

Paper 320-2013

Reporting Tips for No Observations

W. Jodi Auyuen, BlueCross BlueShield of Minnesota

ABSTRACT

Good SAS® application designs include reporting accurate information, delivering in a timely manner, meeting business needs, and presenting in an easily grasped manner. We design reports to meet these goals and hopefully to cover potential questions. One of the frequently asked questions is: I used to receive a session, say tourists from Japan, why I don't see that session for the week of March 14, 2011? Even though we don't need to code "No visitors from Japan due to Tsunami on March 11, 2011", we could at least provide a generic message like "No data returned for this session." so business users won't wonder whether they missed the page or question the accuracy of the development work.

INTRODUCTION

SAS offers functions and capabilities for developers to design reports and to transform data into meaningful information for business users. If there is no data, developers can and certainly should make it clear on the report.

This paper will share a few reporting examples on two scenarios: one is when there are no observations in the data set, and the second scenario is when there are no observations in a reporting category (dimension or classification variable).

NO OBSERVATIONS IN THE DATASET¹

First of all, in order to know whether or not there are observations in the data set, we can count the rows by using PROC SQL to create a SAS macro variable, Nbr_Obs as in the following example.

```
PROC SQL NOPRINT;
    SELECT COUNT(*) INTO :Nbr_Obs
    FROM Japan;
QUIT;
```

However, the PROC SQL above requires all the observations from the incoming data set be read, which is time consuming for large data sets. It will be more efficient to use DICTIONARY tables, suggested by Mr. Art Carpenter².

```
PROC SQL NOPRINT;
    SELECT NOBS INTO:Nbr_Obs FROM DICTIONARY.TABLES
    WHERE LIBNAME='WORK' AND MEMNAME=UPCASE('Japan')
    ;
QUIT;
```

Next, when Nbr_Obs=0, developer could write a statement, "No Tourists from: Japan" to explain that there are no observations.

```
DATA NoObs;
    text="No Tourists from: Japan";
RUN;

PROC REPORT DATA=NoObs;
    COLUMN text;
    DEFINE text / ' ';
    TITLE1 "Tourists from: Japan";
RUN;
```

Suppose there are multiple data sets, one for each country with tourism. Using these two blocks of statements above, we can offer three options for report layout:

¹ Carpenter, Arthur L., 2013, "How Do I . . .?"

There is more than one way to solve that problem; Why continuing to learn is so important", presented at the 2013 SAS Global Forum Conference, Paper 029-2013.

² Carpenter Arthur, L., ., <http://support.sas.com/publishing/authors/carpenter.html>

1. To create separate PDF files for different Countries of Origin, so they can forward to different branch offices in those countries.
2. To create one PDF file, including all Countries of Origin separated by page breaks.
3. To create one Excel file, including all Countries of Origin separated by worksheets.

Option #1: To create separate PDF files for different Countries of Origin, so they can forward to different branch offices in those countries.

The macro code below, `SeparatePDF`, first reads in the dataset, and creates a macro variable, `Nbr_Obs`, to count number of observations. Based on that, if it is zero, it writes a text string: "No tourists from: Country" and uses PROC REPORT to write it out on the output file. Otherwise, if there is data, it uses PROC REPORT to list tourists' name and city on the output file. The ODS statements are inside the macro, so separate PDF files are created each time the macro is executed.

```
%MACRO SeparatePDF(myout) ;
    ODS LISTING CLOSE;
    ODS PDF file="/shrproj/sastemp/Tourists_&myout..pdf" style=sasweb notoc;

    PROC SQL NOPRINT;
        SELECT NOBS INTO:Nbr_Obs FROM DICTIONARY.TABLES
            WHERE LIBNAME='WORK' AND MEMNAME=UPCASE("&myout.");
    QUIT;

    %IF &Nbr_Obs=0 %THEN
        %DO;
            DATA NoObs;
                text="No Tourists from: &myout.";
            RUN;

            PROC REPORT DATA=NoObs;
                COLUMN text;
                DEFINE text / ' ';
                TITLE1 "Tourists from: &myout.";
            RUN;
        %END;
    %ELSE
        %DO;
            PROC REPORT DATA=&myout.;
                COLUMN name city;
                TITLE1 "Tourists from: &myout.";
            RUN;
        %END;
    ODS PDF CLOSE;
    QUIT;
%MEND SeparatePDF;

%SeparatePDF(Canada) ;
%SeparatePDF(NewZealand) ;
%SeparatePDF(Taiwan) ;
%SeparatePDF(Japan) ;
```

In the destination folder below, these PDF files are generated, shown in Figure 1. You can customize file names and destination paths also.



