Paper 059-31

A Better Means — The ODS Data Trap

Myra A. Oltsik, Mediamark Research Inc., White Plains, NY, USA
Peter Crawford, Crawford Software Consultancy Limited, Croydon, UK

ABSTRACT

When we check the statistics of the numeric variables in a dataset, we want to keep those results in another dataset that we can reference over the course of a project. But this is not easily accomplished. Submit the following SAS®[1] statement –

```
proc means data=sashelp.class; var Age Height Weight; run;
```

– and the list output shows you six columns of information for the three variable rows. If you were to use ODS to create a dataset of this output by adding the following statement before it –

```
ods output summary=means_summary;
```

– you might expect to get a dataset with six variables and three rows of data. You don't. Instead, as of SAS® v9.1.3, you get a dataset with just one record and 18 variables, i.e., one variable for each combination of input variable and column. Instead of waiting for SAS® to fix this ODS problem, we created a macro to produce the results we needed. Along the way, by adding some optional parameters, we think we've built a better "data trap".

INTRODUCTION

The output for the proc means statement above looks like this:

```
Variable     N            Mean         Std Dev         Minimum         Maximum
------------------------------------------------------------------------------
Age         19      13.3157895       1.4926722      11.0000000      16.0000000
Height      19      62.3368421       5.1270752      51.3000000      72.0000000
Weight      19     100.0263158      22.7739335      50.5000000     150.0000000
------------------------------------------------------------------------------
```

This is exactly what we'd like our dataset to look like internally. But if we add the ODS statement and then run –

```
proc print data=means_summary;
run;
```

– we get the following instead:

```
Obs    VName_Age    Age_N      Age_Mean     Age_StdDev       Age_Min

 1     Age          19     13.315789474   1.4926721594            11

Obs       Age_Max   VName_Height   Height_N    Height_Mean   Height_StdDev

 1             16   Height             19     62.336842105   5.1270752466

Obs    Height_Min    Height_Max   VName_Weight   Weight_N    Weight_Mean

 1           51.3            72   Weight             19     100.02631579
```

[1] 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 registered trademarks or trademarks of their respective companies.

```
                Weight_
Obs              StdDev      Weight_Min      Weight_Max

 1    22.773933494             50.5             150
```

This does us no good. We can't use it to merge with other data, nor can we make comparisons between the original variables. We need a better "data trap".

START WITH THE STANDARDS

SAS has always supplied a way to output the results of PROC MEANS to a dataset:

```
proc means data=SASHELP.CLASS noprint;
   output
      out=var_means(drop=_FREQ_ _TYPE_)
   ;
run;
```

produces:

VIEWTABLE: SUMMARY STATISTICS

	STAT	Age	Height	Weight
1	N	19	19	19
2	MIN	11	51.3	50.5
3	MAX	16	72	150
4	MEAN	13.315789474	62.336842105	100.02631579
5	STD	1.4926721594	5.1270752466	22.773933494

We have to use PROC TRANSPOSE to get the dataset in the structure needed:

```
proc transpose data=var_means out=trans_mean(rename=(_NAME_=NAME));
   id _STAT_;
run;
```

VIEWTABLE: Work.Trans_mean

	NAME	N	MIN	MAX	MEAN	STD
1	Age	19	11	16	13.315789474	1.4926721594
2	Height	19	51.3	72	62.336842105	5.1270752466
3	Weight	19	50.5	150	100.02631579	22.773933494

If this is all you need, a macro fix isn't really necessary. But what if, instead of the Standard Deviation (STD), you want the SUM? The following code produces errors in the log:

```
proc means data=SASHELP.CLASS noprint;
   output
      out=var_means(drop=_FREQ_ _TYPE_)
      N=
      Min=
      Max=
      Mean=
      Sum=
   ;
run;

WARNING: Variable Age already exists on file WORK.VAR_MEANS.
WARNING: Variable Height already exists on file WORK.VAR_MEANS.
WARNING: Variable Weight already exists on file WORK.VAR_MEANS.
```

and the dataset contains only the N statistics:

VIEWTABLE: Work.Var_means

	Age	Height	Weight
1	19	19	19

One way to get the standard statistics for more than one variable is to specify the variable names for each statistic as follows:

```
proc means data=SASHELP.CLASS noprint;
   var
      Age Height Weight
   ;
   output
      out=var_means(drop=_FREQ_ _TYPE_)
      N(Age Height Weight)    = N_Age N_Height N_Weight
      Min(Age Height Weight)  = Min_Age Min_Height Min_Weight
      Max(Age Height Weight)  = Max_Age Max_Height Max_Weight
      Mean(Age Height Weight) = Mean_Age Mean_Height Mean_Weight
      Sum(Age Height Weight)  = Sum_Age Sum_Height Sum_Weight
   ;
run;
```

The AUTONAME option is available for PROC MEANS, but any variable whose name is longer than 23 characters can get cut off when the name of the statistic is added by AUTONAME. This makes the results less easy to combine and compare, so the output variables should be explicitly stated in this example. Even so, instead of producing a dataset with a row for each variable – Age, Height, and Weight – you get a dataset with just one record and 15 variables:

VIEWTABLE: Work.Var_means

	N_Age	N_Height	N_Weight	Min_Age	Min_Height	Min_Weight	Max_Age	Max_H
1	19	19	19	11	51.3	50.5	16	

etc. And the more variables analyzed, the more cumbersome the process becomes.

Another way to get these statistics for more than one variable is to create one output file for each statistic:

```
proc means data=SASHELP.CLASS noprint;
   var
      Age Height Weight
   ;
   output out=var_n(drop=_FREQ_ _TYPE_) N=;
   output out=var_min(drop=_FREQ_ _TYPE_) Min=;
   output out=var_max(drop=_FREQ_ _TYPE_) Max=;
   output out=var_mean(drop=_FREQ_ _TYPE_) Mean=;
   output out=var_sum(drop=_FREQ_ _TYPE_) Sum=;
run;
```

This still requires each dataset to be transposed and then merged, or set into one dataset and then transposed. The process begs for a macro to handle it.

A Better Way

The macro %***better_means***, at minimum, requires only the dataset name, and assumes that dataset is found in the WORK library. Invoking the macro,

```
%better_means(data=class_info);
```
[2]

[2] See Appendix for CLASS_INFO sample dataset. This dataset is our own expansion of the dataset SASHELP.CLASS which comes with SAS.

will, by default, create a new dataset in the work library called CLASS_INFO_MEANS. That dataset will include all the PROC MEANS statistics[3] excluding SUMWGT, which requires a weight variable. It will be sorted by VARNUM, the order that the numeric variables appear in the dataset. In addition, the dataset will have a field called PCT_POP that gives the percentage of the records that are non-missing. Finally, the macro prints the dataset to the output window.

MEANS FOR class_info

Obs	VARNUM	NAME	N	PCT_POP	MEAN	STD	MIN	MAX	CSS	CV
1	3	Age	19	100%	13.316	1.4927	11.00	16	40.11	11.210
2	4	Height	19	100%	62.337	5.1271	51.30	72	473.16	8.225
3	5	Weight	19	100%	100.026	22.7739	50.50	150	9335.74	22.768
4	6	Spanish_Speaker	17	89%	0.294	0.4697	0.00	1	3.53	159.687
5	7	Takes_Bus	19	100%	0.579	0.5073	0.00	1	4.63	87.617
6	8	Attend_Summer_Camp	18	95%	0.556	0.5113	0.00	1	4.44	92.036
7	9	GPA	19	100%	2.789	0.7325	1.75	4	9.66	26.259
8	10	No_Adults_HH	19	100%	1.526	0.6118	1.00	3	6.74	40.082
9	11	No_Kids_HH	19	100%	2.105	0.9366	1.00	4	15.79	44.488
10	12	Years_in_School_System	18	95%	3.611	1.0369	1.00	5	18.28	28.714

