PharmaSUG 2018 - Paper AD18

A time saving approach to track data issues

Aishhwaryapriya Elamathivadivambigai, Seattle Genetics, Inc., Bothell, WA

ABSTRACT

Data issues in raw datasets are a menace in statistical programming. The earlier they are identified and fixed, the smoother the programming process. Edit checks performed by the Data Management team often do not evaluate the data for compliance to CDISC standards. For instance, cases like missing values in a required variable or incorrect visit values according to the protocol could be missed by the DM's screening process. Many such issues are detected during the TFL generation process, thus affecting the quality of the outputs, and not to mention the time consumed to program around them. Such data errors that elude DM's initial edit checks need to be detected at an earlier stage to produce accurate and precise outputs. This paper proposes a way to detect such data issues upstream and list them in an Excel document for investigation by the programmer, thus saving a significant amount of time for programmers and allowing production of higher quality outputs.

INTRODUCTION

Raw datasets are a collection of data, from electronic case report forms (eCRF). Data Management puts the data entered in the CRF into appropriate raw datasets that are in turn used to generate SDTM, ADAM and eventually Tables, Listings and Figures (TLFs).

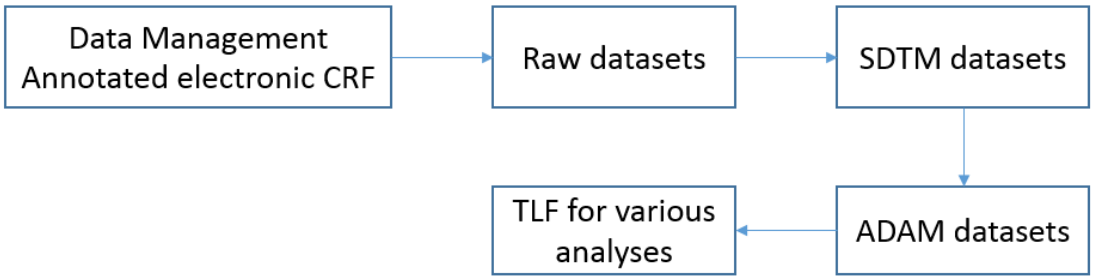

Figure 1: Flow of clinical trial process from eCRF to TLF (Table, Listing, and Figure) generation

Figure 1 shows the workflow of the clinical trial process from eCRF to TLF generation. Data Management performs edit checks to make sure the captured data is complete and accurate. These edit checks verify the 'Entry required' field of the CRF to make sure all the variables with Entry required as 'Y' are not missing and evaluates for obvious logical flaws/errors in the data like onset date < date of birth, seriousness of AE is 'No' but AE toxicity grade 4 or 5 is selected etc. These are just a few examples of the checks Data Management performs.

Table: AE ****Entry Restrict CRADE user role**

Field	Data Type	Codelist	Entry Required	SDV Required	Design Notes
LABEL					See label above
AESEQ	Num, 8		N	N	Autopopulated starting with '1'
AETERM	Char, 200		Y	Y	Dictionary=MedDRA
START	Num, 8	start	Y	Y	

Figure 2: CRF page showing different attributes of AE variables

Figure 2 shows the CRF page that lists all the attributes of AE variables along with Entry required field.

AE					
	SD0002	FDAC018	NULL value in AEDECOD variable marked as Required	Error	10
	SD0009	FDAC206	No qualifiers set to 'Y', when AE is Serious	Error	486
	SD0080	FDAC208	AE start date is after the latest Disposition date	Error	413
	SD0090	FDAC209	AESDTH is not 'Y', when AEOUT='FATAL'	Error	15
	SD1082	FDAC036	Variable length is too long for actual data	Error	18

Figure 3: Pinnacle 21's CDISC validator report for AE SDTM dataset

Figure 3 shows an example Pinnacle 21 report for AE dataset. Pinnacle 21's CDISC validator helps us to check if the SDTM and ADAM datasets follow the CDISC guidelines. The errors listed above are not always miscoding in SDTM datasets, they can be because of any data issues in the raw datasets. CDISC validator report will help us to identify some of the data issues in the raw datasets after they have been mapped into SDTM and ADAM datasets.

