Paper GH-11

Creating Correlated Variable Tables Dynamically

John Barrow, Aref Dajani, U.S. Census Bureau, Suitland, MD

ABSTRACT

Certain survey imputation methods rely on a correlated variable. In this instance, it is advantageous to have a summary of the correlation matrix. During a recent imputation study of the Annual Survey of Local Government Finances, a two column static table was generated containing a column of all variables and a column of their highest correlated variables. To generate this table, PROC SQL and PROC CORR were used as well as macro coding. The table was used together with the %SCAN macro function to impute variables using their highest correlated variable. The power of this macro is that the table is generated dynamically with each imputation implementation. Thus, if variables are added to a survey or study, an updated table will be created without additional effort. The authors present three solutions: one using PROC ODS, one using arrays, and one using macros.

INTRODUCTION

PROC CORR computes the Pearson correlation coefficients for a data set. It has an option BEST=*n* that is used to return the *n* highest correlates for all variables. When BEST=6 the OUTPUT window shows the user the five highest correlated variables and the variable itself (denoted Variable). This result is more useful in a data set versus in the OUTPUT window. Also, the results from PROC CORR displayed in the OUTPUT window are different from the results captured in the SAS® data set. Figure 1 shows the PROC CORR results in the OUTPUT window whereas Figure 2 shows the SAS data set results. The task in this paper is to take the results from the OUTPUT window (when Best=6) and place them into a SAS data set, while keeping the process dynamic. The final table should have six columns: one containing all variables followed by five columns containing its five highest correlated covariates in descending order. This paper presents three solutions: arrays, macros, and PROC ODS. The array and macro solutions demonstrate different ways to sort individual columns or rows of a data set. PROC ODS is the most efficient method. It quickly creates data sets from OUTPUT results.

The CORR Procedure
Pearson Correlation Coefficients, N = 376
Prob > |r| under H0: Rho=0

CY_19T	CY_19T	CY_44T	CY_W01	CY_34T	CY_24T	CY_E23
	1.00000	0.99962	0.99879	0.99489	0.98489	0.98367
		<.0001	<.0001	<.0001	<.0001	<.0001
CY_19U	CY_19U	CY_49U	CY_39U	CY_29U	CY_I89	CY_W31
	1.00000	0.99994	0.99865	0.99717	0.98687	0.98610
		<.0001	<.0001	<.0001	<.0001	<.0001

Figure 1: Partial PROC CORR results from the OUTPUT window

VIEWTABLE: Pearson Correlation Matrix

	TYPE	_NAME_	CY_19T	CY_19U	CY_24T	CY_29U	CY_34T
1	CORR	CY_19T	1	0.9195425578	0.9848895384	0.9183184155	0.9948945315
2	CORR	CY_19U	0.9195425578	1	0.861413777	0.997168476	0.8959715941
3	CORR	CY_24T	0.9848895384	0.861413777	1	0.8615137665	0.9927762528
4	CORR	CY_29U	0.9183184155	0.997168476	0.8615137665	1	0.8953057058
5	CORR	CY_34T	0.9948945315	0.8959715941	0.9927762528	0.8953057058	1
6	CORR	CY_39U	0.9251397778	0.9986492689	0.8692695455	0.9980901798	0.9031326076
7	CORR	CY_44T	0.9996168588	0.9122281917	0.9890648912	0.911200681	0.9962690915
8	CORR	CY_49U	0.9183321349	0.9999417974	0.8599238479	0.9977993323	0.8945460881

Figure 2: Partial PROC CORR results from a SAS data set

DISCLAIMER

The contents of this paper are the work of the authors and do not necessarily represent the views of the U.S. Census Bureau.

DATA

The data used in this paper were from the 2007 Census of Governments: Finance. These are public data available at the U.S. Census Bureau website. Instructions to obtain the data and create the input data set REPORTED are available in the Appendix. These data are not confidential. Each observation contains State Code, Type of Government, Item Code, Amount, Coefficient of Variation, and Year of Survey. Data for each unit were placed on a single row with Item Code as column names. All other variables were excluded for this paper. The variables of interest have CY_ attached to them. Some examples are CY_19T, CY_19U, CY_A01, and CY_F91. The name of this data set is REPORTED.

