Using the COMPUTE Block in PROC REPORT

Jack Hamilton, Kaiser Foundation Health Plan, Oakland, California

ABSTRACT

COMPUTE blocks add a great deal of power to PROC REPORT by allowing programmatic changes to be made for each individual data cell. This paper will describe the basics of PROC REPORT, and show how COMPUTE blocks may be used to tailor the contents and formatting of the output.

A completed version of this paper will be available after WUSS at

http://www.excursive.com/sas/

The expanded paper will contain more details on the inner workings of PROC REPORT, along with whatever I add as a result of suggestions at WUSS.

INTRODUCTION

This paper is intended to acquaint you with the basics of PROC REPORT, and in particular with some of the things you can do using compute blocks.

PROC REPORT combines features of PROC PRINT, PROC SUMMARY, PROC SORT, and the data step – it can sort and summarize data and perform calculations.

You can't do anything with PROC REPORT that you can't do with a combination of PROC SUMMARY, PROC SORT, and data steps, but PROC REPORT may be much easier to use. It particularly excels in two areas: the display of relational (or "normalized") data, and control of appearance through ODS.

This paper is a tutorial; rather than giving detailed instructions about each statement and option in PROC REPORT, I will present working examples of features you might want to use.

A SAMPLE DATA SET

In this paper I will use a subset of the SASHELP.PRDSALE data set which ships with SAS. It has a reduced number of variables and a much smaller number of observations.

There are several system options I like to use by default:

```
options msglevel=i errorcheck=strict nocenter;
```

I also use some settings which make it easier to include results in the paper:

```
options ls=80 ps=40;
options nodate nonumber;
options formchar="|----|+|---+=|-/\<>*";
```

To create the sample data set, I use PROC SQL, selecting only 1 out of every 75 observations:

```
proc sql;
  create table smallprod as
    select   country,
                  region,
                  prodtype,
                  product,
          actual label='' format=comma10.2,
          predict label='' format=comma10.2,
                  month
```

```
    from    sashelp.prdsale
    where    mod(monotonic(), 75) = 0
    order   by ranuni(94612);
quit;
```

Printing the output (with an old-fashioned PROC PRINT) shows us what the data set looks like:

```
title 'PROC PRINT';
proc print data=smallprod;
run;
```

prints:

```
PROC PRINT

Obs COUNTRY    REGION     PRODTYPE   PRODUCT         ACTUAL    PREDICT MONTH

  1 CANADA     EAST       FURNITURE  BED             983.00     851.00  Jun
  2 GERMANY    EAST       OFFICE     CHAIR           197.00     747.00  Mar
  3 U.S.A.     EAST       OFFICE     DESK            415.00     763.00  Dec
  4 CANADA     EAST       OFFICE     DESK            234.00     452.00  Sep
  5 CANADA     WEST       OFFICE     TABLE           778.00     231.00  Dec
  6 CANADA     WEST       OFFICE     CHAIR           838.00      98.00  Jun
  7 U.S.A.     EAST       FURNITURE  SOFA            662.00     566.00  Mar
  8 GERMANY    EAST       FURNITURE  BED             770.00     110.00  Sep
  9 U.S.A.     WEST       OFFICE     CHAIR           425.00     296.00  Mar
 10 GERMANY    WEST       OFFICE     DESK            875.00     890.00  Sep
 11 GERMANY    EAST       OFFICE     DESK            625.00     953.00  Dec
 12 GERMANY    WEST       OFFICE     TABLE           881.00     817.00  Dec
 13 U.S.A.     EAST       OFFICE     CHAIR           862.00      21.00  Jun
 14 U.S.A.     WEST       FURNITURE  BED             550.00     369.00  Jun
 15 CANADA     WEST       FURNITURE  SOFA            142.00     583.00  Mar
 16 U.S.A.     WEST       OFFICE     DESK            655.00     912.00  Sep
 17 CANADA     EAST       OFFICE     CHAIR           670.00     679.00  Mar
 18 U.S.A.     EAST       FURNITURE  BED             153.00      37.00  Sep
 19 GERMANY    WEST       FURNITURE  BED             468.00     576.00  Jun
```

The remaining examples in this paper will not show all of the detail lines unless they are necessary to express a point.

If we run PROC REPORT with virtually no options, we'll get similar (but not identical) output:

```
title 'Plain PROC REPORT';
proc report data=smallprod nowindows missing;
run;
```

```
Plain PROC REPORT

                         Product                                          Mon
  Country      Region     type       Product         ACTUAL     PREDICT    th
  CANADA       EAST       FURNITURE  BED             983.00      851.00  Jun
  GERMANY      EAST       OFFICE     CHAIR           197.00      747.00  Mar
  U.S.A.       EAST       OFFICE     DESK            415.00      763.00  Dec
<LINES DELETED>
  GERMANY      WEST       FURNITURE  BED             468.00      576.00  Jun
```

There's no OBS column, labels are used instead of variable names, and the Month label is split, but otherwise they're pretty similar.

