

# The Sparse Option in PROC FREQ: Little Known, Yet Powerful

By Chris Moriak, AstraZeneca, Wilmington, DE

## Abstract

Have you ever needed to know all the possible combinations among specific variables within a data set? If you have ever needed to zero-fill a table, then the answer is yes. If you have, then you probably used PROC SQL or some kind of data step. Did you know PROC FREQ can provide you with all possible combinations among specific variables within a data set in one simple table statement? The little known SPARSE option in PROC FREQ accomplishes this feat quickly and neatly.

## The SPARSE Option

The SPARSE option in PROC FREQ is not properly named. It is neither meager nor thin in its ability. It is a very powerful option in the table statement. Simply stated, the SPARSE option provides “all possible combinations of levels of the variables in the table, even when some combination levels do not occur in the data.”<sup>1</sup> This is a huge help when trying to zero-fill a data set. Although current versions of SAS® now contain the new PRELOADFMT option for TABULATE and REPORT, for some types of reports, this new option will not work.

## Syntax

```
PROC FREQ data=data set;
  table variable1*variable2 /
  SPARSE out=new_data;
```

## Example 1

Clinical wants to display a descriptive statistics table by treatment, sex and race. They want to display results for every possible treatment, sex, and race combination asked for on a case report form, even though all combinations do not exist in the actual data.

## Example 1: Code

```
data demo ;
  input Ptno Treat $ Sex $
  Race $ Age;
  cards;
  100 Treat1 M White 23
  101 Treat1 F White 43
  102 Treat1 M White 34
  103 Treat2 M Black 21
  104 Treat2 M White 56
  105 Treat2 F Black 33
  ;
run;
```

```
** Find All Combinations;
proc freq data=demo;
  table treat*sex*race / SPARSE
  out=all noprint ;
run;

** Add missing combinations;
data final;
  set demo all(where=(count=0));
run;

proc sort data=final ;
  by treat sex race;
run;

proc means data=final missing
  noprint;
  by treat sex race;
  var age ;
  output out=final2 n=N min=Min
  max=Max;
run;

proc print data=final2 noobs;
  var treat sex race n min max;
run;
```

## Example 1: Output

Treat	Sex	Race	N	Min	Max
Treat1	F	Black	0	.	.
Treat1	F	White	1	43	43
Treat1	M	Black	0	.	.
Treat1	M	White	2	23	34
Treat2	F	Black	1	33	33
Treat2	F	White	0	.	.
Treat2	M	Black	1	21	21
Treat2	M	White	1	56	56

The observations with N=0 are the combinations that do not exist in the data, but they have been added to the report via the SPARSE option in PROC FREQ.

## Example 2

Clinical wants a report that shows a collected value by center, visit, and planned time. Visit 1 has only one planned time: 00:00. Visit 2, 3, and 4 have three planned times: 00:00, 0:15, and 0:30. Clinical wants to see the all possible planned times with the corresponding visits. However, you discover that one center for some reason does not have all the possible combinations of planned

times for the corresponding visits. Also, the new PRELOADFMT option will not handle this type of problem because it will give visit 1 all PTM values instead of just 00:00.

### Example 2: Code

```
data test ;
  format PTM time5.;
  input Ptno Center $ Visit PTM time5.
         Value ;
cards;
100 01 1 00:00 23
100 01 2 00:00 41
100 01 2 00:15 43
100 01 2 00:30 34
101 02 1 00:00 21
101 02 2 00:00 22
101 02 2 00:30 33
102 01 1 00:00 23
102 01 2 00:00 22
102 01 2 00:15 43
102 01 2 00:30 34
103 02 1 00:00 21
103 02 2 00:00 24
103 02 2 00:30 33
;
run;

** Find All Combinations Visit 1;
proc freq data=test;
  where visit=1;
  table Center*visit*ptm / SPARSE
         out=all1 noprint ;
run;

** Find All Combinations not Visit 1;
proc freq data=test;
  where visit ne 1;
  table Center*visit*ptm / SPARSE
         out=all2 noprint ;
run;

** Add missing combinations;
data final;
  set test
        all1(where=(count=0))
        all2(where=(count=0));
run;

proc sort data=final ;
  by Center visit ptm;
run;

proc means data=final missing
  noprint;
  by Center visit ptm;
```

```
var Value ;
output out=final1 n=N min=Min
       max=Max;
run;

proc print data=final1 noobs;
  var Center visit ptm n min max;
  format ptm time5.;
run;
```

### Example 1: Output

Center	Visit	PTM	N	Min	Max
01	1	0:00	2	23	23
01	2	0:00	2	22	41
01	2	0:15	2	43	43
01	2	0:30	2	34	34
02	1	0:00	2	21	21
02	2	0:00	2	22	24
02	2	0:15	0	.	.
02	2	0:30	2	33	33

Each center and Visit 1 has a PTM record for 00:00, while each center and non-Visit 1 has PTM records for 00:00, 00:15, and 00:30.

### Conclusion

Sometimes, it pays to read the SAS program manuals, even when you think you already know how to use a PROC. You may find a diamond in the rough such as the SPARSE option in PROC FREQ.

### References

SAS Institute Inc., SAS ® Procedures Guide, Version 6, Third Edition, Cary, NC: SAS Institute Inc., 1990.

### Contact Information

The author welcomes comments, suggestions, or questions:

Chris Moriak  
 Quantitative Decision Sciences  
 AstraZeneca  
 1800 Concord Pike  
 PO Box 15437  
 Wilmington, DE 19850

Tel: 302-885-9977  
 E-mail: christopher.moriak@astrazeneca.com

<sup>1</sup> SAS Institute Inc., SAS ® Procedures Guide, Version 6, Third Edition, Cary, NC: SAS Institute Inc., 1990, p. 335.