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® 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:

<table>
<thead>
<tr>
<th>Variable</th>
<th>N</th>
<th>Mean</th>
<th>Std Dev</th>
<th>Minimum</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age</td>
<td>19</td>
<td>13.3157895</td>
<td>1.4926722</td>
<td>11.0000000</td>
<td>16.0000000</td>
</tr>
<tr>
<td>Height</td>
<td>19</td>
<td>62.3368421</td>
<td>5.1270752</td>
<td>51.3000000</td>
<td>72.0000000</td>
</tr>
<tr>
<td>Weight</td>
<td>19</td>
<td>100.0263158</td>
<td>22.7739335</td>
<td>50.5000000</td>
<td>150.0000000</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Obs</th>
<th>VName_</th>
<th>Age</th>
<th>Age_N</th>
<th>Age_Mean</th>
<th>Age_StdDev</th>
<th>Age_Min</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Age</td>
<td>19</td>
<td>13.315789474</td>
<td>1.4926721594</td>
<td>11</td>
<td></td>
</tr>
</tbody>
</table>

```
<table>
<thead>
<tr>
<th>Obs</th>
<th>Age_Max</th>
<th>VName_</th>
<th>Height</th>
<th>Height_N</th>
<th>Height_Mean</th>
<th>Height_StdDev</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>16</td>
<td>Height</td>
<td>19</td>
<td>62.336842105</td>
<td>5.1270752466</td>
<td></td>
</tr>
</tbody>
</table>
```

<table>
<thead>
<tr>
<th>Obs</th>
<th>Height_Min</th>
<th>Height_Max</th>
<th>VName_</th>
<th>Weight</th>
<th>Weight_N</th>
<th>Weight_Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>51.3</td>
<td>72</td>
<td>Weight</td>
<td>19</td>
<td>100.02631579</td>
<td></td>
</tr>
</tbody>
</table>

\(^1\) 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:

<table>
<thead>
<tr>
<th><em>STAT</em></th>
<th>Age</th>
<th>Height</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>19</td>
<td>19</td>
<td>19</td>
</tr>
<tr>
<td>2</td>
<td>11</td>
<td>51.3</td>
<td>50.5</td>
</tr>
<tr>
<td>3</td>
<td>16</td>
<td>72</td>
<td>150</td>
</tr>
<tr>
<td>4</td>
<td>13.315789474</td>
<td>62.336842105</td>
<td>100.02331579</td>
</tr>
<tr>
<td>5</td>
<td>1.4926721594</td>
<td>5.1270752466</td>
<td>22.773933494</td>
</tr>
</tbody>
</table>

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;
```

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:

```
NAME | N  | MIN | MAX  | MEAN   | STD    |
-----|----|-----|------|--------|--------|
1    | 19 | 11  | 16   | 13.315789474 | 1.4926721594 |
2    | 19 | 51.3| 72   | 62.336842105  | 5.1270752466  |
3    | 19 | 50.5| 150  | 100.02331579  | 22.773933494  |
```
One way to get the standard statistics for more than one variable is to specify the variable names for each statistic as follows:

```sql
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:

```
<table>
<thead>
<tr>
<th>N_Age</th>
<th>N_Height</th>
<th>N_Weight</th>
<th>Min_Age</th>
<th>Min_Height</th>
<th>Min_Weight</th>
<th>Max_Age</th>
<th>Max_Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>19</td>
<td>19</td>
<td>11</td>
<td>51.3</td>
<td>50.5</td>
<td>16</td>
<td></td>
</tr>
</tbody>
</table>
```

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:

```sql
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,