This paper proposes a way to detect the data issues that elude Data Management edit checks before the creation of SDTM datasets. Different checks have been included in this paper based on the CDISC Validator report and data issues observed in the raw datasets

METHODOLOGY

This section describes in detail the method to check for data issues in AE and DS SDTM datasets. AE and DS datasets are created by combining several raw datasets. Each raw dataset is scanned to figure out potential issues that would create problems with SDTM/ADAM datasets and eventually the outputs.

CHECKING DATA ISSUES IN RAW DATASETS RELATED TO AE

1. Required variables in AE are AETERM and AEDECOD, where AETERM is the topic variable. Null values in either of the two variables would results in an error in the CDISC validator. Null values can be due to a missing value in raw dataset itself.

```
if aeterm = ' '  then notes=strip(subject)||' AETERM(required variable) is missing in R.AE';

if aeterm_pt = ' ' then notes=strip(subject)||' AETERM_PT(AEDECOD-required variable) is missing
in R.AE';
```

   The above code segment shows an instance where AETERM, a variable in the raw dataset AE has a null value. AETERM and AETERM_PT are used to create AETERM and AEDECOD in SDTM dataset respectively. R in R.AE represents the raw datasets library.

2. Onset period of the adverse event (START) is a critical parameter in all AE related analyses. This variable can be verified by comparing AE start date and first dose date from R.AE and R.EX respectively.

```
if start='Started before the signing of consent' then do;
        if aestdtc ne . then do;
                if aestdtc > datepart(iedtc) then notes=strip(subject)||' Data discrepancy in
                AE.START or CONSENT.IEDTC';
         end;
end;
if start='Started after consent but before first dose of any study treatment' then do;
        if aestdtc ne . then do;
                if aestdtc < datepart(iedtc) and aestdtc > exstdtc then notes=strip(subject)||' Data
                discrepancy in AE.START or CONSENT.IEDTC or first dose date of any study
                drug';
```

```
            end;
        end;
        if start='Started after first dose of any study treatment' then do;
            if aestdtc ne . and aestdtc < exstdtc then notes=strip(subject)||' Data discrepancy in AE
            start date or EX first dose date';
        end;
```

Consent date (IEDTC) and first dose date (EXSTDTC) are captured in R.CONSENT and R.EX raw datasets respectively. Similarly we can verify the onset time variable in R.AE by comparing the values with time of first dose in R.EX.

3. Severity (AESEV) and Outcome of an Event (AEOUT) are two related variables which should follow the CTCAE guidelines.

```
if ( aesev='Grade 5' and aeout ^= 'Fatal' ) or ( aesev ^= 'Grade 5' and aeout = 'Fatal' ) then
notes=strip(subject)||' AESEV and AEOUT not matching';
```

Severity of Grade 5 refers to death/Fatal outcome according to CTCAE (Common Terminology Criteria for Adverse Events).

4. Treatment emergent flag is a conditional variable in ADAM Basic data structure. It is defined as an adverse event that occurred after the exposure to study drug or an adverse event whose severity has increased after the exposure to study drug. The AE date variables play an important role in this derivation, they can be checked at the raw dataset level to avoid issues with treatment emergent flag or AE outputs.

```
if (aestdtc ne . and aeendtc ne .) and (aeendtc < aestdtc) and (aeendtc ne aestdtc) then
notes=strip(subject)||' AE end date before AE start date';
```

This issue can occur when data is entered incorrectly or when the date is incomplete.

5. The outcome of an adverse event (AEOUT) is related to the death date of the patient when the outcome is fatal. The death date needs to be captured when the outcome is fatal and related to the adverse event.

```
if aeout = 'Fatal' and deathdt= . and aespid ne ' ' then notes=strip(subject)||' AEOUT is fatal,
deathdt in r.eos is missing';

 if aeout = 'Fatal' and deathdt ne . and aespid = ' ' then notes=strip(subject)||' AEOUT is fatal,
deathdt is present and aespid in r.eos is missing';
```

Death date is captured in R.EOS (End of Study) dataset. AESPID identifies which AETERM resulted in a fatal outcome in this scenario. Reporting the number of deaths in a study is an important part of safety analysis. This code segment helps us to identify possible issues with capturing death information.

6. The issues that are captured in Steps 1-5 are listed in a dataset. Which are then investigated before SDTM/ADAM generation. Collecting all the possible issues will help us in addressing CDISC validator results.

