

Using PROC REPORT to Generate “Impossible” Totals

Andrea J. Decker

Systems Seminar Consultants, Madison, WI

Abstract

PROC REPORT can be manipulated to generate summary lines at many different levels. The SUMMARIZE option of the BREAK statement performs these calculations for many different statistics. Problems arise when percentages, weighted percentages, and weighted averages need to be summarized to a certain level. This paper will describe problems involved and solutions to calculate these “impossible” totals.

Introduction

PROC REPORT is a very powerful and easy procedure used to generate very attractive reports. The programmer controls the appearance and content of every column, thus making it more flexible to use than the PROC PRINT. The data step can also be used to create the same reports, but in most cases requires more coding, and the programmer must spend time counting columns for placement of data. PROC REPORT also has the ability to create summary lines wherever they are needed. Problems arise when this procedure is used to calculate summary lines which contain percentages.

The data used to generate the reports will be sample credit information about 16 people who have purchased a certain number of items within a particular department in one of two stores. Their credit balance is shown along with the annual percentage rate and the number of items purchased within the last year. The variable STORE contains a store code, either a 10 or 20. The DEPT will have a code for the department within the store. The PROC REPORT will translate the department code into a character string with a format.

```
PROC FORMAT;
  VALUE $DEPTFMT
    '201' = 'Clothing'
    '202' = 'Hardware'
    '203' = 'Shoes'
    '204' = 'Sports'
  ;
RUN;
```

The cardholder’s name is stored in the NAME variable. The BALANCE contains the current

balance on the account. The APR represents the annual percentage rate. The number of purchases in the last year is recorded in the NUMPURCH variable.

Calculating Simple Percentages

Percentage calculations, such as Annual Percentage Rate, must be handled in a special way when using PROC REPORT. The bulk of the problem lies in the summary lines the PROC produces. By default the summary lines produce a SUM statistic, or simply adding up the column. This is not the value desired.

The first attempt at a PROC REPORT will illustrate the problem with summarizing percentages. The PROC statement contains some options. The NOWINDOWS option is used to send the output directly to the output window without the use of the report window. The SPLIT= option is used to control where the column headings are split onto the next line. The HEADLINE option will underline the header, and the HEADSKIP will skip a line before beginning the detail lines of the report.

```
PROC REPORT DATA=ANDREA.RPTTEST NOWINDOWS
  SPLIT='*' HEADLINE HEADSKIP;
```

A COLUMNS statement is coded next. This statement will control the order of the columns of the report. The order of the variables in the COLUMNS statement corresponds to the order of the columns in the report. In this instance, the variables listed all come from the SAS data set. It is possible to compute a column and to create an alias for a variable, however they are not needed for this report.

```
COLUMNS STORE DEPT NAME
  NUMPURCH BALANCE APR;
```

The DEFINE statements are where the attributes are defined for each column of the report. As mentioned before, the programmer has the ability to control every detail of each column. Variables defined as ORDER will be ordered by PROC REPORT, and require no sorting beforehand. An ORDER= option will give control of whether the column is ordered by the formatted or unformatted, internal, value. Numeric variables

defined as ANALYSIS can be summarized at certain breaking points. The column width is explicitly given with the WIDTH= option. By default there will be 2 physical columns between each report column, and the columns are centered on the page if the center option is on. Values can be formatted using the FORMAT= option. The DEPT column uses the previously created format \$DEPTFMT. to give a descriptive label to a department code. Any column headings are specified within quotes.

```

DEFINE STORE      / ORDER
                   WIDTH=5
                   'STORE*NAME';
DEFINE DEPT       / ORDER
                   ORDER=FORMATTED
                   WIDTH=15
                   FORMAT=$DEPTFMT.
                   'DEPARTMENT';
DEFINE NAME       / ORDER
                   ORDER=INTERNAL
                   WIDTH=10
                   'CARDHOLDER*NAME';
DEFINE NUMPURCH  / ANALYSIS
                   WIDTH=10
                   'NUMBER OF*PURCHASES';
DEFINE BALANCE   / ANALYSIS
                   WIDTH=15
                   FORMAT=DOLLAR15.2
                   'CURRENT*BALANCE';
DEFINE APR       / ANALYSIS
                   WIDTH=10
                   FORMAT=PERCENT10.2
                   'APR';