```sql
%better_means(data=class_info);
```

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 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 |

<table>
<thead>
<tr>
<th>Obs</th>
<th>LCLM</th>
<th>NMISS</th>
<th>P1</th>
<th>P5</th>
<th>P10</th>
<th>P25</th>
<th>P50</th>
<th>P75</th>
<th>P90</th>
<th>P95</th>
<th>P99</th>
<th>QRANGE</th>
<th>RANGE</th>
<th>PROBT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12</td>
<td>0</td>
<td>11.00</td>
<td>11.00</td>
<td>11.00</td>
<td>12.00</td>
<td>13.00</td>
<td>15.00</td>
<td>15</td>
<td>16</td>
<td>16</td>
<td>3.00</td>
<td>5.00</td>
<td>0.000000</td>
</tr>
<tr>
<td>2</td>
<td>60</td>
<td>0</td>
<td>51.30</td>
<td>51.30</td>
<td>56.30</td>
<td>57.50</td>
<td>62.80</td>
<td>66.50</td>
<td>69</td>
<td>72</td>
<td>72</td>
<td>9.00</td>
<td>20.70</td>
<td>0.000000</td>
</tr>
<tr>
<td>3</td>
<td>90</td>
<td>0</td>
<td>50.50</td>
<td>50.50</td>
<td>77.00</td>
<td>84.00</td>
<td>99.50</td>
<td>112.50</td>
<td>133</td>
<td>150</td>
<td>150</td>
<td>28.50</td>
<td>99.50</td>
<td>0.000000</td>
</tr>
<tr>
<td>4</td>
<td>0.0952</td>
<td>0</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1.00</td>
<td>1.00</td>
<td>0.020061</td>
</tr>
<tr>
<td>5</td>
<td>0.3771</td>
<td>0</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1.00</td>
<td>1.00</td>
<td>0.000098</td>
</tr>
<tr>
<td>6</td>
<td>0.3459</td>
<td>1</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1.00</td>
<td>1.00</td>
<td>0.000250</td>
</tr>
<tr>
<td>7</td>
<td>2.4981</td>
<td>0</td>
<td>1.75</td>
<td>1.75</td>
<td>2.00</td>
<td>2.00</td>
<td>2.75</td>
<td>3.25</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>1.25</td>
<td>2.25</td>
<td>0.000000</td>
</tr>
<tr>
<td>8</td>
<td>1.2829</td>
<td>0</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>2.00</td>
<td>2.00</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>1.00</td>
<td>2.00</td>
<td>0.000000</td>
</tr>
<tr>
<td>9</td>
<td>1.7327</td>
<td>0</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>2.00</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>2.00</td>
<td>3.00</td>
<td>0.000000</td>
</tr>
<tr>
<td>10</td>
<td>3.1860</td>
<td>1</td>
<td>1.00</td>
<td>1.00</td>
<td>2.00</td>
<td>3.00</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>4.00</td>
<td>4.00</td>
<td>0.000000</td>
</tr>
</tbody>
</table>

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

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

```sas
%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:

```sas
%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.
MEANS FOR class_info

<table>
<thead>
<tr>
<th>Obs</th>
<th>VARNUM</th>
<th>NAME</th>
<th>N</th>
<th>PCT_POP</th>
<th>MIN</th>
<th>MAX</th>
<th>MEAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>Age</td>
<td>19</td>
<td>100%</td>
<td>11</td>
<td>16</td>
<td>13.3158</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>Spanish_Speaker</td>
<td>17</td>
<td>89%</td>
<td>0</td>
<td>1</td>
<td>0.2941</td>
</tr>
</tbody>
</table>

and the dataset view is:

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

```plaintext
%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 do 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.

PROC MEANS will produce errors if these steps are not taken.

MEANS FOR class_info

<table>
<thead>
<tr>
<th>Obs</th>
<th>Sex</th>
<th>Spanish_Speaker</th>
<th>VARNUM</th>
<th>NAME</th>
<th>N</th>
<th>PCT_POP</th>
<th>MIN</th>
<th>MAX</th>
<th>MEAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>.</td>
<td>3 Age</td>
<td></td>
<td>19</td>
<td>100%</td>
<td>11.00</td>
<td>16</td>
<td>13.3158</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>.</td>
<td>9 GPA</td>
<td></td>
<td>19</td>
<td>100%</td>
<td>1.75</td>
<td>4</td>
<td>2.7895</td>
<td></td>
</tr>
</tbody>
</table>

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

```plaintext
%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 do 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.

**ADDITION CLASS**

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

```plaintext
%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.
### 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 its own record. Its robustness allows you to add weight and class variables to suit your needs.

