PharmaSUG 2024 - Paper DV-186

Enhanced Spider Plot in Oncology

Ilya Krivelevich, Cixin Steven He, Binbin Zhang-Wiener, and Wenyin Lin, Eisai Inc

ABSTRACT

Graphs are an integral part of modern data analysis of clinical trial. Viewing data in a graph together with the tabular results of statistical analysis can greatly improve understanding of the collected data. Graphical data display provides insight into trends and correlations that are simply not possible with tabular data; the visual representation can very often be the most informative way to understand the insight results.

The spider plot of change in tumor size from baseline is one of the more common graphs in oncology studies. Unlike the waterfall graph, which displays the maximum change from baseline for each subject, the spider plot allows us to visualize change from baseline over the time. Per our experience, the spider plots could also display other clinical meaningful information, such as time-point responses, study drug dosage, and some subject level information, for example, the value of best overall response. This additional information can be very helpful for reviewers.

The demonstrations in this paper are based on RECIST 1.1 evaluation criteria and can be easily adapted to any other tumor evaluation criteria.

CHAPTER 1. TRADITIONAL (SIMPLE) SPIDER PLOT WITHOUT ENHANCEMENTS

Spider plots are easier to interpret for a small number of subjects. If the number of subjects is large, the plot may be more messy and harder to read. In a spider plot, every subject is represented as a line that starts from the same point. These lines may look like the legs of spider; each “leg” represents a unique subject. Subjects usually stay on a study treatment different amount of time (till progression, death, or discontinuation for any other reasons) in oncology clinical trials, so the spider’s legs are expected to be different in length.

For simplicity, let’s consider 4 subjects for whom we need to create a spider plot. According to the protocol, subjects should be evaluated every 6 weeks (every 42 days), but some minor deviations from this schedule are very likely. Also, subjects may skip one or more evaluations for different reasons.

The sums of diameters of target lesions can be found in SDTM domain TR (Variable TRSTRESN where TRTESTCD = ‘SUMDIA’). Based on these collected data, we can easily derive percentage change from baseline for each evaluation. We should assume that baseline measurements correspond to study day 1, and, obviously, the percentage change from baseline for these first records for each subject is equal to zero. See Table 1 for the data used in this chapter.

subjid (Subject ID)	evaluation_day (Study Evaluation Day)	percent_change (Percentage Change from Baseline)
1001	1 (Baseline)	0
1001	42	5
1001	84	-10
1001	126	-25
1001	168	-55
1001	210	-90
1001	252	-100
1001	290	-80
1002	1 (Baseline)	0
1002	41	10
1002	85	5
1002	125	-20
1002	169	-35
1002	209	-25

1002	253	-40
1003	1 (Baseline)	0
1003	40	-35
1003	80	-40
1003	130	-100
1003	170	-100
1003	210	-60
1004	1 (Baseline)	0
1004	45	15
1004	140	-5
1004	190	-35

Table 1. Percentage Change from Baseline

Let's name this dataset timepoints. Simple SAS code using SERIES statement from PROC SGPLOT produces a simple spider plot (see Figure 1) that we will use as a starting point for future enhancement:

```
proc sgplot data = timepoints noautolegend;
  series x = evaluation_day y = percent_change
    /
    group = subjid
    lineattrs = (pattern = solid color = black)
    ;
```