Obs	LCLM	NMISS	P1	P5	P10	P25	P50	P75	P90	P95	P99	QRANGE	RANGE	PROBT
1	12.7220	0	11.00	11.00	11.0	12.0	13.00	15.00	15	16	16	3.00	5.00	0.000000
2	60.2972	0	51.30	51.30	56.3	57.5	62.80	66.50	69	72	72	9.00	20.70	0.000000
3	90.9664	0	50.50	50.50	77.0	84.0	99.50	112.50	133	150	150	28.50	99.50	0.000000
4	0.0952	2	0.00	0.00	0.0	0.0	0.00	1.00	1	1	1	1.00	1.00	0.020061
5	0.3771	0	0.00	0.00	0.0	0.0	1.00	1.00	1	1	1	1.00	1.00	0.000098
6	0.3459	1	0.00	0.00	0.0	0.0	1.00	1.00	1	1	1	1.00	1.00	0.000250
7	2.4981	0	1.75	1.75	2.0	2.0	2.75	3.25	4	4	4	1.25	2.25	0.000000
8	1.2829	0	1.00	1.00	1.0	1.0	1.00	2.00	2	3	3	1.00	2.00	0.000000
9	1.7327	0	1.00	1.00	1.0	1.0	2.00	3.00	4	4	4	2.00	3.00	0.000000
10	3.1860	1	1.00	1.00	2.0	3.0	4.00	4.00	5	5	5	1.00	4.00	0.000000

Obs	STDERR	SUM	KURT	SKEW	T	UCLM	USS	VAR
1	0.34244	253.0	-1.11093	0.06361	38.8847	13.910	3409.00	2.228
2	1.17623	1184.4	-0.13897	-0.25967	52.9971	64.377	74304.92	26.287
3	5.22470	1900.5	0.68336	0.18335	19.1449	109.086	199435.75	518.652
4	0.11391	5.0	-1.16571	0.99361	2.5820	0.493	5.00	0.221
5	0.11637	11.0	-2.11464	-0.34789	4.9749	0.781	11.00	0.257
6	0.12052	10.0	-2.19938	-0.24447	4.6098	0.765	10.00	0.261
7	0.16805	53.0	-1.14773	0.33011	16.5995	3.081	157.50	0.537
8	0.14035	29.0	-0.31188	0.70311	10.8750	1.770	51.00	0.374
9	0.21487	40.0	-0.02682	0.67956	9.7980	2.478	100.00	0.877
10	0.24440	65.0	1.12007	-0.86768	14.7754	4.036	253.00	1.075

If you preferred the dataset be sorted by the variable name, and not printed, you can invoke the macro as follows:

```
%better_means(data=class_info,sort=NAME,print=N);
```

LIMITING THE OUTPUT

If you don't want all the statistics for all numeric variables in the dataset, there are two additional parameters to help you do so:

```
%better_means(data=class_info,stts=n min max mean,varlst=Age
Spanish_Speaker);
```

[3] Descriptive statistics: CSS, CV, KURTOSIS|KURT, LCLM, MAX, MEAN, MIN, N, NMISS, RANGE, SKEWNESS|SKEW, STDDEV |STD, STDERR, SUM, SUMWGT, UCLM, USS, VAR. Quantile statistics: MEDIAN|P50, P1, P5, P10, Q1|P25, Q3|P75, P90, P95, P99, QRANGE. Hypothesis testing: PROBT, T.

Your printed output is:

```
MEANS FOR class_info

Obs    VARNUM    NAME                N    PCT_POP    MIN    MAX      MEAN

 1        3      Age                19      100%      11     16    13.3158
 2        6      Spanish_Speaker    17       89%       0      1     0.2941
```

and the dataset view is:

VIEWTABLE: STATS FOR class_info 09JAN2006:12:27:04.08

	VARNUM	NAME	N	PCT_POP	MIN	MAX	MEAN
1	3	Age	19	100%	11	16	13.315789474
2	6	Spanish_Speaker	17	89%	0	1	0.2941176471

If you provide a weight variable and request all statistics (or include KURT and/or SKEW in the stts= parameter),

```
%better_means3(data=class_info,sort=NAME,print=N,stts=mean min p25 p50 p75
max sumwgt,wghts=WGT);
```

the KURT and SKEW statistics will not be generated. Alternatively, if you no not provide a weight variable and request all statistics (or include SUMWGT in the stts= parameter), the SUMWGT statistic will not be generated. PROC MEANS will produce errors if these steps are not taken.

