# Sample *45698:* Estimating the Variance of a Variable in a Finite Population

## Overview

The finite population variance of a variable provides a measure of the amount of variation in the corresponding attribute of the study population's members, thus helping to describe the distribution of a study variable. Whether you are studying a population's income distribution in a socioeconomic study, rainfall distribution in a meteorological study, or scholastic aptitude test (SAT) scores of high school seniors, a small population variance is indicative of uniformity in the population while a large variance is indicative of a more diverse population. Another use for the population variance is to determine sample size. For example, the U.S. Environmental Protection Agency uses estimated population variances from pilot studies such as the Environmental Monitoring and Assessment Program–Surface Waters Northeast Lakes Pilot study to assist in planning future sampling strategies (Courbois and Urquhart; 2004).

Suppose you have data that were sampled according to some complex survey design. The SURVEYMEANS procedure enables you to estimate finite population totals, means, and ratios in addition to the design-based variances of the estimated quantities, but it does not directly estimate the finite population variance of a variable. However, because a variance can be expressed mathematically as a total, you can easily estimate the finite population variance $S^2$ of a variable by using PROC SURVEYMEANS plus a little SAS programming.

Whenever you estimate a population parameter such as a mean or a variance, you should also report the precision of the estimate. The most commonly reported measure of precision is the variance (or its square root, the standard error). The survey analysis procedures in SAS/STAT software currently provide three different variance estimation methods for complex survey designs: the Taylor series linearization method, the delete-one jackknife method, and the balanced repeated replication (BRR) method. This example demonstrates how to use all three methods to estimate the variance $V(\hat{S}^2)$.

Because the finite population parameter of interest in this example is the variance of a variable, the measure of precision of the estimate is the variance of a variance. Therefore, as you consider the example, it is important to keep in mind the distinction between the two different meanings of the word variance. In one context, a variance is estimated in order to describe the distribution of a variable. A variance used in this context is denoted $S^2$ and its estimator is denoted $\hat{S}^2$. In the other context, a variance is estimated in order to describe the sampling distribution of an estimator. A variance used in this context is denoted $V(\hat{S}^2)$ and its estimator is denoted $\hat{V}(\hat{S}^2)$.

## Analysis

Suppose you want to estimate the variance of a variable $y$ from a finite population using data that were sampled according to some complex survey design. The finite population variance of $y$ is

$$S^2 = \frac{1}{N-1}\sum_{i=1}^{N}(y_i-\bar{y})^2 \qquad (1)$$

where $N$ is the total number of elements in the population, $y_i$ is the $i$th observation of the variable $y$, and $\bar{y}$ is the population mean of $y$. A sample-based estimator of $S^2$ is

$$\hat{S}^2 = \frac{1}{\hat{N}-1}\sum_{k=1}^{n}\frac{(y_k-\hat{\bar{y}})^2}{\pi_k} \qquad (2)$$

where $\hat{N} = \sum_{k=1}^{n}\frac{1}{\pi_k}$ is an estimator of the population total $N$, $\hat{\bar{y}} = \frac{1}{\hat{N}}\sum_{k=1}^{n}\frac{y_k}{\pi_k}$ is an estimator of the population mean, $n$ is the number of elements in the sample, and $\pi_k$ is the probability that element $k$ is observed in the sample.

To estimate $\hat{S}^2$, you first estimate both $\hat{N}$ and $\hat{\bar{y}}$ with PROC SURVEYMEANS. Next, you generate a variable (call it $z$) such that each observation $z_k$ is equal to

$$z_k = \frac{1}{\hat{N}-1}(y_k-\hat{\bar{y}})^2 \qquad k=1,\ldots,n \qquad (3)$$

Now you use PROC SURVEYMEANS to estimate the total of $z$. The estimated weighted total of $z$ is equal to $\hat{S}^2$. However, the variance of the weighted total of $z$ that is computed by PROC SURVEYMEANS, regardless of which VARMETHOD= option you select, is not equal to $\hat{V}(\hat{S}^2)$, the variance of the estimate $\hat{S}^2$. Computing $\hat{V}(\hat{S}^2)$ requires some additional SAS programming.

## Using the Taylor Series Linearization Method to Estimate $V(\hat{S}^2)$

To estimate $V(\hat{S}^2)$ by using the Taylor series linearization method, construct a variable $u$, such that

$$u_k = \frac{(y_k-\hat{\bar{y}})^2-\hat{S}^2}{\hat{N}-1} \qquad (4)$$

where $\hat{S}^2$ is computed as in equation (2). Use PROC SURVEYMEANS to estimate the total (and the variance of the total) of $u$. The total that is computed by PROC SURVEYMEANS is of no interest, but the variance of the total is equal to $\hat{V}(\hat{S}^2)$, the variance of the estimate $\hat{S}^2$ (Särndal, Swensson, and Wretman 1992 , chap. 5.5).

The following steps summarize how you estimate $S^2$, the finite population variance of a variable $y$, and $V(\hat{S}^2)$, the variance of the finite population variance estimator (using the Taylor series linearization method):

1. Use PROC SURVEYMEANS to estimate the sample mean of the variable $y$, and save the estimated mean. PROC SURVEYMEANS also computes the sum of the sampling weights, which is the value of $\hat{N}$ in the analysis. Save that value also; it is used in the construction of $z$.

2. Using the sample mean from step 1, construct the variable $z$ as in equation (3).

3.

Using the sample mean from step 1 and the estimate of $S^2$ obtained in step 3, construct the variable $u$ as in equation (4).

5.

Use PROC SURVEYMEANS to estimate the weighted total of the variable $u$. The estimated variance of this total obtained from PROC SURVEYMEANS is an estimator of the variance of $\hat{S}^2$.

## Example

### Ice Cream Study Data Set

This example uses the IceCreamStudy data set from the example "Stratified Cluster Sample Design" in the chapter "The SURVEYMEANS Procedure" of the *SAS/STAT User's Guide*.

The study population is a junior high school with a total of 4,000 students in grades 7, 8, and 9. In the original example, researchers want to know how much these students spend weekly for ice cream, on the average, and what percentage of students spend at least $10 weekly for ice cream. This example measures the variability of the students' expenditures by estimating $S^2$, the variance of the variable that contains the students' expenditures.

Suppose that every student belongs to a study group and that study groups are formed within each grade level. Each study group contains between two and four students. Table 1 shows the total number of study groups and the total number of students for each grade.

Table 1 Study Groups and Students by Grade

| Grade | Number of Study Groups | Number of Students |
|---|---|---|
| 7 | 608 | 1,824 |
| 8 | 252 | 1,025 |
| 9 | 403 | 1,151 |

It is quicker and more convenient to collect data from students in the same study group than to collect data from students individually. Therefore, this study uses a stratified clustered sample design. The primary sampling units are study groups. The list of all study groups in the school is stratified by grade level. From each grade level, a sample of study groups is randomly selected, and all students in each selected study group are interviewed. The sample consists of eight study groups from the 7th grade, three groups from the 8th grade, and five groups from the 9th grade.

The SAS data set IceCreamStudy saves the responses of the selected students:

```
data IceCreamStudy;
   input Grade StudyGroup Spending Weight @@;
   datalines;
7  34  7 76.0   7  34  7 76.0   7 412  4 76.0   9  27 14 80.6
7  34  2 76.0   9 230 15 80.6   9  27 15 80.6   7 501  2 76.0
9 230  8 80.6   9 230  7 80.6   7 501  3 76.0   8  59 20 84.0
7 403  4 76.0   7 403 11 76.0   8  59 13 84.0   8  59 17 84.0
8 143 12 84.0   8 143 16 84.0   8  59 18 84.0   9 235  9 80.6
8 143 10 84.0   9 312  8 80.6   9 235  6 80.6   9 235 11 80.6
9 312 10 80.6   7 321  6 76.0   8 156 19 84.0   8 156 14 84.0
7 321  3 76.0   7 321 12 76.0   7 489  2 76.0   7 489  9 76.0
7  78  1 76.0   7  78 10 76.0   7 489  2 76.0   7 156  1 76.0
7  78  6 76.0   7 412  6 76.0   7 156  2 76.0   9 301  8 80.6
;
```

Table 2 identifies the variables contained in the data set IceCreamStudy.

