PharmaSUG 2021 - Paper EP-191

Only Get What You Need - To Simplify Analysis Data Validation Report from PROC COMPARE Output

Wenjun He
The Emmes Company, LLC, Rockville, MD

ABSTRACT

Independent programming is a gold standard validation method to generate accurate analysis data sets in biotechnology and pharmaceutical companies. A timely and efficient validation of derivative analysis datasets from dynamic source data is demanding for clinical programmers in phase I trials. SAS® software provides PROC COMPARE as a useful tool to identify differences between two data sets. In order to efficiently keep monitoring the validation status of the whole set of a study trial specific analysis datasets which are constantly updated to match the dynamic source data, this paper introduces the techniques to speed up identifying the discrepancies between each pair of production and validation datasets in various SAS® libraries and simplify the generation of a straightforward validation report by extracting and reorganizing the output from PROC COMPARE.

INTRODUCTION

Independently double programming is the gold standard to generate data validation in pharmaceutical industrials. Analysis Data Model (ADaM) is one of the required standards for data submission to US Food and Drug Administration (FDA) and much more effort is now focused on the accuracy of analysis datasets. In clinical Phase 1 trials, due to the trials' fast pace in subject-recruitment, ever-vigilant attention required to adverse events related to safety during the stage of dose escalation, and limited time in report preparation for safety monitor committee for the halting of trials or study termination, it is much more important to keep track of the constant updates in the source data and monitor the accuracy for data validation. The process of double programming and data validation is presented in Figure 1.

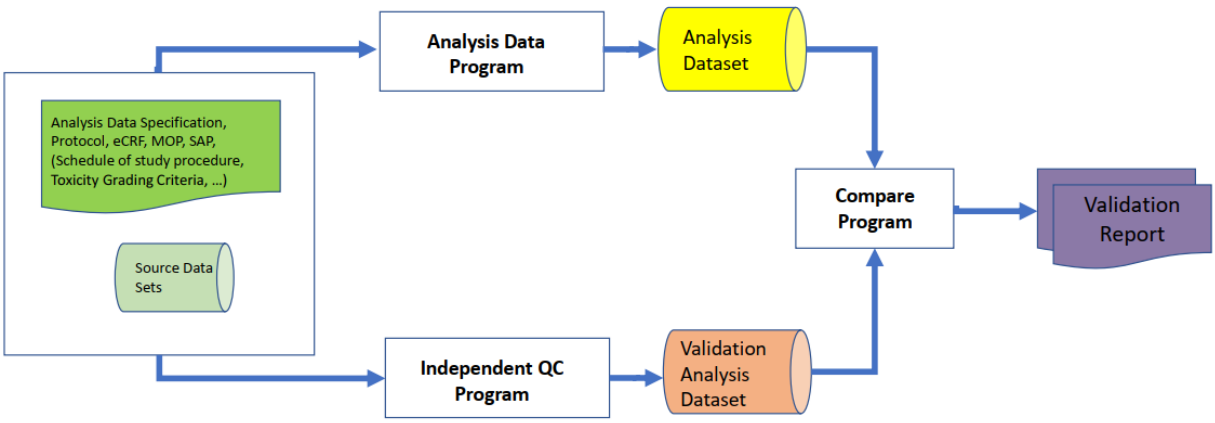

Figure 1. Analysis Data Set Validation Process Flow.

PROC COMPARE is a useful procedure in SAS that allows users to compare two datasets, to compare variables against variables of the same dataset or two datasets. In the syntax, the most commonly used options in "PROC COMPARE statement <OPTIONS>;" are "BASE=" and "COMPARE" option to specify the dataset used as the base dataset and the one for comparison. Thus, the one-to-one based comparison is the standard practice when multiple datasets compared, although it is time-consuming.

In the case of analysis data sets, the production and validation datasets are always saved in two different directories. It is time consuming to report the comparison summary if the report generation is achieved manually on a one-by-one basis. In the following section will be demonstrated the design of the workflow of the comparison, the programming and the generation of the final validation report using the output results of PROC COMPARE in datasets, which include all the information regarding the difference in the observation numbers, the attributes of variables, and the value of the variables in all pairs of analysis data sets to be compared in two separate SAS libraries.

CASE STUDY

ASSIGNMENT - TO GENERATE THE ANALYSIS DATA VALIDATION REPORT WITH THE DIFFERENCES BETWEEN TWO LIBRARIES

Following are two libraries including multiple study-specific analysis data sets – The BASE library for production data sets and the COMPARE library for validation data sets. The library for validation data lacks the counterpart for the data set ADAE in the library for production data.