VIEWTABLE: Paper.Reported

	CY_19T	CY_19U	CY_24T	CY_29U	CY_34T	CY_39U	CY_44T
1	511890810	1706373113	83868442	302596468	47447841	180097211	553770734
2	317964450	541465616	62819873	98599956	32365073	60384769	354657491
3	193926360	1164907497	21048569	203996512	15082768	119712442	199113243
4	77604179	168133157	6135786	28827947	5463504	17128679	77529085
5	76397985	457727150	11670045	65357122	6556145	44843695	81480344
6	1801709	24909719	144597	3396060	187025	2619102	1759281
7	38122487	232415386	3098141	45363711	2876094	24177084	38344533
8	0	281722085	0	61051672	0	30943882	0

Table 1: Partial Input Data Set, REPORTED

DICTIONARY TABLES

To keep the table creation process dynamic, PROC SQL is used. SQL queries the dictionary tables and places the variable names in a macro variable. This eliminates the need to update code if a variable is deleted/added from the data set. Dictionary tables are SAS views created within a session containing metadata regarding libraries, data sets, macros, and external files. This information is located in the SAS view named VCOLUMN which is located in the library SASHELP. From that view, memname specifies the data set being referenced.

```
PROC SQL NOPRINT;
  SELECT NAME, NAME, NAME, COUNT(*)
  INTO :varlist SEPARATED BY ' ', :varlistcomma SEPARATED BY ',',
       :varlistquote SEPARATED BY '","', :numvars
  FROM SASHELP.VCOLUMN
  WHERE LIBNAME='PAPER' AND MEMNAME='REPORTED';
QUIT;
```

Figure 3: SAS SQL used to query Dictionary table SAShelp.vcolumn

The macro variable varlist contains all variable names separated by spaces. The macro variable varlistcomma contains all variable names separated by commas, and the macro variable varlistquote contains all variable names separated by quotations. The last two are only used in the array solution.

PROC CORR

PROC CORR computes the Pearson correlation coefficients for REPORTED. NOSIMPLE suppresses printing simple descriptive statistics, NOMISS excludes observations with missing values, and NOPRINT suppresses displayed output. The OUTP= option prints the Pearson correlations to a data set. The use of the macro variable varlist is used here as a text substitution for the list of variables in REPORTED.

```
PROC CORR DATA=PAPER.REPORTED NOSIMPLE NOMISS NOPRINT OUTP=PAPER.CORRS
          (WHERE=(_TYPE_="CORR"));
  VAR &varlist.;
RUN;
```

Figure 4: SAS code to generate Correlation matrix

ARRAY SOLUTION

Within a DATA step, the correlation table is placed into an array. This process remains dynamic by using the macro variables &numvars and &varlist to populate the array.

```
ARRAY var {&numvars} &varlist;
ARRAY corrvname {&numvars} $ ("&varlistquote");
```

Figure 5: ARRAY Definitions

The SAS code in figure 5 creates arrays named var and corrvname. Each has dimension equal to the value of numvars. The array var is composed of the variables in varlist, while corrvname is populated by the names of those variables from varlistquote. Two arrays are necessary since var contains numeric variables while corrvname contains character. Var is used to sort the correlation coefficients within each row using the LARGEST function. Corrvname is used to assign the appropriate variable name that corresponds to its correlation coefficient.

```
DO z=1 TO &numvars;
  IF (var{z}=LARGEST(2,&varlistcomma)) THEN Best1=corrvname{z};
  IF (var{z}=LARGEST(3,&varlistcomma)) THEN Best2=corrvname{z};
  IF (var{z}=LARGEST(4,&varlistcomma)) THEN Best3=corrvname{z};
  IF (var{z}=LARGEST(5,&varlistcomma)) THEN Best4=corrvname{z};
  IF (var{z}=LARGEST(6,&varlistcomma)) THEN Best5=corrvname{z};
END;
```

Figure 6: SAS code to determine highest five correlates

MACRO SOLUTION

The macro program top5 splits the full correlation table into columns. Each column is placed into a data set entitled COLUMN*n* where *n*=1,…,358. Once the columns are in separate data sets they are sorted in descending order. Finally, only the first six observations are kept in each column. These six rows are the variable itself and its highest five correlated variables. The preceding paragraph is executed in a %DO loop as in Figure 7. This requires a macro program.

```
%MACRO top5;
  %DO i=1 %TO &numvars;
    DATA COLUMN&i. (KEEP=%SCAN(&varlist.,&i.) _NAME_);
      SET PAPER.CORRS;
      %SCAN(&varlist.,&i.)=ABS(%SCAN(&varlist.,&i.));
    RUN;

    PROC SORT DATA=COLUMN&i.;
      BY DESCENDING %SCAN(&varlist.,&i.);
    RUN;

    DATA COLUMN&i.;
      SET COLUMN&i.;
      IF (_N_ LE 6);
    RUN;
  %END;
```

