SAS Macro Program to Prepare Data Documentation with Tailored Descriptions of Various Variable Types

Keiko I. Powers, Ph.D., J. D. Power and Associates, Agoura Hills, CA

ABSTRACT

Knowing your data is an essential part of data analyses. However, we often find it time-consuming to grasp the detailed information of each variable, particularly when the data are large and contain different types of variables, e.g., character or numeric. I have written a SAS macro program that automatically detects the variable types and provides a tailored summary suitable for each variable. For example, if the variable contains addresses or phone numbers, the program lists the first 50 cases, which is sufficient to understand the variable format. If the variable contains zip codes or state names, the program provides frequencies of each category. If the variable is continuous and numeric, e.g., sales volumes, frequencies based on 25 sub-ranges are provided. A one-time execution of the program provides documentation that contains summaries for all the variables in the data. The documentation is a useful tool for quickly identifying appropriate statistical analyses for each variable.

INTRODUCTION

Statistical analysts often deal with new or unfamiliar data and find it time-consuming to grasp the detailed information for each variable, particularly when the data are large and contain different types of variables, i.e., character or numeric. A common way to get summary information of each variable with SAS is PROC FREQ for character or categorical variables and PROC MEANS (or PROC SUMMARY) or PROC UNIVARIATE for continuous variables. These procedures are fast and easy ways to obtain an overall picture of each variable. In particular, when the data contain less than 1000 observations or so, these procedures can provide fairly complete information of each variable.

However, when the data are large, several problems make it hard to obtain a comprehensive picture of each variable. Here are some difficulties I experienced in the past:

1. For continuous variables with a large number of observations, it is hard to see the frequency distribution with PROC UNIVARIATE or PROC MEANS. It is also hard to determine if some extreme values are a few outlier cases or if they form a group of cases having the same extreme value. For example, when I was examining a bank deposit variable with PROC UNIVARIATE, I had two sets of numbers -- zeros and '99999' -- for extreme values. Because the sample size was very large (n>100,000), I could not tell how many cases had zeros or '99999' by looking at the summary table from PROC UNIVARIATE.

2. When I was given raw data provided by someone else, I often discovered that the distinction between numeric versus character variables was not set up for statistical analyses. For example, I had variables such as incomes or the number of children listed as character variables on the data documentation, when they in fact had just numbers with no characters. In order to find out if these variables do contain characters, I had to go through three steps -- first read these fields as character variables, list some observations or run PROC FREQ, and then read them as numeric if I found out

that they are actually numeric. You could read these variables as numeric and let SAS replace all the characters with missing codes. However, it is possible that some characters are in fact there representing some important information. Therefore, it is always safe to go through the three-step procedure, if the variable is identified as a character type in the documentation.

3. Sometimes, it is hard to tell a priori if the character variable can be examined better with PROC FREQ or by simply printing some of the observations. For example, I can usually prepare a useful summary of zip codes by using PROC FREQ. However, I could have a very long list of zip code frequencies if the variable 'zipcode' includes the 4-digit extension. In such a case, I would first print 50 or 100 cases, examine the variable format, and maybe use only the 5-digit zip codes for analyses. This means several steps are required before finding out the variable characteristics.

4. The variables in the data are not usually organized in a user-friendly way for statistical analysts. Numeric and character variables are scattered at random. Therefore, we would have to manually pick variables of each type and list under PROC FREQ, PROC MEANS (OR UNIVARIATE) or other PROCs. These setups make data management very tedious.

After I encountered these problems repeatedly, I came up with the idea of a small SAS program that produces all the necessary data summaries with just one program execution. I have written a SAS macro program that automatically detects the variable types and provides a tailored summary suitable for each variable in three different ways. For example, if the variable contains addresses or people's names, the program lists the first 50 cases, which is sufficient for us to understand the variable format. If the variable contains 5-digit zip codes or state names, the program provides frequencies of each category. If the variable is continuous and numeric, e.g., sales volumes, or product prices, frequencies based on 25 grouped values are provided. A one-time execution of the program provides documentation that contains summaries for all the variables in the data. The documentation is a useful tool for quickly identifying appropriate statistical analyses for each variable.

PROGRAM PROCEDURE CHART

Figure 1. gives a flowchart of the SAS variable summary program. If the data are flat ASCII or raw data, it is assumed that you have already read in with DATA/INPUT statements before the first step of this program. Basically, the procedure consists of three major steps.