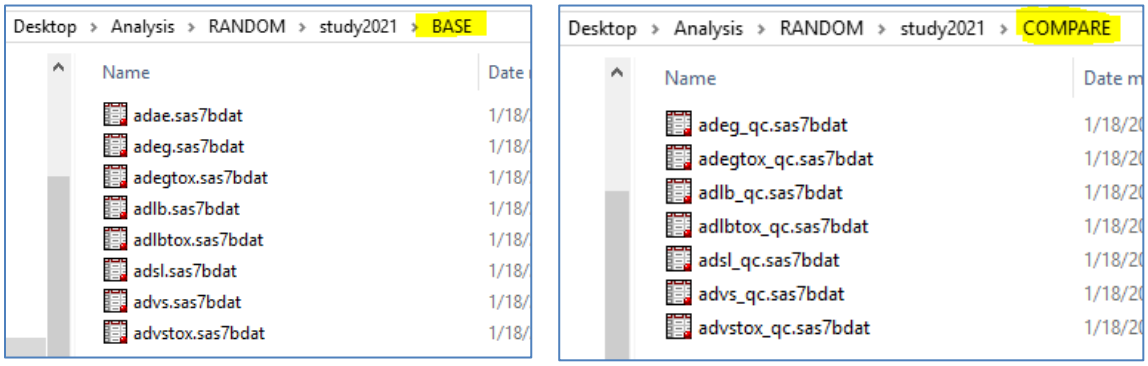

Figure 2. Two SAS Libraries to Be Compared.

OBJECTIVE - THE OUTPUT TABLE(S) ARE EXPECTED TO CONTAIN THE FOLLOWING COMPONENTS

I am firstly listing the following goals in the simplified output that include all the necessary information used to compare each pair of production dataset and validation dataset in two libraries.

1. To identify the datasets in various SAS libraries and list the uncommon datasets.
2. To list the number of observations, the number of variables, uncommon variables, modification date/time for each pair of two datasets in comparison, as well as using the system macro variable SYSINFO to summarize each comparison.
3. To capture the differences among variables compared for each pair of data sets and include all the details in one dataset.

EXAMPLE OUTPUTS FOR COMPARISON OF TWO SAS LIBRARIES INCLUDING DATA SETS WITH DISCREPANCIES

As shown in the following Table 1, in which the difference between production dataset and validation dataset is summarized side-by-side, regarding the nubmer of obseravtions, number of variables, the

modification date/time for each data set, as well as the system macro variable SYSINFO, which is a compact reporting system produced by PROC COMPARE to issuing return codes based on the outcome of comparison.

Analysis Data Set	Number of Observations (Production Data)	Number of Observations (Validation Data)	Difference in the Number of Observations	Observation Number	Modification Date/Time (Production Data)	Modification Date/Time (Validation Data)	Number of Variables (Production Data)	Number of Variables (Validation Data)	&SYSINFO
ADAE	212	-	Missing COMPARE Dataset	BASEOBS = 212	18JAN21:04:31:33	-	63	-	-
ADEG	2850	2850	N/A	BASEOBS = 2850 COMPAREOBS = 2850	18JAN21:04:24:59	18JAN21:04:24:59	69	69	44
ADEGTOX	138	618	Difference Detected in Number of Observations (480)	BASEOBS = 138 COMPAREOBS = 618	18JAN21:04:24:59	18JAN21:04:24:59	35	35	4284
ADLB	5456	5456	N/A	BASEOBS = 5456 COMPAREOBS = 5456	18JAN21:04:24:58	18JAN21:04:24:59	58	58	4156
ADLBTOX	5997	5993	Difference Detected in Number of Observations (4)	BASEOBS = 5997 COMPAREOBS = 5993	18JAN21:04:24:59	18JAN21:04:24:59	37	37	4188
ADSL	49	43	Difference Detected in Number of Observations (6)	BASEOBS = 49 COMPAREOBS = 43	18JAN21:04:24:58	18JAN21:04:24:58	70	71	14460
ADVS	1632	1636	Difference Detected in Number of Observations (4)	BASEOBS = 1632 COMPAREOBS = 1636	18JAN21:04:24:59	18JAN21:04:24:59	54	54	4284
ADVSTOX	2557	2570	Difference Detected in Number of Observations (13)	BASEOBS = 2557 COMPAREOBS = 2570	18JAN21:04:24:59	18JAN21:04:25:00	37	53	6332

Table 1. PROC COMPARE Validation Report: Production Data vs. Validation Data.

In Table 2, more details about the differences between these two SAS libraries are demonstrated, including the common datasets and the missing datasets, as well as the differences in attributes and values of all the variables in these two libraries being summarized.

Discription	Number of Data Sets or Variables
NUMBER OF DATASETS IN THE BASE DIRECTORY:: ADAE ADEG ADEGTOX ADLB ADLBTOX ADSL ADVS ADVSTOX	8
NUMBER OF DATASETS IN THE COMPAREDIRECTORY:: ADEG ADEGTOX ADLB ADLBTOX ADSL ADVS ADVSTOX	7
NUMBER OF DATASETS IN BOTH DIRECTORIES:: ADEG ADEGTOX ADLB ADLBTOX ADSL ADVS ADVSTOX	7
NUMBER OF DATASETS MISSING IN COMPARE DIRECTORY:: ADAE	1
NUMBER OF DATASETS WITH DIFFERENCES	8
DATASETS WITH DIFFERENCES IN THE NUMBER OF OBSERVATIONS	5
NUMBER OF VARIABLES WITH DIFFERENCE IN NAMES	19
NUMBER OF VARIABLES WITH DIFFERENCE IN LENGTH	50
NUMBER OF VARIABLES WITH DIFFERENCE IN LABELS	169
NUMBER OF VARIABLES WITH DIFFERENCE IN FORMAT	100
NUMBER OF VARIABLES WITH DIFFERENCE IN INFORMAT	96
NUMBER OF VARIABLES WITH DIFFERENCE IN TYPE (CHAR/ NUM)	23
TOTAL NUMBER OF VARIABLES WITH DIFFERENCES REPORTED	96

Table 2. PROC COMPARE Validation Summary Report.

