

Answering the Right Question with the Right PROC

Carrie Mariner, Afton-Royal Training & Consulting, Richmond, VA

ABSTRACT

When your boss comes to you and says "I need this report by tomorrow!" do you know what SAS PROC to choose? Starting with the most basic reports (FREQ and MEANS), and working up to more complex topics (REPORT), this paper will help you choose your PROC based on understanding the real question and how to answer it. The basics of reporting syntax will be covered for each PROC.

INTRODUCTION

Whether you just started with SAS or have been programming for years, there are times when you need a report, but are not sure which PROC to use. The first step is to think about the real question you are trying to answer, since the question itself will lead you to the most appropriate PROC. The most common questions are

- 1) How many are there?
- 2) How much is there?
- 3) Can you make a report that looks like this?

Once you identify the question that is really being asked, it becomes easier to choose a correct PROC and move on to making your report more sophisticated and better looking.

HOW MANY ARE THERE?

The question "how many" is usually easy to identify, although it can be hidden in layers of detail. A simple example would be "How many employees work for our company?" Only slightly more complex is "How many employees are in each department?" When the question gets hidden it often looks like this "In the Human Resources department, how many employees are paid over \$50,000 per year?" Notice that the phrase "how many" is normally still in there somewhere. An example without the key phrase would be "Are there more employees in Human Resources or in Sales?" When you stop to think about this, however, you will realize that you need to know how many are in each department first. And once you find "how many" in your reporting question, you can turn to PROC Freq.

PROC FREQ

Let's say that the question is "In the Human Resources department, how many employees are paid over \$50,000 per year?" Since we recognize the "how many" aspect, we start our investigation with PROC Freq. At first, we simply ask for frequency tables of the two variables we are interested in, department (Dept) and Salary. On the TABLES statement, we list the names of those two variables, separated by a space.

```
Proc Freq data=EmployeeInfo;
  tables Salary Dept;
Run;
```

The results look like this.

Salary				
Salary	Frequency	Percent	Cumulative Frequency	Cumulative Percent
23000	1	11.11	1	11.11
25000	1	11.11	2	22.22
35000	2	22.22	4	44.44
36000	1	11.11	5	55.56
40000	1	11.11	6	66.67
45000	1	11.11	7	77.78
50000	1	11.11	8	88.89
63000	1	11.11	9	100.00

Dept				
Dept	Frequency	Percent	Cumulative Frequency	Cumulative Percent
HR	3	33.33	3	33.33
Operations	3	33.33	6	66.67
Sales	3	33.33	9	100.00

This is helpful, but doesn't contain the vital information we need. We now know that there are three employees in Human Resources, and that only one employee is paid over \$50,000, but we don't know if that employee is in Human Resources. We could choose to look at both variables in one table by inserting an asterisk (*) between the variable names. However, given the distribution of salaries, the resulting table wouldn't look very nice or be very easy to interpret. Instead, we decide to limit the table to only the department of interest using a WHERE statement.

```
Proc Freq data=EmployeeInfo;
  tables Salary;
  where Dept='HR';
Run;
```

The results look like this.

Salary				
Salary	Frequency	Percent	Cumulative Frequency	Cumulative Percent
25000	1	33.33	1	33.33
35000	1	33.33	2	66.67
45000	1	33.33	3	100.00

Now we can see that there aren't any employees in HR who make more than \$50,000. This report may be good enough in some cases, since it does answer the question that was asked. However, it is often wise to provide slightly more information than you were asked for, to cut down on those pesky follow up questions which require follow up reports! And it is also wise to make your reports look nicer than is really required, since this makes you look better, too. In this case, it would be nice to have our salaries come out in groups rather than as individual values, which we can accomplish by creating a format especially for the salary variable using PROC Format.

```
Proc Format;
  Value Money low-<40000='Less Than $40,000'
              40000-50000='$40,000 to $50,000'
              50000<-high='More Than $50,000';
Run;
```

Notice that for each original group of values (called a range) on the left of the equals sign, we assign a label that describes the group. The '-' and '<' symbols in the original values beg for further explanation. To use these, remember that any value next to a '-' symbol (called a range indicator) is included in the range, while any value next to a '<' symbol is not included in the range. Thus in our example, both 40000 and 50000 are included in the middle category, while 50000 is not included in the last category.