```

In order to create summary lines, a BREAK statement must be coded. The BREAK statement gives control over where the summary line will occur. The statements below will create a summary line once the department and store values change, resulting in department and store total lines. The SUMMARIZE option on the BREAK statement will produce the statistics for all ANALYSIS variables. By default, the SUM statistic is used. As the columns have been defined above, the NUMPURCH, BALANCE, and APR will be added to achieve a summary. SUPPRESS will keep from printing the value of the particular BREAK variable on the summary line. The SKIP option will skip a line after the summary line. The OL option will allow a single line over the department totals, as the DOL will allow a double line over the store totals.

```

BREAK AFTER DEPT / SUMMARIZE SUPPRESS
                  SKIP OL;

```

```

BREAK AFTER STORE / SUMMARIZE SUPPRESS
                   SKIP DOL;
RUN;

```

(see Figure #1 for output results)

As mentioned previously, all analysis variables have the SUM statistic associated with them. The break statements do all the summarization of the analysis variables. An average of the APR column would make more sense here, since we do not want the addition of the percentages. By adding the MEAN statistic to the Define statement of the APR column, the summarized lines will show the average of the column, not the sum.

```

DEFINE APR       / ANALYSIS
                   MEAN
                   WIDTH=10
                   FORMAT=PERCENT10.2
                   'APR';
BREAK AFTER DEPT / SUMMARIZE SUPPRESS
                  SKIP OL;
BREAK AFTER STORE / SUMMARIZE SUPPRESS
                  SKIP DOL;

```

(see Figure #2 for output results)

The BREAK statements can remain the same. The SUMMARIZE option will now use the average of the APR.

Reporting at a higher level (summarizing detail lines into departments and showing only store totals) requires a little more coding. In order to summarize the detail lines down to just department and store totals, we need to include a PROC SUMMARY beforehand. The NWAY option is used to get the highest level of interaction between the store and department. The SAS data set created from the OUTPUT statement contains the detail lines summarized into departments within the two stores.

```

PROC SUMMARY DATA=ANDREA.RPTTEST NWAY;
  CLASS STORE DEPT;
  VAR NUMPURCH BALANCE APR;
  OUTPUT OUT=SUMM1OUT
         SUM(NUMPURCH)=PURCHTOT
         SUM(BALANCE)=BALTOT
         MEAN(APR)=MEANAPR;
RUN;

```

The PROC REPORT has not changed. SUMM1OUT will be fed into the procedure to create the report. This new dataset has different variable names, but the new define statements

look identical to their unsummarized counterparts. The MEAN statistic is coded into the MEANAPR column to provide us with the average of the APR, not the sum, in the summary lines.

```
PROC REPORT DATA=SUMM1OUT NOWINDOWS
      SPLIT='*' HEADLINE HEADSKIP;
  COLUMNS STORE DEPT PURCHTOT
      BALTOT MEANAPR;
  DEFINE STORE / ORDER
      WIDTH=5
      'STORE*NAME';
  DEFINE DEPT / ORDER
      ORDER=FORMATTED
      WIDTH=15
      FORMAT=$DEPTFMT.
      'DEPARTMENT';
  DEFINE PURCHTOT / ANALYSIS
      WIDTH=10
      'NUMBER OF*PURCHASES';
  DEFINE BALTOT / ANALYSIS
      WIDTH=15
      FORMAT=DOLLAR15.2
      'CURRENT*BALANCE';
  DEFINE MEANAPR / ANALYSIS
      MEAN
      WIDTH=10
      FORMAT=PERCENT10.2
      'APR';
  BREAK AFTER STORE / SUMMARIZE SUPPRESS
      SKIP OL;
RUN;
      (see Figure #3 for output results)
```

Only one BREAK statement was coded, since the detail lines are summarized already by department. We only need the STORE totals to be computed.

Computing Weighted Percentages

Getting simple averages of percentages is not complicated. What becomes more difficult is when the columns represent a weighed percentage. PROC REPORT does support a WEIGHT statement, but this statement will weight all analysis variables. This is not desired because only one variable needs to be weighed. Another way to get a weighted variable is to use the PROC SUMMARY. The same problem exists here, all variables will be weighted. Therefore, two PROC SUMMARY steps are needed.

