PharmaSUG 2018 – Paper EP-25

A SAS® Macro to Create Validation Summary of Dataset Report

Zemin Zeng, Sanofi, Bridgewater, NJ

ABSTRACT

This paper will introduce a short SAS macro developed at work to create summary report of dataset validation. The code of the SAS macro is included in the paper and can be easily modified to fit reader's need. The macro is warmly welcomed by the programming group at work and has proven to be very useful and efficient.

INTRODUCTION

In Pharmaceutical industry, a statistical programming team is usually assigned to support multiple ongoing clinical studies. It is very demanding for a programming lead to manage programming activities effectively, keep a clear big picture of the team progress and document study related files in a well-organized way. Inspired by Chen's paper (Chen, Y. PharmaSUG, 2010), we developed a short SAS macro to create validation summary of dataset report. We shall share the details of the macro in this paper.

The organization of the paper would be the following. First we will introduce the settings of the macro and the target output macro creates. Second we will state the key steps in the macro with detailed explanation of technical parts. Finally the code of the macro is attached in the Appendix. We hope our paper will be easily adopted by interested users to serve their needs.

SETTING OF THE MACRO

In a particular clinical later phase study, our statistical programming team need to create large amount of SDTM and ADaM datasets and then independently validated them. It is very desirable to have a central file to document the details of dataset creation and validation status. As shown in below Table **1**.

Table 1: Study level SDTM/ADaM data validation report

Page 1 of 1

Project Code / Study Number / Analysis: SARxxxxx / EFCxxxx / CSR_2

Summary of validation Dataset Report

Primary Data	Owner	Creation Date	Validation Output	Validation Date	Validator	Validation Findings	Validation Findings Clean Data
AE	Programmer1	02Jan2018:17:54:34	VAE_DIFF	02Jan2018:20:42:05	Programmer2	Matched!	Matched!
DM	Zemin	02Jan2018:17:53:41	VDM_DIFF	02Jan2018:20:41:52	Programmer3	Difference Records: 30%	Difference Records: 50%
DS	Zemin	02Jan2018:17:54:00	VDS_DIFF	02Jan2018:21:07:09	Programmer3	Validation Date is Before Primary	Validation Date is Before Primary
ADSL	Zemin	03Jan2018:16:23:47	VADSL_DIFF	03Jan2018:16:29:19	Programmer4	Validation Output Data Not Exist	Validation Output Data Not Exist

The dataset name, primary programmer and validator columns come from the study assignment Excel sheet. At the study planning stage, we will make up with a tracking excel file for datasets in which all datasets need to be created are listed and corresponding primary and validation programmers are assigned (like **Table 2** below).

The last column information in **Table 1** is based on validation comparison and data clean status. In our studies, we receive monthly data transfer along with data clean status provided by data management. The clean status file is a patient level data, basically says which patient's data is cleaned (identified by USUBJID). At the beginning of the study, usually there are only a few patients. As study progresses to the end like final database lock milestone, all patients in the study should be cleaned.

The column 'Validation Output' is the dataset name of the comparison output data. In the validation dataset program, we ask validator to output the difference dataset with name convention Vxx_DIFF where xx is the name of the main dataset. For example, in ADAE dataset validation program, the output difference dataset will be named as VADAE_DIFF. We ask individual programmer to include code like below:

```
title2 "COMPARISON: ADAE between primary and validation";
proc compare data=add.adae compare=qcd.v_adae listall out=qcd.vadae_diff outnoequal
  outbase outcomp outdif criterion=0.0000001;
  id usubjid AEREFID aedecod aseq astdt asttm;
run;
```

Table 2: Assignments on SDTM/ADaM dataset creation and validation

Dataset Name	Statistical Programmer Name	Program Name	Target Completion Date	Date Ready for QC	Validation Programer Name	QC program Name
AE	Programmer1	ae.sas			Programmer2	v_ae.sas
DS	Zemin	ds.sas			Programmer3	v_ds.sas
DM	Zemin	dm.sas			Programmer3	v_dm.sas
ADSL	Zemin	adsl.sas			Programmer4	v_adsl.sas

