Paper 089-2011

“Repo” Your Missing Data Using PROC REPORT

Ethan Miller, SRI International, Menlo Park, Ca

ABSTRACT

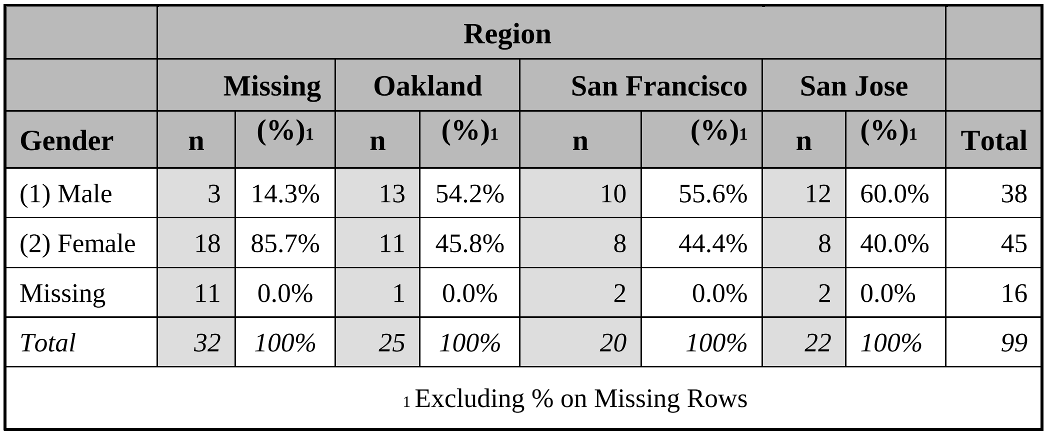
PROC REPORT is a great procedure for producing reports suitable for clients. This paper covers the use of PROC REPORT to generate simple frequency counts and percents that include missing values (similar to PROC FREQ with the MISSING OPTION), and a second percent that excludes missing observation from the calculation (similar to PROC FREQ with the MISSPRINT OPTION). This paper will also include descriptions of how to (1) create totals using compute blocks; (2) format the report; (3) add titles, footnotes, subscripts, and superscripts; and (4) specify various output options.

INTRODUCTION

Are you tired of the same old PROC FREQ and PROC TABULATE output? Are you tired of having no control over your output? Well, there is a better way to do your reporting: PROC REPORT. It can be yours today for just a small investment of time. For reporting purposes, PROC FREQ and PROC TABULATE work great; however, PROC REPORT is better for creating high quality standard reports. PROC REPORT provides flexibility with formatting options and the ability to produce customized statistics. Another important feature is that it allows the user to produce reports that show missing data without including them in the percentages.

Figure 1 below is the output table that will be discussed in this paper. It shows a crosstab of two demographic variables. In this example, it is gender by region. ODS (Output Delivery System) is used to create a PDF of the table. The “Total” row and column are both created using PROC REPORT. The column totals are calculated using a compute block in PROC REPORT. The shading of the n columns, superscripts, and footnotes are formatting options that are also created from PROC REPORT. Counts for the missing data are shown, but are not included in the percentages. This can be done easily in PROC FREQ; however, in PROC FREQ you do not have much control over the formatting of the output table. To my knowledge excluding missing from the percentages cannot be easily done using PROC TABULATE. PROC REPORT will exclude missing from the percentages while allowing control over the formatting of the table.

Figure 1: PROC REPORT Output Discussed throughout this Paper.



	Region								
	Missing		Oakland		San Francisco		San Jose		
Gender	n	(%)[1]	n	(%)[1]	n	(%)[1]	n	(%)[1]	Total
(1) Male	3	14.3%	13	54.2%	10	55.6%	12	60.0%	38
(2) Female	18	85.7%	11	45.8%	8	44.4%	8	40.0%	45
Missing	11	0.0%	1	0.0%	2	0.0%	2	0.0%	16
Total	*32*	*100%*	*25*	*100%*	*20*	*100%*	*22*	*100%*	*99*