In Table 3, all of the differences for each variable are listed and categorized by dataset, attribute, and value. Only excerpt(s) of this long table are shown as below. The value of &SYSINFO should match the column of “Description in the Difference”. For &SYSINFO that is less than 64, the datasets in comparison only have differences in the attributes of each variables (e. g. the variables in ADEG). However, the large

&SYSINFO is always resulted from the differences in the values of variables in comparison (e. g. the variables in ADSL).

DATASET	Details for the Difference: Variable Name or Number of Observations	Observation Number	Description in the Difference	&SYSINFO
ADAE	BASEOBS = 212	-	Missing COMPARE Dataset	-
ADEG	BASEOBS = 2850 COMPAREOBS = 2850	-	N/A	44
	A1HI	-	Difference Detected in LABEL: BASE = Analysis Range 1 Upper Limit-AVAL Gr1 COMPARE =	44
	A1LO	-	Difference Detected in LABEL: BASE = Analysis Range 1 Lower Limit-AVAL Gr1 COMPARE =	44
	A2HI	-	Difference Detected in LABEL: BASE = Analysis Range 2 Upper Limit-AVAL Gr2 COMPARE =	44
	A2LO	-	Difference Detected in LABEL: BASE = Analysis Range 2 Lower Limit-AVAL Gr2 COMPARE =	44
	A3LO	-	Difference Detected in LABEL: BASE = Analysis Range 3 Lower Limit-AVAL Gr3 COMPARE =	44
ADSL	COHORT	-	Difference Detected in LABEL: BASE = Cohort COMPARE =	14460
	TRTSTM	-	Difference Detected in LABEL: BASE = Time of First Exposure to Treatment COMPARE =	14460
	SAFFL	-	Difference Detected in LABEL: BASE = Safety Population Flag COMPARE =	14460
	AARMGR1	-	Difference Detected in LABEL: BASE = Analysis Arm Group 1 COMPARE =	14460
	BMIU	-	Difference Detected in LABEL: BASE = Body Mass index Units COMPARE = Standard Units	14460
	DCSREAS	-	Difference Detected in LABEL: BASE = Reason for Discontinuation from Study COMPARE =	14460
	DCSREASP	-	Difference Detected in LABEL: BASE = Reason Spec for Discont from Study COMPARE =	14460
	TRTSTM	39	Difference in VALUE Detected: ADSL.TRTSTM BASE = 8:00 COMPARE = .	14460
	SAFFL	39	Difference in VALUE Detected: ADSL.SAFFL BASE = Y COMPARE = N	14460
	AEFL	43	Difference in VALUE Detected: ADSL.AEFL BASE = Y COMPARE = N	14460
	AERELFL	34	Difference in VALUE Detected: ADSL.AERELFL BASE = N COMPARE = Y	14460

Table 3. PROC COMPARE Validation Listing Report.

EXAMPLE OUTPUTS FOR COMPARISON OF TWO SAS LIBRARIES WITH DATA SETS TO MATCH PERFECTLY

The following output Table 4 – 6 are for the scenario in which each pair of the datasets in two SAS libraries is a perfect match.

Analysis Data Set	Number of Observations (Production Data)	Number of Observations (Validation Data)	Difference in the Number of Observations	Observation Number	Modification Date/Time (Production Data)	Modification Date/Time (Validation Data)	Number of Variables (Production Data)	Number of Variables (Validation Data)	&SYSINFO
ADAE	212	212	N/A	BASEOBS = 212 COMPAREOBS = 212	18JAN21:04:31:33	18JAN21:04:31:33	63	63	0
ADEG	2850	2850	N/A	BASEOBS = 2850 COMPAREOBS = 2850	18JAN21:04:24:59	18JAN21:04:24:59	69	69	0
ADEGTOX	138	138	N/A	BASEOBS = 138 COMPAREOBS = 138	18JAN21:04:24:59	18JAN21:04:24:59	35	35	0
ADLB	5456	5456	N/A	BASEOBS = 5456 COMPAREOBS = 5456	18JAN21:04:24:58	18JAN21:04:24:58	58	58	0
ADLBTOX	5997	5997	N/A	BASEOBS = 5997 COMPAREOBS = 5997	18JAN21:04:24:59	18JAN21:04:24:59	37	37	0
ADSL	49	49	N/A	BASEOBS = 49 COMPAREOBS = 49	18JAN21:04:24:58	18JAN21:04:24:58	70	70	0
ADVS	1632	1632	N/A	BASEOBS = 1632 COMPAREOBS = 1632	18JAN21:04:24:59	18JAN21:04:24:59	54	54	0
ADVSTOX	2557	2557	N/A	BASEOBS = 2557 COMPAREOBS = 2557	18JAN21:04:24:59	18JAN21:04:24:59	37	37	0

Table 4. PROC COMPARE Validation Report: Production Data vs. Validation Data for Perfect-Match.

