Paper 067-2009

Another Way to Make Use of Variable Labels

Stephanie R. Thompson, Rochester Institute of Technology, Rochester, NY

ABSTRACT

Variable labels are a useful part of SAS®. They allow you to put a more descriptive name to the sometimes cryptic variable names encountered as a programmer. Used in many procedures automatically, labels can make reporting much easier. However, what if you want to put a variable label into a different variable? How can it be done? The simple answer is CALL LABEL. This routine allows you to easily drop variable labels into another variable.

INTRODUCTION

Many times datasets are provided with labels to make reporting easier and less cryptic. These labels are used automatically by many procedures and this makes output easier to read and understand. Sometimes it is necessary to use the information in the label differently. In this case, after compiling survey data provided from an external source the questions were a desired part of the output. The verbatim of each question was contained in the label for the variable name representing the question. The customer was requesting her reports in Microsoft Excel so just exporting SAS output using the variable names would not be beneficial in this case since there was no easy way to tie the variable to the question easily once the workbook was created. A way to capture the contents of the label was needed. A search of http://support.sas.com returned an example using the CALL LABEL routine. This simple bit of code solved the problem and was easy to incorporate into the program.

GENERAL CALL LABEL SYNTAX

CALL LABEL has a very simple syntax that is similar to CALL SYMPUT and it is shown below:

```
call label(svar, namelabel);
```

Here “svar” is the name of the variable that has a label that which is being stored in the variable “namelabel.” It is advisable to set the length of namelabel to accommodate the longest text in the labels. Otherwise the contents of namelabel will be truncated. The labels are long in this case since they were the full text of the questions from the survey. Labels can be up to 256 characters so assigning any length longer than that will not be of benefit. In the event that svar does not have a label, the name of the variable (svar in this case) will be stored in namelabel. This is a good feature since null values will not be created.

PROGRAM EXAMPLE

This is the DATA step from a larger program that used CALL LABEL:

```
data nssetmp;
length tempvar $20 namelabel $250;
set nsse1 (keep = colname cname &svar);
where &svar is not missing;
tempvar = put(&svar, &fmt1);
call label(&svar,namelabel);
run;
```

This section of code was within a MACRO loop which iterated for each variable that needed to have the label put into the variable namelabel. The base data set contained about one hundred variables with labels. Having the labels contained in a variable made the export of the data to Excel much easier and provided information that was already in an understandable format for the customer.

ALTERNATIVES

Another option is the VLABEL function. This function works in the same way as the CALL LABEL routine but an expression cannot be used as an argument within the function. A direct variable name reference or an array reference can be used as the argument. The form of the function is below.

```
   vlabel(varname)
```

Varname is the name of the variable that contains the label you want to assign to another variable. In code, it could be used as follows to replace CALL LABEL from the previous example:

```
   data nssetmp;
   length tempvar $20 namelabel $250;
   set nsse1 (keep = colname cname &svar);
   where &svar is not missing;
   tempvar = put(&svar, &fmt1);
   namelabel = vlabel(&svar);
   run;
```

Both CALL LABEL and the VLABEL function produce the same results. VLABEL will assign a default length of 200 to any variable that has not previously been assigned a length. If you do not need a length of 200, assigning a length can reduce the size to what you need. You may also need to increase the length using VLABEL since labels can be up to 256 characters.

VLABELX is similar to VLABEL in syntax, but it can evaluate an expression to determine the name of the variable whose label will be returned. VLABELX cannot have an array reference as an argument.

CONCLUSION

CALL LABEL is a simple way to put variable labels into another variable. It allows you to use the information from the label in ways other than just displaying them in reports or other output.

REFERENCES

Specific reference located on SAS support website: http://support.sas.com/kb/24/664.html

CONTACT INFORMATION

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

Stephanie R. Thompson
Rochester Institute of Technology
Nine Lomb Memorial Drive
Rochester, NY 14623-5603
Work Phone: (585)475-7237
Fax: (585) 475-7950
Email: Stephanie.Thompson@rit.edu
Web: http://finweb.rit.edu/irps/