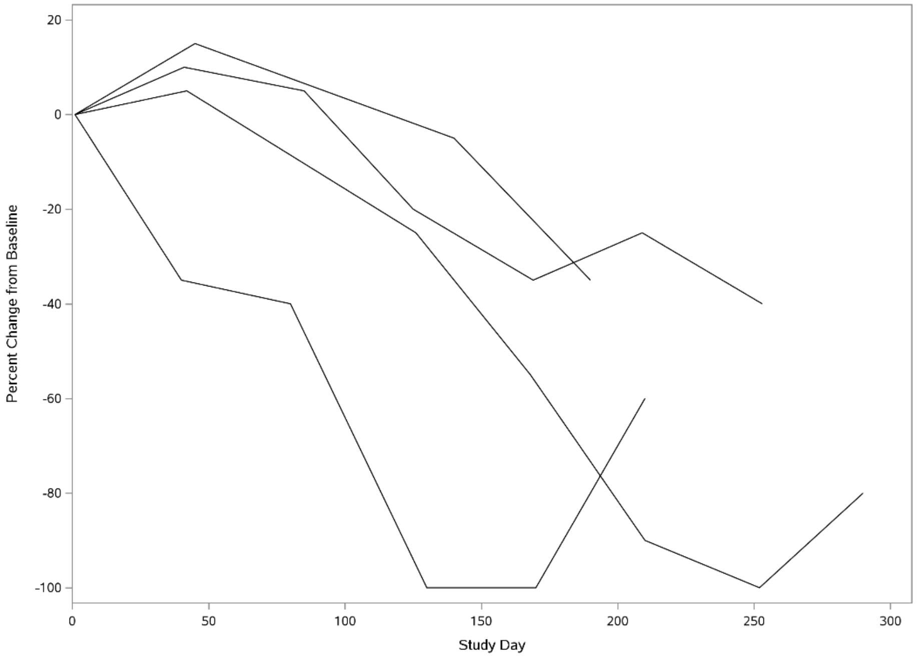

Figure 1. Spider plot without enhancements

As we can see, the produced output is not informative. Much more information can be added into the figure using different enhancements.

CHAPTER 2. ADDING REFERENCE LINES

To make the figure more informative, we recommend using horizontal reference lines at 0 to emphasize the direction of change from baseline. Also, horizontal reference lines at +20 and -30 reflect RECIST 1.1 thresholds for Stable Disease in comparison with baseline. If different evaluation criteria are in use and these criteria have different Stable Disease thresholds, the reference lines should be drawn at different values.

The recommended statement to be added is:

```
    refline -30 0 20 / axis = Y lineattrs = (color = ligr pattern = dash);
```

This statement will transform our figure into:

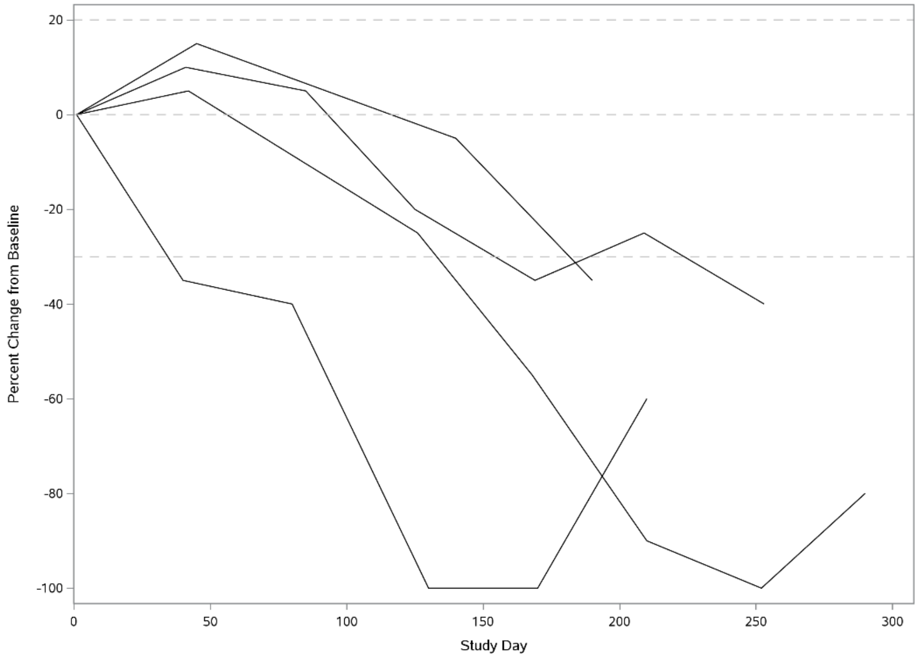

Figure 2. Spider plot with reference lines

CHAPTER 3. ADDING RESULTS OF TIMEPOINT RESPONSES

As we can see, our figure looks like a set of polygonal chains (spider legs), where each chain represents a separate subject in a study. Each point of the chain corresponds to a timepoint assessment. During such assessments, physicians evaluate subjects and assign timepoint response based on a timepoint finding. In this case, to provide more information about timepoint assessment, we should update Table 1 to Table 2, which contains not only the sum of diameters but results of response timepoint evaluations. Such results are usually collected in SDTM domain RS.

