

User-Defined Functions that Concatenate Statistics for Standard Reporting

Martha Wetzel, Emory University

ABSTRACT

Statisticians often need to report results such as means/confidence intervals, medians/quartiles, and counts/percentages in a format that differs substantially from the data sets produced by SAS®. Academic journals typically expect that tables display related results (e.g., the confidence interval associated with a mean) in a single cell with parentheses around select statistics. However, SAS procedures output data sets containing each statistic in a separate column. As a result, the statistician must either copy and paste results, which introduces an opportunity for error, or spend time writing lengthy concatenation statements. Without proper formatting, concatenation results in dropped trailing zeros, requiring additional manual changes. In order to increase efficiency, custom functions can be used to combine statistics into a character string following a standard pattern. To that end, four custom functions built with SAS Macro language are provided here:

1. **MEANCI**: Returns a concatenated variable of mean and confidence interval in the format “mean (lower confidence level - upper confidence level)”
2. **MEANSTD**: Returns a concatenated variable of mean and standard deviation in the format “mean (standard deviation)”
3. **MEDQ**: Returns a concatenated variable of medians and quartiles in the format “median (Q1, Q3)”
4. **NPCT**: Returns a concatenated variable of counts and percentages in the format “count (percent %)”

INTRODUCTION

This paper provides custom functions built in the SAS macro language that streamline results reporting. These functions concatenate results into frequently requested displays and allow the user to specify the number of decimals reported.

The code presented in this paper uses SAS Macro language to create short macros that can be used in open code similarly to standard functions. The macro language is leveraged to allow for default values for the arguments. The primary difference between standard SAS functions and functions created with macro language is that the latter must be called using a percent sign, e.g., %function.

A simple illustration of a variable created with %MeanSTD, named Mean_STD, is provided in Output 1. This example is based on UNIVARIATE procedure output from the SASHELP.HEART dataset and is described in further detail in a subsequent section.

Output 1: Create Variable Using %MeanSTD

Variable Name	Mean	Standard Deviation	Mean_STD
Height	64.813184701	3.5827073708	64.81 (3.58)
Weight	153.08668076	28.915426091	153.09 (28.92)

USAGE NOTES

The full code for all four functions is provided in the Appendix. The code must be run prior to calling these functions. The most efficient method for utilizing these functions is to save the code in the appendix as a .SAS file. The user should call the macro into their program with a %include statement at the beginning of a program.

The default values for the variable name parameters are the variable names automatically generated in the output from PROC UNIVARIATE and PROC FREQ. However, alternative variable names can be designated in the function calls.

CONTINUOUS VARIABLES

DATA PREPARATION

Data must be summarized prior to employing these functions. The examples below will use output from PROC UNIVARIATE and a DATA step to create confidence intervals (CIs):

```
/* Prepare data by calculating summary statistics */
proc univariate data = sashelp.heart outtable = UnivarOutput noprint;
    var Height Weight;
run;

/* Prepare data by calculating confidence intervals */
data AddCL;
    set UnivarOutput;
    MyLCL = _MEAN_ - 1.96*_STD_;
    MyUCL = _MEAN_ + 1.96*_STD_;
run;
```

%MEDQ

The function %MEDQ creates a string variable that shows the median and quartiles as “Median (Q1, Q3).” The names of input variables have the standard variable names from PROC UNIVARIATE as default values. Example output is shown in Output 2.

Syntax

%MEDQ (<med=, low=, high=, decimals= >)

Optional Arguments

- Med: Name of variable containing the median. Default = _MED_
- Low: Name of variable containing the low value. Default = _Q1_
- High: Name of variable containing the low value. Default = _Q3_
- Round: Y/N, round the results. Default = N
- Decimals: Number of decimals to round to. Default = 0

Example

Use the function:

```
data ForReport;
    set UnivarOutput (keep = _VAR_ _MEDIAN_ _Q1_ _Q3_ _MIN_ _MAX_ );
    MedianResult = %medq (round = Y);
run;
```

Output 2: Results from %MEDQ Function

Variable Name	25th Percentile (Lower Quartile)	50th Percentile (Median)	75th Percentile (Upper Quartile)	MedianResult
Height	62.25	64.5	67.5	65 (62, 68)
Weight	132	150	172	150 (132, 172)

Usage Note

This function can also be used for displaying medians with minimums and maximums:

```
MedianRange = %medq(Low=_MIN_, high=_MAX_);
```

%MEANCI AND %MEANSTD

The functions %MEANCI and %MEANSTD return string results containing the mean and either the CIs or standard deviation (STD) in parentheses.

Syntax

%MEANCI (<Mean=, LCL=, UCL=, Decimals= >)

%MEANSTD(<Mean=, STD=, Decimals= >)

Optional Arguments

%MEANCI

- Mean: Name of variable containing the mean. Default = _MEAN_
- LCL: Name of variable containing the lower confidence level. Default = _LCL_
- UCL: Name of variable containing the upper confidence level. Default = _UCL_
- Decimals: Number of decimals to round to. Default = 2

%MEANSTD

- Mean: Name of variable containing the mean. Default = _MEAN_
- STD: Name of variable containing the STD. Default = _STD_
- Decimals: Number of decimals to round to. Default = 2

Example

```
data ForOutput;  
  set AddCL (keep = _VAR_ _MEAN_ MyUCL MyLCL _STD_ );  
  Mean_CI = %meanCI (LCL=MyLCL, UCL=MyUCL);  
  Mean_STD = %meanstd;  
run;
```

Output 3: Output from %MEANCI and %MEANSTD

Variable Name	Mean	Standard Deviation	MyLCL	MyUCL	Mean_CI	Mean_STD
Height	64.813184701	3.5827073708	57.791078254	71.835291148	64.81 (57.79 - 71.84)	64.81 (3.58)
Weight	153.08668076	28.915426091	96.412445623	209.7609159	153.09 (96.41 - 209.76)	153.09 (28.92)

CATEGORICAL VARIABLES

The %NPCT function is designed to create a variable in the style of "Count (Percent%)."

