Paper HOW07

PROC REPORT Basics: Getting Started with the Primary Statements

Arthur L. Carpenter
California Occidental Consultants, Oceanside, California

ABSTRACT

The presentation of data is an essential part of virtually every study and there are a number of tools within SAS® that allow the user to create a large variety of charts, reports, and data summaries. PROC REPORT is a particularly powerful and valuable procedure that can be used in this process. It can be used to both summarize and display data, and is highly customizable and highly flexible.

Unfortunately for many of those just starting to learn PROC REPORT, the terms “customizable” and “flexible” often seem to be euphemisms for “hard to learn”. Fortunately PROC REPORT is NOT as hard to learn as it appears. All you really need to have in order to get started is a basic knowledge of a few primary statements.

In this introduction to PROC REPORT you will learn to use the PROC REPORT statement and a few of its key options. Several of the supporting statements, including COLUMN, DEFINE, BREAK, and RBREAK, and their primary options will also be covered.

KEYWORDS

PROC REPORT, COLUMN, DEFINE, BREAK, RBREAK, COMPUTE

INTRODUCTION

The syntax for PROC REPORT is quite different from most other base procedures. In most instances the PROC statement is followed by the COLUMN statement which lists the variables of interest. This is in turn followed by a DEFINE statement for each variable in the COLUMN statement. The basic syntax looks something like:

```
PROC REPORT DATA= datasetname <options>;
COLUMN variable list and column specifications;
DEFINE column / column usage and attributes;
COMPUTE column; compute block statements; ENDCOMP;
RUN;
```

The COLUMN statement is used to identify all variables used in the generation of the table. This statement is followed by the DEFINE statement which specifies how the column is to be used and what its attributes are to be. One DEFINE statement is used for each variable in the COLUMN statement.

Although not discussed in this paper, the COMPUTE statement is used to start the definition of a compute block. The compute block is terminated with a ENDCOMP. The compute block has a variety of uses including the creation of new columns and performance of column specific operations.

The following PROC step shows the code to create a simple REPORT table.

```
title1 'Using Proc REPORT';
title2 'Simple Report';
* Simple report;
proc report data=sasclass.clinics nofs;
  columns region lname fname wt;
  define region / display;
  define lname / display;
  define fname / display;
  define wt / display;
  run;
```

```
Using Proc REPORT
Simple Report

  re
  gi                first      weight
  on  last name     name    in pounds
  5   Rose          Mary          215
  6   Nolan         Terrie        187
  9   Tanner        Heidi         177
  2   Saunders      Liz           109
  4   Jackson       Ted           201
  5   Pope          Robert        158
  8   Olsen         June          158
  4   Maxim         Kurt          179
```

You can see that this report resembles the output from a PROC PRINT, however there are several distinct differences. These include:

- there is no OBS column

- variable labels are used instead of column names
- it is possible to calculate summary statistics and new columns with PROC REPORT
- column headers are adjusted to column width not the other way around
- the procedure is potentially interactive (turned off with the NOFS option)

USING THE COLUMN STATEMENT

The COLUMN statement is used not only to identify the variables of interest, but to also add headers and to group variables. The primary function of the COLUMN statement is to provide a list of variables for REPORT to operate against, and these variables are listed in the order (left to right) that they are to appear on the report.

In addition to listing the variables, you can do a number of other things on the COLUMN statement as well. For instance you can use a comma to attach a statistic to a variable. The default statistic is the SUM.

```
* Using the Comma;
proc report data=sasclass.clinics nofs;
column region lname fname wt,mean;
define region / display;
define lname / display;
define fname / display;
define wt / display;
title1 'Using Proc REPORT';
title2 'Using the Comma';
run;
```

```
Using Proc REPORT
Using the Comma

 re                            weight
 gi                  first   in pounds
 on   last name      name         mean
 5    Rose           Mary          215
 6    Nolan          Terrie        187
 9    Tanner         Heidi         177
 2    Saunders       Liz           109
 4    Jackson        Ted           201
 5    Pope           Robert        158
 8    Olsen          June          158
 4    Maxim          Kurt          179
```

Statistics include the same suite of statistics that are available through procedures such as MEANS, SUMMARY, and UNIVARIATE. Of course doing statistics on a single observation, as was done here, is generally silly. Later we will see how this technique becomes useful when we group observations with the DEFINE statement.