Discription	Number of Data Sets or Variables
NUMBER OF DATASETS IN THE BASE DIRECTORY:: ADAE ADEG ADEGTOX ADLB ADLBTOX ADSL ADVS ADVSTOX	8
NUMBER OF DATASETS IN THE COMPAREDIRECTORY:: ADAE ADEG ADEGTOX ADLB ADLBTOX ADSL ADVS ADVSTOX	8
NUMBER OF DATASETS IN BOTH DIRECTORIES:: ADAE ADEG ADEGTOX ADLB ADLBTOX ADSL ADVS ADVSTOX	8
NUMBER OF DATASETS WITH NO DIFFERENCES	8

Table 5. PROC COMPARE Validation Summary Report for Perfect-Match.

DATASET	Details for the Difference: Variable Name or Number of Observations	Observation Number	Description in the Difference	&SYSINFO
ADAE	BASEOBS = 212 COMPAREOBS = 212	-	N/A	0
ADEG	BASEOBS = 2850 COMPAREOBS = 2850	-	N/A	0
ADEGTOX	BASEOBS = 138 COMPAREOBS = 138	-	N/A	0
ADLB	BASEOBS = 5456 COMPAREOBS = 5456	-	N/A	0
ADLBTOX	BASEOBS = 5997 COMPAREOBS = 5997	-	N/A	0
ADSL	BASEOBS = 49 COMPAREOBS = 49	-	N/A	0
ADVS	BASEOBS = 1632 COMPAREOBS = 1632	-	N/A	0
ADVSTOX	BASEOBS = 2557 COMPAREOBS = 2557	-	N/A	0

Table 6. PROC COMPARE Validation Listing Report for Perfect-Match.

STEPS AND LOGIC IN PROGRAMMING

To get the output tables as listed above, the following logic or steps are used for programming.

Step 1: Set up the libraries which have datasets to be compared:
Step 2: Read in datasets from each library and output the observation number/variable name for each pair of datasets side-by-side. In SASHELP library, SASHELP.VTABLE and SASHELP.VCOLUMN are used to capture dataset's number of observations, name, and other metadata such as variable name, length, label, informat, format.
Step 3: To see whether two data sets match or not, the first thing to confirm is to check the observation number in each data set. The following program is to use BASEOBS and COMPAREOBS to generate one summary table OBS to show the common/uncommon datasets as well as the difference in the number of observations for the datasets in comparison.
Step 4: Using the two data sets generated in STEP 2, BASEVAR and COMPAREVAR, to generate another dataset VAR containing the differences in the variable(s)' characteristics is generated after TRANSPOSE procedure.
Step 5: Comparing the data using PROC COMPARE - Using macro to compare on a one-by-one basis, and then combining all output data sets for form one ALLCOMPARE dataset.
Step 6: Combine OBS, VAR, ALLCOMPARE to generate a final data set.
Step 7: Using PROC REPORT to generate report.

CONCLUSION

This paper uses the techniques to extract and reorganize the output from PROC COMPARE as well as the SASHELP library and the system macro variable &SYSINFO to simplify the validation reports for the datasets present in various SAS libraries, making it easy to efficiently keep monitoring the validation status of the whole set of a study trial specific analysis datasets.

CONTACT INFORMATION

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

Wenjun He
The Emmes Company, LLC
whe@emmes.com

APPENDIX

THE EXAMPLE PROGRAM

STEP 1

```
%let BASE=C:\Users\whe\Desktop\Analysis\RANDOM\study2021\BASE;
%let COMPARE=C:\Users\whe\Desktop\Analysis\RANDOM\study2021\COMPARE;
libname BASE "&BASE.";
libname COMPARE "&COMPARE.";

%macro YNFORMAT(FORMAT=);
%if %upcase(&FORMAT) in(YES, Y) %then %do; options FMTSEARCH =(BASE COMPARE); %end;
%if %upcase(&FORMAT) in(NO, N) %then %do; options nofmterr; %end;
%mend YNFORMAT;
%YNFORMAT(FORMAT=Y);
```

STEP 2

```
%macro getmeta(lib);
proc sql;
     create table &lib.obs as
     select distinct memname, nobs as &lib.OBS, modate as &lib.modat, nvar as
&lib.nvar
     from sashelp.vtable
        where libname="&lib." and memtype="DATA"
        order by memname;

        create table &lib.var as
        select memname, name, length, label, format, informat, xtype
        from sashelp.vcolumn
        where libname="&lib." and memtype="DATA"
        order by memname, name;
quit;

data &lib.obs;
     set &lib.obs;
     %if &lib.=COMPARE %then %do;
     memname=scan(memname, 1, "_");
        %end;
run;
proc sort;
    by memname;
run;

data &lib.var;
     set &lib.var;
```

```
     %if &lib.=COMPARE %then %do;
     memname=scan(memname, 1, "_");
        %end;
run;
proc sort;
     by memname;
run;
%mend getmeta;
%getmeta(BASE);
%getmeta(COMPARE);
```

