

When PROC COMPARE is not enough - the fast check of variables discrepancy through thousands of observations

Roman Chekhun, Quartesian, Kyiv, Ukraine

ABSTRACT

An important part of dataset programming is its validation. One of the common moments that programmers encounter is a mismatch due to the different number of records in the dataset. This can happen due to a large number of variables in supplementary datasets, the quality of the collected raw data, or other conditions.

This paper presents a macro that makes it possible to pinpoint all inconsistent values between the datasets on the Production and QC sides. It allows key discrepancies to be identified more quickly. Equally important, there is no limit to the number of inconsistency records displayed. The limit of displayed 500 values in PROC COMPARE sometimes makes it difficult to find the root cause of the discrepancy.

The main convenience and difference from the standard PROC COMPARE procedure is the compactness of the displayed result in the form of a table with both a specific parameter and a group of parameters.

INTRODUCTION

When updating raw data, most of the time can be taken up by the validation process. For the fastest detection of inconsistencies in the most compact form, you can use this macro. Basically, this macro is designed to work with SDTMs and ADaMs, but with certain parameter settings, it can also be used for ad-hoc requests.

MACRO DESCRIPTION

This macro "subj_check" is an addition to PROC COMPARE when there are two datasets with a different number of the observations, or the total number of observations is equal but different for some --TESTCD (on production and QC sides) and needs to find out for which specific subjects it is different. This macro is only able to detect differences in the number of observations per subject/parameter between a dataset on Production and on QC side. It does not look at e.g. timepoint or result values, and will thus not detect differences in these. The macro details the following:

- Number of observations for each subject.
- Showing subjects and number of observations if it's different.
- XYTESTCD variable for the Findings Observations Class (XY: the domain letters).
- Showing subjects and number of observations for each non-consistent XYTESTCD (XY: the domain letters).
- Difference in number of observations for each value of QNAM in Supplemental Qualifiers.
- Difference in number of observations for each subject with specific QNAM in Supplemental Qualifiers.

Only up to two variables actually play a role in the comparison:

- for SDTM only the variables USUBJID and XYTESTCD (XY: the domain letters),
- for ADaM (non-SUPP) only USUBJID and -if present- PARAMCD, and
- for SUPP only QNAM and USUBJID for specified QNAM values.

ADVANTAGES OVER PROC COMPARE

The main advantage is how information is presented. The compact table display irrelevant data what can help fast find inconsistencies number of observations for each subject and parameter. Using PROC COMPARE displays detailed information in a long list. When there is a different number of observations on the Production and QC sides, first, it's needed to remove this rude inconsistency. Also, an additional advantage is the ability to save results of comparing in datasets with subsequent use.

HOW THE MACRO WORKS

The macro “subj_check” contains three parameters.

```
%subj_check(domain, opt=, qnam=);
```

Two macro parameters are mandatory: dataset name (“domain”) and dataset type (“opt”). The third parameter (“qnam”) is optional and applies only to Supplemental Qualifiers dataset. For the first parameter “domain” in the upper case specify the name of SDTM (DM, LB, TR, etc.) or ADaM (ADSL, ADAE, etc.). For the second parameter “opt” specify the type of dataset with the next available values: “sdm”, “adam” or “supp”. For the third parameter specify values of QNAM in a quoted string. For several values separate them by space.

There are three blocks inside the macro. First, macro checks the type of dataset. If there “adam” type, it checks for the presence of the “PARAMCD” variable. Depending on this, the type of output may differ - by USUBJID or by USUBJID and PARAMCD. If there “sdm” type, it checks if the domain is included in the Findings Observations Class. Depending on this, the type of output may differ - by USUBJID or by USUBJID and --TESTCD. If there “supp” type, it checks if “qnam” parameter was specified or not. Depending on this, the type of output may differ - by USUBJID and QNAM if “qnam” parameter is specified, or by QNAM otherwise. The full code of the macro is listed in the Appendix.

EXAMPLES OF COMPARISON RESULTS

Data from the free-access CDISC-SDTM sample study [2] was used to demonstrate how the macro works.

DIFFERENCE IN NUMBER OF OBSERVATIONS FOR EACH SUBJECT

Let’s take the AE domain as an example. In the macro for the first parameter specify the “AE” domain, for the second parameter specify its type – “sdm”.

```
%subj_check(AE, opt=sdm);
```

With such specified parameters, the macro will display the result in the form of a table. Only subjects and the number of observations for each of them will be shown here from the production dataset AE and the QC dataset AE. For SDTMs that are not included in the Findings Observations Class, the table will always look like in Output 1. Accordingly, in this way you can also check ADaMs dataset. Write the name of ADaM in the parameters of the macro and specify “opt = adam” in the type.

