

Generating SAS® Reports and the Collation Problem

Imelda C. Go, Richland County School District One, Columbia, SC

ABSTRACT

Default SAS output may produce reports that are not in a desired order. For example, let X and Y be two groups. PROC PRINT and PROC FREQ that use BY-group processing (for X and Y) will produce the following output sequence: PROC PRINT for X, PROC PRINT for Y, PROC FREQ for X, and PROC FREQ for Y. The desired output sequence is output from PROC PRINT for X, PROC FREQ for X, PROC PRINT for Y, and PROC FREQ for Y. Rearranging the output manually can achieve the desired order, but why waste precious time? This paper exemplifies a number of programming approaches to the collation problem. Using SAS macros to solve this problem can be advantageous. Macros generalize repetitive routines and can reduce the number of statements needed.

INTRODUCTION

The term *collate* refers to collecting or arranging (pages) in proper order. SAS provides the convenience of BY-group processing in many of its procedures. BY-group processing produces output for the different groups of interest, but the default output may not be in the desired order.

BY-GROUP PROCESSING

For example, consider a school district that has 50 schools. There is the need to compile a set of reports for each school. Each set of reports for a school consists of output from various SAS procedures generated *by school*. All data for the schools are in the data set called MASTER. A set of reports for a school can be obtained from output generated by the following procedures:

```
PROC SORT DATA=MASTER; BY SCHOOL;
PROC PRINT DATA=MASTER; BY SCHOOL; PAGEBY SCHOOL;
  VAR SSN LAST FIRST MI EXEMPT;
  TITLE 'Student Names and Exemption Status';
PROC MEANS DATA=MASTER; BY SCHOOL;
  VAR GPA;
  TITLE 'Average GPA';
PROC FREQ DATA=MASTER; BY SCHOOL;
  TABLES STANINE;
  TITLE 'Distribution of Reading Test Stanines';
```

The output will be generated in the following order: all the PROC PRINT output, all the PROC MEANS output, and then all the PROC FREQ output. All the output needed to make a set of reports for each school is available. However, the output from all the procedures is not sorted by school because the output is only sorted by school *within* each procedure.

Solving this problem using manual paper collation could be tolerable for a few schools or for a few SAS procedures. It becomes less tolerable as the number of schools increases and as more procedures with BY-group processing are involved. Some SAS programming can easily solve the collation problem. This may result in using

more programming statements, but it can eliminate human error that may result from manual paper collation.

The PROC MEANS statement above does not produce output *paged by* school. This is another aspect of the collation problem that can be solved by the following solutions.

SOLUTION 1

Generate reports using data sets that contain data from one school only. Suppose that there are 50 schools (e.g., School 1, School 2, ..., School 50). The schools are identified by a variable named SCHOOL. The value of SCHOOL is equal to the school's number (e.g., SCHOOL=1 for School 1, etc.). Create 50 data sets (e.g., data set SCHOOL1 contains data for School 1, etc.) where each data set contains data for only one school. Use each data set to generate reports for the corresponding school. If SAS macros are not used, the program code may lengthen considerably.

```
OPTIONS PAGENO=1;
DATA SCHOOL1;
  SET MASTER;
  IF SCHOOL=1;
PROC SORT DATA=SCHOOL1; BY SCHOOL;
PROC PRINT DATA=SCHOOL1; BY SCHOOL;
  VAR SSN LAST FIRST MI EXEMPT;
  TITLE 'Student Names and Exemption Status';
PROC MEANS DATA=SCHOOL1; BY SCHOOL;
  VAR GPA;
  TITLE 'Average GPA';
PROC FREQ DATA=SCHOOL1; BY SCHOOL;
  TABLES STANINE;
  TITLE 'Distribution of Reading Test Stanines';

:

OPTIONS PAGENO=1;
DATA SCHOOL2;
  SET MASTER;
  IF SCHOOL=2;
PROC SORT DATA=SCHOOL2; BY SCHOOL;
PROC PRINT DATA=SCHOOL2; BY SCHOOL;
  VAR SSN LAST FIRST MI EXEMPT;
  TITLE 'Student Names and Exemption Status';
PROC MEANS DATA=SCHOOL2; BY SCHOOL;
  VAR GPA;
  TITLE 'Average GPA';
PROC FREQ DATA=SCHOOL2; BY SCHOOL;
  TABLES STANINE;
  TITLE 'Distribution of Reading Test Stanines';

:

:

OPTIONS PAGENO=1;
DATA SCHOOL50;
  SET MASTER;
  IF SCHOOL=50;
PROC SORT DATA=SCHOOL50; BY SCHOOL;
PROC PRINT DATA=SCHOOL50; BY SCHOOL;
  VAR SSN LAST FIRST MI EXEMPT;
  TITLE 'Student Names and Exemption Status';
PROC MEANS DATA=SCHOOL50; BY SCHOOL;
  VAR GPA;
  TITLE 'Average GPA';
PROC FREQ DATA=SCHOOL50; BY SCHOOL;
  TABLES STANINE;
  TITLE 'Distribution of Reading Test Stanines';
```

