

SAS Macros to check and solve common problems in Oncology studies

Julia Chen, Huchison MediPharma

1. Abstract

When conducting an Oncology trial, our company will invest numerous time and money on it, meanwhile, the designs and endpoints are more difficult and complex than other therapeutic areas. In this situation, it is very important for us to oversight, check whether the source data has recorded information timely and correctly.

In this paper, several macros will be discussed to check the source data's reliability authenticity and usability based on CDISC SDTM. These macros are: 1) Judge whether TR summary statistics is correct if the source data have collected statistics(Sum of Diameter, Percent Change from Baseline in Sum of Diameter, etc); 2) Judge the correctness of tumor assessment on target, non-target and new lesion; 3) Judge whether the overall response recorded in RS is consistence with TR using the Response Evaluation Criteria in Solid Tumors version 1.1 (RECIST 1.1); 4) Judge if RSDTC is no earlier than the corresponding TRDTC at the same time point; 5) BOR without confirmation on CR/PR derivation and validation. These macros will help us detect the data issues quickly and timely to ensure the data quality and efficacy accuracy.

2. Introduction

Based on RECIST (Response Evaluation Criteria in Solid Tumor) v1.1 and SDTM structure, tumor assessment will be recorded in SDTM domains:

- 1) TU : Tumor Identification
- 2) TR : Tumor Results
- 3) RS: Response

with the tumor response on target/non-target/new lesion to derive overall response at each visit and BOR(Best Overall Response) during the study.

Tumor assessments are very critical to Oncology efficacy endpoints: the correctness of tumor results (CR, PR, SD, PD, etc) and event/censor date will affect a lot on whether the trial can be succeed or failed. So, the data quality and variable logic required us to check through carefully.

The main focus of the paper is the application of the SAS macros on checking the consistency of tumor assessment based on different SDTM domains, overall evaluation from target, non-target and new lesion and deriving BOR without confirmation.

3. Macros Application

3.1 Check the correctness of Sum of Diameters

Based on RECIST 1.1, when performing the evaluation of target lesions, we need to compare the sum of diameters of each visit with baseline or the smallest sum on study for Partial Response (PR) or Progressive Disease (PD). Whether the sum of diameters is calculated correctly by the investigator will affect the evaluation result of target lesion.

We can select *TRTESTCD*= 'DIAMETER', calculate the sum of *TRSTRESN* per subject per visit per evaluator when *TRGRPID* = 'TARGET' and compare it with *TRSTRESN* when *TRTESTCD*= "SUMDIAM" or the corresponding variable from other efficacy dataset if the sum of diameters is not recorded in TR.

```
%target_sum_check(inds1=tr, /*Input the dataset which has the sum records*/
cond1 = %str(where TRGRPID=' TARGET' and TRTESTCD = 'SUMDIAM'), /*Selection criteria from above dataset1*/
vis1 = visitnum, /*Visit variable name1*/
diam1 = trstresn, /*Sum of Diameter variable name*/
eval1 = TREVALLID, /*Evaluator variable1*/
inds2=tr, /*Input the dataset which has the individual records*/
cond2 = %str(where TRGRPID = "TARGET" and TRTESTCD = "DIAMETER"), /*Selection criteria from above dataset2*/
vis2 = visitnum, /*Visit variable name2*/
diam2 =trstresn, /*Diameter variable name*/
eval2 = TREVALLID, /*Evaluator variable1*/
outds=final2, /*Output dataset name*/
crit=0.1): /*Your tolerance of the deviation ratio*/
```

After calling the macro (*See attachment 1*), the final dataset will list the records when 'The deviation is too big!', 'Sum of Diameters has been recorded but individual records hasn't shown.' or 'Individual diameters have been recorded but Sum of Diameters haven't.' as below:

USUBJID	sum_of	eval	visn	sum_cal	judge	diff_pct
402-S3001	25	INVESTIGATOR		.	Sum has been recorded while individual record not	
410-S3004	0	INVESTIGATOR		.	Sum has been recorded while individual record not	

And from above result, we can go back to query DM on the cause of inconsistency. Percent Change from Baseline can be checked followed the same logic.

