ABSTRACT
PROC REPORT is a tool for tabulating and reporting the contents of SAS® data sets. PROC REPORT’s strength is its ability to create table columns, modify individual cell values, and use information from the different rows of the report. This flexibility allows users to create tables that show cumulative totals and cumulative percent. Cumulative totals are used to report values such as the number of responses year-to-date and the percent of a total for all periods to date. Two PROC REPORT features that allow this are the “COMPUTE” block, and the distinction that PROC REPORT makes between report variables and data variables. This paper discusses how PROC REPORT builds a report, the different kinds of “COMPUTE” blocks and how they function, and what report and data variables are and how they work. The code to produce one and two level tables containing cumulative row totals and percent are shown with explanatory comments.

Key word: PROC REPORT, Cumulative total, Cumulative Percent

INTRODUCTION
PROC REPORT is an excellent tool for preparing standard reports. An advantage of PROC REPORT over PROC TABULATE is its ability to define the columns of a report and manipulate the rows of the report when producing nonstandard reports. One important nonstandard report used by statisticians is a table of cumulative totals or percent. Cumulative totals are needed when displaying both the number of responses to a survey and the cumulative responses to date by month. This paper discusses the following PROC REPORT features: (1) the difference between a report variable and a data variable, (2) how PROC REPORT builds a report, and (3) the use of the compute statement.

PROC REPORT is flexible enough to allow calculation of cumulative total and percent. A complete two level table showing cumulative totals and percent at each level is shown in Figure 1. The code for the final table is illustrated by starting with a simple table then adding more complexity. A basic PROC REPORT table is shown in Figure 2 and a more complex one level table is shown in Figure 3. For each table the code is given and annotated to explain what is being done and why. The code to generate the artificial data used in the example is given in the Appendix.

Figure 1 below shows the final report produced here. It is different from most reports because the last two columns give a cumulative total and percent. The basic characteristics of a cumulative total are:
1. the cumulative total for the first row is always equal to the sum from the first row,
2. the cumulative total from the last row is always equal to the sum of all row totals,
3. the cumulative total for a row is equal to the row total plus the cumulative total of the previous row, and
4. when row totals are positive the cumulative total for any row is greater than or equal to the cumulative total of the previous row.

HOW PROC REPORT WORKS
PROC REPORT is a cross between PROC PRINT, PROC SUMMARY, and the data step. It combines features of all of them. An important part of PROC REPORT is how it uses different types of variables and how the compute block processes those variables.

Data Variables versus Report Variables
PROC REPORT has two different types of variables: data variables and report variables. Data variables are variables that appear only within a COMPUTE block and do not appear on the COLUMN statement. Data variables retain their value until specifically assigned a value. Once assigned a value, they retain that value until it is changed. Data step variables retain...
their values between COMPUTE blocks for the same report line and between report lines. This is similar to how data step variables act when a retain statement is used. Report variables are variables that appear in the COLUMN statement that defines the variables appearing in the report and the order in which they appear. A report variable may or may not appear in a compute block. Report variables are initialized to missing at the beginning of each row of the report; this is similar to how data set variables are handled in the data step. There are two different varieties of report variables: (1) variables that come from the input data set (e.g. Sales) and (2) COMPUTED variables that are created (e.g. YTD). Computed variables are created from left to right as they appear on the COLUMN statement. Data variables can use report variables, other data variables, or both.

**COMPUTE BLOCK**

A compute block is one or more programming statements that appear between COMPUTE and ENDCOMP statements. A COMPUTE block can be associated with a report item (data set variable, statistic, or computed variable), or a location (before or after a group of observations, or at the top or bottom of a page or of the complete report). When a compute block is used with a report variable it can be used either to define the computed variable (a variable not on the input data set), to modify a report variable, or to define or change the display characteristics of a report variable. Examples of the use of a compute block are given below:

To create a new report variable.

```sas
COMPUTE PROFIT;
  PROFIT = REVENUE.SUM - EXPENSES.SUM
ENDCOMP;
```

To change the scale of a report variable from the input data set.

```sas
COMPUTE BRITISH_POUNDS ;
  BRITISH_POUNDS = DOLLARS.SUM / 1.49;
  /*CONVERSION RATE 1.49 DOLLARS = 1 POUND */;
ENDCOMP;
```

To add footnote at the bottom of each page of the table.

```sas
COMPUTE AFTER PAGE;
  Line "See list of symbols at end of table";
ENDCOMP;
```

To add symbols at the end of the table.

```sas
COMPUTE AFTER;
  Line "(na) Not Available";
  Line "(z) Zero";
ENDCOMP;
```

**HOW PROC REPORT BUILDS A REPORT**

This is an abbreviated version of a detailed step-by-step description in Chapter 32, The PROC REPORT Procedure, in the version 8 SAS procedures guide. PROC REPORT is built in two distinct steps.

In the first step, PROC REPORT creates a temporary data set. The data set includes summary data by group, order and across variables. It also includes summary records for RBREAK, BREAK, and COMPUTE BEFORE | AFTER statements associated with a report variable. Creation of this preliminary summary data gives the report access to universe or group total before the first line of the report or group is printed.

