PharmaSUG 2023 - Paper QT-020

A Macro to Identify Repeating SAS® BY Variables in a MERGE

Timothy J. Harrington, Navitas Data Sciences, Princeton, NJ

ABSTRACT

This paper describes how, when performing a MERGE between two SAS® datasets, observations causing a message

'NOTE: MERGE statement has more than one data set with repeats of BY values'

are identified. This paper is intended for SAS programmers with a basic or higher level of expertise.

INTRODUCTION

SAS MERGEs are best performed on data sets with unique BY variable values in one or both of the data sets being merged, ie: one-to-one, one-to-many, or many-to-one merges. When there are occurrences of duplicate BY variables in both of the data sets (many-to-many) there is an ambiguity as to whether each repeating BY variable in one dataset should be matched to which of the duplicated key observations in the other. To indicate this situation a message ' MERGE statement has more than one data set with repeats of BY values' is written to the SASLOG file. Although this message shows a duplicated BY variable issue, it does not identify which observations are at fault. This macro performs a MERGE on two input data sets and if there are repeats of any of the BY variables, lists the observation numbers involved in both of the datasets, the values of the repeating BY variables, and the number of occurrences of different repeated BY variables to the SASLOG file.

EXAMPLE ILLUSTRATION

The following example is the MERGE of two Adverse Event (AE) SAS datasets, HEADACHE and FEVER, each of which include the PATIENT ID, VISITNUM, VISIT, and AE Severity Text, AE_TEXT1 for HEADACHE and AE_TEXT2 for FEVER.

Input Dataset 1: Headaches by Visit

Obs	PATIENT	VISITNUM	VISIT	AE_TEXT1
1	2497	11	BASELINE	Headache: Mild
2	6069	11	BASELINE	Headache: Mild
3	6720	11	BASELINE	Headache: Mild
4	6720	12	WEEK 2	Headache: Severe
5	6720	12	WEEK 2	Headache: Severe
6	6939	11	BASELINE	Headache: Moderate
7	8625	11	BASELINE	Headache: Mild
8	8625	11	BASELINE	Headache: Mild
9	8625	12	WEEK 2	Headache: Mild
10	8625	12	WEEK 2	Headache: Mild
11	8655	11	BASELINE	Headache: Moderate
12	8855	12	WEEK 2	Headache: Mild
13	8855	12	WEEK 2	Headache: Mild
14	8855	12	WEEK 2	Headache: Moderate
15	8855	12	WEEK 2	Headache: Moderate
16	8855	12	WEEK 2	Headache: Mild

Input Dataset 2: Incidence of Fever by Visit

Obs	PATIENT	VISITNUM	VISIT	AE_TEXT2
1	2497	11	BASELINE	Fever: 100F
2	2497	11	BASELINE	Fever: 100F
3	2497	11	BASELINE	Fever: 103F
4	2497	12	WEEK 2	Fever: 100F
5	2497	12	WEEK 2	Fever: 99F
6	2497	12	WEEK 2	Fever: 99F
7	2497	12	WEEK 2	Fever: 100F
8	2497	13	WEEK 3	Fever: 102F
9	2497	13	WEEK 3	Fever: 103F
10	2497	14	WEEK 4	Fever: 99F
11	6069	11	BASELINE	Fever: 100F
12	6069	11	BASELINE	Fever: 99F
13	6069	11	BASELINE	Fever: 102F
14	6069	11	BASELINE	Fever: 102F
15	6069	12	WEEK 2	Fever: 103F
16	6069	13	WEEK 3	Fever: 105F
17	6069	14	WEEK 4	Fever: 99F
18	6069	14	WEEK 4	Fever: 99F
19	6720	11	BASELINE	Fever: 99F
20	6720	11	BASELINE	Fever: 100F
21	6720	11	BASELINE	Fever: 99F
22	6720	12	WEEK 2	Fever: 99F
23	6720	12	WEEK 2	Fever: 99F
24	6720	13	WEEK 3	Fever: 103F
25	6720	14	WEEK 4	Fever: 100F
26	6939	11	BASELINE	Fever: 99F
27	6939	12	WEEK 2	Fever: 99F
28	6939	13	WEEK 3	Fever: 99F
29	6939	14	WEEK 4	Fever: 100F
30	6939	14	WEEK 4	Fever: 99F
31	6939	14	WEEK 4	Fever: 99F
32	6939	14	WEEK 4	Fever: 101F
33	6939	15	WEEK 5	Fever: 101F
34	6939	15	WEEK 5	Fever: 99F
35	6939	15	WEEK 5	Fever: 105F
36	8625	11	BASELINE	Fever: 104F
37	8625	11	BASELINE	Fever: 99F
38	8625	11	BASELINE	Fever: 99F
39	8625	12	WEEK 2	Fever: 100F
40	8625	13	WEEK 3	Fever: 103F
41	8625	14	WEEK 4	Fever: 99F
42	8855	11	BASELINE	Fever: 103F
43	8855	12	WEEK 2	Fever: 102F

