SESUG Paper 281-2019

Cutting Out the Middle Man – Bypassing ADaM to Calculate Tumor Response Inputs using RECIST 1.1 Guidelines and Collected Data in SDTM Format

Amber Frazier, PPD

ABSTRACT

Confirming vendor results? Providing recon listings to the client? No time or need for ADaM? Cut out the middle man! A high-level overview on how to use CRF data in SDTM format to reconcile tumor response assessment using Response Evaluation Criteria in Solid Tumor (RECIST 1.1) Guidelines, from collecting the basic required data points to deriving the Best Overall Response. Tumor response categorization is key in monitoring patient disease progression and best overall response is frequently an analysis endpoint in Oncology studies. The RECIST 1.1 Guidelines offer a standard process for data collection and calculation for target, non-target, and new lesion progression based on assessments at timed intervals, generally 6-12 weeks apart. Efficient data analysis starts with effective data collection using simple case report forms (CRF) and ensuring all required data items are accounted for. Ideally the CRF data is structured to easily fit into SDTM format using Tumor Identification, Tumor Results and Tumor Response domains (TU, TR, and RS). The RECIST 1.1 worksheets summarize the information needed and can be formatted for SDTM translation.

INTRODUCTION

For statistical programmers, oncology can be a complex therapeutic area in which to develop analytic code. The contributing data collection can also be complex, requiring several stages of data collection at specific time points. For this reason, the Response Evaluation Criteria in Solid Tumors (RECIST) was developed in 2000 and created a standard by which the data are collected and processed, for consistency across industry contributors including government agencies, investigators, and statisticians. RECIST creates the common language between these groups and a guide to more straightforward data collection. In 2008, RECIST 1.1 was released, including a simpler data collection method using worksheets, but a more complex programmatic derivation to interpret them. In SDTM, the data is collected in 3 domains, Tumor Response (TR), Tumor Identification (TU), and Disease Response (RS). For many clinical trials the development of an analysis database is required; combining these 3 domains into analysis datasets can provide more easily programmable endpoints of overall results and disease status. But what if there is a need to derive these interpretations without an analysis database, using only raw data? This paper will touch on basic programming techniques that can be used to create or confirm the basic RECIST 1.1 criterion.

The criterion that will be discussed are applied to target lesions only. Per RECIST 1.1, five target lesions are chosen for evaluation. This is meant to be a high-level overview derivation of the RECIST 1.1 inputs for determining tumor response.

- Sum of longest diameters (SLD)
- Nadir
- Percent change from baseline
- Percent change from Nadir

1. SUM OF LONGEST DIAMETERS (SLD)

Sum of longest diameters is derived at each timepoint. This will be a sum of the target lesion long axis (mm) and lymph node short axis (mm). In SDTM, this will correspond to the results field for the diameter measurements. This value SLD can be evaluated in the same manner as any standard lab data, showing a timepoint result, baseline, change from baseline, and percent change from baseline.

There are many ways to derive a summation of data points, but simplicity is key for the beginner programmer. In SDTM format the data is collected in a vertical format where each tumor measurement is recorded in its own row, so using PROC SQL will be the way to go. Utilizing the GROUP statement, the sums can be derived per subject, per visit.

SUBJECT	VISIT	TRLNKID	TRTESTCD	TRSTRESN
101	BASELINE	R1-T01	TARGET	11
101	BASELINE	R1-T02	TARGET	13
101	BASELINE	R1-T03	TARGET	10
101	BASELINE		SLD	34
101	VISIT 1	R1-T01	TARGET	11
101	VISIT 1	R1-T02	TARGET	11
101	VISIT 1	R1-T03	TARGET	10
101	VISIT 1		SLD	32

Figure 1. Example of TR domain after creating SLD.

```
PROC SQL;
    create table CREATE_SLD as select
    SUBJECT, VISIT, VISITNUM, SUM(TRSTRESN) as SLD, "SLD" as TRTESTCD
    from SDTM.TR
    group by SUBJECT, VISIT, VISITNUM;
QUIT;
```

2. NADIR

What is Nadir? Nadir means "the lowest point". Within RECIST 1.1, Nadir refers to the smallest sum of the longest diameters value (SLD) which has occurred on-treatment prior to that timepoint. This can be derived programmatically using a couple clever techniques.

To calculate Nadir on a vertical data table like Section 1.2, choose the SLD records for the calculation. Nadir is calculated by determining the smallest sum which occurred prior to that visit and is retained until a smaller value takes its place. Using a RETAIN statement, the previously selected value can be carried vertically downward, populating the subsequent data rows for each subject. Due to the nature of the calculation, Nadir is undeterminable at baseline and will equal the baseline sum at the next evaluable visit.