SOME BASIC OPTIONS

There are some basic options I almost always use with PROC REPORT: NOWINDOWS, MISSING, HEADLINE, and HEADSKIP.

NOWINDOWS tells PROC REPORT not to go into interactive mode. This applies only if you're running SAS interactively. It doesn't hurt to specify it in batch.

MISSING is similar to the MISSING option in PROC SUMMARY, PROC MEANS, PROC TABULATE, and PROC FREQ. It tells SAS not to silently delete observations with missing values in their classification variables. For some reason, statisticians seem to think this is OK, but it's almost never appropriate for business or personal use – your numbers won't add up right. So always use the MISSING option, just in case your data have missing values. It's a good habit to get into.

HEADLINE tells PROC REPORT to print an underline below the column headers. HEADSKIP tells PROC REPORT to skip a line after the header. As we'll see later, there's another way to achieve this effect, this but is the easy way.

```
proc report data=smallprod
      nowindows missing headline headskip;
run;
```

prints:

```
                          Product                                                Mon
  Country      Region      type         Product          ACTUAL      PREDICT     th
  ------------------------------------------------------------------------------

  CANADA       EAST        FURNITURE    BED              983.00       851.00  Jun
  GERMANY      EAST        OFFICE       CHAIR            197.00       747.00  Mar
  U.S.A.       EAST        OFFICE       DESK             415.00       763.00  Dec
<LINES DELETED>
  GERMANY      WEST        FURNITURE    BED              468.00       576.00  Jun
```

THE COLUMN STATEMENT

The COLUMN statement is used to tell PROC REPORT which variables you want to print, and in what order. It's similar to the VAR statement in PROC PRINT.

```
proc report data=smallprod
      nowindows missing headline headskip;
  column country region product month predict actual;
run;
```

prints:

```
                                     Mon
  Country      Region      Product     th      PREDICT      ACTUAL
  ------------------------------------------------------------

  CANADA       EAST        BED         Jun      851.00      983.00
  GERMANY      EAST        CHAIR       Mar      747.00      197.00
  U.S.A.       EAST        DESK        Dec      763.00      415.00
<LINES DELETED>
  GERMANY      WEST        BED         Jun      576.00      468.00
```

THE DEFINE STATEMENT – ORDER

You tell PROC REPORT how to handle specific columns using the DEFINE statement. It has a number of options; one commonly used option is ORDER, which tells PROC REPORT to sort the data:

```
proc report data=smallprod
      nowindows missing headline headskip;
  column country region product month predict actual;
  define country / order;
  define region / order;
run;
```

prints:

```
                                      Mon
  Country      Region      Product      th      PREDICT      ACTUAL
  ------------------------------------------------------------------

  CANADA       EAST        BED          Jun      851.00      983.00
                           DESK         Sep      452.00      234.00
                           CHAIR        Mar      679.00      670.00
               WEST        TABLE        Dec      231.00      778.00
                           CHAIR        Jun       98.00      838.00
                           SOFA         Mar      583.00      142.00
  GERMANY      EAST        CHAIR        Mar      747.00      197.00
<LINES DELETED>
```

As you can see, it's possible to order by more than one column. Columns are ordered by their appearance in the COLUMN statement, not by the physical order of the DEFINE statements.

THE DEFINE STATEMENT – GROUP

The GROUP option on the DEFINE statement tells PROC REPORT that you want to create a SUMMARY report rather than a detail report, summarizing by the variables with DEFINE / ORDER:

```
proc report data=smallprod
         nowindows missing headline headskip;
   column country region product month predict actual;
   define country / group;
   define region / group;
run;
```

prints:

```
                                      Mon
  Country      Region      Product      th      PREDICT      ACTUAL
  ------------------------------------------------------------------

  CANADA       EAST        BED          Jun      851.00      983.00
                           DESK         Sep      452.00      234.00
                           CHAIR        Mar      679.00      670.00
               WEST        TABLE        Dec      231.00      778.00
                           CHAIR        Jun       98.00      838.00
                           SOFA         Mar      583.00      142.00
  GERMANY      EAST        CHAIR        Mar      747.00      197.00
<LINES DELETED>
```

But wait! That looks just like the previous output? What happened?

If you look in the log, you'll see these lines:

```
NOTE: Groups are not created because the usage of PRODUCT is DISPLAY.
```

What that's telling you is the product variable might have multiple values within each group. Since PROC REPORT has no way of knowing what you want displayed, it just gives up and doesn't group at all.

