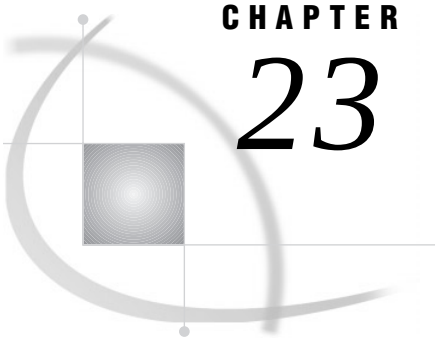


CHAPTER

23

BY-Group Processing in the DATA Step

Definitions

BY-group processing
is a method of processing observations from one or more SAS data sets that are grouped or ordered by values of one or more common variables. The most common use of BY-group processing in the DATA step is to combine two or more SAS data sets by using the BY statement with a SET, MERGE, MODIFY, or UPDATE statement.

BY variable
names a variable or variables by which the data set is sorted or indexed. All data sets must be ordered or indexed on the values of the BY variable if you use the SET, MERGE, or UPDATE statements. If you use MODIFY, data does not need to be ordered. However, your program might run more efficiently with ordered data. All data sets that are being combined must include one or more BY variables. The position of the BY variable in the observations does not matter.

BY value
is the value or formatted value of the BY variable.

BY group
includes all observations with the same BY value. If you use more than one variable in a BY statement, a BY group is a group of observations with the same

combination of values for these variables. Each BY group has a unique combination of values for the variables.

FIRST.variable and *LAST.variable*
are variables that SAS creates for each BY variable. SAS sets FIRST.*variable* when it is processing the first observation in a BY group, and sets LAST.*variable* when it is processing the last observation in a BY group. This allows you to take different actions, based on whether processing is starting for a new BY group or ending for a BY group. For more information, see "How the DATA Step Identifies BY Groups" on page 308.

For more information about BY-Group processing, see Chapter 24, "Reading, Combining, and Modifying SAS Data Sets," on page 315. See also *Combining and Modifying SAS Data Sets: Examples*.

Syntax

Use one of the following forms for BY-group processing:

BY *variable(s)*;

BY <GROUPFORMAT> <DESCENDING> *variable(s)* <NOTSORTED>;

where

variable
names each variable by which the data set is sorted or indexed.

Note: All data sets must be ordered or indexed on the values of the BY variable if you process them using the SET, MERGE, or UPDATE statements. If you use the MODIFY statement, your data does not need to be ordered. However, your program might run more efficiently with ordered data. All data sets that are being combined must include the BY variable or variables. The position of the BY variable in the observations does not matter. △

GROUPFORMAT
uses the formatted values, instead of the stored values, of the variable when you reference FIRST.*variable* and LAST.*variable* in a DATA step. If you have more that one variable in the BY group, GROUPFORMAT applies only to the variable that immediately follows it.

DESCENDING
indicates that the data sets are sorted in descending order (largest to smallest) by the variable that is specified. If you have more that one variable in the BY group, DESCENDING applies only to the variable that immediately follows it.

NOTSORTED
specifies that observations with the same BY value are grouped together but are not necessarily stored in alphabetical or numeric order.

For complete information about the BY statement, see *SAS Language Reference: Dictionary*.

Understanding BY Groups

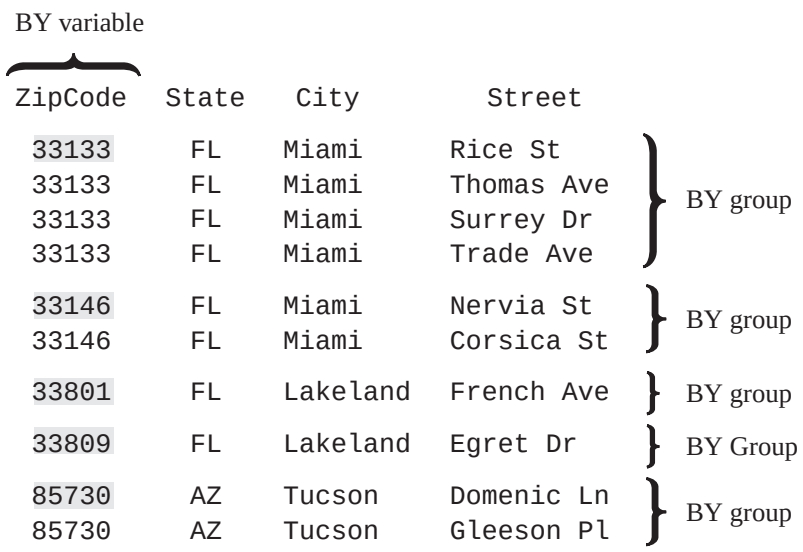
BY Groups with a Single BY Variable

The following figure represents the results of processing your data with the single BY variable ZipCode. The input SAS data set contains street names, cities, states, and ZIP codes that are arranged in an order that you can use with the following BY statement:

```
by ZipCode;
```

The figure shows five BY groups each containing the BY variable ZipCode. The data set is shown with the BY variable ZipCode printed on the left for easy reading, but the position of the BY variable in the observations does not matter.