The first step looks identical to the previous example, but the average of the APR variable is

removed. The sum of the number of purchases and balances is calculated at store-department level. NWAY statements are used in both steps, as we are only interested in the highest level of interaction between STORE and DEPT.

```
PROC SUMMARY DATA=ANDREA.RPTTEST NWAY;
  CLASS STORE DEPT;
  VAR NUMPURCH BALANCE;
  OUTPUT OUT=SUMM2A
      SUM(NUMPURCH)=PURCHTOT
      SUM(BALANCE)=BALTOT;
RUN;
```

In order to get the APR to be weighted by the balance, we must use a second step to compute the weighted variable and roll up the data to store-department level.

```
PROC SUMMARY DATA=ANDREA.RPTTEST NWAY;
  CLASS STORE DEPT;
  WEIGHT BALANCE;
  VAR APR;
  OUTPUT OUT=SUMM2B
      MEAN(APR)=WTDAPR;
RUN;
```

The resulting SAS data sets must be merged together by STORE and DEPT to give one SAS data set we can plug into the PROC REPORT.

```
DATA SUMM2;
  MERGE SUMM2A
      SUMM2B;
  BY STORE DEPT;
RUN;
```

The resulting SAS data set contains the store and departments along with the total purchases, total balances, and an annual percentage rate weighted by the balance.

```
PROC REPORT DATA=SUMM2 NOWINDOWS
      SPLIT='*' HEADLINE HEADSKIP;
  COLUMNS STORE DEPT PURCHTOT
      BALTOT WTDAPR;
  DEFINE STORE / ORDER
      WIDTH=5
      'STORE*NAME';
  DEFINE DEPT / ORDER
      ORDER=FORMATTED
      WIDTH=15
      FORMAT=$DEPTFMT.
      'DEPARTMENT';
  DEFINE PURCHTOT / ANALYSIS
      WIDTH=10
      'NUMBER OF*PURCHASES';
```

```

DEFINE BALTOT / ANALYSIS
                WIDTH=15
                FORMAT=DOLLAR15.2
                'CURRENT*BALANCE';
DEFINE WTDAPR / ANALYSIS
                MEAN
                WIDTH=10
                FORMAT=PERCENT10.2
                'WEIGHTED*APR';
BREAK AFTER STORE / SUMMARIZE SUPPRESS
                SKIP OL;
RUN;

```

(see Figure #4 for output results)

The MEAN statistic was included with the WTDAPR definition. If it were not, the percentages would have been added. But, this is not the desired result, since the summary lines are the average of the weighted percentages, not the weighted percentage at the store level. This is possible, but requires quite a bit more coding and becomes quite complex. We eventually want to let PROC REPORT calculate the summary line. The formula for the Store Weighted APR is the addition of the store-department contribution to the weighted APR for the particular store. The formula for the store-department piece of the store weighted APR is the store-department weighted APR multiplied by the store-department balance divided by the store balance.

The information must be summarized with the same techniques used before, along with attaching a store balance to each observation in order to perform the necessary calculations.

PROC SUMMARY is used to generate two SAS data sets in one step. NWAY was used in the previous examples to keep only the highest level of interaction. If we take this option off, we get all levels. The `_TYPE_=2` is the individual store totals. This SAS data set will contain the sum of the balances for each store, which is the denominator of the Store Weighted APR formula. The second SAS data set contains the `_TYPE_=3`, or the highest level of interaction. This SAS data set contains the details for the report, and also the store-department balance for the numerator of the Store Weighted APR formula.

```

PROC SUMMARY DATA=ANDREA.RPTTEST;
CLASS STORE DEPT;
VAR NUMPURCH BALANCE;
OUTPUT OUT=SUMM2A(WHERE=( _TYPE_=3))

```

```

SUM(NUMPURCH)=PURCHTOT
SUM(BALANCE)=BALTOT;
OUTPUT OUT=SUMM2A2(WHERE=( _TYPE_=2))
SUM(BALANCE)=STBALTOT;
RUN;

```

Again PROC SUMMARY is used to create a SAS data set which has the APR weighted by balance. NWAY is used because the only the highest level of interaction between STORE and DEPT are needed.

```

PROC SUMMARY DATA=ANDREA.RPTTEST NWAY;
CLASS STORE DEPT;
WEIGHT BALANCE;
VAR APR;
OUTPUT OUT=SUMM2B
        MEAN(APR)=WTDAPR;
RUN;

```

The SAS data sets, which contain the highest level of interaction, are merged together. This dataset will look identical to the SAS data set that was created for the previous report.

```

DATA SUMM2;
MERGE SUMM2A
        SUMM2B;
BY STORE DEPT;
RUN;

```

The balance for the store totals needs to be attached to the SAS data set just created in order for the weighted APR column to be computed correctly. The SAS data set SUMM2A2, which contains the store balance totals, is now merged with the previously created set. The only variables being added to the merge from SUMM2A2 are the balance and the store number. This will prevent any necessary variables to be overridden. The piece of the Store Weighted APR is also calculated in this data step.

```

DATA SUMM2;
MERGE SUMM2
        SUMM2A2(KEEP=STBALTOT STORE);
BY STORE;
TOTAPR=(WTDAPR*BALTOT)/STBALTOT;
RUN;