subjid (Subject ID)	evaluation_day (Study Evaluation Day)	percent_change (Timepoint Percent Change from Baseline)	response (Timepoint Response)
1001	1 (Baseline)	0	
1001	42	5	SD
1001	84	-10	SD
1001	126	-25	SD
1001	168	-55	PR
1001	210	-90	PR
1001	252	-100	CR
1001	290	-80	PD
1002	1 (Baseline)	0	
1002	41	10	SD
1002	85	5	SD
1002	125	-20	SD
1002	169	-35	PR
1002	209	-25	SD
1002	253	-40	PR
1003	1 (Baseline)	0	
1003	40	-35	PR

1003	80	-40	PR
1003	130	-100	CR
1003	170	-100	CR
1003	210	-60	PD
1004	1 (Baseline)	0	
1004	45	15	SD
1004	140	-5	SD
1004	190	-35	PR

Table 2. Percentage Change from Baseline and Results of Timepoint Evaluations

The set of possible values for timepoint response is limited, as it is dictated by the evaluation criteria. For example, with RECIST 1.1 criteria, the possible values can be CR (Complete Response), PR (Partial Response), SD (Stable Disease), PD (Progressive Disease), or NE (Not Evaluable). Result of timepoint evaluation non-CR/non-PD is not relevant to our figure, because our figure includes only subjects with measurable disease, while non-CR/non-PD can be assigned only to subjects without measurable target lesions. If other (not RECIST 1.1) evaluation criteria are employed, then the set of possible values should be in accordance with the criteria guidance.

We can certainly have results of Timepoint Percent Change from Baseline and corresponding results of Timepoint Responses in the same observation of data set timepoints. However, we propose to have 2 major parts in this data set; this will allow us to have entries in the legend in the desired order (CR, PR, SD, PD, NE) instead of the default alphabetical order. For each possible values of timepoint response, we should have a temporary corresponding numerical equivalent (for example, 1 for CR, 2 for PR, 3 for SD, 4 for PD, 9 for NE); this numerical equivalent will be used for sorting purposes. Let's name this additional variable response_num. For example, for subject 1003 we will have the records below:

subjid	part	evaluation_day	percent_change	response	response_num
1003	1	1	0		
1003	1	40	-35		
1003	1	80	-40		
1003	1	130	-100		
1003	1	170	-100		
1003	1	210	-60		
1003	2	40	-35	PR	2
1003	2	80	-40	PR	2
1003	2	130	-100	CR	1
1003	2	170	-100	CR	1
1003	2	210	-60	PD	4

We also recommend not using the variable percent_change in part 2 and substituting it with the variable response_y with the same values. This newly created variable response_y will be used as a vertical coordinate for the markers, representing results of timepoint evaluations.

To control appearance of the marker indicating a particular response, we should use attribute map. Let's create this attribute map for possible value of response:

```
data attribute_map_responses;
  length id $11 value $20 markersymbol $15;
  input id value markersymbol;
  datalines;
  evaluations CR DiamondFilled
  evaluations PR CircleFilled
  evaluations SD TriangleFilled
  evaluations PD SquareFilled
  evaluations NE StarFilled
  ;
run;
```

Now, let's sort our data set timepoints in the needed order:

```
proc sort data = timepoints;
  by part response_num subjid evaluation_day;
run;
```

If we return to subject 1003, we will have records for this subject in the needed order:

subjid	part		evaluation_day	percent_change	response	response_num	response_y
1003	1		1	0			
1003	1		40	-35			
1003	1		80	-40			
1003	1		130	-100			
1003	1		170	-100			
1003	1		210	-60			
1003	2		130		CR	1	-100
1003	2		170		CR	1	-100
1003	2		40		PR	2	-35
1003	2		80		PR	2	-40
1003	2		210		PD	4	-60

To display results of timepoint evaluations, we need to use the SCATTER statement from PROC SGPLOT.

The whole PROC SGPLOT SAS procedure (without statements for title, footnotes, or axes) now looks like this:

```
proc sgplot data = timepoints d dattrmap = attribute_map_responses
  noautolegend;
  series x = evaluation_day y = percent_change
    /
    group = subjid
    lineattrs = (pattern = solid color = black)
    ;
  scatter x = evaluation_day y = response_y
    /
    group = response nomissinggroup
    attrid = evaluations
    name = 'the_evaluations'
    markerattrs = (size = 10 color = black)
    ;
  keylegend 'the_evaluations'
    /
    exclude = (' ')
    across = 1
    border
    location = inside
    position = SW
    outerpad = (bottom = 10 px)
    title = 'Results of Timepoint Assessments'
    titleattrs = (size = 9)
    valueattrs = (size = 8)
    ;
    refline -30 0 20 / axis = Y lineattrs = (color = ligr pattern = dash);
```