3.2 Check the consistency of different SDTM domain

3.2.1 Check the consistency of non-target/target lesion between TR and RS

If the datasets have retrieved the correct information, the non-target lesion should be recorded the same from TR to RS per subject per visit per evaluator. Based on RECIST 1.1, the assessment of non-target lesion in RS will be classified as *CR*, *Non-CR/Non-PD*, *Not evaluated*, *PD*, while the assessment in TR is not that standardized, especially the study has records from both independent assessor and investigator. To solve this situation, we can use format to preprocess the data to make there is a link between two datasets as below:

```

proc format;
value $trntgt
'ABSENT' = 1
'PRESENT' = 4
'Not assessable' = 3
'UNEQUIVOCAL PROGRESSION' = 4
'QUIVOCAL PROGRESSION' = 2
;
run;

proc format;
value $rsntgt
'CR' = 1
'NON-CR/NON-PD' = 2
'NE' = 3
'PD' = 4
;
run;

```

In above formats, we have records from both central reading and investigator, when central reading record 'ABSENT', it means complete response when there is a non-target lesion in prior visit while 'PRESENT' is consistence with progression disease, if not recorded then it is neither CR nor PD, when the records are from investigator as 'UNEQUIVOCAL PROGRESSION', it corresponds to PD, while 'QUIVOCAL PROGRESSION' corresponds to 'NON-CR/NON-PD', after calling the macro (*See attachment 2*),:

```

%nontarget_check(inds1=tr, /*Input the tr dataset*/
cond1 = %str(where TRGRPID='NON-TARGET' and VISIT='SCREENING'),/*Selection criteria from dataset1*/
vis1 = VISITNUM, /*Visit variable name1*/
resp1 =TRSTRESC, /*Assessment variable name1*/
eval1=TREVALID, /*Evaluator variable1*/
inds2=rs, /*Input the RS dataset*/
cond2 = %str(where RSTESTCD='NTRGRESP' and not missing(visitnum)),/*Selection criteria from dataset2*/
vis2 = VISITNUM, /*Visit variable name2*/
resp2 =RSSTRESC, /*Assessment variable name2*/
eval2=RSEVALID, /*Evaluator variable2*/
outds=final); /*Output dataset name*/

```

And the output dataset will list the inconsistency as 'Only in RS', 'Only in TR', 'Not Equal'

usubjid	visitn	eval	tr_rslt	TRSTRESC	RSSTRESC	rs_rslt	judge
H3-01-S3002	9.01	INVESTIGATOR	4	UNEQUIVOCAL PROGRESSION	NON-CR/NON-PD	2	Not Equal
H3-01-S3005	5.01	RADIOLOGIST 1	4	ABSENT	NON-CR/NON-PD	2	Not Equal
H3-01-S3005	7.01	RADIOLOGIST 1	4	ABSENT	NON-CR/NON-PD	2	Not Equal
H3-01-S3005	9.01	RADIOLOGIST 1	4	ABSENT	NON-CR/NON-PD	2	Not Equal
H3-01-S3005	11.01	RADIOLOGIST 1	4	ABSENT	NON-CR/NON-PD	2	Not Equal
H3-01-S3005	13.01	RADIOLOGIST 1	4	ABSENT	NON-CR/NON-PD	2	Not Equal
H3-01-S3005	19.01	RADIOLOGIST 1	4	ABSENT	NON-CR/NON-PD	2	Not Equal
H3-01-S3006	13.01	INVESTIGATOR			NE	3	Only in RS
H3-01-S3007	3.01	RADIOLOGIST 1	4	PRESENT			Only in TR

By using above output, we can query site if they have recorded the correct information and if not, what is the best way to make it right and prevent the same issue in future.

Target lesion can be checked as the same logic.

3.2.3 Check the consistency of new lesion between TU and RS

As SDTMIG writes: the TU domain represents data that uniquely identifies tumors, a record in the TU domain contains the following information: a unique tumor ID value; anatomical location of the tumor; method used to identify the tumor; role of the individual identifying the tumor; and

timing information. If there is a new lesion record in TU with location, there should be a record in RS for the same patient/visit/evaluator.