VIEWTABLE: STATS FOR class_info 20DEC2005:15:27:39.46

	NAME	MEAN	MIN	P25	P50	P75	MAX	SUMWGT
1	Age	13.4537356	11	12	14	15	16	8.201904114
2	Attend_Summer_Camp	0.53547851	0	0	1	1	1	7.966071614
3	GPA	2.74970098	1.75	2.25	2.75	3.25	4	8.201904114
4	Height	62.379648	51.3	57.5	63.5	66.5	72	8.201904114
5	No_Adults_HH	1.51772066	1	1	1	2	3	8.201904114
6	No_Kids_HH	2.07199145	1	1	2	2	4	8.201904114
7	Spanish_Speaker	0.38782242	0	0	0	1	1	7.376517286
8	Takes_Bus	0.53206143	0	0	1	1	1	8.201904114
9	Weight	99.5979721	50.5	84	103	113	150	8.201904114
10	Years_in_School_System	3.62370417	1	3	4	4	5	7.435727801

ADDING CLASS

Some may want to save the statistics by a class variable, and %***better_means*** allows you to do so. The output of

```
%better_means3(data=class_info,stts=n min max mean,varlst=Age GPA,clss=Sex
Spanish_Speaker);
```

will include the _TYPE_ variable, with the overall mean when _TYPE_ = 0. The printed output, if requested, will be printed by _TYPE_ so that each combination of class variables are in their own block.

```
MEANS FOR class_info

_TYPE_=0

                Spanish_
Obs    Sex       Speaker    VARNUM    NAME      N    PCT_POP     MIN      MAX      MEAN

  1                  .         3      Age      19     100%      11.00      16     13.3158
  2                  .         9      GPA      19     100%       1.75       4      2.7895
```

TYPE=1

Obs	Sex	Spanish_ Speaker	VARNUM	NAME	N	PCT_POP	MIN	MAX	MEAN
3		.	3	Age	2	100%	14.00	16.00	15.0000
4		.	9	GPA	2	100%	2.25	3.00	2.6250
5		0	3	Age	12	100%	11.00	15.00	13.0833
6		0	9	GPA	12	100%	1.75	4.00	2.9167
7		1	3	Age	5	100%	11.00	15.00	13.2000
8		1	9	GPA	5	100%	2.00	3.25	2.5500

TYPE=2

Obs	Sex	Spanish_ Speaker	VARNUM	NAME	N	PCT_POP	MIN	MAX	MEAN
9	F	.	3	Age	9	100%	11.00	15	13.2222
10	F	.	9	GPA	9	100%	2.00	4	2.7500
11	M	.	3	Age	10	100%	11.00	16	13.4000
12	M	.	9	GPA	10	100%	1.75	4	2.8250

TYPE=3

Obs	Sex	Spanish_ Speaker	VARNUM	NAME	N	PCT_POP	MIN	MAX	MEAN
13	F	0	3	Age	6	100%	12.00	15.00	13.5000
14	F	0	9	GPA	6	100%	2.00	4.00	2.8750
15	F	1	3	Age	3	100%	11.00	15.00	12.6667
16	F	1	9	GPA	3	100%	2.00	3.25	2.5000
17	M	.	3	Age	2	100%	14.00	16.00	15.0000
18	M	.	9	GPA	2	100%	2.25	3.00	2.6250
19	M	0	3	Age	6	100%	11.00	15.00	12.6667
20	M	0	9	GPA	6	100%	1.75	4.00	2.9583
21	M	1	3	Age	2	100%	13.00	15.00	14.0000
22	M	1	9	GPA	2	100%	2.00	3.25	2.6250

CONCLUSION

When you use ODS with PROC MEANS, what you think you'll see is not what you get. Using this macro will produce a dataset with the means' statistics you want, with each variable's data in it's own record. Its robustness allows you to add weight and class variables to suit your needs.

CONTACT INFORMATION

Myra A. Oltsik
myra.oltsik@gfk.com
VP, Analytic Programming and Databases
Database Marketing/DemandMatch CoE
MediaMark Research Inc.
500 Mamaroneck Ave, Suite 103
Harrison, NY 10528-1600

Peter Crawford
Peter.Crawford@blueyonder.co.uk
Crawford Software Consultancy Limited
31 Sefton Road
Croydon
CR0 7HS
UK