The `pageno=1` option is useful for resetting the page numbering. All reports related to a particular school will start from page 1 and be numbered consecutively.

All the procedures in the example used BY-group processing even if each data set involved only one school. Using the BY statement is a convenient way of printing the value of variable `SCHOOL` at the top of the page.

SOLUTION 2

Use some SAS macro statements to generalize the repetitive code in Solution 1. The macro parameter values for `BEGIN` and `END` correspond to the numbers that represent the first and last schools that need reports.

```
%MACRO REPORTS (BEGIN, END);
%DO INDEX=&BEGIN %TO &END;
  OPTIONS PAGENO=1;
  DATA SCHOOL&INDEX;
    SET MASTER;
    IF SCHOOL=&INDEX;
  PROC PRINT DATA=SCHOOL&INDEX;
    BY SCHOOL;
    VAR SSN LAST FIRST MI EXEMPT;
    TITLE 'Student Names and Exemption Status';
  PROC MEANS DATA=SCHOOL&INDEX;
    BY SCHOOL;
    VAR GPA;
    TITLE 'Average GPA';
  PROC FREQ DATA=SCHOOL&INDEX;
    BY SCHOOL;
    TABLES STANINE;
    TITLE 'Distribution of Reading Test Stanines';
%END;
%MEND REPORTS;
**THE FOLLOWING STATEMENT GENERATES REPORTS FOR 50 SCHOOLS;
%REPORTS(1,50)
```

It is convenient to have the macro parameters `BEGIN` and `END` because there might be a need to generate reports for just one or a few schools. If a set of reports was needed for the 3rd school only, the statement `%REPORTS(3,3)` will generate the reports for just the 3rd school. If there is a need to generate the reports for the 14th through 25th schools, then the statement `%REPORTS(14,25)` will generate the reports for those schools.

SOLUTION 3

The first two solutions involved 50 different data sets to generate the reports for the schools. As an alternative, the `WHERE` statement can be used so that the only data set needed for the procedures is `MASTER`.

```
%MACRO REPORTS (BEGIN, END);
%DO INDEX=&BEGIN %TO &END;
  OPTIONS PAGENO=1;
  PROC PRINT DATA=MASTER;
    BY SCHOOL;
    VAR SSN LAST FIRST MI EXEMPT;
    WHERE SCHOOL=&INDEX;
    TITLE 'Student Names and Exemption Status';
  PROC MEANS DATA=MASTER;
    BY SCHOOL;
    VAR GPA;
    WHERE SCHOOL=&INDEX;
    TITLE 'Average GPA';
  PROC FREQ DATA=MASTER;
    BY SCHOOL;
    TABLES STANINE;
    WHERE SCHOOL=&INDEX;
    TITLE 'Distribution of Reading Test Stanines';
%END;
%MEND REPORTS;
```

This paper illustrates some solutions to the collation problem. It does not claim that these solutions provide the most ideal or the only way to deal with collation problems.

CONCLUSION

The collation problem can be solved using simple SAS programming. Solutions typically require adding more programming statements. The number of additional statements needed may vary and depend on whether or not macro statements were used. Macro statements generalize repetitive routines and can reduce the number of statements needed.

REFERENCES

SAS Institute Inc. (1990), *SAS Language Reference, Version 6, First Edition*, Cary, NC: SAS Institute Inc.

SAS Institute Inc. (1990), *SAS Programming Tips: A Guide to Efficient SAS Processing*, Cary, NC: SAS Institute, Inc.

TRADEMARK NOTICE

SAS is a registered trademark or trademark of the SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Imelda C. Go
Office of Research and Evaluation
Richland County School District One
1616 Richland St.
Columbia, SC 29201

Tel.: (803) 733-6079
Fax: (803) 929-3873
icgo@juno.com