Figure 7: SAS MACRO program top5 (part 1)

There are 358 data sets that contain six observations. Next they are appended into one data set. The appended data sets contain the variables with the next highest correlated in the following row. To create the six column correlation table, the program uses an indicator variable, *c*. The CEIL function is used to assign the same value to each of these variables. A data step with a MERGE statement creates the desired table, merging by *c*.

```
DATA ALLCOLUMNS;
  SET %DO i=1 %TO &numvars;
        COLUMN&i.
      %END;
      ;
RUN;

DATA  %DO i=0 %TO 5;
        CTDATA&i. (KEEP= Best&i. c)
      %END;
      ;
  SET ALLCOLUMNS;
  x=MOD(_N_,6)-1;
  IF x=-1 THEN x=5;
  c=CEIL(_N_/6);
  %DO i=0 %TO 5;
    IF x=&i. THEN
      DO;
        Best&i. = _NAME_;
        OUTPUT CTDATA&i.;
      END;
  %END;
RUN;
```

Figure 8: SAS macro program top5 (part 2)

The six column correlation table is made with a MERGE statement within a DATA step. After creating the correlation table, PROC DATASETS is used to delete the unnecessary *n* column data sets.

```
        DATA CTABMACRO (RENAME=(Best0=Variable));
          MERGE %DO i=0 %TO 5;
                  CTDATA&i. (KEEP= Best&i. c)
                %END;
                ;
          BY c;
          DROP c;
        RUN;

        PROC SORT DATA=CTABMACRO;
          BY Variable;
        RUN;

        PROC DATASETS NOLIST;
          DELETE COLUMN:;
        QUIT;
%MEND;
%top5;
```

Figure 9: SAS macro program (part 2)

ODS TRACE SOLUTION

PROC ODS is the most elegant way to solve the problem of creating correlation tables as SAS data sets. When the ODS TRACE statement is submitted, SAS outputs a record of each output object to the log. This information is then used to create a SAS data set from the information displayed in the OUTPUT window. It is necessary to submit the PROC CORR statement before the ODS OUTPUT in order to obtain the Name of the output data set.

```
Output Added:
-------------
Name:       PearsonCorr
Label:      Pearson Correlations
Template:   base.corr.StackedMatrixNCH
Path:       Corr.PearsonCorr
-------------
```

Figure 10: SAS Log results

Figure 10 shows the Log results after submitting PROC CORR with ODS TRACE=ON. The output object, NAME, (also from Figure 10) is then used with an ODS OUTPUT statement to create a SAS data set containing the results from the procedure. These results are viewable in the OUTPUT window, as shown in Figure 1. Table 2 shows the data set created by the ODS OUTPUT statement. The desired result is a table containing the columns Variable and Best2 through Best6 which is addressed using the KEEP= statement. The full code is listed below in Figure 11.

```
ODS OUTPUT PEARSONCORR=PAPER.CTABODS (KEEP=variable best2-best6
       RENAME=(best2=Best1 best3=Best2 best4=Best3 best5=Best4 best6=Best5));

PROC CORR DATA=PAPER.REPORTED NOSIMPLE BEST=6 NOMISS OUTP=CORRS;
  VAR &varlist;
RUN;

ODS LISTING;
ODS TRACE OFF;
```

Figure 11: SAS Code used to generate Figures 7, 8 and Table 2

VIEWTABLE: Pearson Correlations

	Variable	Best1	Best2	Best3	Best4	Best5	Best6	R1	R2	R3	R4	R5
1	CY_19T	CY_19T	CY_44T	CY_W01	CY_34T	CY_24T	CY_E23	1.00000	0.99962	0.99879	0.99489	0.98489
2	CY_19U	CY_19U	CY_49U	CY_39U	CY_29U	CY_I89	CY_W31	1.00000	0.99994	0.99865	0.99717	0.98687
3	CY_24T	CY_24T	CY_34T	CY_44T	CY_F44	CY_X11	CY_T09	1.00000	0.99278	0.98906	0.98888	0.98696
4	CY_29U	CY_29U	CY_39U	CY_49U	CY_19U	CY_W31	CY_Z00	1.00000	0.99809	0.99780	0.99717	0.98847
5	CY_34T	CY_34T	CY_44T	CY_19T	CY_W01	CY_24T	CY_F44	1.00000	0.99627	0.99489	0.99380	0.99278
6	CY_39U	CY_39U	CY_49U	CY_19U	CY_29U	CY_I89	CY_Z00	1.00000	0.99867	0.99865	0.99809	0.98910
7	CY_44T	CY_44T	CY_19T	CY_W01	CY_34T	CY_24T	CY_E23	1.00000	0.99962	0.99851	0.99627	0.98906
8	CY_49U	CY_49U	CY_19U	CY_39U	CY_29U	CY_W31	CY_I89	1.00000	0.99994	0.99867	0.99780	0.98710
9	CY_52T	CY_52T	CY_53T	CY_Q12	CY_M50	CY_M44	CY_M30	1.00000	0.99844	0.97114	0.96621	0.96520
10	CY_53T	CY_53T	CY_52T	CY_Q12	CY_M50	CY_M44	CY_M05	1.00000	0.99844	0.96983	0.96846	0.96236

