Paper PO-04

Do You Have Too Much Class?

Janet Willis, Rho Inc., Chapel Hill NC

ABSTRACT

Do you ever use the SAS® MEANS procedure? Do you use PROC MEANS with a CLASS statement to calculate frequencies for several variables at the same time without having to sort your data set first? Have you ever stopped to check if any of your variables that you are calculating the frequencies on are missing? If your answer to the first question is yes and your answer to the second question is no then you need to take a closer look at your code. Do you know that using the TYPE statement "TYPES category*visit*(var1 var2 var3 var4);" in your PROC MEANS when all of these variables are CLASS variables can result in errors? What's MISSING?

Learn what can possibly go wrong, why, and how to avoid errors in your frequencies. Alternative solutions that provide different output structures are included to meet your programming needs. Examples include using PROC MEANS with CLASS and TYPES statements, with and without the MISSING option, as well as PROC MEANS with CLASS and VAR statements.

INTRODUCTION

It is common to need to know the number of responses received broken down by site and treatment. Using recycled code, working under tight timelines, and not knowing your data are a few things that can lead to errors in your results. Understanding how your code works is critical to quality reporting of your data. Often PROC MEANS with a CLASS statement is used to calculate the number of responses for different variables. If you have more than one variable that you want this count for and your data contains missing values for any of these variables you need to proceed with caution.

SAMPLE DATA

Let's take a look at an example. Using the data in Table 1 you are asked to calculate the number of subjects in each population within site and treatment. As you can see there are missing values for some population flags:

Site	SubjID	Treatment	Consented	Evaluable	PerProt
1	1001	A	1	1	1
1	1002	B	1	.	.
1	1003	B	1	1	1
1	1004	A	1	1	.
1	1005	A	1	1	1
2	2001	B	1	1	1
2	2002	B	1	1	1
2	2003	A	1	1	1
2	2004	A	1	1	1
2	2005	B	1	.	.

Table 1. Sample Data

METHOD 1: PROC MEANS WITH CLASS AND TYPES STATEMENTS

Let's use PROC MEANS with CLASS and TYPES statements to find the counts:

```
PROC FORMAT;
  value site
    1 = 'Hospital X'
    2 = 'Hospital Y'
    ;
  value $treat
    'A' = 'Treatment 1'
    'B' = 'Treatment 2'
    ;
  value consent
    1 = 'Consented'
    ;
  value eval
    1 = 'Evaluable'
    ;
  value perprot
    1 = 'Per Protocol'
    ;
run;

PROC MEANS noprint COMPLETETYPES nway
           data=enter(keep=site treatment consented evaluable perprot);
  format site site.
         treatment $treat.
         consented consent.
         evaluable eval.
         perprot perprot.;
  CLASS site treatment consented evaluable perprot/PRELOADFMT order=data;
  TYPES site*treatment*(consented evaluable perprot);
  OUTPUT out=method1;
run;

options pageno=1;
PROC PRINT data=method1 uniform;
  title1 'Method 1:  PROC MEANS Output';
  title2 'Using CLASS Variables SITE TREATMENT CONSENTED EVALUABLE PERPROT';
  title3 'TYPES SITE*TREATMENT*(CONSENTED EVALUABLE PERPROT)';
run;
```

An advantage of using the CLASS statement is that we don't need to sort the data set before the PROC statement. Using the COMPLETETYPES option along with the PRELOADFMT option gives us all possible values of the variables in the output regardless of whether they appear in the data. In Output 1. Method 1 Incorrect Results you can see that only records with non-missing values for all three variables CONSENTED, EVALUABLE and PERPROT are included in the counts, resulting in incorrect counts for the individual variables.

Method 1: PROC MEANS Output
Using CLASS Variables SITE TREATMENT CONSENTED EVALUABLE PERPROT
TYPES SITE*TREATMENT*(CONSENTED EVALUABLE PERPROT)

Obs	site	treatment	consented	evaluable	perprot	_TYPE_	_FREQ_
1	Hospital X	Treatment 1	.	.	Per Protocol	25	2
2	Hospital X	Treatment 2	.	.	Per Protocol	25	1
3	Hospital Y	Treatment 1	.	.	Per Protocol	25	2
4	Hospital Y	Treatment 2	.	.	Per Protocol	25	2
5	Hospital X	Treatment 1	.	Evaluable	.	26	2
6	Hospital X	Treatment 2	.	Evaluable	.	26	1
7	Hospital Y	Treatment 1	.	Evaluable	.	26	2
8	Hospital Y	Treatment 2	.	Evaluable	.	26	2
9	Hospital X	Treatment 1	Consented	.	.	28	2
10	Hospital X	Treatment 2	Consented	.	.	28	1
11	Hospital Y	Treatment 1	Consented	.	.	28	2
12	Hospital Y	Treatment 2	Consented	.	.	28	2

Output 1. Method 1 Incorrect Results

METHOD 2: PROC MEANS WITH CLASS AND TYPES STATEMENTS AND MISSING OPTION

Let's continue to use PROC MEANS with CLASS and TYPES statements but let's add the MISSING option:

```
PROC MEANS noprint COMPLETETYPES nway MISSING
           data=enter(keep=site treatment consented evaluable perprot);
  format site site.
         treatment $treat.
         consented consent.
         evaluable eval.
         perprot perprot.;
  CLASS site treatment consented evaluable perprot/PRELOADFMT order=data;
  TYPES site*treatment*(consented evaluable perprot);
  OUTPUT out=method2;
run;

options pageno=1;
PROC PRINT data=method2 uniform;
  title1 'Method 2:  PROC MEANS Output';
  title2 'Using CLASS Variables SITE TREATMENT CONSENTED EVALUABLE PERPROT with
MISSING Option';
  title3 'TYPES SITE*TREATMENT*(CONSENTED EVALUABLE PERPROT)';
run;
```

In Output 2. Method 2 Correct Results you can see that all records for all three CLASS variables are included in the counts, even records where one or more of the CLASS variables have a missing value. By adding the MISSING

option SAS considers missing as a valid level for CLASS variables. The results are correct counts for the individual variables!