Table 2 Variables in IceCreamStudy Data Set

| Variable | Description |
|---|---|
| Grade | Student's grade (strata) |
| StudyGroup | Student's study group (PSU) |
| Spending | Student's expenditure per week for ice cream, in dollars |
| Weight | Sampling weights |

The SAS data set StudyGroup is created to provide PROC SURVEYMEANS with the sample design information shown in Table 1. The variable Grade identifies the strata, and the variable _TOTAL_ contains the total number of study groups in each stratum.

```
data StudyGroups;
   input Grade _total_;
   datalines;
7 608
8 252
9 403
;
```

### Step 1: Compute $\hat{\bar{y}}$ and $\hat{N}$

Use PROC SURVEYMEANS to obtain an estimate of the sample mean. Specify the MEAN and STACKING options in the PROC SURVEYMEANS statement. The STACKING option causes the procedure to create an output data set with a single observation. This table structure makes it easy in later steps to identify the saved stored in the data set Statistics. The data set Summary contains the sum of the sampling weights, the number of strata, and the number of clusters. The sum of the sampling weights is needed to compute $\hat{S}^2$; the number of strata and the number of clusters are used later when computing confidence limits for $\hat{S}^2$.

```
proc surveymeans data=IceCreamStudy mean stacking ;
   weight Weight;
   strata Grade;
   cluster StudyGroup;
   var Spending;
   ods output Statistics = Statistics
              Summary = Summary;
run;
```

The following DATA step saves the sample mean of the variable Spending in a macro variable named Spending_Mean:

```
data _null_;
   set Statistics;
   call symput("Spending_Mean",Spending_Mean);
run;
```

The next DATA step saves the sum of the sampling weights in a macro variable named N, the number of strata in a macro variable named H, and the number of clusters in a macro variable named C:

```
data Summary;
   set Summary;
   if Label1="Sum of Weights" then call symput("N",cValue1);
   if Label1="Number of Strata" then call symput("H",cValue1);
   if Label1="Number of Clusters" then call symput("C",cValue1);
run;
```

**Step 2: Construct the Variable $z$**

Construct the variable $z$ in a DATA step by using the macro variables Spending_Mean and N:

```
data Working;
   set IceCreamStudy;
   z=(1/(&N-1))*(Spending-&Spending_Mean)**2;
run;
```

**Step 3: Estimate the Total of $z$**

Use PROC SURVEYMEANS to estimate the weighted total of the variable $z$. Specify the SUM and STACKING options in the PROC SURVEYMEANS statement. The ODS OUTPUT statement saves the statistics table to a data set named Result.

```
proc surveymeans data = Working sum stacking;
   weight Weight;
   var z;
   ods output Statistics = Result(keep=z_Sum);
run;
```

The following DATA step retrieves the estimated total of z and stores it in a macro variable named Variance. The total of z is equal to $\hat{S}^2$.

```
data _null_;
   set Result;
   call symput("Variance",z_Sum);
run;
```

**Step 4: Construct the Variable $u$**

Construct the variable $u$ by using the macro variables Spending_Mean, N, and Variance:

```
data Taylor;
   set IceCreamStudy;
   u=(1/(&N-1))*((Spending-&Spending_Mean)**2 - &Variance);
run;
```

**Step 5: Estimate the Total of $u$**

Use PROC SURVEYMEANS to estimate the total of the variable $u$. Specify the SUM, VARSUM, TOTAL=, and STACKING options in the PROC SURVEYMEANS statement. The VARSUM option computes the variance of the total. In this step, the computation of interest is the variance of the estimated total rather than the total itself. Therefore, the sampling design must be appropriately represented in the SURVEYMEANS procedure. The TOTAL= option is specified to enable the procedure to apply a finite population correction in the variance computation. The STRATA statement specifies that the strata be identified by the variable Grade, and the CLUSTER statement specifies that cluster membership be identified by the variable StudyGroup. The ODS OUTPUT statement saves the statistics table in a data set named TaylorResult.

```
proc surveymeans data = Taylor sum varsum stacking total=StudyGroups;
   strata Grade;
   cluster StudyGroup;
   weight Weight;
   var u;
   ods output Statistics = TaylorResult;
run;
```

that is symmetric about the estimated parameter. Confidence intervals constructed in this manner have good coverage properties, however negative lower confidence limits are possible. There are alternative methods for computing confidence intervals that will exclude the possibility of negative lower confidence limits. For example, if the study variable is approximately normally distributed, confidence limits can be computed using a chi-square distribution. Another possibility is to to use the $t$ distribution with the lower confidence limit computed as $max\left(0, \hat{S}^2 + t_{(\alpha/2, df)}\sqrt{\hat{V}(\hat{S}^2)}\right)$. In the simple case that is presented in this example, the latter method is acceptable. However, there are situations where it is not. Whatever method you choose, it is important that the confidence intervals be constructed in a manner that is consistent with any assumptions you make about the underlying data and the parameter estimation method.

```
%let df=%eval(&C - &H);

data TaylorResult;
   set TaylorResult(rename=(u_VarSum=Variance
                   u_StdDev=StdErr));
   Estimate=&Variance;
   LowerCL= Estimate + StdErr*TINV(.025,&df);
   UpperCL= Estimate + StdErr*TINV(.975,&df);
   label Estimate=Population Variance Estimate
         Variance=Variance of Estimate
         StdErr=Standard Error of Estimate
         LowerCL=Lower Confidence Limit
         UpperCL=Upper Confidence Limit;
   Variable='Spending';
run;
```

Use PROC PRINT to print the contents of the data set TaylorResult:

```
title 'Parameter Estimates';

proc print data=TaylorResult label noobs;
   var Variable Estimate Variance StdErr LowerCL UpperCL;
run;

title ;
```

Output 1 displays the results. The estimate of the population variance of the variable Spending is 28.46. The variance of the estimate is 27.87. The standard error of the estimate is 5.28, and the estimated lower and upper 95% confidence limits are 17.05 and 39.86, respectively.

**Output 1 Estimate of Population Variance**

Parameter Estimates

| Variable | Population Variance Estimate | Variance of Estimate | Standard Error of Estimate | Lower Confidence Limit | Upper Confidence Limit |
|---|---|---|---|---|---|
| Spending | 28.4604 | 27.869473 | 5.279155 | 17.0555 | 39.8653 |

## Using the Delete-One Jackknife Method to Estimate $V(\hat{S}^2)$

The delete-one jackknife resampling method of variance estimation deletes one primary sampling unit (PSU) at a time from the full sample to create $R$ replicates, where $R$ is the total number of PSUs. In each replicate, the sample weights of the remaining PSUs are modified by the jackknife coefficient $\alpha_r$. The modified weights are called replicate weights.

If $\hat{S}_r^2$ is the estimate of $S^2$ obtained by using only the data and the replicate weights from the $r$th replicate, the jackknife variance estimate $\hat{V}(\hat{S}^2)$ is

$$\hat{V}(\hat{S}^2) = \sum_{r=1}^{R} \alpha_r \left(\hat{S}_r^2 - \hat{S}^2\right)^2 \qquad (5)$$

with $R - H$ degrees of freedom, where $\alpha_r$ is the jackknife coefficient for the $r$th replicate, $R$ is the number of replicates, and $H$ is the number of strata (or $R - 1$ when there is no stratification). See the section "Jackknife Method" in the chapter "The SURVEYMEANS Procedure" of the *SAS/STAT User's Guide* for more details.

Recall that when you construct $z_k$, you use estimates of $\hat{\bar{y}}$ and $\hat{N}$ that are computed by using the full sample. However, the jackknife variance estimator requires that the $\hat{S}_r^2$ be computed from the $r$th replicate. Thus, the jackknife estimate of the variance of the total of $z$ is not equal to the jackknife estimate of the variance of $\hat{S}^2$.