	NOTES	RAW_DATASET
1	10001-0043 AETERM(required variable) is missing in R.AE	r.ae
2	10001-0043 AETERM_PT(AEDECOD-required variable) is missing in R.AE	r.ae
3	10001-0070 AE End date before AE start date	r.ae
4	10001-0070 Data discrepancy in AE.START or CONSENT.IEDTC	r.ae
5	10003-0127 AEOUT is fatal, deathdt in r.eos is missing	r.ae
6	10004-0027 AETERM(required variable) is missing in R.AE	r.ae
7	10004-0165 AETERM(required variable) is missing in R.AE	r.ae
8	10004-0167 AETERM(required variable) is missing in R.AE	r.ae
9	10005-0148 Data discrepancy in AE start date or EX first dose date	r.ae
10	10007-0152 Data discrepancy in AE start date or EX first dose date	r.ae
11	10015-0114 AETERM(required variable) is missing in R.AE	r.ae
12	10015-0137 AETERM(required variable) is missing in R.AE	r.ae

Figure 4 The dataset *aecheck* listing data issues related to Adverse Events

Figure 4 shows the dataset *aecheck,* which specifies the data issues along with the subject number and the raw dataset involved. According to this dataset, the required variable (AETERM) is missing for a few subjects that would cause the Pinnacle 21's validator to show errors. Resolving the issue before CDISC validator saves times in checking SDTM datasets, as the problem was identified before the creation of SDTM datasets.

CHECKING DATA ISSUES IN RAW DATASETS RELATED TO DS

SDTM Disposition (DS) dataset is a combination of different raw datasets that include consent, discontinuation, end of treatment, end of study and long term follow up. This segment includes the checks performed across all the raw datasets mentioned above. Like the dataset *aecheck*, the data issues from this segment are also stored in a separate dataset.

7. The consent raw dataset mainly contains information on informed consent signed date, patient enrollment and the reason(s) for not enrolled. The date of informed consent plays a very important role in deliverables and assigned as DSDTC for DSTERM 'Informed consent signed'.

```
if iedtc_yyyy=. and nmiss(iedtc_mm,iedtc_dd)=0 then notes=strip(subject)||' Year part is missing in iedtc in r.consent';

if iedtc_mm=. and nmiss(iedtc_yyyy,iedtc_dd)=0 then notes=strip(subject)||' Month part is missing in iedtc in r.consent';

if iedtc_dd=. and nmiss(iedtc_mm,iedtc_yyyy)=0 then notes=strip(subject)||' Date part is missing in iedtc in r.consent';

if nmiss(iedtc_yyyy,iedtc_mm,iedtc_dd)=3 then notes=strip(subject)||' Consent date in r.consent is missing';
```

   IEDTC in the above code represents the date the informed consent was signed. Incomplete date variables in raw datasets are usually data entry errors. The date can be completely missing or day/month/year can be missing. Any of these scenarios creates a problem in outputs and needs to be checked in advance.

8. R.CONSENT records the eligibility criteria and if the patients met those criteria. Inconsistencies in the data can cause a problem in the outputs related to disposition.

```
if strip(ieorres)='No' and strip(ietestcd)=' ' then notes=strip(subject)||' Patient did not meet eligibility criteria, but criteria not met is missing';
```

   IETESTCD in R.CONSENT captures the reason the patient did not meet eligibility criteria. This variable mainly affects the SDTM dataset IE if missing, and needs to be recorded in advance so data

management can be notified about the issue.

9. Once the eligibility criteria are met, the patient is enrolled into the study or if the patient is not enrolled, the reason for the latter is captured in the raw datasets.

```
if dsenroll='No' and dsrsn=' ' and aespid=' ' and dsrsnoth=' ' and dsrsninv=' ' then
notes=strip(subject)||' Patient did not enroll into the study but reason for not enrolling is missing';
```

DSRSN denotes the reason for patient not enrolled, AESPID is the Adverse Event that caused the patient not to enroll, DSRSNOTH captures any other reason for the patient to not enroll and DSRSNINV denotes investigator decision to not enroll the patient. One of the above variables is populated if the patient did not enroll; a missing value in all of them when patient did not enroll is a data issue.