STEP 3

```
/*Side-by-side compare the number of observations and varibles in two data sets
compared*/
data OBS;
     merge BASEOBS (in=a) COMPAREOBS (in=b);
        by memname;
        format DIFF $200. Variables $200.;
        if BASEOBS NE COMPAREOBS & not missing(BASEOBS) & not missing(COMPAREOBS)
          then DIFF = propcase("DIFFERENCE DETECTED IN NUMBER OF OBSERVATIONS (
")||strip(put(abs(BASEOBS - COMPAREOBS),8.))||" )";
     if BASEOBS = . then do; DIFF = propcase("MISSING BASE DATASET"); Variables =
"COMPAREOBS = "|| strip(put(COMPAREOBS,8.)); end;
     if COMPAREOBS = . THEN do; DIFF =propcase("MISSING COMPARE DATASET"); Variables =
"BASEOBS = "|| strip(put(BASEOBS,8.)); end;
     if DIFF NE "" and SCAN(DIFF,1) = propcase("DIFFERENCE") then Variables = "BASEOBS
= "|| strip(put(BASEOBS,8.)) ||"  COMPAREOBS = "|| strip(put(COMPAREOBS,8.));
        if DIFF EQ "" then do; DIFF = "N/A"; Variables = "BASEOBS = "||
strip(put(BASEOBS,8.)) ||"  COMPAREOBS = "|| strip(put(COMPAREOBS,8.)); end;
     DIFF=tranwrd(DIFF, "In", "in");
     DIFF=tranwrd(DIFF, "Of", "of");
        DIFF=tranwrd(DIFF, "Base", "BASE");
        DIFF=tranwrd(DIFF, "Compare", "COMPARE");
run;
proc sort;
    by MEMNAME ;
run;
```

STEP 4

```
%macro vartrans(ds=);
    proc transpose data=&ds.var out=&ds.var (drop=_label_) prefix=&ds.;
            var name length label format informat xtype;
               by memname name;
       run;
       proc sort;
           by memname name _name_;
       run;
%mend vartrans;
%vartrans(ds=BASE);
%vartrans(ds=COMPARE);

/*Side-by-side compare the varibles in two data sets compared*/
data VAR;
     merge BASEVAR COMPAREVAR;
        by memname name _name_;
run;

data VAR;
     merge OBS (IN = A WHERE = (BASEOBS = . OR COMPAREOBS = .) KEEP = MEMNAME BASEOBS
COMPAREOBS) VAR;
        by  memname ;
     rename NAME = Variables;
```

```
     if NOT A;
     if BASE1 NE COMPARE1;
     DIFF = propcase("DIFFERENCE DETECTED IN ") || upcase(strip(_NAME_))||":  BASE = "
|| strip(BASE1) || "   COMPARE = " || strip(COMPARE1);
       DIFF = tranwrd(DIFF, "In", "in");
     keep NAME DIFF MEMNAME;
run;
```

STEP 5

```
/* Using proc compare to compare data sets */
data datasets;
     merge BASEOBS (IN = A KEEP = MEMNAME) COMPAREOBS (IN = B KEEP = MEMNAME);
       by MEMNAME;
       if A and B;
run;
proc sql;
     select distinct(MEMNAME), count(distinct(MEMNAME)) into :ds_list separated by
"~", :ds_cnt
       from DATASETS
       order MEMNAME;
quit;
%put &ds_list;
%put &ds_cnt;

%macro comp;
   %global NO_DIFF;
   %let NO_DIFF = 0 ;
   %do i = 1 %to &DS_CNT;
      %let DS = %scan(&DS_LIST., &i., ~);
           proc compare BASE = BASE.&DS.
                        COMPARE = COMPARE.&DS._qc
                        OUTBASE OUTCOMP  OUTDIF  OUTNOEQUAL
                        OUT = COMPAREOUT;
           run;
                %let _SYSINFO=&SYSINFO;
                %put &_SYSINFO;

      PROC SQL NOPRINT;
           SELECT COUNT(*) INTO :COMPARECNT
           FROM COMPAREOUT;
      QUIT; %PUT &COMPARECNT. ;

      %IF &COMPARECNT. = 0 and &_SYSINFO=0 %THEN %DO;
      %LET NO_DIFF = %EVAL(&NO_DIFF. + 1 ) ;
      %put &no_diff.;
      DATA COMPARE;
              comp_SYSINFO=&_SYSINFO;     MEMNAME = "&DS.";
      RUN;
      %end;

         %ELSE %IF &COMPARECNT. = 0 and &_SYSINFO < 64 %THEN %DO;
      DATA COMPARE;
              comp_SYSINFO=&_SYSINFO;     MEMNAME = "&DS.";
              DIFF=""; Variables=""; _OBS_=.;
      RUN;
         %END;

      %ELSE %DO;
      OPTIONS OBS = 12;
      PROC SORT DATA = COMPAREOUT OUT = ALL ; BY _OBS_ _TYPE_ ; RUN;
      OPTIONS OBS = MAX;
      PROC TRANSPOSE DATA = ALL OUT = ALL; VAR _ALL_; BY _OBS_ _TYPE_; RUN;
      DATA BASE COMP ;
```

```
        SET ALL;
         IF _TYPE_ = "BASE" THEN OUTPUT BASE ;
         IF _TYPE_ = "COMPARE" THEN OUTPUT COMP;
      RUN;
      PROC SORT DATA = BASE ; BY _NAME_ _OBS_ _LABEL_;
      PROC SORT DATA = COMP ; BY _NAME_ _OBS_ _LABEL_;
      RUN;

      DATA COMPARE;
           MERGE BASE (DROP = _TYPE_ _LABEL_ RENAME = (COL1 = BASE))
                 COMP (DROP = _TYPE_ _LABEL_ RENAME = (COL1 = COMPARE))
           ;
           BY _NAME_ _OBS_ ;
           WHERE _NAME_ NOT IN ("_TYPE_" "_OBS_");
           IF BASE NE COMPARE ;
           DIFF = propcase("DIFFERENCE IN VALUE DETECTED:
")||upcase("&DS..")||strip(_NAME_)||(" BASE = ") ||STRIP(BASE)||"   COMPARE =
"||STRIP(COMPARE);
                  DIFF= tranwrd(DIFF, "In", "in");
                  DIFF= tranwrd(DIFF, "Value", "VALUE");
           RENAME _NAME_ = Variables;
           MEMNAME = "&DS.";     comp_SYSINFO=&_SYSINFO;
      RUN;

      DATA COMPARE;
         SET COMPARE;
         BY Variables _OBS_ ;
         RETAIN N 0;
         IF FIRST.Variables THEN N = 0;
           N + 1;
         IF N = 6 THEN DO ;
           _OBS_ = .G ;
           DIFF = propcase("DIFFERENCE IN VALUE DETECTED:
")||upcase(strip(MEMNAME))||"."||strip(Variables)||PROPCASE(": MORE THAN 5
OBSERVATIONS HAVE DIFFERENCES");
                DIFF = tranwrd(DIFF, "In", "in");
                DIFF= tranwrd(DIFF, "Value", "VALUE");
         END;
         KEEP Variables DIFF MEMNAME  _OBS_  comp_SYSINFO;
         PROC SORT; BY Variables _OBS_ ;
       RUN;
       %END;

      /*combing all compare out*/
      %IF &I. = 1 %THEN %DO;
      DATA ALLCOMPARE;
           SET COMPARE;
      RUN;
      %END;
      %ELSE %DO;
      DATA ALLCOMPARE;
           FORMAT MEMNAME $50. Variables $100. DIFF $5000.;
           SET COMPARE ALLCOMPARE;
      RUN;
      %END;
       %end;
%mend comp;
%comp;

proc sql;
     create table syscode as
       select distinct MEMNAME, comp_SYSINFO from ALLCOMPARE;
```

```
        create table OBS_code as
        select a.*, b.comp_SYSINFO from OBS as a left join SYSCODE as b
               on a.MEMNAME = b.MEMNAME;

        create table VAR_code as
        select c.*, d.comp_SYSINFO from VAR as c left join SYSCODE as d
               on c.MEMNAME = d.MEMNAME;

quit;
```