KEY STEPS IN THE MACRO

The macro %**validation_data** (See **APPENDIX** for the complete code) has only one input parameter which is primary dataset Library. In order to create summary report as in **Table 1** for both SDTM and ADaM datasets, the macro will be called two times, one is for SDTM dataset library, and the second one is the ADaM dataset library. If just calls the macro once (say for SDTM library), then the summary report will only consists of SDTM dataset validation information. Below are the steps the macro will perform:

Step 1: check the input library datasets, create macro variable to loop through each dataset;

Step 2: get each dataset file attributes. The key SAS function will be FOPTNAME (see the link in the reference for the detailed information). %sysfunc(FOPTNSME(fileid)) returns the number of attributes of a given file, %sysfunc(FOPTNSME(fileid, i)) returns the ith position attribute name, %sysfunc(FINFO(fileid, attribute name)) returns the attribute name value (See **Table 3** for complete list of attribute names and their values). In the macro output **Table 1**, Creation date and Validation date columns are both from the Last Modified value in **Table 3** of primary and validation datasets, respectively.

Table 3: File attributes from FOPTNAME

	A	B
1	**Name**	**Value**
2	Filename	/sasmeta/home/.../SDTM/DATA/ae.sas7bdat
3	Owner Name	xxxx
4	Group Name	users
5	Access Permission	-rw-rw-r--
6	Last Modified	19Dec2017:18:54:39
7	File Size (bytes)	3907584

Step 3: Subset dataset with only clean patients, the last column in the macro output **Table 1** is the comparison results of subset datasets. Based on the study needs, this step can be skipped if clean patient list is not applicable.

Step 4: Analyze the comparison results (saved in vxx_diff dataset) between primary and validation datasets. The macro analyzes the difference dataset (eg. vadae_diff), the contents of the last two columns in the macro output **Table 1** are from the analysis results. We list the possible values of the last second column in the macro output **Table 1** as below. The same applies to the last column with an addition restriction to clean patients.

Validation Findings	Conditions
Matched	If update difference dataset has observation 0.
Difference Records: xx%	xx is the number of records of difference dataset divide 3.
Validation Date is Before Primary	If the validation date/time is prior to the primary dataset.
Validation Output Data Not Exist	If validation dataset doesn't exist

CONCLUSION

The SAS macro **%validation_data** is widely used in many clinical studies at work. After every data refresh and dry run, the macro output gives the programming team a nice summary of validation status, inform individual programmer to look into their data for potential issues. The macro output records the details of study dataset validation and serves as a good study document for important study milestones, like final database lock. We highly recommend peers to develop similar macro to manage dataset validations for their clinical studies.

REFERENCES

Chen Yang (2010), *Let your title macro report study progress*, Proceedings of the PharmaSUG 2010, http://www.pharmasug.org/cd/papers/AD/AD07.pdf

SAS FOPTNAME Function,
http://support.sas.com/documentation/cdl/en/lrdict/64316/HTML/default/viewer.htm#a000209587.htm

ACKNOWLEDGMENTS

CONTACT INFORMATION

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

Zemin Zeng, Sanofi, Bridgewater, NJ
Email: *zengzemin@gmail.com* / *zemin.zeng@sanofi.com*

