Paper 10600-2016

You Can Bet On It, The Missing Rows are Preserved with PRELOADFMT and COMPLETETYPES

Christopher J. Boniface, U.S. Census Bureau;
Janet L. Wysocki, U.S. Census Bureau

ABSTRACT

Do you write reports that sometimes have missing categories across all class variables? Most times the customer or sponsor wants to see those missing categories in the report. Some programmers will write all sorts of additional data step code in order to show the zeroes for the missing rows or columns. Did you ever ponder that there must be an easier way to accomplish this? Well, PROC MEANS and PROC TABULATE in conjunction with PROC FORMAT can handle this situation with a couple of powerful options. With PROC TABULATE, we can use the PRELOADFMT and PRINTMISS options in conjunction with a user-defined format with PROC FORMAT to accomplish this task. With PROC SUMMARY, we can use the COMPLETETYPES option to get all the rows with zeroes.

The Census Bureau produces special tabulations for many sponsors. Often the sponsor will want a report with various counts across various categorical variables. Sometimes the crossing of certain class variables results in no observations. However, the sponsor wants to see these missing rows or columns in the table. By default, PROC TABULATE and PROC MEANS will omit these missing categories from the result. This paper will show two easy examples of how to get those missing rows or columns in the table. We'll present a special tabulation where the sponsor wants to see occupation code by county within each state. However, not all occupations are filled in each state resulting in missing rows. To solve this, we'll show one example using PROC TABULATE with the PRELOADFMT option in tandem with a user-defined format created in PROC FORMAT. Secondly, we'll show another solution using PROC SUMMARY with COMPLETETYPES to secure the missing categories. The final result is that all the state tables will have the same number of rows and all of the occupations listed.

INTRODUCTION

The U.S. Census Bureau's 2010 Census Special Tabulation Program provides data users with the option to have user-defined tabulations created from decennial census microdata on a cost-reimbursable basis. When requesting a special tabulation, the sponsor should provide a preliminary, general specification of the data needed. We will ask them some specific questions, and then work with them to develop a final, detailed specification that documents their data needs and geographic requirements. For additional information on the U.S. Census Bureau's Special Tabulation Program, see http://www.census.gov/population/www/cen2010/spec-tab/.

We use SAS® to tabulate our decennial special tabulations. In general, we use PROC TABULATE or PROC SUMMARY to generate our special tabulations report. Many sponsors of custom special tabulations request crosstabs of various categorical variables. Sometimes the crossing of certain class variables results in no observations. However, the sponsor wants to see these missing rows or columns in the table. This paper will explore two solutions to this problem. One solution will use PROC TABULATE with the PRELOADFMT option in tandem with a user-defined format created in PROC FORMAT. The second solution will use PROC SUMMARY with the COMPLETETYPES option to preserve the missing categories.

Sample Case Study

A sponsor has requested a decennial special tabulation from the Census Bureau. They want to see counts of all occupations by county for all states for a particular Decennial Census. They want one file per state. They indicate that they want an Excel table that displays all occupations in the rows of the table and all counties going across as the column. Additionally, they request that all occupations be shown in the table even if no one held a particular occupation in a particular state. Thus, they want to see the same number of rows in each state table. That is, they

want to see all occupations in each table.[1]

SOLUTION 1: PROC TABULATE/PRELOADFMT/PRINTMISS

PROC TABULATE is used in our first solution. PROC TABULATE is a powerful tool in doing tabulations and is used quite a bit here to not only compute the tabulation counts, but to output a report with rows and columns. A basic table statement with two class variables in it, separated by a comma, will produce a table with rows and columns. The PROC TABULATE code below essentially produces the counts and outputs a report with rows and columns. The class statements list the two categorical variables in our study, occupation and state county codes. The occupation values will appear in the row dimension of the table, since it is listed before the comma in the table statement. The state county values will appear in the column dimension of the table, since it is listed after the comma in the table statement. Note also that we have a PROC FORMAT for all the many occupation codes. There are hundreds of codes, but for display purposes, we're just showing six of them. The output of the PROC TABULATE below is shown in Table1. The output in Table 1 looks fine and good, until we take a closer look. Where are the Pumping station operators, Shuttle car operators, and Military officer occupations in the table? They are not there because no one held these jobs in any county in the state. By default, PROC TABULATE will output only the values of the categories that have at least one occurrence in the data. The missing categories/rows are deleted by default. We create an output SAS dataset called sums, and any crossing that does not exist in the data will not output to the SAS dataset. Thus, Occupation Code 975 for County A displays as a missing value by default, since there are no occurrences for this crossing. Note: the MLF option on the class statement in tandem with the FORMAT statement allows the labels for the occupation codes to show in the Occupation column. Without either, the actual codes (973,974) would show instead of the labels in the Occupation column.

```
/*Job Category Titles with Census 2000 Codes – partial listing for display*/
proc format;
    value $occf (notsorted multilabel)
     '965' = 'Pumping station operators'
     '972' = 'Refuse and recyclable material collectors'
     '973' = 'Shuttle car operators'
     '974' = 'Tank car, truck, and ship loaders'
     '975' = 'Material moving workers, all other'
     '980' = 'Military officer and special tactical operations leaders/managers';
run;

proc tabulate data=recodes out=sums;
 class occ / order=data mlf;
 class stcou / order=data mlf ;
 var count;
 table occ, stcou * (count*sum);
 weight pwt;
 format occ $occf. ;
run;
```

