Paper 85-27

Stop Madly Merging: Proc Print to the Rescue!

Catherine Lindsey, TRW, Atlanta, GA

ABSTRACT

Merging multiple datasets correctly can be a formidable challenge to an inexperienced SAS/BASE user. How will you determine if the merge was successful? Do all the records match up as was intended? One of the most useful, underutilized tools in SAS to help answer these and other merging questions is proc print. Proc print can be most beneficial when multiple datasets need to be merged, such as for a large study or project with several linked forms, each recorded in a separate dataset. By creating an indicator variable (variables of the same value given to every observation in each specific dataset) for each form to be merged, proc print can be used to uncover incomplete merges. In some cases, what may first seem to be a merging error may not be an error at all. If, for example, one of the original forms (before merging) was missing all information for one or more observations, then the indicator variable corresponding to that form will be missing after the merge. By running a proc print, specifying only those observations where at least one indicator variable is missing, merging gaps and missing data can be easily identified and checked against the original form dataset.

INTRODUCTION

When large amounts of data are collected, it is often more practical to create several linked datasets rather than a single large dataset. These smaller datasets require less memory, are far more manageable, and are linked together with common unique identifier variables. Creating a single workable dataset with only essential information from these multiple datasets can be simple and painless by doing a few quick checks using proc print.

For the following demonstration, imagine this scenario: A longitudinal study has been conducted to assess risk factors for lung disease. For the purpose of this paper, only variables collected at baseline will be considered. Dataset SCREENER contains screening information used to determine if the subject was eligible for the study (example variables: AGE, GENDER, and INCOME). Dataset BXLINE contains the study baseline questionnaire which collected information about health practices, risk behaviors, and lifestyle (SMOKE, DRVISIT, and EXERCISE). After completing the baseline questionnaire, subjects were seen by a physician for a routine physical exam. Information collected during this exam is stored in Dataset EXAM (LUNGS, PULSE, and BP). At each of four study sites, subjects were assigned a unique identification variable; to correctly identify a certain subject, both the site and identification variables are needed. Both of these variables are included in every dataset (SITE, ID).

A dataset containing variables from all three datasets for men aged 40-50 is needed for analysis. The first step, before beginning to actually create this dataset, is to become very familiar with each dataset separately. A proc contents should be run, and frequencies should be obtained for different important variables to check for missing or unusual variables. Has the data been cleaned? If so, outliers and questionable values should be rectified. In this example, the datasets SCREENER, BXLINE, and EXAM have been cleaned, but it is still important to look at variable frequencies and note for which variables any missing values occur (there are no missing values for this example).

After the initial dataset scrutiny and familiarization, the process of creating the analytical dataset can begin. For this dataset, the investigator only needs certain variables (all variables listed in the description above, a total of nine). Each dataset needs to be prepared individually.

APPLICATION AND PROGRAMMING

Starting with the SCREENER dataset, AGE, GENDER, and INCOME are kept by using the *keep* option that appears after the data statement. Men not aged 40-50 are deleted from the dataset (now to be called ONE, in a temporary dataset creation), and an indicator variable is created: SCREENER=1. Each observation remaining in dataset ONE is given a variable called SCREENER and for all observations, the value of SCREENER is equal to one. The libname statement is necessary to define where permanent datasets are stored. Notice that the KEEP statement includes the identification variables (SITE, ID) and the newly created indicator variable (SCREENER) in addition to the three requested variables that were selected from the SCREENER dataset.

```
libname lung "a:\";
data one (keep=site id screener age gender
         income);
set lung.screener;
if (age<40 or age>50) then DELETE;
screener=1;
proc sort;
by site id;
run;
```

Dataset ONE, containing information from SCREENER is now ready to be merged. Similar programming is prepared for both the BXLINE and EXAM datasets below. Note the creation of indicator variables with names indicating the original dataset where the data is located.

```
data two (keep=site id bxline smoke drvisit
         exercise);
set lung.bxline;
bxline=1;
proc sort;
by site id;
run;

data three (keep=site id exam lungs pulse bp);
set lung.exam;
exam=1;
proc sort;
by site id;
run;
```

Datasets TWO and THREE are now also ready to merge, each with their respective indicator variables. Following is merge programming for the creation of Dataset FOUR. Dataset ONE, containing the SCREENER information is being selected and forced into FOUR due to the age cut specification. When the merge occurs only the subjects matching those already determined to be between ages 40 and 50 and having screener information will be included in FOUR. The *(in=one)* option in the

merge statement and the *if one;* statement instruct SAS to do just this.

```
data four;
merge one (in=one) two three;
if one;
by site id;
run;
```

Notice that the merge-by variables are the variables required to identify a subject and the variables by which all three datasets were sorted. Dataset FOUR has now been created, containing all nine requested variables (AGE, DRVISIT, etc.), two identification variables (SITE, ID), and three indicator variables (SCREENER, BXLINE, EXAM).

Now, checks must be performed to ensure the data merged properly. The first step is to use proc print and the indicator variables to make sure that all subjects have information from all datasets. The easiest way to do this is to use a request asking for only those observations where one of the indicator variables is not equal to one to be printed. Only the indicator variables and the identification variables will be printed out in the output. Note that due to our inclusion of Dataset ONE (above), all observations should have SCREENER=1.

```
proc print data=four;
where screener ne 1 or bxline ne 1 or exam ne
     1;
var site id screener bxline exam;
run;
```

After running this code, the log should be checked to see how many observations were printed out. This number is the number of subjects (observations) that do not have data from all three forms. Then, look at the output. Example output:

SITE	ID	SCREENER	BXLINE	EXAM
2	32	1	1	.
2	54	1	1	.
4	22	1	.	1

Looking at the output, two subjects are missing the EXAM data and one is missing the BXLINE data. To see if the data exists in the corresponding original dataset and determine if the data is truly missing or a merging error occurred, another proc print can be run.

```
proc print data=LUNG.EXAM;
where (site=2 and id=32) or (site=2 and
     id=54);
run;

proc print data=LUNG.BXLINE;
where (site=4 and id=22);
run;
```

A quick glance at the log or output will show if any records were identified. If not, then these records were most likely merged correctly, and the subject does not have information. It is important to match up these missing records with missing variable values. For example, in this case, two subjects did not have EXAM data. Therefore, there should be at least 2 missing values for the variables LUNG, PULSE, and BP (in addition to the indicator variable EXAM).

After running these proc print checks and uncovering any discrepancies, dataset FOUR is ready for final checks (missing values, skip patterns, etc) and analysis.

CONCLUSION

Happy merging!

CONTACT INFORMATION

Any questions or comments may be addressed to:

Catherine Lindsey
CDC (TRW/CISSS)
1600 Clifton Road, NE
Mailstop E46
Atlanta, GA 30333

(w) 404-639-3279
(f) 404-639-8640
col8@cdc.gov