USUBJID	QC_num	PROD_num
CDISC01.100008	22	12
CDISC01.100014	17	12
CDISC01.200001	22	12
CDISC01.200002	22	12

Output 1. Difference in number of observations for each subject

DIFFERENCE IN NUMBER OF OBSERVATIONS FOR --TESTCD FOR EACH SUBJECT

This is exactly the case for the Findings Observations Class SDTMs. For example, consider the LB domain on the production and QC sides. Specify the “LB” domain and “sdm” type in the macro’s parameters.

```
%subj_check(LB, opt=sdm);
```

Since this is a Findings Observations Class domain, the macro contains a check by --TESTCD. As a result of the macro execution, there is a table where the LBTESTCD, USUBJID, and number of observations from production and QC datasets are indicated. The results are shown in Output 2.

Similarly, also ADaM dataset ADLB, with variable PARAMCD, can be compared.

```
%subj_check(ADLB, opt=adam);
```

LBTESTCD	USUBJID	QC_test_num	PROD_test_num
BILI	CDISC01.100008	2	1
BILI	CDISC01.100014	2	1
BILI	CDISC01.200001	2	1
BILI	CDISC01.200002	2	1
BUN	CDISC01.100008	2	1
BUN	CDISC01.100014	2	1
BUN	CDISC01.200001	2	1
BUN	CDISC01.200002	2	1
GLUC	CDISC01.100008	6	3
GLUC	CDISC01.200001	6	3
GLUC	CDISC01.200002	6	3
HCT	CDISC01.100008	2	1
HCT	CDISC01.100014	2	1
HCT	CDISC01.200001	2	1
HCT	CDISC01.200002	2	1
HGB	CDISC01.100008	2	1
HGB	CDISC01.100014	2	1
HGB	CDISC01.200001	2	1
HGB	CDISC01.200002	2	1
LYM	CDISC01.100008	2	1
LYM	CDISC01.100014	2	1

Output 2. Difference in number of observations for LBTESTCD for each subject

DIFFERENCE IN NUMBER OF OBSERVATIONS FOR EACH SUBJECT WITH SPECIFIC QNAM IN SUPPLEMENTAL QUALIFIERS

For example, let's take SUPPAE on the production and QC sides. In the parameters of the macro specify the domain "AE", in the type specify the "supp" and in the third parameter specify which values of QNAM we are interested in. There can be one or several of them, separated by a space in quotes.

```
%subj_check (AE, opt=supp, qnam ="SUPPVAR1" "SUPPVAR4");
```

As a result of executing the macro, we will get a table where the number of observations from production and QC datasets is displayed for each subject for each specified value of QNAM. The results are shown in Output 3.

USUBJID	QNAM	QC_num	PROD_num
820101	SUPPVAR1	29	9
820101	SUPPVAR4	25	14
820104	SUPPVAR1	11	6
820104	SUPPVAR4	16	11
820105	SUPPVAR1	7	
820105	SUPPVAR4	3	
820106	SUPPVAR1	16	
820106	SUPPVAR4	17	
820107	SUPPVAR1	4	
820107	SUPPVAR4	5	
820201	SUPPVAR1	14	
820201	SUPPVAR4	1	

Output 3. Difference in number of observations for each subject with specific QNAM “SUPPVAR1” and “SUPPVAR4”

DIFFERENCE IN NUMBER OF OBSERVATIONS FOR EACH VALUE OF QNAM IN SUPPLEMENTAL QUALIFIERS

The same number of QNAM parameters means that production and QC datasets have the same number of unique QNAM values. Let's take the same SAPPAE as an example. Specify the “AE” domain and the “supp” type in the macro parameters.

```
%subj_check (AE, opt=supp);
```

As a result of executing the macro, we get a table with a list of the values of the QNAM variable and the number of observations for each value from the production dataset and QC dataset. This way, you can understand by which values there is a discrepancy. The results are shown in Output 4.

The different number of QNAM parameters is the opposite case of the previous one. Here, for example, in the SUPPAE dataset there are two “SUPP11” and “SUPP44” values, which are not present in the SUPPAE production dataset. Parameters of the macro are set in the same way. In the resulting table, a blank or 0 means that there are no such values of QNAM in the corresponding dataset. The results are shown in Output 5.

QNAM	QC_num	PROD_num
SUPPVAR1	208	37
SUPPVAR2	208	37
SUPPVAR4	112	34
SUPPVAR5	7	4
SUPPVAR6	221	36
SUPPVAR7	104	34
SUPPVAR8	135	32
SUPPVAR9	225	35

Output 4. Same number of QNAM values

QNAM	QC_num	PROD_num
SUPP11	112	
SUPP44	104	
SUPPVAR1	208	37
SUPPVAR2	208	37
SUPPVAR4	0	34
SUPPVAR5	7	4
SUPPVAR6	221	36
SUPPVAR7	0	34
SUPPVAR8	135	32
SUPPVAR9	225	35

Output 5. Different number of QNAM values

CONCLUSION

This paper presents a tool that simplifies the comparison of the number of observations per subject(/parameter) between two datasets, as are typically compared during validation. The limitations of this macro: it looks only at a few key variables and works only if the number of observations between compared datasets is different. The examples are shown different cases in which it can be used. Knowing these cases helps to identify and use the best one to fix discrepancies.