10. Discontinuation from a treatment or a part of the treatment can be due to various factors. All the reasons are captured in the raw datasets. In the SDTM disposition dataset, the reason for discontinuation is commonly stored in the required variable DSTERM. A more detailed description for the reason for discontinuation is also collected in the CRF to link to other raw datasets. For example, if the reason for discontinuation is adverse event, the specific AESPID that reflects the adverse event that caused the subject to discontinue the study is also collected.

```
if dsterm='Adverse Event' and aespid=' ' then notes=strip(subject)||' Reason for discont. is AE but
the Adverse Event is not specified';

if dsterm='Investigator Decision' and invdecsp=' ' then notes=strip(subject)||' Reason for discont.
is Investigator decision but details not specified';

if dsterm='Patient Decision' and ptdecsp=' ' then notes=strip(subject)||' Reason for discont. is
Patient decision but details not specified';
```

The above code segment helps us to check if the reason is accurately specified. The data can be accurate only if both the variables for reason for discontinuation are collected.

11. End of treatment page captures all information related to the patient's last day of treatment and the reason for the end of treatment.

```
if dsyn='Yes' and dsdtc_raw=' ' then notes=strip(subject)||' EOT visit performed but date is
missing';
```

The date variable captured in this page is a critical point in the study as it marks the last day of study drug administration. A missing value in the date when the EOT visit was performed is a data issue and there is no way of knowing for sure if the EOT visit was performed, as this can be a data entry error.

12. The following code helps us to check if the date variable is fully populated when the EOT visit was performed. Incomplete date can be a problem while calculating the duration of treatment and eventually affect all the outputs related to it.

```
if dsyn='Yes' then do;

    if dsdtc_yyyy=. and nmiss(dsdtc_mm,dsdtc_dd)=0 then notes=strip(subject)||' Year part is
    missing in dsdtc in r.eot';

    if dsdtc_mm=. and nmiss(dsdtc_yyyy,dsdtc_dd)=0 then notes=strip(subject)||' Month part
    is missing in dsdtc in r.eot';

    if dsdtc_dd=. and nmiss(dsdtc_mm,dsdtc_yyyy)=0 then notes=strip(subject)||' Date part is
    missing in dsdtc in r.eot';

    if nmiss(dsdtc_yyyy,dsdtc_mm,dsdtc_dd)=3 then notes=strip(subject)||' EOT date in r.eot
    is missing';

end;
```

13. The long-term follow up CRF page contains follow up information after end of treatment. The following code checks the date when the patient was contacted for a follow up.

```
if dscontact='Yes' then do;
    if dsdtc_yyyy=. and nmiss(dsdtc_mm,dsdtc_dd)=0 then notes=strip(subject)||' Year part is missing in dsdtc in r.ltfu';
    if dsdtc_mm=. and nmiss(dsdtc_yyyy,dsdtc_dd)=0 then notes=strip(subject)||' Month part is missing in dsdtc in r.ltfu';
    if dsdtc_dd=. and nmiss(dsdtc_mm,dsdtc_yyyy)=0 then notes=strip(subject)||' Date part is missing in dsdtc in r.ltfu';
    if nmiss(dsdtc_yyyy,dsdtc_mm,dsdtc_dd)=3 then notes=strip(subject)||' LTFU date in r.ltfu is missing';
end;
```

14. End of study page captures all information related to the last contact date of a patient. It captures the death date and the reason for death if the patient was not alive during the last contact.

```
if alive='No' and (aespid =' ' and dscause=' ' ) then notes=strip(subject)||' Patient is not alive but cause of death is missing';
```

The above code segment checks for the cause of death when the alive variable is “No”. This is because of the inconsistency in data that will affect the outputs.

15. The following code segments checks for missing values or incomplete date variables.

```
if alive='No' then do;
    if deathdt_yyyy=. and nmiss(deathdt_mm,deathdt_dd)=0 then notes=strip(subject)||' Year part is missing in deathdt (r.eos)';
    if deathdt_mm=. and nmiss(deathdt_yyyy,deathdt_dd)=0 then notes=strip(subject)||' Month part is missing in deathdt (r.eos)';
    if deathdt_dd=. and nmiss(deathdt_mm,deathdt_yyyy)=0 then notes=strip(subject)||' Day part is missing in deathdt (r.eos)';
    if nmiss(deathdt_yyyy,deathdt_mm,deathdt_dd)=3 then notes=strip(subject)||' Death date in r.eos is missing when alive=No';
end;

if dsdtc_yyyy=. and nmiss(dsdtc_mm,dsdtc_dd)=0 then notes=strip(subject)||' Year part is missing in EOS date (r.eos)';

if dsdtc_mm=. and nmiss(dsdtc_yyyy,dsdtc_dd)=0 then notes=strip(subject)||' Month part is missing in EOS date (r.eos)';

if dsdtc_dd=. and nmiss(dsdtc_mm,dsdtc_yyyy)=0 then notes=strip(subject)||' Day part is missing in EOS date (r.eos)';

if nmiss(dsdtc_yyyy,dsdtc_mm,dsdtc_dd)=3 then notes=strip(subject)||' EOS date is missing';
```