In the example, we have new lesion recorded with *RSTESTCD* = 'NEWLPROG' from independent assessor and *RSTESTCD* = 'NEWLSN' and *RSORRES* = 'Y' from investigator in RS while both have *TUORRES* = 'NEW' in TU.

```
%new_check(inds1=tu,
            cond1=%str(where TUORRES='NEW' and not missing(TULOC)),
            vis1=visitnum,
            eval1 = TUEVALID,
            inds2=rs,
            cond2=%str(where (RSTESTCD='NEWLSN' and rsorres="Y") or (RSTESTCD='NEWLPROG') ),
            vis2=visitnum,
            eval2 = RSEVALID,
            outds=final);
```

After setting below criteria in macro(See attachment 3), we can get output with records when inconsistency exists as below.

subjid	visitr	eval	RS	TU	judge
H3-01-S3003	5.01	RADIOLOGIST 2	NE		RS Overall shows new lesion But no records in TU
H3-01-S3003	7.01	RADIOLOGIST 2	NE		RS Overall shows new lesion But no records in TU
H3-01-S3003	13.01	RADIOLOGIST 2		EQUIVOCAL	New lesion has been recorded in TU, but overall doesnt show in RS
H3-01-S3004	5.01	RADIOLOGIST 1	UNEQUIVOCAL		RS Overall shows new lesion But no records in TU
H3-01-S3004	7.01	RADIOLOGIST 1	UNEQUIVOCAL		RS Overall shows new lesion But no records in TU
H3-01-S3004	9.01	RADIOLOGIST 1	UNEQUIVOCAL		RS Overall shows new lesion But no records in TU
H3-01-S3004	11.01	RADIOLOGIST 1	UNEQUIVOCAL		RS Overall shows new lesion But no records in TU
H3-01-S3004	11.01	RADIOLOGIST 2	UNEQUIVOCAL		RS Overall shows new lesion But no records in TU
H3-01-S3013	7.01	RADIOLOGIST 2	EQUIVOCAL		RS Overall shows new lesion But no records in TU
H3-01-S3013	9.01	RADIOLOGIST 2	EQUIVOCAL		RS Overall shows new lesion But no records in TU

3.3 Check Overall Response based on Target/Non-Target/New lesion at each visit

The overall response for each patient at certain time point will depend on the findings of both target and no-target lesions, and also will include the occurrence of any new lesion. The below is the summary table of overall response at certain time point per RECIST guideline.

Target lesions	Non-target lesions	New lesions	Overall response
CR	CR	No	CR
CR	Non-CR/non-PD	No	PR
CR	Not evaluated	No	PR
PR	Non-PD or not all evaluated	No	PR
SD	Non-PD or not all evaluated	No	SD
Not all evaluated	Non-PD	No	NE
PD	Any	Yes or No	PD
Any	PD	Yes or No	PD
Any	Any	Yes	PD

CR = complete response, PR = partial response, SD = stable disease, PD = progressive disease, and NE = inevaluable.

Due to the different design of CRF, the input term of 'CR', 'PR', 'SD' and so on are different. To avoid the difference, we will use 'proc format' to set a standard in different studies as below:

```

proc format;
  value $tgt
    'CR' = 1
    'PR' = 2
    'SD' = 3
    'PD' = 4
    'NA' = 5
;
run;

proc format;
  value $ntgt
    'CR' = 1
    'Non-CR/Non-PD' = 2
    'PD' = 3
    'Not assessable' = 4
;
run;

proc format;
  value $new
    'N' = 1
    'Y' = 2
    'NE' = 3
;
run;

```

Based on the algorithm in above table, we can call macro:

usubjid	visitnum	rsevalid	tgt	ntgt	new	overall	judge
01-S3001	5.01	RADIOLOGIST 1	SD	NON-CR/NON-PD	N	PD	SD
01-S3003	3.01	RADIOLOGIST 2	SD		Y	SD	PD
01-S3003	5.01	RADIOLOGIST 2	SD		NE	SD	
01-S3003	7.01	RADIOLOGIST 2	SD		NE	SD	
01-S3013	7.01	RADIOLOGIST 1	PR	NON-CR/NON-PD	N	PD	PR

From above result, we can see the correct overall response for the first record should be SD instead of PD and following records the same.

3.4 Check if the assessment date is no earlier than image date

All measurements should be recorded in metric notation, using calipers if clinically assessed. The same method of assessment and the same technique should be used to characterize each identified and reported lesion at baseline and during follow-up. No matter what methods (X-ray, CT, MRI, etc) is used to measure, the assessment date should be later than or on the latest image date within a patient/visit/evaluator.