Figure 23.1 BY Groups for the Single BY Variable ZipCode



The first BY group contains all observations with the smallest BY value, which is 33133; the second BY group contains all observations with the next smallest BY value, which is 33146, and so on.

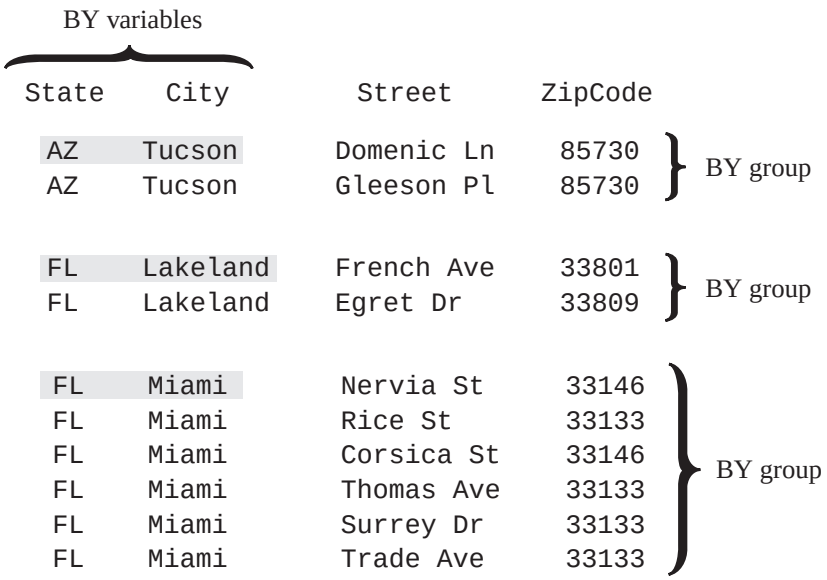
BY Groups with Multiple BY Variables

The following figure represents the results of processing your data with two BY variables, State and City. This example uses the same data set as in "BY Groups with a Single BY Variable" on page 305, and is arranged in an order that you can use with the following BY statement:

```
by State City;
```

The figure shows three BY groups. The data set is shown with the BY variables State and City printed on the left for easy reading, but the position of the BY variables in the observations does not matter.

Figure 23.2 BY Groups for the BY Variables State and City



The observations are arranged so that the observations for Arizona occur first. The observations within each value of State are arranged in order of the value of City. Each BY group has a unique combination of values for the variables State and City. For example, the BY value of the first BY group is **`AZ Tucson`**, and the BY value of the second BY group is **`FL Lakeland`**.

Invoking BY-Group Processing

You can invoke BY-group processing in both DATA steps and PROC steps by using a BY statement. For example, the following DATA step program uses the SET statement to combine observations from three SAS data sets by interleaving the files. The BY statement shows how the data is ordered.

```
data all_sales;
   set region1 region2 region3;
   by State City Zip;
   … more SAS statements …
run;
```

This section describes BY-group processing for the DATA step. For information on BY-group processing with procedures, see the *SAS Procedures Guide*.

Determining If the Data Requires Preprocessing

Before you process one or more SAS data sets using grouped or ordered data with the SET, MERGE, or UPDATE statements, you must check the data to determine if they

require preprocessing. They require no preprocessing if the observations in all of the data sets occur in one of the following patterns:

- ascending or descending numeric order
- ascending or descending character order
- not alphabetically or numerically ordered, but grouped in some way, such as by calendar month or by a formatted value.

If the observations are not in the order that you want, you must either sort the data set or create an index for it before using BY-group processing.

If you use the MODIFY statement in BY-group processing, you do not need to presort the input data. Presorting, however, can make processing more efficient and less costly.

You can use PROC SQL views in BY-group processing. For complete information, see the *SAS Guide to the SQL Procedure: Usage and Reference*.

SAS/ACCESS Users: If you use views or librefs, refer to the SAS/ACCESS documentation for your operating environment for information about using BY groups in your SAS programs.

