PharmaSUG 2017 - Paper PO06

Statistician's secret weapon: 20 ways of detecting raw data issues

Lixiang Larry Liu, Eli Lilly and Company, Indianapolis, IN

ABSTRACT

Unclean clinical raw data is always statistician and statistical programmer's nightmare for all the downstream SDTM, ADaM and TFLs development work. Raw data issues could mess up the programming logic, create OpenCDISC reject, error, warning messages, and worst of all, if incorrect data is analyzed, study team could draw wrong or inaccurate conclusions regarding drug's safety and efficacy, which could put patients' safety in jeopardy and have significant impact on company's financial status.

This paper will review 20 effective ways of detecting raw data issues. Since they are applied to the drug dispense, labs, and safety related CRF data (Including adverse event, medical history, concomitant therapy, drug exposure etc.), which are common for all clinical trials, these methods and their associated SAS programs could be easily used for clinical trial studies across different therapeutic areas.

INTRODUCTION

In the past decade, significant progress has been made in data management's edit checks for cleaning the raw data. However, these edit checks have their limitations for only checking certain items within each CRF data, and there is no edit checks by comparing data across different CRF forms, which is an important part of detecting raw data issues. In addition, data management could not finish cleaning the data until database lock, but all the statistical group related deliverables, including SDTM, ADaM and TFLs work, need to have their programs developed and validated before database lock. The unclean raw data is always statistician and statistical programmer's nightmare, because the raw data issues could mess up all these programs' logics, create OpenCDISC reject, error, warning messages, and worst of all, if the data could not be 100% cleaned at the database lock, and if incorrect data is analyzed, study team could draw wrong or inaccurate conclusions regarding drug's safety and efficacy, which could put patients' safety in jeopardy and have significant impact on company's financial status.

Statistical group should take the initiative and start to check the raw data issues at each test data transfer before the database lock. This can place them ahead of the game, and understand what caused the troubles in all the downstream work and the OpenCDISC reject, error, warning messages, this can also enable them to catch the raw data issues that data management group might not be able to detect, and consequently, increase the quality of the data.

In the following section, this paper will review 20 effective ways of detecting raw data issues. Since they are applied to the drug dispense, labs, and safety related CRF data (Including adverse event, medical history, concomitant medication, drug exposure etc.), which are common for all clinical trials, these methods and their associated SAS programs could be easily used for clinical trial studies across different therapeutic areas.

ISSUES & SOLUTIONS

1. Not coded adverse event (AE) terms

For the AE CRF form, patients enter the verbatim terms first, which is the AETERM variable in both the raw data and in SDTM, then the coder will code them based on the most current Medical Dictionary for Regulatory Activities (MedDRA) at that time, and pull out the MedDRA Lowest Level Term Code (LLTCD). Since the MedDRA is updated twice a year in March and September, if the trial lasts more than 6 months, there might be different versions (which is called the coding version) of MedDRA in the data. If the AE term is already coded, there will be the LLTCD available for SDTM programmers to use for merging with the most current version of the MedDRA (which is called the reporting version) and to pull out the AEDECOD. If the AE term is not coded, then the LLTCD will not be available to merge with the most

current MedDRA, and the AEDECOD will be missing in SDTM, which will cause an error/reject message in SDTM OpenCDISC report. The following is an example of the SDTM OpenCDISC report from the Pinnacle 21 Enterprise, there are 4 not coded AE terms in the raw data, which causes 4 records of null value in AEDECOD in SDTM, and this is flagged by the OpenCDISC check as shown below:

Pinnacle 21 Enterprise Validation Report

Issue Summary

Dataset	Rule ID	Publisher ID	Message	FDA	PMDA	Found
AE						
	SD0002	FDAC018	NULL value in AEDECOD variable marked as Required	Error	Reject	4

By checking how many AETERM are not coded, statistician and statistical programmer can have a good understanding about the coding progress, and communicate this with the coders and data management group in real time to ensure all the AETERM are coded at the database lock. These un-coded AE terms could be easily identified by the following SAS code, please note the raw dataset name might vary across different studies:

```
data un_coded_AE;
  set raw.ae3001a; /* all the raw data are in the 'raw' library */
  where AEDECOD = '';
run;
```

2. AE terms with missing severity

AE severity is generally classified as either 'Mild', 'Moderate' or 'Severe', it is needed in the downstream TFLs, and it is needed to derive the treatment emergent adverse event (TEAE) flag in ADAE ADaM dataset. Missing AE severity will also cause an error message in the SDTM OpenCDISC report. The following code could identify these records:

```
data AE_missing_SEV;
  set raw.ae3001b;
  where AESEV = '';
run;
```