16. The date of end of study is recorded when the patient is alive along with the reason for end of study. The following code checks for missing reason for end of study when end of study date is populated.

```
if dsdtc_raw ne ' ' and dsterm=' ' then notes=strip(subject)||' Reason for End of study is missing when EOS date is present';
```

.

17. The issues are documented in the dataset *dscheck*.

	NOTES	RAW_DATASET
1	10003-9202 Consent date in r.consent is missing	r.consent
2	10006-9028 Patient did not meet eligibility criteria, but criteria not met is missing	r.consent
3	10006-9115 Consent date in r.consent is missing	r.consent
4	10001-0073 Reason for discont. is AE but the adverse event is not specified	r.disc
5	10001-0039 EOT visit performed but date is missing	r.eot
6	10001-0142 LTFU date in r.ltfu is missing	r.ltfu
7	10002-0078 Patient is not alive but cause of death is missing	r.eos
8	10002-0086 Patient is not alive but cause of death is missing	r.eos
9	10004-0055 Death date in r.eos is missing when alive=No	r.eos
10	10010-0052 Reason for End of study is missing when EOS date is present	r.eos

Figure 5 The dataset *dscheck*

Figure 5 shows the dataset *dscheck* that lists all the data issues in raw datasets involved in the creation of DS SDTM dataset.

18. The above steps can be repeated to check for potential data issues in all raw datasets. All findings are exported to an excel sheet which is updated often to keep track of all the data issues.

	A	B
1	RAW_DATASET	DATA_ISSUES
2	r.ae	10001-0043 AETERM(required variable) is missing in R.AE
3	r.ae	10001-0043 AETERM_PT(AEDECOD-required variable) is missing in R.AE
4	r.ae	10001-0070 AE End date before AE start date
5	r.ae	10001-0070 Data discrepancy in AE.START or CONSENT.IEDTC
6	r.ae	10003-0127 AEOUT is fatal, deathdt in r.eos is missing
7	r.ae	10004-0027 AETERM(required variable) is missing in R.AE
8	r.ae	10004-0165 AETERM(required variable) is missing in R.AE
9	r.ae	10004-0167 AETERM(required variable) is missing in R.AE
10	r.ae	10005-0148 Data discrepancy in AE start date or EX first dose date
11	r.ae	10007-0152 Data discrepancy in AE start date or EX first dose date
12	r.ae	10015-0114 AETERM(required variable) is missing in R.AE
13	r.ae	10015-0137 AETERM(required variable) is missing in R.AE
14	r.consent	10003-9202 Consent date in r.consent is missing
15	r.consent	10006-9028 Patient did not meet eligibility criteria, but criteria not met is missing
16	r.consent	10006-9115 Consent date in r.consent is missing
17	r.disc	10001-0073 Reason for discont. is AE but the adverse event is not specified
18	r.eot	10001-0039 EOT visit performed but date is missing
19	r.ltfu	10001-0142 LTFU date in r.ltfu is missing
20	r.eos	10002-0078 Patient is not alive but cause of death is missing

Figure 6 Excel sheet listing data issues

Figure 6 shows the excel sheet after combining datasets containing all data issues.

CONCLUSION

Data issues are a hindrance in producing quality outputs and as new data comes in the risk of data issues increases. It is essential that we adopt a technique to check for all data issues before mapping the raw datasets to SDTM datasets. With the proposed technique, all data issues are identified as new data comes in, thus averting subsequent delays in checking for data issues manually.

ACKNOWLEDGMENTS

I take this opportunity to thank my manager Jay Gadhiya for his timely suggestions and advices.

CONTACT INFORMATION

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

Name	: Aishhwaryapriya Elamathivadivambigai
Enterprise	: Seattle Genetics, Inc.
Address	: 21823 - 30th Drive S.E. Bothell, WA 98021
Work Phone	: 425-527-2668
E-mail	: aelamathivadivambiga@seagen.com