Paper CC-06

Summarizing Character Variables Using SAS® Proc Report

Priya Suresh – RTI International, RTP, NC

Elizabeth Heath – RTI International, RTP, NC

ABSTRACT

The SAS report procedure, Proc Report, is feature rich and is ideal for summarizing numerical values and flagging data outliers. Sometimes the values to be summarized are character variables, such as locations or names. In this paper, we will present a nifty technique to summarize character variables using SAS Proc Report, show how to flag outlier values, and point out a few idiosyncrasies of Proc Report.

INTRODUCTION

SAS Proc Report makes it easy to provide summary values and other statistics that are very useful for managing data collection. The output can include information such as the total number of completed cases broken out by project specific grouping categories. There are occasions where additional information needs to be "summarized" to assist in managing data collection, such as which person owns the majority of the cases within a data collection region, but the variables that contain such information are usually of character data types. This paper provides a nifty technique using SAS Proc Report to provide summary information even for character variables.

This paper is intended for SAS users who are familiar with the breakout summary row and compute block in Proc Report. For a good introduction to Proc Report syntax, and to learn about the breakout summary row and compute block, see the papers by David Lewandowski and Arthur L. Carpenter that are listed in the References section in this paper.

SAMPLE DATA

A survey is conducted in selected counties within two states. Each county has a specific number of cases to be worked. Counties within the same state are grouped together and assigned a StateID. Each county is assigned to a caseworker based on their caseload and familiarity with the county. Data collection is summarized daily, for each county and state, as shown in Exhibit 1, with the number of cases assigned in column NumCases, counts per type of current case status in columns Ineligibles through Pending, and percentage of eligible cases that have been completed in column CompRate.

SESUG Caseworker Workload Report

stateID	StateAbbr	Username	county	Status	StatusDesc	NumCases	Ineligibles	Eligibles	Completes	NonResponse	Pending	CompRate	Comments
S037	NC	PSuresh	Wake	765	Final Non-Interview	3	0	3	0	3	0	0	
	NC	PSuresh	Orange	763	Final Complete	1	0	1	1	0	0	100	
	NC	PSuresh	Durham	765	Final Non-Interview	1	0	1	0	1	0	0	
S037		***PSuresh***				***5***	***0***	***5***	***1***	***4***	***0***		***SubTotal***
S051	VA	EHeath	Louisa	763	Final - Complete	66	37	29	29	0	0	100	Cases GT 49
	VA	EHeath	Caroline	763	Final - Complete	1	0	1	1	0	0	100	
	VA	JDoe	Lee	768	Final - Partial Complete	1	0	1	1	0	0	100	
S051		***EHeath***				***68***	***37***	***31***	***31***	***0***	***0***		***Cases GT 49***

Exhibit 1: Sample Report for illustrative purposes

BREAK STATEMENT

In order to display the summary statistics for each state, the syntax for the break statement is as follows

break after StateID /ol summarize skip style(SUMMARY)=[background='CXCC99FF'. font_weight=bold];

STATISTIC FOR A NUMERIC VARIABLE

Proc Report provides techniques for summarizing the counts of numeric variables such as total number of cases and current case status. For example, to display the sum of the ineligibles across all counties within the state in the breakout summary line, one would use the following standard syntax:

define ineligibles / analysis **sum;**

COMPUTE STATEMENT

Proc Report's compute statement provides a way to highlight any anomalous characteristic of the data. For example, one reporting requirement for Exhibit 1 was to have the phrase "Cases gt 49' in the comments field if the county had more than 49 cases. In the code snippet below, the logic in the compute statement block provides the syntax to populate the Comments field.

```
/*
```

- with the phrase 'Cases gt 49' when the number of cases is more than 49,
- as blank if the number of cases is less than 49, or
- with the phrase 'SubTotal' if it is a summary row and the number of cases is less than 49.

```
*/
Compute comments/character length=20;
    If   NumCases.sum >  49 then  /* currently column 7 is # of cases, alternatively you can use _c7_ > 49  */
            comments='Cases GT 49';
    else if  _break_=' ' then
            comments=' ';
    else
            comments='SubTotal';
```

These *idiosyncrasies* are noteworthy.

1. The fields to the *left* of the Comments field can be referred to by column number or by variable name. *The fields to the right cannot be referred to by variable names but only by column numbers.*
2. If the variable is an analysis variable, you have to refer to it with a compound name, meaning, variablename.statistic. In this example, we use NumCases.sum. Alternately, you could just refer to it by its column number, as _c7_. If you try to use _c7_.sum, the code will not work.