APPENDIX: THE CODE OF THE MACRO

```
/*****************************************************************
* Program Name               : validation_data.sas
* Program Purpose            : To create dataset validation report
* Compound/Study/Analysis    : SARxxxxx/xxxxx/CSR
* Program Author             : Zemin Zeng
* Program creation date      : 2017-01-15
* Input data files           : Library name for primary datasets
* Output data files          : Validation_data_report_SDD/ADD
* System : SAS version 9.4 -  WISE environment
* Macro call Sample          : %validation_data(dtpart=SDD)
*****************************************************************/

/* ========================= Beginning of Code ================== */
```

```
%macro validation_data(dtpart=SDD);
* Step 1: check the input library datasets, create macro variable to loop through each
dataset;
 proc sql;
   create table toc as
   select libname, memname, memtype from dictionary.members
   where libname="&dtpart" and memtype='DATA';
 quit;

*** get number of datasets and save it in macro variable &nod;
data _null_;
   set toc end=eof; if eof then call symput('nod', compress(_n_));
run;


* Step 2: get each dataset file attributes;
%macro att(outdt=, pre=p);
   DATA &outdt (keep=&pre.dt &pre.own &pre.tm nkey);
       fid = FOPEN("_file"); numopts = FOPTNUM(fid);
    DO j = 1 TO numopts;
        optname = FOPTNAME(fid,i); optval = FINFO(fid, optname);
       if j = 1 then txt=optval;
       &pre.dt=upcase(scan(reverse(scan(reverse(txt), 1 , '/')), 1, '.'));
        if j = 2 then &pre.own=optval; if j = 5 then &pre.tm=optval;
    END;
       nkey=&i;
   RUN;
  %mend;

/*To get the subset of cleaned patients*/
proc sort data=sdd.cleanpid out=cleanpid ; by usubjid; where upcase(clean)='YES'; run;
*** Loop through each dataset in TOC;
%do i=1 %to &nod;
   data _null_; set toc; if _n_=%eval(&i);
     call symput('dtset', lowcase(trim(left(memname))));
   run;

/*if corresponding validation dataset not exist*/
%IF not %sysfunc(exist(qcd.v&dtset._diff)) %then %do;
  %put _ERR: Validation of &dtpart..&dtset output diff dataset does not exist!;
   filename _file "%sysfunc(pathname(&dtpart))/&dtset..sas7bdat";
  %att(outdt=prim, pre=p);

  data _rpt; set prim(keep=p: nkey); length vdt vown vtm $200; vdt='NA'; vown='NA';
     vtm='NA'; vobsr=.; vobs1=.; vobsc=.;
     %if &dtpart=SDD %then %do; sdtm=0; %end;
     %else %do; sdtm=1; %end;
  run;
%end;
%else %do;
  %put v&dtset._diff.sas7bdat exist;

 * Step 3: Subset dataset with only clean patients;
  %let dsid = %sysfunc(open(qcd.v&dtset._diff));
  %if (&dsid) %then %do;
     %if %sysfunc(varnum(&dsid,USUBJID)) %then %do;
           proc sort data=qcd.v&dtset._diff out=v&dtset._diff; by usubjid; run;

           data cln_&dtset._diff;
              merge cleanpid(keep=usubjid in=a) v&dtset._diff(in=b);
             by usubjid; if a and b;
            run;
        %end;
        %else %do ;
          data cln_&dtset._diff;  set  qcd.v&dtset._diff; run;
       %end;
```

```
 %end;
%let rc = %sysfunc(close(&dsid));

/*to subset to cleanpid--primary*/
 %let dsid = %sysfunc(open(&dtpart..&dtset));
 %if (&dsid) %then %do;
    %if %sysfunc(varnum(&dsid,USUBJID)) %then %do;

          data pc&dtset;
            merge cleanpid(keep=usubjid in=a) &dtpart..&dtset(in=b);
           by usubjid; if a and b;
          run;
       %end;
       %else %do ;
       data pc&dtset;  set &dtpart..&dtset; run;
    %end;
 %end;
%let rc = %sysfunc(close(&dsid));

 %LET DSID = %SYSFUNC(OPEN(qcd.v&dtset._diff));
   %LET NUMOBS =%SYSFUNC(ATTRN(&DSID,NLOBS)); %LET RC = %SYSFUNC(CLOSE(&DSID));
 %LET DSID = %SYSFUNC(OPEN(&dtpart..&dtset));
  %LET PNUMOBS =%SYSFUNC(ATTRN(&DSID,NLOBS)); %LET RC = %SYSFUNC(CLOSE(&DSID));
 %LET DSID = %SYSFUNC(OPEN(work.cln_&dtset._diff));
  %LET CNUMOBS =%SYSFUNC(ATTRN(&DSID,NLOBS)); %LET RC = %SYSFUNC(CLOSE(&DSID));