Parentheses can be used to create collections of variables or statistics. The following example calculates the mean, of one observation, for two different variables. Notice that the statistic is still associated with these variables through the use of the comma as in the previous example.

```
* Using parentheses;
proc report data=sasclass.clinics nofs;
column region lname fname (wt edu),mean;
define region / display;
define lname / display;
define fname / display;
define wt / display;
define edu / display;
title1 'Using Proc REPORT';
title2 'Using Parentheses to Form Groups';
run;
```

```
Using Proc REPORT
Using Parentheses to Form Groups

 re                            weight    years of
 gi                  first   in pounds  education
 on   last name      name         mean       mean
 5    Rose           Mary          215         13
 6    Nolan          Terrie        187         15
 9    Tanner         Heidi         177         12
 2    Saunders       Liz           109         12
 4    Jackson        Ted           201         18
 5    Pope           Robert        158         16
 8    Olsen          June          158         16
 4    Maxim          Kurt          179         13
```

DEFINING COLUMNS

The COLUMN statement is insufficient to provide total control of the appearance of individual columns and how they are to be used. This task falls to the DEFINE statement.

DEFINING TYPES OF COLUMNS
The define statement lists the name of the column to which it applies and the attributes (following a “/”) that are to be used. A primary attribute of the DEFINE statement is the ‘type’ of column. The type refers to how REPORT is to handle the variable. Each variable has a default type. However I have found that it is usually a good idea to include a DEFINE statement for each column even if all of its desired attributes are the defaults.

DEFINE types include:

• display	Shows or displays the value of the variable (default for character variables).
• group	Consolidate observations using this variable.
• analysis	Used in calculations with a statistic (default type for numeric variables).
• computed	Specifies a variable **not** on the incoming data set that is to be created in a compute block.
• order	Sorts the data and forms groups when summary statistics are requested.

- across   Used to create groups across the page rather than down the page.

USING DEFINE TYPE GROUP
GROUP is used for the consolidation of rows, usually for summary purposes. Notice that since we are grouping on on the variable REGION, we are summarizing across individuals so the variables LNAME and FNAME would not make sense and have been removed from the COLUMN statement.

```
proc report data=sasclass.clinics nofs;
column region (wt edu),mean;
define region / group;
define wt / display;
define edu / display;
title1 'Using Proc REPORT';
title2 'Define Type GROUP';
run;
```

The resulting table has one row for each unique value of REGION and the statistics (mean weight and mean years of education) now reflect a summary of all the patients within that region.

```
Using Proc REPORT
Define Type GROUP

  re     weight   years of
  gi  in pounds  education
  on       mean       mean
  1         195         10
  10  172.33333  13.666667
  2       107.8       13.6
  3       145.8       14.2
  4   159.14286         15
  5      157.75       14.5
  6         198         14
  7         151         15
  8         160         14
  9       187.8         12
```

USING DEFINE TYPE GROUP WITH DISPLAY
When GROUP is used with a DISPLAY variable, grouping occurs without consolidation. This means that we are once again summarizing across a single observation. This silly here but notice how the value of REGION is displayed - once at the start of each group.

```
proc report data=sasclass.clinics nofs;
column region sex wt,(n mean);
define region / group;
define sex / display;
define wt / analysis;
define edu / display;
title1 'Using Proc REPORT';
title2 'Using Display and GROUP Together';
run;
```

Notice that SEX is still in ‘INTERNAL’ order, and that the individual observations are displayed.

