PharmaSUG 2018 – Paper AA-02

Patient-Level Longitudinal Analysis Plots Showing Adverse Event Timelines and Dose Titration Levels

John R Gerlach, Dataceutics, Inc., Pottstown, PA USA
Keith Brown, Dataceutics, Inc., Pottstown, PA USA

ABSTRACT

The analyses in a clinical trial often include subject-specific longitudinal plots that depict, for example, a lab measurement taken at various times during a study period. Because the study involves dose titration levels, the plots might include vertical lines indicating dose levels, which can increase and decrease during the study period. Perhaps the analysis includes another component: the timelines of adverse events, if any. Both the vertical reference lines and horizontal time lines are superimposed on the longitudinal plot. Obviously, the task becomes more intricate since each plot is subject-specific, having different measurements, dose titration levels, and adverse events. This paper explains how to produce this intricate graphical analysis.

INTRODUCTION

Imagine a longitudinal plot consisting of an analysis variable measured across time, such as a laboratory measurement taken on scheduled visits (i.e. study days) during a clinical trial, each plot representing a single subject in the study. A plot shows a locus that represents how a variable of interest progresses over time, along the Y and X axes, respectively. There would be N-plots each identified by the Subject Identifier in the subtitle. Thus far, the intended deliverable would be rather easy to produce using SAS/Graph® and the Macro Language.

The task becomes more difficult when installing vertical reference lines that represent the dose levels, which can increase or decrease during the study period for a given subject. Even more demanding, however, the graphical analysis includes the timelines of the subject's adverse events. Although it is likely that a subject will have at least one adverse event, the proposed SAS® solution should accommodate the possibility that the subject has none. This paper explains two techniques for creating such plots: using the GPLOT procedure with the Annotate Facility; and using the SGPLOT procedure without any annotate component.

DELIVERABLE

Observe the longitudinal plot in Figure 1. The plot depicts lab measurements of a single subject during a period of twenty-four weeks. However, besides the analysis variable of interest, the focus of the clinical analysis concerns the superimposed dose titration levels and adverse events, if any, during that period. Consequently, the desired plot would include vertical reference lines indicating the dose level, colored and labeled accordingly, as well as horizontal time lines, in tandem with the timely lab measurements, taken during scheduled visits.

Because the plot is subject-specific, each plot can look extremely different. Although the curve of the plot line might increase over time, an anticipated result for this hypothetical study, the dose titration levels and the adverse events can be extremely different for each subject. For instance, a dose level might be lowered due to an adverse event, which increased later during the study period. Also, a subject might have a dozen adverse events, whatever they might be, or no adverse events at all. Another feature of the analysis concerns the order of the adverse events, from first occurrence in a crescendo fashion. Finally, if the adverse event does not have an end date, it is assumed to be ongoing, thus, the time line continues to the end of the study period, as shown in Figure 1.

The plot contains two Y-axes: one for the adverse events and the other for the lab measurements. For this discussion, the range limit of the lab measurements is fixed, from 0 to 100. However, the Y-axis denoting the adverse events requires more work, obviously, which is discussed later in further detail. The X-axis range starts at week 0 and ends at week 24. The FORMAT procedure below creates the NWK format that supplants the value of zero for 'BL' to denote the Baseline visit.

```
proc format;
   value nwk 0 = 'BL';
run;
```

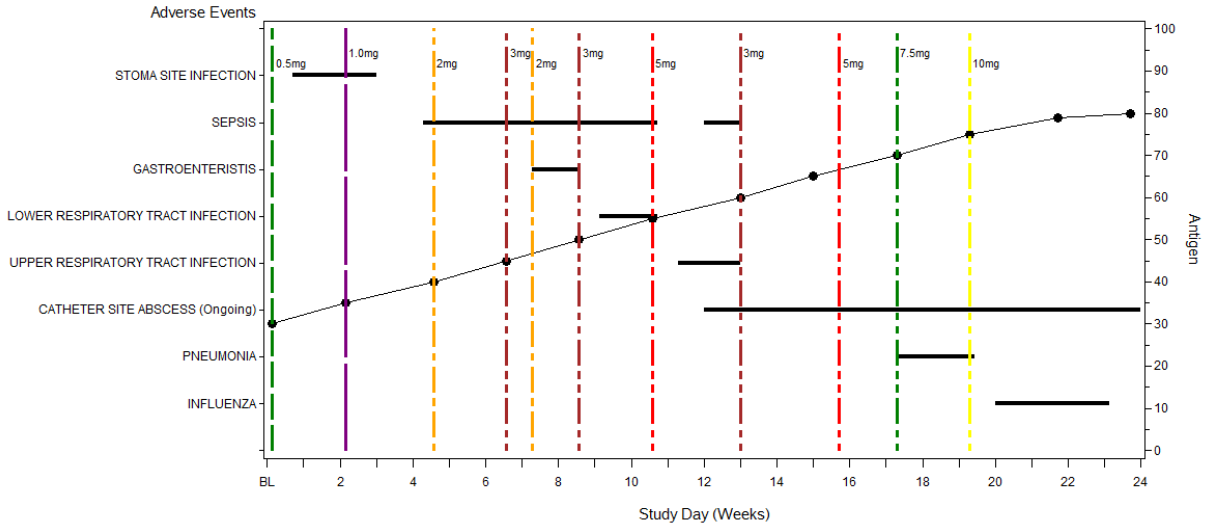

Figure 1. Lab measurements for Subject 1001 during a 24-week Study Period, showing Drug Titration Levels and Adverse Events in order of first occurrence.

