PhUSE 2016

Paper CC10

Keeping Track of Database Changes During Database Lock

Sanjiv Ramalingam, Biogen Inc., Cambridge, USA

ABSTRACT

Higher frequency of data transfers combined with greater likelihood of changes to the database at the time of database lock necessitates the tracking of changes in the database. To ensure updates of only resolved data issues, datasets between the transfers must be compared. In this paper, a macro is presented that compares the datasets between the transfers and presents a report in the form of an excel spreadsheet. The first tab lists any datasets that are present in one transfer but not the other. Further processing is only done for datasets common to both the transfers. A second tab is created to present cases where there were differences in the number of records for that dataset between the transfers. Typically close to database lock, differences in number of records should be minimal if any at all. Subsequent tabs are created under the domain name if the values of any of the variables changed between each domain. Variables included in one transfer but not the other are covered in the same tab within the domain name. The macro is created taking advantage of %CALL EXECUTE and ODS tagsets and only the directory path of the transfer is required to run the macro.

INTRODUCTION

Greater number of data queries are raised and addressed at the time of database lock than at any other time of the clinical study. Most reporting at this stage of the clinical trial is for the clinical study report and hence critical to track database changes. To help programmers liaise with their data management counterparts and ensure that only agreed upon changes are reflected in the database it is helpful to have a tool that compares the current data transfer with the previous transfer. With heavily outsourced models it is highly recommended that such data comparisons be implemented to ensure quality. Both RAW and SDTM datasets should be compared as there have been experiences when no issues were reported with raw data but differences in SDTM data between transfers were found after un-blinding amongst SDTM variables that were unrelated to un-blinding. A macro was created specifically to aid this process. The macro creates an excel spreadsheet with different tabs. It is possible that a transfer may include or drop domains with respect to the previous transfer. To capture this scenario a tab called 'Missing Domain' is created as illustrated below.

Obs	src	Domain name
1	In current directory but not previous directory	RA
2	In current directory but not previous directory	SS1
3	In previous directory but not current directory	DNA
4	In previous directory but not current directory	DS3

Missing domain

Further processing is only done for datasets common to both the transfers. Another tab called 'N_mismatches' is created for domains where there were differences in the number of records between the transfers as illustrated below.

Obs	domain	old_N	new_N	difference
1	QS	683874	683657	-217
2	SUPPAE	12071	12080	9
3	SUPPCM	15460	15473	13
4	SUPPDS	1798	1796	-2
5	SUPPQS	169716	169708	-8
6	SV	6430	6428	-2

N_Mismatches AE CM DA DS DV FA SUPPDV

Subsequent tabs are created under the domain name only if the values of any of the variables changed between the domain transfers as illustrated below for the AE tab. A PROC COMPARE was used to obtain the comparison report. The OUTDIF option was used to write an observation that contains the differences for each pair of matching observations with the affected variables amongst the records highlighted using "XX" in the report. In the highlighted record below, the "XX" is marked below the date 2015-06-22 indicating that the date was changed from 2015-06-17 to 2015-06-22.

L	M	N	O	P	Q	R	S	T	U	V
AESEV	AESE	AEACN	AEREL	AEOUT	AECONTR	EPOCH	AESTDTC	AEENDTC	AESTDY	AEENDY
MILD	N	DOSE NOT CHANGED	NOT RELATED		N	BLINDED TREATMENT	2015-04-02	2015-04-02	10	10
MILD	N	DRUG INTERRUPTED	NOT RELATED		N	BLINDED TREATMENT	2015-04-02	2015-04-02	10	10
.....		.XXX.XX.XXXXXX..	.....	.....		.....	.....	.....	E	E
MODERATE	N	DRUG INTERRUPTED	NOT RELATED		Y	BLINDED TREATMENT	2014-11-02	2014-11-04	35	37
MODERATE	N	DRUG INTERRUPTED	NOT RELATED		N	BLINDED TREATMENT	2014-11-02	2014-11-04	35	37
.....		.....	.....	.....	X	.....	.....	.....	E	E
SEVERE	N	DOSE NOT CHANGED	NOT RELATED	NOT RECOVERED/NOT RESOLVED	N	FOLLOW-UP	2015-08-04		141	.
SEVERE	N	DRUG WITHDRAWN	NOT RELATED	NOT RECOVERED/NOT RESOLVED	N	FOLLOW-UP	2015-07-21		127	.
.....		.XXX.XX.XXX.XXXX	.....	.....		.....	.....X.XX	.....	-14	E
MODERATE	N	DRUG WITHDRAWN	RELATED		N	FOLLOW-UP	2015-10-09	2015-10-21	64	76
MODERATE	N	DRUG WITHDRAWN	RELATED		N	FOLLOW-UP	2015-10-09	2015-10-21	64	76
.....		.....	.....	.....		.....	.....	.....	E	E
MODERATE	N	DRUG INTERRUPTED	NOT RELATED		N	BLINDED TREATMENT	2015-06-12	2015-06-17	85	90
MODERATE	N	DRUG INTERRUPTED	NOT RELATED		N	BLINDED TREATMENT	2015-06-12	2015-06-22	85	95
.....		.....	.....	.....		.....	.....	XX	E	5
MODERATE	N	DOSE NOT CHANGED	RELATED		N		2015-04	2015-05	.	.
MODERATE	N	DOSE NOT CHANGED	RELATED		N		2015-04	2015-05	.	.