### CONTACT INFORMATION

**Myra A. Oltisik**  
[myra.oltisik@gfk.com](mailto:myra.oltisik@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](mailto:Peter.Crawford@blueyonder.co.uk)  
Crawford Software Consultancy Limited  
31 Sefton Road  
Croydon  
CR0 7HS  
UK

### Table

<table>
<thead>
<tr>
<th>Obs</th>
<th>Sex</th>
<th>Spanish Speaker</th>
<th>VARNUM</th>
<th>NAME</th>
<th>N</th>
<th>PCT_POP</th>
<th>MIN</th>
<th>MAX</th>
<th>MEAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>.</td>
<td>3 Age</td>
<td>2</td>
<td>100%</td>
<td>2</td>
<td>14.00</td>
<td>16.00</td>
<td>15.0000</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>.</td>
<td>9 GPA</td>
<td>2</td>
<td>100%</td>
<td>2</td>
<td>2.25</td>
<td>3.00</td>
<td>2.6250</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>3 Age</td>
<td>12</td>
<td>100%</td>
<td>1</td>
<td>11.00</td>
<td>15.00</td>
<td>13.0833</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>9 GPA</td>
<td>12</td>
<td>100%</td>
<td>1</td>
<td>1.75</td>
<td>4.00</td>
<td>2.9167</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>3 Age</td>
<td>5</td>
<td>100%</td>
<td>1</td>
<td>11.00</td>
<td>15.00</td>
<td>13.2000</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>9 GPA</td>
<td>5</td>
<td>100%</td>
<td>1</td>
<td>2.00</td>
<td>3.25</td>
<td>2.5500</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>F</td>
<td>.</td>
<td>3</td>
<td>100%</td>
<td>1</td>
<td>11.00</td>
<td>15</td>
<td>13.2222</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>F</td>
<td>.</td>
<td>9</td>
<td>100%</td>
<td>2</td>
<td>2.00</td>
<td>4</td>
<td>2.7500</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>M</td>
<td>.</td>
<td>3</td>
<td>100%</td>
<td>1</td>
<td>11.00</td>
<td>16</td>
<td>13.4000</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>M</td>
<td>.</td>
<td>9</td>
<td>100%</td>
<td>1</td>
<td>1.75</td>
<td>4</td>
<td>2.825</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Obs</th>
<th>Sex</th>
<th>Spanish Speaker</th>
<th>VARNUM</th>
<th>NAME</th>
<th>N</th>
<th>PCT_POP</th>
<th>MIN</th>
<th>MAX</th>
<th>MEAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>F</td>
<td>0 Age</td>
<td>6</td>
<td>100%</td>
<td>2</td>
<td>12.00</td>
<td>15.00</td>
<td>13.5000</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>F</td>
<td>0 GPA</td>
<td>6</td>
<td>100%</td>
<td>2</td>
<td>2.00</td>
<td>4.00</td>
<td>2.8750</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>F</td>
<td>1 Age</td>
<td>3</td>
<td>100%</td>
<td>1</td>
<td>11.00</td>
<td>15.00</td>
<td>12.6667</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>F</td>
<td>1 GPA</td>
<td>3</td>
<td>100%</td>
<td>1</td>
<td>2.00</td>
<td>3.25</td>
<td>2.5000</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>M</td>
<td>.</td>
<td>3</td>
<td>100%</td>
<td>2</td>
<td>14.00</td>
<td>16.00</td>
<td>15.0000</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>M</td>
<td>.</td>
<td>9</td>
<td>100%</td>
<td>2</td>
<td>2.25</td>
<td>3.00</td>
<td>2.6250</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>M</td>
<td>0 Age</td>
<td>6</td>
<td>100%</td>
<td>2</td>
<td>11.00</td>
<td>15.00</td>
<td>12.6667</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>M</td>
<td>0 GPA</td>
<td>6</td>
<td>100%</td>
<td>1</td>
<td>1.75</td>
<td>4.00</td>
<td>2.9583</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>M</td>
<td>1 Age</td>
<td>2</td>
<td>100%</td>
<td>1</td>
<td>13.00</td>
<td>15.00</td>
<td>14.0000</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>M</td>
<td>1 GPA</td>
<td>2</td>
<td>100%</td>
<td>1</td>
<td>2.00</td>
<td>3.25</td>
<td>2.6250</td>
<td></td>
</tr>
</tbody>
</table>
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   M1463.5102.5103.002130.589554328
James   M1257.3 83.00114.002240.224384063
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 90.00112.002110.294710158
Louise  F1256.3 77.01012.001230.559787725
Mary    F1566.5112.00112.252140.886572091
Philip  M1672.0150.0 12.251320.235832500
Robert  M1264.8128.00013.002130.322308053
Ronald  M1567.0133.01013.252140.598718166
Thomas  M1157.5 85.00013.752340.222437109
William M1566.5112.0001.751430.500028901
;
run;
MACRO PROGRAM