Appendix

SAMPLE DATABASE

```
data class_info;
   infile datalines;
   input
      Name                    $8.
      Sex                     $1.
      Age                     2.
      Height                  4.1
      Weight                  5.1
      Spanish_Speaker         1.
      Takes_Bus               1.
      Attend_Summer_Camp      1.
      GPA                     4.2
      No_Adults_HH            1.
      No_Kids_HH              1.
      Years_in_School_System  1.
      WGT                     11.9
   ;
datalines;
Alfred  M1469.0112.50112.7511 0.766176313
Alice   F1356.5 84.00104.002430.610406102
Barbara F1365.3 98.00013.751350.009657545
Carol   F1462.8102.50112.001210.058077408
Henry   M1463.5102.5 103.002130.589554328
James   M1257.3 83.00114.003240.224384063
Jane    F1259.8 84.50002.502240.887209392
Janet   F1562.5112.50103.002240.494762153
Jeffrey M1362.5 84.01002.001240.270432161
John    M1259.0 99.50002.502150.125581299
Joyce   F1151.3 50.51113.252240.545268647
Judy    F1464.3 90.00112.001150.294710158
Louise  F1256.3 77.01012.001230.559787725
Mary    F1566.5112.01012.251240.886572091
Philip  M1672.0150.0 1 2.251320.235832500
Robert  M1264.8128.00113.002130.322308053
Ronald  M1567.0133.01013.251240.598718166
Thomas  M1157.5 85.00103.752340.222437109
William M1566.5112.00001.751430.500028901
;
run;
```