Reported Occupation Counts by County

Data: Counties in State X. Data based on a sample

		County A	County B	County C	County D
Occupation	**Occupation Code**				
Refuse and recyclable material collectors	972	80	95	130	40
Tank car, truck, and ship loaders	974	15	20	30	25
Material moving workers, all other	975	.	10	45	65

[1] All population counts displayed are fictitious

Table 1. Table with the missing rows not showing

How can we get the missing row(s) to appear? There are two powerful options in PROC TABULATE that will solve this problem: PRELOADFMT and PRINTMISS. The code below shows the solution. Essentially, you need to use these two options in tandem. The PRELOADFMT needs to be an option on the class statement of your categorical values. Moreover, you need to specify a format in the format statement. Also, you need to specify the (NOTSORTED and MULTILABEL) options in the PROC FORMAT for the $occf. format. Lastly and most important, you need the PRINTMISS option on the TABLE statement. Effectively, you are telling SAS to output all of the occupation codes in the occupation format ($occf.) regardless of whether there is an occurrence or not in the data. Thus, all occupations codes will appear in the rows of the table. Furthermore, by specifying the PRINTMISS option, any missing values in a particular cell of the table will be output to the sums dataset and will display as a zero instead of a missing value in the table.

Table 2 shows the output of the following PROC TABULATE. This time, the Pumping station operators, Shuttle car operators, and Military officers appear as rows in the table. Note that all values for these occupations are zero. Also, other cells that previously showed a missing value have changed to a zero.

```
proc tabulate data=recodes out=sums;
 class occ / order=data preloadfmt mlf;
 class stcou / order=data preloadfmt mlf ;
 var count;
 table occ, stcou * (count*sum)  /printmiss;
 weight pwt;
  format occ $occf. ;
 run;
```

Reported Occupation Counts by County

Data: Counties in State X. Data based on a sample

		County A	County B	County C	County D
Occupation	Occupation Code				
Pumping station operators	965	0	0	0	0
Refuse and recyclable material collectors	972	80	95	130	40
Shuttle car operators	973	0	0	0	0
Tank car, truck, and ship loaders	974	15	20	30	25
Material moving workers, all other	975	0	10	45	65
Military officer and special tactical operations leaders/managers	980	0	0	0	0

Table 2. Table with the missing rows showing

SOLUTION 2: PROC SUMMARY/COMPLETETYPES

PROC SUMMARY is used in our second solution. As with PROC TABULATE, PROC SUMMARY will not output missing categories of the class variables involved in our crosstab of occupation with county. The basic PROC SUMMARY code is shown below.

```
proc summary data=recodes nway;
 class occ county;
 var count;
 output out=outrecodes1 sum=;
run;
```

Take notice the NWAY option on the PROC SUMMARY line. This option will allow the highest level of tabulation. For our example, this will show observations at the county for each occupation. The CLASS statement lists the variables occ and county. The VAR statement sums the count variable. The OUTPUT statement outputs a summed dataset name outrecodes using the option sum=. Table 3 shows the PROC SUMMARY output and the fact that the Shuttle car operators, Pumping station operators, and Military officer occupations are missing from the output.

Reported Occupation Counts by County

Data: Counties in State X. Data based on a sample

	Occupation Code	County	Count
Refuse and recyclable material collectors	972	A	80
Refuse and recyclable material collectors	972	B	95
Refuse and recyclable material collectors	972	C	130
Refuse and recyclable material collectors	972	D	40
Tank car, truck, and ship loaders	974	A	15
Tank car, truck, and ship loaders	974	B	20
Tank car, truck, and ship loaders	974	C	30
Tank car, truck, and ship loaders	974	D	25
Material moving workers, all other	975	B	10
Material moving workers, all other	975	C	45
Material moving workers, all other	975	D	65

Table 3. Table with the missing rows not showing

PROC SUMMARY has its own solution for this problem and it is with the option COMPLETETYPES. Similar to the PRELOADFMT and PRINTMISS options with PROC TABULATE, the COMPLETETYPES option will output all values of a categorical variable even if there are no values for a particular crossing. When you use the COMPLETETYPES option on the PROC SUMMARY statement, all combinations of the class variables will appear in the output. In this case, all combinations of the crossings for occupation and county will appear in the output. Using the option MISSING = 0 will zero fill the missing observations. The limitation of using this procedure however, is that at least one observation for a particular occupation code needs to exist within the dataset. Thus, the following code will solve part of the problem, but not all of it. Adding the COMPLETETYPES option will add a row for occupation code 975 county A, since there is at least one observation already for occupation code 975. However, we still don't have any zero filled observations for occupation codes 965, 973 and 980. How can we get those rows in the table?