Once we have our format created, we can work on the details of our PROC Freq step. We need to apply the format we created with a format statement (remember the '.' after the format name to distinguish it from a variable name). We also add a label statement to further describe the variables for our report readers. Finally, rather than limiting the report to HR, we create a two-dimensional table using the asterisk between the Salary and Dept variables. Because PROC Freq will by default add several unwanted statistics to the table when we define two dimensions, we also add the nocol, norow, and nopercnt options to remove those statistics.

```
Proc Freq data=EmployeeInfo;
  tables Salary*Dept/nocol norow nopercnt;
  format Salary money.;
  label Salary='Salary Categories'
        Dept='Department';
Run;
```

The final results look like this.

Table of Salary by Dept				
Salary(Salary Categories)	Dept(Department)			Total
Frequency	HR	Operations	Sales	
Less Than \$40,000	2	2	1	5
\$40,000 to \$50,000	1	0	2	3
More Than \$50,000	0	1	0	1
Total	3	3	3	9

HOW MUCH?

The question “how much” can be a little harder to recognize than “how many” because of the many different aspects of measurement people use. The main difference between “how many” and “how much” is the type of analysis variable or at least the way we are using the analysis variable. For a variable that is discrete or logically groups itself into discrete categories for the report, “how many” and PROC Freq are natural. When we want to describe the amount of a variable (how much) we are normally dealing with a continuous variable which we don't plan to categorize. Examples of “how much” include “How much are we paying our employees all together?”, “What's the average salary we pay?”, and “What's the lowest salary we are paying in each rank of our pay scale?” Most “how much” questions can be answered with PROC Means. Of course, other PROCs such as PROC Summary would also work, but for most cases a simple PROC Means is sufficient.

PROC MEANS

Let's consider the question “What's the lowest salary we are paying in each rank of our pay scale?” We start with a simple PROC Means to investigate the variable salary.

```
Proc Means data=EmployeeInfo;
  var Salary;
Run;
```

The results look like this.

Analysis Variable : Salary				
N	Mean	Std Dev	Minimum	Maximum
9	39111.11	12383.91	23000.00	63000.00

For our example, we need to modify the report by including only the minimum instead of the default statistics, which we can specify as “min” on the PROC Means statement. We also want to look at salary within categories of the Rank variable. This could be done by first sorting the data by Rank and then using a BY statement with the variable Rank. However, PROC Means provides a simpler alternative, a Class statement. It produces the same statistics in a similar format, but does not require sorting first.

```
Proc Means data=EmployeeInfo min;
  var Salary;
  class Rank;
Run;
```

The final results look like this.

Analysis Variable : Salary		
Rank	N Obs	Minimum
1	5	23000
2	3	40000
3	1	63000

CAN YOU?

There are many occasions when no simple “how many” or “how much” question is behind a reporting request. Instead, you are asked something like “Can you produce a report that looks like this?” The person requesting it may sketch a report on paper or show you an old report to duplicate. In these cases, you need a powerful and flexible reporting tool to let you create any rows and columns you need, containing whatever statistics are needed. The two powerhouses of SAS reporting are PROC Report and PROC Tabulate. PROC Report is fundamentally a column-based tool, while PROC Tabulate is two dimensional at heart. That said, many reports can be generated with either tool, although some lend themselves more easily to one or the other. Because PROC Report is column based, it can be easier to learn when you are just starting, so we'll focus on PROC Report.

PROC REPORT

Let's suppose the requester hands you a report that looks like the following and asks you to recreate it using SAS.

