PharmaSUG 2020 - Paper DV-283

Programming Technique for Line Plots with Superimposed Data Points

Chandana Sudini, Merck & Co., Inc, Rahway, NJ, USA
Bindya Vaswani, Merck & Co., Inc, Rahway, NJ, USA

ABSTRACT

Line graphs are mainly plotted by connecting associated data points over a specified time interval to portray an overall trend. Despite the existence of many visualization methods and techniques, line plots continue to be a simple way of displaying quantitative data patterns in exploratory analyses.

Line plots can be used to present either summary statistics such as mean, standard deviation of a population or individual subject level data over a time period. In a scenario where we need to understand the impact of concomitant medication (CM) on the laboratory measurements (LB) for each subject, presenting line plots with subject level data from those two sources can be extremely challenging, since the y-axis value at the time of CM administration may be unknown.

In order to accomplish this, we propose using the properties of a straight line to predict the potential y-value at the time of the CM administration. We can derive these predicted values by programmatically fitting a coordinate between the LB data points, before and after the CM administration (using mathematical concept of slope and constant of straight line). Once the predicted values are calculated, plugging those values into the annotation data step to generate line plots with superimposed data points, resulting in a meaningful representation of the data being analyzed. This paper details the SAS logic required to generate these line plots.

INTRODUCTION WITH BACKGROUND:

A line graph is a type of graph which displays information as a set of line segments connecting adjacent data points. It is like a scatter plot except that the measurement points are ordered by the x-axis values and joined with straight line segments.

Line plots can be used to present summary statistics of data, such as, Safety or Patient Reported Outcome (ePRO) etc. They can also be used to present Individual subject level data over a time period in early stage trials or when analyzing a small subset of study population.

The analysis of interest is generally performed using one type of clinical data source, such as Labs or Adverse events. Within a single line graph, if there is a need to analyze the impact of one clinical data point on another, for example, effect of concomitant medications (CM) on subsequent laboratory assessments (LB), it would be difficult to plot data from two different sources on the same graph. As a possible solution, we propose to superimpose CM data points over LB line graph. If the CM and LB values are collected during the same scheduled visits, then we can plot CM data by using study day for x-axis and retaining the same LB values for the y-axis.

However, CM may not be necessarily administered during the scheduled LB visits. In that case, although CM study day can be used for the x-axis, we do not have corresponding y-axis values to present this data point on the LB line graph. Using mathematical concept of slope and constant of straight line, we can derive the y-axis values by programmatically fitting a coordinate between the LB data points, before and after the CM administration.

BASIC LINE GRAPH:

PROC SGPLOT statements can be used to create a basic line graph to present lab data by subject. We have considered following variables from hypothetical LB data, as input to generate the proposed graph.

Input LB data for a sample of 3 subjects, with LBDY derived from the collected LB Date and LBVAL being the collected LB Result:

```
data lb;
   input usubjid $ lbdy lbval;
   datalines;
1      5     8
1      10    14
1      15    17
1      20    18
1      25    18
2      5     5
2      10    4
2      15    12
2      20    13
2      25    14
3      5     15
3      10    20
3      15    22
3      20    24
3      25    26
;
```

SAS Code and resultant plot for the Basic Lab Line Graph:

```
data lb1;
      set lb;
      x=lbdy;
      y=lbval;
      byval=lbdy;
run;

ods graphics on/reset=all width=20in height=8.5in imagemap=on border=off
SUBPIXEL;

title1 h=14pt "Basic Line Graph with Lab Data";
proc sgplot data=lb pad=(bottom=30) NOAUTOLEGEND;
      series X=X Y=Y / curvelabel CURVELABELATTRS= (Color=Green
      Family="Arial" Size=12
      Style=Italic Weight=Bold) group=USUBJID
            lineattrs= (color=grey pattern=solid thickness=2)
            markers markerattrs= (size=15 symbol=circle color=blue)
            tip=(USUBJID);
      xaxis values= (2 to 30 by 2) minor MINORCOUNT=5 min=0 max=5500
      offsetmax=0.05 label="Lab Value" LABELATTRS= (Family=Arial Size=14pt
      Style=Italic Weight=Bold) valueattrs=(size=12pt weight=bold);
      yaxis values= (0 to 30 by 5) fitpolicy=none
                  label="Study Day" LABELATTRS= (Family=Arial Size=12pt
                  Style=Italic Weight=Bold) offsetmin=0.05
                  minor valueattrs= (size=10pt weight=bold) display=all;
run;
```

