Simple methods to pull the diagonal out of a correlation matrix

Laura L. M. Thornton, Animal Improvement Programs Laboratory, Agricultural Research Service, USDA, Beltsville, MD

ABSTRACT

When using large data sets with many variables, it is often necessary to find how the variables are correlated. However, many times, the cross-correlations among all the variables are not as important as the diagonal of a correlation matrix where both the var and with clauses are invoked. A simple and effective use of macro variable references can be employed to display only the diagonal using the noprint option for the PROC CORR and a single dataset step. While the simple statistics are not printed prior to the diagonal of the matrix, they can be obtained using a PROC MEANS prior to the invocation to the correlation procedure. The methodology for two ways of extraction of singular pieces from the matrix and various adaptations are presented.

INTRODUCTION

Often correlation matrices have a diagonal of 1.00, while the off-diagonals provide the useful information that a researcher is searching for. However, when using variables in the correlation program that are indirectly correlated and a measurement of the strength of relationship is needed, the diagonal is the most useful portion of the matrix. Where many variables are used, the entire correlation matrix may become unwieldy and cumbersome, spanning many pages, making it very difficult to read the pertinent information. One strategy is to reduce the number of variables included in the analysis through the use of the var and with statements. The var denotes the main variables for the correlation, while the with statement identifies the variables within the data set that the main variables are to be correlated with for the matrix. Occasionally, even when a similar strategy is employed, the resulting matrix may still be several variables wide and long, and the main objective is still to look at how a particular variable is correlated to a similar dependent variable. It is then necessary to pull only the diagonal in order to consider only those correlations of use, particularly when many variables are identified with the use of var and with statements.

OBTAINING A DIAGONAL WITHOUT A MACRO

When only one data set is present or a single file is to be processed, a macro utility is not necessary and may actually be more cumbersome. However, the following is not flexible and cannot easily be altered to take off-diagonal values; it is however very quick to pull a diagonal from a fairly large correlation matrix. Because the code is not flexible, use it when the diagonal of the correlation matrix is the desired output.

```
proc means data=match;
 var varble1-varble16 newvar1-newvar16;
proc corr noprob data=match outp=sav.outcorr;
 var varble1-varble16; with newvar1-newvar16;
 title "Comp corr btwn dep var";
run;
data sav.outcorr;
 array varble{16};
 set sav.outcorr(where=(_type_ = 'CORR'));
 j+1;
 cor = varble{j};
 keep _name_ cor; format cor 8.3;
run;
proc print; id _name_;
```

Often, the simple statistics are useful diagnostics in checking data quality; putting the means prior to the correlation call makes a neat and clean presentation so long as only the variables used in the correlation are presented. Of course, standard options can be added to format the means as desired. Including the noprob option on the correlation call causes the output data set to contain only the correlation matrix, so the use of arrays can make extracting the diagonal fairly quick, so long as the data set size is reasonable. Because the noprint option is not specified, the entire matrix will be printed so that you can make sure the diagonal prints correctly. Adding the noprint option should save printed output and space in the .lst file and can be done after preliminary testing to make sure the program is working correctly.

This code can be altered to pull the diagonal from a covariance matrix or any other matrix where _TYPE_ is defined in the output.

MACRO UTILITY TO OBTAIN A DIAGONAL

Occasionally, several data sets may have the same type of data, but merging is not an option because of issues such as comparison across time or regions. Even though the variables are similar, each data set must be processed separately. In order to prevent simple coding errors, such as forgetting to change a variable in the var or with sections, a macro can be used to efficiently run all data sets and create printed copies where only the desired output (in this case, the diagonal) is present. It is important to note that the variables must be done in such a way that a numerical sequence can be assigned to the variables, or the %do loop will not run.

```
%macro nexcor (fn);

Title "&fn corr btwn dep var";
proc means data=&fn;
    var test1-test3 nv1-nv3;
proc corr data=&fn noprint outp=pearcorr;
  var test1-test3; with nv1-nv3;
  data corrlate;
  set pearcorr (firstobs=4);
 %do i=1 %to 3 ;
  if _NAME_="nv&i" then nvxtest=test&i;
 %end;
 keep _NAME_ nvxtest;
proc print data=corrlate;
run; quit;

%mend;
%nexcor(testfile)
```

The noprint option is used to suppress printing a matrix the size of the number of input variables; in this case, it is three by three. The noprob option is not included because the code specifically pulls only the variables where the name is nv&i so the probabilities are not included. This macro only works when sequential variables are presented (e.g. data has variables 1 through 60 and cross-correlation variables of 1 through 60). It is generally not necessary to include the semicolon following a macro call.

