar=4 then do;
    color='black';
    text='A';
  end;
  else if dayvar=5 then do;
    color='orange';
    text='V';
  end;
  else if dayvar=6 then do;
    color='brown';
    text='Y';
  end;
  else if dayvar=7 then do;
```

Easy-to-use Macros to Create a Line Plot with Error Bars Including Functions of Automated Jittering and Adjusting Axis Ranges, continued

```
        color='cyan';
        text='D';
    end;
else if dayvar=8 then do;
    color='magenta';
    text='N';
end;
else if dayvar=9 then do;
    color='purple';
    text='O';
end;
else if dayvar=10 then do;
    color='yellow';
    text='P';
end;
else do;
    color='black';
    text='M';
end;
if statindex=1 then do;
    function='symbol';
    x=newxvar;
    y=&yvar;
    size=1;
    if dayvar ne 3 then style='marker ';
    else style='markere';
    if y ne . then output;
end;
if statindex=1 then do;
    function='move';
    size=1;
    line=1;
    x=newxvar;
    y=&yvar;
    output;
end;
if statindex in (2, 3) then do;
    function='draw';
    size=1;
    line=1;
    x=newxvar;
    y=&yvar;
    output;
    x=newxvar-(jitinc/3);
    output;
    x=newxvar+(jitinc/3);
    output;
    x=newxvar;
    output;
end;
if statindex=4 then do;
    function='draw';
    size=1;
    line=1;
    x=newxvar;
    y=&yvar;
    output;
```

Easy-to-use Macros to Create a Line Plot with Error Bars Including Functions of Automated Jittering and Adjusting Axis Ranges, continued

```

end;
run;

/* Set up treatment text to go in header */
%if &panelvar^= %then %do;
  data trtdset;
    set thissubj;
    length tempvar $ 256;
    tempvar=trim(left(label(&lasttrt)))||'='||trim(left(put(treatvar,?? treatvariable.)));
    call symput('trttext',tempvar);
  run;
%end;

/* Create dummy data to go in header box so no graph is formed */
data thissubjtemp;
  set thissubj (drop=newxvar);
  newxvar=90;
run;

axis3 order=(-99 to -90 by 1);

/* Create individual subject graph, suppressing the legend */
/* Suppress symbol shapes since they are done in the legend and mins and maxs should not
have shapes */
title ' ';
symbol1 i=join h=1.5 color=red v=none;
symbol2 line=1 i=join color=green v=none;
symbol3 line=1 i=join color=blue v=none;
symbol4 line=1 i=join color=black v=none;
symbol5 line=1 i=join color=orange v=none;
symbol6 line=1 i=join color=brown v=none;
symbol7 v=none line=1 i=join color=cyan;
symbol8 v=none line=1 i=join color=magenta;
symbol9 v=none line=1 i=join color=purple;
symbol10 v=none line=1 i=join color=yellow;
proc gplot data=thissubj gout=grph_cat;
  format newxvar xvariable. dayvar dayvariable. newxvar xvariable.;
  plot &yvar*newxvar=dayvar/ name="Subj&pagecnt.a&modfour" haxis=axis1 vaxis=axis2 anno=anno
nolegend;
run;
quit;

/* Add the symbol shapes to the symbol statements from the anno dataset so they appear in
the legend. */
title ' ';
symbol1 i=join h=1.5 value=C f=marker color=red;
symbol2 value=U f=marker line=1 i=join color=green;
symbol3 value=U f=markere line=1 i=join color=blue;
symbol4 value=A f=marker line=1 i=join color=black;
symbol5 value=V f=marker line=1 i=join color=orange;
symbol6 value=Y f=marker line=1 i=join color=brown;
symbol7 value=D f=marker line=1 i=join color=cyan;
symbol8 value=N f=marker line=1 i=join color=magenta;
symbol9 value=O f=marker line=1 i=join color=purple;
symbol10 value=P f=marker line=1 i=join color=yellow;
data annoby;
  set thissubj;

```

Easy-to-use Macros to Create a Line Plot with Error Bars Including Functions of Automated Jittering and Adjusting Axis Ranges, continued

```

        if _n_=1;
        length text $ 200 color $ 10 function $ 8 style $ 20;
        function='label';
        xsys='3'; ysys='3';
        style="'Times New Roman';
        x=50;
        y=94;
        color='black';
        %if &lastby^= %then %do;
            text=trim(left(put(bycntvar,?? byvariable.)));
        %end;
        size=1.5;
        output;
    run;

    /* Create header box with dummy data, title and header information, as well as the legend */
    proc gplot data=thissubtemp gout=grph_cat;
        title1 j=c h=2.05 font="Times New Roman" "%unquote(&title)";
        format dayvar dayvariable.;
        plot &yvar*newxvar=dayvar/ name="leg&pagecnt.a&modfour" haxis=axis3 vaxis=axis2 noframe
noaxes anno=annoby
            legend=legend1;
    run;
    quit;