The following steps summarize how you estimate $\hat{S}^2$, the finite population variance of a variable $y$, and $V(\hat{S}^2)$, the variance of the finite population variance estimator (using the delete-one jackknife method):

1. Use PROC SURVEYMEANS to estimate the sample mean $\hat{\bar{y}}$ and the sum of the weights $\hat{N}$ for the full sample. Save both estimates as they are used in the construction of $z$.

2. Construct $z_k$ as in equation (3), using the full-sample estimates of $\hat{\bar{y}}$ and $\hat{N}$ obtained in step 1.

3. Use PROC SURVEYMEANS to estimate the weighted total of the variable $z$. Save the estimated total, which is the full-sample estimate of the population variance ($\hat{S}^2$). When you estimate the total, specify the VARMETHOD=JACKKNIFE option and the OUTWEIGHTS= and OUTJKCOEFS= method-options in the PROC SURVEYMEANS statement. Both the OUTWEIGHTS= and OUTJKCOEFS= data sets are used in later steps.

5.

For each replicate, using the estimates for $\hat{\bar{y}}_r$ and $\hat{N}_r$ that were obtained in step 4, construct the variable $z$ such that

$$z_{kr} = \frac{1}{\hat{N}_r - 1}(y_{kr} - \hat{\bar{y}}_r)^2 \qquad k = 1,\ldots,n \qquad r = 1,\ldots,R \tag{6}$$

6.

Use PROC SURVEYMEANS to estimate the weighted total of $z$ by replicate, and save the estimates for later use. The estimated weighted total of $z_r$ is equal to $\hat{S}^2_r$ for the $r$th replicate.

7.

Construct a variable (call it $u$) by using the estimates $\hat{S}^2_r$ from step 6, the jackknife coefficients, and the full-sample estimate $\hat{S}^2$ from step 3 such that

$$u_r = \alpha_r\left(\hat{S}^2_r - \hat{S}^2\right)^2 \qquad r = 1,\ldots,R$$

8.

Use PROC SURVEYMEANS to estimate the unweighted total of the variable $u$ from step 7. The estimated unweighted total of $u$ is $\hat{V}(\hat{S}^2)$, the delete-one jackknife estimate of the variance of $\hat{S}^2$.

## Example

This example uses the same IceCreamStudy data set that was described in the section Ice Cream Study Data Set and reproduces the steps described in the section Using the Delete-One Jackknife Method to Estimate $V(\hat{S}^2)$ . Steps 1 and 2 are identical to the first two steps in the previous example but are repeated here for completeness.

### Step 1: Compute $\hat{\bar{y}}$ and $\hat{N}$ for the Full Sample

Use PROC SURVEYMEANS to obtain an estimate of the sample mean. Specify the MEAN and STACKING options in the PROC SURVEYMEANS statement. The WEIGHT statement specifies that the variable Weight contain the sampling weights. The STRATA statement specifies that the variable Grade identifies strata membership. The CLUSTER statement specifies that the variable StudyGroup identifies cluster (or PSU) membership. The ODS OUTPUT statement creates output data sets for the statistics and data summary tables, to be named Statistics and Summary, respectively. The sample mean is stored in the data set Statistics. The data set Summary contains the sum of the sampling weights and the number of strata. The sum of the sampling weights is needed to compute $\hat{S}^2$; the number of strata is used later when computing confidence limits for $\hat{S}^2$.

```
proc surveymeans data=IceCreamStudy mean stacking ;
   weight Weight;
   strata Grade;
   cluster StudyGroup;
   var Spending;
   ods output Statistics = Statistics
              Summary = Summary;
run;
```

The following DATA step saves the sample mean of the variable Spending in a macro variable named Spending_Mean:

```
data _null_;
   set Statistics;
   call symput("Spending_Mean",Spending_Mean);
run;
```

The next DATA step saves the sum of the sampling weights in a macro variable named N and the number of strata in a macro variable named H:

```
data Summary;
   set Summary;
   if Label1="Sum of Weights" then call symput("N",cValue1);
   if Label1="Number of Strata" then call symput("H",cValue1);
run;
```

### Step 2: Construct the Variable $z$ by Using the Full-Sample Estimates of $\hat{\bar{y}}$ and $\hat{N}$

Construct the variable $z$ in a DATA step by using the macro variables Spending_Mean and N:

```
data Working;
   set IceCreamStudy;
   z=(1/(&N-1))*(Spending-&Spending_Mean)**2;
run;
```

### Step 3: Estimate the Total of $z$ for the Full Sample

Use PROC SURVEYMEANS to estimate the weighted total of the variable $z$. Specify the SUM and STACKING options in the PROC SURVEYMEANS statement. Also specify the VARMETHOD=JACKKNIFE option with the OUTJKCOEFS= and OUTWEIGHTS= method-options. The OUTJKCOEFS= method-option saves the jackknife coefficients in a SAS data set named Jkcoefs. The OUTWEIGHTS= method-option saves the replicate weights in a SAS data set named Jkweights.

In this step you must fully specify the sampling design so that the jackknife coefficients and replicate weights are computed correctly. The STRATA statement specifies that the strata be identified by the variable Grade. The CLUSTER statement specifies that the PSUs be identified by the variable StudyGroup. The WEIGHT statement specifies

```
                 varmethod=JACKKNIFE(outjkcoefs=Jkcoefs
                                     outweights=Jkweights);
   strata Grade;
   cluster StudyGroup;
   weight Weight;
   var z;
   ods output Statistics = Result
              VarianceEstimation=VarianceEstimation;
run;

data _null_;
   set Result;
   call symput("Variance",z_Sum);
run;
```

You can see from the "Variance Estimation" table in Output 2 that there are 16 replicates.

**Output 2 Estimate of Population Variance**

The SURVEYMEANS Procedure

| Data Summary | |
|---|---|
| **Number of Strata** | 3 |
| **Number of Clusters** | 16 |
| **Number of Observations** | 40 |
| **Sum of Weights** | 3162.6 |

| Variance Estimation | |
|---|---|
| **Method** | Jackknife |
| **Number of Replicates** | 16 |

The next DATA step retrieves the number of replicates and stores the value in a macro variable named R:

```
data _null_;
   set VarianceEstimation;
   where label1="Number of Replicates";
   call symput("R",cvalue1);
run;

%let R=%eval(&R);
```

The data set Jkcoefs has 16 observations, one for each replicate. The $r$th observation contains the jackknife coefficient for the $r$th replicate. The data set Jkweights contains the original variables from the IceCreamStudy data set and 16 new variables named RepWgt_1 through RepWgt_16; there are $n = 40$ observations.

### Step 4: Compute $\hat{\bar{y}}$ and $\hat{N}$ for Replicate Samples

Before computing $\hat{\bar{y}}_r$ and $\hat{N}_r$, use the following DATA step to convert the data set Jkweights from wide form to long form; doing so enables you to use BY-group processing with PROC SURVEYMEANS.

```
data Long(drop= RepWt_1 - RepWt_&R Z);
  set Jkweights;
  array num (*) RepWt_1 - RepWt_&R;
  do replicate=1 to dim(num);
    Jkweight=num(replicate);
   output;
  end;
run;
```

The data set Long has $40 \times 16 = 640$ observations. There are 16 copies of the original variables from the IceCreamStudy data set stacked on top of each other, and each copy is identified by the variable Replicate. Instead of the 16 replicate weight variables, RepWgt_1 through RepWgt_16, there is now one variable, Jkweight, which is constructed by stacking the variables RepWgt_1 through RepWgt_16 on top of each other. Thus, the first 40 observations contain a copy of the original variablesand the contents of RepWgt_1, and the variable Replicate has a value of 1. The second 40 observations contain a copy of the original variables and the contents of RepWgt_2, and the variable Replicate has a value of 2. The remaining observations are constructed and identified similarly.

