The PRINT Procedure

ABSTRACT 461
INTRODUCTION 461
SPECIFICATIONS 462
  PROC PRINT Statement 462
  BY Statement 465
  ID Statement 465
  PAGEBY Statement 466
  SUM Statement 466
  SUMBY Statement 470
  VAR Statement 470
DETAILS 470
  Page Format 470
  Printed Output 471
EXAMPLES 471
  Example 1: Printing a Simple Report 471
  Example 2: Using the BY Statement 473
  Example 3: Using the PAGEBY Statement 474
  Example 4: Using the SUM Statement 476
  Example 5: Using the SUM Statement with a BY Statement 477
  Example 6: Using the SUMBY Statement 479
  Example 7: Using Identical BY and ID Statements 481

ABSTRACT

The PRINT procedure prints the observations in a SAS data set, using all or some of the variables. PROC PRINT can also print totals and subtotals for numeric variables.

INTRODUCTION

The PRINT procedure prints a listing of the values of some or all of the variables in a SAS data set. You can produce customized reports using procedure options and statements; for example, when you use a BY statement, PROC PRINT separates observations into groups defined by the BY variables. The PRINT procedure prints totals of the values of numeric variables when you use the SUM statement.
SPECIFICATIONS

The following statements control the PRINT procedure:

PROC PRINT <option-list>;
  VAR variable-list;
  ID variable-list;
  BY variable-list;
  PAGEBY BY-variable;
  SUMBY BY-variable;
  SUM variable-list;

PROC PRINT Statement

PROC PRINT <option-list>;

Table 27.1 summarizes the options supported by the PROC PRINT statement. Detailed explanations follow the table in alphabetic order:

Table 27.1  Summary of PROC PRINT Statement Options

<table>
<thead>
<tr>
<th>Class</th>
<th>Option</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specifying data set</td>
<td>DATA=</td>
<td>names the SAS data set to print</td>
</tr>
<tr>
<td>Formatting output</td>
<td>DOUBLE</td>
<td>double-spaces the printed output</td>
</tr>
<tr>
<td></td>
<td>NOOBS</td>
<td>suppresses the observation number in the output</td>
</tr>
<tr>
<td></td>
<td>UNIFORM</td>
<td>formats all pages uniformly</td>
</tr>
<tr>
<td></td>
<td>LABEL</td>
<td>uses variables’ labels as column headings</td>
</tr>
<tr>
<td></td>
<td>SPLIT=</td>
<td>splits labels used as column headings across multiple lines</td>
</tr>
<tr>
<td></td>
<td>N</td>
<td>prints the number of observations in the data set, in each BY group, or both</td>
</tr>
<tr>
<td></td>
<td>ROUND</td>
<td>rounds values to the number of decimal places specified for a variable in a FORMAT statement or, if you do not specify a format, to two decimal places</td>
</tr>
</tbody>
</table>

DATA= SAS-data-set
names the SAS data set to print. If you do not use the DATA= option, PROC PRINT uses the most recently created data set.

DOUBLE
D
double-spaces the printed output.
LABEL

uses variables' labels as column headings. You can specify labels in LABEL statements either in the DATA step that creates the data set or in the PROC PRINT step. See SAS Language: Reference, Version 6, First Edition for a description of the LABEL statement.

If you do not specify the LABEL option, or if a variable does not have a label, PROC PRINT uses the variable's name as the column heading. If you specify the LABEL option and at least one variable has a label, PROC PRINT prints all column headings horizontally. Therefore, using the LABEL option may increase the number of pages of output.

PROC PRINT splits labels if necessary in order to conserve space. Use the SPLIT= option to control where these splits occur.

The PRINT procedure uses labels for BY variables in the header it prints above each BY group, but it uses the variables' names when identifying subtotals and totals (see Output 27.9).

N

prints the number of observations in the data set at the end of the printed output.

If you use a BY statement with the N option, PROC PRINT displays the number of observations in each BY group but not the total number of observations in the data set. However, if you use a BY statement and a SUM statement with the N option, PROC PRINT displays the number of observations in each BY group and the total number of observations in the data set.

NOOBS

suppresses the observation number in the output. Use the NOOBS option when you do not use an ID statement but do not want to print the observation numbers. (The ID statement also suppresses the printing of observation numbers).

ROUND

R

rounds values to the number of decimal places specified for a variable in a FORMAT statement or, if you do not specify a format, to two decimal places. If you specify the ROUND option, the PRINT procedure rounds variables before summing them.

SPLIT='split-character'

S='split-character'

splits labels used as column headings across multiple lines where the split character appears. PROC PRINT does not print the split character.