In this case, there were only six cross-correlation variables, but when many more are present, altering each on in the step can be time consuming and frustrating. As with the case in the previous code, where _NAME_ is defined within the output of a procedure, the step can be altered to extract the diagonal from the output matrix.

MULTIPLE DIAGONAL PULLS

Sometimes, several types of variables are correlated to a single set of other variables. The code previously presented is easily altered to extract the diagonals for several groups of variables. As before, the variables but be numbered sequentially for the program to run correctly as written here.

The code to produce output for multiple diagonals is very similar to the macro above:

```
%macro nexcor2 (fn);

Title "&fn corr of vartype1, vartype2, and
vartype3 with dependant var4";
proc means data=&fn;
  var var1_1-var1_15 var2_1-var2_15 var3_1-
  var3_15 var4_1-var4_15;
proc corr data=&fn noprint outp=pearcorr;
  var var4_1-var4_15;
    with var1_1-var1_15 var2_1-var2_15
    var3_1-var3_15;
data corrlate;
 set pearcorr (firstobs=4);
%do i=1 %to 15 ;
 if _NAME_="var1&i" then var1xvar4=var4&i;
 if _NAME_="var2&i" then var2xvar4=var4&i;
 if _NAME_="var3&i" then var3xvar4=var4&i;
%end;
  keep _NAME_ var1xvar4 var2xvar4 var3xvar4;
proc print data=corrlate;
run; quit;

%mend;
%nexcor2(filename)
```

This program will create 3 columns of output that contain only diagonals of the correlations of all var type 1 to var type 4, of all var type 2 to var type 4, and of var type 3 to var type 4. Missing variables occur in each output column; those are due to the cross-correlation of each variable to the others included in the output. A reduction of the size of the printed output and space saved in the .lst file occurs by using this strategy. Where there would be three 15 by 15 matrices, there is now a single output of three columns. As before, the code can be altered to work anywhere the output data set includes _NAME_ as an automatic label.

The previous examples are macros to make variable processing simple; it is not suggested that the macros be removed. Removing the macro variables would require each variable combination to be named individually prior to printing. When dealing with many variables, that alternative would be time consuming and frustrating. A single file will run as easily as multiple files.

CONCLUSIONS

SAS provides many ways to do different tasks. Working with large matrices is sometimes a daunting task, but can be simplified by extracting only information that is useful. The code presented here illustrates a couple of ways to extract a single diagonal line, whether a main diagonal or an off diagonal, for use of necessary comparisons between correlations. By extracting only the diagonal or a single off diagonal, the size of the printed output and space for a .lst file are greatly reduced. Perhaps more importantly, the results are easier to read and are presented in a clear and concise manner.

The code presented here is specific to the correlation procedure within SAS, but it can easily be altered to pull the diagonal or off diagonal from matrices generated by other SAS programs.

REFERENCES:

SAS Institute Inc (1999), "SAS Macro Language: Reference version 9," Cary, NC: SAS Institute Inc.

Carpenter, A. (2004) Carpenter's Complete Guide to the SAS® Macro Language, Second Edition

Delwiche, L. D. and S. J. Slaughter (2003) The Little SAS® Book: A primer, Third Edition

SAS Online Documentation (SAS V9.1)

Numerous postings on the SAS listserv:
http://www.listserv.uga.edu/archives/sas-l.html

ACKNOWLEDGMENTS

SAS is a Registered Trademark of the SAS Institute, Inc. of Cary, North Carolina.

I would be greatly remiss if I did not recognize George Wiggans and Ashley Sanders of AIPL for their assistance in writing and debugging the code for me. When I first learned these tools, both were willing to give me a starting place and when I really messed it up, helped me make the code work. Both are very giving of their expertise and help me every time I need assistance in order to be a better SAS programmer.

CONTACT INFORMATION

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

Laura L. M. Thornton
Animal Improvement Programs Laboratory, ARS, USDA
Room 306, Building 005, BARC-West
10300 Baltimore Avenue
Beltsville, MD 20705-2350 USA
Phone: 301.504.5068
Fax: 301.504.8092
Email: laura@aipl.arsusda.gov
Web: http://aipl.arsusda.gov/

****