Next, sort the data set Long by Replicate:

```
proc sort data=Long out=Long;
   by Replicate;
run;
```

Use PROC SURVEYMEANS to estimate the mean of Spending by Replicate. Doing so produces the estimates of $\bar{y}_r$ and $N_r$ for each replicate. The WEIGHT statement specifies that the sampling weights be contained in the variable Jkweight. The ODS OUTPUT statement saves the sample means ($\hat{\bar{y}}_r$) in a SAS data set named JKMeans and saves the sum of the replicate weights ($\hat{N}_r$) in a data set named JKN. By default, the means are stored in a variable named Mean and the sum of the replicate weights are stored in a variable named N.

```
   by Replicate;
   ods output Statistics = JKMeans(keep=Replicate Mean)
              Summary = JKN;
run;
```

**Step 5: Construct the Variable $z$ for the Replicate Samples**

Before you can construct the variable $z$ for the replicate samples, you must merge the data sets JKMeans and JKN with Long, by Replicate:

```
proc sort data=JKMeans out=JKMeans;
   by Replicate;
run;

data JKN(keep=N replicate );
   set JKN(rename=(nvalue1=N));
   where Label1="Sum of Weights";
run;

proc sort data=JKN out=JKN;
   by Replicate;
run;

data Long;
   merge Long JKN JKMeans;
   by Replicate;
run;
```

Now construct the variable $z$ using the merged data set:

```
data Long;
   set Long;
   z=(1/(N-1))*(Spending-Mean)**2;
run;
```

**Step 6: Estimate the Total of $z$ for Replicate Samples**

Use PROC SURVEYMEANS to estimate the total of the variable $z$ by Replicate. The WEIGHT statement specifies that the sampling weights be contained in the variable Jkweight. You do not need to specify the STRATA and CLUSTER statements. The ODS OUTPUT statement saves the estimated totals in the variable JKEstimate in a SAS data set named Statistics. The estimated totals are the estimates $\hat{S}_r^2$ for each replicate.

```
proc surveymeans data=Long sum stacking;
   weight Jkweight;
   var z;
   by Replicate;
   ods output Statistics=Statistics(drop=Z_StdDEV rename=(Z_Sum=JKEstimate));
run;
```

**Step 7: Construct the Variable $u$**

Before you can construct the variable $u$, you must sort and merge, by Replicate, the data sets Statistics and Jkcoefs:

```
proc sort data=Statistics out=Statistics;
   by Replicate;
run;

proc sort data=Jkcoefs out=Jkcoefs;
   by Replicate;
run;

data Statistics;
   merge Statistics Jkcoefs;
   by Replicate;
run;
```

The data set Statistics now contains the jackknife coefficients $\alpha_r$ in the variable JKcoefficients and the estimates $\hat{S}_r^2$ in the variable JKEstimate. Construct the variable $u$ by using these variables and the full-sample estimate $\hat{S}^2$ that is saved in the macro variable Variance:

```
data Statistics;
   set Statistics;
   u=JKcoefficient*(JKEstimate-&Variance)**2;
run;
```

**Step 8: Estimate the Total of $u$**

Use PROC SURVEYMEANS to compute the unweighted total of $u$. Specify the SUM option in the PROC SURVEYMEANS statement. The ODS OUTPUT statement saves the total in a variable named Variance in a SAS data set named JKResult.

```
proc surveymeans data=Statistics sum;
   var u;
   ods output Statistics=JKResult(rename=(sum=Variance));
run;
```

being analyzed (Spending).

```
%let df=%eval(&R-&H);

data JKResult;
   set JKResult;
   StdErr=sqrt(Variance);
   Estimate=&Variance;
   LowerCL= Estimate + StdErr*TINV(.025,&df);
   UpperCL= Estimate + StdErr*TINV(.975,&df);
   label Estimate=Population Variance Estimate
         Variance=Variance of Estimate
         StdErr=Standard Error of Estimate
         LowerCL=Lower Confidence Limit
         UpperCL=Upper Confidence Limit;
   Variable='Spending';
run;
```

Use the PRINT procedure to print the contents of the data set JKResult:

```
title 'Parameter Estimates';

proc print data=JKResult label noobs;
   var Variable Estimate Variance StdErr LowerCL UpperCL;
run;

title ;
```

Output 3 displays the results. The estimate of the population variance for the variable Spending is 28.46. The variance of the estimate is 30.27, and the standard error of the estimate is 5.50. The estimated lower and upper 95% confidence limits are 16.57 and 40.35, respectively.

**Output 3 Estimate of Population Variance**

Parameter Estimates

| Variable | Population Variance Estimate | Variance of Estimate | Standard Error of Estimate | Lower Confidence Limit | Upper Confidence Limit |
|---|---|---|---|---|---|
| Spending | 28.4604 | 30.267500 | 5.50159 | 16.5750 | 40.3459 |

---

## Using the BRR Method to Estimate $V(\hat{S}^2)$

The BRR method requires that the full sample be drawn by using a stratified sample design with two PSUs per stratum. If $H$ is the total number of strata, the total number of replicates $R$ is the smallest multiple of four that is greater than $H$. Each replicate is obtained by deleting one PSU per stratum according to the corresponding Hadamard matrix and adjusting the original weights for the remaining PSUs. The new weights are called replicate weights.

If $\hat{S}^2_r$ is the estimate of $S^2$ obtained by using only the data and the replicate weights from the $r$th replicate, the BRR variance estimate $\hat{V}(\hat{S}^2)$ is

$$\hat{V}(\hat{S}^2) = \frac{1}{R}\sum_{r=1}^{R}\left(\hat{S}^2_r - \hat{S}^2\right)^2 \qquad (7)$$

with $H$ degrees of freedom. See the section "Balanced Repeated Replication (BRR) Method" in the chapter "The SURVEYMEANS Procedure" of the *SAS/STAT User's Guide* for more details.

Recall that when you construct $z_k$, you use estimates of $\hat{\bar{y}}$ and $\hat{N}$ that are computed by using the full sample. However, the BRR variance estimator requires that the $\hat{S}^2_r$ be computed from the $r$th replicate. Thus, the BRR estimate of the variance of the total of $z$ is not equal to the BRR estimate of the variance of $\hat{S}^2$.

The following steps summarize how you estimate $S^2$, the finite population variance of a variable $y$, and $V(\hat{S}^2)$, the variance of the finite population variance estimator (using the BRR method):

1.

Use PROC SURVEYMEANS to estimate the sample mean $\hat{\bar{y}}$ and the sum of the weights $\hat{N}$ for the full sample. Save both estimates for later use: they are used in the construction of $z$.

2.

Construct $z_k$ as in equation (3) by using the full-sample estimates of $\hat{\bar{y}}$ and $\hat{N}$ obtained in step 1.

3.

Use PROC SURVEYMEANS to estimate the weighted total of the variable $z$, and save the estimated total. This total is the full-sample estimate of the population variance ($\hat{S}^2$). When you estimate the total, specify the VARMETHOD=BRR option and the OUTWEIGHTS= method-option in the PROC SURVEYMEANS statement. The OUTWEIGHTS= SAS data set is used in later steps. Also save the number of strata $H$ and the number of replicates $R$ for later use.

4.

For each replicate, use PROC SURVEYMEANS to estimate the sample mean $\hat{\bar{y}}_r$ and the sum of the weights $\hat{N}_r$ by using only the data and replicate weights for the $r$th replicate. Save the estimates for later use.

5.

For each replicate, using the estimates for $\hat{\bar{y}}_r$ and $\hat{N}_r$ that were obtained in step 4, construct the variable $z$ such that

Use PROC SURVEYMEANS to estimate the weighted total of $z$ by replicate, and save the estimates for later use. The estimated weighted total of $z_r$ is equal to $\hat{S}_r^2$ for the $r$th replicate.

7.

Construct a variable (call it $u$) by using the estimates $\hat{S}_r^2$ from step 6, the number of replicates $R$, and the full-sample estimate $\hat{S}^2$ from step 3 such that

$$u_r = \frac{1}{R}\left(\hat{S}_r^2 - \hat{S}^2\right)^2 \qquad r = 1, \ldots, R$$

8.

Use PROC SURVEYMEANS to estimate the unweighted total of the variable $u$ from step 7. The estimated unweighted total of $u$ is $\hat{V}(\hat{S}^2)$, the BRR estimate of the variance of $\hat{S}^2$.