In the second step, PROC REPORT created the report line by line. It initializes all data step variables to missing and then sequentially constructs each row of the report. All report variables are initialized to missing. The value of each report variable is determined from left-to-right; values for computed variables come from executing statements in the compute block. Only report variables to the left of the variable being computed can be used in the calculation. All other values are obtained from the temporary file created at the start.

**A SIMPLE TABLE**

The final table starts with the basic table in Figure 2. Table One is columns two, three and four of Table Three in Figure 1. This simple table and the code that produces it is given below.

**The Basic Table and SAS Code**

The basic table is just the standard PROC REPORT table.

**FIGURE 2**

**TABLE ONE: 1998 SALES BY QUARTER**

<table>
<thead>
<tr>
<th>QUARTER</th>
<th>TOTAL</th>
<th>PERCENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)</td>
<td>(2)</td>
<td>(3)</td>
</tr>
<tr>
<td>1</td>
<td>$960</td>
<td>17.78%</td>
</tr>
<tr>
<td>2</td>
<td>$1,705</td>
<td>31.59%</td>
</tr>
<tr>
<td>3</td>
<td>$964</td>
<td>17.86%</td>
</tr>
<tr>
<td>4</td>
<td>$1,769</td>
<td>32.77%</td>
</tr>
</tbody>
</table>

$5,397 100.0%

**FIGURE 3**

**TABLE TWO: 1998 SALES BY QUARTER and YEAR TO DATE**

<table>
<thead>
<tr>
<th>QUARTER</th>
<th>TO DATE</th>
<th>PERCENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)</td>
<td>(2)</td>
<td>(3)</td>
</tr>
<tr>
<td>1</td>
<td>$960</td>
<td>17.78%</td>
</tr>
<tr>
<td>2</td>
<td>$2,375</td>
<td>48.71%</td>
</tr>
<tr>
<td>3</td>
<td>$3,328</td>
<td>67.23%</td>
</tr>
<tr>
<td>4</td>
<td>$5,136</td>
<td>100.00%</td>
</tr>
</tbody>
</table>

$5,397 100.0%

**Things To Notice in the Table One Code**

Several things to notice in this generic PROC REPORT code are:

- The use of the split character "*" to produce the column headers,
- The use of an alias to produce the column percent, and
- The use of the RBREAK command to produce the grand total for all groups.

**A MORE COMPLICATED TABLE:**

**CUMULATIVE TOTAL AND PERCENT COLUMNS**

Figure 3 shows Table One with a cumulative total and percent column added by making use of PROC REPORT data variables and the COMPUTE BEFORE blocks.
Table Two is produced by adding too and modifying the code used to produce Table One.

```sas
PROC REPORT DATA=quarter nowd SPLIT="*" OUT=two;
WHERE(year=98);
TITLE1 "FIGURE 3";
TITLE2 "TABLE TWO: 1998 SALES BY QUARTER and YEAR TO DATE";
COLUMN quarter ('SALES ' "BY QUARTER " sales sales=pct) ('YEAR TO DATE ' YTD cumptotal )
);
DEFINE quarter / GROUP FORMAT=S. 'QUARTER+(1)'; CENTER;
DEFINE SALES / ANALYSIS FORMAT=DOLLAR8. SUM 'Total+(2)';
DEFINE PCT / ANALYSIS PCT SUM FORMAT=PERCENT8.2 'Percent+(3)';
DEFINE YTD / COMPUTED "TOTAL+(4)" FORMAT=DOLLAR8.0;
DEFINE cumptotal / computed 'PERCENT+(5)' FORMAT=PERCENT9.2;
COMPUTE BEFORE; @
total=sales.sum; +
cumtotal=0; @
ENDCOMP;

COMPUTE YTD ; @
cumtotal+sales.sum; @
YTD=cumtotal; @
IF _BREAK_="_RBREAK_" THEN YTD=.; @
ENDCOMP;

COMPUTE cumptotal;
cumptotal=ytd/total; @
ENDCOMP;

RBREAK AFTER /SUMMARIZE UL OL;
RUN;

THINGS TO NOTICE IN THE TABLE TWO CODE

The key to calculating cumulative totals and percent is to save the group total at the beginning and to calculate the cumulative total correctly. This requires the use of data variables and a COMPUTE BEFORE block.

The initial “COMPUTE BEFORE” block creates a summary record that contains the total for all report variables on the input data set. This is the source of the information on total sales used as the denominator in calculating the cumulative percent.

The variable total is a data variable that is set to the value of sales for all records included in the table.

The variable cumptotal is a second data variable that is initialized to zero. It is a variable that will later contain the cumulative total.

The value of the report variable YTD calculates the value of the report variable that holds the cumulative total.

For each record on the temporary data set, the value of sales for the current record (sales.sum) is added to the data variable cumptotal that holds the sum of all previous records.

The value of the report variable YTD is set equal to the current value of the data variable cumptotal. Notice that the cumulative total for the last record in the group will always be equal to the grand total for variable in column (2).

When summary records are created either by a COMPUTE BLOCK, BREAK, or RBREAK statement, a SAS variable called _BREAK_ is set. When the summary record is for the entire data set and created by a RBREAK statement, the value of _BREAK_ is _RBREAK_. When the summary record is created because of either a BREAK or COMPUTE BLOCK associated with a variable, the value of _BREAK_ is the name of the variable. The value of the _BREAK_ variable is used to recognize a summary line of table. Since the sum of the cumulative total and percent of values of the individual rows has no meaning, the sum is set to missing with this statement.

This compute block calculates the cumulative percent. It uses the report variable YTD containing the cumulative total and the data variable total containing the grand total. Notice that the cumulative total for that last record is always equal to 100 percent.

The output data set is the temporary data set used to create the final PROC REPORT after the report variables have been created or modified. A PROC PRINT of that file is given below.