By calling the macro %rstrdate_check(See attachment 4), we can get all the records with assessment data (RSBTC) earlier than image date(TRDTC) as below:

USUBJID	RSEVALID	VISITNUM	CATE	rsdt	maxtrdt
S3002	RADIOLOGIST 1	3.01	TARGET	2016-05-11	2016-05-13
S3002	RADIOLOGIST 1	5.01	TARGET	2016-07-07	2016-07-08
S3002	RADIOLOGIST 1	7.01	TARGET	2016-08-30	2016-08-31
S3002	RADIOLOGIST 1	9.01	TARGET	2016-10-25	2016-10-26
S3002	RADIOLOGIST 2	3.01	TARGET	2016-05-11	2016-05-13

3.5 Derive BOR without confirmation on CR/PR

The best overall response is determined once all the data for the patient is known. The macro is generated when s where confirmation of complete or partial response IS NOT required. The

special note is that when SD is believed to be best response, it must also meet the protocol specified minimum time from baseline. If the minimum time is not met when SD is otherwise the best time point response, the patient's best response depends on the subsequent assessments.

After selecting *RSTESTCD*='OVRLRESP' and the minimum time from baseline to SD should be 53 days, we call below macro(*See attachment 5*):

```
%bor(inds=rs, cond1=%str(where RSTESTCD="OVRLRESP"), dura = 53, eval =RSEVALID, outds=final);
```

And get the result of BOR per patient per evaluator:

USUBJID	RSEVALID	min	AVALC
01-S3001	INVESTIGATOR	3	SD
01-S3001	RADIOLOGIST 1	3	SD
01-S3001	RADIOLOGIST 2	3	SD

4. Conclusion

It is very important for programmers, statisticians and DMs to understand the therapeutic knowledge and characteristics, so that to ensure the programming logic and data quality. Besides the points the paper has checked, there are still a lot of aspects can be checked combining SDTM and RECIST 1.1.

5. Reference

New response evaluation criteria in solid tumors: Revised RECIST guideline (version 1.1)

SDTM Version 1.3 SDTM Implementation Guide Version 3.2

Attachment 1

```
%macro target_sum_check(inds1=sdtm.tr,                /*Input the dataset which has the sum records*/
                        cond1 = ,                    /*Selection criteria from above dataset1*/
                        vis1 = visitnum,            /*Visit variable name1*/
                        diam1 = trstresn,          /*Sum of Diameter variable name*/
                        eval1=,                    /*Evaluator variable1*/
                        inds2=,                    /*Input the dataset which has the individual records*/
                        cond2 = ,                    /*Selection criteria from above dataset2*/
                        vis2 = visitnum,            /*Visit variable name2*/
                        diam2 =trstresn,          /*Diameter variable name*/
                        eval2=,                    /*Evaluator variable1*/
                        outds=,                    /*Output dataset name*/
                        crit=);                    /*Your tolerance of the deviation*/

data rs;
  set &inds1;
run;

data tr;
  set &inds2;
run;

data sum_crf(rename=(&diam1 = sum_crf) rename=(&vis1 = visitn) %if %length(&eval1) > 0 %then %do; rename=(&eval1 = eval) %end.);
  set rs;
  &cond1;
  keep usubjid &vis1 &diam1 %if %length(&eval1) > 0 %then %do; &eval1 %end;;
run;

proc sort data = sum_crf;
  by usubjid visitn %if %length(&eval1) > 0 %then %do; eval %end;;
run;

data sum_cal_t(rename= (&vis2 = visitn) rename= (&diam2 = diameter) %if %length(&eval2) > 0 %then %do; rename=(&eval2 = eval) %end.);
  set tr;
  &cond2;
  keep usubjid &vis2 &diam2 %if %length(&eval2) > 0 %then %do; &eval2 %end;;
run;

proc sql;
  create table sum_cal as
  select distinct usubjid, visitn, sum(diameter) as sum_cal %if %length(&eval2) > 0 %then %do; ,eval %end;
  from sum_cal_t group by usubjid,visitn %if %length(&eval2) > 0 %then %do; ,eval %end;;
quit;

data &outds;
  merge sum_crf(in=a) sum_cal(in=b);
  by usubjid visitn %if %length(&eval2) > 0 %then %do; eval %end;;
  length judge $50.;
  if rmiss(sum_crf,sum_cal) ^=2;
  if a and not b then judge = "Sum has been recorded while individual record not";
  if b and not a then judge = "Individual has been shown while sum not";
  if not missing(sum_cal) and not missing(sum_crf) and sum_cal ne 0 then diff_pct = (sum_crf - sum_cal)/sum_cal;
  if diff_pct ne . and abs(diff_pct) > &crit then judge = "Deviation is too big!";
  if not missing(judge);
run;

%mend;
```