When this happens, you probably want to delete the problematic column:

```
proc report data=smallprod
      nowindows missing headline headskip;
  column country region predict actual;
  define country / group;
  define region / group;
run;
```

prints:

```
Country      Region          PREDICT      ACTUAL
-------------------------------------------------

CANADA       EAST          1,982.00    1,887.00
             WEST            912.00    1,758.00
GERMANY      EAST          1,810.00    1,592.00
             WEST          2,283.00    2,224.00
U.S.A.       EAST          1,387.00    2,092.00
             WEST          1,577.00    1,630.00
```

FORMATTED GROUPS

You can also apply a format to a grouped variable, and the report will be sorted and grouped by the formatted value rather than the internal value:

```
proc format;
  value $Continent
    'U.S.A.', 'CANADA' = 'NA'
    'GERMANY'          = 'EU';
run;
proc report data=smallprod
      nowindows missing headline;
  column country region predict actual;
  define country / group format=$continent. width=7;
  define region / group;
run;
```

prints:

```
Country  Region          PREDICT      ACTUAL
-----------------------------------------
EU       EAST          1,810.00    1,592.00
         WEST          2,283.00    2,224.00
NA       EAST          3,369.00    3,979.00
         WEST          2,489.00    3,388.00
```

OTHER STATISTICS

The most common statistic is probably SUM, but other statistics (MEAN, STD, MIN, MAX, and so forth) are also available and can be specified in the DEFINE statement (only one at a time):

```
proc report data=smallprod
      nowindows missing headline headskip;
  column country region predict actual;
  define country / group;
  define region / group;
  define predict / mean;
  define actual / mean;
run;
```

prints:

```
Country      Region          PREDICT      ACTUAL
-------------------------------------------------

CANADA       EAST            660.67       629.00
             WEST            304.00       586.00
GERMANY      EAST            603.33       530.67
             WEST            761.00       741.33
U.S.A.       EAST            346.75       523.00
             WEST            525.67       543.33
```

REPORT LEVEL SUMMARIES

You can use the RBREAK statement to create a summary line at the bottom of the report:

```
proc report data=smallprod
      nowindows missing headline;
  column country region predict actual;
  define country / group;
  define region / group;
  rbreak after / summarize skip ol;
run;
```

prints:

```
Country     Region          PREDICT      ACTUAL
-------------------------------------------------
CANADA      EAST          1,982.00    1,887.00
            WEST            912.00    1,758.00
GERMANY     EAST          1,810.00    1,592.00
            WEST          2,283.00    2,224.00
U.S.A.      EAST          1,387.00    2,092.00
            WEST          1,577.00    1,630.00
                        ----------  ----------
                          9,951.00   11,183.00
```

You can also use RBREAK BEFORE to print a summary at the beginning of the report, but this is less common.

GROUP LEVEL SUMMARIES

You can use the BREAK statement to create a summary after (or before) each summary level you have defined with GROUP options:

```
proc report data=smallprod
      nowindows missing headline;
  column country region predict actual;
  define country / group;
  define region / group;
  break after country / summarize skip ol;
run;
```

prints:

```
Country     Region          PREDICT      ACTUAL
-------------------------------------------------
CANADA      EAST          1,982.00    1,887.00
            WEST            912.00    1,758.00
----------                ----------  ----------
CANADA                    2,894.00    3,645.00

GERMANY     EAST          1,810.00    1,592.00
            WEST          2,283.00    2,224.00
----------                ----------  ----------
GERMANY                   4,093.00    3,816.00

U.S.A.      EAST          1,387.00    2,092.00
            WEST          1,577.00    1,630.00
----------                ----------  ----------
U.S.A.                    2,964.00    3,722.00
```

You can, of course, combine both BREAK and RBREAK:

```
proc report data=smallprod
      nowindows missing headline;
  column country region predict actual;
  define country / group;
  define region / group;
  break after country / summarize skip ol;
  rbreak after / summarize skip dol;
run;
```

prints:

```
Country     Region          PREDICT       ACTUAL
-------------------------------------------------
CANADA      EAST          1,982.00    1,887.00
            WEST            912.00    1,758.00
----------                ----------  ----------
CANADA                    2,894.00    3,645.00

GERMANY     EAST          1,810.00    1,592.00
            WEST          2,283.00    2,224.00
----------                ----------  ----------
GERMANY                   4,093.00    3,816.00

U.S.A.      EAST          1,387.00    2,092.00
            WEST          1,577.00    1,630.00
----------                ----------  ----------
U.S.A.                    2,964.00    3,722.00

                          ==========  ==========
                          9,951.00   11,183.00
```