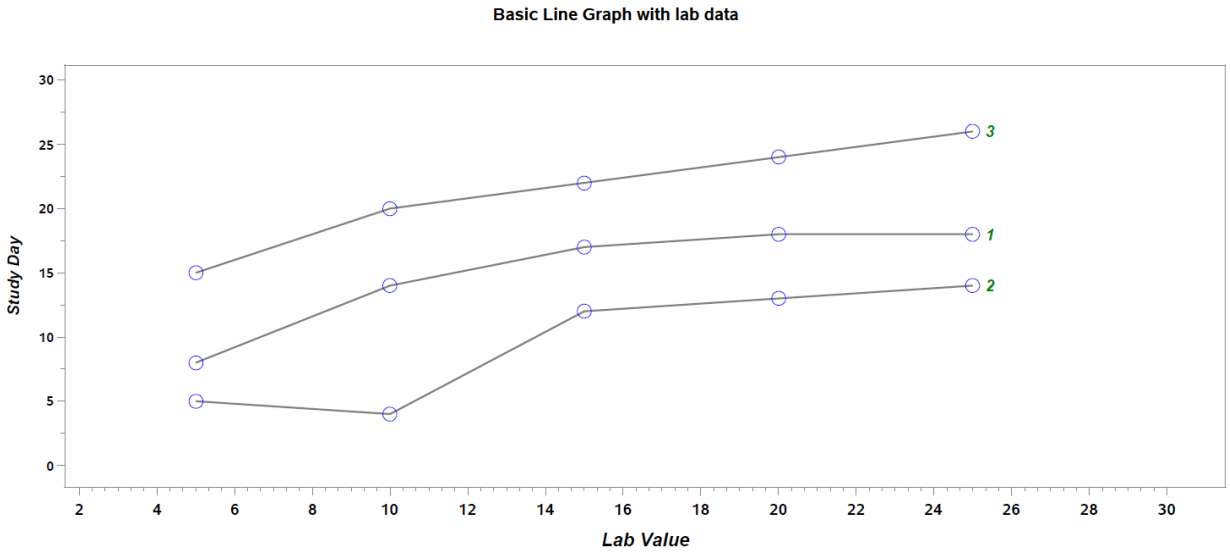


Figure 1: Basic Line Graph with Lab data

Figure 1 displays lab values collected during scheduled visits for a sample of 3 subjects.
As the next step, we would like to see the impact of administered CM on the same line graph. Our objective is to programmatically apply the mathematical concepts outlined below to calculate the missing y-axis values for the CM data points.

MATHEMATICAL CONCEPTS AND PROPOSED LOGIC:

A line graph displays information as a set of straight-line segments connected by adjacent (x, y) intercepts.
The equation of Straight line:

y = mx + c
(x, y) is the coordinate point
c is the constant
m is the slope of the straight line

The slope or gradient of a line is a numerical representation of the relationship between the x and y variables. The slope between the coordinates/intercepts indicates whether the trend is about to decrease or increase.

If we have the values of adjacent coordinate points of a straight line, we can derive the slope of that line using the formula:

m = *change in y* /change in x = (*y2 − y1)*/ (x2 − x1)

The next step would be to plug in the slope and any one set of coordinate points (x1, y1) or (x2, y2) into the straight-line equation to derive the constant of that line.

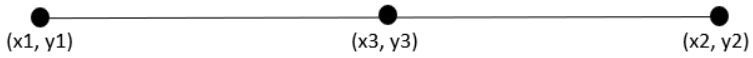
y1=m*x1+c ➔ c=y1-m*x1

Once we have the slope, constant and value for one of the coordinate points, by using the straight-line equation again, we can easily calculate the value of the other missing coordinate.

y3=m*x3+c

This algorithm will enable us to predict the potential y-value at the time of the CM administration in our example being presented in this paper.

Assuming (x1, y1), (x2, y2) are adjacent LB coordinates, and x3 is a known CM data point, y3 can be derived and used for the CM coordinate (x3, y3).



PROGRAMMING LOGIC TO GENERATE SUPERIMPOSED LINE GRAPH:

LB data has already been described in the basic plot section. Further, we have considered following variables from hypothetical CM data, as added input to generate the proposed graph.