When these two data sets are merged using the following SAS code the resulting dataset COMBINED, listed below, is created. Observations are kept when the IN numeric variables H and F are both 1, that is when there are matching BY variables in both of the datasets (Inner join).

```
proc sort data=headache;
  by patient visitnum;
run;

proc sort data=fever;
  by patient visitnum;
run;

data combined;
  merge headache(in=h) fever(in=f);
  by patient visitnum;
  if h and f;
run;
```

Obs	PATIENT	VISITNUM	VISIT	AE_TEXT1	AE_TEXT2
1	2497	11	BASELINE	Headache: Mild	Fever: 100F
2	2497	11	BASELINE	Headache: Mild	Fever: 100F
3	2497	11	BASELINE	Headache: Mild	Fever: 103F
4	6069	11	BASELINE	Headache: Mild	Fever: 100F
5	6069	11	BASELINE	Headache: Mild	Fever: 99F
6	6069	11	BASELINE	Headache: Mild	Fever: 102F
7	6069	11	BASELINE	Headache: Mild	Fever: 102F
8	6720	11	BASELINE	Headache: Mild	Fever: 99F
9	6720	11	BASELINE	Headache: Mild	Fever: 100F
10	6720	11	BASELINE	Headache: Mild	Fever: 99F
11	6720	12	WEEK 2	Headache: Severe	Fever: 99F
12	6720	12	WEEK 2	Headache: Severe	Fever: 99F
13	6939	11	BASELINE	Headache: Moderate	Fever: 99F
14	8625	11	BASELINE	Headache: Mild	Fever: 104F
15	8625	11	BASELINE	Headache: Mild	Fever: 99F
16	8625	11	BASELINE	Headache: Mild	Fever: 99F
17	8625	12	WEEK 2	Headache: Mild	Fever: 100F
18	8625	12	WEEK 2	Headache: Mild	Fever: 100F
19	8855	11	BASELINE	Headache: Moderate	Fever: 103F
20	8855	12	WEEK 2	Headache: Mild	Fever: 102F
21	8855	12	WEEK 2	Headache: Mild	Fever: 102F
22	8855	12	WEEK 2	Headache: Moderate	Fever: 102F
23	8855	12	WEEK 2	Headache: Moderate	Fever: 102F
24	8855	12	WEEK 2	Headache: Mild	Fever: 102F

There are 24 observations where PATIENT and VISITNUM match, but there are two cases where there are multiple occurrences of the same PATIENT number and VISITNUM in both of the two datasets. The resulting output to the SASLOG file is:

```
NOTE: MERGE statement has more than one data set with repeats of BY values.
NOTE: There were 16 observations read from the data set WORK.HEADACHE.
NOTE: There were 43 observations read from the data set WORK.FEVER.
NOTE: The data set WORK.COMBINED has 24 observations and 5 variables.
```

In this example, the reason why there is the 'Repeats of BY values' message is because PATIENT 6720 has two VISITNUM=12 (WEEK 2) visits in each of HEADACHE (observations 4 and 5) and FEVER