Although the variable EDU appears in a DEFINE statement, it is not used and does not appear in the report. Only variables in the COLUMN statement will appear in the report, and the 'extra' DEFINE statement is ignored (no errors are generated, however there is a warning written to the LOG).

```
Using Proc REPORT
Using Display and GROUP Together

      p
      a
      t
      i
      e
      n
      t
  re  s
  gi  e    weight in pounds
  on  x          n       mean
  1   M          1        195
      M          1        195
      M          1        195
      M          1        195
  10  F          1        163
      M          1        177
      M          1        177
      F          1        163
      M          1        177
      M          1        177
  2   F          1        105
      F          1        109
      F          1        115
      M          1        105
      M          1        105
```

USING ACROSS WITH GROUP

You can transpose rows into columns by using the ACROSS define type. In this example SEX is defined as an across variable. As a result a column will be formed for each distinct value taken on by SEX. In this case 'F' and 'M'.

```
* Types of the Define Statement;
proc report data=sasclass.clinics nofs;
column region sex wt,(n mean);
define region / group;
define sex / across;
define wt / analysis;
title1 'Using Proc REPORT';
title2 'Define Types GROUP and ACROSS';
run;
```

The ACROSS also necessarily groups within SEX. Consequently since REGION is also a grouping variable we will be grouping for SEX within REGION. Since we did not specify what we wanted to display under the column headings F and M we get the count (SUM is the default statistic) within each group.

```
Using Proc REPORT
Define Types GROUP and ACROSS

  re
  gi   pati    weight in pounds
  on   F  M           n        mean
  1    .  4           4         195
  10   2  4           6   172.33333
  2    6  4          10       107.8
  3    5  5          10       145.8
  4    4  *          14   159.14286
  5    5  3           8      157.75
  6    4  6          10         198
  7    .  4           4         151
  8    4  .           4         160
  9    2  8          10       187.8
```

The report itself, however, is a bit unclear. Although SEX is grouped within REGION, the variable WT is not grouped within SEX. Consequently the N and MEAN for WT is for the whole region and does not summarize for a specific value of SEX. Obviously this something that we may want to do and an example that summarizes within REGION and SEX is shown below.

OTHER DEFINE STATEMENT OPTIONS

A number of supplemental options can be used with the design statement to augment the display of the information. Some of these other options include:

•	format	Specifies how the information within this column is to be formatted.
•	width	Width of column, number of characters to allocate.
•	noprint	This column is not to be displayed.
•	flow	Text that is wider than the 'width=' option wraps within the block (split characters are supported).
•	*statistic*	Statistic to be calculated (analysis variables).

Several of these options have been applied in the following example. It is often easier to associate statistics with columns through the DEFINE statement, rather than the COLUMN statement. Notice that since EDU is an analysis variable and no statistic is specified, the SUM statistic is calculated.

The column SEX has been given a type of ACROSS, which causes a separate column to be displayed for each unique value of SEX. Since we have not specified what those columns are to contain, REPORT counts the number of males and females.

```
proc report data=sasclass.clinics nofs;
column region sex edu wt;
define region / group width=6;
define sex / across width=2;
define wt / analysis mean format=6.2;
define edu / analysis;
title1 'Using Proc REPORT';
title2 'Other Define Statement Options';
run;
```

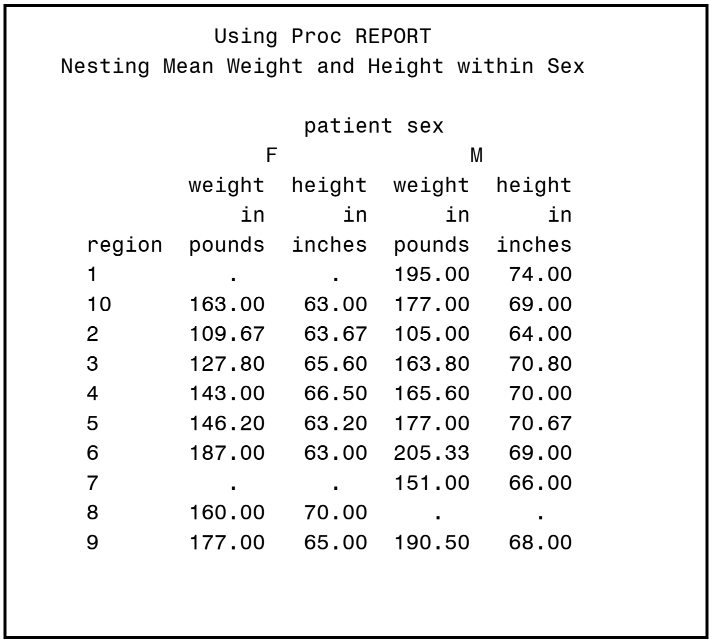
```
Using Proc REPORT
Other Define Statement Options

                                 weight
            patien    years of       in
  region   F    M    education   pounds
  1        .    4           40   195.00
  10       2    4           82   172.33
  2        6    4          136   107.80
  3        5    5          142   145.80
  4        4   10          210   159.14
  5        5    3          116   157.75
  6        4    6          140   198.00
  7        .    4           60   151.00
  8        4    .           56   160.00
  9        2    8          120   187.80
```