    /* If all subject in treatment are through, then go through and create empty graphs for
remaining
graphs on the current page */
    %if "&modfour" ^="&graphspage" %then %do;
        %do %until("&modfour"="&graphspage");

            proc gslide gout=grph_cat name="Subj&pagecnt.a&modfour";
                title1 j=c h=1.05 font=swiss " ";
            run;
            quit;

            %let modfour=%eval(&modfour+1);
        %end;
    %end;

%end; /* End of by-processing */

/* Create output file and replay graphs into file */
ods pdf file="%unquote(&outputpathfilename)";
%local m n;
%do m=1 %to &pagecnt;
    %if "&graphspage"="4" %then %do;
        proc greplay igout=grph_cat gout=grph_cat tc=grph_cat nofs ;
            template=newtemp ;
            treplay 1:Subj&m.a1 2:Subj&m.a2 3:Subj&m.a3 4:Subj&m.a4 5:leg&m.a1;
        run;
        quit;
    %end;
    %if "&graphspage"="2" %then %do;
        proc greplay igout=grph_cat gout=grph_cat tc=grph_cat nofs ;
            template=newtemp ;

```

Easy-to-use Macros to Create a Line Plot with Error Bars Including Functions of Automated Jittering and Adjusting Axis Ranges, continued

```
        treplay 1:Subj&m.a1 2:Subj&m.a2 3:leg&m.a1;
    run;
    quit;
%end;
%if "&graphspage"="1" %then %do;
    proc greplay igout=grph_cat gout=grph_cat tc=grph_cat nofs ;
        template=newtemp ;
        treplay 1:Subj&m.a1 2:leg&m.a1;
    run;
    quit;
%end;
%end;
ods pdf close;

/* Delete graphs from template so nothing is carried over when program is re-run */
%do n=1 %to &pagecnt;
    %if "&graphspage"="4" %then %do;
        proc greplay igout=grph_cat nofs;
            delete Subj&n.a1 Subj&n.a2 Subj&n.a3 Subj&n.a4 leg&n.a&modfour;
        quit;
    %end;
    %if "&graphspage"="2" %then %do;
        proc greplay igout=grph_cat nofs;
            delete Subj&n.a1 Subj&n.a2 leg&n.a&modfour;
        quit;
    %end;
    %if "&graphspage"="1" %then %do;
        proc greplay igout=grph_cat nofs;
            delete Subj&n.a1 eg&n.a&modfour;
        quit;
    %end;
%end;

%end;

%mend LinePlotMeanMedian;
```

### SAS CODE OF %jitterx:

```
%macro jitterx(dsn=,          /* Input Dataset */
               outdsn=,      /* Output Dataset */
               xvar=,        /* X variable (must be numeric) */
               trtvar=);    /* line by variable (can be numeric or character) */

/* This macro will take an input dataset with an x variable and a treatment variable and
   based on the increments between x's and the number of lines, will create a new x variable
   (called jitterx) that will be the jittered form of the original x variable */

/* Find differences in x values of data */
proc sort data=&dsn out=xdiff (keep=&xvar) nodupkey;
    by &xvar;
run;

data xdiff;
    set xdiff end=eof;
    retain prevx;
```

Easy-to-use Macros to Create a Line Plot with Error Bars Including Functions of Automated Jittering and Adjusting Axis Ranges, continued

```
    if _n_=1 and eof then diff=0;
    else diff=&xvar-prevx;
    prevx=&xvar;
run;

/* Find smallest difference in x values in the data */
proc univariate data=xdiff noprint;
    var diff;
    output out=jit min=jitmin;
run;

/* Count the treatments */
proc sort data=&dsn out=trtdsn (keep=&trtvar) nodupkey;
    by &trtvar;
run;

data trtdsn;
    set trtdsn;
    cnttrtvar=1;
run;

proc univariate data=trtdsn noprint;
    var cnttrtvar;
    output out=cnttrt n=trtcnt;
run;

%local trtcnt;

data cnttrt;
    set cnttrt;
    call symput('trtcnt',trtcnt);
run;

/* Jitrng is the maximum spread for jittering (either .35 or .5) */
/* If there are 0 treatments or 1 treatment, then no jittering needed */
/* Jittering increments are based on the jit range and the number of treatments being jittered */
%local jitrng;
%global jitinc;

data jit;
    set jit;
    trtcnt=&trtcnt;
    if trtcnt lt 4 then jitrng=jitmin*.35;
    else jitrng=jitmin*.5;
    if trtcnt in (0, 1) then do;
        jitrng=0;
        jitinc=0;
    end;
    else do;
        jitinc=jitrng/(trtcnt-1);
    end;
    call symput('jitrng',jitrng);
    call symput('jitinc',jitinc);
run;

proc sort data=&dsn out=&outdsn;
    by &trtvar;
```

## Easy-to-use Macros to Create a Line Plot with Error Bars Including Functions of Automated Jittering and Adjusting Axis Ranges, continued