[1] Excluding % on Missing Rows

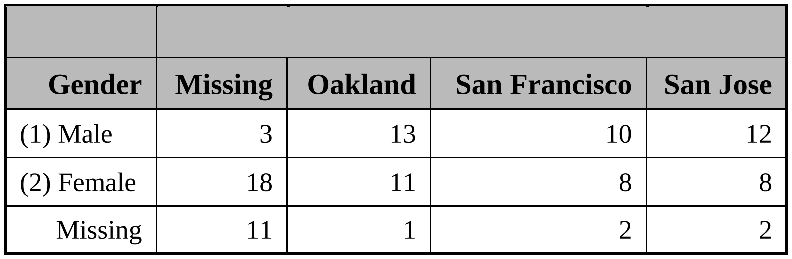
This paper examines the use of PROC REPORT to produce the table in Figure 1. The code illustrates how to generate missing counts while excluding them from percentages. Additionally, the following capability will be discussed: creation of column and row totals, footnotes, superscripts, column shading, and ODS to produce a PDF. The data set used to create the table was made up by the author, and it contains a variable for gender and a variable for region.

GETTING STARTED: No Frills PROC REPORT

The following code generates Figure 2. It is a crosstab of gender by region, and shows basic counts for the two variables. Counts of the variables are the default for reporting if no statistic is defined.

```
❶ods pdf file = "Path\SUGI_1.pdf";
proc report nowd❷ data=path.SUGIdata missing❸;
      ❾col gender  region  ;
      ❹define gender/group❺ id❻ "Gender"❼;
      define region  / across❽ "Region" ;
run;
❶ods pdf close;
```

Figure 2: Output for a Basic Crosstab using PROC REPORT.



Gender	**Missing**	**Oakland**	**San Francisco**	**San Jose**
(1) Male	3	13	10	12
(2) Female	18	11	8	8
Missing	11	1	2	2

How It Works

❶ ODS is used to put out the report in a PDF called SUGI_1.pdf.
❷ NOWD sends the report to the output destination instead of the interactive window.
❸ The MISSING option is used to include and report missing data.
❹ The DEFINE statement controls the attributes, options, and formatting for the variables in the report. After the slash the variable's function in the report is defined as well as formatting options.
❺ GROUP is used to put unique values of the variable in rows, and it can only be used on a variable from the incoming data set.
❻ ID is an option that allows variable values (row values) to be repeated every time the report breaks when it is wider than the page. This can happen when there are a lot of data columns.
❼ A quoted string after the slash serves as the label of a particular variable in the report.
❽ ACROSS is used to show each unique response in columns, and the variable it is defining must be on the incoming data set as well.
❾ The COL line is where the variables being reported are listed. The order of the variables matters here. For example, if gender is entered after region then it would appear on the right side of the report after San Jose.

Adding Statistics

The following code is used to generate the Figure 3. Figure 3 is the same crosstab as in Figure 2 except that n and percent have been added. Notice that the report is counting missing values, but not including them in the percent.

```
ods pdf file = "C:\Documents and Settings\emiller\Desktop\WUSSES_2.pdf";
proc report nowd data=path.wussdata missing;
      col gender  region , ❶ (n gender = genderpct❷)❸  ;
      define gender/group id "Gender";
      define region  / across "Region"  ;
      define genderpct /analysis❹ pctn❻ format=percent7.1 "%";
      define n/ format=8. ❺;
run;
ods pdf close;
```

Figure 3: Crosstab of Gender by Region with N and Percent.

	Region							
	Missing		Oakland		San Francisco		San Jose	
Gender	n	%	n	%	n	%	n	%
(1) Male	3	14.3%	13	54.2%	10	55.6%	12	60.0%
(2) Female	18	85.7%	11	45.8%	8	44.4%	8	40.0%
Missing	11	0.0%	1	0.0%	2	0.0%	2	0.0%