Preprocessing Input Data

Sorting

You can use the SORT procedure to change the physical order of the observations in the data set. You can either replace the original data set, or create a new, sorted data set by using the OUT= option of the SORT procedure. In this example, PROC SORT rearranges the observations in the data set INFORMATION based on ascending values of the variables State and ZipCode, and replaces the original data set.

```
proc sort data=information;
   by State ZipCode;
run;
```

As a general rule, when you use PROC SORT, specify the variables in the BY statement in the same order that you plan to specify them in the BY statement in the DATA step. For a detailed description of the default sorting orders for numeric and character variables, see the SORT procedure in the *SAS Procedures Guide*.

Indexing

You can also ensure that observations are processed in ascending numeric or character order by creating an index based on one or more variables in the SAS data set. If you specify a BY statement in a DATA step, SAS looks for an appropriate index. If one exists, SAS automatically retrieves the observations from the data set in indexed order.

Note: Because indexes require additional resources to create and maintain, you should determine if their use significantly improves performance. Depending on the nature of the data in your SAS data set, using PROC SORT to order data values can be more advantageous than indexing. For an overview of indexes, see “SAS Indexes” on page 433. △

How the DATA Step Identifies BY Groups

In the DATA step, SAS identifies the beginning and end of each BY group by creating two temporary variables for each BY variable: FIRST.*variable* and LAST.*variable*. These temporary variables are available for DATA step programming but are not added to the output data set. Their values indicate whether an observation is

- the first one in a BY group
- the last one in a BY group
- neither the first nor the last one in a BY group
- both first and last, as is the case when there is only one observation in a BY group.

You can take actions conditionally, based on whether you are processing the first or the last observation of a BY group.

When an observation is the first in a BY group, SAS sets the value of the FIRST.*variable* to 1. For all other observations in the BY group, the value of the FIRST.*variable* is 0. Likewise, if an observation is the last in a BY group, SAS sets the value of LAST.*variable* to 1. For all other observations in the BY group, the value of LAST.*variable* is 0. If the observations are sorted by more than one BY variable, the FIRST.*variable* for each variable in the BY statement is set to 1 at the first occurrence of a new value for the variable.

This example shows how SAS uses the FIRST.*variable* and LAST.*variable* to flag the beginning and end of four BY groups. Six temporary variables are created within the program data vector. These variables can be used during the DATA step, but they do not become variables in the new data set.

In the figure that follows, observations in the SAS data set are arranged in an order that can be used with this BY statement:

```
by State City ZipCode;
```

SAS creates the following temporary variables: FIRST.State, LAST.State, FIRST.City, LAST.City, FIRST.ZipCode, and LAST.ZipCode.

Figure 23.3 FIRST. and LAST. Values for Four BY Groups

Observations in Four BY Groups				Corresponding FIRST. and LAST. Values					
State	City	ZipCode	Street	FIRST. State	LAST. State	FIRST. City	LAST. City	FIRST. Zipcode	LAST. Zipcode
AZ	Tucson	85730	Gleeson Pl	1	1	1	1	1	1
FL	Miami	33133	Rice St	1	0	1	0	1	0
FL	Miami	33133	Thomas Ave	0	0	0	0	0	0
FL	Miami	33133	Surrey Dr	0	0	0	0	0	1
FL	Miami	33146	Nervia St	0	0	0	0	1	0
FL	Miami	33146	Corsica St	0	1	0	1	0	1
OH	Miami	45056	Myrtle St	1	1	1	1	1	1

Processing BY-Groups in the DATA Step

Overview

The most common use of BY-group processing in the DATA step is to combine two or more SAS data sets using a BY statement with a SET, MERGE, MODIFY, or UPDATE statement. When processing these statements, SAS reads one observation at a time into the program data vector. With BY-group processing, SAS selects the observations from the data sets according to the values of the BY variable or variables. After processing all the observations from one BY group, SAS expects the next observation to be from the next BY group.

The BY statement modifies the action of the SET, MERGE, MODIFY, or UPDATE statement by controlling when the values in the program data vector are set to missing. During BY-group processing, SAS retains the values of variables until it has copied the last observation it finds for that BY group in any of the data sets. Without the BY statement, the SET statement sets variables to missing when it reads the last observation from any data set, and the MERGE statement does not set variables to missing after the DATA step starts reading observations into the program data vector.

You can process observations conditionally by using the subsetting IF or IF-THEN statements, or the SELECT statement, with the temporary variables FIRST.*variable* and LAST.*variable* (set up during BY-group processing). For example, you can use them to perform calculations for each BY group and to write an observation when the first or the last observation of a BY group has been read into the program data vector.