Figure 1. Output Files with Customized File Names

Option #2. To create one PDF file, including all Countries of Origin, separated by page breaks

In order to keep all output in one PDF, Option #2 opened the PDF destination first, then used the macro, OnePDF, to generate each output, and then closed the PDF destination after executing the macros %OnePDF for all the Countries of Origin.

```
ODS LISTING CLOSE;
ODS PDF file="/shrproj/sastemp/Tourists.PDF" style=sasweb ;
%MACRO OnePDF(myout);

    PROC SQL NOPRINT;
        SELECT NOBS INTO:Nbr_Obs FROM DICTIONARY.TABLES
            WHERE LIBNAME='WORK' AND MEMNAME=UPCASE("&myout.");
    QUIT;

    %IF &Nbr_Obs=0 %THEN
        %DO;
            DATA NoObs;
                text="No Tourists from: &myout.";
            RUN;

            PROC REPORT DATA=NoObs;
                COLUMN text;
                DEFINE text / ' ';
                TITLE1 "Tourists from: &myout.";
            RUN;
        %END;
    %ELSE
        %DO;
            PROC REPORT DATA=&myout.;
                COLUMN name city;
                TITLE1 "Tourists from: &myout.";
            RUN;
        %END;
%MEND OnePDF;

%OnePDF(Canada);
%OnePDF(NewZealand);
%OnePDF(Taiwan);
%OnePDF(Japan);
ODS PDF CLOSE;
QUIT;
```

You can create multiple PDF files using Option #2 with newfile option (highlighted in the ODS PDF statement below); however, SAS assigns generic file names, shown in Figure 2. If you prefer to customize file names, you might want to stay with Option #1, specifying the ODS statements within the macro.

```
ODS PDF file="/shrproj/sastemp/Tourists1.PDF" style=sasweb newfile=proc;
```



Figure 2. Output Files with Generic File Names.

Option #3. To create one Excel file, including all Countries of Origin separated by worksheets.

This third option uses the ExcelXP tagset to output different reports as different Excel worksheets in one Excel file. Pairing the `sheet_interval='none'` and `sheet_name` options gives absolute control over the worksheet tab names. By default the ExcelXP destination creates a new worksheet for every table created in the SAS output³.

In general, Excel allows business users to do further analysis in Excel, like sorting or filtering. And in comparison, PDF reports are easy to print feature and are used to discourage editing. However, the newer versions of Adobe do allow editing. We often choose PDF for reports which are sent out externally, versus choosing Excel for further analysis internally.

```
ODS LISTING CLOSE;
ODS tagsets.ExcelXP file="/shrproj/sastemp/Tourists.xls" style=sasweb options
(embedded_titles='yes' sheet_interval='none');

%MACRO OutputExcel(myout);

  PROC SQL NOPRINT;
    SELECT NOBS INTO:Nbr_Obs FROM DICTIONARY.TABLES
      WHERE LIBNAME='WORK' AND MEMNAME=UPCASE("&myout.");
  QUIT;

  %IF &Nbr_Obs=0 %THEN
    %DO;
  DATA NoObs;
    text="No Tourists from: &myout.";
  RUN;

  ODS tagsets.ExcelXP options(sheet_interval='none' sheet_name="&myout.");

  PROC REPORT DATA=NoObs;
    COLUMN text;
    DEFINE text / ' ' style(column)={cellwidth=6cm};
    TITLE1 "Tourists from: &myout.";
  RUN;
  %END;
%ELSE
  %DO;
  ODS tagsets.ExcelXP options(sheet_interval='none' sheet_name="&myout.");

  PROC REPORT DATA=&myout.;
    COLUMN name city;
    DEFINE name / display 'Name' style(column)={cellwidth=5cm};
    DEFINE city /display 'City' style(column)={cellwidth=5cm};
    TITLE1 "Tourists from: &myout.";
  RUN;
  %END;

%MEND OutputExcel;

%OutputExcel(Canada);
%OutputExcel(NewZealand);
%OutputExcel(Taiwan);
%OutputExcel(Japan);

ODS tagsets.ExcelXP CLOSE;
QUIT;
```

³ Gebhart, Eric, "The Devil Is in the Details: Styles, Tips and Tricks That Make Your Microsoft Excel Output Look Great!", SESUG Paper CS-050, Page 6, Available at <http://analytics.ncsu.edu/sesug/2008/CS-050.pdf>