As in the previous example the summary statistics, in this case for WT and EDU, are for the whole region and do not apply to individual values of SEX. In the following example we do summarize across both variables by nesting WT and EDU within SEX.

NESTING VARIABLES
In the previous example, the columns resulting from the ACROSS variable (SEX) contain only the patient counts. What if we want to control what is to be displayed? We have already seen how parentheses can be used to create groups of variables, and how the comma can be used to form associations. Effectively these associations nest the items on the right of the comma within the item(s) on the left.

In the following example SEX is defined as an ACROSS variable. Within each value of SEX the mean height and weight is displayed. The association is formed by nesting the analysis variables within the ACROSS variable in the COLUMN statement.

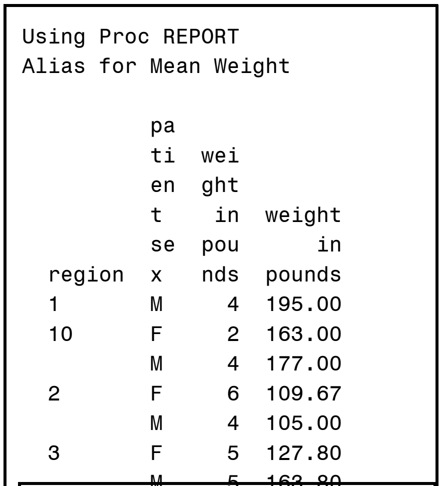
```
* Nesting variables;
proc report data=sasclass.clinics nofs;
column region sex,(wt ht);
define region / group width=6;
define sex / across width=3;
define wt / analysis mean format=6.2;
define ht / analysis mean format=6.2;
title1 'Using Proc REPORT';
title2 'Nesting Mean Weight and Height within Sex';
run;
```



```
                 Using Proc REPORT
    Nesting Mean Weight and Height within Sex

                        patient sex
                    F                 M
            weight  height  weight  height
                in      in      in      in
    region  pounds  inches  pounds  inches
    1            .       .  195.00   74.00
    10      163.00   63.00  177.00   69.00
    2       109.67   63.67  105.00   64.00
    3       127.80   65.60  163.80   70.80
    4       143.00   66.50  165.60   70.00
    5       146.20   63.20  177.00   70.67
    6       187.00   63.00  205.33   69.00
    7            .       .  151.00   66.00
    8       160.00   70.00       .       .
    9       177.00   65.00  190.50   68.00
```

USING VARIABLE ALIASES
When you need to use one variable in more than one way, it is possible to create an alias for that variable.

```
* Using Define Statement Aliases;
proc report data=sasclass.clinics nofs;
column region sex wt wt=wtmean;
define region / group width=6;
define sex / group width=2;
define wt / analysis n format=3.;
define wtmean / analysis mean format=6.2;
title1 'Using Proc REPORT';
title2 'Alias for Mean Weight';
run;
```



```
Using Proc REPORT
Alias for Mean Weight


           pa
           ti  wei
           en  ght
           t    in  weight
           se  pou      in
  region   x   nds  pounds
  1        M     4  195.00
  10       F     2  163.00
           M     4  177.00
  2        F     6  109.67
           M     4  105.00
  3        F     5  127.80
           M     5  163.80
```

In this example the alias is defined in the COLUMN by following the variable to be 'aliased' with an equal sign and the alias *e.g.* WTMEAN. The alias is then used in a DEFINE statement as if it was any other variable on the COLUMN statement. This allows us to show two different statistics associated with the same variable while giving us the freedom of two separate DEFINE statements.