Current record
Previous record
Difference

MACRO PARAMETERS

Only four parameters are involved, three of which pertain to locations namely;

CRTDIR= < directory location of current data transfer>,

PREVDIR= <directory location of previous transfer>,

OUTDIR= <directory location where report is to be saved>

The fourth parameter is for variables that need to be dropped during PROC COMPARE. While this macro parameter is not required during SDTM comparisons it will be required when comparing raw data as raw data contains variables that are not collected in the eCRF nor mapped to SDTM format.

DROPVAR= <variables to be dropped for comparison>

METHODOLOGY

There are two main parts to the macro. The first part deals with the creation of the first tab of the report where differences in the number of records between domains amongst the transfers are calculated. It is in the first part that a flag is created per domain if the numbers of records between the domains are equal. This flag serves as a trigger for the data comparison to be executed for those domains in the second part. Comparison will only be done between domains that had an equal number of records between the transfers. Though the entire macro has not been presented in the paper all core components and code to build one has been shared.

PART ONE

A list of all datasets in both the current and previous transfer directory is first obtained using PROC CONTENTS. For every domain a macro call is made to obtain the difference in the number of records amongst the transfers. The macro call is automated using CALL EXECUTE. A key feature integral to automating essentially involves making SAS® write SAS code. This can be achieved by first creating a text string of the macro call and then invoking the macro call using CALL EXECUTE[1]. There are alternative ways of automating macro calls[2]. This section of the code is described below.

```
/* get list of all datasets in current directory*/
proc contents data=&crtdir._all_ out=lst1(keep=memname) noprint;
run;

proc sort data=lst1 nodupkey;
       by memname;
run;

/* get list of all datasets in previous directory*/
proc contents data=&prevdir._all_ out=lst2(keep=memname) noprint;
run;
```

```
proc sort data=lst2 nodupkey;
      by memname;
run;
/* determine datasets common to both transfers(lst) and only in either of the
transfers. Datasets(lst) common to only both transfers will be used for
determining mismatches in records or values */

data lst inlsta inlstb;
      merge lst1(in=a) lst2(in=b);
      by memname;
      if a and b then output lst;
      if a and not b then output inlsta;
      if b and not a then output inlstb;
run;
/* create dataset to highlight datasets in one transfer but not the other */
data misdom;
      length src $100;
      set inlsta(in=a) inlstb(in=b);
      if a then src='In current directory but not previous directory';
      if b then src='In previous directory but not current directory';
run;
/* create excel report for missing domains */
ods Tagsets.ExcelXP file="&outdir/report_&sysdate..xml"  ;
ods tagsets.ExcelXP options(Sheet_Name="Missing domain");
title 'Domains missing between Transfers';

proc print data=misdom label;
run;

/* macro for determining difference in records per domain between transfers */
%macro allcmp1(inds=,nds=);
/* sort individual domains using all variables available in the domain */
      proc sort data=&prevdir.&inds out=src1;
            by _all_;
      run;

      proc sort data=&crtdir.&inds out=src2;
            by _all_;
      run;
/* set both datasets together and id records from each domain */
      data src;
            set src1(in=a) src2(in=b);
            if a then src="src1";
            if b then src="src2";
      run;
/* determine counts for each datasets, transpose and then determine the difference
in number of records */
      proc freq data=src noprint;
            table src/out=cnt(keep=src count);
      run;

      proc transpose data=cnt out=nds;
            var count;
            id src;
      run;

      data nds;
            length domain $100;
            set nds;
            rename src1=old_N src2=new_N;
            domain="&inds";
            difference=src2-src1;
```

```
        if difference ne 0 then flag="Y";
        run;

        proc sql noprint;
               create table &nds as
               select domain,old_n,new_n,difference,flag
               from
               nds
               ;
        quit;

/* delete temporary datasets */
        proc datasets library=work noprint;
               delete cnt src1 src2;
        quit;

%mend;

/* create macro call by concatenating text required and assigning temporary datasets
*/
data x1;
        length allcmp1 $400;
        set lst end=eof;
        nds=cats('ds',_n_);
        allcmp1=cats('%allcmp1(inds=',memname,',nds=ds',_n_,')');
        call execute(allcmp1);
run;

/* combine all datasets and split them based on differences =0(cmp) or >0 (nocmp) */
proc sql noprint;
        select nds into:nds_ separated by ' '
        from x1;
quit;


/* datasets that had same number of records will be saved to nocmp */
/* PROC COMPARE will only be done for domains that had same number of observations */
data cmp nocmp;
        set &nds_;
        if flag eq "Y" then output nocmp;
        else output cmp;
run;
```