You do not need to use both the LABEL and SPLIT= options because the SPLIT= option implies that labels are to be used. For example, the statements

    proc print data=class split='*';
    label x='This Is*a Label';
    run;

print this label for X:

    This Is
    a Label
You can use blanks in a label, and you can use multiple split characters within a label. For example, the following statements produce **Output 27.1**:

```plaintext
data test;
  input x $ aa;
  cards;
  aaa bbb ccc ddd ;
  run;

proc print data=test split='*';
  label x='This *Is*a Label';
  title 'Using Multiple Split Characters and Blanks in a Label';
  run;
```

**Output 27.1** Using Multiple Split Characters in a Label

<table>
<thead>
<tr>
<th>OBS</th>
<th>AAA</th>
<th>BBB</th>
<th>CCC</th>
<th>DDD</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>aaa</td>
<td>bbb</td>
<td>ccc</td>
<td>ddd</td>
</tr>
</tbody>
</table>

To print a column with no heading, use the split character as the variable's label. For example, these statements produce **Output 27.2**:

```plaintext
data test;
  input x $ aa;
  cards;
  AAA BBB CCC DDD ;
  run;

proc print data=test split='*';
  label x='*';
  title 'Printing a Column without a Heading';
  run;
```

**Output 27.2** Printing a Column without a Heading

<table>
<thead>
<tr>
<th>OBS</th>
<th>AAA</th>
<th>BBB</th>
<th>CCC</th>
<th>DDD</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>AAA</td>
<td>BBB</td>
<td>CCC</td>
<td>DDD</td>
</tr>
</tbody>
</table>
Note: PROC PRINT does not split labels of BY variables in the header preceding each BY group even if you specify the SPLIT= option. Instead, PROC PRINT uses the split character as part of the label.

UNIFORM

Formats all pages uniformly. When you specify the UNIFORM option, PROC PRINT normally reads the data set twice. The first time PROC PRINT reads the data set, it determines the width of the data so that it can print the data with a suitable field width. The PRINT procedure prints the data as it reads them the second time. However, if all the variables in the data set have formats that explicitly specify a field width (for example, BEST12., but not BEST.), PROC PRINT reads the data set only once.

BY Statement

BY variable-list;

When you use a BY statement with the PRINT procedure, the procedure prints a separate analysis for each BY group.

When a BY statement appears, the PRINT procedure expects the input data set to be sorted in order of the BY variables or to have an appropriate index. If your input data set is not sorted in ascending order, you can do one of the following:

- Use the SORT procedure with a similar BY statement to sort the data.
- If appropriate, use the BY statement option NOTSORTED or DESCENDING.
- Create an index on the BY variables you want to use. For more information on creating indexes and using the BY statement with indexed data sets, see Chapter 17, "The DATASETS Procedure."

The LABEL and SPLIT= options have no effect on BY variables when PROC PRINT places them in headers above each BY group. However, if the list of ID variables exactly matches the list of BY variables, PROC PRINT produces a specially formatted report, illustrated in Output 27.11. In this case, PROC PRINT honors the LABEL and SPLIT= options with BY variables because the same variables are also ID variables.

ID Statement

ID variable-list;

When you use an ID statement, the PRINT procedure uses the formatted values of the ID variables instead of observation numbers to identify observations in the output.

When an observation is too long to print on one line, the PRINT procedure prints the values of the ID variables at the beginning of every line containing data values for the observation. The ID list is too large if it does not allow enough room to print at least one variable that is not in the ID list.

If the list of ID variables exactly matches the list of BY variables, PROC PRINT produces a specially formatted report, illustrated in Output 27.11.
PAGEBY Statement

```lang-sas
PAGEBY BY-variable;
```

The PAGEBY statement begins printing a new page whenever the value of the specified BY variable changes or whenever the value of any BY variable listed before it in the BY statement changes.

The PAGEBY variable is one of the BY variables appearing in the BY statement in the PROC PRINT step. You must use a BY statement when you use the PAGEBY statement.

For example, the following SAS statements print a new page whenever the value of either X or Y changes, but not when the value of Z changes:

```sas
proc print;
  by x y z;
  pageby y;
run;
```

See Example 3 and Example 5 for illustrations of the use of the PAGEBY option.

SUM Statement

```lang-sas
SUM variable-list;
```

The SUM statement specifies variables whose values are to be totaled.

If you specify in a SUM statement a variable that is not listed in the VAR statement, PROC PRINT adds the variable to the VAR list.

When you use a BY statement and a SUM statement in the same PROC PRINT step, the PRINT procedure subtotals the SUM variables for each BY group containing more than one observation and totals them over all BY groups. Consider this example, which uses only one BY variable, X:

```sas
libname test 'SAS-data-library';

data test.a;
  input x y z;
  cards;
  1 1 1
  2 1 1
  2 1 2
  3 1 1
  3 1 2
  3 3 1
  3 3 2
  3 4 1
  3 4 2
  4 1 1
;
run;

proc print data=test.a;
  by x;
  sum z;
  title 'Sum Variable Z Whenever X Changes Value';
run;
```
These SAS statements create and print the data set TEST.A shown in Output 27.3. Notice the following features in the output:

1. The BY statement groups the data separately for each value of the BY variable, X, and prints a header for each BY group.
2. The combination of the BY statement and the SUM statement sums the values of the SUM variable, Z, for each BY group containing more than one observation.
3. The SUM statement sums the values of Z for the entire data set.