3. Duplicated AE group ID

AE group ID (AEGRPID) is expected to be a unique value for each adverse event per subject, this variable also severs as the linkage between different SDTM domains depending on the study design, e.g. it is the linkage between AE and FA domains. Duplicated AE group IDs within each subject's records is a raw data issue and will cause an error message in SDTM OpenCDISC report, the following SAS code will be able to output the records with this issue.

```
proc sort data=raw.ae3001a out=ae3001a dupout=dup_aegrpid nodupkey;
  by subjid aegrpid aesev;
run;
```

4. The exact time information is missing for AEs that occurred on the same day as the first dosing date

The exact time information for an AE is not required except the ones occurred on the same day as the first dosing of the study drug. If the AE occurred before the first dosing, then the AE would be a pre-existing condition and it was not caused by the drug or study procedure; if the AE occurred after the first

dosing, then the AE would be a TEAE and could be caused by the study drug or study procedure. If the AE and the first dosing occurred on the same day, the only way to tell whether it is a pre-existing condition or TEAE is to compare the dosing time and the AE starting time. This information is important to downstream SDTM, ADaM and TFLs work, since there are several TEAE related tables for pretty much all the clinical trials. The following SAS code will be able to detect these records.

```
proc sql;
 /* read in AE raw dataset ae3001b, and create the AE start date
 variable aestdt */

    create table ae as
    select * , input(catx('-',AESTDATYY,AESTDATMO,AESTDATDD), yymmdd10.) as
              aestdt format=yymmdd10.
    from raw.ae3001b(where=(aegrpid>.))  order by subjid, aegrpid, aespid;

 /* read in exposure raw dataset ec1001, and create the exposure start  date
 variable dostdt */

    create table ex as
    select distinct subjid, input(catx('-',ECSTDATYY,ECSTDATMO,ECSTDATDD),
              yymmdd10.) as dosdt format=yymmdd10.
    from raw.ec1001 (where=(ectpt='W0'));

 /* merge the AE and exposure datasets created above by subject ID and
 AE/exposure start date, then subset the data by keeping only the records
 with missing hour or minute information */

    create table ae_ex as
    select a1.* , a2.dosdt
    from ae a1 inner join ex a2
    on a1.subjid=a2.subjid and a1.aestdt=a2.dosdt having AESTTIMHR in('-
     99', '') or AESTTIMMI in('-99', '') order by subjid, aegrpid, aespid;
quit;
```

5. Concomitant medication (CM) not coded

Similar to the issue discussed in #1, an un-coded CM will create trouble in the downstream SDTM, ADaM, TFLs programming and cause an SDTM OpenCDISC error, the following code will detect these records.

```
data un_coded_CM;
  set raw.CM1001; /* all the raw data are in the 'raw' library */
  where CMDECOD = '';
run;
```

6. Treatment exposure date and time not in sequential order

When a patient fills in the exposure form with all the date and time information, mistake could happen by filling in the wrong day, month or year. The following table presents two examples: for patient 1001, the exposure date at week 36 was on April 29th 2016, which is earlier than the drug taken date at week 34. By looking at each exposure records, we can tell that the drug was expected to be taken every other week, therefore, the correct exposure date at week 36 should be 29MAY2016, the patient must have entered the wrong month. Similar case for patient 1002, the exposure time at week 8 is earlier than the exposure time at week 6, we can tell that the date at week 6 should be 17NOV2015 instead of 17DEC2015. After detecting these issues, statistical function group should communicate with the data management group and have the site correct these records.

Subject	Exposure Timepoint	Exposure Date/time
1001	W28	01APR2016:05:35:00
	W30	15APR2016:12:00:00
	W32	29APR2016:03:00:00
	W34	13MAY2016:12:15:00
	W36	29APR2016:15:00:00
	W38	10JUN2016:12:00:00
1002	W0	05OCT2015:10:17:00
	W0	05OCT2015:10:18:00
	W2	19OCT2015:10:30:00
	W4	02NOV2015:12:56:00
	W6	17DEC2015:12:30:00
	W8	01DEC2015:04:36:00
	W10	15DEC2015:07:41:00
	W12	04JAN2016:18:50:00

The following SAS code will pull out these troublesome records:

```
/* read in the exposure raw data set, and create the exposure date
   variable ADT, and create the treatment visit ID variable, The ectpt
   variable has values of W0, W4, W8, etc., 'W' stands for 'WEEK' */
 data ex1001;
    set raw.ec1001;
    adt=input(catx('-',ECSTDATYY, ECSTDATMO, ECSTDATDD), yymmdd10.) ;
     trtvisid=input(compress(ectpt,'W'), best.);
 run;

   /* sort the data by subject ID and the derived treatment visit ID so
that the data is in exposure time sequence order */
 proc sort data=ex1001;
   by subjid trtvisid;
 run;

   /* Pull out the records where the later exposure records having an
earlier date by using the lag function*/
 data ex_issue;
    set ex1001;
    by subjid;
    if subjid=lag(subjid) and . < adt < lag(adt);
 run;
```