DATA MART

The analysis considers only the Intent-to-Treat (ITT) Population, denoted by the variable ITTFL, which is found in the several ADaM data sets, listed below, indicating the unit of analysis and the required variables. Because it is possible for a subject to have no adverse events, the ADSL data set ensures the inclusion of those subjects not found in ADAE. Each subject will have lab measurements and exposure data taken on scheduled visits.

- ADSL (Subject Info) *One record per subject.*
  USUBJID ITTFL
- ADLB (Lab Tests) *One record per lab measurement per scheduled visit date per subject.*
  USUBJID ITTFL PARAMCD AVISITN LBDY AVAL
- ADEX (Exposure) *One record per actual dose per subject.*
  USUBJID ITTFL TRTA ASTDT
- ADAE (Adverse Events) *One record per event per subject.*
  USUBJID ITTFL AEBODSYS AEDECOD TRTEMFL ASTDY

LAB MEASUREMENTS AND ADVERSE EVENTS

The WHERE statement in the SORT step specifies the selection criteria: ITT population, a specific lab test, and scheduled visits from 1 through 12. Notice how the WHERE statement discerns a scheduled visit by comparing the value of the INT function on AVISITN with the actual value; that is, if the integral value is different, such as 2.0 versus 2.1, then it must be an unscheduled visit.

```
proc sort data=adam_adlb(keep=usubjid ittfl paramcd avisitn lbdy aval)
   out=lb01(keep=usubjid lbdy aval);
   by usubjid lbdy;
   where ittfl eq 'Y' and paramcd eq 'ANITGEN'
      and avisitn between 1 and 12 and int(avisitn) eq avisitn
      and lbdy is not null and aval is not null;
run;
```

After obtaining the lab data, the DATA step below computes the study week denoting the X-axis. Table 1 shows the final lab data set that contributes to the graphical analysis, listing the study day and respective study week, along with the analysis variable, the variables LBDY, STUDYWK, and AVAL, respectively.

```
data lb02;
   set lb01;
   studywk = lbdy / 7;
run;
```

Subject 1001			**Subject 1002**		
LBDY	STUDYWK	AVAL	LBDY	STUDYWK	AVAL
1	1	30	1	1	60
15	2	35	15	2	50
32	4	40	29	4	45
46	6	45	43	6	35
60	8	50	60	8	35
74	10	55	74	10	55
91	13	60	90	12	60
105	15	65	104	14	80
121	17	70	121	17	82
135	19	75	135	19	75
152	21	79	151	21	79
170	24	80	171	24	80

Table 1. Listing of Lab data for two subjects.

Processing the adverse events requires a lot more effort for several obvious reasons, such as a subject having no adverse events or, perhaps, having an event that is “Ongoing.” For the purpose of this discussion, subjects have at most a dozen adverse events. Once again, the WHERE statement in the SORT step implements the selection criteria, namely a Treatment Emergent adverse event that has a start study day. The following DATA step merges the initial AE01 data set with ADSL to ensure that *all* ITT subjects are represented, even those not having any adverse events, in which case the variable AEDECOD is valued “No SAEs”. For those adverse events not having an end study day, the AEDECOD variable is modified by appending the text “(Ongoing)”, such as “Bronchtis (Ongoing)”. Thus far, the timelines do not exist.

```
proc sort data=adam_adae(keep=usubjid astdy aendy aebodsys
   aedecod trtemfl) out=ae01;
   by usubjid aebodsys aedecod;
   where aebodsys is not null and aedecod is not null
      and trtemfl eq 'Y' and astdy is not null;
run;

data ae02;
   merge adam_adsl(in=adsl keep=usubjid ittfl
      where=(ittfl eq 'Y')) ae01(in=ae);
      by usubjid;
   if adsl;
   if not ae
      then do;
         astdy    = 0;
         aendy    = 0;
         aedecod  = 'No SAEs';
         end;
   if aendy eq .
      then aedecod = trim(aedecod) || ' (Ongoing)';
run;
```

The timeline for a given adverse event obviously requires knowing *at least* the start study day. But what happens if a subject has *more than one* instance of the same event during the study period? Thus, there can be several line segments, each representing a timeline for the same event. For example, a subject experiences nausea on study day 20 through 30, then again on study day 40 through 50, and so on. It is necessary to discern the pair-wise Start/End study days in order to determine each timeline.

The following TRANSPOSE procedure processes the AE Start and End study days separately (shown below) using a BY-statement for each subject's adverse event, for which there can be more than one occurrence. If a subject has the same adverse event more than once, then the transposed data set with have those Start and End study day variables, named as follows: ASTDY1,…, ASTDY*n* and AENDY1,…,AENDY*n*, specified by the PREFIX options of the TRANSPOSE statement.

```
proc transpose data=ae02(drop=aendy)
   out=astdy(drop=_label_ _name_) prefix=astdy;
   var astdy;
   by usubjid aebodsys aedecod;
run;

proc transpose data=ae02(drop=astdy)
   out=aendy(drop=_label_ _name_) prefix=aendy;
   var aendy;
   by usubjid aebodsys aedecod;
run;
```

The following DATA step merges the two data sets ASTDY and AENDY by the subjects' adverse event. Thus, an observation in the data set AE05 represents the timelines for all the occurrences of an adverse event, per subject. Depending on the maximum number of instances for an adverse event, the data set might contain the following variables:

[USUBJID AEBODSYS AEDECOD] *AESTDY1 AEENDY1 AESTDY2 AEENDY2*

```
data ae05;
   merge astdy aendy;
      by usubjid aebodsys aedecod;
run;
```

In order to formulate the timeline of an adverse event, it is necessary to process the Start / End study days in a pairwise fashion (e.g. ASTDY1 with AENDY1), using parallel arrays in a subsequent Data set. But how many pairs are there? The following SQL step uses the COLUMNS Dictionary table to discern the number of variables denoting the Start study day, which is stored in the macro variable NSDAYS.

```
proc sql noprint;
   select left(put(count(*),3.)) into :nsdays
      from dictionary.columns
      where libname eq 'WORK' and memname eq 'AE05'
         and upcase(name) like 'ASTDY%';
quit;
```

Before formulating the timelines, it is necessary to change the order of the AE data set from **Subject / Event** to **Subject / Event / Study Day** in ascending order. Afterwards, the subsequent DATA step computes the timelines for all adverse events using parallel arrays and a DO-loop that iterates as many times as there are occurrences of a particular adverse event (e.g. Bronchitis).

```
proc sort data=ae05 out=ae06;
   by usubjid astdy1 aendy1;
run;

data ae07;
   array astdys{*} astdy1-astdy&nsdays.;        ← Parallel arrays.
   array aendys{*} aendy1-aendy&nsdays.;
   set ae06;
      by usubjid;
```

```
   if first.usubjid
      then nae = 0;
   nae+1;                                          ← The nth AE.
   do i = 1 to dim(astdys);
      if astdys{i} ge 0                            ← None or at least one AE.
         then do;
            astdy = astdys{i};                     ← Start Day of AE.
            if aendys{i} ne .
               then aendy = aendys{i};
             * else aendy = astdys{i};
            astwk = astdy / 7;                     ← Start Week of AE.
            if aendy eq .
               then aenwk = 24;                    ← Ongoing AE.
               else if aendy eq 0                  ← No AE's.
                  then aenwk = 0;
                  else aenwk = (aendy+1) / 7;      ← End Week of AE.
            output;
            end;
      end;
   keep usubjid nae aedecod astwk aenwk astdy aendy;
   format astwk aenwk 8.2;
run;
```

For a given subject, each adverse event will be assigned an ordinal value (NAE) beginning with the first occurrence. The DO-loop traverses the parallel arrays, using each pairwise Start / End study day variables to determine the timeline of the adverse event, by converting the Study Day to Study Week, ranging from 0 to 24.

Table 2 shows the results for two subjects, each having at least one observation. The events for Subject 1001 are ordered by occurrence and has an adverse event "CATHETER SITE ABSCESS" that includes the text "(Ongoing)" because the end study day is unknown. Subject 1002 has no adverse events at all; consequently, the AEDECOD variable is assigned the text value "No SAEs" with no measurable timeline.

USUBJID	NAE	AEDECOD	ASTDY	AENDY	ASTWK	AENWK
1001	1	STOMA SITE INFECTION	5	20	0.7	3.0
	2	SEPSIS	30	69	4.3	10.0
	2	SEPSIS	84	100	12.0	14.4
	3	GASTROENTERISTIS	51	59	7.3	8.6
	4	LOWER RESPIRATORY TRACT INFECTION	64	78	9.1	11.3
	5	UPPER RESPIRATORY TRACT INFECTION	79	95	11.3	13.7
	6	CATHETER SITE ABSCESS (Ongoing)	84	.	12.0	24.0
	7	PNEUMONIA	121	137	17.3	19.7
	8	INFLUENZA	140	161	20.0	23.1
1002	1	No SAEs	0	0	0.0	0.0

Table 2. Listing of Adverse events for two subjects.

The data sets representing the lab data and the adverse events are now suitable for graphical analysis. And, typical of SAS, there is more than one way to produce these intricate plots, in this case using either the SAS/Graph procedures **GPLOT** or **SGPLOT**, each having different requirements. Also, keep in mind that the plot has two distinct Y-axes: the left axis representing adverse events, if any, and the right axis representing the lab measurements, an ordinal variable and a continuous variable, respectively. The GPLOT solution is discussed first, then the SGPLOT approach.

THE GPLOT SOLUTION

Prior to discussing the X,Y1, and Y2 components of the plot, consider the vertical dose reference lines that depict a subject's tolerance of a study drug with the intent of titrating to the highest dose level of 10 mg. The dose levels may increase and decrease during the study period, depending on the subject's response to the treatment, perhaps prompting an adverse event. Also, as part of the graphical analysis, the dose levels are color-coded and labeled. The SORT procedure and subsequent DATA step obtain and process the dosing data needed for the analysis. Table 3 shows that Subject 1001 had two instances where the dose level was decreased twice, then later increased, eventually reaching the target dose level of 10 mg.

```
proc sort data=adam_adex(keep=usubjid trta astdt astdy) out=reflines01;
   by usubjid astdt astdy;
   where astdt is not null and astdy ge 1;
run;

data reflines02;
   set reflines01(rename=astdy=studydy);
      by usubjid notsorted trta;
   studywk = studydy / 7;
   dose    = trta;
   if first.trta
      then output;
   keep usubjid dose trta studydy studywk;
   format studydy studywk 5.;
run;
```

USUBJID	DOSE	STUDYDY	STUDYWK	
1001	0.5	1.0	0.1	
1001	1.0	15.0	2.1	
1001	2.0	32.0	4.6	
1001	3.0	46.0	6.6	
1001	2.0	51.0	7.3	← Down titration
1001	3.0	60.0	8.6	
1001	5.0	74.0	10.6	
1001	3.0	91.0	13.0	← Down titration
1001	5.0	110.0	15.7	
1001	7.5	121.0	17.3	
1001	10.0	135.0	19.3	

Table 3. Listing of Dose data showing increase and decreasing titration.