## Example

### The MUNIsurvey Data Set

This example uses the MUNIsurvey data set from the section "Variance Estimation Using Replication Methods" in the chapter "The SURVEYMEANS Procedure" of the *SAS/STAT User's Guide*. The data are not shown here, but a SAS program that generates the data is included in the sample SAS code that you can download for this example.

In the original example, the San Francisco Municipal Railway (MUNI) conducted a survey to estimate the average waiting time for MUNI subway system's passengers. This example estimates the variance of the passengers' waiting time.

The study uses a stratified cluster sample design. Each MUNI subway line is a stratum. The subway lines included in the study are 'J-Church,' 'K-Ingleside,' 'L-Taraval,' 'M-Ocean View,' 'N-Judah,' and the street car 'F-Market & Wharves.' The MUNI vehicles in service for these lines during a day are the primary sampling units. Within each stratum, two vehicles (PSUs) are randomly selected. Then the waiting times of passengers for a selected MUNI vehicle are collected.

The collected data are saved in the SAS data set MUNIsurvey. Table 3 identifies the variables contained in the data set.

Table 3 Variables in MUNIsurvey Data Set

| Variable | Description |
|---|---|
| Line | The MUNI line that a passenger is riding (strata) |
| Vehicle | The vehicle that a passenger is boarding (PSU) |
| Waittime | The time (in minutes) that a passenger waited |
| Weight | Sampling weights |

### Step 1: Compute $\hat{\bar{y}}$ and $\hat{N}$ for the Full Sample

Use PROC SURVEYMEANS to obtain estimates of the sample mean ($\hat{\bar{y}}$) and the sum of the sampling weights ($\hat{N}$) for the full sample. Specify the MEAN and STACKING options in the PROC SURVEYMEANS statement. The WEIGHT statement specifies that the variable Weight contain the sampling weights. The STRATA statement specifies that the variable Line identify stratum membership. The CLUSTER statement specifies that the variable Vehicle identify PSU or cluster membership. The ODS OUTPUT statement produces output data sets for the statistics and data summary tables, to be named Statistics and Summary, respectively. The sample mean is stored in the data set Statistics, and the sum of the sampling weights is stored in the data set Summary.

```
proc surveymeans data=MUNIsurvey mean stacking ;
   weight Weight;
   strata Line;
   cluster Vehicle;
   var Waittime;
   ods output Statistics = Statistics
              Summary = Summary;
run;
```

The following DATA step saves the sample mean ($\hat{\bar{y}}$) of the variable Waittime in a macro variable named Waittime_Mean:

```
data _null_;
   set Statistics;
   call symput("Waittime_Mean",Waittime_Mean);
run;
```

The next DATA step saves the sum of the sampling weights in a macro variable named N and the number of strata in a macro variable named H:

```
data Summary;
   set Summary;
   if Label1="Sum of Weights" then call symput("N",cValue1);
   if Label1="Number of Strata" then call symput("H",cValue1);
run;
```

### Step 2: Construct the Variable $z$ by Using the Full-Sample Estimates of $\hat{\bar{y}}$ and $\hat{N}$

Construct the variable $z$ in a DATA step by using the macro variables Waittime_Mean and N:

```
run;
```

### Step 3: Estimate the Total of $z$ for the Full Sample

Use PROC SURVEYMEANS to estimate the total of the variable $z$. Specify the SUM and STACKING options in the PROC SURVEYMEANS statement. Also specify the VARMETHOD=BRR(OUTWEIGHTS=) option. The OUTWEIGHTS= method-option saves the replicate weights in a SAS data set named BRRweights.

In this step you must fully specify the sampling design so that the replicate weights are computed correctly. The STRATA statement specifies that the strata be identified by the variable Line. The CLUSTER statement specifies that the PSUs be identified by the variable Vehicle. The WEIGHT statement specifies that the full-sample sampling weights be contained in the variable Weight. The ODS OUTPUT statement saves the statistics table to a data set named Estimate and the variance estimation table to a data set named VarianceEstimation.

```
proc surveymeans data=Working sum stacking
                 varmethod=brr(outweights=BRRweights);
   strata Line;
   cluster Vehicle;
   weight Weight;
   var z;
   ods output Statistics = Estimate
              VarianceEstimation=VarianceEstimation;
run;
```

**Output 4 Estimate of Population Variance**

The SURVEYMEANS Procedure

| Data Summary | |
|---|---|
| **Number of Strata** | 6 |
| **Number of Clusters** | 12 |
| **Number of Observations** | 1937 |
| **Sum of Weights** | 143040 |

| Variance Estimation | |
|---|---|
| **Method** | BRR |
| **Number of Replicates** | 8 |

You can see from Output 4 that there are eight replicates and 1,937 observations.

The data set BRRweights contains the original variables from the Munisurvey data set and eight new variables named RepWgt_1 through RepWgt_8.

The following DATA step retrieves the estimated total of the variable $z$ and stores it in a macro variable named Variance. The total of the variable $z$ is equal to $\hat{S}^2$.

```
data _null_;
   set Estimate;
   call symput("Variance",Z_Sum);
run;
```

The next DATA step retrieves the number of replicates and stores the value in a macro variable named R: The number of replicates is used later to construct the variable $u$.

```
data _null_;
   set VarianceEstimation;
   where label1="Number of Replicates";
   call symput("R",cvalue1);
run;

%let R=%eval(&R);
```

### Step 4: Compute $\hat{\bar{y}}$ and $\hat{N}$ for Replicate Samples

Before computing $\hat{\bar{y}}_r$ and $\hat{N}_r$, use the following DATA step to convert the data set BRRweights from wide form to long form; doing so enables you to use BY-group processing with PROC SURVEYMEANS.

```
data Long(drop= RepWt_1 - RepWt_&R Z);
  set BRRweights;
  array num (*) RepWt_1 - RepWt_&R;
  do replicate=1 to dim(num);
    BRRweight=num(replicate);
   output;
  end;
run;
```

The data set Long has $1,937 \times 8 = 15,496$ observations. There are eight copies of the original variables from the Munisurvey data set stacked on top of each other, and each copy is identified by the variable Replicate. Instead of the eight replicate weight variables, RepWgt_1 through RepWgt_8, there is now one variable, BRRweight, which is constructed by stacking the variables RepWgt_1 through RepWgt_8 on top of each other. Thus, the first 1,937 observations contain a copy of the original variables and the contents of RepWgt_1, and the variable Replicate has a value of 1. The second 1,937 observations contain a copy of the original variables and the contents of RepWgt_2, and the variable Replicate has a value of 2. The remaining observations are constructed and identified similarly.

```
run;
```

Use PROC SURVEYMEANS to estimate the mean of Waittime by Replicate. Doing so produces the estimates of $\bar{y}_r$ and $N_r$ for each replicate. The WEIGHT statement specifies that the sampling weights be contained in the variable BRRweight. The ODS OUTPUT statement saves the sample means in a SAS data set named BRRMeans and the sum of the replicate weights in a data set named BRRN.

```
proc surveymeans data=Long mean;
   weight BRRweight;
   var Waittime;
   by Replicate;
   ods output Statistics = BRRMeans(keep=Replicate Mean)
              Summary = BRRN;
run;
```

**Step 5: Construct the Variable $z$**

Before you can construct the variable $z$, you must merge the data sets BRRMeans and BRRN with Long by Replicate:

```
proc sort data=BRRMeans out=BRRMeans;
   by Replicate;
run;

data BRRN(keep=N replicate );
   set BRRN(rename=(nvalue1=N));
   where Label1="Sum of Weights";
run;

proc sort data=BRRN out=BRRN;
   by Replicate;
run;

data Long;
   merge Long BRRN BRRMeans;
   by Replicate;
run;
```

Now construct the variable $z$ using the merged data set:

```
data Long;
   set Long;
   z=(1/(N-1))*(Waittime-Mean)**2;
run;
```