MACRO PROGRAM

```
/*****************************************************************************/
/* PROGRAM:       better_means                                               */
/* AUTHORS:       Myra A. Oltsik and Peter Crawford                          */
/* ORIGINAL DATE: 12/20/05                                                   */
/* PURPOSE:       Create a dataset with PROC MEANS statistics, with each record being   */
/*                one variable. Print stats if needed, too. Fixes ODS problems.          */
/*                                                                           */
/* NOTE:          This macro has special handling for N, SUMWGT, KURT and SKEW.          */
/*                Also:   STDEV, Q1, MEDIAN, Q3 are referred as STD, P25, P50, P75.      */
/*****************************************************************************/
/*****************************************************************************/
/* MACRO PARAMETERS:                                                         */
/*    required:  none                                                        */
/*    optional:  print   -- whether or not to print results to output        */
/*               data    -- dataset name to be analysed                      */
/*               sort    -- sort order choice of the file of MEANS, by VARNUM or NAME   */
/*               stts    -- indicate which statistics should included in the output     */
/*               varlst  -- list of variables for means if not all numeric vars in file */
/*               clss    -- variable(s) for a class statement                */
/*               wghts   -- variable for a weight statement                  */
/*    defaults:                                                              */
/*               data    -- &syslast  (most recently created data set)       */
/*               print   -- Y                                                */
/*               sort    -- VARNUM                                           */
/*               stts    -- _ALL_                                            */
/*               varlst  -- _ALL_                                            */
/*                                                                           */
/* Created Macro Variables:                                                  */
/*               locals  --  see inline comments at %local statement         */
/* Creates Data Sets                                                         */
/*               results are written to &data._means                         */
/*               many data sets are created in the work library  all prefixed _better_  */
/*               but unless the testing option is set, the work data stes are deleted   */
/*                                                                           */
/* SAMPLES:                                                                  */
/*   %better_means(data=test); print all default statistics in a dataset     */
/*   %better_means(data=sashelp.class,stts=MEAN SUM); print only MEAN and SUM stats     */
/*   %better_means(data=sashelp.gnp,print=N,sort=NAME,stts=MIN MAX,varlst=INVEST        */
/*      EXPORTS); suppress list printing, limit output statistics and variables, and    */
/*      sort on NAME                                                         */
/*   %better_means(data=sasuser.shoes,clss=PRODUCT); run all stats by PRODUCT field     */
/*   %better_means(data=sasuser.weighted,wghts=WGT); run all stats weighted on WGT      */
/*****************************************************************************/
%macro
   better_means(
      data   = &syslast ,
      print  = Y,
      sort   = VARNUM,
      stts   = _ALL_,
      varlst = _ALL_,
      clss   =  ,
      wghts  =  ,
      testing= no,             /* any other value will preserve the _better_: data sets */
/*****************************************************************************/
/* PROVIDE THE COMPLETE PROC MEANS STATISTIC LIST (FROM ONLINE-DOC) IF NONE STATED.     */
/*****************************************************************************/
      _stts  = N MEAN STD MIN MAX CSS CV LCLM NMISS
               P1 P5 P10 P25 P50 P75 P90 P95 P99 QRANGE RANGE
               PROBT STDERR SUM SUMWGT KURT SKEW T UCLM USS VAR
    );
   %local
      vLexist                          /* EXISTENCE OF LABELS ON INPUT DATASET          */
      s                                /* POINTER TO STATISTIC IN THE STATISTICS LIST   */
      stato                            /* HOLDER OF AN INDIVIDUAL STATISTIC NAME :-
                                          USED IN STATISTIC TABLE NAME, AND
                                          USED IN THE IN= VARIABLE DATASET OPTION       */
      full                             /* INDICATOR IN OUTPUT LABEL WHEN ALL STATS USED.*/
   ;
/*****************************************************************************/
/* PUT STATS AND VAR PARAMETER LIST INTO UPPER CASE.                         */
/*****************************************************************************/
   %let varlst = %upcase(&varlst);
   %let stts   = %upcase(&stts);
   %let data   = &data ;                          /* RESOLVE &syslast, WHEN DEFAULTED */
/*****************************************************************************/
/* GET THE NAMES/NUMBERS OF ALL VARIABLES INTO A LOOKUP FORMAT IF SORT ORDER = VARNUM.  */
/*****************************************************************************/
   %if &sort eq VARNUM %then %do;
      proc contents data= &data out= _better_cols noprint;
      run;
      data _better_cntl;
         retain
            FMTNAME '_bm_VN'
            TYPE    'I'
            HLO     'U'
         ;
         set _better_cols( keep= NAME VARNUM  rename=( VARNUM=LABEL ));
         START = upcase( NAME) ;
      run;
      proc format cntlin= _better_cntl;
      run;
   %end;
```

```
/*****************************************************************************/
/* PROCESS STATISTICS CONDITIONS / COMBINATIONS                               */
/*****************************************************************************/
   %if &stts = _ALL_ or %length(&stts) = 0 %then %do;