 %LET DSID = %SYSFUNC(OPEN(work.pc&dtset));
  %LET CPNUMOBS =%SYSFUNC(ATTRN(&DSID,NLOBS)); %LET RC = %SYSFUNC(CLOSE(&DSID));

 %IF &NUMOBS ne 0 %THEN %DO; %put The number of Observations= &NUMOBS;  %END;

 filename _file "%sysfunc(pathname(&dtpart))/&dtset..sas7bdat";
 %att(outdt=prim, pre=p);

 filename _file "%sysfunc(pathname(QCD))/v&dtset._diff.sas7bdat";
 %att(outdt=val, pre=v);

 data _rpt;
   merge prim(keep=p: nkey) val(keep=v: nkey);
   by nkey;
   vobs1=&NUMOBS;  vobsr=%sysevalf(100*(&NUMOBS/&PNUMOBS)/3);
   %if &CPNUMOBS ne 0 %then %do; vobsc=%sysevalf(100*(&CNUMOBS/&CPNUMOBS)/3); %end;
   %if &CPNUMOBS = 0 %then %do; vobsc=9999; %end;
   %if &dtpart=SDD %then %do; sdtm=0; %end; %else %do; sdtm=1; %end;
 run;
%end; /*end with validation dataset not zero obs*/
proc append base=_result data=_rpt force;
run;
%end; /*loop through dataset end*/

/*************************************************
           Reporting  handling
*************************************************/

*Step 4: Analyze the comparison results;
data _result;
 set _result;
 length  des desc $50;
 if vobsr=. then des='Validation Output Data Not Exist';
 else if vobsr=0 then des='Matched!';
 else des='Difference Records: ' || strip(put(vobsr, 5.2)) || '%';
 if vobsc=. then desc='Validation Output Data Not Exist';
 else if vobsc=0 then desc='Matched!';
 else desc='Difference Records: ' || strip(put(vobsc, 5.2)) || '%';
 if vobsc=9999 then desc='No Clean Data';
 Sig='';
 if length(ptm)=18 then do;
```

```
   _pdt1=substr(ptm, 1, 9); _pdt2=substr(ptm, 10); dt=input(strip(_pdt1), date9.);
   _pdtc=strip(put(dt, yymmdd10.)) || "T" || _pdt2;
 end;
 if length(vtm)=18 then do;
   _vdt1=substr(vtm, 1, 9); _vdt2=substr(vtm, 10); _vdt=input(strip(_vdt1), date9.);
   _vdtc=strip(put(_vdt, yymmdd10.)) || "T" || _vdt2;
   if _vdtc<_pdtc then do;
      des='Validation Date is Before Primary';
         desc='Validation Date is Before Primary';
   end;
 end;
 format _vdt yymmdd10.;
run;

proc print data=_result; var sdtm pdt  pown ptm vdt vown vtm des: sig; run;


*to output listing use gstars macro;
 %if &dtpart=ADD %then %do;
/*To get the primary and validator's names from the tracking sheet */
 data _track(keep=sdtm pdt pown vown);
  infile "%sysfunc(pathname(doco))/valid_data.csv"  truncover firstobs=2;
  input line $200.;
  sdtm=input(scan(line, 1, ','), best.); pdt=scan(line, 2, ','); pown=scan(line, 3,
',','); vown=scan(line, 4, ',');
 run;

 proc sort; by sdtm pdt; run;

 data _result;
    merge _result(in=a drop=pown vown) _track;
    by sdtm pdt; if a;
  run;
/*Call Sanofi macro to create rft output*/
%gslist(filename = Validation_data_report
        ,pgmname = validation
        ,location=QC
       ,dataset  = _result
       ,columns  =  pdt  pown ptm vdt vtm vown  des desc
       ,vardef   = pdt     [wrap=y      label="Primary \par Data"]
                   pown    [wrap=y      label="Owner"]
                   ptm     [wrap=y      label="Creation \par Date"]
                  vdt      [wrap=y      label="Validation \par Output"]
                  vtm      [wrap=y      label="Validation \par Date"]
                  des      [wrap=y      label="Validation \par Findings"]
                  vown     [wrap=y      label="Validator"]
             desc     [wrap=y      label="Validation \par Findings \par Clean Data"]
       , varby        =
       , varbylab     =
       ,orderby    =
       ,firstonly  =
       ,breakafter =
       ,title1     =%str(Summary of validation  Dataset Report)
       ,options   = dest=app fontsize=8 stretch=Y orient=l Freflist=N &GlbOpt4GS
                        colsize=0.6 0.6 1 0.8 1 0.6 1.6 1.6 );

  %end;
options mprint mlogic missing='';
*Stop the macro;
%exit:
%mend;

/*Macro call to output validation report*/
%validation_data(dtpart=SDD);
%validation_data(dtpart=ADD);
```