The AE data set contains only an ordinal value denoting the *n*th adverse event (NAE) and the study week (STUDYWK), which contributes nothing with respect to a timeline. The values of NAE will be formatted into actual text, for example: "PNEUMONIA" or "No SAEs." Even though the data set contains the Start/End study week of an adverse event, there is no mechanism inside GPLOT for drawing these timelines. Thus, it is necessary to use the Annotate Facility. But first, the following DATA step creates the Control Input data set NAEF that will be used to map the ordinal value of NAE to a recognizable adverse event.

The Control Input data set actually represents all subjects in the study. However, when generating a plot for a given subject, a subset will be created based on the unique subject identifier (USUBJID), then processed by the FORMAT procedure to create the NAEF format specific for a subject. For the GPLOT solution, the NAEF format includes the HLO variable to accommodate the Y-axis whose values range from *0 to n+1*, rather than *1 to n*, in order to make the plot more aesthetically appealing.

```
data naef;
   retain fmtname 'NAEF' type 'N';
   set ae07;
      by usubjid nae;
   if first.nae
      then do;  start=nae;  label=aedecod;            output;  end;
   if last.usubjid
      then do;  start=.;    label=' ';      hlo='O';   output;  end;
   keep usubjid fmtname type start label hlo;
run;
```

USUBJID	FMTNAME	TYPE	START	LABEL	HLO
1001	NAEF	N	1	STOMA SITE INFECTION	
1001	NAEF	N	2	SEPSIS	
1001	NAEF	N	3	GASTROENTERISTIS	
1001	NAEF	N	4	LOWER RESPIRATORY TRACT INFECTION	
1001	NAEF	N	5	UPPER RESPIRATORY TRACT INFECTION	
1001	NAEF	N	6	CATHETER SITE ABSCESS (Ongoing)	
1001	NAEF	N	7	PNEUMONIA	
1001	NAEF	N	8	INFLUENZA	
1001	NAEF	N	.		O
1002	NAEF	N	1	No SAEs	
1002	NAEF	N	.		O

Table 4. Control-Input data set to create the subject-specific format needed to write the adverse events.

For a given subject, the FORMAT procedure creates the NAEF format using the NAEF Control Input data set along with the WHERE data set option, selecting only those observations for that subject, shown below. Table 5 shows the NAEF format for Subject 1001, which is generated by the FMTLIB option of the FORMAT statement.

```
proc format cntlin=naef(where=(usubjid eq "&usubjid.")) fmtlib;
   select naef;
   title2 'NAEF Format';
run;
```

FORMAT NAME: NAEF LENGTH: 33 NUMBER OF VALUES: 9
MIN LENGTH: 1 MAX LENGTH: 40 DEFAULT LENGTH: 33 FUZZ: STD

START	END	LABEL (VER. V7|V8 27NOV2017:14:29:23)
1	1	STOMA SITE INFECTION
2	2	SEPSIS
3	3	GASTROENTERISTIS
4	4	LOWER RESPIRATORY TRACT INFECTION
5	5	UPPER RESPIRATORY TRACT INFECTION
6	6	CATHETER SITE ABSCESS (Ongoing)
7	7	PNEUMONIA
8	8	INFLUENZA
OTHER	**OTHER**	

Table 5. The NAEF format for Subject 1001.

Two annotate data sets are required: one for the dose reference lines, and one for the actual timelines for the adverse events, named ANNO_DOSE and ANNO_AEDUR, respectively. The abridged DATA step below generates the information needed for the dose reference lines and their labels. The dose labels are positioned in an attempt to avoid the text of one dose level (e.g. 1.0 mg) from running into an adjacent dose level. This is accomplished by adjusting the X and Y values in the annotate data set. In essence the labels alternate, up and down, in an ordered fashion so that the next dose level is either higher or lower than the preceding one. Table 6 contains a partial listing of the annotate data set. Notice the Y-values in Table 6 alternate between 92.5 and 95.0 so that the labels do not collide with each other. Also, the X-values are increased by a constant value so that the dose label appears adjacent to the vertical reference line.

```
data anno_dose;
   retain decr 5;
   length function color text $8;
   retain xsys '2' ysys '2' hsys '1' when 'a';
   set reflines02;
   select(trta);
      when(0.50) do;
         function = 'move';  y = 0;  x = studywk;   line = 5;
            color = 'green';  size = 0.50;                              output;
         function = 'draw';  y = 100;                                   output;
         function = 'label'; y = 100-(1.5*decr); x=x+0.10; line = .;
            color = 'black'; position='6'; size = 2.5; text='0.5mg';    output;
         end;
      when(1.0) do;
         function = 'move';  y = 0;  x = studywk;   line = 7;
            color = 'purple'; size = 0.50;                              output;
         function = 'draw';  y = 100;                                   output;
         function = 'label'; y = 100-(1.0*decr); x=x+0.10; line = .;
            color = 'black'; position='6'; size = 2.5; text='1.0mg';    output;
         end;

       :          :          :          :          :          :

      otherwise;
      end;
   keep usubjid function color xsys ysys hsys when x y size line text position;
run;
```

X	Y	FUNCTION	COLOR	LINE	TEXT
0.1429	0.0	move	green	5	
	100.0	draw	green	5	
0.2429	92.5	label	black	.	0.5mg
2.1429	0.0	move	purple	7	
	100.0	draw	purple	7	
2.2429	95.0	label	black	.	1.0mg
4.5714	0.0	move	orange	9	
	100.0	draw	orange	9	
4.6714	92.5	label	black	.	2mg
6.5714	0.0	move	brown	15	
	100.0	draw	brown	15	
6.6714	95.0	label	black	.	3mg

Table 6. Partial listing of the Annotate data set for labeling Dose levels.

