Paper CC76

Manage Variable Lists for Improved Readability and Modifiability

David H. Abbott, Veterans Affairs Health Services Research

ABSTRACT

Lists of variables occur frequently in statistical analysis code, for example, lists of explanatory variables, variables used as rows in demographics tables, and so forth. These lists may be long, say 10-30 or more variables and the same list, or a major portion of it, may occur in multiple places in the code. The lists are often replicated using copy and paste by the programmer during program composition. Readers of the code may find themselves doing repeated "stare and compare" to determine if the list in location A is really the same list as in location B or location C. Simply adding a variable to the list may require changing numerous lines of code since the list occurs in the code numerous times. If managed naively, variable lists can impair code readability and modifiability.

The SAS® macro facility provides the tools needed to eliminate repeated entry of lengthy variable lists. Related groups of variables can be assigned to macro variables and the macro variables concatenated as needed to generate the list of variables needed at different points in the code. Certain SAS macros can be used to programmatically alter the list, for example, remove specific variables from the list (not needed for a given regression) or change the delimiter character to comma (when the list is used with PROC SQL). The macro variable names can express the purpose of the groups of variables, e.g. ExplanVars , OutcomeVars, DemographicOnlyVars, etc. Employing this approach makes data analysis code easier to read and modify.

Problem/Opportunity

Lists of variables tend to occur frequently in SAS code, especially in data cleaning and analysis programs. For example, regression models involve the lists of explanatory variables and data cleaning programs contain lists of variables to be checked for certain properties. Typically, these lists, or major portions of them, occur more than once and perhaps many times. A SAS programmer who is reviewing/modifying a program needs to determine whether or not the contents of the several lists are the same and, if not, how they differ. The samenesses and differences of the several lists are not apparent and the effort involved in checking them for equivalence can be substantial. Further, the lists may have sublists in common and that is not apparent to code readers without tedious stare and compare work.

Proposed Solution

These difficulties with lists of variables can be addressed using basic elements of the SAS macro facility:

- Assign each useful grouping of variables to a macro variable
- Name the macro variables informatively
- Represent variable lists in the program using the macro variables.

That is the strategy. Now, let's look at the implementation details via an illustrative example...

Example Code (Before)

```
Title1 Explore distributions of independent
variables; proc freq data=ChemoData;
  table race gender region prevCancer     stage grade insur/missing;
...;
proc                univariate
  data=ChemoData;  var  age  BMI
  dateDiag income; histogram;
...
```

```
Title1 Check for multicolinearity;
proc reg data=ChemoData;
  model chemoRecvd = age BMI dateDiag income/vif;

Title1 Perform logistic regression;
proc logistic data=ChemoData simple desc;
   class race gender region prevCancer stage    grade insur /param=ref desc;
   model chemoDi = race gender region prevCancer stage grade insur age BMI
    dateDiag income     /scale=none rsq lackfit;
```

Solution Steps

The several steps in applying the proposed solution are:

- Identify the useful groupings of variables – in this simple example it suffices to group according to whether the variables are continuous or categorical
- Assign each group to a macro variable – see "Assignment" below
- Name the macro variables informatively
- Use as needed in executable code – see "Example Code (After)"

Assignment

```
%let cateVars=race gender region prevCancer stage grade
insur; %let contVars=age BMI dateDiag income;
...
```

Example Code (After)

```
Title1 Explore distributions of independent
variables; proc freq data=ChemoData;
  table &cateVars /missing;
...;
proc univariate data=ChemoData;
  var &contVars;
  histogram;
Title1 Check for multicolinearity in continuous IVs;
proc reg data=ChemoData;
   model chemoRecvd = &contVars /vif;
run;
Title1 Perform logistic regression;
proc logistic data=ChemoData simple desc;
   class  &cateVars /param=ref desc;
   model  chemoDi = &cateVars &contIndVars       /scale=none rsq lackfit
```

Comparison of Before/After

The after code is arguably significantly easier to read and clearly easier to modify. Adding an additional categorical variable to the analysis requires only one word of code to be changed in the After code and three words to be changed in the Before code. More importantly, with the After code, no list comparison need to be made and verifying correctness of the change is more straight-forward.