Our figure after submitting the SAS code above will be transformed into:

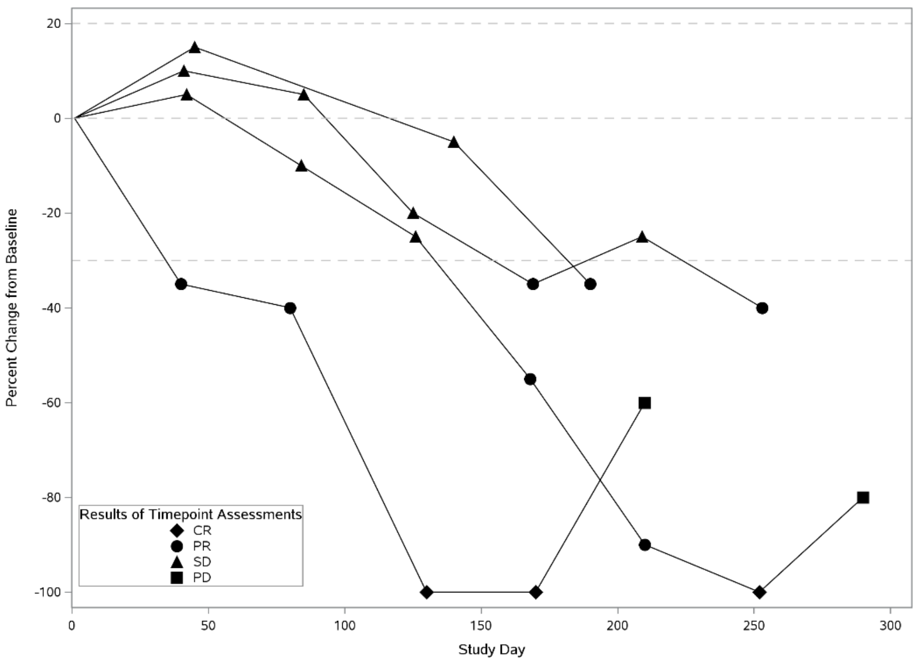

Figure 3 Spider plot with reference lines and results of timepoint responses

CHAPTER 4: ADDING DOSAGE INFORMATION

During the study, daily dosage information is collected for each treated subject. This information is usually stored in SDTM domain EX. During the treatment, the physician may make decisions to reduce or increase a patient's dose following protocol. This decision is based mostly on the patient's medical condition. The physician can also make a decision about treatment interruption. The study protocol has a list of allowed daily dose levels. The protocol allows the physician to make such decisions about dose changing or temporary dose interruption at any day of the treatment, not necessarily on the same dates as the efficacy evaluations.

Let's consider that the starting daily dosage was 10 mg for our 4 subjects. The protocol allows an increase of the daily dose to 15 mg or a sequential decrease to 10 mg or 5 mg. In cases of drug interruptions, SDTM domain EX provides us with a daily dosage of 0 mg.

The full information about daily dosage for our 4 subjects can be seen below in Table 3. Let's name this data set, created from SDTM domain EX and containing records about daily dosing, exposure.

subjid (Subject ID)	start_interval_day (Study Day of Treatment Interval Start)	end_interval_day (Study Day of Treatment Interval End)	dosage (Daily Dosage)
1001	1	99	10 mg
1001	100	299	15 mg
1002	1	149	10 mg
1002	150	199	0 mg
1002	200	299	5 mg
1002	300	349	10 mg
1003	1	44	10 mg
1003	45	64	5 mg
1003	65	219	10 mg
1003	220	299	15 mg
1004	1	79	10 mg
1004	80	155	15 mg

Table 3. Daily Dosing Information from SDTM domain EX

If we want the dosing information to be incorporated into our spider plot, we need to adjust the dosing records. This is a process with several steps.