Input CM data for the same sample of 3 subjects, with CMSTDY derived from the collected CM Start Date:

```
data cm;
   input usubjid $ cmstdy;
   datalines;
1      7
1      17
2      12
2      22
3      17
3      23
;
```

SAS data steps:

1. Assign LB study day and result values into (x, y) variables.

```
data lb1;
   set lb;
   x=lbdy;
   y=lbval;
   byval=lbdy;
run;
```

2. Assign CM study day into x3 variable, assign y3 as missing, since it is not available at this point.

```
data cm1;
   set cm(rename=(cmstdy=cmdy));
   x3=cmdy;
   y3=.;
   byval=cmdy;
run;
```

3. Combine LB and CM data vertically by subject and study day.

```
data final;
   set lb1(in=a) cm1(in=b);
   by usubjid byval;
   if a then dat="LB";
   else dat="CM";
run;
```

4. Sort the data by subject and ascending order of study day, retain (x1, y1) values of the LB data points, as the initial coordinate that forms the straight line.

```
proc sort data=final;
   by usubjid byval;
run;

data final1;
   set final;
   by usubjid byval;
   retain x1 y1;
   if first.usubjid then do;
         x1=.;
         y1=.;
   end;
   if dat="LB" then do;
         x1=x;
         y1=y;
   end;
run;
```

5. Sort the data by subject and descending order of study day, retain (x2, y2) values of the LB data points, as the second set of coordinates required to form the straight line.

```
proc sort data=final1;
   by usubjid descending byval;
run;

data final2;
   set final1;
   by usubjid descending byval;
   retain x2 y2;
   if first.usubjid then do;
         x2=.;
         y2=.;
   end;
   if dat="LB" then do;
         x2=x;
         y2=y;
   end;
run;
```

6. Now that we have (x1, y1), (x2, y2) from LB data, we can calculate slope of the line as m = (y2-y1)/(x2-x1).

7. Next compute constant either by substituting (x1, y1) or (x2, y2) into straight line equation c = y1-(m*x1).

8. As we have slope, constant and CM study day as x3, potential y3 value can be calculated as y3 = (m*x3) + c.

```
data final3;
   set final2;
   if dat="CM" then do;
         M = (y2-y1)/(x2-x1);
         C = y1-(m*x1);
         y3 =(m*x3) + c;
   end;
run;
```

```
proc sort data=final3;
   by usubjid byval;
run;
```

The above data steps will yield the following dataset, to be further used as input for the PROC SGPLOT to generate the superimposed line graph.

	usubjid	lbdy	lbval	cmdy	byval	x	y	x1	y1	x2	y2	dat	x3	y3	M	C
1	1	5	8	.	5	5	8	5	8	5	8	LB	.	.	.	.
2	1	.	.	7	7	.	.	5	8	10	14	CM	7	10.4	1.2	2
3	1	10	14	.	10	10	14	10	14	10	14	LB	.	.	.	.
4	1	15	17	.	15	15	17	15	17	15	17	LB	.	.	.	.
5	1	.	.	17	17	.	.	15	17	20	18	CM	17	17.4	0.2	14
6	1	20	18	.	20	20	18	20	18	20	18	LB	.	.	.	.
7	1	25	18	.	25	25	18	25	18	25	18	LB	.	.	.	.
8	2	5	5	.	5	5	5	5	5	5	5	LB	.	.	.	.
9	2	10	4	.	10	10	4	10	4	10	4	LB	.	.	.	.
10	2	.	.	12	12	.	.	10	4	15	12	CM	12	7.2	1.6	-12
11	2	15	12	.	15	15	12	15	12	15	12	LB	.	.	.	.
12	2	20	13	.	20	20	13	20	13	20	13	LB	.	.	.	.
13	2	.	.	22	22	.	.	20	13	25	14	CM	22	13.4	0.2	9
14	2	25	14	.	25	25	14	25	14	25	14	LB	.	.	.	.
15	3	5	15	.	5	5	15	5	15	5	15	LB	.	.	.	.
16	3	10	20	.	10	10	20	10	20	10	20	LB	.	.	.	.
17	3	15	22	.	15	15	22	15	22	15	22	LB	.	.	.	.
18	3	.	.	17	17	.	.	15	22	20	24	CM	17	22.8	0.4	16
19	3	20	24	.	20	20	24	20	24	20	24	LB	.	.	.	.
20	3	.	.	23	23	.	.	20	24	25	26	CM	23	25.2	0.4	16
21	3	25	26	.	25	25	26	25	26	25	26	LB	.	.	.	.