REFERENCES

1. Michael Stackhouse, Franklin E. Menius Jr., "An Automated Macro to Compare Data Transfers" - PharmaSUG 2018 - Paper AD-13.
2. CDISC Sample domains: <https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/51B6NK> .

CONTACT INFORMATION

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

Roman Chekhun

Quartesian

Email: roman.chekhun@quartesian.com

Web: <https://www.quartesian.com>

Brand and product names are trademarks of their respective companies.

APPENDIX

The whole source code of the macro is listed below.

```
ods html;

options minoperator mindelimiter=', ';

%macro subj_check(domain, opt=, qnam=);

%let domain = %upcase(&domain.);

%let adam_param = "";

%if &opt=adam %then %do;

    proc contents data=&domain. varnum out=varlist(keep=NAME) noprint;

    run;

    proc sql noprint;

        select name into: adam_param trimmed from varlist where name="PARAMCD";

    quit;

%put &adam_param;

%end;

%if &opt=sdtm and &domain in (LB, DA, DD, EG, IE, IS, MB, MS, MI, MO, PC, PP, PE, QS,
RP, SC, SS, TU, TR, RS, VS) %then %do;

    proc sql ;
```

```

create table &domain._compare as

select distinct coalesce(qc&&domain..&domain.testcd,h.PROD_testcd) as
&domain.TESTCD, coalesce(qc&&domain..usubjid,h.subject) as USUBJID,
count(qc&&domain..&domain.testcd) as QC_test_num , h.PROD_test_num

from qc&domain. full join (select distinct &&domain..&domain.testcd as
PROD_testcd, &domain..usubjid as subject, count(&&domain..&domain.testcd) as
PROD_test_num from &domain. group by &&domain..usubjid,
&&domain..&domain.testcd) h

on qc&&domain..&domain.testcd = h.PROD_testcd and qc&domain..usubjid=h.subject

group by qc&&domain..usubjid, qc&&domain..&domain.testcd;

select * from &domain._compare where QC_test_num^=PROD_test_num;

quit;

%end;

%else %if &opt=adam and &adam_param=PARAMCD %then %do;

proc sql ;

create table &domain._param_check as

select distinct coalesce(qc&&domain..paramcd,h.PROD_testcd) as PARAMCD,
coalesce(qc&domain..usubjid,h.subject) as USUBJID,
count(qc&&domain..paramcd) as QC_test_num , h.PROD_test_num

from qc&domain. full join (select distinct &&domain..paramcd as PROD_testcd,
&domain..usubjid as subject, count(&&domain..paramcd) as PROD_test_num

from &domain.

group by &&domain..usubjid, &&domain..paramcd) h

on qc&&domain..paramcd = h.PROD_testcd and qc&domain..usubjid=h.subject

group by qc&&domain..usubjid, qc&&domain..paramcd;

select * from &domain._param_check where QC_test_num^=PROD_test_num;

quit;

%end;

%else %if &opt=sdtm or (&opt=adam and &adam_param^=PARAMCD) %then %do;

proc sql;

create table &domain._subj_check as select distinct qc&domain..usubjid,
count(qc&&domain..usubjid) as QC_num, h.PROD_num

from qc&domain.

full join (select distinct &&domain..usubjid , count(&&domain..usubjid) as
PROD_num

from &domain. group by &&domain..usubjid) h

on qc&domain..usubjid = h.usubjid

```

```

        group by qc&&domain..usubjid;

select * from &domain._subj_check where QC_num^=PROD_num;

quit;

%end;

%if &opt=supp %then %do;

    %if %length(&qnam)=0 %then %do;

        proc sql;

            create table &domain._qnam_check as

                select distinct coalesce(a.qnam, h.qnam) as QNAM, count(a.qnam) as QC_num,
                h.PROD_num

                from suppqc&domain. a full join (select distinct (qnam), count(qnam) as
                PROD_num from supp&domain. group by qnam) h

                on a.qnam = h.qnam

                group by a.qnam;

            select * from &domain._qnam_check where QC_num^=PROD_num;

            quit;

        %end;

    %else %if %length(&qnam)^=0 %then %do;

        proc sql;

            create table &domain._subj_qnam_check as

                select distinct coalesce(a.usubjid, h.usubjid) as USUBJID,
                coalesce(a.qnam, h.qnam) as QNAM, count(qval) as QC_num, h.PROD_num

                from suppqc&domain. a full join

                (select usubjid, qnam, count(qval) as PROD_num from supp&domain. where
                qnam in (&qnam.) group by usubjid, qnam) h

                on a.usubjid = h.usubjid and a.qnam = h.qnam

                where a.qnam in (&qnam.)

                group by a.usubjid, a.qnam;

            select * from &domain._subj_qnam_check where QC_num^=PROD_num;

            quit;

        %end;

    %end;

%mend subj_check;

```