Table 2: SAS data set showing results of PROC CORR

CONCLUSION

Three six column tables were created using different methods: arrays, macros, and PROC ODS. PROC ODS has wide ranging applications since it provides a way to easily convert results in the Output window to a SAS data set. PROC ODS is the fastest, simplest solution of the three presented here. It allows for easy use of the BEST=*n* option and therefore does not require additional work to generalize. The array solution is next in order of simplicity, but still uses macro coding and subtle quotations to populate the arrays. The macro solution is the most complex. Both arrays and macros offer the flexibility to generalize the method to *n* columns. A subset of the final table is presented in Table 3.

VIEWTABLE: Paper.Ctabarray

	Variable	Best1	Best2	Best3	Best4	Best5
1	CY_19T	CY_44T	CY_W01	CY_34T	CY_24T	CY_E23
2	CY_19U	CY_49U	CY_39U	CY_29U	CY_I89	CY_W31
3	CY_24T	CY_34T	CY_44T	CY_F44	CY_X11	CY_T09
4	CY_29U	CY_39U	CY_49U	CY_19U	CY_W31	CY_Z00
5	CY_34T	CY_44T	CY_19T	CY_W01	CY_24T	CY_F44
6	CY_39U	CY_49U	CY_19U	CY_29U	CY_I89	CY_Z00
7	CY_44T	CY_19T	CY_W01	CY_34T	CY_24T	CY_E23
8	CY_49U	CY_19U	CY_39U	CY_29U	CY_W31	CY_I89
9	CY_52T	CY_53T	CY_Q12	CY_M50	CY_M44	CY_M30
10	CY_53T	CY_52T	CY_Q12	CY_M50	CY_M44	CY_M05

Table 3: Partial SAS Data set showing all variables and their 5 highest correlates

REFERENCES

SAS Institute Inc. "PROC CORR Statement" Base SAS® 9.2 Procedures Guide: Statistical Procedures, Third Edition.
http://support.sas.com/documentation/cdl/en/procstat/63104/HTML/default/viewer.htm#procstat_corr_sect004.htm

SAS Institute Inc. "Definition of a DICTIONARY Table" Base SAS® 9.2 Procedures Guide: Statistical Procedures, Third Edition.
http://support.sas.com/documentation/cdl/en/lrcon/62955/HTML/default/viewer.htm#a002300184.htm

SAS Institute Inc. "ODS TRACE Statement" Base SAS® 9.2 Procedures Guide: Statistical Procedures, Third Edition.
http://support.sas.com/documentation/cdl/en/odsug/61723/HTML/default/viewer.htm#a002233618.htm

SAS Institute Inc. "DELETE Statement" Base SAS® 9.2 Procedures Guide: Statistical Procedures, Third Edition.
http://support.sas.com/documentation/cdl/en/proc/61895/HTML/default/viewer.htm#a000247666.htm

