Paper 88-28

Continuous or Not: How One Can Tell

Vatsala Karwe, Mathematica Policy Research, Princeton, New Jersey

ABSTRACT

The program vcont.sas contains code to identify categorical and continuous variables in a dataset. It produces frequencies for categorical variables, and summary statistics for continuous variables. The program was developed in SAS® version 8.1, on a windows platform. It uses basic macro programming code and is intended for use by any sas user with an elementary knowledge of SAS.

INTRODUCTION

When a new data set becomes available, one wishes to examine it by looking at the frequencies of all the categorical variables, and at some summary measures for continuous variables. Usually, one would like to avoid a proc freq on variables that have a very large number of values – for example, an identification variable such as a name variable, or say, for "height" measured in inches. This paper gives a sas macro that looks at each variable in the data set, determines whether it is categorical or continuous, and produces a proc freq output if it is categorical, a proc means if it is a continuous variable. This is done for all numeric variables. For character variables taking on a small number of categories, a proc freq is produced. A list containing the names of all character variables that have a large number of distinct values in the dataset is also produced.

THE MAIN IDEAS IN THE PROGRAM

OBTAINING VARIABLE CATEGORIES

Obtaining the number of distinct values a variable takes on, (the "categories") is the main objective of the program. This is done by using the chisq option in a proc freq on the variable. The output dataset created by this option contains the variable df_pchi, which is the degrees of freedom associated with that variable, and is one less than the number of categories the variable takes on.

This is done using the following code in the macro:

```
%do ii=1 %to &numvars;
proc freq data=&&data&kk noprint;
output out=w&&varn&ii chisq;
tables &&varn&ii/ chisq; run;
%end;
```

The above code is applied to each variable. A dataset containing the variable name and its corresponding degrees of freedom (df) is created and appended onto the previous variable's output. In this way a dataset varwdf&kk is produced which contains the names of all the variable and corresponding df values. After this, it is a simple matter to ascertain whether the variable is categorical or continuous: if the df is large, it is a continuous variable, if it is small, it is a categorical variable. The user can decide how to define "large" and "small" by choosing an appropriate value for the macro variable &cutoff.

MAXIMIZING EFFICIENCY

For most datasets, and for most variables within that dataset, about 100 records would be sufficient to reveal the different categories of a variable taking on a small number of distinct values. It would be a waste of effort to look at, say 10,000 records, when in fact 100 would be sufficient to decide upon the continuous or categorical nature of a variable.

For this reason, the program proceeds in a stepwise manner. This remains an internal process, and in no way affects the performance of the program other than in improving the efficiency of the program. It requires the user to specify one macro variable –obsgrpsize, but does not require any other input or user-interaction.

The macro is set up to first look at an "obsgrpsize" number of records. This can be chosen by the user as the value of the macro variable &obsgrpsize. Then, the dataset is set into smaller datasets, data1, data2, and so on. Data1 contains obsgrpsize number of records. Data2 contains (2*obsgrpsize) number of records, and for general N, dataN contains (N*obsgrpsize) number of records. For example, if we begin with a dataset that contains 673 records, and we set obsgrpsize=200, then we get four datasets with data1 containing the first 200 records, data2 containing the first 400 records, data3 containing the first 600 records, and finally data4 containing all the records.

The macro first performs the chi-square tests on data1, a dataset containing a small number of records and all the variables from the original dataset. After looking at data1, two sets of variables are identified – one is the set of continuous variables, the other non-continuous. I am calling the second set "non-continuous" because these may be continuous, except that all the distinct values may not have appeared in the first set of obsgrpsize records in data1. The macro then looks at data2, drops the continuous variables identified from looking at data1, and examines the remaining "non-continuous" variables for continuity. If more continuous variables are found, these are dropped from data3, and so data3 is processed only for a smaller set of "non-continuous" variables. At each step, a smaller set of variables is examined. In the final step, all records in the original dataset are checked for continuity of the "non-continuous" variables that remain after the penultimate step.

GENERIC PROPERTIES OF THE PROGRAM

I have tried to make the macro as generic as possible. It works for any sas data set. It looks at any kind of variable, character or numeric. There a few options that the user would need to set before running the macro. These are:

[1] Specify the data set to be examined.

[2] Specify the value of macro variable &cutoff .

As explained above, &cutoff identifies the number of categories beyond which it is desired to consider a variable continuous.

[3] Specify the value of the macro variable &obsgrpsize.

To maximize efficiency of the macro, &obsgrpsize should be chosen to be the number of records (beginning from the first) that would capture the continuous/categorical nature of most of the variables. For most datasets, &obsgrpsize equal to 100 would work and would be a good choice.

[4] Specify choice of output of PROC MEANS or PROC UNIVARIATE for the numeric continuous variables in the data set by setting the macro variable &contproc equal to either "MEANS" or to "UNIVARIATE".

CREATING THE DATA SET CONTAINING CONTINUOUS VARIABLES

The program runs sequentially on data1, data2,…,dataN. At each step I, a macro variable &contvarsI is created from a proc sql step containing the names of the continuous variables. It would have been a simple matter to take the original dataset, and just "keep" these continuous variables in contvars1, contvars2,…It turns out that if any of the strings &contvarsI is empty, the KEEP statement becomes "keep=" and this results in the entire original dataset being "kept". For this reason, I created a BLANKDUMMY variable and converted every blank KEEP statement into keep=BLANKDUMMY. This circumvented the problem of a blank

KEEP statement. Later, I dropped the BLANKDUMMY variable. A PROC CONTENTS for the data set, restricted to the character variables, produces a list of the character variables in the original dataset. Finally, a PROC MEANS or a PROC UNIVARIATE is run on the identified continuous variables according to the value of &contproc set by the user.

CREATING THE DATASET CONTAINING CATEGORICAL VARIABLES

This dataset exists as the last dataset (– say data4 in the example given above -) that was processed. All continuous variables have been dropped from data4 by the end of the program. It is then a simple matter of performing a PROC FREQ on all variables in this dataset to get the frequencies for all categorical variables (both character and numeric) in the original dataset.

CONCLUSION

The input to this program is any sas data set. The output contains the following four elements:

[1] A list of character variable names that take on a large number of values.

[2] Frequency tables for character variables that take on a small number of values.

[3] Frequency tables for numeric variables that take on a small number of distinct values (categorical variables).

[4] PROC MEANS or PROC UNIVARIATE output on numeric variables that take on a large number of distinct values (continuous variables).

This program will make it easy to address that common first need for a look at descriptive statistics on all variables in a new or unfamiliar data set. Whereas looking at small datasets is usually easy by adhoc methods, carrying out the simplest task on a large dataset can often be daunting. This program should make it easy to take an initial look at all variables in a large dataset by providing frequencies for categorical variables and summary statistics for continuous ones.

REFERENCES

Carpenter’s Complete Guide to the SAS® Macro Language. 1998. Art Carpenter, SAS Institute Inc.

Sas® Macro Programming Made Easy. 2001. Michele M. Burlew, SAS Institute Inc.

ACKNOWLEDGMENTS

I wish to thank Mathematica Policy Research for providing project work and reference material that led to and helped in the development of this macro. Also, many thanks to my family for their patience while I developed this code.

CONTACT INFORMATION

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

Vatsala Karwe
Mathematica Policy Research
P.O. Box 2393
Princeton, NJ 08543-2393
Work Phone: (609) 275-2399
Fax: (609) 799-0005
Email: vkarwe@mathematica-mpr.com

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.