PART TWO

The second part involves the comparison (PROC COMPARE) between datasets that had the same number of observations between transfers. One of the key issues when developing this section of the macro was the sorting order used for data comparison. To solve this issue, the _ALL_ keyword was used and it worked for all domains except for laboratory data. It was the experience of the author that a separate sorting order was needed to make a useful comparison for laboratory data namely <SUBJID LBCAT LBTESTCD LBTEST LBDTC VISITNUM VISIT LBORRES>. For every domain a macro call is made to obtain the data comparison. The macro call is automated using CALL EXECUTE and works in the same way as described in part one. The logic to determine the parsing and validity of variables to be dropped for that domain (macro parameter, DROPVAR) has not been covered in the paper. Logic to highlight variables present in one transfer but not the other per domain is also included. The macro variable DROPVAR used in the code below is different from one explained in the macro parameters section. The section of code related to part two is described below.

```
%macro allcmp(inds=);

       proc sort data=prevdir.&inds out=src1;
              by _all_;
       run;

       proc sort data=crtdir.&inds out=src2;
              by _all_;
       run;

       %if %upcase(&inds)=LB  %then %do;
              proc sort data=prevdir.lb out=src1;
               by usubjid lbcat lbtestcd lbtest lbdtc visitnum visit lborres;
              run;

              proc sort data=crtdir.lb out=src2;
               by usubjid lbcat lbtestcd lbtest lbdtc visitnum visit lborres;
              run;
       %end;

proc compare base=src1
       %if (&dropvar)^= %then %do;(drop=&dropvar) compare=src2(drop=&dropvar)
       %end;
       %else %if (&dropvar)= %then %do;compare=src2 %end;
              outnoequal outbase outcomp outdiff noprint out=compare;
run;

proc datasets library=work noprint;
delete src1 src2;
quit;

 /* Determine commonality of variables */
proc contents data=crtdir.&inds out=content_1(keep=name) noprint;
run;

proc contents data=prevdir.&inds out=content_2(keep=name) noprint;
run;

data content_1;
       set content_1;
       name=upcase(name);
run;

data content_2;
       set content_2;
       name=upcase(name);
run;

proc sort data=content_1;
by name;
run;

proc sort data=content_2;
by name;
run;

data crtdir prevdir;
       merge content_1(in=a) content_2(in=b);
       by name;
       if a and not b then output crtdir;
       if b and not a then output prevdir;
run;
```

```
      data crtdir;
             set crtdir;
             src="CURRENT TRANSFER";
      run;

      data prevdir;
             set prevdir;
             src="OLD TRANSFER";
      run;

  /* Determining common variables and create output for variables not common */
      ods tagsets.ExcelXP options(Sheet_Name="&inds");

      proc print data=compare;
      run;

      title 'Variables in Current transfer but NOT in old transfer ';
      proc print data=crtdir;
      run;

      title 'Variables in old transfer but NOT in current transfer ';
      proc print data=prevdir;
      run;

      proc datasets library=work noprint;
      delete    crtdir prevdir;
      quit;
  %mend;

/* invoke macro for data comparison */

      data x;
             length allcmp $200;
             set cmp end=eof;
             allcmp=cats('%allcmp(inds=',domain,')');
             call execute(allcmp);
      run;
```

CONCLUSION

Even though the title suggests the timing of the changes be tracked around database lock, this process can be implemented at earlier stages to ensure that the current transfer is not the same as the previous transfer before further analysis.

REFERENCES

[1] Wang, Hui. Creating data driven SAS code with CALL EXECUTE. PharamSUG2015, Orlando, FL.
[2] Ramalingam, Sanjiv. Automating the pooling of variables across multiple datasets using PROC SQL and SAS macro. PharmaSUG2009, Portland, OR.

CONTACT INFORMATION

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

Sanjiv Ramalingam
Biogen Inc.
300 Binny Street
Cambridge, MA - USA
Email: Sanjiv.ramalingam@biogen.com