**Step 6: Estimate the Total of $z$ for the Replicate Samples**

Use PROC SURVEYMEANS to estimate the total of the variable $z$ by Replicate. The WEIGHT statement specifies that the variable BRRweight contain the sampling weights. You do not need to specify the STRATA and CLUSTER statements. The ODS OUTPUT statement saves the estimated totals in the variable BRREstimate in a SAS data set named Statistics. The estimated totals are the estimates $\hat{S}_r^2$ for each replicate.

```
proc surveymeans data=Long sum stacking;
   weight BRRweight;
   var z;
   by Replicate;
   ods output Statistics=Statistics(drop=Z_StdDEV
                                    rename=(Z_Sum=BRREstimate));
run;
```

**Step 7: Construct the Variable $u$**

```
data Statistics;
   set Statistics;
   u=(1/&R)*(BRREstimate-&Variance)**2;
run;
```

**Step 8: Estimate the Total of $u$**

Use PROC SURVEYMEANS to compute the unweighted total of $u$. Specify the SUM option in the PROC SURVEYMEANS statement. The ODS OUTPUT statement saves the total in a variable named Variance in a SAS data set named BRRResult.

```
proc surveymeans data=Statistics sum;
   var u;
   ods output Statistics=BRRResult(rename=(sum=Variance));
run;
```

The following DATA step computes the standard error of the estimate and the upper and lower 95% confidence limits. The confidence limits for this example are computed by using a $t$ distribution with H=6 degrees of freedom. The variable Estimate is generated and assigned the estimated value of $\hat{S}^2$, which is stored in the macro variable Variance. The data set is also prepared for printing.

```
data BRRResult;
   set BRRResult;
   Variable='Waittime';
   label Estimate=Population Variance Estimate
         Variance=Variance of Estimate
         StdErr=Standard Error of Estimate
         LowerCL=Lower Confidence Limit
         UpperCL=Upper Confidence Limit;
run;
```

Use the PRINT procedure to print the contents of the data set BRRResult:

```
title 'Parameter Estimates';

proc print data=BRRResult label noobs;
   var Variable Estimate Variance StdErr LowerCL UpperCL;
run;

title ;
```

Output 5 displays the results. The estimate of the population variance for the variable Waittime is 18.02. The variance of the estimate is 2.17, and the standard error of the estimate is 1.47. The estimated lower and upper 95% confidence limits are 14.41 and 21.63, respectively.

**Output 5 Estimate of Population Variance**

Parameter Estimates

| Variable | Population Variance Estimate | Variance of Estimate | Standard Error of Estimate | Lower Confidence Limit | Upper Confidence Limit |
|---|---|---|---|---|---|
| Waittime | 18.0196 | 2.172780 | 1.47404 | 14.4128 | 21.6264 |

## Appendix: Estimating the Finite Population Standard Deviation $\hat{S}$ and Computing $\hat{V}(\hat{S})$ by Using the Delta Method

After you have an estimate of the finite population variance $\hat{S}^2$ of a variable and a design-based estimator of the variance $V(\hat{S}^2)$, you can estimate the finite population standard deviation of the variable and a design-based estimator of its variance by means of a simple transformation. Specifically, an estimator of the finite population standard deviation is

$$S \equiv g\left(S^2\right) = \left(S^2\right)^{\frac{1}{2}}$$

and, by application of the so-called *delta method*, an estimator of the variance of $\hat{S}$ is

$$V(\hat{S}) = V\left(\hat{S}^2\right)\left[g\prime\left(S^2\right)|_{S^2=\hat{S}^2}\right]^2 = V\left(\hat{S}^2\right)\left(\frac{1}{4\hat{S}^2}\right)$$

where $g\prime(S^2)|_{S^2=\hat{S}^2}$ is the derivative of $g(S^2)$ with respect to $S^2$ evaluated at $\hat{S}^2$. Substituting the sample-based estimators $\hat{S}^2$ and $\hat{V}(\hat{S}^2)$ for $S^2$ and $V(\hat{S}^2)$, respectively, yields

$$\hat{S} = \left(\hat{S}^2\right)^{\frac{1}{2}}$$

and

$$\hat{V}(\hat{S}) = \hat{V}\left(\hat{S}^2\right)\left(\frac{1}{4\hat{S}^2}\right)$$

### Example

Consider the BRR example provided in the section Using the BRR Method to Estimate $V(\hat{S}^2)$ . The estimation results are stored in the data set BRRResult. To compute the finite population standard deviation, its variance, and confidence limits, perform the transformations in the following DATA step. Note that the order of the first two assignment statements is critical.

```
data BRRStdDev;
   set BRRResult;
   Variance=(1/(4*Estimate))*Variance;
   Estimate=sqrt(Estimate);
   StdErr=sqrt(Variance);
   LowerCL= Estimate + StdErr*TINV(.025,&H);
   UpperCL= Estimate + StdErr*TINV(.975,&H);
   label Estimate=Population Standard Deviation Estimate;
run;
```

Use the PRINT procedure to print the contents of the data set BRRStdDev:

```
title 'Parameter Estimates';

proc print data=BRRStdDev label noobs;
   var Variable Estimate Variance StdErr LowerCL UpperCL;
run;
```

error of the estimate is 0.17. The estimated lower and upper 95% confidence limits are 3.82 and 4.67, respectively.

**Output 6 Estimate of Population Standard Deviation**

Parameter Estimates

| Variable | Population Standard Deviation Estimate | Variance of Estimate | Standard Error of Estimate | Lower Confidence Limit | Upper Confidence Limit |
|---|---|---|---|---|---|
| Waittime | 4.24495 | 0.030145 | 0.17362 | 3.82011 | 4.66979 |

## References

Courbois, J.-Y. P. and Urquhart, N. S. (2004), “Comparison of Survey Estimates of the Finite Population Variance,” *Journal of Agricultural, Biological, and Environmental Statistics*, 9(2), 236–251.