NIFTY TECHNIQUE FOR SUMMARIZING A CHARACTER VARIABLE

Another reporting requirement for Exhibit 1 was to show the name of the caseworker with the most number of counties within a state in the summary line. Remember that the caseworker name is the character variable Username and one cannot compute a "statistic" for a character variable.

How does one achieve this?

1) Recode the caseworker names to a hidden variable that is numeric, let's call it Userid. The noprint option makes it a hidden variable.

```
define Userid /analysis mode noprint;
```

2) Generate a format definition for Userid; let's call it UserF.

```
proc format;
  value UserF
```

```
1 = "PSuresh"
2 = "EHeath"
3 = "JDoe"
;
```

3) Use the “Mode” statistic for that variable to identify the caseworker who has the majority of the cases within the state.

```
define Userid /analysis mode noprint;
```

4) Add the field Userid to the list of variables. Note that it is in column 14 *before the Comments field*.

```
Col     StateID StateAbbr Username County Status
        StatusDesc NumCases Ineligibles Eligibles Completes
        NonResponse Pending CompRate Userid Comments;
```

5) In the compute block of the comment field, use the UserF format to populate the ‘Caseworker assigned’ column for the summary row.

```
if upcase(_break_) eq upcase('StateID') then
        username = put ( Userid.mode, userF.);
```

Please note the following:

1. The Userid column has to be to the left of the comments column because Comments is a computed variable and the mode of the userid will not be known otherwise. (Idiosyncratic behavior.)
2. In the break statement, _break_ has the value “stateID”. This is one of the exceptions in SAS code, where it is case-sensitive on the break statement. In most other places its case-insensitive – i.e. we can type it in as STateId or stateid, and the code works, but in the break statement check you have to check it exactly like it was defined. In order to compare the values, we make it case-insensitive by capitalizing both _break_ and the string to be checked. (Idiosyncratic behavior.)
3. In the compute block, the specific statistic for a variable is referred as *variableName.Statistic . Thus, the syntax, Userid.mode, determines the mode of the variable userID.*

Voila! The summary column has the name of the caseworker who has the majority of the counties within the state; if nobody has a majority, then it is left blank or missing.

In Exhibit 1, caseworker PSuresh worked all of the NC counties and the caseworker name PSuresh appears in the summary row. Because caseworker EHeath has two counties and JDoe has one county, EHeath appears in the summary for VA. In effect, we tri*cked Proc Report to summarize a character variable with this technique.*

FULL SYNTAX FOR PROC REPORT

```
proc Report data=InDSN ;
        Col StateID StateAbbr Username County Status
                StatusDesc NumCases Ineligibles Eligibles Completes
                NonResponse Pending CompRate Userid Comments;
         define StateID /group;
         define StateAbbr /display;
         define Username /display;
         define County /display ;
         define Status /display ;
         define StatusDesc /display ;
         define NumCases /analysis sum;
        define Ineligibles /analysis sum;
        define Eligibles /analysis sum;
        define Completes /analysis sum;
        define NonResponse /analysis sum;
        define Pending /analysis sum;
        define CompRate /display ;

        define Userid /analysis mode noprint;
        define Comments/'Comments' computed;

        break after StateId/ol summarize   skip style(SUMMARY)=[background=CXCC99FF  font_weight=bold] ;

Compute comments/character length=20;
        if NumCases.sum > 49 then
                comments= ' Cases GT 49';
        else if  _break_=' '  then
                Comments=' ';
        else
                comments='SubTotal';

        if upcase(_break_) eq upcase('StateID') then        /* originally this variable was defined as stateID so it is
being upcased to match the value of this in the break statement */
                username =  put ( Userid.mode, userF.);

endcomp;
run;
```

CONCLUSION

The SAS Proc Report can be used to summarize not only numeric variables but also character variables with some creative manipulation of the data in the compute block of the procedure.

REFERENCES

Lewandowski, David, 2008, 079-2008 A Step-by-Step Introduction to PROC REPORT, *Proceedings of the SAS Global Forum Conference*, San Antonio, TX, March 16-19, 2008.

Carpenter, Arthur L., 2012, 242-2012 PROC REPORT Basics: Getting Started with the Primary Statements, *Proceedings of the SAS Global Forum Conference*, Orlando, FL, April 22-25, 2012.

RECOMMENDED READING

- Base SAS Procedures Guide

CONTACT INFORMATION

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

Priya Suresh
Phone: (919) 541-7428
E-mail: psoh@rti.org

Elizabeth Heath
Phone: (919) 485-2786
E-mail: eah@rti.org

Our mailing address:
RTI International
PO Box 12194
RTP NC 27709-2194