Attachment 2

```
%macro nontarget_check(inds1=,          /*Input the dataset which has the individual records*/
                      cond1 = ,        /*Selection criteria from above dataset1*/
                      vis1 = ,         /*Visit variable name1*/
                      resp1 = ,        /*Variable name*/
                      eval1=,         /*Evaluator variable1*/
                      inds2=,         /*Input the dataset which has the overall response*/
                      cond2 = ,        /*Selection criteria from above dataset2*/
                      vis2 = ,         /*Visit variable name2*/
                      resp2 = ,        /*Variable name 2*/
                      eval2=,         /*Evaluator variable2*/
                      outds=);         /*Output dataset name*/

data tr;
  set &inds1;
run;

data rs;
  set &inds2;
run;

data tr_ntgt(rename = (&vis1 = visitn) %if %length(&eval1) > 0 %then %do; rename=(&eval1 = eval) %end.);
  set tr;
  &cond1.;
  if not missing(&vis1);
  temp = put(&resp1,$trntgt.);
  if temp = "1" then tr_ntgt_rslt = 1;
  if temp = "2" then tr_ntgt_rslt = 2;
  if temp = "3" then tr_ntgt_rslt = 3;
  if temp = "4" then tr_ntgt_rslt = 4;
  keep usubjid &vis1 &resp1 tr_ntgt_rslt %if %length(&eval1) > 0 %then %do; &eval1 %end;;
run;

proc sql;
  create table ntgt_cal as
  select distinct usubjid, visitn, max(tr_ntgt_rslt) as tr_rslt, &resp1 %if %length(&eval1) > 0 %then %do; ,eval %end;
  from tr_ntgt group by usubjid,visitn %if %length(&eval1) > 0 %then %do; ,eval %end;
  order by usubjid,visitn %if %length(&eval1) > 0 %then %do; ,eval %end;;
quit;

data rs_ntgt(rename = (&vis2 = visitn) /*rename = (rsvisit = visit)*/%if %length(&eval2) > 0 %then %do; rename=(&eval2 = eval) %end.);
  set rs;
  &cond2;
  temp = put(&resp2,$rsntgt.);
  if temp = "1" then rs_rslt = 1;
  if temp = "2" then rs_rslt = 2;
  if temp = "3" then rs_rslt = 3;
  if temp = "4" then rs_rslt = 4;
  keep usubjid &vis2 &resp2 rs_rslt %if %length(&eval2) > 0 %then %do; &eval2 %end;;
run;

proc sort data=rs_ntgt nodupkey;
  by usubjid visitn %if %length(&eval2) > 0 %then %do; eval %end;;
run;

data &outds;
  merge ntgt_cal(in=a) rs_ntgt(in=b);
  by usubjid visitn %if %length(&eval2) > 0 %then %do; eval %end;;
  if a and not b then judge = "Only in TR";
  if b and not a and &resp2 ne "Not assessable" then judge = "Only in RS";
  if a and b and (rs_rslt ^= tr_rslt) then judge = "Not Equal";
  if not missing(judge);
run;

%mend;
```

Attachment 3

```
%macro new_check(inds1=, cond1=, vis1=, eval1=, inds2=, cond2=, vis2=, eval2=, outds=);
data tu_new(rename=(&vis1=visitn) %if %length(&eval1) > 0 %then %do; rename=(&eval1 = eval) %end.);
  set &inds1;
  &cond1.;
  keep usubjid TU &vis1 %if %length(&eval1) > 0 %then %do; &eval1 %end; ;
run;

proc sort data=tu_new nodupkey;
  by usubjid visitn %if %length(&eval2) > 0 %then %do; eval %end;;
run;

data rs_new(rename=(&vis2=visitn) %if %length(&eval2) > 0 %then %do; rename=(&eval2 = eval) %end.);
  set &inds2;
  &cond2.;
  keep usubjid RSSTRESC &vis2 %if %length(&eval2) > 0 %then %do; &eval2 %end;;
run;

proc sort data=rs_new nodupkey;
  by usubjid visitn %if %length(&eval2) > 0 %then %do; eval %end;;
run;

data &outds;
  merge rs_new(in=a) tu_new(in=b);
  by usubjid visitn %if %length(&eval2) > 0 %then %do; eval %end;;
  length judge $100;
  if b and not a then judge='New lesion has been recorded in TU, but overall doesnt show in RS';
  if a and not b then judge='RS Overall shows new lesion But no records in TU';
  if not missing(judge) and not missing(visitn);
run;
%mend;
```