Step 1: PROC CONTENTS to retain the following variable information. The information is kept as macro variables.

- total number of variables (SAS variable name: varnum)
- numeric vs. character (SAS variable name: type)
- variable name (SAS variable name: name)

The remaining part of the program is processed for each variable by a DO loop.

Step 2: PROC FREQ to retain the number of distinct values or fields for each variable. The information on the number of values/fields is kept as a macro variable.

- *If the variable has less than 50 distinct fields (both numeric and character) go to Step 3a*
- *If the variable is numeric and has more than 50 distinct values go to Step 3b*
- *If the variable is a character type and has more than 50 distinct fields go to Step 3c*

Step 3a: If the variable has less than 50 distinct fields (both numeric and character).

- PROC FREQ to obtain a complete frequency table

Step 3b: If the variable is numeric and has more than 50 distinct values.

- PROC RANK to assign all the observations to 25 value-range categories.
- PROC MEANS by the value-range categories – this will provide # of cases, mean, minimum, maximum, range and variance for each category.

Step 3c: If the variable is a character type and has more than 50 distinct fields.

- PROC PRINT with OBS = 50 to print the first 50 observations

Steps 2 and 3 are repeated for all the variables, producing a one-page summary for each variable.

Optional Step: Finally, you can add

```
PROC MEANS;
VAR _NUMERIC_ ;
```

This will provide the overall mean as well as the minimum, maximum, variance, and range of all the numeric variables in the data. Alternatively, PROC UNIVARIATE can provide additional information about the variables that might be useful for some statistical analyses. If you review and compare outputs from Step 3b and from the optional step, you are able to obtain a lot of information about the numeric variables.

CONCLUSION

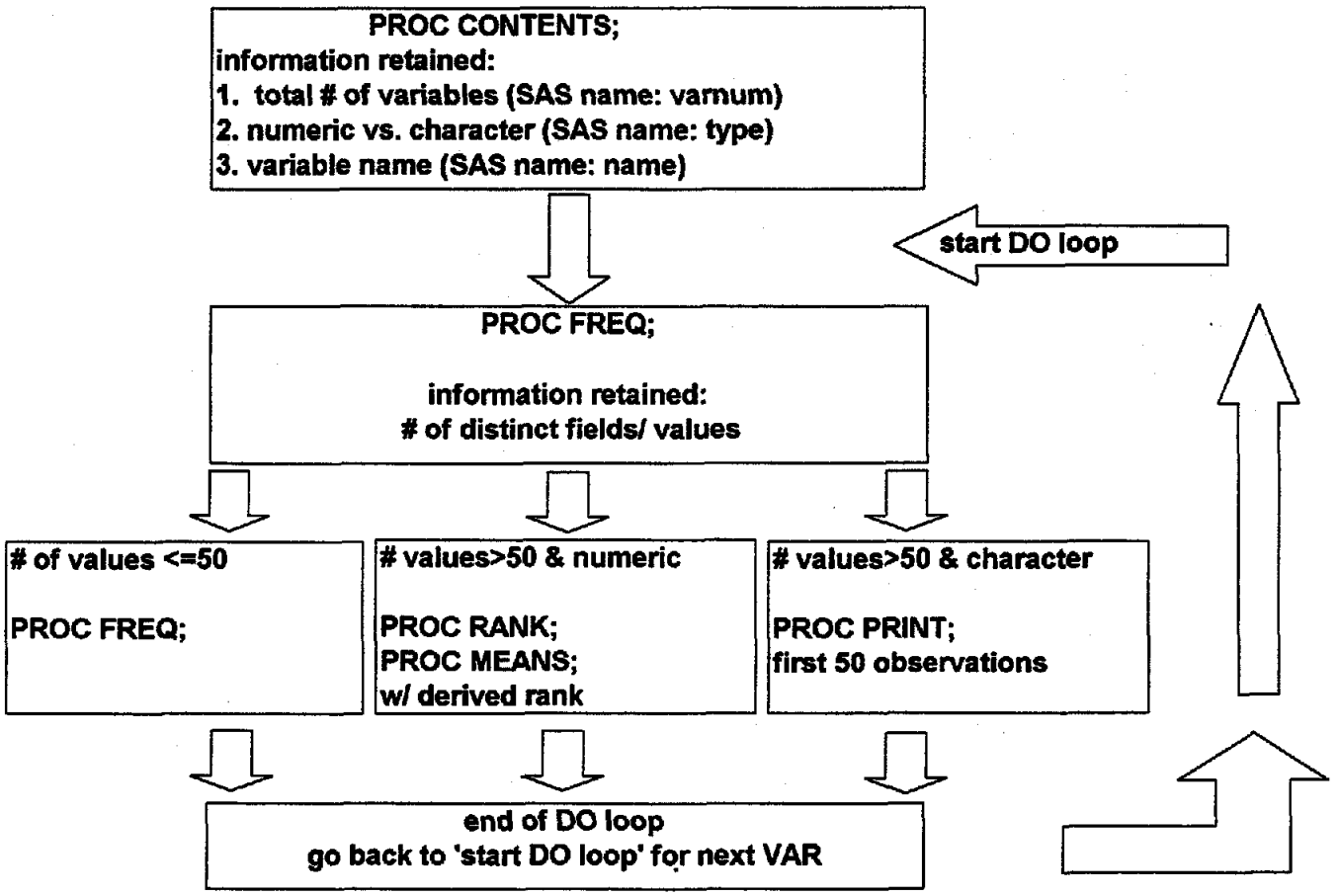
Clearly, a main contribution of this program is the way all these familiar and powerful SAS PROCs – PROC CONTENTS, PROC FREQ, PROC MEANS, and PROC RANK – are combined to produce a user-friendly summary of all the variables with one SAS execution. Statistical consultants/analysts are often expected to complete many different analyses within a limited time. To allocate as much time as possible to analyses rather than to data management, it is essential to get a clear idea about the variables as efficiently as possible. In this sense, this small macro program has been a fairly flexible and useful tool that can provide a comprehensive variable summary in a short time.