ACROSS VARIABLES

Often, you want print values data values across the page, in columns, rather than down the page, in rows. The ACROSS option allows you to do this:

```
proc report data=smallprod
      nowindows missing headline headskip;
  column country region predict actual;
  define country / group;
  define region / across;
  rbreak after / summarize skip ol;
run;
```

prints:

```
                    Region
Country     EAST        WEST          PREDICT       ACTUAL
-------------------------------------------------------------

CANADA             3           3     2,894.00    3,645.00
GERMANY            3           3     4,093.00    3,816.00
U.S.A.             4           3     2,964.00    3,722.00
           ----------  ----------  ----------  ----------
                  10           9     9,951.00   11,183.00
```

The default statistic is N, the number of observations; in this case, the regions total to 19. If you want to calculate statistics for another column, use the COMMA operator:

```
proc report data=smallprod
      nowindows missing headline headskip;
  column country region, predict actual;
  define country / group;
  define region / across;
  rbreak after / summarize skip ol;
run;
```

prints:

```
                      Region
                    EAST        WEST
  Country         PREDICT     PREDICT      ACTUAL
  ------------------------------------------------

  CANADA         1,982.00      912.00    3,645.00
  GERMANY        1,810.00    2,283.00    3,816.00
  U.S.A.         1,387.00    1,577.00    3,722.00
                ----------  ----------  ----------
                 5,179.00    4,772.00   11,183.00
```

You can use parentheses for grouping to get several statistics for one across group:

```
proc report data=smallprod
      nowindows missing headline headskip;
  column country region, (predict actual);
  define country / group;
  define region / across;
  rbreak after / summarize skip ol;
run;
```

prints:

```
                                Region
                      EAST                      WEST
  Country         PREDICT      ACTUAL     PREDICT      ACTUAL
  ------------------------------------------------------------

  CANADA         1,982.00    1,887.00      912.00    1,758.00
  GERMANY        1,810.00    1,592.00    2,283.00    2,224.00
  U.S.A.         1,387.00    2,092.00    1,577.00    1,630.00
                ----------  ----------  ----------  ----------
                 5,179.00    5,571.00    4,772.00    5,612.00
```

You can also put across groups side by side:

```
proc report data=smallprod
      nowindows missing headline headskip;
  column country (region prodtype) , (predict);
  define country / group;
  define region / across;
  define prodtype / across;
  rbreak after / summarize skip ol;
run;
```

prints:

```
                        Region                Product type
                          EAST         WEST    FURNITURE      OFFICE
Country                PREDICT      PREDICT     PREDICT     PREDICT
------------------------------------------------------------------

CANADA                1,982.00       912.00    1,434.00    1,460.00
GERMANY               1,810.00     2,283.00      686.00    3,407.00
U.S.A.                1,387.00     1,577.00      972.00    1,992.00
                     ----------   ----------  ----------  ----------
                      5,179.00     4,772.00    3,092.00    6,859.00
```

Or nest them:

```
proc report data=smallprod
      nowindows missing headline headskip;
  column country (region , prodtype) , (predict);
  define country / group;
  define region / across;
  define prodtype / across;
  rbreak after / summarize skip ol;
run;
```

prints:

```
                                  Region
                        EAST                        WEST
                    Product type                Product type
                  FURNITURE      OFFICE    FURNITURE      OFFICE
Country             PREDICT     PREDICT     PREDICT     PREDICT
------------------------------------------------------------------

CANADA               851.00    1,131.00       583.00      329.00
GERMANY              110.00    1,700.00       576.00    1,707.00
U.S.A.               603.00      784.00       369.00    1,208.00
                    ----------  ----------  ----------  ----------
                     1,564.00    3,615.00    1,528.00    3,244.00
```

HEADER OPTIONS

At this point, there's a lot of text on the page, and it might be useful to draw some lines to help show the groupings:

```
proc report data=smallprod
      nowindows missing headline
      split='!';
  column country ('= Groups =' region prodtype), (predict);
  define country / group;
  define region / across '- Region -';
  define prodtype / across '- Product -!- Type -';
  rbreak after / summarize skip ol;
run;
```

prints:

```
                  ==================== Groups ====================
                                              ------ Product -------
                  ------- Region -------     -------- Type --------
                        EAST         WEST    FURNITURE      OFFICE
Country              PREDICT      PREDICT     PREDICT     PREDICT
------------------------------------------------------------------
CANADA             1,982.00       912.00    1,434.00    1,460.00
GERMANY            1,810.00     2,283.00      686.00    3,407.00
U.S.A.             1,387.00     1,577.00      972.00    1,992.00
```

```
----------  ----------  ----------  ----------
  5,179.00    4,772.00    3,092.00    6,859.00
```