(observations 22 and 23). There is also PATIENT 8625 with two VISITNUM=11 (BASELINE) visits in HEADACHE (observations 7 and 8) and three VISITNUM=11 visits in FEVER (observations 36,37, and 38). Multiples of the same BY groups in both input datasets will generate this message. The same is true when there are three or more datasets being MERGEd and any two of the datasets have the repeating BY variable values. This message is shown only once regardless of how many sets of repeating BY values there are, there is no further information given, hence finding each of the observations with the repeating BY variable values can be a difficult task, particularly when the dataset has a very large number of observations.

Whenever there are multiple matches of the same BY variable values in two (or more) of the input datasets there is an ambiguity as to which of the repeating value matches may be required to be output. If PROC SQL were used there would be an output observation created for every combination, the Cartesian product, of the repeating BY values. For example, if there were two repeating BY values in one dataset and two in the other there would be four output observations created.

```
proc sql noprint;
   create table combined
     as select a.patient, a.visitnum, a.visit, a.ae_text1, b.ae_text2
       from headache a inner join fever b
         on a.patient=b.patient and a.visitnum=b.visitnum
           order by patient, visitnum;
quit;
```

Using the above HEADACHE and FEVER observations with this PROC SQL code results in 2x2 and 2x3 output observations in COMBINED, these are the Cartesian products, with each FEVER observation appearing with each BY variable matching HEADACHE observation. The OBS column shows the source observation numbers in HEADACHE and FEVER respectively.

Obs	PATIENT	VISITNUM	VISIT	AE_TEXT1	AE_TEXT2
4 / 22	6720	12	WEEK 2	Headache: Severe	Fever: 99F
4 / 23	6720	12	WEEK 2	Headache: Severe	Fever: 99F
5 / 22	6720	12	WEEK 2	Headache: Severe	Fever: 99F
5 / 23	6720	12	WEEK 2	Headache: Severe	Fever: 99F

Obs	PATIENT	VISITNUM	VISIT	AE_TEXT1	AE_TEXT2
7 / 36	8625	11	BASELINE	Headache: Mild	Fever: 99F
7 / 37	8625	11	BASELINE	Headache: Mild	Fever: 99F
7 / 38	8625	11	BASELINE	Headache: Mild	Fever: 104F
8 / 36	8625	11	BASELINE	Headache: Mild	Fever: 99F
8 / 37	8625	11	BASELINE	Headache: Mild	Fever: 99F
8 / 38	8625	11	BASELINE	Headache: Mild	Fever: 104F

When a DATA step MERGE is used, with the same columns with the same type of the join, in this case an inner join, only part of the Cartesian product is output and there is an ambiguity as to which of the observations in the Cartesian product are output. This happens to be ordered by the observation sequence in FEVER for only the first of the repeating BY groups in HEADACHE. AE_TEXT2='Fever: 104F' is the first of the three observations for PATIENT 8625 in FEVER at VISITNUM=11 VISIT='WEEK 2'.

Obs	PATIENT	VISITNUM	VISIT	AE_TEXT1	AE_TEXT2
4 or 5 / 22 or 23 ?	6720	12	WEEK 2	Headache:	Fever: 99F
4 or 5 / 22 or 23 ?	6720	12	WEEK 2	Headache:	Fever: 99F

Obs	PATIENT	VISITNUM	VISIT	AE_TEXT1	AE_TEXT2
7 or 8 / 36, 37, or 38 ?	8625	11	BASELINE	Headache: Mild	Fever: 104F
7 or 8 / 36, 37, or 38 ?	8625	11	BASELINE	Headache: Mild	Fever: 99F
7 or 8 / 36, 37, or 38 ?	8625	11	BASELINE	Headache: Mild	Fever: 99F

A way of foreseeing this 'many-to-many' situation occurring is to pre-check each of the datasets being merged before the MERGE takes place, looking for multiple occurrences of BY variable values and reporting where such occurrences occur in both datasets.