The DATA step below creates the annotate data set ANNO_AEDUR that contains the timelines for all adverse events, for all subjects. Here again, the appropriate records will be selected per the subject identifier, per patient profile. The aforementioned AE07 data set supplies the needed information for annotation, specifically the variables: USUBJID, NAE denoting the *n*th adverse event, ASTWK and AENWK, the start / end week of the study, respectively. The annotate data set contains pairs of observations, MOVE to a specific X,Y coordinate, then DRAW to the destination X-coordinate, along the same Y-coordinate denoting the *n*th adverse event. Table 7 highlights the two timelines for the adverse event Sepsis, for Subject 1001; whereas, Subject 1002 has no adverse events.

```
data anno_aedur;
   length function color $8;
   retain xsys '2' ysys '2' hsys '1' when 'a' color 'black' line 1 size 1.0;
   set ae07;
   function = 'move';  y = nae;   x = astwk;    output;
   function = 'draw';  y = nae;   x = aenwk;    output;
   keep usubjid page function color xsys ysys hsys when line size x y;
run;
```

USUBJID	FUNCTION	Y	X	COLOR	LINE
1001	move	1	0.7143	black	1
	draw	1	3.0000	black	1
	move	2	4.2857	black	1
	draw	2	10.0000	black	1
	move	2	12.0000	black	1
	draw	2	14.4286	black	1
	move	3	7.2857	black	1
	draw	3	8.5714	black	1
	move	4	9.1429	black	1
	draw	4	11.2857	black	1
	move	5	11.2857	black	1
	draw	5	13.7143	black	1
	move	6	12.0000	black	1
	draw	6	24.0000	black	1
	move	7	17.2857	black	1
	draw	7	19.7143	black	1
	move	8	20.0000	black	1
	draw	8	23.1429	black	1
1002	move	1	0.0000	black	1
	draw	1	0.0000	black	1

Table 7. Annotate data set representing timelines of adverse events.

THE SGPLOT SOLUTION

Unlike the GPLOT solution, the SGPLOT procedure requires a separate data set for the reference lines; that is, each dose level is represented by its own variable, such as MG5 representing dose 5 mg and contains the study week for that instance of that dose level. The REFLINES statement of the SGPLOT uses these several variables in order to specify attributes to each reference line, including: color, pattern, and label. The DATA step below creates the relevant variables that indicate the location of the vertical reference lines along the X-axis. The SELECT/WHEN statement assigns a value to only one MG variable per observation, which produces a peculiar data set, as shown in Table 8, yet makes sense for the SGPLOT procedure. Notice that the values of the MG variables are reasonably the same as the respective values denoting the study week (STUDYWK).

```
data reflines02;
   length dosec $5;
   array doses[7] mg0_5 mg1 mg2 mg3 mg5 mg7_5 mg10 (7*.);
   set reflines01;
      by usubjid notsorted trta;
   studywk = astdy / 7;
   studydy = astdy;
   dose    = trta;
   dosec   = cats(dose,'mg');
   call missing(of doses[*]);
   select(dosec);
      when('0.5mg')   mg0_5 = studywk;    when('1mg')     mg1   = studywk;
      when('2mg')     mg2   = studywk;    when('3mg')     mg3   = studywk;
      when('5mg')     mg5   = studywk;    when('7.5mg')   mg7_5 = studywk;
      when('10mg')    mg10  = studywk;    otherwise put _error_;
     end;
  keep usubjid dose dosec trta studydy studywk mg:;
  format studydy studywk 5.1;
run;
```

USUBJID	STUDYWK	DOSE	MG0_5	MG1	MG2	MG3	MG5	MG7_5	MG10
1001	0.1	0.5	0.1	.	.	.	.	.	.
1001	2.1	1.0	.	2.1	.	.	.	.	.
1001	4.6	2.0	.	.	4.6	.	.	.	.
1001	6.6	3.0	.	.	.	6.6	.	.	.
1001	7.3	2.0	.	.	7.3	.	.	.	.
1001	8.6	3.0	.	.	.	8.6	.	.	.
1001	10.6	5.0	.	.	.	.	10.6	.	.
1001	13.0	3.0	.	.	.	13.0	.	.	.
1001	15.7	5.0	.	.	.	.	15.7	.	.
1001	17.3	7.5	.	.	.	.	.	17.3	.
1001	19.3	10.0	.	.	.	.	.	.	19.3

Table 8. Listing of component data set to include dose reference lines.