How It Works

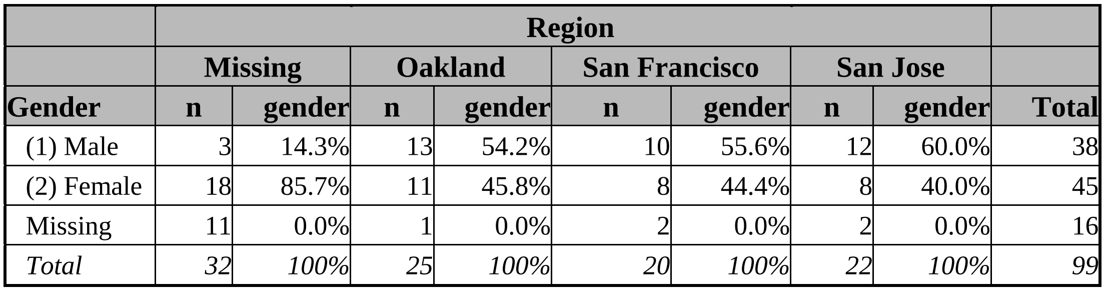
❶ Variables on the left side of the comma are the GROUP and ACROSS variables; variables on the right side of the comma are the analysis variables.
❷ Genderpct is a copy of gender that will be used to create the percentages. It is sometimes called an alias. N is now required within the parenthesis because we are also reporting percent. N is the default if no other statistic is being reported, but if you report anything other than N you will only report the new statistic. Therefore, both N and the new statistic need to be listed in the COLUMN statement.
❸ The variables within the parentheses on the right side of the comma, when reported, are nested within each value of the ACROSS variable and GROUP variable being reported on the left side of the comma. In this case for every unique value of region and gender we will get an N and a percent.
❹ Genderpct is defined as an analysis variable. This variable will be used to generate a statistic, in this case the percent of gender by region.
❺ As mentioned earlier, N is the default for PROC REPORT if no other statistic is specified. Now that percent is being reported we need to explicitly define N. Since N is a keyword, we do not have to define its function after the slash. To the right of the slash is the format we would like to use for displaying N.
❻ PCTN is the keyword used to return the percent for gender. The reason the percent is 0 on the missing data is that you can not create a percent on missing data. If percent missing is desired you would not create the alias variable Genderpct, but instead you would keep N in the parenthesis and add PCTN after the slash when you define N.

Adding Totals

The following code is used to generate Figure 4. Figure 4 is the same as Figure 3 except total column and row totals have been added.

```
ods pdf file = "C:\Documents and Settings\emiller\Desktop\WUSSES_3.pdf";
proc report nowd data=path.wussdata missing;
       col gender dumrow❷ region , (n gender = genderpct) total❻ ;
       define gender/group id noprint❺;
       define dumrow /computed "Gender"❸ ;
       define region  / across "Region";
       define genderpct /analysis pctn format=percent7.1  ;
       define total/computed 'Total'❼ ;
       define n/format=8. ;
       ❹compute dumrow /char length=10;
              if _BREAK_="_RBREAK_" then dumrow = 'Total';
              else dumrow = put(gender,gender.);
       endcomp;
       ❽compute total;
              total = sum(_c3_,_c5_,_c7_,_c9_);
       endcomp;
       ❶rbreak after / summarize;
run;
ods pdf close;
```

Figure 4: Crosstab of Gender by Region with Column and Row Totals.



	Region								
	Missing		Oakland		San Francisco		San Jose		
Gender	n	gender	n	gender	n	gender	n	gender	Total
(1) Male	3	14.3%	13	54.2%	10	55.6%	12	60.0%	38
(2) Female	18	85.7%	11	45.8%	8	44.4%	8	40.0%	45
Missing	11	0.0%	1	0.0%	2	0.0%	2	0.0%	16
Total	*32*	*100%*	*25*	*100%*	*20*	*100%*	*22*	*100%*	*99*

How It Works

❶ RBREAK AFTER creates an extra row at the end of the report. RBREAK AFTER also executes first, so before the report is created the extra row is added, even before anything is computed. RBREAK AFTER adds a line after the last row and SUMMARIZE creates total counts and percents. Unfortunately, you can not put a label on the last row when summarizing in this way. The work around is to create a dummy variable, in this case dumrow.

❷ Dumrow is added to COL. Dumrow will be populated with the values of gender, so it will go after gender but before region.

❸ Dumrow is defined as a computed variable, which is a variable that is created in a COMPUTE BLOCK in PROC REPORT and has not come from the incoming data set. Dumrow will take the values of gender, so we label it as gender.

