PROC MEANS versus PROC SQL for Descriptive Statistics Generation of Weighted Data

Keiko I. Powers, Ph.D., J. D. Power and Associates, Westlake Village, CA

ABSTRACT

When sample representativeness is a key issue (e.g., survey data), research projects/analyses are often based on weighted statistics. Many SAS® procedures such as PROC MEANS offer the WEIGHT option, which makes calculation of weighted statistics effortless. It is often the case, however, that researchers need to review both weighted and un-weighted statistics to closely examine the data. In addition, various statistics such as weighted/un-weighted means have to be merged with the original data for further analysis. When considering these various needs, it is important to understand the pros and cons between the approach based on PROC MEANS and one based on PROC SQL so that researchers can choose the option that best fits their analysis goals. The paper compares and contrasts these two approaches with illustrative examples.

INTRODUCTION

For various applied research that relies on sample data, it is often difficult to obtain randomly selected samples with sufficiently large sample sizes due to various reasons, such as cost limitations, unavailability of respondents, etc. For survey research, sampling is often based on stratified sampling and various statistical analyses are based on quantitative approaches with sampling weights. To derive weighted means and related statistics such as confidence intervals for survey data, PROC SURVEYMEAN is the ideal procedure – it is a tailored procedure for survey based analysis, which allows users to calculate such measures as the mean, standard error, and confidence interval with programming codes similar to the PROC MEANS setup. It is the recommended approach for analysis of survey data, as the procedure properly adjusts for various survey specifications, such as strata and sampling weights.

With applied quantitative research, in addition to weighted statistics, researchers often find it useful to examine 'simple' or un-weighted statistics of the sample data. For example, un-weighted means and standard deviations, etc., provide valuable information in assessing the soundness of the sample data. Because of this reason, applied researchers often work on both weighted and un-weighted statistics, and PROC MEANS is a powerful tool with its versatile data generation capability. In addition to PROC MEANS, another approach to derive various statistics values is based on PROC SQL. Though the type of statistics being offered is somewhat limited compared to PROC MEANS, it calculates means, standard deviations, standard errors fairly efficiently with simple commands. These procedures both offer a simple and powerful approach to key statistics generation, but PROC SQL is found to be more efficient in generating un-weighted and weighted key statistics and handling subsequent data management that are essential for applied research. On the other hand, if various additional statistics measures, such as median, percentiles, etc are in need, PROC MEANS is a much better choice. In this paper, pros and cons of approaches based on PROC SQL and PROC MEANS are described in detail with illustrative cases using small example data.

EXAMPLE DATA

Below is a hypothetical data set with three segments (n = 11, 7, 2 for Segment 1, 2, 3), each having a weight of 30, 44, and 57 respectively. Various statistical measures are derived for the variable, SCORE in the following sections. All the analyses using PROC MEANS and PROC SQL (as well as PROC SURVEYMEAN) are based on this data set so that comparisons of the SAS outputs are easier and more meaningful.

```
data d1;
input segment score weight @@;
datalines;
1     5       30
1     5       30
1     5       30
2     6       44
2     7       44
2     6       44
```

```
1     5       30
1     7       30
1     6       30
1     7       30
1     5       30
1     6       30
1     3       30
2     5       44
2     5       44
2     4       44
3     6       57
3     7       57
2     4       44
1     7       30
;
run;
```

COMPARING STATISTICS GENERATION BETWEEN PROC MEANS AND PROC SQL

PROC MEANS – FREQ VERSUS WEIGHT STATEMENT FOR WEIGHTED MEANS AND STANDARD DEVIATIONS

With PROC MEANS, there are two ways to calculate weighted means – one with FREQ and the other with WEIGHT statement. Both statements give the same weighted mean, but they provide different values for other statistics, such as standard deviations. Differences between the two setups are illustrated below with analyses based on the sample data.

The first step calculates un-weighted statistics of SCORE by setting up codes with FREQ or WEIGHT statement excluded -

```
proc means data=d1 stderr;
var score;
output out=mout0;
title 'no weight applied';
run;

proc print data=mout0 noobs;
title 'no weight applied';
run;;
```

This 'no-weight' setup prints the following outputs –

```
no weight applied

The MEANS Procedure

Analysis Variable : score

   Std Error
  ----------
   0.2562380
  ----------
```

```
no weight applied

_TYPE_    _FREQ_    _STAT_     score

   0        20       N        20.0000
   0        20       MIN       3.0000
   0        20       MAX       7.0000
   0        20       MEAN      5.5500
   0        20       STD       1.1459
```

Next step is to derive weighted statistics for SCORE. In the case of calculating weighted means with PROC MEANS, both FREQ and WEIGHT statements provide the same weighted mean. On the other hand, the standard deviation and the standard error are different between the two setups as shown below.