7. Duplicated drug dispense records

Drug dispense data plays an important role on the drug compliance rate calculation, but sometimes the data might be not as clean as we would like it to be. The dispense center could enter the data by mistake. One of the quality controls is to check the duplicated dispense records. For example, if we only expect one drug dispense per study visit, but we see two drug dispenses at certain visit for some subjects, then it would be an indicator of data issue. These records could be easily identified by "proc sort nodupkey" or by the "proc freq" and pull out the counts per study visit.

8. Dispense time is later than the dosing time

We would assume it should be always the case that the patient should get the study drug first, i.e. drug dispense first, then took the drug after he/she got the drug, otherwise, where the drug could come from? In real world data, it is not always the case, drug dispense data could be messy. Patient could call the drug dispense center and complain that the drug expired, or the syringe is leaking, and asks for additional dispense. Or the patient could be on vacation for a long time, and asks for additional drug dispenses. In all these scenarios, if there is no clear instruction for drug dispense center to record the visit ID, it could produce some messy data, and could end up some records with dispense time is later than the dosing time at some visits. These records could be easily identified by comparing the dispense date with the dosing date for each visit per subject, and communicate with data management and find out the root cause of each issue.

9. Randomized patients with no drug dispense or no drug exposure

Once patient got randomized, the patient will be in the intent to treat (ITT) population, however, for one reason or another, after the randomization, patient might discontinue the study without the drug dispense, or with the drug dispense, but without any drug exposure. The study team should have a good understanding of who these patients are, and what caused the no drug dispense/exposure. These patients could be identified by comparing the randomization dataset and the drug dispense or drug exposure dataset.

10. First dose was not taken at the designated visit or date

The first dose date and time information is important because it is the cutoff point between the baseline and post baseline for all the lab, vital, questionnaires measurements. For AEs, prior to the dosing time, it is pre-existing condition, after that, it is TEAE. If the first dosing time was not at the designated visit for some patients, then it could impact the ADaM specs and programming. All this kind of records should be closely examined and make sure there is no data entry error. These records could be identified with the study knowledge of the baseline visit number and the first. SAS function.

11. Missing disposition date(s)

Some study may only have study disposition date, but if the study drug has long half-life, the trial could also have the treatment disposition, then patients still come in for clinical visits for a certain period of time to monitor the drug safety. For this kind of trials, they have both treatment disposition and study disposition. These disposition dates are important because it involves the treatment duration or study duration calculation in the downstream ADaM and TFLs work, they are required information on the CRF form and could not be missing at the final database lock. These records could be easily identified if either the day, month or year information is missing on the disposition CRF raw dataset.

12. Duplicated disposition records or incorrect DSDECOD/DSTERM was selected

If the subject completed the study, then both the treatment disposition and study disposition's DSDECOD should be "COMPLETED". For screen failures, there should be only one study disposition record with the DSDECOD as "SCREEN FAILURE", and DSTERM lists the reason for the screen failure. For early discontinued patients, the situation is tricky because the clinical sites need to choose the right DSDECOD and DSTERM for each patient, and they quite often got confused and selected multiple reasons for the same patient's disposition, which could create duplicated disposition records. Clinical site personals could also select inconsistent information between DSDECOD and DSTERM for these patients. For all early discontinued patients' disposition records, the common practice is that the statistics team, data management team and medical team get together and go through each patient one by one, and make sure each disposition's DSDECOD and DSTERM are consistent and make sense.

13. Duplicated lab test records

For most clinical trials, at least a few dozens of lab parameters' data are generated, sometimes all the lab data could come from a central lab, sometimes they split between the central lab and local labs. In many cases, if the clinical site has concerns on some lab measurement, a re-test might be taken, or the patient came in for an unscheduled visit; in some other cases, the sample in one tube might be damaged or spilled in the middle of the tests, and the lab technician had to pull out another sample tube, which might create two different lab records at the same visit. Overall, the lab dataset is large and messy, we have to approach it with caution and look for data issue flags. One issue that is easy to detect is the duplicated lab test records at the same visit for the same lab parameter for the same subject. Since we only need one record for either the baseline or the post baseline, these records need to be checked closely and communicate with data management accordingly.

14. Lab collection date is after the study disposition date

Once patients had the study disposition, either completed the study or early discontinued the study, there should be no data collected any more, therefore, it is generally a raw data issue if we have some lab records' collection date after the study disposition date, this could create downstream ADaM and TFLs programming issues, and needs to be communicate with data management to have these data checked.