The PROC SUMMARY CODE below using COMPLETETYPES will solve part of the problem. Table 4 shows a row for occupation code 975, county A.

```
options missing = 0;
proc summary data=recodes nway completetypes;
 class occ county;
 var count;
 output out=outrecodes2 sum=;
run;
```

Reported Occupation Counts by County

Data: Counties in State X. Data based on a sample

	Occupation Code	County	Count
Refuse and recyclable material collectors	972	A	80
Refuse and recyclable material collectors	972	B	95
Refuse and recyclable material collectors	972	C	130
Refuse and recyclable material collectors	972	D	40
Tank car, truck, and ship loaders	974	A	15
Tank car, truck, and ship loaders	974	B	20
Tank car, truck, and ship loaders	974	C	30
Tank car, truck, and ship loaders	974	D	25
Material moving workers, all other	975	A	0
Material moving workers, all other	975	B	10
Material moving workers, all other	975	C	45
Material moving workers, all other	975	D	65

Table 4. Table with some of the missing rows showing

To show the occupation codes for those occupations where no counts exist in any of the counties, we need to use PRELOADFMT along with COMPLETETYPES. Just like the example with PROC TABULATE used in Solution 1, we need to set the stage and tell SAS the viable occupation codes using PRELOADFMT. We also need to specify a format for the occ variable on the FORMAT statement.

```
/*Job Category Titles with Census 2000 Codes - partial listing for display*/
proc format;
    value $occf (notsorted multilabel)
     '965' = 'Pumping station operators'
     '972' = 'Refuse and recyclable material collectors'
     '973' = 'Shuttle car operators'
     '974' = 'Tank car, truck, and ship loaders'
     '975' = 'Material moving workers, all other'
     '980' = 'Military;
run;
```

```
options missing = 0;
proc summary data=recodes nway completetypes;
 class occ county /preloadfmt;
 var count;
 output out=outrecodes2 sum=;
format occ $occf.
run;
```

Reported Occupation Counts by County

Data: Counties in State X. Data based on a sample

Occupation	Occupation Code	County	Count
Pumping station operators	965	A	0
Pumping station operators	965	B	0
Pumping station operators	965	C	0
Pumping station operators	965	D	0
Refuse and recyclable material collectors	972	A	80
Refuse and recyclable material collectors	972	B	95
Refuse and recyclable material collectors	972	C	130
Refuse and recyclable material collectors	972	D	40
Shuttle car operators	973	A	0
Shuttle car operators	973	B	0
Shuttle car operators	973	C	0
Shuttle car operators	973	D	0
Tank car, truck, and ship loaders	974	A	15
Tank car, truck, and ship loaders	974	B	20
Tank car, truck, and ship loaders	974	C	30
Tank car, truck, and ship loaders	974	D	25
Material moving workers, all other	975	A	0
Material moving workers, all other	975	B	10
Material moving workers, all other	975	C	45
Material moving workers, all other	975	D	65
Military officer and special tactical operations leader/managers	980	A	0
Military officer and special tactical operations leader/managers	980	B	0
Military officer and special tactical operations leader/managers	980	C	0
Military officer and special tactical operations leader/managers	980	D	0

Table 5. Table with all missing rows showing

CONCLUSION

As we have shown, you can display all missing categories of a class variable in your tables. Don't get caught with missing rows or columns in your tables. We presented two solutions to the problem of missing observations using PROC TABULATE and PROC SUMMARY. In the first solution, we use PROC TABULATE with the PRELOADFMT and PRINTMISS options in tandem with a user-defined format created with PROC FORMAT to ensure that all occupations appear in the rows regardless of whether or not they're in the data. In the second solution, we use PROC SUMMARY with the COMPLETETYPES option to preserve the missing occupation codes in the data. The missing rows are always preserved with PRELOADMT and PRINTMISS in PROC TABULATE and with COMPLETETYES in PROC SUMMARY. These solutions are available starting with SAS 8. You can bet on it!

CONTACT INFORMATION

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

Name: Christopher J. Boniface
U.S. Census Bureau
Washington D.C. 20233
Work Phone: (301)763-5769
E-mail: christopher.j.boniface@census.gov

Name: Janet L. Wysocki
U.S. Census Bureau
Washington D.C. 20233
Work Phone: (301)763-2446
E-mail: janet.l.wysocki@census.gov