The component data set REFLINES02 is concatenated along with the two other data sets representing the lab and adverse events (AE07 and LB02), each using the WHERE data set option in order to select subject-specific observations, creating a heterogeneous collection of data. Table 9 shows a partial listing of the data set used to produce the plot for Subject 1001. The FORMAT statement includes the NAEF format that translates the ordinal variable NAE to the text, either an adverse event or “No SAE’s”. Also, keep in mind that this DATA step would be executed inside a macro denoting the *n*th subject when generating the patient profiles.

```
data plotds_&usubjid;
   set ae07(keep=usubjid nae astwk aenwk    where=(usubjid eq "&usubjid.")
       lb02(keep=usubjid aval studywk       where=(usubjid eq "&usubjid."))
       reflines02(keep=usubjid studywk
          dosec mg: rename=(studywk=dosewk  where=(usubjid eq "&usubjid."));
   format nae naef.;
run;
```

							------ Dose (MG) Variables ------						
NAE	ASTWK	AENEK	STUDYWK	AVAL	DOSEC	DOSEWK	0.5	1	2	3	5	7.5	10.0
1	0.71	3.00	.	.		.	.	.	.	.	.	.	.
2	4.29	10.00	.	.		.	.	.	.	.	.	.	.
2	12.00	14.43	.	.		.	.	.	.	.	.	.	.
3	7.29	8.57	.	.		.	.	.	.	.	.	.	.
4	9.14	11.29	.	.		.	.	.	.	.	.	.	.
5	11.29	13.71	.	.		.	.	.	.	.	.	.	.
6	12.00	24.00	.	.		.	.	.	.	.	.	.	.
7	17.29	19.71	.	.		.	.	.	.	.	.	.	.
8	20.00	23.14	.	.		.	.	.	.	.	.	.	.
.	.	.	0.1	30		.	.	.	.	.	.	.	.
.	.	.	2.1	35		.	.	.	.	.	.	.	.
.	.	.	4.6	40		.	.	.	.	.	.	.	.
.	.	.	6.6	45		.	.	.	.	.	.	.	.
.	.	.	8.6	50		.	.	.	.	.	.	.	.
.	.	.	10.6	55		.	.	.	.	.	.	.	.
.	.	.	13.0	60		.	.	.	.	.	.	.	.
.	.	.	15.0	65		.	.	.	.	.	.	.	.
.	.	.	17.3	70		.	.	.	.	.	.	.	.
.	.	.	19.3	75		.	.	.	.	.	.	.	.
.	.	.	21.7	79		.	.	.	.	.	.	.	.
.	.	.	23.7	80		.	.	.	.	.	.	.	.
.	.	.	.	.	0.5mg	0.1	0.14	.	.	.	.	.	.
.	.	.	.	.	1mg	2.1	.	2.14	.	.	.	.	.
.	.	.	.	.	2mg	4.6	.	.	4.57	.	.	.	.
.	.	.	.	.	3mg	6.6	.	.	.	6.57	.	.	.
.	.	.	.	.	2mg	7.3	.	.	7.29	.	.	.	.
.	.	.	.	.	3mg	8.6	.	.	.	8.57	.	.	.
.	.	.	.	.	5mg	10.6	.	.	.	.	10.6	.	.
.	.	.	.	.	3mg	13.0	.	.	.	13.0	.	.	.
.	.	.	.	.	5mg	15.7	.	.	.	.	15.7	.	.
.	.	.	.	.	7.5mg	17.3	.	.	.	.	.	17.3	.
.	.	.	.	.	10mg	19.3	.	.	.	.	.	.	19.3

Table 9. Partial listing of plot data set for Subject 1001.

Besides the rather awkward plot data set, the SGPLOT procedure needs additional information for the situation when a subject has no adverse events. The matter concerns the COLOR option of the VECTOR statement, shown below, that draws the actual adverse event timelines. If the subject has no adverse events, then this option must be designated the color white so that nothing shows; black otherwise. Using the ordinal variable NAE, the SQL step below determines whether a subject has one or more than one adverse event. Subject 1001 has AEVALUES: 8,7,6,5,4,3,2,1; whereas, Subject 1002 has AEVALUES of 1, which equates to “No SAE’s”; hence, the COLOR option will be assigned WHITE, rather than BLACK.

```
vector x=aenwk y=nae / xorigin=astwk yorigin=nae noarrowheads
   lineattrs=(pattern=solid color=&aecolor. thickness=1%);

proc sql noprint;
  select distinct nae into :aevalues separated by ','
     from ae07
     where usubjid eq "&usubjid."
     order by nae desc;
quit;

%if &aevalues eq 1
   %then %let aecolor=white;      ← No SAE’s.
   %else %let aecolor=black;      ← Actual adverse events.
```

The SGPLOT step below contains thirteen statements needed to generate the desired plot, which includes seven REFLINE statements for the vertical dose reference lines. The text highlighted in yellow in juxtaposition to each statement indicates the particular task in building the plot. Consider these statements in the context of the graphical component, as follows:

- Plot line SCATTER, SERIES
- Adverse Event Timelines VECTOR
- Axis Definitions XAXIS, YAXIS, Y2AXIS
- Dose Reference Lines REFLINE < *MG-variable* >

```
proc sgplot data=plotds_&usubjid noautolegend;
   scatter x=studywk  y=aval
      / y2axis markerattrs=(symbol=circlefilled
        color=black);                                    ← Antigen plot points.
   series x=studywk  y=aval
      / y2axis lineattrs=(pattern=solid color=black);    ← Antigen plot line.
   vector x=aenwk y=nae
      / xorigin=astwk yorigin=nae noarrowheads
        lineattrs=(pattern=solid
           color=&aecolor thickness=1%);                 ← AE Duration timelines.
   xaxis integer values=(0 to 24 by 2)
        minor minorcount=1
        label='Study Day (Weeks)' tickvalueformat=nwk.; ← X-axis: Study Weeks.
   y2axis integer values=(0 to 100 by 10)
        minor minorcount=1 label='Antigen';             ← Y2 axis for Antigen.
   yaxis integer label='Adverse Events' labelpos=top
        tickvalueformat=naef.
        values=(&aevalues) reverse;                     ← Y-axis for AEs.
   refline mg0_5 / axis=x label=dosec
        labelpos=min lineattrs=(color=green
        pattern=5  thickness=0.5%);                     ← 0.5mg Dose.
   refline mg1 / axis=x label=dosec
        labelpos=min lineattrs=(color=purple
        pattern=7  thickness=0.5%);                     ← 1mg Dose.
   refline mg2 / axis=x label=dosec
        labelpos=min lineattrs=(color=orange
        pattern=9  thickness=0.5%);                     ← 2mg Dose.
   refline mg3 / axis=x label=dosec
      labelpos=min lineattrs=(color=brown
      pattern=11 thickness=0.5%);                       ← 3mg Dose.
   refline mg5 / axis=x label=dosec
      labelpos=min lineattrs=(color=red
      pattern=15 thickness=0.5%);                       ← 5mg Dose.
   refline mg7_5 / axis=x label=dosec
      labelpos=min lineattrs=(color=green
      pattern=15 thickness=0.5%);                       ← 7.5mg Dose.
   refline mg10 / axis=x label=dosec
      labelpos=min lineattrs=(color=yellow
      pattern=15 thickness=0.5%);                       ← 10mg Dose.
quit;
```

The VECTOR statement that draws the adverse event timelines specifies the Y*X co-ordinates, that is, the *n*th adverse event and the END study week of the event, and includes the XORIGIN and YORIGIN options that indicates the START of the timeline. The versatility of the VECTOR statement give SGPLOT and advantage over the GPLOT procedure that requires the Annotate Facility. However, the dose reference lines seem to be less versatile and more cumbersome as compared to the implementation using GPLOT. For example, the dose labels are positioned on the same line, rather than alternating to avoid the text from colliding. Also, the plot data sets for GPLOT and SGPLOT are very different, the first using a more intuitive plot and annotate data sets while the latter uses a more heterogeneous data set.

PATIENT PROFILES

Prior to producing the graphical patient profiles, the Data _null_ step below creates the macro variables that identify the subjects, as well as how many.

```
data _null_;
   retain n 0;
   set ae07(keep=usubjid) end=eof;
      by usubjid;
   if first.usubjid
      then do;
         n+1;
         call symput('sub' || left(put(n,3.)), trim(usubjid));
         end;
   if eof
      then call symput('nplots', left(put(n,3.)));
run;
```

There are two macros that produce the plots: a driver macro, called ***%genplots***, and its subject-specific subordinate macro, called ***%genplot***. The driver macro, shown below, applies to both SAS solutions, which simply executes the subordinate macro for as many times as there are subjects. Notice the double ampersand that denotes deferred addressing in the Macro Language, translating the macro variable SUB1 to subject identifier 1001.

```
%macro genplots;
   %do i = 1 %to &plots.;
      %genplot(usubjid=&&sub&i.);
      %end;
%mend genplots;
```

The subordinate macro *%genplot* differs significantly between the GPLOT and SGPLOT solutions, as shown in the lists below. Only the creation of the NAEF format, mapping the ordinal variable NAE to the text of an adverse event, appears in both solutions. Otherwise, the GPLOT solution relies more on annotate data sets and SAS/Graph statements; whereas, the SGPLOT approach relies more on the plot data set and the statements in SGPLOT.

The *%genplot* macro has one parameter USUBJID that identifies one subject, as defined in the driver macro. Depending on the proposed method, the *%genplot* macro has several components outlined below.

GPLOT PROCEDURE

- Identify maximum number of adverse events plus 1.

```
proc sql noprint;
   select max(nae)+1 into :maxnae
      from ae07
      where usubjid eq "&usubjid.";
quit;
```

- Obtain the NAEF format for *n*th subject.

```
proc format cntlin=naef(where=(usubjid eq "&usubjid."));
run;
```

- Create plot data set consisting of adverse events and lab data only. The plot data set contains only three variables: STUDYWK, NAE, AND AVA, as shown ion Table 10.

```
data plotds;
   set ae07(keep=usubjid nae astwk rename=astwk=studywk
      where=(usubjid eq "&usubjid."))
   lb03(keep=usubjid aval studywk where=(usubjid eq "&usubjid."));
run;
```

STUDYWK	NAE	AVAL
0.71	1	.
4.29	2	.
:	:	:
17.29	7	.
20.00	8	.
0.00	.	30
2.00	.	35
:	:	:
19.00	.	75
21.00	.	79
24.00	.	80

Table 10. Partial listing of plot data set for Subject 1001.