The following example computes annual payroll by department. It uses IF-THEN statements and the values of FIRST.*variable* and LAST.*variable* automatic variables to reset the value of PAYROLL to 0 at the beginning of each BY group and to write an observation after the last observation in a BY group is processed.

```
options pageno=1 nodate linesize=80 pagesize=60;

proc sort data=salaries out=temp;
   by Department;
data budget (keep=Department Payroll);
   set temp;
      by Department;
   if WageCategory='Salaried' then YearlyWage=WageRate*12;
   else if WageCategory='Hourly' then YearlyWage=WageRate*2000;
      /* Set FIRST.variable to 1 if this is a new department */
      /* in the BY group.                                    */
   if first.Department then Payroll=0;
   Payroll+YearlyWage;
      /* Set LAST.variable to 1 if this is the last department */
      /* in the current BY group.                               */
   if last.Department;
run;

proc print data=budget;
   format Payroll dollar10.;
   title 'Annual Payroll by Department';
run;
```

Output 23.1 Output from Conditional BY-Group Processing

```
                          Annual Payroll by Department                    1

                       Obs    Department       Payroll

                        1         BAD          $952,000
                        2         DDG          $448,000
                        3         PPD          $522,000
                        4         STD          $496,000
```

Data Grouped by Ascending Order

This example reads data that is in ASCENDING order and adds 13 to the inventory number. The input data must be sorted in ascending order by the values of the BY variable.

```
options pageno=1 nodate linesize=80 pagesize=60;

data current_inventory;
   set inventory;
   by Name;
   Number=Number+13;
run;

proc print data=current_inventory;
   title 'Current Inventory';
run;
```

Output 23.2 Output in Ascending Order of BY Variable

```
                            Current Inventory                             1

                          Obs    Name      Number

                           1     Fern        63
                           2     Hosta       31
                           3     Ivy         24
                           4     Moss        21
                           5     Rose        19
                           6     Vinca       16
```

Data Grouped by Descending Order

```
options pageno=1 nodate linesize=80 pagesize=60;

data current_inventory;
   set inventory;
   by descending Name;
   Number=Number+33;
run;

proc print data=current_inventory;
   title 'Number of Plants on Order';
run;
```

Output 23.3 Output in Decreasing Order of BY Variable

```
                          Number of Plants on Order                    1

                            Obs    Name      Number

                             1     Vinca       36
                             2     Rose        39
                             3     Moss        41
                             4     Ivy         44
                             5     Hosta       51
                             6     Fern        83
```

Data Not in Alphabetic or Numeric Order

In BY-group processing, you can use data arranged in an order other than alphabetic or numeric, such as by calendar month or by category. To do this, use the NOTSORTED option in a BY statement when you use a SET statement. The NOTSORTED option in the BY statement tells SAS that the data is not in alphabetic or numeric order, but that it is arranged in groups by the values of the BY variable. You cannot use the NOTSORTED option with the MERGE statement, the UPDATE statement, or when the SET statement lists more than one data set.

This example assumes that the data is grouped by MONTH. The subsetting IF statement conditionally writes an observation, based on the value of LAST.month. This DATA step writes an observation only after processing the last observation in each BY group.

```
data total_sale(drop=sales);
   set region.sales
   by month notsorted;
   total+sales;
   if last.month;
run;
```

Data Grouped by Formatted Values

Use the GROUPFORMAT option in the BY statement to ensure that

- formatted values are used to group observations when a FORMAT statement and a BY statement are used together in a DATA step
- the FIRST.*variable* and LAST.*variable* are assigned by the formatted values of the variable.

The GROUPFORMAT option is valid only in the DATA step that creates the SAS data set. It is particularly useful with user-defined formats.

This example uses the FORMAT procedure, the GROUPFORMAT option, and the FORMAT statement to create and print a simple data set. The input TEST data set is sorted by ascending values. The NEWTEST data set is arranged by the formatted values of the variable Score.

```
options pageno=1 nodate linesize=80 pagesize=60;

   /* Create a user-defined format */
proc format;
   value Range 1-2='Low'
               3-4='Medium'
               5-6='High';
run;

   /* Create the SAS data set */
data newtest;
   set test;
   by groupformat Score;
   format Score Range.;
run;

   /* Print using formatted values */
proc print data=newtest;
   title 'Score Categories';
   var Name Score;
   by Score;
run;
```

Output 23.4 Grouping Observations By Using Formatted Values

```
                                Score Categories                                1

---------------------------------- Score=Low ----------------------------------

                              Obs    Name    Score

                               1     Jon      Low


--------------------------------- Score=Medium ---------------------------------

                             Obs     Name      Score

                              2     Anthony   Medium
                              3     Miguel    Medium
                              4     Joseph    Medium


---------------------------------- Score=High ---------------------------------

                              Obs    Name    Score

                               5     Ian     High
                               6     Jan     High
```

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS Language Reference: Concepts*, Cary, NC: SAS Institute Inc., 1999. 554 pages.

SAS Language Reference: Concepts

ISBN 1–58025–441–1

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st printing, November 1999