APPENDIX

The Census Data used to create the data set REPORTED can be found at http://www.census.gov/govs/estimate/historical_data_2007.html#state_local. Select the "State by Type of Government – Public Use Format" ZIP file. This will download a text file, 07statetypepu.txt. Place that file in a folder named PAPER. It may be necessary to change the file locations in the code below.

```
OPTIONS NOMLOGIC NOMPRINT NOSYMBOLGEN;
ODS TRACE ON;

LIBNAME PAPER 'H:\Sas Paper';  /* This should be changed to the user's location. */
%GLOBAL numvars;

/* Clean input data */

DATA PAPER.INPUT;
  INFILE "H:\Sas Paper\07statetypepu.txt";  /* Change to the user's location. */
  INPUT state_code $ 1-2
        type_of_gov $ 3
        item_code $ 5-7
        amount 9-20
        ;
RUN;

PROC TRANSPOSE DATA=PAPER.INPUT OUT=PAPER.REPORTED PREFIX=CY_;
  BY state_code type_of_gov;
  ID item_code;
RUN;

DATA PAPER.REPORTED;
  SET PAPER.REPORTED (DROP= CY_A54 CY_K: CY_Y10 CY_Y15 CY_Y50 CY_Y54 CY_Z45
                            state_code type_of_gov _NAME_);
RUN;

/* Create macro variables */

PROC SQL NOPRINT;
  SELECT NAME, NAME, NAME, COUNT(*)
  INTO :varlist SEPARATED BY ' ', :varlistcomma SEPARATED BY ',',
       :varlistquote SEPARATED BY '","', :numvars
  FROM SASHELP.VCOLUMN
  WHERE LIBNAME='PAPER' AND MEMNAME='REPORTED';
QUIT;

/* Create Correlation Matrix */
```

```
PROC CORR DATA=PAPER.REPORTED NOSIMPLE NOMISS NOPRINT OUTP=PAPER.CORRS
             (WHERE=(_TYPE_="CORR"));
  VAR &varlist.;
RUN;

/* ARRAY SOLUTION ************************************************************/

DATA PAPER.CTABARRAY (RENAME=(_NAME_=Variable));
  SET PAPER.CORRS;
  ARRAY var {&numvars} &varlist;
  ARRAY corrvname {&numvars} $ ("&varlistquote");
  DO y=1 TO &numvars;
    var{y}=ABS(var{y});
  END;
  DO z=1 TO &numvars;
    IF (var{z}=LARGEST(2,&varlistcomma)) THEN Best1=corrvname{z};
    IF (var{z}=LARGEST(3,&varlistcomma)) THEN Best2=corrvname{z};
    IF (var{z}=LARGEST(4,&varlistcomma)) THEN Best3=corrvname{z};
    IF (var{z}=LARGEST(5,&varlistcomma)) THEN Best4=corrvname{z};
    IF (var{z}=LARGEST(6,&varlistcomma)) THEN Best5=corrvname{z};
  END;
  DROP y z;
  KEEP _NAME_ Best1-Best5;
RUN;

/* MACRO SOLUTION ************************************************************/

%MACRO top5;
  %DO i=1 %TO &numvars;
    DATA COLUMN&i. (KEEP=%SCAN(&varlist.,&i.) _NAME_);
      SET PAPER.CORRS;
      %SCAN(&varlist.,&i.)=ABS(%SCAN(&varlist.,&i.));
    RUN;

    PROC SORT DATA=COLUMN&i.;
      BY DESCENDING %SCAN(&varlist.,&i.);
    RUN;

    DATA COLUMN&i.;
       SET COLUMN&i.;
       IF (_N_ LE 6);
    RUN;
  %END;

  DATA ALLCOLUMNS;
    SET COLUMN:;
  RUN;

  DATA %DO i=0 %TO 5;
         CTDATA&i. (KEEP= Best&i. c)
       %END;
       ;
    SET ALLCOLUMNS;
    x=MOD(_N_,6)-1;
    IF x=-1 THEN x=5;
    c=CEIL(_N_/6);
    %DO i=0 %TO 5;
      IF x=&i. THEN
        DO;
          Best&i. = _NAME_;
          OUTPUT CTDATA&i.;
        END;
    %END;
  RUN;

  DATA PAPER.CTABMACRO (RENAME=(best0=Variable));
```

```
    MERGE  %DO i=0 %TO 5;
              CTDATA&i. (KEEP= Best&i. c)
           %END;
           ;
    BY c;
    DROP c;
  RUN;

  PROC SORT DATA=PAPER.CTABMACRO;
    BY Variable;
  RUN;

  PROC DATASETS NOLIST;
    DELETE COLUMN:;
  QUIT;
%MEND;

%top5;

/* PROC ODS SOLUTION ********************************************************/

ODS OUTPUT PEARSONCORR=PAPER.CTABODS (KEEP=Variable best2-best6
           RENAME=(best2=Best1 best3=Best2 best4=Best3 best5=Best4 best6=Best5));

PROC CORR DATA=PAPER.REPORTED NOSIMPLE BEST=6 NOMISS OUTP=CORRS;
  VAR &varlist;
RUN;

ODS LISTING;
ODS TRACE OFF;
```

ACKNOWLEDGEMENTS

Thank you to Suzanne Dorinski and Mary Ann Koller who tirelessly withstood John's seemingly endless barrage of questions.

CONTACT INFORMATION

SAS is learned largely by doing. Feel free to contact the presenting author with your questions or comments.

John Barrow
Email: john.barrow@census.gov
Phone: (301) 763-9967