Handling Alterations/Variations

In a long SAS program representing a comprehensive analysis, several types of alterations to/variations of the variables lists may need to be accommodated. Some possibilities are:

- Replace a variable with a different version, e.g., change the set of categories used for a variable
- Convert a variable list to being comma delimited (e.g.,for use with PROC SQL)
- Eliminate statistically uninformative variables
- Add variables for a portion of the analysis

These types of alterations can all be achieved without resorting to cutting and pasting previously enumerated sets of variables and again forcing code readers to do "stare and compare" to verify correctness. In the flow of the program, these alterations can be made explicit, e.g.,

```
explanVars2=%replaceWord(inStr=explanVars, swapIn=age3c, swapOut=age5c)
```

results in the age variable with 3 age categories replacing the age variable with 5 categories in the list of explanatory variables.

Macros of interest

Of course, the %replaceWord macro is not standard and needs to be acquired and compiled in the session and two additional macros prove useful in modifying lists of variables:

```
%macro replaceWord( /*replace a word in a string of
  characters*/ inStr=, /*list of words to be altered*/
  swapOut= /*word to be replaced in the list*/,
  swapIn= /*word replacing the swapOut word*/)
%* Author: David H. Abbott;

%macro removeWords(/*remove a list of words from a
    string*/ baseList=, /*list of words to be altered*/
    remList=, /*list of words to be removed*/ );
%* Author: David H. Abbott;

 %macro seplist (/*emit a list of words separated by a delimiter*/
    Items /* list  of items, separated by indlm */
   , indlm = %str( ) /* string that delimits each item of items */
   , dlm   = %str(,)/*string that delimits list of items emitted */
   , prefix=/* string to place before each item */
   , nest=/*nesting character, e.g. Q for single quote*/
   , suffix=/* string to place after each item */);
%* Author: Richard A. DeVenezia;
```

Programmers can try their hand at implementing these macros or procure them from the authors.

Take Aways

- Long list of variables need occur only once
- Readability benefits when lists are named
- Modifiability benefits when occur only once
- Both benefit when alterations to lists are explicit

References

Carpenter, AL, 2004. "Storing and Using a List of Values in a Macro Variable". PNWSUG 2004. http://www.lexjansen.com/pnwsug/2004/c_cc_storing_and_using_a_lis.pdf

Hu, Jiangtang, 2013. "List Processing with SAS: A Comprehensive Survey". SESUG 2013. http://analytics.ncsu.edu/sesug/2013/BtB-18.pdf

Rozhetskin, Dmitry, 2010. "Choosing the Best Way to Store and Manipulate Lists in SAS®". WUSS 2010. http://www.wuss.org/proceedings10/coders/2972_9_COD-Rozhetskin.pdf

SAS Macros by Richard A. DeVenezia
http://www.devenezia.com/downloads/sas/macros/index.php

ACKNOWLEDGMENTS

The views expressed in this paper are those of the author and do not necessarily reflect the position or policy of the Department of Veterans Affairs or the United States government.

Without the leadership and encouragement of Dr. Dawn Provenzale director of the Cooperative Studies Program Epidemiology Center at the Durham VA Medical Center, this work could not have occurred. She takes a strong interest in fostering many dimensions of excellence in her employees.

CONTACT INFORMATION

Name	David H. Abbott
Enterprise	Center for Health Services Research in Primary Care
Address	Durham Veterans Affairs Medical Center
	HSR&D Service (152)
	508 Fulton St.
City, State ZIP	Durham, NC 27705
Work Phone:	919-286-0411
E-mail:	david.abbott@va.gov