15. Medical history (MH) term not coded

Similar to the issue discussed in #1, an un-coded MH term will create trouble in the downstream SDTM, ADaM, TFLs programming and cause an SDTM OpenCDISC error, the following code will be able to detect these records.

```
data un_coded_MH;
  set raw.MH7001;
  where MHDECOD = '';
run;
```

16. Visit number and visit date are not in sequential order

When patients enter the study visit dates into the CRF, they could enter the wrong visit date by pressing the wrong key on the computer, either the day, month or year could be wrong. The following table has shown two examples: the visit 5 date is earlier than visit 4 date, the patient might have entered the wrong month there; visit 8 date is earlier than visit 7 date, the patient might accidently entered the wrong year there, all these records should be brought to the attention of the data management and ask the clinical site to fix the data issues.

Subject	Visit	Visit Date
1001	1	2012-09-08
	2	2012-09-17
	3	2012-10-19
	4	2012-11-16
	5	2012-10-11
	6	2013-01-04
	7	2013-02-02
	8	2012-03-04
	9	2013-03-28

The following SAS code could be used to detect this kind of records:

```
/* read in the raw study visit dataset, derive the visit date and visit ID
    variables */
data sv1001;
   set raw.sv1001;
   visdt=input(catx('-',VISDATYY,VISDATMO,VISDATDD), yymmdd10.);
   visid=input(compress(blockid,,'kd'), best.);
   keep subjid visid blockid visdt;
run;

  /* sort the data by subject ID, visit ID and visit date */
proc sort data=sv1001;
   by subjid visid visdt;
run;

  /* using the lag function to pull out the records with the issue */
data visit_issue;
  set sv1001;
  if subjid=lag(subjid) and visdt < lag(visdt);
run;
```

17. Age is missing

Age is an important variable for all clinical trials because it is used in the demographics table, and in the analyses by age subgroups. In many cases, age might be a covariate in the statistical model. Most trials will collect either the birthday information, or just the birth year, then derive the age for each patient based on either the inform consent date, the randomization date or the first dosing date. If age is missing in the DM domain, then either the birthday/birth year was not collected on CRF, or the informed consent/randomization/first dosing date was not collected on CRF, in either case, data management needs to contact the clinical site and have the needed information collected.

18. Race is missing or all races are selected

Similar to the age variable, race is needed for the demographics table, in the analyses by race subgroups, and it could be a covariate in the statistical model. In some situations, patient really did not know what the right race should be selected, for example, the two parents might have different races, and patient just left the race as missing, or selected many different races. We also see some patients just selected all races in some clinical trials, which created another issues. Statistical team and data management team should work with the clinical sites to have all the needed race information collected.

19. Protocol deviation start date is after the study disposition date

Protocol deviation monitor report is an important raw data source for the downstream DV domain and related ADaM and TFLs work. It is not uncommon to see the incorrect day, month and year were entered. One way to check the data is to make sure the protocol deviation start date should be no later than the study disposition date, because the study team will not monitor the patient any more after study disposition. In the listed table below, there is one record in the data that having protocol deviation start date after the study disposition date, which created a warning message in the SDTM OpenCDISC report.

Dataset	Rule ID	Publisher ID	Message	FDA	PMDA	Found
DV	SD1202		DVSTDTC date is after RFPENDTC	NA	Warning	1

20. Total treatment duration is much longer than the study designated time window

If patient's total treatment duration is much longer than expected per the study design, it is an indicator of something might be wrong with the data. For example, if the trial lasts for 52 weeks, but some patient's treatment duration is 60 weeks long, then the study team might need to look at each study visit and see where the extra eight weeks come from. If the patient got the same amount of drug dispenses as other patients, but had eight more weeks somewhere in the course of the trial, it will have some impact on both the efficacy and safety analyses. In many of these cases, they were caused by some raw data issues and could be fixed, otherwise, the study team could mark these cases as protocol violation, and exclude them from the per protocol population analyses. If there is any exposure duration analysis, there should be variable of treatment duration in one of the ADaM datasets, and this variable could be used to pull out these records. The cutoff value of the number of extra weeks/days varies based on the study design, and it should be a decision from the study team.

CONCLUSION

This paper reviewed 20 different ways of finding raw data issues, which could be applied to vast majority of clinical trials. These methods will empower statisticians and statistical programmers to detect the raw data issues early, which will lead to high quality of data at database lock, and consequently, the study team can draw reliable conclusions on both the safety and efficacy data, which will benefit the patients.

CONTACT INFORMATION

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

Lixiang Larry Liu
liu_lixiang@lilly.com