```

The SAS data set is now ready to be reported on. The columns are the same as previous reports, but with one addition, TOTAPR. This is the value created from the previous data step. In the DEFINE statement for this variable, NOPRINT option is specified. This will prevent this

column from showing up on the final report. PROC REPORT will add the TOTAPR column to create the weighted APR for the store summary lines.

```

PROC REPORT DATA=SUMM2 NOWINDOWS
      SPLIT='*' HEADLINE HEADSKIP;
  COLUMNS STORE DEPT PURCHTOT
      BALTOT WTDAPR TOTAPR;
  DEFINE STORE / ORDER
      WIDTH=5
      'STORE*NAME';
  DEFINE DEPT / ORDER
      ORDER=FORMATTED
      WIDTH=15
      FORMAT=$DEPTFMT.
      'DEPARTMENT';
  DEFINE PURCHTOT / ANALYSIS
      WIDTH=10
      FORMAT=COMMA10.0
      'NUMBER OF*PURCHASES';
  DEFINE BALTOT / ANALYSIS
      WIDTH=15
      FORMAT=DOLLAR15.2
      'CURRENT*BALANCE';
  DEFINE WTDAPR / ANALYSIS
      WIDTH=10
      FORMAT=PERCENT10.2
      'WEIGHTED*APR';
  DEFINE TOTAPR / ANALYSIS
      NOPRINT;
  BREAK AFTER STORE / OL SUPPRESS SKIP;
  COMPUTE AFTER STORE ;
      LINE @33 PURCHTOT.SUM COMMA10.0
      +2 BALTOT.SUM DOLLAR15.2
      +2 TOTAPR.SUM PERCENT10.2;
  ENDCOMP;
RUN;

```

(see Figure #5 for output results)

The BREAK statement does not contain the SUMMARIZE option. The summary lines will be computed using a COMPUTE block. The COMPUTE block contains a LINE statement. The LINE statement works almost like the PUT

statement does in the data step, but does not support some of its options. Absolute and relative pointers are used to point to where the information will be placed on the report, and formats are used to insert commas, dollar signs, and percent signs. The variables are referred to by two level names. The first part is the variable name, with the second being the statistic for summarization. The PURCHTOT and BALTOT totals will be the same as when a SUMMARIZE option was used on the BREAK statement. The WTDAPR column is not used. What is shown instead is the sum of the TOTAPR column. This is the correct weighted APR at the store level.

In Summary

PROC REPORT is a very easy and powerful reporting procedure. It can create from very simple reports with simple summary lines, to extremely complex reports with complex summary lines. When used with PROC SUMMARY and some additional computing statements, PROC REPORT can be manipulated to produce calculations often thought “impossible.”

Trademark Notice

SAS, PROC REPORT, and PROC SUMMARY are registered trademarks of the SAS Institute Inc., Cary, NC, USA and other countries.

Any questions or comments regarding the paper may be directed to the author:

Andrea J. Decker
 Systems Seminar Consultants
 2997 Yarmouth Greenway Dr.
 Madison, WI 53711
 Phone: (608) 278-9964
 Fax: (608) 278-0065
 E-mail: adecker@sys-seminar.com

Appendix A - Output

Figure 1:

PROC REPORT WITH INCORRECT APR TOTALS (MEAN) 1
 SUMMARIZED BY STORE AND DEPARTMENT

STORE NAME	DEPARTMENT	CARDHOLDER NAME	NUMBER OF PURCHASES	CURRENT BALANCE	APR
10	Clothing	Alicia	16	\$3,985.39	11.00%
		Bill	13	\$2,985.83	10.00%
			29	\$6,971.22	21.00%
	Hardware	Brian	15	\$982.54	12.00%
		Kara	17	\$2,904.14	12.00%
			32	\$3,886.68	24.00%
	Shoes	Angie	29	\$873.29	10.00%
		Jaime	16	\$2,356.82	12.00%
			45	\$3,230.11	22.00%
	Sports	Carolyn	15	\$1,210.49	13.00%
		Eric	21	\$3,902.93	9.00%
			36	\$5,113.42	22.00%
			=====	=====	=====
			142	\$19,201.43	89.00%
20	Clothing	Andrea	8	\$1,143.53	9.00%
		James	23	\$763.62	8.00%
			31	\$1,907.15	17.00%
	Hardware	Joanne	10	\$1,923.49	13.00%
		Mark	20	\$3,903.82	11.00%
			30	\$5,827.31	24.00%
	Shoes	Amy Sue	15	\$494.50	12.00%
		Rebecca	20	\$892.25	14.00%
			35	\$1,386.75	26.00%
	Sports	Julie	9	\$245.03	14.00%
		Matthew	18	\$782.65	10.00%
			27	\$1,027.68	24.00%
			=====	=====	=====
			123	\$10,148.89	91.00%