      %let stts = &_stts ;
      %let full = FULL STATS;
   %end;
   %if  %length(&wghts) %then %do;
      %* remove KURT and Skew  when weights are present;
      %let stts = %sysfunc( tranwrd( &stts, KURT, %str( ) ));
      %let stts = %sysfunc( tranwrd( &stts, SKEW, %str( ) ));
      %let full = STATS ;
   %end;
   %else %do;
      %* remove SUMWGT  when no weights present ;
      %let stts = %sysfunc( tranwrd( &stts, SUMWGT, %str( ) ));
      %let full = STATS ;
   %end;

/*****************************************************************************/
/* RUN PROC MEANS ON VARIABLES WITH OUTPUT FILE FOR EACH STATISTIC REQUESTED. MERGE     */
/* DATASET OF LIST OF NUMERIC VARIABLES AND THEIR VARNUM.                     */
/*****************************************************************************/
   proc means data= &data noprint missing;
      %if &varlst ne _ALL_ & %length(&varlst) %then %do;
         var   &varlst;
      %end;
      %if %length(&clss) %then %do;
         class &clss;
      %end;
      %if %length(&wghts) %then %do;
         weight &wghts;
      %end;
      %let s     =              1  ;
      %let stato = %scan( &stts, 1 );
      %do %while( %length(&stato) > 0 );          /* USING %LENGTH() FOR &STATO WORDS SIGNIFICANT TO %IF/%WHILE */
         output out= _better_&stato &stato= ;
         %let s     = %eval(        &s +1 );
         %let stato = %scan( &stts, &s    );
      %end;
   run;

   data _better_means1;
      length
         _BETTER_ $32.                          /* STATS IDENTITY */
      ;
      set                                       /* ALL THOSE OUTPUT DATASETS FROM PROC MEANS */
      %let    stato = %scan( &stts, 1 );
      %let    s     =               1  ;
      %do %while(    %length(&stato) gt  0 );
         _better_&stato( in= _in_&stato )       /* NEED IN= VARIABLE TO IDENTIFY INPUT DATA */
         %let s     = %eval(        &s +1 );
         %let stato = %scan( &stts, &s    );
      %end;
      ;
      by _TYPE_ &clss;
      %let    stato = %scan( &stts, 1 );        /* GENERATE _BETTER_ TO IDENTIFY EACH ROW OF RESULTS */
      %let    s     =              1  ;
      %do %while( %length(&stato) > 0 );
         if _in_&stato then _BETTER_ = "%upcase( &stato )" ; else
         %let s     = %eval(        &s +1 );
         %let stato = %scan( &stts, &s    );
      %end;
      ;
   run;

   proc transpose data=_better_means1  out=_better_means2;
      by _TYPE_ &clss ;
      id _BETTER_ ;
   run;
/*****************************************************************************/
/* FROM SAS FAQ # 1806: MACRO TO CHECK IF THE VARIABLE EXISTS IN A DATASET.   */
/*****************************************************************************/
   %macro varcheck(varname,dsname);
      %local dsid vindex rc;
      %let dsid = %sysfunc(open(&dsname,is));

      %if &dsid EQ 0 %then %do;
         %put ERROR: (varcheck) The data set "&dsname" could not be found;
      %end;
      %else %do;
         %let vindex = %sysfunc(varnum(&dsid,&varname));
      %end;

      %let rc = %sysfunc(close(&dsid));
      &vindex
   %mend varcheck;

   %let vLexist = %varcheck(_LABEL_,_better_means2);
/*****************************************************************************/
/* CREATE BASIS FOR OUTPUT DATASET BASED ON DIFFERENT CONDITIONS AND PARAMETER CHOICES. */
/*****************************************************************************/
   %macro inL( list, seek )/ des= "Return TRUE, if &seek in &list, blank delimited";
       %sysfunc( indexw( &list, &seek ))
   %mend inL ;
```

```
   %macro now( fmt= datetime21.2 ) / des= "Timestamp";
       %sysfunc( datetime(), &fmt )
   %mend  now;

   data _better_means_out;
      length
         _TYPE_ 3.      ;
      retain                                        /* TO FIX ORDER OF THE FIRST FEW */
         &clss
         %if &sort eq VARNUM %then %do;
            VARNUM
         %end;
         NAME
         %if &vLexist ne 0 %then %do;               /* ADD IF TRANSPOSED DATASET CONTAINS THE LABEL VARIABLE */
            LABEL
         %end;
         %if %inL(&stts,N) %then %do;               /* ADD % NOT MISSING IF STATISTIC "N" REQUESTED */
            N
            PCT_POP
            PCT_DEN
         %end;
      ;
      set _better_means2(rename=(
         _NAME_  = NAME
         %if &vLexist ne 0 %then %do;
            _LABEL_ = LABEL
         %end;
      ));
      %if %inL(&stts,N) %then %do;
         format
            PCT_POP percent.4
         ;
         if NAME = "_FREQ_" then do;
            PCT_DEN = N ;
            delete;
         end;
         else do;
            if PCT_DEN then PCT_POP = N / pct_den ;
         end;
         drop
            PCT_DEN
         ;
      %end;
      %else %do;
         if NAME = "_FREQ_" then delete;
      %end;
      %if &sort eq VARNUM %then %do;
         VARNUM = input(NAME,_bm_VN.);
      %end;
      NAMEU = upcase(NAME) ;
   run;

/*****************************************************************************/
/* CREATE FINAL DATASET WITH ALL STATISTICS, SORTED AS REQUESTED ON INVOCATION.*/
/*****************************************************************************/
   proc sort data= _better_means_out
      out= &data._means(label= "&FULL FOR &data %NOW" drop= NAMEU
      %if %length(&clss) = 0 %then %do;
         _TYPE_
      %end;
      );
      by _TYPE_ &clss &sort;
   run;

/*****************************************************************************/
/* IF PRINTED OUTPUT IS REQUESTED, DO SO HERE.                               */
/*****************************************************************************/
   %if &print = Y %then %do;
      proc print data=&data._means;
         title3 "MEANS FOR &data";
         %if %length(&clss) > 0 %then %do;
            by _TYPE_;
         %end;
      run;
   %end;

   %if &testing = no  %then %do;
/*****************************************************************************/
/* CLEAN UP REMAINING TEMPORARY DATASETS.                                    */
/*****************************************************************************/
      proc datasets lib= work nolist;
         delete    _better_:        ;
      run; quit;
   %end;
%mend better_means;
```