Step 1

STDT domain EX contains dosing information for each subject as a set of intervals. Usually for each subject in data set exposure, the next interval starts one day after the previous interval ended. For example, for subject 1001, the first dosing interval is till day 99, the second interval starts from day 100. However, if we want to put dosing information into the plot, we should not have gaps between dosing information intervals. To eliminate these gaps, we should use a study day when dosage was changed as a Study Day of Treatment Interval End value in our figure (we will see 1 day increase in value of end_interval_day in this adjusted table in comparison with values, extracted from SDTM domain EX). Let's name exposure1 the exposure data set, adjusted after this step.

subjid (Subject ID)	start_interval_day (Study Day of Treatment Interval Start)	end_interval_day (Study Day of Treatment Interval End)	dosage (Daily Dosage)
1001	1	100	10 mg
1001	100	300	15 mg
1002	1	150	10 mg
1002	150	200	0 mg
1002	200	300	5 mg
1002	300	350	10 mg
1003	1	45	10 mg
1003	45	65	5 mg
1003	65	220	10 mg
1003	220	300	15 mg
1004	1	80	10 mg
1004	80	156	15 mg

Table 3a. Daily Dosing Information from SDTM domain EX adjusted for using in spider plot

Of course, this adjustment is done only to produce needed figure and does not indicate that at the same study day (like day 100 for subject 1001) patient had 2 different dosages.

Step 2

Because each subject's spider plot displays information collected no later than the date of last efficacy evaluation, let's drop records with start_interval_day after the date of last efficacy evaluation from the data set exposure1. We will get data set exposure2.

For example, for subject 1002, the last exposure record with start_interval_day = 300 will be dropped because the last efficacy evaluation for this subject was done on study day 253.

Step 3

Data set exposure2 from the previous step may have chronologically latest exposure intervals for each subject that end after the date of last efficacy evaluation. For the same reason as above (because the spider plot for each subject displays information collected no later than the date of last efficacy evaluation), we should cut this chronologically latest exposure interval back to the day of latest efficacy evaluation. We will get data set exposure3.

If we return to our subject 1002, the latest exposure interval for this subject will now be from study day 200 to study day 253 (in data set exposure2, this interval was from day 200 to day 300).

Step 4

For each subject, if the chronologically latest exposure interval from data set exposure3 ends before the date of latest efficacy evaluation, an additional exposure record will be added starting from the chronologically latest end_interval_day for this subject and ending with the study day of last efficacy evaluation. This interval will have a value of dosage = 0 mg. We will get data set exposure4 with these newly added records.

For example, for subject 1004, we will get an additional record with an interval from study day 156 to study day 190 and a value of dosage = 0 mg.

Step 5

This step is more challenging to implement than the previous 4 steps. If a dosing interval is completely located between 2 consecutive efficacy evaluations, this dosing interval should be present without change in the data set exposure5 that we expect to get after this step. However, if a dosing interval starts between one pair of consecutive efficacy evaluations and ends between a different pair of consecutive efficacy evaluations, this dosing interval should be split into several dosing intervals in such a way that each new dosing interval can be placed completely inside of a pair of consecutive efficacy evaluations.

This idea is easy to digest by looking at some exposure intervals for subject 1003. One of the exposure intervals provided from dataset exposure1 is an interval from study day 65 to study day 220 with a daily dosage of 10 mg. After step 4, this exposure interval has an end day = 210. From the data set timepoints we know that this subject has the following days of efficacy evaluation: day 80, day 130, day 170, and day 210. In this case, the exposure interval between day 65 and day 210 will be split into 4 dosage intervals, each of them having a daily dosage of 10 mg:

- between day 65 and day 80
- between day 80 and day 130
- between day 130 and day 170
- between day 170 and day 210

After the end of this step, for each subject we will have a consecutive set of exposure intervals such that all these intervals will be inside of pairs of consecutive evaluation days.

Step 6

This step is needed to derive percent change from baseline in sum of target lesion diameter for each dosing interval in the recently created data set exposure5. Assuming linearity of changing in sum of diameters between evaluations, we can derive a value of percent change of sum of diameters for the start and end of each dosing interval, using preliminary derived slope for each interval from the chapter 1 data set timepoints.

Let's name this last data set about exposure exposure_final. Each exposure interval in this data set continues from study day start_x till study day end_x, and for each of these exposure intervals, we have percent change in sum of target lesion diameters from start_y till end_y. We also recommend having one additional variable in the data set exposure_final – the variable dosage_num for numerical equivalent of the variable dosage. This will allow us to sort the data set as needed to control entries in the legend.

For example, for subject 1003 we will have dosage intervals as:

start_x	end_x	slope	start_y	end_y	dosage	dosage_num
1	40	-0.89744	0	-35	10 mg	10
40	45	-0.125	-35	-35.625	10 mg	10
45	65	-0.125	-35.625	-38.125	5 mg	5
65	80	-0.125	-38.125	-40	10 mg	10
80	130	-1.2	-40	-100	10 mg	10
130	170	0	-100	-100	10 mg	10
170	210	1	-100	-60	10 mg	10

Each of the intervals in this data set exposure_final is located entirely between two consecutive efficacy evaluations.