Method 2: PROC MEANS Output
Using CLASS Variables SITE TREATMENT CONSENTED EVALUABLE PERPROT with MISSING Option
TYPES SITE*TREATMENT*(CONSENTED EVALUABLE PERPROT)

Obs	site	treatment	consented	evaluable	perprot	_TYPE_	_FREQ_
1	Hospital X	Treatment 1	.	.	Per Protocol	25	**2**
2	Hospital X	Treatment 1	.	.	.	25	1
3	Hospital X	Treatment 2	.	.	Per Protocol	25	**1**
4	Hospital X	Treatment 2	.	.	.	25	1
5	Hospital Y	Treatment 1	.	.	Per Protocol	25	**2**
6	Hospital Y	Treatment 1	.	.	.	25	0
7	Hospital Y	Treatment 2	.	.	Per Protocol	25	**2**
8	Hospital Y	Treatment 2	.	.	.	25	1
9	Hospital X	Treatment 1	.	Evaluable	.	26	**3**
10	Hospital X	Treatment 1	.	.	.	26	0
11	Hospital X	Treatment 2	.	Evaluable	.	26	**1**
12	Hospital X	Treatment 2	.	.	.	26	1
13	Hospital Y	Treatment 1	.	Evaluable	.	26	**2**
14	Hospital Y	Treatment 1	.	.	.	26	0
15	Hospital Y	Treatment 2	.	Evaluable	.	26	**2**
16	Hospital Y	Treatment 2	.	.	.	26	1
17	Hospital X	Treatment 1	Consented	.	.	28	**3**
18	Hospital X	Treatment 2	Consented	.	.	28	**2**
19	Hospital Y	Treatment 1	Consented	.	.	28	**2**
20	Hospital Y	Treatment 2	Consented	.	.	28	**3**

Output 2. Method 2 Correct Results

METHOD 3: PROC MEANS WITH CLASS AND VAR STATEMENTS

Now let's look at another method to get these results. Let's continue to use PROC MEANS with the CLASS statement but use the VAR statement instead of TYPES:

```
PROC MEANS noprint COMPLETETYPES nway
           data=enter(keep=site subjid treatment consented evaluable perprot);
  format site site.
         treatment $treat.
         consented consent.
         evaluable eval.
         perprot perprot.;
  CLASS site treatment/PRELOADFMT order=data;
  VAR consented evaluable perprot;
  OUTPUT out=method3 n=ncons neval nperprot;
run;

options pageno=1;
PROC PRINT data=method3 uniform;
  title1 'Method 3:  PROC MEANS Output';
  title2 'Using CLASS Variables SITE TREATMENT';
  title3 'Using VAR Variables CONSENTED EVALUABLE PERPROT';
  title4 'Format of Output Data Set is Ready for Merges';
run;
```

In Output 3. Method 3 Correct Results you can see that we have the same counts as in Output 2. Method 2 Correct Results, however, the output data set shown in Output 3. Method 3 Correct Results is structured differently.

Method 3: PROC MEANS Output
Using CLASS Variables SITE TREATMENT
Using VAR Variables CONSENTED EVALUABLE PERPROT
Format of Output Data Set is Ready for Merges

Obs	site	treatment	_TYPE_	_FREQ_	ncons	neval	nperprot
1	Hospital X	Treatment 1	3	3	3	3	2
2	Hospital X	Treatment 2	3	2	2	1	1
3	Hospital Y	Treatment 1	3	2	2	2	2
4	Hospital Y	Treatment 2	3	3	3	2	2

Output 3. Method 3 Correct Results

The structure of the output data set produced when using the VAR statement is more user friendly and can easily be merged with other data sets or used in a display.

CONCLUSION

As you can see from this simple example, missing data can cause incorrect results if you aren't careful. It's good practice to include the MISSING option when using CLASS and TYPES statements in PROC MEANS to avoid erroneous results due to missing values. Reducing the number of CLASS variables can help prevent insufficient memory issues when using the NWAY option. Using the VAR statement instead of TYPES reduces the number of variables in your CLASS statement.

Since programs are often developed and validated on small amounts of data extra care must be given to assure that missing data that may show up in your data later will not cause incorrect results. We've also seen that there is more than one way to produce correct results. Take time to think about what structure you need your output data set in and if your data set is sorted or not. Then proceed with the method that most efficiently meets your programming needs.

RECOMMENDED READING

Base SAS® Procedures Guide

CONTACT INFORMATION

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

Janet Willis
Rho, Inc.
6330 Quadrangle Drive
Chapel Hill, NC 27517
919-595-6617 (work)
919-408-0999 (fax)
Janet_Willis@rhoworld.com