- Specify SYMBOL statements for plot line with dots.

```
symbol1               color=black   value=none   height=1.0;  * AE's ;;
symbol2 line=1 i=join color=black   value=dot    height=1.0;  * Antigen ;;
```

- Specify three AXIS statements for X, Y, and Y2 axes.

```
axis1 order=&maxnae. to 0 by -1
   minor=none label=(h=1.0 j=right f=swissx "Adverse Events");
axis2 order=0 to 24 by 2
   minor=(number=1 h=0.5) label=(h=1.0 j=c f=swissx "Study Day (Weeks)");
axis3 order = 0 to 100 by 10
   minor=(number=1 h=0.5) major=(h=0.8)
   label=(angle=-90 h=1.0 f=swissx color=black 'Antigen');
```

- Generate plot.

```
proc gplot data=plotds;
   plot  nae  * studywk / vaxis=axis1 haxis=axis2
      anno=anno_aedur(where=(usubjid eq "&usubjid."));
   plot2 aval * studywk / vaxis=axis3  overlay
     anno=anno_dose(where=(usubjid eq "&usubjid.") nolegend;
   format nae naef. studywk nwk.;
run; quit;
```

SGPLOT PROCEDURE

- Create macro variable denoting values of ordinal NAE variable to determine whether the COLOR option of the VECTOR statement should be white (No SAE's) or black (actual adverse events).

```
proc sql noprint;
   select distinct nae into :aevalues separated by ','
      from ae07
      where usubjid eq "&usubjid.";
quit;
%if &aevalues eq 1
   %then %let aecolor = white;
   %else %let aecolor = black;
```

- Create the NAEF format for the *n*th subject.

```
proc format cntlin=naef(where=(usubjid eq "&usubjid.")) fmtlib;
   select naef;
run;
```

- Create plot data set (See Table 9) consisting of adverse events, lab data, and dose reference lines.

```
data plotds_&usubjid;
   set ae07(keep=usubjid nae astwk aenwk where=(usubjid eq "&usubjid."))
       lb02(keep=usubjid aval studywk where=(usubjid eq "&usubjid."))
       reflines02(keep=usubjid studywk dosec mg:
          rename=(studywk=dosewk)  where=(usubjid eq "&usubjid."));
```

```
   format nae naef.;
run;
```

- Create the plot (See code above). Note: No SYMBOL or AXIS statements required.

Figure 2, generated by the GPLOT method, shows the profile for Subject 1002 who has no adverse events as indicated by the text "No SAE's". Notice how the position of the dose labels go up and down in order to avoid any collision of the text. The SGPLOT method places the dose labels outside of the actual plot along the same line. Using the LABELLOC=INSIDE option of the REFLINE statement would place the dose labels inside the plot, albeit not in an alternating fashion. Moreover, the LABELLOC option is available only after SAS®, Version 9.4.

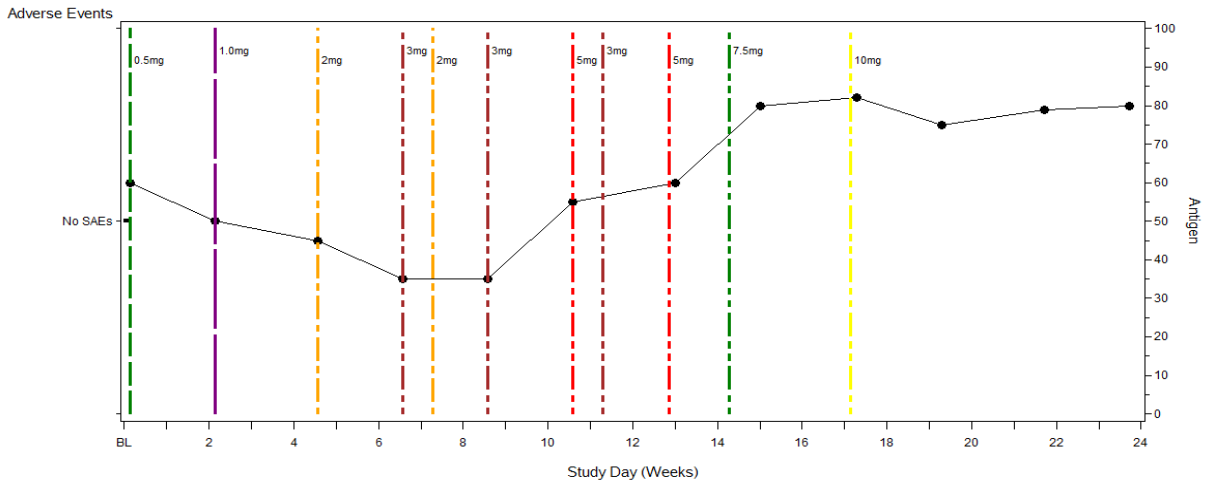

Figure 2. Subject 1002 having Lab measurements and Does Titration Levels, but **no** Adverse Events.

CONCLUSION

The proposed intricate plot superimposes dose titration reference lines and adverse event timelines over a longitudinal plot representing a lab measurement, thereby depicting the relationships of several factors in tandem. The first solution explained the use of GPLOT along with two annotate data sets: for dose reference lines and adverse event timelines. The second solution discussed the use of SGPLOT with an agglomeration of data, albeit without any annotate data sets. Depending on the actual analysis or possible enhancement, one method might be preferred over the other For example, a patient-profile might contain many adverse events that would require pagination; or, perhaps the analysis variable might represent a log-transformation that would require subject-specific calibration of the Y-axis. Both methods have their advantages, even if only a programmer's preference.

REFERENCES

Matange, Sanjay, "*Patient Profile Graphs Using SAS***".** Paper 160-2013, SAS Global Forum 2013.

Okerson, Barbara B. "*Fun with Timelines: Doing More with SAS/Graph Proc GPLOT.*" Paper 218-27, SUGI 27.

"*How to Create a Timeline Plot* | *SAS Code Fragments*", from https://stats.idre.ucla.edu/sas/code/how-to-create-a-timeline-plot/ (accessed October 25, 2017).