Syntax

%NPCT (<count=, percent=, decimals=, indecimal= >)

Optional Arguments

- Count: Name of variable containing the count. Default = COUNT
- Percent: Name of the variable containing the percentage. Default = PERCENT
- Decimals: Number of decimals to round to. Default = 2
- Indecimal: (Y/N) This refers to whether the percent variable is stored as a decimal or as a number greater than one. (For example, PROC FREQ outputs the percent variable as a number greater than one). Set to Y if the percent variable is stored as a decimal. Default = Y.

Example

```
data Categorical;
    /* Create example values */
    Count = 10;
    PercentDec = .5; /* 50% stored as decimal */
    PercentGT1 = 50; /* 50% stored as number greater than 1 */
    Result_Dec = %npct(percent=PercentDec);
    Result_GT1 = %npct(percent=PercentGT1, indecimal=N);
run;
```

Output 4: Variables Created with %NPCT

Count	PercentDec	PercentGT1	Result_Dec	Result_GT1
10	0.5	50	10 (50.00%)	10 (50.00%)

RTF OUTPUT

Once the statistics are concatenated into the desired style, it is simple to output a publication-ready table using the REPORT procedure, as shown below:

```
OPTIONS ORIENTATION=PORTRAIT MISSING = "" NODATE;
ODS RTF STYLE=JOURNAL FILE= "C:\My Folder\My Table &SYSDATE..DOC";

PROC REPORT DATA= ForOutput HEADLINE HEADSKIP CENTER
STYLE (REPORT)={ JUST=CENTER} nowd ;
```

```

COLUMNS _VAR_ Mean_CI;
DEFINE _VAR_/order "Variable" order=internal ;
DEFINE Mean_CI/ "Mean (95% CI)" DISPLAY STYLE(COLUMN)={JUST = C};

RUN;

ODS RTF CLOSE;

```

Output 5: PROC REPORT Example

<i>The SAS System</i>	
<i>Variable</i>	<i>Mean (95% CI)</i>
Height	64.81 (57.79 - 71.84)
Weight	153.09 (96.41 - 209.76)

CONCLUSION

Using basic SAS language, summary statistics can be output to SAS data sets that store one statistic in each column. However, many clients prefer publication-ready results, which typically require concatenating multiple columns and displaying certain statistics in parentheses. This paper presents four functions that streamline this process for presenting means, medians, and counts with percentages.

ACKNOWLEDGMENTS

Many thanks for the support, feedback, and user acceptance testing performed by the Pediatric Biostatistics Core at Emory University, comprised of Courtney McCracken, Janet Figueroa, Michael Kelleman, Scott Gillespie, Amanda Thomas, Erika Bettermann, and Ishaan Dave.

CONTACT INFORMATION

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

Martha Wetzel
 Emory University
 Martha.wetzel@emory.edu
<https://github.com/mpwetzel/SAS4Academia>

APPENDIX

Code for all four functions:

```

/* Format means and confidence interval as "mean (LCL - UCL)" */
%macro meanci(mean=_MEAN_, /* Name of variable containing mean */
  lcl=_LCL_, /* Name of variable containing lower confidence level */
  ucl=_UCL_, /* Name of variable containing upper confidence level */
  decimals=2) /* Number of decimal places to round to */

  cat(strip(put(&mean., 8.&decimals.)), " (", strip(put(&lcl.,
  8.&decimals.)), " - ", strip(put(&ucl., 8.&decimals.)), ")");

```

```

%mend meanci;

/* Format means and standard deviation as "mean (STD)" */
%macro meanstd (mean=_MEAN_, /* Name of variable containing mean */
  STD=_STD_, /* Name of variable containing the standard deviation */
  decimals=2) /* Number of decimal places to round to */

  cat(strip(put(&mean., 8.&decimals.)), " (", strip(put(&std.,
8.&decimals.)), " )");

%mend meanstd;

/* Format medians and quartiles/ranges as "median (Q1, Q2)" */
%macro medq (med=_MEDIAN_, /* Name of variable containing the median */
  Low=_Q1_, /* Name of variable containing the low value */
  High=_Q3_, /* Name of variable containing the high value */
  round=N, /* Y/N should the results be rounded */
  decimals=0) /* Number of decimal places to round to */

  %if %upcase(&round.)= N %then %do;
    cat(strip(&med.), " (", strip(&low.), ", ", strip(&high.), " )");
  %end;
  %else %if %upcase(&round.) = Y %then %do;
    cat(strip(put(&med., 8.&decimals.)), " (", strip(put(&low.,
8.&decimals.)), ", ", strip(put(&high., 8.&decimals.)), " )");
  %end;
%mend medq;

/* Format counts and percentages as "N (%)" */
%macro npct (count=COUNT, /* Name of variable containing the counts */
  percent=PERCENT, /* Name of variable containing the percentages */
  decimals=2, /* Number of decimal places to round to */
  indecimal=Y); /* Set to Y if the percentage variable is a decimal, set
to N if percent variable is
greater than 1 (e.g., proc freq output) */

  cat(strip(&COUNT.), " (", strip(put(
  %if %upcase(&indecimal.) = Y %then %do;
    &PERCENT.
  %end;

  %else %do;
    &PERCENT./100
  %end;
  , PERCENT8.&decimals.)), " )");

%mend npct;

```