COLUMN ALIASES

Column Aliases allow you to use a data set variable for more than one purpose:

```
proc report data=smallprod
      nowindows missing headline split='!';
  column country
         predict predict=predictmean
         actual actual=actualmean;
  define country / group;
  define actual / sum 'Actual!Total';
  define actualmean / mean 'Actual!Mean';
  define predict / sum 'Predict!Total';
  define predictmean / mean 'Predict!Mean';
  rbreak after / summarize skip ol;
run;
```

prints:

```
                Predict     Predict       Actual       Actual
Country           Total        Mean        Total         Mean
-------------------------------------------------------------
CANADA         2,894.00      482.33    3,645.00       607.50
GERMANY        4,093.00      682.17    3,816.00       636.00
U.S.A.         2,964.00      423.43    3,722.00       531.71
              ----------  ----------  ----------  ----------
               9,951.00      523.74   11,183.00       588.58
```

You can combine aliases and across variables to obtain subtotal columns plus a total column:

```
proc report data=smallprod
      nowindows missing headline split='!';
  column country
         region , actual actual=actualsum;
  define country / group;
  define region / across '- Region -';
  define actual / sum;
  define actualsum / sum 'Total!Actual';
  rbreak after / summarize skip ol;
run;
```

prints:

```
              ------- Region -------
                    EAST        WEST       Total
Country           ACTUAL      ACTUAL      Actual
-----------------------------------------------
CANADA         1,887.00    1,758.00    3,645.00
GERMANY        1,592.00    2,224.00    3,816.00
U.S.A.         2,092.00    1,630.00    3,722.00
              ----------  ----------  ----------
               5,571.00    5,612.00   11,183.00
```

FORMATTED OUTPUT

You can use the STYLE option to apply styles to ODS destinations:

```
title 'Applying STYLES';
proc report data=smallprod
        nowindows missing headline
        style(header)=[font_style=roman];
   column country region, (predict actual);
   define country / group;
   define region / across style=[font_face=courier];
   rbreak after / summarize skip ol style=[font_weight=bold];
run;
```

prints:

	Region			
	EAST		WEST	
Country	PREDICT	ACTUAL	PREDICT	ACTUAL
CANADA	1,982.00	1,887.00	912.00	1,758.00
GERMANY	1,810.00	1,592.00	2,283.00	2,224.00
U.S.A.	1,387.00	2,092.00	1,577.00	1,630.00
	5,179.00	**5,571.00**	**4,772.00**	**5,612.00**

This and the remaining styled reports use the HTML destination with the built-in style sasdocPrinter.

COMPUTED COLUMNS
Well, finally, I hear you saying.

Computed columns allow you to calculate the value of a column using a subset of the data step language. You specify the COMPUTED option on the DEFINE statement to tell PROC REPORT that you are using a computed column, and code a COMPUTE … ENDCOMP block (known as a compute block) to calculate the value of the variable:

```
proc report data=smallprod
        nowindows missing headline headskip pspace=1;
   column country region product month predict actual diff;
   define country / order;
   define region / order;
   define month / order 'Mon' order=internal;
   define predict / display;
   define actual / display;
   define diff / computed format=comma10.2;
   compute diff;
        diff = actual - predict;
   endcomp;
run;
```

prints:

```
   Country      Region       Product      Mon      PREDICT       ACTUAL         diff
   ------------------------------------------------------------------------------
   CANADA       EAST         CHAIR        Mar       679.00       670.00        -9.00
                             BED          Jun       851.00       983.00       132.00
                             DESK         Sep       452.00       234.00      -218.00
<LINES DELETED>
```

There's a very important restriction on variables in compute blocks: You can only use variables to the left of the current column in your calculations (you can assign a value to a variable to the right, but you can't read it).

Another important restriction is that unqualified columns must have usage DISPLAY. Here's what happens when you try to perform calculations on a numeric column with the default usage of ANALYSIS:

```
proc report data=smallprod
        nowindows missing headline headskip pspace=1;
   column country region product month predict actual diff;
   define country / order;
   define region / order;
   define month / order 'Mon' order=internal;
   define diff / computed format=comma10.2;
   compute diff;
      diff = actual - predict;
   endcomp;
run;
```

The log shows

```
NOTE: Variable predict is uninitialized.
NOTE: Variable actual is uninitialized.
NOTE: Variable predict is uninitialized.
NOTE: Variable actual is uninitialized.
NOTE: There were 19 observations read from the data set WORK.SMALLPROD.
```

And the DIFF column always contains 0.