(1) FREQ statement setup

- Source codes -

```
proc means data=d1 stderr;
var score;
freq weight;
output out=mout1;
title 'with freq option';
run;

proc print data=mout1 noobs;
title 'with freq option';
run;
```

- Procedure outputs -

```
with freq option

The MEANS Procedure

Analysis Variable : score

   Std Error
   __________
   0.0403771
   __________

with freq option

_TYPE_    _FREQ_    _STAT_     score

   0        752      N        752.000
   0        752      MIN        3.000
   0        752      MAX        7.000
   0        752      MEAN       5.584
   0        752      STD        1.107
```

(2) WEIGHT statement setup

- Source codes -

```
proc means data=d1 stderr;
```

```
var score;
weight weight;
output out=mout2;
title 'with weight option';
run;

proc print data=mout2 noobs;
title 'with weight option';
run;
```

- Procedure outputs -

```
with weight option

The MEANS Procedure

Analysis Variable : score

   Std Error
  ___________
   0.2538508
  ___________


with weight option

_TYPE_    _FREQ_    _STAT_     score

   0         20      N          20.000
   0         20      MIN         3.000
   0         20      MAX         7.000
   0         20      MEAN        5.584
   0         20      STD         6.961
   0         20      SUMWGT    752.000
```

In both WEIGHT and FREQ statement cases, calculation of the standard deviation starts with the same CSS – corrected sum of squares, or the weighted sum of squares with respect to the weighted mean (see Base SAS® 9.1.3 Procedure Guide for more details). The difference in the standard deviations is due to the default setup difference in the divisor, or the VARDEF option. With the FREQ setup, VARDEF is the weighted counts (or 752, with the current data example), whereas it is (n-1), sample size minus 1, or 19 with the WEIGHT setup. The standard errors of the two setups have the same difference with respect to the default divisor. As a result, the standard deviation with the WEIGHT setup is considerably larger, and the standard error is considerably smaller with the FREQ setup, both providing incorrect statistics in this case. If the analysis purpose is to have un-weighted statistics including the sample size, the mean, standard deviation, and standard error as well as the weighted mean and weighted counts – typically a minimum set of statistics needed for applied research – the results in these example cases indicate the need for several output files from PROC MEANS. In addition, after output files for various statistics are created from multiple PROC MEANS, these data files need to be merged step by step to get the desired end result for further analysis. These data management steps based on the PROC MEANS approach will be described in more detail in a later section of the paper.

PROC SQL – FOR DERIVING VARIOUS UN-WEIGHTED AND WEIGHTED STATISTICS IN ONE PROC SETUP

PROC SQL, though limited in the number of statistics being offered compared to PROC MEANS, has several tailored commands for calculating aggregated values, such as means and standard deviations. The following PROC SQL statements are to calculate sample sizes (both un-weighted and weighted), means (both un-weighted and weighted), standard deviations, and standard errors of the variable SCORE in the example data.

- Source codes -

```
proc sql;
create table sql1 as
select count(score) as sample_size,
mean(score) as regmean,
(sum(score)/sum(1)) as unwtdmean,
std(score) as unwtdstd,
stderr(score) as unwtdstderr,
(sum(score*weight)/sum(weight)) as wtdmean,
sum(weight) as sumwt
from d1;
quit;

proc print data=sql1 noobs;
title 'from proc sql - overall';
run;
```

The PROC SQL above creates statistics output data, SQL1, and below is the printed output from the PROC PRINT step. The SQL1 data file contains the un-weighted count (the sample size), un-weighted mean, un-weighted standard deviation, un-weighted standard error, weighted mean and weighted count. It should be noted that both REGMEAN and UNWTDMEAN provide the same un-weighted mean, though they are based on different PROC SQL codes – i.e., mean(score) versus sum(score)/sum(1). The statistical values below are the same with those from PROC MEANS setups shown in the earlier section.

- Procedure outputs -

```
from proc sql - overall

sample_
  size   regmean  unwtdmean  unwtdstd  unwtdstderr  wtdmean  sumwt

   20      5.55       5.55     1.14593    0.25624    5.58378   752
```

PROC SURVEYMEANS - GENERATING SURVEY MEAN AND STANDARD ERROR

The codes below provide key statistics generation using PROC SURVEYMEANS for survey data with stratified sampling.