SUBJECT	VISIT	VISIT NUMBER (X)	TRTESTCD	TRSTRESN	NADIR_CALC
101	BASELINE	1	SLD	34	
101	VISIT 1	2	SLD	32	34
101	VISIT 2	3	SLD	28	32
101	VISIT 3	4	SLD	30	28
101	VISIT 4	5	SLD	31	28

Figure 2. Choose SLD records to calculate Nadir.

In order to evaluate the Nadir, a temporary field is needed to hold the current lowest value. Because the Nadir is the lowest value prior to the visit, the values that need to be considered are all previous records.

To simplify the calculation, organize the data into horizontal rows by transposing by subject. For this example, a visit number is assigned so that in comparison to the variable name it can be used to determine previous visit values. For programming ease, add a prefix which adds meaning to the variable name, like "VIS_". Using the transposed dataset, an array can be used to create a dynamic calculation of previous minimum value.

SUBJECT	VIS_1	VIS_2	VIS_3	VIS_4	VIS_5
101	34	32	28	30	31

Figure 3. The dataset is transposed before calculating Nadir.

Create an array with the visit numbers by searching for and removing the assigned prefix of the ID columns. Use the VNAME function to get the variable name and convert it into a useable visit number value by stripping the prefix.

Nadir(x) = MINIMUM (Nadir(x-1), SLD(x-1)) where non-missing results are considered. If one result is missing, then it should be ignored in the calculation.

Nadir(x) where x=1, or baseline, is null. There is no previous lowest value or SLD, x=0 does not exist.

Nadir(x) where x=2 will be equal to the SLD at baseline, because Nadir is missing at baseline and only non-missing results are considered.

Nadir(x) where x>2 is the minimum non-missing value of the previous Nadir value and the current value in check, SLD(x-1). Nadir(x) = MIN(Nadir(x-1), SLD(x-1)). If SLD is missing, then the previous Nadir will be retained.

Picture a pyramid type calculation to determine the lowest value from a previous visit. Using the transposed dataset, output one row for each visit that needs a calculation. Then only values from previous visits are considered (shown shaded below). Using arrays and the VNAME function, this can be accomplished by comparing the VISITNUM with the suffix on the ID column variable name and choose values where the variable suffix is less than the visit number.

SUBJECT	VIS_1	VIS_2	VIS_3	VIS_4	VIS_5	VISITNUM	PREVIOUS NADIR	CURRENT NADIR
101	34	32	28	30	31	**1**	-	-
101	34	32	28	30	31	**2**	34	34
101	34	32	28	30	31	**3**	34	32
101	34	32	28	30	31	**4**	32	28
101	34	32	28	30	31	**5**	28	28

Figure 4. The shaded areas show the available choices for Nadir for each VISITNUM record.

…<open code>…

```
do i=1 to dim(VIS);                  *(VIS) is the array of visit ID variable names;
retain NADIR;                        *NADIR is the running lowest value prior to the current
                                     visit;
  if VIS[i] ne . then do;            *VIS[i] is the SLD of the visit ID variable;
  VISITNUM = VNUM[i];                *VNUM[i] is the converted visit number created using
                                     VNAME. Use this to determine the inputs for the Nadir
                                     evaluation.
   if i = 1 then NADIR = .;          *If occurrence is the first, there is no Nadir;
   else if i > 1 then do;            *Else if the visit is after baseline;
   PREV = VIS [i-1];                 *Set lag value, this becomes the evaluable SLD for the
                                     current record;
```

```
    NADIR = MIN(PREV, NADIR);   *Determine if Nadir or the previous SLD is the minimal populated
                                value.  Note: SAS will give log warnings for missing values.
    TRTESTCD = 'NADIR';         *Set a TRTESTCD value for the record;
     TRSTRESN = NADIR;          *Option to create the result value from the Nadir derivation;
    end; output;                *Output this visit record. Since i=1 to dim(VIS) will give one
                                record for each visit, this data can be easily rejoined with the
                                original input dataset.
```

…<open code>…

SUBJECT	VISIT	VISITNUM (X)	TRTESTCD	TRSTRESN
101	BASELINE	1	NADIR	34
101	VISIT 1	2	NADIR	34
101	VISIT 2	3	NADIR	32
101	VISIT 3	4	NADIR	28
101	VISIT 4	5	NADIR	28

Figure 5. Nadir is evaluated and a test code (TRTESTCD) is created for the TR domain.