STEP 6

```
/* final data set to combine the data set OBS, VAR, ALLCOMPARE */

DATA FINAL;
  * FORMAT MEMNAME $50. Variables $100. DIFF $5000.;
   SET OBS_code (KEEP = MEMNAME Variables DIFF comp_SYSINFO)  VAR_code ALLCOMPARE;
   FORMAT MEMNAME $50. Variables $100. DIFF $5000.;
   /*SORTING*/
   IF index(DIFF, "Missing")>0 THEN DO;
      IF index(DIFF, "BASE")>0 THEN SORT1 = 1 ;
      ELSE IF index(DIFF, "COMPARE")>0 THEN SORT1 = 2 ;
   END;

   ELSE DO;
     SORT1 = 3;
     IF INDEX(DIFF, "Number of Observations")>1 THEN SORT2=1;
        ELSE IF INDEX(DIFF, "N/A")>0 THEN SORT2=0;
     ELSE IF INDEX(DIFF, "in NAME") THEN SORT2 = 2;
     ELSE IF INDEX(DIFF, "in LENGTH") THEN SORT2 = 3;
     ELSE IF INDEX(DIFF, "in LABEL") THEN SORT2 = 4;
     ELSE IF INDEX(DIFF, "in FORMAT") THEN SORT2 = 5;
     ELSE IF INDEX(DIFF, "in INFORMAT") THEN SORT2 = 6;
     ELSE IF INDEX(DIFF, "in XTYPE") THEN SORT2 = 7;
     ELSE SORT2 = 8;
     IF _OBS_ = .G THEN _OBS_ = .;

   END;
PROC SORT; BY SORT1 MEMNAME SORT2 Variables _OBS_;
RUN;
```