NO OBSERVATIONS IN THE CATEGORY (DIMENSION)

Another scenario that might cause business users to question is when there are no observations in a category. By default, SAS will not display that category on the report. Here is one example: the customer wants to list their three classes by age groups: the first class is for students in the age range 0-10, the second class is for students in the age range 11-15, and the third class is for students at age 18.

```
PROC FORMAT;
    VALUE AgeFmt (NOTSORTED)
        LOW-10 = '0-10'
        11-15 = '11-15'
        18     = '18'
    ;
RUN;

DATA session;
    SET SASHELP.CLASS (KEEP=age name);
    ClassAge=PUT(age, AgeFmt.);
RUN;

PROC FREQ DATA=session;
    TABLES ClassAge/ NOPERCENT NOCUM OUT=OriginalList;
RUN;
```

The FREQ Procedure

ClassAge	Frequency
16	1
11-15	18

The developer sent this result to customers, and they asked, “I want to see my first class for students under 10 and the third class for students at age 18. I was not sure the missing of two categories was because of the bug in your design, or...” The developer answered, “There are no students for those two classes.” One way to resolve this confusion is we can modify the code to make it self-explanatory.

One approach is to add following PROC FORMAT. It reads “LABEL” from format library FMTLIB. And we can merge it with student enrollee list:

```
PROC FORMAT FMTLIB OUT=ClassAgeList (RENAME=(LABEL=ClassAge) KEEP=LABEL) NOPRINT;
RUN;

PROC SORT DATA=OriginalList;
    BY ClassAge;
RUN;

DATA CompleteList;
    MERGE ClassAgeList (IN=a) OriginalList (IN=b);
    BY ClassAge;
    IF b AND NOT a THEN ClassAge=CATX(' - ', 'NEW CLASS', ClassAge);
    IF a OR b THEN OUTPUT CompleteList;
RUN;

OPTIONS MISSING=0;

PROC REPORT data=CompleteList;
    COLUMN ClassAge Count;
    DEFINE ClassAge / DISPLAY 'Class Age Range';
    DEFINE count / DISPLAY 'Number Of Students';
RUN;
```

Class Age Range	Number Of Students
NEW CLASS - 16	1
0-10	0
11-15	18
18	0

Not only does it now display all the classes as requested, it also informs when there is one student falling outside the class age ranges.

CONCLUSION

The beauty of development is to be challenged to present reports in a self-explanatory way and easy for interpretation so to help business users to take actions or to make decisions quick. Even if there is no data, we would like the reports to reflect all possible levels.

REFERENCES

Carpenter, Arthur L., <http://support.sas.com/publishing/authors/carpenter.html>

Carpenter, Arthur L., 2013, ““How Do I . . . ?”

There is more than one way to solve that problem; Why continuing to learn is so important”, presented at the 2013 SAS Global Forum Conference, Paper 029-2013.

DelGobbo, Vincent, “Tips and Tricks for Creating Multi-Sheet Microsoft Excel Workbooks, the Easy Way with SAS”, Proceedings of the SAS Global Forum 2008, Available at <http://www2.sas.com/proceedings/forum2008/192-2008.pdf>

Gebhart, Eric, “The Devil Is in the Details: Styles, Tips and Tricks That Make Your Microsoft Excel Output Look Great!”, SESUG Paper CS-050, Available at <http://analytics.ncsu.edu/sesug/2008/CS-050.pdf>

Flavin, Justina M, Carpenter, Arthur L., “Is the Legend in your SAS/GRAPH Output Telling the Right Story?”, Available at http://www.caloxy.com/papers/54-c-legends_in_sasgraph_output.pdf

Cheng Alice M., “Is the Legend in your SAS/Graph Output Still Telling the Right Story?” Available at http://www.wuss.org/proceedings11/Papers_Cheng_A_74931.pdf

ACKNOWLEDGMENTS

I would like to thank Ann Schomisch, manager of Business Intelligence, who encouraged me to present. I also want to express my appreciation to Art Carpenter for his mentoring and valuable feedback on this paper. I learned good new tips from him. Many thanks are also due to Susan Trelstad and Barb Miller for reviewing this paper.

CONTACT INFORMATION

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

W. Jodi Auyuen
 BlueCross Blue Shield of Minnesota
 3535 Blue Cross Road, M325
 Eagan, Minnesota 55024
 E-mail: wuon_j_auyuen@bluecrossmn.com
 Web:

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. © indicates USA registration.

Other brand and product names are trademarks of their respective companies.