Figure 2: Input data set for the PROC SGPLOT procedure

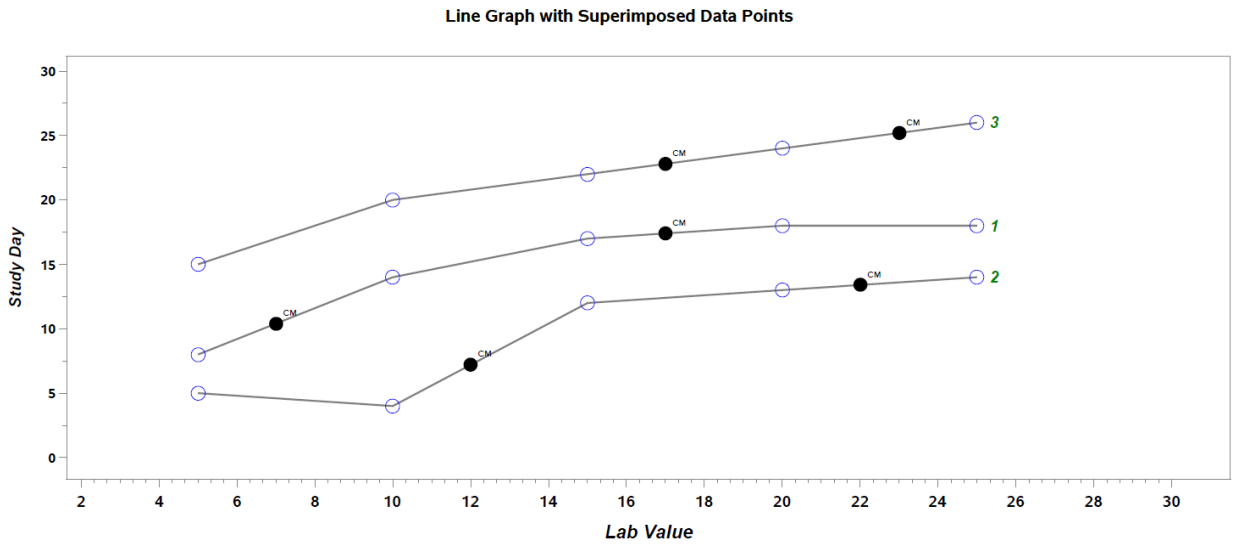


Figure 3: Line Graph with Superimposed Data Points

SAS SGPLOT code used to generate Superimposed Line Graph:

```
ods graphics on/reset=all width=20in height=8.5in imagemap=on border=off
SUBPIXEL;
title1 h=14pt "Line Graph with Superimposed Data Points";
proc sgplot data=final3 pad=(bottom=30) NOAUTOLEGEND;
            series X=X Y=Y / curvelabel CURVELABELATTRS= (Color=Green
            Family="Arial" Size=12 Style=Italic Weight=Bold) group=USUBJID
            lineattrs= (color=grey pattern=solid thickness=2)
            markers markerattrs= (size=15 symbol=circle color=blue)
            tip=(USUBJID);

      xaxis values= (2 to 30 by 2) minor MINORCOUNT=5 min=0 max=5500
      offsetmax=0.05 label="Lab Value" LABELATTRS= (Family=Arial Size=14pt
      Style=Italic Weight=Bold) valueattrs= (size=12pt weight=bold);

      yaxis values= (0 to 30 by 5) fitpolicy=none label="Study Day"
            LABELATTRS= (Family=Arial Size=12pt Style=Italic Weight=Bold)
            offsetmin=0.05 minor valueattrs= (size=10pt weight=bold)
            display=all;
      scatter X=X3 Y=Y3/group=usubjid datalabel=dat markerattrs= (size=15
      symbol=circlefilled color=black);
run;
```

Scatter is the additional SGPLOT statement that plots the CM data points over the LB line graph.

CONCLUSION:

With this approach, using simple mathematical concepts we were able to generate line plots with superimposed data points, resulting in a meaningful representation of the data being analyzed. Although, we have used LB and CM to demonstrate the programming technique in this paper, this methodology can be implemented for any pair of clinical data points.

ACKNOWLEDGMENTS

The authors would like to thank their Director Jeff Xia for his support and suggestions on this paper.

RECOMMENDED READING

- Base SAS Procedures
- SAS ODS Graphical Procedures

CONTACT INFORMATION

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

Chandana Sudini
Merck &Co., Inc.
Rahway, NJ-07065
Email: chandana.sudini@merck.com

Bindya Vaswani
Merck &Co., Inc.
Rahway, NJ-07065
Email: bindya.vaswani@merck.com