Attachment 4

```
%macro ovrresp(inds=,tgt_cd=,ntgt_cd=,new_cd=,all_cd=,visitn=,var=,eval=,outds=);
data rs;
    set &inds;
run;

data tgt ntgt new ovlr;
    set rs;
    if RSTESTCD = "&tgt_cd" then output tgt;
    if RSTESTCD = "&ntgt_cd" then output ntgt;
    if RSTESTCD = "&new_cd" then output new;
    if RSTESTCD = "&all_cd" then output ovlr;
    keep usubjid &visitn &var %if %length(&eval) > 0 %then %do; &eval %end;;
run;

%sort(inds=tgt,outds=tgt, by =%str(usubjid &visitn ));
%sort(inds=ntgt,outds=ntgt, by =%str(usubjid &visitn ));
%sort(inds=new,outds=new, by =%str(usubjid &visitn ));
%sort(inds=ovlr,outds=ovlr, by =%str(usubjid &visitn ));

%if %length(&eval) > 0 %then %do;
%sort(inds=tgt,outds=tgt, by =%str(usubjid &visitn &eval));
%sort(inds=ntgt,outds=ntgt, by =%str(usubjid &visitn &eval));
%sort(inds=new,outds=new, by =%str(usubjid &visitn &eval));
%sort(inds=ovlr,outds=ovlr, by =%str(usubjid &visitn &eval));
%end;

data &outds;
    merge tgt(rename = (&var = tgt)) ntgt(rename = (&var = ntgt))
          new(rename = (&var = new)) ovlr(rename = (&var = overall));
    by usubjid &visitn %if %length(&eval) > 0 %then %do; &eval %end;;
    length judge $50.;
    tgtn = put(tgt,$tgt.);
    ntgtn = put(ntgt,$ntgt.);
    newn = put(new,$new.);
    if ntgtn="3" then judge = "Progression of Disease (PD)";
    if tgtn="4" then judge = "Progression of Disease (PD)";
    if newn = "2" then judge = "Progression of Disease (PD)";
    if tgtn="3" and ntgtn ne "3" and newn = "1" then judge = "Stable Disease (SD)";
    if tgtn="2" and upcase(ntgtn) ne "3" and newn = "1" then judge = "Partial response (PR)";
    if tgtn="5" and upcase(ntgtn) ne "3" and newn = "1" then judge = "Not assessable";
    if tgtn="3" and ntgtn ne "3" and missing(newn) then judge = "Stable Disease (SD)";
    if tgtn="2" and upcase(ntgtn) ne "3" and missing(newn) then judge = "Partial response (PR)";
    if overall ne judge;
run;
%mend;
```