EXAMPLE SOLUTION

The following macro pre-checks two datasets to be merged for repeating BY variable values occurring in both datasets and reports on any that are found. The macro takes the following input parameters:

```
ds1= Name (and libname) of the first input dataset (dataset a)
ds2= Name (and libname) of the second input dataset (dataset b)
join= Type of join (e.g. 'a and b' for inner join, this is the default),
outds= Name (and libname) of the created output dataset
byvars= BY variables to be sorted and merged on
reportn= Maximum number of repeat BY variable groups to report (default=10)
```

The macro performs the following:

1. Sorts the first input dataset by the BY variables
2. Records the observation number _n_ and a running count of each distinct BY values occurrence
3. Adds an input dataset identifier
4. Creates an output dataset of each of the last observation of distinct BY variable values where the occurrence count is >= 2
5. Repeats steps 1 to 4 for the second input dataset
6. Creates a third dataset by setting the two created datasets with these output observations together
7. Sorts this combined dataset by the BY variables and the source dataset identifier (input dataset A=1 or dataset B=2)
8. Creates a fourth dataset of the observations where the same BY variable values occur with both input dataset identifiers (1 or 2), this is where there are two observations with the same BY variable values, one from input dataset 1 and the other from input dataset 2, and *both* have a count of >= 2.
9. Performs the MERGE with the specified join on the original input datasets
10. Uses the fourth output dataset to report each occurrence (up to the REPORT parameter maximum) of repeating BY variables, including the source datasets and the observation number of the first repeat in both of the datasets, as well as the number of repeats in each dataset and the names and values of the repeating BY variables.

MACRO CODE

```
%macro mergemny(ds1= , ds2= , join= , outds= , byvars= , reportn=10);

%local i nbyvars;

%if &join= %then %do; /* Default is inner join */
  %let join=a and b;
%end;

%do i=1 %to 2;

 proc sort data=&&ds&i;
   by &byvars;
 run;

 data t&i(keep=&byvars dsid obsn repn);
   set &&ds&i;
   by &byvars;
   retain repn obsn 0; /* Repeat count, _n_ at first distinct BY values */
   dsid=&i; /* Source dataset identifier 1 or 2 */
   if first.%scan(&byvars,-1,' ') then do;
     obsn=_n_; /* Observation number of the first new BY variable values */
   end;
   repn=repn*(first.%scan(&byvars,-1,' ')=0)+1; /* BY values repeat count*/
   if last.%scan(&byvars,-1,' ')=1 and repn ge 2; /* Count at last repeat*/
run;

%end;

 data t3;
   set t1 t2;
 run;

 proc sort data=t3;
   by &byvars dsid;
 run;

 data t4;
   set t3;
   by &byvars dsid; /* Only keep multiple counts in both datasets */
   if first.%scan(&byvars,-1,' ')+last.%scan(&byvars,-1,' ') ne 2;
 run;

 data _null_; /* Get the number of BY variable names */
   call symput('nbyvars',put(countw("&byvars",' '),8.));
 run;

 data &outds; /* Perform the MERGE using numeric a and b as IN variables */
   merge &ds1(in=a) &ds2(in=b);
   by &byvars;
   if &join;
 run;
```

```
 data _null_; /* Report on the duplicated BY variables found, if any */
   set t4 end=eof;
   by &byvars dsid;
   retain nocc 0;       /* Occurrence of a group of BY variable repeats */
   length dsname $100;  /* Source dataset name */
   if _n_ ge 1 and nocc<&reportn then do;
     dsname=left(upcase(scan("&ds1!&ds2",dsid,'!')));
     repn=repn-1; /* Number of repeats, 2 sets of same BY values=1 repeat*/
     obsn=obsn+1; /* Observation number of the first repeat */
     put 'NOTE: Dataset ' dsname 'has ' repn 'duplicates at _N_=' obsn ': '
       @ ;
     if dsid=1 then do;
%do i=1 %to &nbyvars; /* List the BY variables and their values */
       put "%upcase(%scan(&byvars,&i,' '))=" %scan(&byvars,&i,' ') @ ;
%end;;
     end;
     put;
     if dsid=2 then do;
       put;
     end;
   end;
   nocc=nocc+last.%scan(&byvars,-1,' ');
   if eof=1 then do;
     put "NOTE: Number of occurrences of duplicated BY variables in
      %upcase(&ds1) + %upcase(&ds2) merge is " nocc '.';
     put;
   end;
 run;

 proc datasets nolist;
   delete t1-t4;
 quit;
 run;

%mend mergemny;
```

