PharmaSUG2011 - Paper DM03

Data Edit-checks Integration using ODS Tagset

Niraj J. Pandya, Element Technologies Inc., NJ
Vinodh Paida, Impressive Systems Inc., TX

ABSTRACT

In the Clinical trials data analysis and reporting work, though the Data Management works hard to ensure high quality data as per the Good Clinical Practices, data issues still can exist. The study programmers extract data for the study and checks for potential data issues to ensure the data integrity by executing edit check programs. This paper demonstrates how to effectively integrate all different edit checks into one output file. Also, this article talks about some important and very common edit checks. This paper incorporates the use of ODS tagset feature that is available in SAS® 9.1.3 and later versions.

INTRODUCTION

Usually it happens that after getting the data snapshot from Data managers with clean data, study programmers find some issues in the data which are not in line with study design and not as per the CRF completion guidelines. It is also often observed that data values are valid values for the variable and from the first glance at a particular data domain, nothing seems odd. But when compared to other related variables in the data, or when multiple domain data is merged, some data records are found not to be in accordance. If such data issues are not cached and corrected, it can lead to erroneous study results which can also cause re-work for data managers as well as for statistical programmers.

Checking for some obvious data issues and reporting them back to data managers for query/correction before statistical analysis is started, can save lot of time and efforts later on for both data managers and programmers. In this article, we will discuss some of very basic still important edit-checks. Also a macro explained in this paper will show how all such edit-checks output can be combined in to one Excel® file using ODS ExcelXP tagset and every time a new edit-check is created/added, how to add a new tab in the resulting Excel file.

DATA EDIT-CHECKS

Let's look at some basic still important data edit-checks and what is the need to have these individual checks in place. Whole purpose to have these checks in place is to identify potential data issues early in the analysis of the data and allow sufficient time to correct them to avoid re-work and delays in reporting timelines.

1. **Partial dosing start and stop dates in study drug dosing data**

   This check identifies any records in the dosing log data where start and/or stop dates are not full. This can be identified by checking if character date variable is not missing but related numeric date variable has missing value. Partial dates can not be used in the derivation of treatment duration or while deriving treatment emergent AE flag for adverse event data. Thus it is important to query such partial dates and correct them. By knowing such cases, you can apply some rules to convert partial dates in to full dates if at all such dates can not be corrected by data managers.

2. **Future dosing dates in study drug dosing data**

   This check identifies any dosing records with either start or stop date as future date. This is an obvious issue that can occur in many other data types and many times is goes unnoticed. This can be easily identified by comparing numeric SAS date variable values with system date or today's date. Not correcting such date values in dosing data can lead to erroneous treatment duration report and can also affect the treatment emergent flag derivation for adverse events reporting which may result in over reporting in this case.

3. **Missing dosing information**

   This check gives a list of subjects that have one or more than one dosing records with missing dosing information like total dosage per day, dosing unit, dosing frequency etc. Simply by checking for missing values for such variables can give a list.

4. Dosing records with overlapping dates and dosing records with gaps.

This check identifies dosing records where there is an overlap in the dosing dates. This can be done by comparing the stop date of one dosing record with the start date of next dosing record. If the stop date of first record is after the start date of next dosing record, then it is a problem if daily dosage values in both these records are different. In this case total daily dosage for the overlapped days can not be determined easily and this should be corrected either by data managers or you have to apply some rule to get proper dosage amount. Also, it is good to have a list of subjects who have gaps of certain days between their two subsequent dosing records. This is not necessarily an issue but there should be some comments in dosing log data for such subject as for what reason the gap occurred if it was not planned as per study design.

5. Subject final summary data vs. study drug dosing data

Subject summary data indicates that subject completed the study or discontinued the study for any reason but subject's last dosing record has missing stop date. This check is important for data consistency and it is also important as this check indicates that dosing data is incomplete.

6. Subject final summary data vs. adverse event data

This check is also for data consistency and can be divided in two parts.

a. Subject summary page indicates that subject discontinued due to adverse event but adverse event data does not show discontinuation.

b. Adverse event data indicates subject discontinuation from study; however final summary data does not indicate final status as discontinuation due to adverse event.