```
Obs qtr sales pct YTD Cumptotal _BREAK_
1 . 5397.27 1.00000 . 5397.27 _RBREAK_ 
2 1 959.59 0.17779 959.59 0.17779
3 2 1704.97 0.31590 2664.56 0.49369
4 3 963.90 0.17859 3628.45 0.67228
5 4 1768.81 0.32772 5397.27 1.00000
6 . 5397.27 1.00000 . 5397.27 _RBREAK_
```

Notice that the PROC REPORT table in Figure 2 has five rows and the data set in Figure 4 has six records. Records 2, 3, 4, and 5 correspond to the four quarter totals and were produced by the “DEFINE quarter / GROUP…” statement. Record 6 corresponds to the year total and was produced by the “RBREAK” statement. The first record is also the total for all records on the data set and was created by the COMPUTE BEFORE…” statement and is not displayed by the PROC REPORT.

FINAL TABLE A Table of Cumulative Totals and Percent for Separate Groups

Table Two was expanded to include information for several years. This final two level table is shown in Figure 1. This same basic information could have been created by using the code for Table Two with a BY statement. This would however have given each group as a separate table rather than all groups in the same table.

Final Table and SAS CODE

The final table is given in Figure 1. This table is the same as Figure 2 except there is a separate table for each year. The code is shown below. The basic change is the addition of a year report variable, a COMPUTE BEFORE YEAR block, and the removal of .

Proc Report data=quarter nowd SPLIT="*" OUT=three;
TITLE1 "FIGURE 1";
TITLE2 "TABLE THREE: SALES BY QUARTER and YEAR TO DATE";
column year quarter
('SALES '(BY QUARTER sales pct)
("YEAR TO DATE " YTD cumptotal )
);
DEFINE year / group format=5. 'YEAR*' '*{(1)*' CENTER;
DEFINE Quarter / group format=8. 'QUARTER'* '*{(2)* CENTER;
DEFINE SALES / ANALYSIS FORMAT=DOLLAR8. SUM 'TOTAL+(3)';
DEFINE PCT / COMPUTED FORMAT=PERCENT8.2 'PERCENT+(4)';
DEFINE YTD / COMPUTED "TOTAL+(5)" FORMAT=....
```
THINGS TO NOTICE IN THE TABLE THREE CODE

Calculating cumulative totals and percent separately for different groups in a PROC REPORT table is almost the same as doing it for one group. The key is to initialize the cumulative total data variables at the start of each group and not to compute a grand total. Key features in the code are discussed below.

This COMPUTE BEFORE block makes the cumulative total for the year available before each quarter record is created.

Creates the data variable total and sets it equal to the value of sales.sum.

/ Creates the data variable cumtot and sets it equal to zero. The value of cumtot is set to zero every time there is a new value for year.

For each record the value of sales is added to cumtot to make cumtot the cumulative total for the year.

The report variable – ytd – is set equal to the data variable cumtot.

When the report line is for the yearly total, the value of _BREAK_ will be “year” and the sum of the cumulative totals and percent for the year are set to missing or blank because the answer is meaningless.

The cumulative percent is calculated for each quarter using the cumulative total for the year (cumtot) and the yearly total (total).

The quarterly percent is calculated by dividing the sum of sales for the quarter (sales.sum) by the total sales for the year (total).

This creates the output data set for PROC REPORT shown in Figure 5. As in table two, note that the table three has 15 rows of data and the output data set has 18 rows of data. The three extra rows of data in the output data set were generated by the COMPUTE BEFORE YEAR block at û and they are used by PROC REPORT to calculate the cumulative percent column.

CONCLUSION

PROC REPORT’s features of data set variables and the COMPUTE block allow users to create flexible tables that are not possible in PROC TABULATE. Knowledge and use of data variables and the COMPUTE BEFORE block allow PROC REPORT users to create more complex and more useful tables. With the additional capabilities added by Output Display System (ODS), even more complex tables can be produced in the future.

REFERENCES


APPENDIX

The code below generates the artificial SAS data set used to produce the examples in this paper. With this data set and included code you should be able to reproduce the tables.

Data quarter;
Do year=97 to 99;
  Do j=1 to 12;
    if j=1 then xx='1dec1997'd;
    QUARTER=QTR( intnx('month',xx,J) );
  Do n=1 to 100;
    sales= normal(123)*20 + 5;
    output;
  End;
End;
RUN;