/*******************************************************************************/
PROGRAM:   better_means
/* AUTHORS:  Myra A. Oltoksik 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. Does fixes ODS problem.
/* </quotes>
/* 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 analyzed */
/* stts -- sort order choice of the file of MEANS by VARNUM or NAME */
/* varlist -- list of variables for means if not all numeric vars in file */
/* wghts -- variable for a weight statement */

/* defaults: */
/* data -- syslast (most recently created data set) */
/* print -- Y */
/* sort -- VARNUM */
/* varlist -- ALL */
/* wghts -- ALL */

/* Created Macro Variables: */
/* locals -- see inline comments at %local statement */
/* Creates Data Sets */
/* results are written to _better_means */
/* many data sets are created in the work library all prefixed _better_ 
but unless the testing option is set, the work data sets are deleted */

/* SAMPLES: */
/* %better_means(data= &data; 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-HNAME, stts=MIN MAX, varlist- INVEST */
/* %better_means(data=sashelp.gnp, print=N, sort-HNAME, stts=MIN MAX, varlist- INVEST */
/* %export); suppress list printing, limit output statistics and variables, and */
/* %sort on NAME */
/* %better_means(data=sasuser.shoes, class=PRODUCT); run all stats by PRODUCT field */
/* %better_means(data=sashelp.gnp, print=N, sort-HNAME, stts=MIN MAX, varlist- INVEST */
/* %better_means(data=sashelp.gnp, print=N, sort-HNAME, stts=MIN MAX, varlist- INVEST */

%macro
better_means(
  data = &syslast ,
  print = Y,
  sort = VARNUM,
  varlist = ALL,
  class = ,
  wghts = ,
  testing= no, /* any other value will preserve the _better_ data sets */
) ;
%local
vlist=
/* EXISTENCE OF LABELS ON INPUT DATASET */
&
stato=
/* POINTER TO STATISTIC IN THE STATISTICS LIST */
&
full=
/* INDICATOR IN OUTPUT LABEL WHEN ALL STATS USED. */

/* PROVIDE THE COMPLETE PROC MEANS STATISTIC LIST (FROM ONLINE-DOC) IF NONE STATED. */
%local

%if

%then %do;
proc contents data= &data; noprint;
run;
data better_cntl;
retain
FMTNAME 'km.VH' TYPE 'I' HLG 'O';
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 class &clss;
  %if %length(&wghts) then weight &wghts;
  %let s = 1;
  %let stato = %scan(&stts, 1);
  %do %while( %length(&stato) > 0 );
    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 */
  by _TYPE_ &clss;
  %let stato = %scan(&stts, 1);
  %let s = 1;
  %do %while( %length(&stato) > 0 );
    if _in_&stato then _BETTER_ = "%upcase( &stato )";
    %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

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
    ;
    %else %do;
      if NAME = "_FREQ_" then delete;
    %end;
    %if %vLexist ne 0 %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;
    %if %inL(&stts,N) %then %do;
      by _TYPE_ &clss &sort;
    %end;
  run;

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

  /* CLEAN UP REMAINING TEMPORARY DATASETS. */
  proc datasets lib= work nolist;
    delete _better_; 
  run; quit;
%mend better_means;