Both of these checks show a list of subjects for whom final summary and adverse event data do not reconcile. If the issue is not corrected, numbers from study discontinuation report and discontinuation due to AEs reports will not match which can raise questions about the data integrity later on while writing clinical study report.

7. Adverse event start date after stop date of adverse event and future dates in adverse event data

This check identifies adverse event data where start date of adverse event is after the stop date of adverse event. This is obviously an issue and it should be queried and corrected. If not corrected, subject will have negative duration for this adverse event. This check also identifies any records with future dates and can be easily identified by comparing AE start and stop date variables with system date or today's date. If such future dates are not corrected, it can lead to wrong AE duration.

8. Adverse events with missing preferred term and body system organ class term

This check gives a list of reported investigator adverse event terms which remain uncoded using MedDRA dictionary. This list should be reported to data managers which can request change/update in MedDRA dictionary for such terms. If they are not mapped correctly, these adverse events records do not get reported on adverse event summary reports by preferred term which results in under reporting of adverse events.

9. Previous drug treatment stop date vs. ongoing flag

This check identifies subject with any previous drug treatment record where stop date is populated which is before the start of study drug and at the same time ongoing flag is also on. Subject either stopped taking this drug treatment or the treatment is ongoing and so in this case either the stop date is wrong or ongoing flag is wrong. This should be queried and corrected and based on correction, this drug treatment might be considered as condrug treatment.

10. Concomitant drug treatment record with start date after stop date

This check gives a list of records where start date of concomitant drug treatment is after the stop date. It seems purely a data entry issue where dates got switched. Such records should be queried and corrected. If this issue is not corrected, such records might go unreported in concomitant drug reports if the wrongly entered stop date (which is most probably the start date) happens to be before the start of study drug. This is because often stop date of drug treatment is compared with start date of study drug to consider the drug treatment as concomitant drug treatment.

11. **Multiple records with different result vs. collection date/time for a subject and for a specific lab parameter**

    This check produces a list of subjects that have multiple records with different results but same unit, collection date and time for a given lab parameter. This is an important issue as only one of these records will be used for lab summary reports and other possibly useful results will be neglected because of data issue. In this case either the result, collection date or collection time is wrong and should be queried. If after query, collection date or time turns out to be different then it could be used in summary report for other visits and time points.

12. **Lab data with missing unit but non-missing result**

    This check produces a list of records where raw lab result is available but raw unit is missing. Such records should be reported to data managers to query them and correct them out. If such records are not corrected, the result conversion in to standard unit for the lab parameter is not possible and such results are not used in lab summary report. In that case we are losing important study results for the summary reports.

13. **Lab data where character test result is available but numeric result is missing**

    This check produces a list of lab records where raw text value of result is available but text to numeric conversion fails and numeric value is missing. This is possible either because unit is entered along with the result in text field or text value of result has some characters like ‘<’ or ‘>’ signs in it. In any case, if this is not corrected, we will lose important test results from lab summary reports and hence this should be reported to data managers for correction.

We just went through some checks which are important for data consistency as well as have major impact on reporting too. You can think of many more other data issues and related checks and it is very important to have such checks in place so that important and useful study results don’t go unreported. Many other study specific checks can be added to this list which is out of scope for this article.

INTEGRATED EXCEL WORKBOOK

Now that we discussed some of data edit-checks, their need and significance, let’s look at how to integrate different outputs from each of such checks in to one place so that it is easy for data managers or for that matter for anyone who is reviewing and tracking them.

With the advancement in SAS over years, new capabilities are added. With the use of SAS version 9.1.3 and later, you can easily create a multi-sheet excel workbook using ODS TAGSET named ExcelXP. This tagset creates a new tab in the resulting excel file every time a new tabular output is created by some SAS procedures. Because of this characteristic, you can create multi-sheet workbook either by having series of procedures that create some tabular output or a single procedure that creates tabular output but with BY group processing. ExcelXP tagset supports numerous options that control both appearance and functionality of the Excel workbook.

By taking the advantage of such features of ExcelXP tagset and making use of macros in SAS, we can easily create an excel file that has first spread sheet with the list and explanation of different edit-checks and then subsequently one spread sheet per check.