❹ Dumrow is computed using a COMPUTE BLOCK. A COMPUTE BLOCK starts with the COMPUTE statement and ends with the ENDCOMP statement. After the slash the type and length of the new variable are defined. In the next line _BREAK_ is the automatic variable that signifies the current row of the report and where it has the value of_RBREAK_ PROC REPORT is on the last row (because we specified AFTER) of the report. The first IF statement is testing whether it is the last row, and if it is, it will be set to 'Total.' Otherwise, dumrow will have the unique formatted values of the gender variable; this is accomplished using a PUT function.

❺ Since the gender column is now coming from dumrow , we no longer need to report the gender variable, so we are using the NOPRINT option to keep the gender variable from being reported.

❻Total is added to the to the COL statement on the other side of the comma to the far right, and because it is outside the parenthesis it will show up as one column on the far right of the report. This total will be the total N for a given row.

❼ Total is defined as a computed variable and labeled as total.

❽ Total is calculated by summing across columns. _n_ denotes the column to be summed. The summing begins on column three even though it looks like it should begin in column two. The reason for this is that even though we are not printing gender, a column for it is still counted in the report.

Make It Fancy

This code is used to generate Figure 1. Figure 1 is the same as Figure 4 except that it has a footnote, superscripts, and column shading.

```
ODS escapechar='^'; ❶
ods pdf file = "C:\Documents and Settings\emiller\Desktop\WUSSES_4.pdf";
proc report nowd data=path.wussdata missing;
       col gender dumrow region , (n gender = genderpct) total ;
       define gender/group id noprint;
       define dumrow /computed "Gender" ;
       define region  / across "Region"  ;
       define genderpct /analysis pctn format=percent7.1 "(%)^{super 1}❷" ;
       define total/computed 'Total' ;
       define n/format=8. style(column)={background=graydd}❹;
       compute dumrow /char length=50;
              if _BREAK_="_RBREAK_" then dumrow = 'Total';
              else dumrow = put(gender,gender.);
       endcomp;
       compute total;
              total = sum(_c3_,_c5_,_c7_,_c9_);
       endcomp;
       rbreak after /  summarize ;
       ❸compute after _page_ /;
              line "^{super 1} Excluding % on Missing Rows";
       endcomp;
run;
ods pdf close;
```

How It Works

❶ An escape character is defined, for this example; it is the carrot. An escape character**,** in this circumstance, will allow us to control style attributes in our final report. It signals that the code in curly brackets following the escape character will not be put out as it appears.
❷ {super 1} preceded by the escape character signals ODS to produce a superscript of 1.
❸ This compute block creates a footnote. The keywords AFTER _PAGE_ are used to create the footnote at the bottom of the report, and if the table is longer than one page it will create a footnote at the bottom of the partial table on each page. Again the ESCAPE CHARACTER is used to create superscript one.
❹ A style is set to gray which is the background of the n column.

CONCLUSION

PROC REPORT is a powerful tool for creating high quality standard reports. PROC REPORT has a whole host formatting options, as well as the ability to produce customized statistics. PROC REPORT also has the ability to produce percentages where observations missing on the row variable are excluded from the calculation of the percent.

REFERENCES

Carpender A (2007). 'Carpenter's Complete Guide to the SAS® REPORT Procedure,' SAS Institute, Inc., Cary, NC

Zender C.L. (2007). 'Funny ^Stuff~ in My Code: Using ODS ESCAPECHAR,' Proceedings of the 2007 Annual SAS® Global Forum Conference, Orlando, Florida

ACKNOWLEDGMENTS

A huge thank you to the Sultan of SAS, Patrick "P-Ditty" Thornton for his Job like patience, time, and encouragement. Also, to Cyndi C-Dubs Williamson, Mary McCracken, Doris Perkins and Camille Marder for being Awesome.

RECOMMENDED READING

Carpender A (2007). 'Carpenter's Complete Guide to the SAS® REPORT Procedure,' SAS Institute, Inc., Cary, NC

CONTACT INFORMATION

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

Ethan Miller
SRI International
333 Ravenswood Ave BS159
Menlo Park, Ca 94025
Work Phone: (650) 859-5726
E-mail: ethan.miller@sri.com