You do not need to sort the data set because the observations are already sorted by the values of X and Y.

Output 27.3 Subtotaling the SUM Variable for Each BY Group

<table>
<thead>
<tr>
<th>X=1</th>
<th>OBS</th>
<th>Y</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>X=2</th>
<th>OBS</th>
<th>Y</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>X</td>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>X=3</th>
<th>OBS</th>
<th>Y</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>4</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>X</td>
<td>9</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>X=4</th>
<th>OBS</th>
<th>Y</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>14</td>
<td></td>
<td>3</td>
</tr>
</tbody>
</table>

When you use a BY statement with multiple BY variables, PROC PRINT subtotals the SUM variables for each BY group containing more than one observation, just as it does if you use only one BY variable. However, it subtotals only those BY variables whose values change when the BY group changes.

Note: When the value of a BY variable changes, the SAS System considers that the values of all variables to its right in the BY statement change also.
Consider the following example, which uses two BY variables, X and Y:

```
libname test 'SAS-data-library';

proc print data=test.a;
  by x y;
  sum x;
  title 'Using the SUM statement with Multiple BY Variables';
run;
```

The output produced by these statements appears in Output 27.4. Notice the following features in the output:

1. The first and last subgroups in the report generate no subtotals because they contain only one observation.
2. When the BY group changes from
   
   \[
   X=2 \quad Y=1
   \]

   to

   \[
   X=3 \quad Y=1
   \]

   PROC PRINT subtotals the SUM variable for both X and Y because the values of both variables change.
3. When the BY group changes from
   
   \[
   X=3 \quad Y=1
   \]

   to

   \[
   X=3 \quad Y=3
   \]

   PROC PRINT subtotals the SUM variable for the subgroup \( Y=1 \) because the value of \( Y \) is changing from 1 to 3. Since the value of \( X \) does not change, PROC PRINT does not calculate subtotals for \( X \).
4. When the BY group changes from
   
   \[
   X=3 \quad Y=4
   \]

   to

   \[
   X=4 \quad Y=1
   \]

   PROC PRINT prints subtotals for the value of \( Z \) for the subgroup \( Y=4 \) and for the subgroup \( X=3 \), which consists of all observations where \( X=3 \) and \( Y \) is 1, 3, or 4.
5. At the very end of the report, PROC PRINT totals the values of \( Z \) over all BY groups.
### Output 27.4 Subtotalling the SUM Variable with Multiple RY Variables

Using the SUM statement with Multiple BY Variables

<table>
<thead>
<tr>
<th>OBS</th>
<th>X</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>9</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

For another example of using the SUM statement with multiple BY variables, see Example 5.
SUMBY Statement

SUMBY BY-variable;

The SUMBY statement prints subtotals for the specified BY variable whenever its value changes or whenever the value of any BY variable listed before it in the BY statement changes.

The SUMBY variable is a numeric variable appearing in the BY statement in the PROC PRINT step. You must use a BY statement when you use the SUMBY statement.

The SUMBY statement limits the printing of subtotals for BY groups. PROC PRINT still groups the data by all BY variables but prints subtotals only when the SUMBY variable changes. Without the SUMBY variable, the PRINT procedure prints subtotals for each BY group containing more than one observation.

If you do not use a SUM statement, PROC PRINT subtotals all the numeric variables in the data set except those listed in the ID and BY statements. If you do use a SUM statement, the PRINT procedure subtotals only the variables listed in that statement.

For example, the following SAS statements subtotal the variables A, B, and C when either X or Y changes value, but not when Z changes value:

```sas
proc print;
  sum a b c;
  by x y z;
  sumby y;
run;
```

See Example 6 for an illustration of the use of the SUMBY statement.

VAR Statement

VAR variable-list;

The VAR statement names the variables to print.

PROC PRINT prints the variables in the order you list them in the VAR statement. If you do not use the VAR statement, the PRINT procedure prints all variables in the data set.

DETAILS

Page Format

PROC PRINT uses an identical format for all observations on a page of output. First, it attempts to print observations on a single line. If it cannot do so, it splits the observations into two or more sections and prints the observation number or the ID variables at the beginning of each line. By default, spacing dictates whether the PRINT procedure prints column headings horizontally or vertically. If you use the LABEL option, the PRINT procedure prints all column headings horizontally, breaking up the report into additional sections if necessary.

The data width of an unformatted character variable is its length or the page size minus the length of the ID variable, whichever is less. The width of an unformatted numeric variable is 12.