STEP 7

```
/* Summary of the final dataset and the output.  */

PROC SQL NOPRINT;

     SELECT COUNT(MEMNAME) INTO : BASECNT FROM BASEOBS ;
        select distinct memname into :baseds separated by " " from BASEOBS ;
     SELECT COUNT(MEMNAME) INTO : COMPARECNT FROM COMPAREOBS ;
        select distinct memname into :compareds separated by " " from COMPAREOBS ;
        select a.memname into :commonds separated by " " from BASEOBS as a, COMPAREOBS
as b where a.memname=b.memname;
        select distinct(a.memname) into :bmisgds separated by " " from BASEOBS as a,
COMPAREOBS as b where a.memname not in(select memname from compareobs);

     SELECT COUNT(DISTINCT MEMNAME) FORMAT = 3. ,
            COUNT(CASE WHEN SORT1 = 1 THEN MEMNAME ELSE "" END ) FORMAT = 3. ,
            COUNT(CASE WHEN SORT1 = 2 THEN MEMNAME ELSE "" END ) FORMAT = 3. ,
            COUNT(CASE WHEN SORT2 = 1 THEN MEMNAME ELSE "" END ) FORMAT = 3. ,
            COUNT(CASE WHEN SORT2 = 2 THEN MEMNAME ELSE "" END ) FORMAT = 3. ,
            COUNT(CASE WHEN SORT2 = 3 THEN MEMNAME ELSE "" END ) FORMAT = 3. ,
            COUNT(CASE WHEN SORT2 = 4 THEN MEMNAME ELSE "" END ) FORMAT = 3. ,
```

```
            COUNT(CASE WHEN SORT2 = 5 THEN MEMNAME ELSE "" END ) FORMAT = 3. ,
            COUNT(CASE WHEN SORT2 = 6 THEN MEMNAME ELSE "" END ) FORMAT = 3. ,
            COUNT(CASE WHEN SORT2 = 7 THEN MEMNAME ELSE "" END ) FORMAT = 3. ,
            COUNT(CASE WHEN SORT2 = 8 THEN MEMNAME ELSE "" END ) FORMAT = 3. ,
            COUNT(CASE WHEN _OBS_ = 99999 THEN MEMNAME ELSE "" END ) FORMAT = 5.
            INTO :DIFF_DS_CNT , :BASEMISS , :COMPAREMISS ,
                 :DIFF_OBS , :DIFF_NAME , :DIFF_LENGTH ,
                 :DIFF_LABEL , :DIFF_FORMAT , :DIFF_INFORMAT ,
                 :DIFF_XTYPE , :DIFF_VAR, :DIFF_GT5
            FROM FINAL;
QUIT;
%put &BASECNT &baseds &COMPARECNT &compareds &commonds &bmisgds;
%put &NO_DIFF &DIFF_DS_CNT &BASEMISS &COMPAREMISS &DIFF_OBS &DIFF_NAME &DIFF_LENGTH
                &DIFF_LABEL &DIFF_FORMAT &DIFF_INFORMAT
                &DIFF_XTYPE &DIFF_GT5 &DIFF_VAR;

/* generate data set SUMMARY */
DATA SUMMARY;
FORMAT DISC $100.PRG $10.;
%MACRO SUMMARY (SORT,PRG,DISC);
    DISC="&DISC.";
    PRG=STRIP(PUT(&&&PRG.,5.));
    SORT=&SORT.;
    OUTPUT;
%MEND SUMMARY ;

%SUMMARY (1,BASECNT,%STR(NUMBER OF DATASETS IN THE BASE DIRECTORY:: &baseds.)) ;
%SUMMARY (2,COMPARECNT,%STR(NUMBER OF DATASETS IN THE COMPAREDIRECTORY:: &compareds.))
;
%SUMMARY (3,DS_CNT,%STR(NUMBER OF DATASETS IN BOTH DIRECTORIES:: &commonds.)) ;
%SUMMARY (4,BASEMISS,%STR(NUMBER OF DATASETS MISSING IN BASE DIRECTORY)) ;
%SUMMARY (5,COMPAREMISS,%STR(NUMBER OF DATASETS MISSING IN COMPARE DIRECTORY::
&bmisgds.)) ;
%SUMMARY (6,NO_DIFF,%STR(NUMBER OF DATASETS WITH NO DIFFERENCES)) ;
%SUMMARY (7,DIFF_DS_CNT,%STR(NUMBER OF DATASETS WITH DIFFERENCES )) ;
%SUMMARY (8,DIFF_OBS,%STR(DATASETS WITH DIFFERENCES IN THE NUMBER OF OBSERVATIONS)) ;
%SUMMARY (9,DIFF_NAME,%STR(NUMBER OF VARIABLES WITH DIFFERENCE IN NAMES)) ;
%SUMMARY (11,DIFF_LENGTH,%STR(NUMBER OF VARIABLES WITH DIFFERENCE IN LENGTH)) ;
%SUMMARY (12,DIFF_LABEL,%STR(NUMBER OF VARIABLES WITH DIFFERENCE IN LABELS)) ;
%SUMMARY (13,DIFF_FORMAT,%STR(NUMBER OF VARIABLES WITH DIFFERENCE IN FORMAT)) ;
%SUMMARY (14,DIFF_INFORMAT,%STR(NUMBER OF VARIABLES WITH DIFFERENCE IN INFORMAT)) ;
%SUMMARY (15,DIFF_XTYPE,%STR(NUMBER OF VARIABLES WITH DIFFERENCE IN TYPE ( CHAR/ NUM
))) ;
%SUMMARY (16,DIFF_GT5,%STR(NUMBER OF VARIABLES WITH DIFFERENCE IN MORE THAN 5
OBSERVATIONS)) ;
%SUMMARY (17,DIFF_VAR,%STR(TOTAL NUMBER OF VARIABLES WITH DIFFERENCES REPORTED)) ;
PROC SORT DATA = SUMMARY (WHERE= (PRG NE "0")) ; BY SORT;
RUN;

ods listing close; ods noresults; ods escapechar="^";
options orientation=landscape nonumber nodate missing="-";
ods rtf file ="C:\Users\whe\Desktop\Analysis\Report\PROC_COMPARE_&sysdate9..rtf"