If you want to perform calculations on an analysis variable, you must qualify the name with the name of a statistic:

```
proc report data=smallprod
      nowindows missing headline;
  column country region product month predict actual diff;
  define country / order;
  define region / order;
  define diff / computed format=comma10.2;
  compute diff;
       diff = actual.sum - predict.sum;
  endcomp;
  break after country / summarize skip ol;
run;
```

prints:

```
  Country      Region       Product       th      PREDICT       ACTUAL         diff
  ---------------------------------------------------------------------------------
  CANADA       EAST         BED          Jun       851.00       983.00       132.00
                            DESK         Sep       452.00       234.00      -218.00
                            CHAIR        Mar       679.00       670.00        -9.00
               WEST         TABLE        Dec       231.00       778.00       547.00
                            CHAIR        Jun        98.00       838.00       740.00
                            SOFA         Mar       583.00       142.00      -441.00
  ----------                             ---   ----------   ----------   ----------
  CANADA                                 Sep     2,894.00     3,645.00       751.00
```

PRINTING AN OBSERVATION NUMBER

Printing something like an observation number is an obvious thing for a long-time SAS user to want to do, and there's an obvious approach:

```
proc report data=smallprod
      nowindows missing headline;
  column Obs country region product month predict;
  define country / order;
  define region / order;
  define obs / computed;
  compute obs;
    obs + 1;
  endcomp;
run;
```

but unfortunately it prints:

```
                                                    Mon
        Obs  Country     Region      Product        th      PREDICT
  ------------------------------------------------------------------
          1  CANADA      EAST        BED           Jun       851.00
          1                          DESK          Sep       452.00
          1                          CHAIR         Mar       679.00
<LINES DELETED>
```

What happened?

It's a bit of a gotcha. PROC REPORT behaves a bit like the data step in that report variables (those listed in the COLUMN statement) are automatically reset to missing at the beginning of each observation. So Obs is always reset to missing, 1 is added, and the result is missing.

The workaround is to use a variable that's not named in the COLUMN statement to hold the value across observations. Such variables are retained by not reset:

```
proc report data=smallprod
      nowindows missing headline;
  column Obs country region product month predict;
  define country / order;
  define region / order;
  define obs / computed;
  compute obs;
    obsno + 1;
      obs = obsno;
  endcomp;
run;
```

prints:

```
                                                    Mon
        Obs  Country     Region      Product        th      PREDICT
  ------------------------------------------------------------------
          1  CANADA      EAST        BED           Jun       851.00
          2                          DESK          Sep       452.00
          3                          CHAIR         Mar       679.00
<LINES DELETED>
```

CREATING AN OUTPUT DATA SET

You can use the OUT= option on the PROC REPORT statement to create an output data set. The data set will have a variable for each column in the COLUMN statement, plus a special variable named _BREAK_:

```
proc report data=smallprod nowindows missing headline
      out=reportout;
  column country region predict actual;
  define country / group;
  define region / group;
  break after country / summarize skip ol;
  rbreak after / summarize skip dol;
run;

proc print data=reportout;
run;
```

prints:

Obs	COUNTRY	REGION	PREDICT	ACTUAL	_BREAK_
1	CANADA	EAST	1,982.00	1,887.00	
2	CANADA	WEST	912.00	1,758.00	
3	CANADA		2,894.00	3,645.00	COUNTRY
4	GERMANY	EAST	1,810.00	1,592.00	
5	GERMANY	WEST	2,283.00	2,224.00	
6	GERMANY		4,093.00	3,816.00	COUNTRY
7	U.S.A.	EAST	1,387.00	2,092.00	
8	U.S.A.	WEST	1,577.00	1,630.00	
9	U.S.A.		2,964.00	3,722.00	COUNTRY
10			9,951.00	11,183.00	_RBREAK_

USING THE _BREAK_ VARIABLE TO CHANGE SUMMARY LINES

The _BREAK_ variable is also available in COMPUTE blocks; you can use it to decide when to change the value of other variables (even though it appears to the right of all other variables in the output data set). Suppose you don't like the default text for summary lines; you can change it:

```
proc report data=smallprod nowindows missing headline;
   column country region predict actual;
   define country / group;
   define region / group;
   break after country / summarize skip ol;
   rbreak after / summarize skip dol;
   compute country;
      if _break_ = 'COUNTRY' then
         country = 'Subtotal';
      else if _break_ = '_RBREAK_' then
         country = 'GrandTotal';
   endcomp;
run;
```

prints:

Country	Region	PREDICT	ACTUAL
CANADA	EAST	1,982.00	1,887.00
	WEST	912.00	1,758.00
Subtotal		2,894.00	3,645.00
GERMANY	EAST	1,810.00	1,592.00
	WEST	2,283.00	2,224.00
Subtotal		4,093.00	3,816.00
U.S.A.	EAST	1,387.00	2,092.00
	WEST	1,577.00	1,630.00
Subtotal		2,964.00	3,722.00

```
==========                ==========  ==========
GrandTotal                  9,951.00   11,183.00
```

COMPUTING A RATIO

Suppose you want to compute the ratio between two summarized variables. You can do that by specifying the variable and statistic in the compute block:

```
proc report data=smallprod nowindows missing headline;
   column country region predict actual ratio;
   define country / group;
   define region / group;
   define ratio / computed format=6.2;
   break after country / summarize skip ol;
   rbreak after / summarize skip dol;
   compute ratio;
      ratio = actual.sum / predict.sum;
   endcomp;
run;
```

prints:

```
Country     Region          PREDICT      ACTUAL   ratio
-------------------------------------------------------
CANADA      EAST          1,982.00    1,887.00    0.95
            WEST            912.00    1,758.00    1.93
----------              ----------  ----------  ------
CANADA                    2,894.00    3,645.00    1.26

GERMANY     EAST          1,810.00    1,592.00    0.88
            WEST          2,283.00    2,224.00    0.97
----------              ----------  ----------  ------
GERMANY                   4,093.00    3,816.00    0.93

U.S.A.      EAST          1,387.00    2,092.00    1.51
            WEST          1,577.00    1,630.00    1.03
----------              ----------  ----------  ------
U.S.A.                    2,964.00    3,722.00    1.26

                        ==========  ==========  ======
                          9,951.00   11,183.00    1.12
```

Notice something important here: the ratios are correct in the summary lines. SAS didn't just add up the ratios from the country/region summary lines – it knew to use the country/region summaries in the country/region ratio calculations, and the grand total summaries in the grand summary ratio calculation. SAS actually maintains four sets of numbers – for the detail lines, groups, and report, plus page totals - and chooses the correct one to use each time it executes the compute block.

PRINTING A LINE

The LINE statement lets your print a text string of your choosing. The line statement is similar in syntax to the PUT statement (warning – but not identical):

```
proc report data=smallprod nowindows missing headline;
   column country region predict actual ratio;
   define country--region / group;
   define ratio / computed format=6.2;
   rbreak after / summarize skip dol;
   compute ratio;
      ratio = actual.sum / predict.sum;
   endcomp;
   compute after country;
      cratio = actual.sum / predict.sum;
      line @10 'Country Ratio is '
           @28 cratio 6.2;
   endcomp;
```

```
run;
```

prints:

```
Country     Region          PREDICT      ACTUAL   ratio
-------------------------------------------------------
CANADA      EAST          1,982.00    1,887.00    0.95
            WEST            912.00    1,758.00    1.93
       Country Ratio is     1.26
GERMANY     EAST          1,810.00    1,592.00    0.88
            WEST          2,283.00    2,224.00    0.97
       Country Ratio is     0.93
U.S.A.      EAST          1,387.00    2,092.00    1.51
            WEST          1,577.00    1,630.00    1.03
       Country Ratio is     1.26
                         ==========  ==========  ======
                          9,951.00   11,183.00    1.12
```

Warning: there are two big gotchas with the LINE statement. The first is that the LINE statement always executes, even if it's inside a false IF block. This means that you can't conditionally put out a blank line. You can, however, put the text you might want to print in a variable, and calculate the variable with an IF statement.

This doesn't do what you want:

```
compute after country;
   cratio = actual.sum / predict.sum;
   if ratio > 1 then
      line @10 'Country Ratio is '
           @28 cratio 6.2;
endcomp;
```

it prints:

```
Country     Region          PREDICT      ACTUAL   ratio
-------------------------------------------------------
CANADA      EAST          1,982.00    1,887.00    0.95
            WEST            912.00    1,758.00    1.93
       Country Ratio is     1.26
GERMANY     EAST          1,810.00    1,592.00    0.88
            WEST          2,283.00    2,224.00    0.97
       Country Ratio is     0.93
U.S.A.      EAST          1,387.00    2,092.00    1.51
            WEST          1,577.00    1,630.00    1.03
       Country Ratio is     1.26
                         ==========  ==========  ======
                          9,951.00   11,183.00    1.12
```

This:

```
compute after country;
   cratio = actual.sum / predict.sum;
   if ratio > 1 then
      linetext = 'Country Ratio is ' || put(cratio, 6.2);
   else
      linetext = ' ';
   line @10 linetext $50.;
endcomp;
```

prints:

```
Country     Region          PREDICT      ACTUAL   ratio
-------------------------------------------------------
CANADA      EAST          1,982.00    1,887.00    0.95
            WEST            912.00    1,758.00    1.93
       Country Ratio is    1.26
```

```
GERMANY      EAST          1,810.00    1,592.00    0.88
             WEST          2,283.00    2,224.00    0.97

U.S.A.       EAST          1,387.00    2,092.00    1.51
             WEST          1,577.00    1,630.00    1.03
        Country Ratio is   1.26
                          ==========  ==========  ======
                          9,951.00    11,183.00    1.12
```

Much closer, but you still have an extra blank line.