Step 7

Now we are almost ready with the figure for this chapter.

To control the color corresponding to each possible dosage (according to the protocol, we can have only 4 different dosages: 0 mg, 5 mg, 10 mg, or 15 mg), we should use attribute map. Let's put in this attribute map for the possible values of daily dose:

```
data attribute_map_doses;
  length id $11 value $20 linecolor $15;
  input id value linecolor;
  datalines;
  doses 0_mg gray
  doses 5_mg violet
  doses 10_mg blue
  doses 15_mg orange
  ;
run;

data attribute_map_doses;
  set attribute_map_doses;
  value = translate (value, ' ', '_');
run;
```

We need to concatenate 2 attribute maps to create a global attribute map:

```
data global_attribute_map;
  set attribute_map_responses attribute_map_doses;
run;
```

This global attribute map is below:

id	value	markersymbol	linecolor
evaluations	CR	DiamondFilled	
evaluations	PR	CircleFilled	
evaluations	SD	TriangleFilled	
evaluations	PD	SquareFilled	
evaluations	NE	StarFilled	
doses	0 mg		gray
doses	5 mg		violet
doses	10 mg		blue
doses	15 mg		orange

Now we can concatenate data set exposure_final (adding one more variable to it: part = 1) with the previously created part 2 of data set timepoints. Let's name this data set obtained after the concatenation process as timepoints_with_exposure.

```
data timepoints_with_exposure;
  set
    exposure_final (in = about_exposure)
    timepoints (in = about_timepoints where = (part = 2))
    ;
  if about_exposure then part = 1;
run;
```

The obtained SAS data set timepoints_with_exposure is almost ready to be used in PROC SGPLOT, but one minor data rearrangement is needed. To control the legend, we need to sort this data set timepoints_with_exposure by part descending dosage_num response_num:

```
proc sort data = timepoints_with_exposure;
  by part descending dosage_num response_num;
run;
```

In the previous chapters of this presentation we used SERIES and SCATTER statements to produce the needed figures. This time we are going to use VECTOR and SCATTER statements inside of PROC SGPLOT:

```
proc sgplot data = timepoints_with_exposure dattrmap = global_attribute_map
  noautolegend;
  vector x = end_x y = end_y
    /
    noarrowheads
    xorigin = start_x
    yorigin = start_y
    lineattrs = (thickness = 3 pattern = solid)
    group = dosage nomissinggroup
    attrid = doses
    name = 'the_doses'
    ;
  scatter x = evaluation_day y = response_y
    /
    group = response nomissinggroup
    attrid = evaluations
    name = 'the_evaluations'
    markerattrs = (size = 10 color = black)
    ;
  keylegend 'the_doses'
    /
    exclude = (' ')
    across = 1
    border
    location = inside
    position = topright
    outerpad = (top = 10 px)
    title = 'Daily Dosage' titleattrs = (size = 9)
    valueattrs = (size = 8)
    ;
  keylegend 'the_evaluations'
    /
    exclude = (' ')
    across = 1
    border
    location = inside
    position = SW
    outerpad = (bottom = 10 px)
    title = 'Results of Timepoint Assessments' titleattrs = (size = 9)
    valueattrs = (size = 8)
    ;
  refline -30 0 20 / axis = Y lineattrs = (color = ligr pattern = dash);
```

Our figure after submitting the SAS code above can be seen below. This enhanced output looks much more informative than the output from the simple spider plot without any enhancement that we had in Chapter 1.