style=RTF;;
title"";
title1 j=l "Protocol XXXX Analysis Data Validation Report" j=r "Testing";
title3 j=c "PROC COMPARE LIBRARIES: Prodution vs. Validation";
footnote j=l "^{style [outputwidth=100% bordertopcolor=black bordertopwidth=2pt]
SYSUSERID.}"
j=r "^{style [outputwidth=100% bordertopcolor=black bordertopwidth=2pt] Page:
^{thispage} of ^{lastpage}}";

%let fsize=2;
PROC REPORT DATA=OBS_code HEADLINE SPACING=1 MISSING SPLIT="*"
```

```
    style(report)=[font_face=arial cellspacing=.5 outputwidth=10in font_size=&fsize.]
       style(header)=[background=lightgrey font_weight=bold font_size=&fsize.]
       style(column)=[font_face=arial just=center font_size=&fsize.];
    COLUMN MEMNAME BASEOBS COMPAREOBS DIFF VARIABLES BASEMODAT COMPAREMODAT BASENVAR
COMPARENVAR comp_SYSINFO;
    DEFINE MEMNAME / DISPLAY "Analysis *Data Set"    STYLE(COLUMN)=[CELLWIDTH=.75IN
just=center];
    define baseobs / display "Number of Observations *(Production Data)"
STYLE(COLUMN)=[CELLWIDTH=.845IN just=center];
    define BASEmodat  / display "Modification Date/Time *(Production Data)"
STYLE(COLUMN)=[CELLWIDTH=1.225IN just=center];
    define BASEnvar    / display "Number of Variables *(Production Data)"
STYLE(COLUMN)=[CELLWIDTH=.845IN just=center];
    define compareobs / display "Number of Observations *(Validation Data)"
STYLE(COLUMN)=[CELLWIDTH=0.845IN just=center];
    define COMPAREmodat  / display "Modification Date/Time *(Validation Data)"
STYLE(COLUMN)=[CELLWIDTH=1.225IN just=center];
    define COMPAREnvar    / display "Number of Variables *(Validation Data)"
STYLE(COLUMN)=[CELLWIDTH=.845IN just=center];
    define diff  / display "Difference in the Number of Observations"
STYLE(COLUMN)=[CELLWIDTH=1.4IN just=left];
    define variables / display "Observation Number" STYLE(COLUMN)=[CELLWIDTH=1.1IN
just=left];
       define comp_SYSINFO / display '&SYSINFO' STYLE(COLUMN)=[CELLWIDTH=0.85IN
just=center];

PROC REPORT DATA = SUMMARY  HEADLINE SPACING=1 MISSING SPLIT="*"
    style(report)=[font_face=arial cellspacing=.5 outputwidth=10in font_size=&fsize.]
       style(header)=[background=lightgrey font_weight=bold font_size=&fsize.]
       style(column)=[font_face=arial just=center font_size=&fsize.];
    COLUMN SORT DISC PRG;
    DEFINE SORT / Noprint ORDER ORDER=DATA ;
    DEFINE DISC / flow STYLE(COLUMN)=[CELLWIDTH=6IN JUST=left]
STYLE(HEADER)=[JUST=center] 'Discription' ;
    DEFINE PRG / STYLE(COLUMN)=[CELLWIDTH=1.5IN] STYLE(HEADER)=[JUST=CENTER] CENTER
'Number of* Data Sets or Variables' ;
RUN;

PROC REPORT DATA = FINAL HEADLINE SPACING=1 MISSING SPLIT="*"
    style(report)=[font_face=arial cellspacing=.5 outputwidth=10in font_size=&fsize.]
    style(header)=[background=lightgrey font_weight=bold font_size=&fsize.]
       style(column)=[font_face=arial just=center font_size=&fsize.];
    where not (missing(DIFF) and missing(Variables) and missing(_OBS_) and SORT2=8);
    COLUMN SORT1 MEMNAME SORT2 Variables _OBS_ DIFF  comp_SYSINFO;
    DEFINE SORT1 / Noprint ORDER ORDER=DATA ;
    DEFINE SORT2 / Noprint ORDER ORDER=DATA ;
    DEFINE MEMNAME / Group ORDER ORDER=DATA STYLE(COLUMN)=[CELLWIDTH=0.75IN]
STYLE(HEADER)=[JUST=center] 'DATASET';
    DEFINE Variables / Group ORDER ORDER=DATA
                       STYLE(COLUMN)=[CELLWIDTH=1.25IN] STYLE(HEADER)=[JUST=center]
'Details for the Difference: Variable Name or Number of Observations' ;
    DEFINE _OBS_ / Group ORDER ORDER=DATA
                   STYLE(COLUMN)=[CELLWIDTH=0.85IN] STYLE(HEADER)=[JUST=center] center
'Observation* Number' ;
    DEFINE DIFF / flow STYLE(COLUMN)=[CELLWIDTH=5.5IN JUST=left]
STYLE(HEADER)=[JUST=center]  'Description in the Difference' ;
       DEFINE comp_SYSINFO / display '&SYSINFO' STYLE(COLUMN)=[CELLWIDTH=0.8IN
just=center];
RUN;
ods rtf close;
title;
footnote;
options missing=.;
```