- Source codes -

```
proc surveymeans data=d1;
strata segment;
var score;
weight weight;
run;
```

- Procedure outputs -

```
The SURVEYMEANS Procedure

            Data Summary

Number of Strata                   3
Number of Observations            20
Sum of Weights                   752
```

Statistics			
Variable	N	Mean	Std Error of Mean
score	20	5.583777	0.247382

Statistics	
Variable	95% CL for Mean
score	5.06184593 6.10570726

As mentioned earlier, PROC SURVEYMEANS should be used for deriving the correct standard error and the confidence interval for survey data, as they are statistically adjusted for survey specifications, such as strata. On the other hand, if the key focus is to obtain the weighted means and they need to be combined with un-weighted statistics – i.e., the values need to be further processed with other data files, etc, - PROC MEANS or PROC SQL is likely to be a better choice, as they provide various statistics as well as more flexible data management capabilities.

PROC MEANS VERSUS PROC SQL – GENERATING ADDITIONAL STATISTICS

When the research study requires various other statistics, such as median or percentiles, PROC MEANS is more practical than PROC SQL. PROC MEANS offers an extensive list of various statistics, and they can be included in an output file simply by listing the key words, as shown below.

- Source codes -

```
proc means data=d1 noprint;
var score;
output out=medianout0 median=;
run;

proc print data=medianout0 noobs;
title 'no weight - median';
run;
```

- Procedure outputs -

```
no weight - median

_TYPE_    _FREQ_    score

   0        20       5.5
```

The example highlights the simplicity of median data generation with PROC MEANS. There is no comparable setup with PROC SQL, and percentile-related statistics have the same problem with PROC SQL.

COMPARING DATA MERGING STEPS BETWEEN PROC MEANS AND PROC SQL

Another factor to consider when comparing PROC MEANS versus PROC SQL for various un-weighted and weighted statistics is data processing functionality, particularly when the statistics data files are to be merged with other data files. The following examples highlight the simplicity of PROC SQL for this purpose compared to PROC MEANS-based setup.

PROC MEANS – MERGING STATISTICS VALUES WITH ORIGINAL DATA FILE

When working on various analyses using weights, we often need to look at both un-weighted and weighted analysis results to thoroughly understand the data. When PROC MEANS procedures are used, we need multiple procedures and DATA steps to obtain and organize both results. For example, if combining the statistical values with the original data (i.e., remerging) is necessary, we need additional data steps to incorporate all in one data file as shown below with the example data case.

- Source codes -

```
proc transpose data=mout0 out=tout0; var score; id _stat_; run;
proc transpose data=mout2 out=tout2; var score; id _stat_; run;

data pmeanout;
if _n_ = 1 then set tout0
(keep = n mean std rename=(n=sample_size mean = unwtdmean std=unwtdstd));
if _n_ = 1 then set tout2(keep = sumwgt mean rename=(sumwgt=sumwt mean = wtdmean));
set d1;
unwtdstderr = unwtdstd/sqrt(sample_size);
run;

proc print data=pmeanout noobs;
var segment score weight sample_size unwtdmean  unwtdstd unwtdstderr wtdmean sumwt;
title 'from proc means & data step - statistics merged with original data';
run;
```

- Procedure outputs -

from proc means & data steps - statistics merged with original data

segment	score	weight	sample_size	unwtdmean	unwtdstd	unwtdstderr	wtdmean	sumwt
1	5	30	20	5.55	1.14593	0.25624	5.58378	752
1	5	30	20	5.55	1.14593	0.25624	5.58378	752
1	5	30	20	5.55	1.14593	0.25624	5.58378	752
2	6	44	20	5.55	1.14593	0.25624	5.58378	752
2	7	44	20	5.55	1.14593	0.25624	5.58378	752
2	6	44	20	5.55	1.14593	0.25624	5.58378	752
1	5	30	20	5.55	1.14593	0.25624	5.58378	752
1	7	30	20	5.55	1.14593	0.25624	5.58378	752
1	6	30	20	5.55	1.14593	0.25624	5.58378	752
1	7	30	20	5.55	1.14593	0.25624	5.58378	752
1	5	30	20	5.55	1.14593	0.25624	5.58378	752
1	6	30	20	5.55	1.14593	0.25624	5.58378	752
1	3	30	20	5.55	1.14593	0.25624	5.58378	752
2	5	44	20	5.55	1.14593	0.25624	5.58378	752
2	5	44	20	5.55	1.14593	0.25624	5.58378	752
2	4	44	20	5.55	1.14593	0.25624	5.58378	752