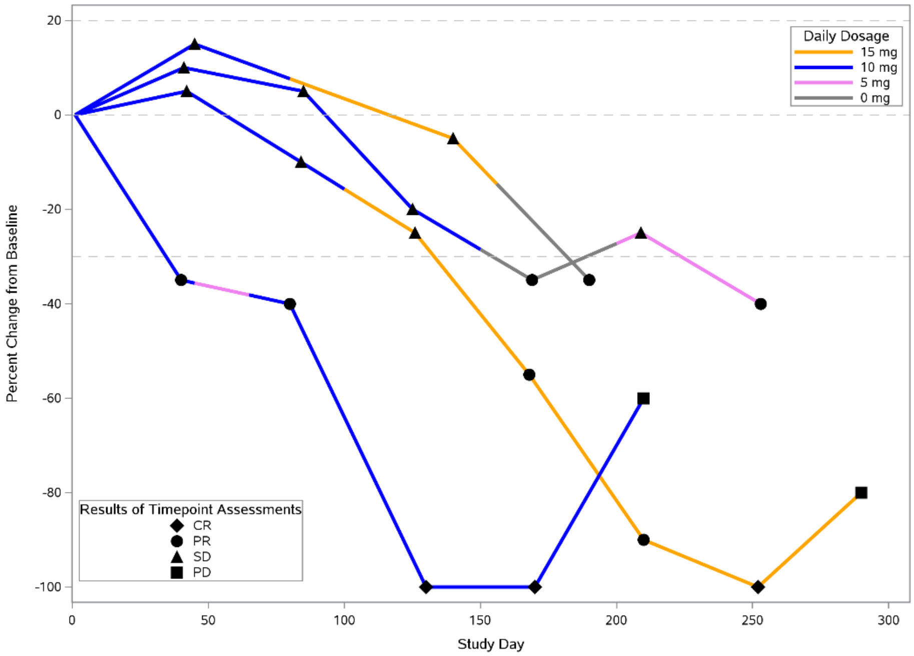

Figure 4. Spider plot with reference lines, results of timepoint responses, and dosage

CHAPTER 5: ADDING SUBJECT-LEVEL INFORMATION AS THE LABELS OF POLYGONAL CHAINS

The earlier produced figure may have further enhancements. It would be very helpful for a reviewer if we can identify each polygonal chain (spider leg) with information about subject ID and possibly some other subject-level information, such as age or gender. For our figure, we will create chain labels combining Subject ID and BOR (Best Overall Response). Let's assume that for this study, RECIST 1.1 evaluation criterion is in use and the study protocol requires BOR to be derived with confirmation of response. One of the efficacy ADaM data sets should keep results of BOR derivation, so there is no need to rederive it during the work on this figure; we can just extract these values from this data set. For our 4 polygonal chains (spider legs), the values of the polygonal chain labels will be:

- 1001 BOR = PR
- 1002 BOR = PR
- 1003 BOR = CR
- 1004 BOR = SD

Now we need to put the label value into the record with the needed dosing interval for each subject. This should be the chronologically latest dosing interval for each subject. To make it possible, another resorting of the previously created data set timepoints_with_exposure is required:

```
proc sort data = timepoints_with_exposure;
  by part subjid start_x;
run;
```

After that, we need to populate an additional variable (let's name it subject_level) with just the suggested values for the chronologically latest record for each subject from part 1. Finally, we are getting the final version of the data set that will be used for our enhanced spider plot. Let's name this data set final. This data set requires resorting to support putting legend entries in the necessary order. We are now ready to produce the final version of our enhanced spider plot. Before we submit our PROC SGPLOT statement, we should take care of the right-side offset of the horizontal axis. Because each spider leg is expected to have a label that occupies some additional space to the right of the end of the leg, the offsetmax option of the xaxis statement should be adjusted accordingly.

The SAS code (including statements for horizontal and vertical axes) that produces the final output is below:

```
proc sgplot data = final dattrmap = global_attribute_map noautolegend;
  vector x = end_x y = end_y
    /
    noarrowheads
    xorigin = start_x
    yorigin = start_y
    lineattrs = (thickness = 3 pattern = solid)
    group = dosage nomissinggroup
    datalabel = subject_label
    datalabelpos = right
    datalabelattrs = (color = black size = 8)
    attrid = doses
    name = 'the_doses'
    ;
  scatter x = evaluation_day y = response_y
    /
    group = response nomissinggroup
    attrid = evaluations
    name = 'the_evaluations'
    markerattrs = (size = 10 color = black)
    ;
  keylegend 'the_doses'
    /
    exclude = (' ')
    across = 1
    border
    location = inside
    position = topright
    outerpad = (top = 10 px)
    title = 'Daily Dosage' titleattrs = (size = 9)
    valueattrs = (size = 8)
    ;
  keylegend 'the_evaluations'
    /
    exclude = (' ')
    across = 1
    border
    location = inside
    position = SW
    outerpad = (bottom = 10 px)
    title = 'Results of Timepoint Assessments' titleattrs = (size = 9)
    valueattrs = (size = 8)
    ;
  refline -30 0 20 / axis = Y lineattrs = (color = ligr pattern = dash);
  xaxis
    type = linear
    label = 'Study Day' labelattrs = (size = 9 pt)
    valueattrs = (size = 8 pt)
    offsetmin = 0 offsetmax = 0.125
    ;
  yaxis
    type = linear
    label = 'Percent Change from Baseline' labelattrs = (size = 9 pt)
    valueattrs = (size = 8 pt)
```

```
         offsetmin = 0.025 offsetmax = 0.025
         ;
   run;
```