Särndal, C. E., Swensson, B., and Wretman, J. (1992), *Model Assisted Survey Sampling*, New York: Springer-Verlag.

```
data IceCreamStudy;
   input Grade StudyGroup Spending Weight @@;
   datalines;
7  34  7 76.0   7  34  7 76.0   7 412  4 76.0   9  27 14 80.6
7  34  2 76.0   9 230 15 80.6   9  27 15 80.6   7 501  2 76.0
9 230  8 80.6   9 230  7 80.6   7 501  3 76.0   8  59 20 84.0
7 403  4 76.0   7 403 11 76.0   8  59 13 84.0   8  59 17 84.0
8 143 12 84.0   8 143 16 84.0   8  59 18 84.0   9 235  9 80.6
8 143 10 84.0   9 312  8 80.6   9 235  6 80.6   9 235 11 80.6
9 312 10 80.6   7 321  6 76.0   8 156 19 84.0   8 156 14 84.0
7 321  3 76.0   7 321 12 76.0   7 489  2 76.0   7 489  9 76.0
7  78  1 76.0   7  78 10 76.0   7 489  2 76.0   7 156  1 76.0
7  78  6 76.0   7 412  6 76.0   7 156  2 76.0   9 301  8 80.6
;
data StudyGroups;
   input Grade _total_;
   datalines;
7 608
8 252
9 403
;
proc surveymeans data=IceCreamStudy mean stacking ;
   weight Weight;
   strata Grade;
   cluster StudyGroup;
   var Spending;
   ods output Statistics = Statistics
              Summary = Summary;
run;
data _null_;
   set Statistics;
   call symput("Spending_Mean",Spending_Mean);
run;
data Summary;
   set Summary;
   if Label1="Sum of Weights" then call symput("N",cValue1);
   if Label1="Number of Strata" then call symput("H",cValue1);
   if Label1="Number of Clusters" then call symput("C",cValue1);
run;

data Working;
   set IceCreamStudy;
   z=(1/(&N-1))*(Spending-&Spending_Mean)**2;
run;
proc surveymeans data = Working sum stacking;
   weight Weight;
   var z;
   ods output Statistics = Result(keep=z_Sum);
run;
data _null_;
   set Result;
   call symput("Variance",z_Sum);
run;
data Taylor;
```

```
   strata Grade;
   cluster StudyGroup;
   weight Weight;
   var u;
   ods output Statistics = TaylorResult;
run;
%let df=%eval(&C - &H);

data TaylorResult;
   set TaylorResult(rename=(u_VarSum=Variance
                       u_StdDev=StdErr));
   Estimate=&Variance;
   LowerCL= Estimate + StdErr*TINV(.025,&df);
   UpperCL= Estimate + StdErr*TINV(.975,&df);
   label Estimate=Population Variance Estimate
         Variance=Variance of Estimate
         StdErr=Standard Error of Estimate
         LowerCL=Lower Confidence Limit
         UpperCL=Upper Confidence Limit;
   Variable='Spending';
run;
title 'Parameter Estimates';

proc print data=TaylorResult label noobs;
   var Variable Estimate Variance StdErr LowerCL UpperCL;
run;

title ;
proc surveymeans data=IceCreamStudy mean stacking ;
   weight Weight;
   strata Grade;
   cluster StudyGroup;
   var Spending;
   ods output Statistics = Statistics
              Summary = Summary;
run;
data _null_;
   set Statistics;
   call symput("Spending_Mean",Spending_Mean);
run;
data Summary;
   set Summary;
   if Label1="Sum of Weights" then call symput("N",cValue1);
   if Label1="Number of Strata" then call symput("H",cValue1);
run;

data Working;
   set IceCreamStudy;
   z=(1/(&N-1))*(Spending-&Spending_Mean)**2;
run;
proc surveymeans data=Working sum stacking
                 varmethod=JACKKNIFE(outjkcoefs=Jkcoefs
                                     outweights=Jkweights);
   strata Grade;
   cluster StudyGroup;
   weight Weight;
   var z;
   ods output Statistics = Result
              VarianceEstimation=VarianceEstimation;
run;
data _null_;
   set Result;
   call symput("Variance",z_Sum);
run;
data _null_;
   set VarianceEstimation;
   where label1="Number of Replicates";
   call symput("R",cvalue1);
run;

%let R=%eval(&R);
data Long(drop= RepWt_1 - RepWt_&R Z);
  set Jkweights;
  array num (*) RepWt_1 - RepWt_&R;
  do replicate=1 to dim(num);
    Jkweight=num(replicate);
   output;
  end;
run;
proc sort data=Long out=Long;
   by Replicate;
run;
proc surveymeans data=Long mean;
   weight Jkweight;
   var Spending;
```

```
proc sort data=JKMeans out=JKMeans;
   by Replicate;
run;
data JKN(keep=N replicate );
   set JKN(rename=(nvalue1=N));
   where Label1="Sum of Weights";
run;
proc sort data=JKN out=JKN;
   by Replicate;
run;
data Long;
   merge Long JKN JKMeans;
   by Replicate;
run;
data Long;
   set Long;
   z=(1/(N-1))*(Spending-Mean)**2;
run;
proc surveymeans data=Long sum stacking;
   weight Jkweight;
   var z;
   by Replicate;
   ods output Statistics=Statistics(drop=Z_StdDEV rename=(Z_Sum=JKEstimate));
run;
proc sort data=Statistics out=Statistics;
   by Replicate;
run;
proc sort data=Jkcoefs out=Jkcoefs;
   by Replicate;
run;
data Statistics;
   merge Statistics Jkcoefs;
   by Replicate;
run;
data Statistics;
   set Statistics;
   u=JKcoefficient*(JKEstimate-&Variance)**2;
run;
proc surveymeans data=Statistics sum;
   var u;
   ods output Statistics=JKResult(rename=(sum=Variance));
run;
%let df=%eval(&R-&H);

data JKResult;
   set JKResult;
   StdErr=sqrt(Variance);
   Estimate=&Variance;
   LowerCL= Estimate + StdErr*TINV(.025,&df);
   UpperCL= Estimate + StdErr*TINV(.975,&df);
   label Estimate=Population Variance Estimate
         Variance=Variance of Estimate
         StdErr=Standard Error of Estimate
         LowerCL=Lower Confidence Limit
         UpperCL=Upper Confidence Limit;
   Variable='Spending';
run;
title 'Parameter Estimates';

proc print data=JKResult label noobs;
   var Variable Estimate Variance StdErr LowerCL UpperCL;
run;

title ;
proc format;
   value $line
      F='F-Market & Wharves'
      J='J-Church'
      K='K-Ingleside'
      L='L-Taraval'
      M='M-Ocean View'
      N='N-Judah';
run;
data p;
   input p @@ ;
   weight=int(120/12+120/9+420/10+120/9+360/15)/2;
   datalines;
0.06 0.053 0.04 0.05 0.10 0.09 0.13 0.12 0.02 0.03 0.04
0.05 0.05 0.055 0.01 0.04 0.05 0.001 0.004 0.005 0.002
;

data f1;
   line='F'; vehicle=1;
   do passenger=1 to 65;
      waittime=rantbl(200,0.06,0.053,0.04,0.05,0.10,0.09, 0.13,
run;

data f2;
   line='F'; vehicle=2;
   do passenger=1 to 102;
      waittime=rantbl(103,0.06,0.053,0.04,0.05,0.10,0.09,0.13,
                      0.12,0.02,0.03,0.04,0.05,0.05,0.055,0.01,
                      0.04,0.05,0.001,0.004,0.005,0.002)-1;
      output;
   end;
run;

data f;
   set f1 f2;
   weight=int(70/15+120/6+420/8+120/7+360/15)/2;
run;

data j1;
   line='J'; vehicle=1;
      do passenger=1 to 101;
         waittime=rantbl(2,0.06,0.003,0.04,0.05,0.10,0.09,0.13,
                         0.12,0.12,0.03,0.04,0.05,0.05,0.055,0.03,
                         0.04,0.05,0.001,0.004,0.025,0.002)-1;
         output;
      end;
run;

data j2;
   line='J'; vehicle=2;
   do passenger=1 to 142;
      waittime=rantbl(7,0.06,0.053,0.04,0.09,0.13,0.05,0.10,0.12,
                      0.02,0.03,0.04,0.05,0.05,0.004,0.005,0.002,
                      0.055,0.01,0.04,0.05,0.001)-1;
      output;
   end;
run;

data j;
   set j1 j2;
   weight=int(120/15+120/9+420/10+120/9+360/15)/2;
run;

data k1;
   line='K'; vehicle=1;
   do passenger=1 to 145;
      waittime=rantbl(111,0.06,0.003,0.04,0.05,0.10,0.09,0.13,0.12,
                      0.12,0.03,0.04,0.05,0.05,0.055,0.03,0.04,0.05,
                      0.001,0.004,0.025,0.002)-1;
      output;
   end;
run;

data k2;
   line='K'; vehicle=2;
   do passenger=1 to 180;
      waittime=rantbl(71,0.06,0.053,0.04,0.09,0.13,0.05,0.10,0.12,
                      0.02,0.03,0.04,0.05,0.05,0.004,0.005,0.002,
                      0.055,0.01,0.04,0.05,0.001)-1;
      output;
   end;
run;

data k;
   set k1 k2;
   weight=int(120/15+120/9+420/10+120/9+360/15)/2;
run;

data L1;
   line='L'; vehicle=1;
   do passenger=1 to 135;
      waittime=rantbl(1110,0.06,0.003,0.05,0.05,0.04,0.05,0.10,0.09,
                      0.13,0.12,0.12,0.03,0.04,0.055,0.03,0.04,0.05,
                      0.001,0.004,0.025,0.002)-1;
      output;
   end;
run;

data L2;
   line='L'; vehicle=2;
   do passenger=1 to 185;
      waittime=rantbl(18,0.02,0.03,0.04,0.055,0.09,0.053,0.04,0.09,
                      0.13,0.05,0.10,0.12,0.04,0.05,0.05,0.004,0.005,
                      0.002,0.025,0.01,0.001)-1;
      output;
   end;
```

```
run;

data m1;
   line='M'; vehicle=1;
   do passenger=1 to 139;
      waittime=rantbl(1150,0.06,0.03,0.05,0.05,0.14,0.05,0.10,0.09,0.03,
                      0.12,0.12,0.03,0.04,0.015,0.03,0.02,0.05,0.001,
                      0.004,0.025,0.002)-1;
      output;
   end;
run;

data m2;
   line='M'; vehicle=2;
   do passenger=1 to 203;
      waittime=rantbl(1008,0.03,0.03,0.05,0.055,0.29,0.053,0.04,0.09,
                      0.13,0.05,0.10,0.12,0.04,0.05,0.02,0.004,0.005,
                      0.002,0.015,0.01,0.001)-1;
      output;
   end;
run;

data m;
   set m1 m2;
   weight=int(70/15+120/9+420/10+120/9+360/15)/2;
run;

data n1;
   line='N'; vehicle=1;
   do passenger=1 to 306;
      waittime=rantbl(1150,0.06,0.04,0.06,0.05,0.14,0.05,0.08,0.09,
                      0.03,0.12,0.12,0.03,0.04,0.015,0.03,0.02,0.05,
                      0.001,0.004,0.025,0.002)-1;
      output;
   end;
run;

data n2;
   line='N'; vehicle=2;
   do passenger=1 to 234;
      waittime=rantbl(1008,0.03,0.05,0.05,0.05,0.07,0.053,0.04,0.03,
                      0.23,0.05,0.10,0.08,0.04,0.05,0.02,0.004,0.005,
                      0.012,0.015,0.02,0.001)-1;
      output;
   end;
run;

data n;
   set n1 n2;
   weight=int(120/12+120/7+420/10+120/7+360/12+300/30);
run;

data MUNIsurvey;
   set f j k l m n;
   format line $line.;
run;
proc datasets nolist;
   delete f j k l m n;
run;
proc surveymeans data=MUNIsurvey mean stacking ;
   weight Weight;
   strata Line;
   cluster Vehicle;
   var Waittime;
   ods output Statistics = Statistics
              Summary = Summary;
run;
data _null_;
   set Statistics;
   call symput("Waittime_Mean",Waittime_Mean);
run;
data Summary;
   set Summary;
   if Label1="Sum of Weights" then call symput("N",cValue1);
   if Label1="Number of Strata" then call symput("H",cValue1);
run;

data Working;
   set MUNIsurvey;
   Z=(1/(&N-1))*(Waittime-&Waittime_Mean)**2;
run;
proc surveymeans data=Working sum stacking
                 varmethod=brr(outweights=BRRweights);
   strata Line;
   cluster Vehicle;
```

```
run;
data _null_;
   set Estimate;
   call symput("Variance",Z_Sum);
run;
data _null_;
   set VarianceEstimation;
   where label1="Number of Replicates";
   call symput("R",cvalue1);
run;

%let R=%eval(&R);
data Long(drop= RepWt_1 - RepWt_&R Z);
  set BRRweights;
  array num (*) RepWt_1 - RepWt_&R;
  do replicate=1 to dim(num);
    BRRweight=num(replicate);
   output;
  end;
run;
proc sort data=Long out=Long;
   by Replicate;
run;
proc surveymeans data=Long mean;
   weight BRRweight;
   var Waittime;
   by Replicate;
   ods output Statistics = BRRMeans(keep=Replicate Mean)
              Summary = BRRN;
run;
proc sort data=BRRMeans out=BRRMeans;
   by Replicate;
run;
data BRRN(keep=N replicate );
   set BRRN(rename=(nvalue1=N));
   where Label1="Sum of Weights";
run;
proc sort data=BRRN out=BRRN;
   by Replicate;
run;
data Long;
   merge Long BRRN BRRMeans;
   by Replicate;
run;
data Long;
   set Long;
   z=(1/(N-1))*(Waittime-Mean)**2;
run;
proc surveymeans data=Long sum stacking;
   weight BRRweight;
   var z;
   by Replicate;
   ods output Statistics=Statistics(drop=Z_StdDEV
                                    rename=(Z_Sum=BRREstimate));
run;
data Statistics;
   set Statistics;
   u=(1/&R)*(BRREstimate-&Variance)**2;
run;
proc surveymeans data=Statistics sum;
   var u;
   ods output Statistics=BRRResult(rename=(sum=Variance));
run;
data BRRResult;
   set BRRResult;
   StdErr=sqrt(Variance);
   Estimate=&Variance;
   LowerCL= Estimate + StdErr*TINV(.025,&H);
   UpperCL= Estimate + StdErr*TINV(.975,&H);
   Variable='Waittime';
   label Estimate=Population Variance Estimate
         Variance=Variance of Estimate
         StdErr=Standard Error of Estimate
         LowerCL=Lower Confidence Limit
         UpperCL=Upper Confidence Limit;
run;
title 'Parameter Estimates';

proc print data=BRRResult label noobs;
   var Variable Estimate Variance StdErr LowerCL UpperCL;
run;

title ;
data BRRStdDev;
   set BRRResult;
   UpperCL= Estimate + StdErr*TINV(.975,&H);
   label Estimate=Population Standard Deviation Estimate;
run;
title 'Parameter Estimates';

proc print data=BRRStdDev label noobs;
   var Variable Estimate Variance StdErr LowerCL UpperCL;
run;

title ;
```