3. PERCENT CHANGE FROM BASELINE

The calculated percent change from baseline will aid in the determination of tumor response when applying RECIST 1.1 criteria. The percent change of baseline for RECIST is generally calculated as:

((VALUE2 / VALUE1) - 1) * 100 = % CHANGE

In this example, VALUE1 is representative of the baseline SLD, and the VALUE2 represents the visit for which the calculation is made.

In this example a simple retain statement can be applied to calculate the percent change from baseline as shown below. While applying the retain function, it is important to reset the values after each subject's calculation to ensure no erroneous values are chosen between subjects. In Section 1.2, the SLD is calculated using PROC SQL. For percent change from baseline, the records for SDL are chosen for the derivation. Sort by subject and visit and apply the calculations. Change from baseline will not be calculated for baseline.

SUBJECT	VISIT	--TESTCD	--STRESN	BASELINE SLD	CHANGE from BASELINE (%)
101	BASELINE	SLD	34	34	
101	VISIT 1	SLD	32	34	-5.88%
101	VISIT 2	SLD	28	34	-17.65%

```
DATA PCNT_CHG_SLD;
    set CREATE_SLD;
    by SUBJECT;
    RETAIN BL_SLD;
    if first.SUBJECT then BL_SLD = SLD;
    else do;
        PCNT_CHG = ((SLD / BL_SLD) - 1) * 100;
    end;
RUN;
```

4. PERCENT CHANGE FROM NADIR

The calculated percent change from Nadir will aid in the determination of tumor response when applying RECIST 1.1 criteria. The percent change of baseline for RECIST is generally calculated as:

((VALUE2 / VALUE1) - 1) * 100 = % CHANGE

In this example, VALUE1 is representative of the current Nadir, and the VALUE2 represents the current SLD. This calculation will be like the calculation applied in Section 3, percent change from baseline.

In this example a lag or retain statement can be applied to calculate the percent change from Nadir. However, due to the nature of the calculation, in this example the percent change from Nadir can only be calculated in the timepoints after Nadir has been established. For that reason, a line of code which derives the percent change can simply be added to the DATA step array used in Section 2.

SUBJECT	VISIT	TRTESTCD	TRSTRESN	NADIR	CHANGE from NADIR (%)
101	BASELINE	SLD	34		
101	VISIT 1	SLD	32	34	-5.88%
101	VISIT 2	SLD	28	32	-12.5%
101	VISIT 3	SLD	30	28	7.14%
101	VISIT 4	SLD	31	28	10.71%

Figure 6. Calculate % Change from Nadir for each SLD at each visit.

…<open code>…

```
NADIR = MIN(PREV, NADIR);
```

*After setting NADIR in the code from Section 2, add a line of code to create the change value.

```
CHG_NADIR = ((VIS[i] / NADIR) - 1) * 100;
```

*Derive the change from Nadir as a new column;

…<open code>…

CONCLUSION

RECIST 1.1 can be complex set of evaluations to determine the overall response and disease progression, but the programming calculations for the input parameters can be simplified using basic SAS functions and data steps no matter what style of data collection is used. Once these derivations are calculated, they can be used to determine the best response per time point as well as the best overall response (BOR). Calculating the responses takes careful consideration, be sure to understand each of the cases and how they should be interpreted.

REFERENCES

[1]. Eisenhauera, E.A., Therasseb, P., Bogaertsc, J., Schwartzd, L.H., Sargente, D., Fordf, R., Danceyg, J., Arbuckh, S., Gwytheri, S., Mooneyg, M., Rubinsteing, L., Shankarg, L., Doddg, L., Kaplanj, R., Lacombec, D., Verweijk, J. 2009. “New response evaluation criteria in solid tumours: Revised RECIST guideline (version 1.1).” European Journal of Cancer.

Available at https://ctep.cancer.gov/protocolDevelopment/docs/recist_guideline.pdf

RECOMMENDED READING

- *PharmaSUG 2019 - Paper BP-227 From Lesion size to Best Response - Implementing RECIST through programming Ankit Pathak, Rang technologies*
- *PharmaSUG 2017 - Paper AD24 SAS Macro for Derivation of Best Overall Response per RECIST 1.1 Bob Zhong, Jiangfan Li, Hong Xie, Peter De Porre, Kenneth Maahs, Kyounghwa Bae Johnson & Johnson, Spring House, PA*

- *Teslenko, Iryna & Belotserkovskiy, Maxim & Kumar, Akhil. (2015). Common Pitfalls of RECIST 1.1 Application in Clinical Trials. Journal for Clinical Studies. 7. 26-32.*

CONTACT INFORMATION

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

Amber Frazier
a.frazier.stats@gmail.com

DISCLAIMER

The content of this paper is the work of the author and do not necessarily represent the opinions, recommendations, or practices of PPD.