This SAS code produces the following figure:

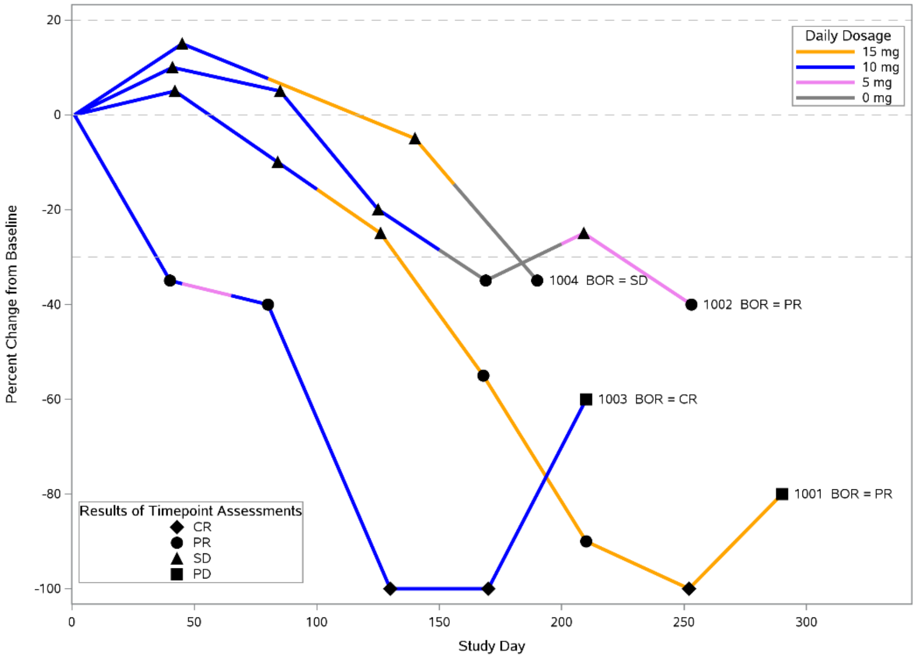

Figure 5. Spider plot with reference lines, results of timepoint responses, dosage, and subject labels

CONCLUSION

SAS has provided us the powerful tools to generate graphs using PROC SGPLOT. This paper shows how to make spider plots more informative using different types of enhancements. Similar results can be achieved using Graph Template Language technique. The enhanced spider plots are very helpful to learn the dynamic changes for tumor response in the treatment. They are recommended for relatively small numbers of subjects per figure (for example, the Dose Escalation part of Phase I studies). These enhanced figures can also be recommended for patient profiles.

REFERENCES

SAS® 9.4 ODS Graphics: Procedures Guide, Sixth Edition. Cary, NC: SAS Institute Inc.

Matange, Sanjay, and Dan Heath. 2011. Statistical Graphics Procedures by Example: Effective Graphs Using SAS®. Cary, NC: SAS Institute Inc.

Matange, Sanjay. 2016. Clinical Graphs Using SAS®. Cary, NC: SAS Institute Inc.

Harris, Kriss, and Richann Watson. 2020. SAS®Graphics for Clinical Trials by Example. Cary, NC: SAS Institute Inc

Nora H. Ruel, M.A., Paul H. Frankel, Ph.D., City of Hope Cancer Center. Graphical Results in Clinical Oncology Studies https://support.sas.com/resources/papers/proceedings16/7520-2016.pdf

Shilpakala Vasudevan, Ephicacy Lifescience Analytics. Visualization of efficacy endpoints in oncology clinical trials https://www.lexjansen.com/pharmasug-cn/2019/DV/Pharmasug-China-2019-DV85.pdf

CONTACT INFORMATION

Your comments and questions are valued and encouraged.

The full SAS code (including code for data manipulation) is available upon request.

Contact the author at:

Ilya Krivelevich
Eisai Inc.
200 Metro Blvd, Nutley, NJ 07110
Phone: 201-543-8283
ilya_krivelevich@eisai.com