This example demonstrates how to use three different variance estimation methods provided in survey analysis procedures in SAS/STAT software for complex survey designs to estimate the variance V(S2).

**Type:** Sample
**Topic:** Analytics ==> Survey Sampling and Analysis
**Date Modified:** 2012-02-16 12:07:52
**Date Created:** 2012-02-16 12:00:12

**Operating System and Release Information**

| Product Family | Product | Host | SAS Release | |
|---|---|---|---|---|
| | | | Starting | Ending |
| SAS System | SAS/STAT | z/OS | | |
| | | OpenVMS VAX | | |
| | | Microsoft® Windows® for 64-Bit Itanium-based Systems | | |
| | | Microsoft Windows Server 2003 Datacenter 64-bit Edition | | |
| | | Microsoft Windows Server 2003 Enterprise 64-bit Edition | | |
| | | Microsoft Windows XP 64-bit Edition | | |
| | | Microsoft® Windows® for x64 | | |
| | | OS/2 | | |
| | | Microsoft Windows 95/98 | | |
| | | Microsoft Windows 2000 Advanced Server | | |
| | | Microsoft Windows 2000 Datacenter Server | | |
| | | Microsoft Windows 2000 Server | | |
| | | Microsoft Windows 2000 Professional | | |
| | | Microsoft Windows NT Workstation | | |
| | | Microsoft Windows Server 2003 Datacenter Edition | | |
| | | Microsoft Windows Server 2003 Enterprise Edition | | |
| | | Microsoft Windows Server 2003 Standard Edition | | |
| | | Microsoft Windows Server 2003 for x64 | | |
| | | Microsoft Windows Server 2008 | | |
| | | Microsoft Windows Server 2008 for x64 | | |
| | | Microsoft Windows XP Professional | | |
| | | Windows 7 Enterprise 32 bit | | |
| | | Windows 7 Enterprise x64 | | |

| | | | | |
|---|---|---|---|---|
| | | Windows 7 Professional 32 bit | | |
| | | Windows 7 Professional x64 | | |
| | | Windows 7 Ultimate 32 bit | | |
| | | Windows 7 Ultimate x64 | | |
| | | Windows Millennium Edition (Me) | | |
| | | Windows Vista | | |
| | | Windows Vista for x64 | | |
| | | 64-bit Enabled AIX | | |
| | | 64-bit Enabled HP-UX | | |
| | | 64-bit Enabled Solaris | | |
| | | ABI+ for Intel Architecture | | |
| | | AIX | | |
| | | HP-UX | | |
| | | HP-UX IPF | | |
| | | IRIX | | |
| | | Linux | | |
| | | Linux for x64 | | |
| | | Linux on Itanium | | |
| | | OpenVMS Alpha | | |
| | | OpenVMS on HP Integrity | | |
| | | Solaris | | |
| | | Solaris for x64 | | |
| | | Tru64 UNIX | | |