Figure 2:

PROC REPORT WITH CORRECT APR TOTALS (MEAN)
SUMMARIZED BY STORE AND DEPARTMENT

STORE NAME	DEPARTMENT	CARDHOLDER NAME	NUMBER OF PURCHASES	CURRENT BALANCE	APR	
10	Clothing	Alicia	16	\$3,985.39	11.00%	
		Bill	13	\$2,985.83	10.00%	
			29	\$6,971.22	10.50%	
	Hardware	Brian	15	\$982.54	12.00%	
		Kara	17	\$2,904.14	12.00%	
			32	\$3,886.68	12.00%	
	Shoes	Angie	29	\$873.29	10.00%	
		Jaime	16	\$2,356.82	12.00%	
			45	\$3,230.11	11.00%	
	Sports	Carolyn	15	\$1,210.49	13.00%	
		Eric	21	\$3,902.93	9.00%	
			36	\$5,113.42	11.00%	
			=====	=====	=====	
				142	\$19,201.43	11.13%
20	Clothing	Andrea	8	\$1,143.53	9.00%	
		James	23	\$763.62	8.00%	
			31	\$1,907.15	8.50%	
	Hardware	Joanne	10	\$1,923.49	13.00%	
		Mark	20	\$3,903.82	11.00%	
			30	\$5,827.31	12.00%	
	Shoes	Amy Sue	15	\$494.50	12.00%	
		Rebecca	20	\$892.25	14.00%	
			35	\$1,386.75	13.00%	
	Sports	Julie	9	\$245.03	14.00%	
		Matthew	18	\$782.65	10.00%	
			27	\$1,027.68	12.00%	
			=====	=====	=====	
				123	\$10,148.89	11.38%

Figure 3:

PROC REPORT WITH CORRECT APR TOTALS (MEAN) 2
SUMMARIZED BY STORE AND DEPARTMENT

STORE NAME	DEPARTMENT	NUMBER OF PURCHASES	CURRENT BALANCE	APR
10	Clothing	29	\$6,971.22	10.50%
	Hardware	32	\$3,886.68	12.00%
	Shoes	45	\$3,230.11	11.00%
	Sports	36	\$5,113.42	11.00%
		-----	-----	-----
		142	\$19,201.43	11.13%
20	Clothing	31	\$1,907.15	8.50%
	Hardware	30	\$5,827.31	12.00%
	Shoes	35	\$1,386.75	13.00%
	Sports	27	\$1,027.68	12.00%
		-----	-----	-----
		123	\$10,148.89	11.38%

Figure 4:

PROC REPORT WITH INCORRECT WEIGHTED APR TOTALS 1
SUMMARIZED BY STORE
SUMMARY LINE SHOWS AVERAGE OF WEIGHTED APR, NOT WEIGHTED AT STORE LEVEL

STORE NAME	DEPARTMENT	NUMBER OF PURCHASES	CURRENT BALANCE	WEIGHTED APR
10	Clothing	29	\$6,971.22	10.57%
	Hardware	32	\$3,886.68	12.00%
	Shoes	45	\$3,230.11	11.46%
	Sports	36	\$5,113.42	9.95%
		-----	-----	-----
		142	\$19,201.43	10.99%
20	Clothing	31	\$1,907.15	8.60%
	Hardware	30	\$5,827.31	11.66%
	Shoes	35	\$1,386.75	13.29%
	Sports	27	\$1,027.68	10.95%
		-----	-----	-----
		123	\$10,148.89	11.13%

Figure 5:

PROC REPORT WITH CORRECT WEIGHTED APR TOTALS

1

SUMMARIZED BY STORE

STORE NAME	DEPARTMENT	NUMBER OF PURCHASES	CURRENT BALANCE	WEIGHTED APR
10	Clothing	29	\$6,971.22	10.57%
	Hardware	32	\$3,886.68	12.00%
	Shoes	45	\$3,230.11	11.46%
	Sports	36	\$5,113.42	9.95%
		-----	-----	-----
		142	\$19,201.43	10.84%
20	Clothing	31	\$1,907.15	8.60%
	Hardware	30	\$5,827.31	11.66%
	Shoes	35	\$1,386.75	13.29%
	Sports	27	\$1,027.68	10.95%
		-----	-----	-----
		123	\$10,148.89	11.24%