Attachment 5

```
%macro bor (inds=, cond1=, dura=, eval=, outds=);

data rs1;
  set &inds;
  &cond1.;
run;

proc sort data=rs1;
  by usubjid %if %length(&eval) > 0 %then %do; &eval %end; visitnum;
run;

data rs2_1;
  merge rs1(in=a) ads1(in=b);
  by usubjid;
  if a and b;
  rsltn = put(RSSTRESC, $orslt.);
  rsdt = input(rsdtc, yymmdd10.);
  format rsdt yymmdd10.;
run;

/*Remove records after first PD*/
data rs2pd;
  set rs2_1;
  where rsltn = "4";
run;

data rs2pd_1(rename=(rsdt=cutday));
  set rs2pd;
  by usubjid %if %length(&eval) > 0 %then %do; &eval %end; visitnum;
  %if %length(&eval) > 0 %then %do; if first.RSEVALID;%end;
  %if %length(&eval) <= 0 %then %do; if first.usubjid;%end;
  keep usubjid rsdt %if %length(&eval) > 0 %then %do; &eval;%end;;
run;

data rs2;
  merge rs2_1 rs2pd_1;
  by usubjid %if %length(&eval) > 0 %then %do; &eval;%end;;
  if not missing(cutday) and rsdt > cutday then delete;
  if not missing(rsltn) then a = input(rsltn, best.);
run;

proc sql;
  create table bor_1 as
  select *, min(a) as min
  from rs2 group by usubjid%if %length(&eval) > 0 %then %do; ,&eval;%end;;
quit;
```

```

/*Check those with SD has the duration from randomization day longer than protocol specified*/
data sd nonsd;
  set bor_1;
  if min = 3 then output sd;
  else if min ^= 3 then output nonsd;
run;

data sd_1;
  set sd;
  if a = min then diff = rsdt - randdt;
run;

proc sql;
  create table sd_2 as
  select *, max(diff) as maxdiff
  from sd_1 group by usubjid%if %length(&eval) > 0 %then %do; ,&eval;%end;;
quit;

data sd_3_1 sd_3_2;
  set sd_2 ;
  if maxdiff < &dura then output sd_3_1;
  if maxdiff >= &dura then output sd_3_2;
run;

proc sql;
  select count(*) into: N
  from sd_3_1;
quit;
%put &N;

%if &N = 0 %then %do;
proc sort data = sd_3_2(keep = usubjid min %if %length(&eval) > 0 %then %do; &eval %end;) out = final_2 nodupkey;
  by usubjid %if %length(&eval) > 0 %then %do; &eval; %end;;
run;

proc sort data = nonsd(keep = usubjid min %if %length(&eval) > 0 %then %do; &eval %end;) out = final_3 nodupkey;
  by usubjid %if %length(&eval) > 0 %then %do; &eval; %end;;
run;

data &outds;
  set final_2 - final_3;
  length AVALC $50.;
  if min = 1 then AVALC = "CR";
  if min = 2 then AVALC = "PR";
  if min = 3 then AVALC = "SD";
  if min = 4 then AVALC = "PD";
  if min = 5 then AVALC = "NE";
run;
%end;

```

```

%if &N ^= 0 %then %do;
proc sort data=sd_3_1;
  by usubjid %if %length(&eval) > 0 %then %do; &eval;%end; visitnum;
run;

data sd_3_1_1;
  set sd_3_1;
  by usubjid %if %length(&eval) > 0 %then %do; &eval;%end; visitnum;
  %if %length(&eval) > 0 %then %do;
  if first.RSEVALID then order = 1;
  else order+1;
  %end;
  %if %length(&eval) <= 0 %then %do;
  if first.usubjid then order = 1;
  else order+1;
  %end;
run;

data sd_3_1_2;
  merge sd_3_1_1 sd_3_1_1(keep = rsltn firstobs = 2 rename = (rsltn=lead_rsltn));
run;

proc sql;
  create table sd_3_1_3 as
  select *, max(order) as bigorder
  from sd_3_1_2 group by usubjid,rseval order by usubjid %if %length(&eval) > 0 %then %do; ,&eval;%end;; order;
quit;

data sd_3_1_4;
  set sd_3_1_3;
  if diff = maxdiff and bigorder = 1 then min = 5;
  else if diff = maxdiff and bigorder ^= 1 then min = lead_rsltn;
  if not missing(min);
run;

proc sort data = sd_3_1_4(keep = usubjid min %if %length(&eval) > 0 %then %do; &eval %end;) out = final_1 nodupkey;
  by usubjid %if %length(&eval) > 0 %then %do; &eval;%end;;
run;
proc sort data = sd_3_2(keep = usubjid min %if %length(&eval) > 0 %then %do; &eval %end;) out = final_2 nodupkey;
  by usubjid %if %length(&eval) > 0 %then %do; &eval;%end;;
run;
proc sort data = nonsd(keep = usubjid min %if %length(&eval) > 0 %then %do; &eval %end;) out = final_3 nodupkey;
  by usubjid %if %length(&eval) > 0 %then %do; &eval;%end;;
run;

data &outds;
  set final_1 - final_3;
  length AVALC $50.;
  if min = 1 then AVALC = "CR";
  if min = 2 then AVALC = "PR";
  if min = 3 then AVALC = "SD";
  if min = 4 then AVALC = "PD";
  if min = 5 then AVALC = "NE";
run;

%end;

proc sort data = &outds;
  by usubjid %if %length(&eval) > 0 %then %do; &eval;%end;;
run;
%end;

```