MACRO OUTPUT

The output generated and written to the SASLOG file, using the HEADACHE and FEVER datasets as input, is listed below. The macro call for this example is:

```
%mergemny(ds1=headache,
          ds2=fever,
          outds=combined,
          byvars=patient visitnum);
```

```
NOTE: MERGE statement has more than one data set with repeats of BY values.
NOTE: There were 16 observations read from the data set WORK.HEADACHE.
NOTE: There were 43 observations read from the data set WORK.FEVER.
NOTE: The data set WORK.COMBINED has 24 observations and 5 variables.

NOTE: Compressing data set WORK.COMBINED increased size by 100.00 percent.
      Compressed is 2 pages; un-compressed would require 1 pages.
NOTE: DATA statement used (Total process time):
      real time           0.00 seconds
      cpu time            0.00 seconds

NOTE: Dataset HEADACHE has 1 duplicates at _N_=5 : PATIENT=6720 VISITNUM=12
NOTE: Dataset FEVER has 1 duplicates at _N_=23

NOTE: Dataset HEADACHE has 1 duplicates at _N_=8 : PATIENT=8625 VISITNUM=11
NOTE: Dataset FEVER has 2 duplicates at _N_=37

NOTE: Number of occurrences of duplicated BY variables in HEADACHE + FEVER
merge is 2.
```

FURTHER CONSIDERATIONS

When using this macro the following points should be taken into account:

- The user must have read access to both of the input datasets and write access to the libname of the output dataset
- The two input datasets are unchanged after completion of the macro execution
- The JOIN parameter, if user specified, must use the variables A and B (the IN variables). Variables named ‘a’ or ‘b’ must not be present in either of the input datasets
- Datasets with names ‘t1’, ‘t2’, ‘t3’, and ‘t4’ should not be used by the calling program or other previously called macros. These temporary datasets are deleted by the macro just before its completion
- Both input datasets are read and checked for duplicates prior to the MERGE. If these datasets are very large there will be a noticeable increase in execution time
- If there are no repeats of the same BY variables’ values in both of the datasets nothing extra is written to the SASLOG.

CONCLUSION

This macro is a suggested solution for identifying repeating key values in both of two datasets. It is probably most helpful when handling two or more datasets with very large numbers of observations, where incidences of repeating BY variables are difficult to find, or for routinely processing data where observations with duplicated key values in two datasets are not expected.

SUGGESTED FURTHER READING

Kuligowski, Andrew T. September 2008. The Nielsen Company. “HELP! - My MERGE Statement Has More Than One Data Set with Repeats of BY Values!” Northeast SAS Users Group (NESUG) Conference, Pittsburgh, PA : Lex Jansen. https://www.lexjansen.com/cgi-bin/xsl_transform.php?x=nesug2008

SAS Institute Inc. Cary, NC. (c) 2001, 2007, 2008, 2012. "Step-by-Step Programming with Base SAS® Software." ISBN 978-1-58025-791-6. Chapter 18: "Merging SAS Data Sets". Available at https://support.sas.com/documentation/cdl/en/basess/58133/PDF/default/basess.pdf

ACKNOWLEDGMENTS

Navitas Data Sciences, 1610 Medical Drive, Suite 300, Pottstown, PA 19464 USA

PharmaSUG 2022 Paper Selection Committee

CONTACT INFORMATION

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

Timothy J Harrington
Navitas Data Sciences
269 501 6093
Timothy.harrington@navitaslifesciences.com
www.navitasdatasciences.com