Dept	Total Salary	Average Start Date	Highest Rank
HR	\$105,000	23NOV2000	2
Operations	\$122,000	26FEB2000	3
Sales	\$125,000	07JAN2001	2

Because the columns contain information about four different variables, PROC Freq and PROC Means are clearly inappropriate for this task. The columns are defined by Dept, Salary, StartDate, and Rank. We start with the simplest PROC Report, only limiting the variables to be included with a keep= dataset option.

```
Proc Report
```

```
data=EmployeeInfo(keep= Dept Salary StartDate Rank) Nowindows;
```

```
Run;
```

Notice the Nowindows option on the PROC Report statement. It turns off the default report generating window, which will only get in the way of our programming. The results look like this.

Dept	Salary	StartDate	Rank
HR	25000	05APR2002	1
HR	35000	06MAR2001	1
HR	45000	03APR1999	2
Sales	35000	04JUL1994	1
Sales	40000	06DEC2004	2
Sales	50000	15AUG2003	2
Operations	63000	06MAR2004	3
Operations	23000	24JUL2003	1
Operations	36000	23SEP1992	1

We successfully included the columns we needed, but we would like the rows to create summary statistics rather than giving us details as has happened. By default, PROC Report generates detail information when any character variable is included as a column. We need to change this, since Dept is a variable in our report and is character.

To change any attribute of a column, or to change the way the variable is used to create the column (called “usage”) we use a Define statement. The default usage for character variables is Listing, which looks like PROC Print. Our other choices here would be Order, which still gives us detail information but puts the rows in sort order, or Group, which both sorts and groups the data. We now add a define statement for Dept, to change usage to Group, and make it a bit prettier. A slash is necessary after the name of the column on the Define statement, and any text in quotes after the slash becomes the new column header.

```
Proc Report
```

```
data=EmployeeInfo(keep= Dept Salary StartDate Rank)
```

```
Nowindows;
```

```
Define Dept/group “Department”;
```

```
Run;
```

The results look like this.

Department	Salary	StartDate	Rank
HR	105000	09SEP2082	4
Operations	122000	19JUN2080	5
Sales	125000	21JAN2083	5

We successfully created the department groups, and SAS gave us summaries of the other three variables, since they were numeric. By default, however, PROC Report wants to give us the Sum statistic for all our numeric variables, which certainly creates nonsense from our dates. We can override this with Column statements for each variable we want to change, defining the alternative statistic as appropriate.

```
Proc Report data=EmployeeInfo Nowindows;  
  Column Dept Salary StartDate Rank;  
  Define Dept      /group "Department";  
  Define Salary    /"Total Salary" format=dollar9.;  
  Define StartDate /mean "Average Start Date";  
  Define Rank      /max "Highest Rank";  
Run;
```

This code also contains a Column statement to define the order of the columns, rather than using keep= and relying on the order of the variables in the data set. This change doesn't affect our report, but does give us additional control over the report if we need it. The results look like this.

Department	Total Salary	Average Start Date	Highest Rank
HR	\$105,000	23NOV2000	2
Operations	\$122,000	26FEB2000	3
Sales	\$125,000	07JAN2001	2

We successfully recreated the original report, with a slight improvement of a nicer column header on the first column. Of course, it is wise to check with the requester to see whether any improvements will be appreciated or will cause trouble instead!

CONCLUSION

The better you understand what a reporting question really is, the easier it will be for you to decide which PROC is most appropriate. When the question is "how many" you can rely on PROC Freq, with customized formats to create categories if needed. When the question is "how much" PROC Means is a simple approach that will address your needs. When the question is "Can you make a report like this?" you may want PROC Report or PROC Tabulate, and most likely either one will be able to provide what you need, if you work hard enough! For most column-based report-duplicating tasks, PROC Report will be easier to understand and code.

REFERENCES

SAS Institute, Inc. 2002. *Base SAS® 9 Procedures Guide*. Cary, NC: SAS Institute, Inc.
SAS Institute, Inc. 2003. *SAS® Programming I: Essentials Course Notes*. Cary, NC: SAS Institute, Inc.

CONTACT INFORMATION

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

Carrie Mariner
Afton-Royal Training & Consulting
11901 Heathmere Crescent
Midlothian, Virginia 23113
Email: AftonRoyal@comcast.net
Web: www.Afton-Royal.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.