3	6	57	20	5.55	1.14593	0.25624	5.58378	752
3	7	57	20	5.55	1.14593	0.25624	5.58378	752
2	4	44	20	5.55	1.14593	0.25624	5.58378	752
1	7	30	20	5.55	1.14593	0.25624	5.58378	752

PROC SQL – NO EXTRA STEP NEEDED FOR MERGING STATISTICS VALUES WITH ORIGINAL DATA

Unlike the PROC MEANS-based approach that required multiple steps to complete data merging, PROC SQL allows us to complete data processing in one PROC setup as shown below. Starting with the PROC SQL setup in the first section, by simply changing the third code line from "SELECT COUNT(SCORE) AS SAMPLE_SIZE," to "SELECT *, COUNT(SCORE) AS SAMPLE_SIZE," (or add '*,' to read in all the data points from data file D1), we can generate an output data file, SQL2, whose data contents are identical to those from PROC MEANS approach, as shown in PROC PRINT output below. In addition to the simple data management procedure, because the formulas for calculating various statistics (e.g., weighted means – 'WTDMEAN' or weighted counts 'SUMWT') are clearly spelled out, this setup is more organized and programmer-friendly.

- Source codes -

```
proc sql;
create table sql2 as
select *, count(score) as sample_size,
mean(score) as regmean,
(sum(score)/sum(1)) as unwtdmean,
std(score) as unwtdstd,
stderr(score) as unwtdstderr,
(sum(score*weight)/sum(weight)) as wtdmean,
sum(weight) as sumwt
from d1;
quit;

proc print data=sql2 noobs;
title 'from proc sql - statistics merged with original data';
run;
```

- Procedure outputs -

from proc sql - statistics merged with original data

segment	score	weight	sample_size	regmean	unwtdmean	unwtdstd	unwtdstderr	wtdmean	sumwt
1	5	30	20	5.55	5.55	1.14593	0.25624	5.58378	752
1	5	30	20	5.55	5.55	1.14593	0.25624	5.58378	752
1	5	30	20	5.55	5.55	1.14593	0.25624	5.58378	752
2	6	44	20	5.55	5.55	1.14593	0.25624	5.58378	752
2	7	44	20	5.55	5.55	1.14593	0.25624	5.58378	752
2	6	44	20	5.55	5.55	1.14593	0.25624	5.58378	752
1	5	30	20	5.55	5.55	1.14593	0.25624	5.58378	752
1	7	30	20	5.55	5.55	1.14593	0.25624	5.58378	752

1	6	30	20	5.55	5.55	1.14593	0.25624	5.58378	752
1	7	30	20	5.55	5.55	1.14593	0.25624	5.58378	752
1	5	30	20	5.55	5.55	1.14593	0.25624	5.58378	752
1	6	30	20	5.55	5.55	1.14593	0.25624	5.58378	752
1	3	30	20	5.55	5.55	1.14593	0.25624	5.58378	752
2	5	44	20	5.55	5.55	1.14593	0.25624	5.58378	752
2	5	44	20	5.55	5.55	1.14593	0.25624	5.58378	752
2	4	44	20	5.55	5.55	1.14593	0.25624	5.58378	752
3	6	57	20	5.55	5.55	1.14593	0.25624	5.58378	752
3	7	57	20	5.55	5.55	1.14593	0.25624	5.58378	752
2	4	44	20	5.55	5.55	1.14593	0.25624	5.58378	752
1	7	30	20	5.55	5.55	1.14593	0.25624	5.58378	752

CONCLUSION

The comparisons between PROC MEANS and PROC SQL in this paper demonstrate the simplicity associated with PROC SQL when data remerging is required after generating un-weighted and weighted statistics. Clearly, PROC MEANS outperforms PROC SQL in the number of statistics being offered for data outputs. Knowing these pros and cons between the two procedures and applying the procedure more appropriate for given analysis goals will help improve programming efficiency and project execution.

REFERENCE

SAS Institute Inc (2004), "SAS Procedures Guide, Version 9.1.3," Cary, NC: SAS Institute Inc.

CONTACT INFORMATION

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

Keiko I. Powers, Ph.D.
J. D. Power and Associates
2625 Townsgate Road
Westlake Village, CA 91361
Work Phone: 805-418-8114
Fax: 805-418-8241
E-mail Address: Keiko.Powers@jdpa.com