ADDING TEXT

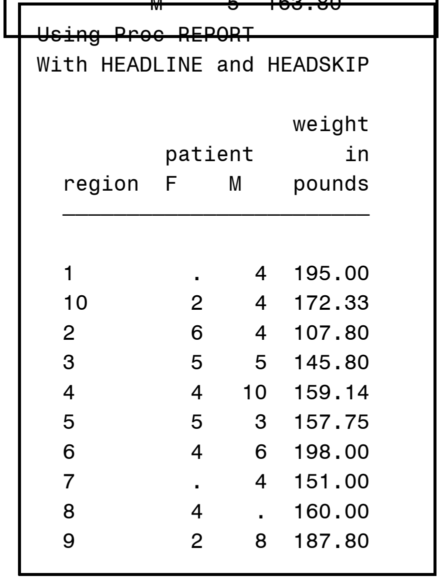
When no other action is taken, column headers are built from either the column label (when present) or the column name. Since text headers can substantially improve the appearance of the report, there is a great deal of flexibility on how text can be added to your report.

There are two major types of headers. Headers can be specified:

- which will span more than one report column.
- for each column individually.

Headers are specified as text strings and can be used on both the COLUMN and the DEFINE statements.

USING HEADLINE AND HEADSKIP OPTIONS
The HEADLINE and HEADSKIP options appear on the PROC REPORT statement,



```
Using Proc REPORT
With HEADLINE and HEADSKIP

                         weight
           patient           in
  region   F     M       pounds
  -----------------------------

  1         .     4      195.00
  10        2     4      172.33
  2         6     4      107.80
  3         5     5      145.80
  4         4    10      159.14
  5         5     3      157.75
  6         4     6      198.00
  7         .     4      151.00
  8         4     .      160.00
  9         2     8      187.80
```

and can be used to add both a space and an underline below the column header text.

```
* Using Headline and Headskip;
proc report data=sasclass.clinics
            nofs headline headskip;
column region sex wt;
define region / group width=6;
define sex / across width=3;
define wt / analysis mean format=6.2;
define edu / display;
title1 'Using Proc REPORT';
title2 'With HEADLINE and HEADSKIP';
run;
```

The HEADLINE option underlines the header area and the HEADSKIP option adds a space. Together they create a nice separation from the two areas of the report. There is not however much flexibility with the appearance or placement of these two separators. If you want more control you may want to use text in the COLUMN or DEFINE statements.

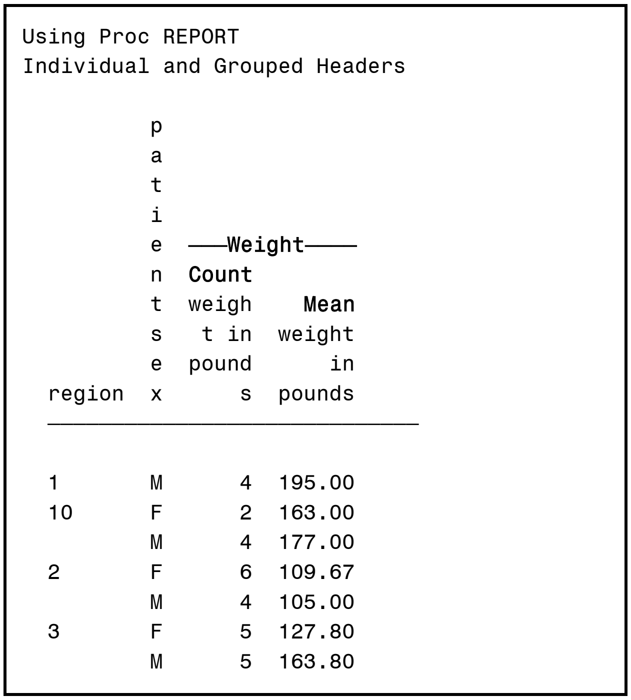
TEXT IN THE COLUMN STATEMENT
Text can be added to the header area by placing quoted strings in the COLUMN statement. Simply place the text string before the variable to which it is to apply.