CONTACT INFORMATION

Keiko I. Powers, Ph.D.
Senior Analyst
J. D. Power and Associates
30401 Agoura Road, Suite200
Agoura Hills, CA 91301

Work phone: (805) 480-3114
Fax: (805) 480-3241
E-mail: Keiko.Powers@powerinfonet.com

Figure 1. Flowchart of SAS Macro Program



APPENDIX

– SAS Program for Tailored Variable Summary Documentation –

```
filename in '/users/kpowers/project/test.dat' lrecl=566;

options nocenter nodate /* mprint obs=300 */;

%let project = Test run;
%let ncat = 50; /* # of categories for deciding orignal vs.recategorized freq*/

data d1; infile in missover;
input

 !!!!!!!!!!! input format here !!!!!!!!!!!!!!!!!!!!

 ;

*****************************************************
**** macro for variable summary start here ***********;

proc contents noprint data=d1 out=vout;
data vout; set vout;
keep varnum name type;
proc sort data=vout out=out; by varnum;
data _null_; set out;
call symput ('NVAR',varnum);
run;

%macro datachck;

%do i = 1 %to &nvar;
data _null_; set out;
if varnum=&i then call symput ("VNAME&i", name);
if varnum=&i then call symput ("VTYPE&i", type);
run;
%end;

%do i = 1 %to &nvar;
data d2; set d1;
keep &&vname&i;
run;

proc freq data=d2 noprint;
table &&vname&i /out=fout;
data _null_; set fout end=eof;
ncount+1;
if eof then call symput ('NVALUE', ncount);
run;

%if &nvalue<=&ncat %then %do;
proc freq data=d2;
table &&vname&i;
title "freq of original (numeric/character) variables";
title2 "-- &project --";
run;
%end;
```

```
%if &nvalue>&ncat and &&vtype&i=1 %then %do;
proc rank data=d2 groups=25;
var &&vname&i;
ranks vrank;
proc sort; by vrank;
proc means; class vrank;
var &&vname&i;
title "freq of numeric variables (grouped to &ncat categories)";
title2 "-- &project --";
run;

 /* proc univariate optional --
proc univariate data=d2;
var &&vname&i;
title "proc univariate of numeric variables";
title2 "-- &project --";
run;
 */
%end;

%if &nvalue>&ncat and &&vtype&i=2 %then %do;
proc print data=d2(obs=&ncat);
title "print first &ncat obs of character variables";
title2 "-- &project --";
run;
%end;

%end;
%mend;
%datachck

/* proc means optional */
proc means data=d1;
var _numeric_;
title "proc means of all numeric variables";
title2 "-- &project --";
run;

endsas;
```