The second gotcha is that you can't align LINE output with columns by name; there's no way to say "put this in the same horizontal location as the X variable". With monospace output, you can just count the columns, but there's no way to line up proportional font output.

I expect that both of these problems will be addressed in a future release of SAS.

CONTROLLING APPEARANCE WITH CALL DEFINE

One of the big winning features of PROC REPORT is the cell-level control it gives you over output appearance through the Output Delivery System. Although it's possible to achieve the same level of control in data step reporting, it's much more complicated – if you can use PROC REPORT, whoever inherits your programs will thank you.

In addition to the static styles shown in an example above, PROC REPORT provides the CALL DEFINE statement, which lets you specify the appearance of individual rows and cells. In the example below, ratios below 1 are shown in red, and ratios outside a given range are in bold:

```
proc report data=smallprod nowindows missing headline;
   column country region predict actual ratio;
   define country--region / group;
   define ratio / computed format=6.2;
   break after country / summarize skip ol;
   rbreak after / summarize skip dol;
   compute ratio;
      ratio = actual.sum / predict.sum;
      if ratio < 1 then
         call define(_col_, 'style', 'style=[foreground=red]');
      if ratio < 0.9 or ratio > 1.1 then
         call define(_col_, 'style', 'style=[font_weight=bold]');
   endcomp;
run;
```

produces:

Country	Region	PREDICT	ACTUAL	ratio
CANADA	EAST	1,982.00	1,887.00	0.95
	WEST	912.00	1,758.00	**1.93**
CANADA		2,894.00	3,645.00	**1.26**
GERMANY	EAST	1,810.00	1,592.00	**0.88**
	WEST	2,283.00	2,224.00	0.97
GERMANY		4,093.00	3,816.00	0.93
U.S.A.	EAST	1,387.00	2,092.00	**1.51**
	WEST	1,577.00	1,630.00	1.03
U.S.A.		2,964.00	3,722.00	**1.26**
		9,951.00	11,183.00	**1.12**

You can also control the format of an entire row; this variation on the Observation Number code creates a grey background under every other row:

```
proc report data=smallprod
         nowindows missing headline ;
   column country region product month predict;
   define country / order;
   define region / order;
   compute region;
      obsno + 1;
      if mod(obsno, 2) = 0 then
         call define(_row_, 'style', 'style=[background=ltgray]');
   endcomp;
run;
```

prints:

Country	Region	Product	Month	PREDICT
CANADA	EAST	BED	Jun	851.00
		DESK	Sep	452.00
		CHAIR	Mar	679.00
	WEST	TABLE	Dec	231.00
		CHAIR	Jun	98.00
		SOFA	Mar	583.00

<LINES DELETED>

DYNAMICALLY CHANGING FORMATS

Someone on SAS-L wanted to know how to display a different number of decimal points for the same field on each line of a report; another variable contained the number of decimal points to display.

Here's some code to do that:

```
data test (keep=decimals value displayed);
   do i = 1 to 5;
      do decimals = 0 to 3;
         value = ranuni(95605) * 10;
         displayed = value;
         output;
      end;
   end;
   stop;
run;

proc report data=test nowindows missing;
   columns value decimals displayed;
   define _all_ / display;
   compute displayed;
      call define(_col_, 'format', '10.' || put(decimals, z1.0));
   endcomp;
run;
```

prints:

```
      VALUE   DECIMALS  DISPLAYED
  3.4648784          0          3
  3.3165982          1        3.3
  2.0278367          2       2.03
  8.0749122          3      8.075
  8.1445687          0          8
  7.6102304          1        7.6
```

```
   5.2384895           2        5.24
   3.8716015           3       3.872
   5.5787562           0           6
<LINES DELETED>
```

ACKNOWLEDGMENTS
I would like to thank Sandy McNeill and Ray Pass for their contributions to my understanding of PROC REPORT, Lauren Haworth and Eric Gebhart for helping me understand ODS, and Richard deVenezia, Ian Whitlock, and the gang at SAS-L for helping me with base SAS in general, and special thanks to Bill Viergever.

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

Jack Hamilton
jfh@alumni.stanford.org:
jack.hamiilton@kp.org
http://www.excursive.com/sas/