When you want to create text to apply to more than one column, parentheses can be used to group text headers that will span columns. The spanning text precedes the list of variables in the group, and text that immediately precedes a variable name becomes text for that column in the report.

When you want to emphasize which text is associated with a variable or groups of variables, special characters, which PROC REPORT will expand to the appropriate width, can be included in the text string. Most commonly dashes are used to provide extended fill lines before and after the text, however other special characters, such as (=_ > <) can also be used to fill space. The special character should be the first and last characters of the string.

```
* Text headers in the COLUMN statement;
proc report data=sasclass.clinics
            nofs headline headskip;
column region sex ('-Weight-'
                   ('Count' wt)
                   ('Mean' wt=wtmean));
define region / group width=6;
define sex / group;
define wt / analysis n format=5.;
define wtmean / analysis mean format=6.2;
title1 'Using Proc REPORT';
title2 'Individual and Grouped Headers';
run;
```

Notice that the text specified in the COLUMN statement is 'added to' the other text, and does not replace the individual column headings. When you want to control the text associated with the individual column headings you will need to modify the DEFINE statement.



```
Using Proc REPORT
Individual and Grouped Headers


        p
        a
        t
        i
        e  ——Weight———
        n  Count
        t  weigh    Mean
        s   t in  weight
        e  pound      in
region  x      s  pounds
___________________________


1       M      4  195.00
10      F      2  163.00
        M      4  177.00
2       F      6  109.67
        M      4  105.00
3       F      5  127.80
        M      5  163.80
```

COLUMN SPECIFIC HEADERS THROUGH THE DEFINE STATEMENT

Text can also be added directly through the DEFINE statements. When you specify a text string in the DEFINE statement it replaces rather than augments the existing column headers.

```
* Text headers in the DEFINE statement;
proc report data=sasclass.clinics
            nofs headline headskip;
column region sex ('-Weight in Pounds-'
                   wt wt=wtmean);
define region / group width=6 'Region';
define sex / group width=6 'Gender' ;
define wt / analysis n format=5. 'Count';
define wtmean / analysis mean format=6.2 'Mean';
title1 'Using Proc REPORT';
title2 'DEFINE Statement Text';
run;
```

```
Using Proc REPORT
DEFINE Statement Text

                      —Weight in Pounds—
  Region  Gender  Count     Mean
  ----------------------------------

  1       M           4   195.00
  10      F           2   163.00
          M           4   177.00
  2       F           6   109.67
          M           4   105.00
```

In this example the label of each variable has been replaced with a text heading. We have now taken control of each of the portions of the header area.

CREATING BREAKS

BREAK OPTIONS

Rather than form one continuous vertical report, it is often advantageous to provide blank lines or white space between logical groups. Often additional summaries are desired at this time as well. Two statements (BREAK and RBREAK) are designed especially for providing these breaks and summaries.

A number of options are available with the BREAK and RBREAK statements. These include:

OL	inserts an overline
DOL	inserts a double overline
UL	inserts an underline
DUL	inserts a double underline
summarize	calculates an across group summary
skip	skips a line after the break
suppress	suppresses the printing or overlines and underlines around the group variables.