```
run;

/* Based on jittering range and the increment, start at first profile treatment and increment x
values
  for each new profile day */
data &outdsn (drop=jitterx_jitrng jitterx_jitinc jitterx_currjit);
  set &outdsn;
  by &trtvar;
  retain jitterx_currjit;
  jitterx_jitrng=&jitrng;
  jitterx_jitinc=&jitinc;
  if _n_=1 then do;
    jitterx_currjit=(jitterx_jitrng/2)*-1;
  end;
  if first.&trtvar and not (_n_=1) then do;
    jitterx_currjit=jitterx_currjit+jitterx_jitinc;
  end;
  jitterx=&xvar+jitterx_currjit;
run;

%mend;
```

### SAS CODE OF %axisrange

```
%macro axisrange(dsn=, /* Input Dataset */
                 yvar=, /* Y variable (must be numeric) */
                 low=, /* Normal Range Low variable or numeric constant to be included in range
                        interval */
                 hi= /* Normal Range High variable or numeric constant to be included in
                      range interval */);

/* This macro takes a numeric variable from an input dataset and calculates valid
axis ranges and outputs them in the form of global macro variables
axisymin, axisymax, and axisyint.
The user can optionally have a low range and/or high range variable or constant that will
be ensured that the range encompasses it. */

data _axisrange;
  set &dsn;
run;

%if "&low" ^= "" %then %do;
  data _axisrange;
    set _axisrange;
    output;
    &yvar=&low;
    output;
  run;
%end;

%if "&hi" ^= "" %then %do;
  data _axisrange;
    set _axisrange;
    output;
    &yvar=&hi;
    output;
  run;
%end;
```

Easy-to-use Macros to Create a Line Plot with Error Bars Including Functions of Automated Jittering and Adjusting Axis Ranges, continued

```
/* Get min and max of variable from dataset */
proc univariate data=_axisrange noprint;
  var &yvar;
  output out=axisrange min=min max=max;
run;

/* Determine from the range of values what the rounding value should be.
   If min value is less than 20% of the max value, then set min to 0.
   Do likewise for max for all negative values. */
data axisrange;
  set axisrange;
  range=max-min;
  if range=0 then range=abs(max);
  if range=0 then range=1;
  /* Determine which factor of 10 to round to */
  if range lt 1.7*(10**floor(log10(range))) then round=(10**floor(log10(range)-1));
  else round=(10**floor(log10(range)));
  if round=. then do;
    finalmin=min;
    finalmax=max;
  end;
  else do;
    finalmin=round(min,round);
    if finalmin gt min then finalmin=finalmin-round;
    finalmax=round(max,round);
    if finalmax lt max then finalmax=finalmax+round;
  end;
  if min ge 0 and max ge 0 and min lt max*.2 then finalmin=0;
  if min le 0 and max le 0 and max gt min*.2 then finalmax=0;
  /* Calculate Interval - range between 1 and 18 */
  finalrange=finalmax-finalmin;
  if round((finalrange/round), 1)=1 then round=round/10;
  if round((finalrange/round), 1)=1 then interval=.5*round;
  if round((finalrange/round), 1)=2 then interval=.5*round;
  if round((finalrange/round), 1)=3 then interval=.5*round;
  if round((finalrange/round), 1)=4 then interval=.5*round;
  if round((finalrange/round), 1)=5 then interval=1*round;
  if round((finalrange/round), 1)=6 then interval=1*round;
  if round((finalrange/round), 1)=7 then interval=1*round;
  if round((finalrange/round), 1)=8 then interval=1*round;
  if round((finalrange/round), 1)=9 then interval=1*round;
  if round((finalrange/round), 1)=10 then interval=1*round;
  if round((finalrange/round), 1)=11 then interval=1*round;
  if round((finalrange/round), 1)=12 then interval=2*round;
  if round((finalrange/round), 1)=13 then interval=1*round;
  if round((finalrange/round), 1)=14 then interval=2*round;
  if round((finalrange/round), 1)=15 then interval=3*round;
  if round((finalrange/round), 1)=16 then interval=4*round;
  if round((finalrange/round), 1)=17 then interval=1*round;
  if round((finalrange/round), 1)=18 then interval=3*round;
  if round((finalrange/round), 1)=19 then interval=1*round;
  if round((finalrange/round), 1)=20 then interval=4*round;
  if round((finalrange/round), 1)=21 then interval=3*round;
  if round((finalrange/round), 1)=22 then interval=2*round;
run;
```



Easy-to-use Macros to Create a Line Plot with Error Bars Including Functions of Automated Jittering and Adjusting Axis Ranges, continued

```
/* Set global macro variables axisymin and axisymax to axis range values */
%global axisymin axisymax axisyint;

%let axisymin=.;
%let axisymax=.;

data axisrange;
  set axisrange;
  call symput('axisymin',finalmin);
  call symput('axisymax',finalmax);
  call symput('axisyint',interval);
run;

%mend;
```