Following example macro code explains how this can be achieved.

```
%macro edit_checks;

    Libname indata "Data location";

    ODS listing close;

 /* Create a dataset which will have the edit check description for each edit check */

    Data checks_list;
         Length Check_no 3 Sheet_name $15 Desc $200;
```

```
        Check_no = 1;
        Sheet_name = "Edit_Check_1";
        Desc = "Description for edit check # 1";
      Output;

        Check_no = 2;
        Sheet_name = "Edit_Check_2";
        Desc = "Description for edit check # 2";
      Output;

      So on keep adding for all other checks

   Run;

%global Sheet_name;

/** Macro to check if any observation exist in output dataset from each edit-check **/

   %macro rpt_obs;

        Proc sql;
           Create table temp as
           select *
           from &syslast;
        Quit;

        %if &sqlobs eq 0 %then %do;

            Data dummy;

            NOTE = 'Note: No records found meeting the edit check criteria';

            Run;

        %end;

   %mend rpt_obs;

/** Build edit-checks in blocks **/

/** Edit Check # 1 **/

   %macro Check_1;

     Enter the Code for performing the check here. Output a dataset containing the result

       %let Sheet_name = Edit_Check_1;
       %rpt_obs;

   %mend;

/** Edit Check # 2 **/

    %macro Check_2;

   Enter the Code for performing the check here. Output a dataset containing the result

       %let sheet_name = Edit_Check_2;
       %rpt_obs;

    %mend;

/** Edit Check # 3 **/

   %macro Check_3;
```

```
    Enter the Code for performing the check here. Output a dataset containing the result

        %let sheet_name = Edit_Check_3;
        %rpt_obs;

    %mend;

So on keep adding a macro code for each subsequent edit-check


 /** Process all of the above edit checks and save the results in an excel workbook **/

  ODS tagsets.ExcelXP path = "Output location" file = "edit_checks.xls" Style = statistical;

  ODS tagsets.ExcelXP options(sheet_name = "Edit_check description");

    Proc Print data = checks_list width = min noobs;

    Run;

    %Global newvar;
    %local i;

 /** Find maximum number of edit checks and put it in a macro variable **/

    Proc Sort data = checks_list out = checks nodupkey;
         By Check_no;
    Run;

    Data _null_;
         Set checks end = eof;
         If eof then call symput('newvar',compress(put(_n_,best.)));
    Run;

 /** Call the macro code for each edit check **/

    %do i = 1 %to %eval(&newvar.);

        %Check_&i.;

        ODS tagsets.ExcelXP options(sheet_name = "&sheet_name");

        Proc print;

        Run;

    %end;

 ODS tagsets.ExcelXP close;


%mend edit_checks;
```

Above mentioned code creates a dataset first with the list of all the edit checks along with a brief description. Then a brief macro code for each check is creating an output dataset with the result from the check. In the end, each of these datasets are being output using print procedure in conjunction with ExcelXP tagset so that each output will be printed in the form of a listing in a different spread sheet of resulting excel file. Name of each spread sheet is also controlled using sheet_name option of ExcelXP tagset.

With some modifications in the above mentioned macro, you can generate edit-checks output which are grouped by different domains or you can add functionality which gives user the flexibility to run some specific checks only and generate the output rather than running all the checks.

CONCLUSION

Quality of data is the core of any clinical trials study analysis and reporting. It is responsibility of not only data managers but every team member to keep their eyes open for catching any possible data issues that can lead to erroneous analysis. Hence it is in the interest of whole study team to have data edit-checks in place to catch possible data issues early in the phase of analysis and reporting rather than having them caught after analysis is done and reports are ready. It saves significant amount of re-work and time. Such checks ensure that major data issues will be caught and it gives the opportunity to correct them in time. It also gives the statistical programmer confidence in the data and allows more time on analysis and reporting and less time looking for data issues. SAS Macro code explained in this article helps integrating outputs from different edit-checks and keeping them all in one place which makes it easy to track such data issues when they are checked periodically.

CONTACT INFROMATION

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

Name: NIRAJ J. PANDYA
Phone: 201-936-5826
E-mail: npandya@elementtechnologies.com

Name: VINODH PAIDA
Phone: 860-333-4178
E-mail: vinodh.saspro@yahoo.com