USING THE BREAK STATEMENT

The BREAK statement is used to create breaks and summaries within the report either before or after specific changes in some grouping variable. The statement identifies which grouping variable defines the groups and whether the break will be 'before' or 'after' each change in the grouping variable.

```
* Using the BREAK statement;
proc report data=sasclass.clinics nofs split='*';
column region sex ('Weight*in Pounds*--'
                   wt wt=wtmean);
define region / group width=6 'Region*--';
define sex / group width=6 'Gender*--' ;
define wt / analysis n format=5. 'Count*--';
define wtmean / analysis mean format=6.2 'Mean*--';
break after region / ol skip summarize suppress;
title1 'Using Proc REPORT';
title2 'BREAK After Each Region';
run;
```

```
Using Proc REPORT
BREAK After Each Region

                         Weight
                        in Pounds
                      -------------
  Region  Gender  Count     Mean
  ------  ------  -----   ------
  1       M           4   195.00
                  -----   ------
                      4   195.00

  10      F           2   163.00
          M           4   177.00
                  -----   ------
                      6   172.33

  2       F           6   109.67
          M           4   105.00
                  -----   ------
                     10   107.80
```

The BREAK statement in this example requests a break 'after' each value of the grouping variable REGION. The attributes of the break include:

- an overline (prints between the last value in the group and the summary)
- a space following the break (SKIP)
- a summary across all of the items within the group (SUMMARIZE)
- suppression of the reiteration of the grouping variables value on the summary line

USING THE RBREAK STATEMENT

The RBREAK statement creates the summary across the entire report and like the BREAK can be placed either before or after the report. The RBREAK statement does not identify a grouping variable because it is applied report wide and is independent of any groups that may or may not otherwise be created.

```
* Using the RBREAK statement;
proc report data=sasclass.clinics nofs split='*';
column region sex ('Weight*in Pounds*--'
                   wt wt=wtmean);
define region / group width=6 'Region*--';
define sex / group width=6 'Gender*--' ;
define wt / analysis n format=5. 'Count*--';
define wtmean / analysis mean format=6.2 'Mean*--';
break after region / ol skip summarize;
rbreak after / dol skip summarize;
title1 'Using Proc REPORT';
title2 'RBREAK At the End of the Report';
run;
```

In this example the report summary appears at the bottom of the report and includes a double over line and a space.

For the RBREAK summary, notice the difference between this and the previous example. In this case SUPPRESS is not included as an option on the BREAK statement, and the value of the grouping variable repeats.

```
           Using Proc REPORT
     RBREAK At the End of the Report

                           Weight
                          in Pounds
                       ______________
  Region  Gender       Count    Mean
  ______  ______       _____  ______
  1       M                4  195.00
  ______               _____  ______
  1                        4  195.00

  10      F                2  163.00
          M                4  177.00
  ______               _____  ______
  10                       6  172.33

  2       F                6  109.67
          M                4  105.00
  ______               _____  ______
  2                       10  107.80

    .... Portions of the report ....
          .... not shown ....

  ______               _____  ______
  9                       10  187.80

                       =====  ======
                          80  161.78
```

SUMMARY

PROC REPORT is a very powerful and flexible reporting tool. Although the statements used in this procedure are substantially different from most other SAS procedures, they are not as difficult to learn as it might first appear to the programmer new to this procedure.

When you are getting started, try to remember that the COLUMN statement is used to declare the variables of interest and that the DEFINE statements provide the details of how to use those variables.

Text can be used to dress up the report and can be declared either on the COLUMN or the DEFINE statements.

Breaks that make the report easier to read or that can additionally create summaries can be created either across the entire report (RBREAK) or for individual values of grouping variables (BREAK).

ABOUT THE AUTHOR

Art Carpenter's publications list includes three books, and numerous papers and posters presented at SUGI and other user group conferences. Art has been using SAS® since 1976 and has served in various leadership positions in local, regional, national, and international user groups. He is a SAS Certified Professional™ and through California Occidental Consultants he teaches SAS courses and provides contract SAS programming support nationwide.

AUTHOR CONTACT

Arthur L. Carpenter
California Occidental Consultants
P.O. Box 430
Oceanside, CA 92085-0430

(760) 945-0613
art@caloxy.com
www.caloxy.com

REFERENCES

Carpenter, Art, 2004, *Carpenter's Complete Guide to the SAS® Macro Language 2nd Edition,* Cary, NC: SAS Institute Inc.,2004.

TRADEMARK INFORMATION

SAS and SAS Certified Professional are registered trademarks of SAS Institute, Inc. in the USA and other countries.
® indicates USA registration.