Paper 2601-2015

Replication Techniques for Variance Approximation

Taylor Lewis, Joint Program in Survey Methodology (JPSM), University of Maryland

ABSTRACT

Replication techniques such as the jackknife and the bootstrap have become increasingly popular in recent years, particularly within the field of complex survey data analysis. The premise of these techniques is to treat the data set as if it were the population and repeatedly sample from it in some systematic fashion. From each sample, or replicate, the estimate of interest is computed, and the variability of the estimate from the full data set is approximated by a simple function of the variability amongst the replicate-specific estimates. An appealing feature is that there is generally only one variance formula (per method), regardless the underlying quantity being estimated. The entire process can be efficiently implemented after appending a series of replicate weights to the analysis data set. As will be shown, the SURVEY family of SAS/STAT® procedures can be exploited to facilitate both the task of appending the replicate weights and approximating variances.

1. INTRODUCTION

This paper is a foray into the increasingly popular class of variance estimators called *replication techniques* (Rust, 1985; Rust and Rao, 1996). These are tools for approximating, or estimating, the variance of any kind of sample-based point estimate. They are particularly handy in samples characterized by any of the four features of complex survey data (Lewis, 2015). While there have been numerous user-written macros presented over the years to conduct these techniques (e.g., Hawkes, 1997; Bienias, 2001; Berglund, 2004), the release of SAS® Version 9.2 ushered in new capabilities that have greatly simplified the amount of syntax required (Mukhopadhyay et al., 2008).

The fundamental idea behind replication is to treat the data set as if it were the population and repeatedly sample from it in some systematic fashion. From each sample, or *replicate*, the quantity of interest is estimated. The variance of the full data set estimate is then estimated as a simple function of the variability amongst the replicate-specific estimates. An appealing feature is that there is generally only one variance formula (per method), no matter the quantity being estimated. As we will see, the process can be efficiently implemented by appending a series of *replicate weights* to the analysis data set.

This paper begins with a brief background section introducing a complex survey data set, one characterized by stratification, clustering, and weighting, housing data from a fictitious mathematical aptitude survey of 16 high school students. Subsequent examples in this paper use this data set to illustrate three of the most commonly used replication techniques in practice: (1) balanced repeated replication; (2) the jackknife; and (3) the bootstrap. In this author's view, a data set of manageable size is essential for the uninitiated to comprehend these techniques, since it allows one to wholly visualize the replicate weights and how they are constructed. We also touch on the multivariate generalization of replication techniques by demonstrating how they can be used to estimate the covariance matrix of linear model parameters. The paper concludes with a brief discussion on degrees of freedom considerations.

2. AN EXAMPLE SURVEY DATA SET

To facilitate the exposition of complex survey data and replication techniques, let us introduce a hypothetical mathematics aptitude survey of high school students. Suppose that a particular high school consists of four classes, grades 9 through 12, and that the students within each class are assigned to one of 10 homerooms, not necessarily of equal size. The ultimate objective is to select a sample of students on which to administer a mathematics examination at the start and end of school year, but supplemental information on these students will also be collected, such as whether or not they received any kind of mathematics tutoring.

Assume that class standing was used to stratify the student population. To make the data collection process run smoother, a sample of two homerooms was selected within each class and, subsequently, two students were selected within each sampled homeroom. Hence, a multi-stage, clustered sample

design was employed. More specifically, this is an example of a two-PSU-per-stratum sample design, which is very common in applied survey research. Although a total of 16 students from the high school were sampled, the overall number of *primary sampling units* (PSUs) is 8 (2 homerooms x 4 grades). A visualization is provided in Figure 1. The population of clusters is represented by the collection of boxes, separated into four rows that delineate four population strata. A colored box indicates the given cluster (i.e., homeroom) was sampled.

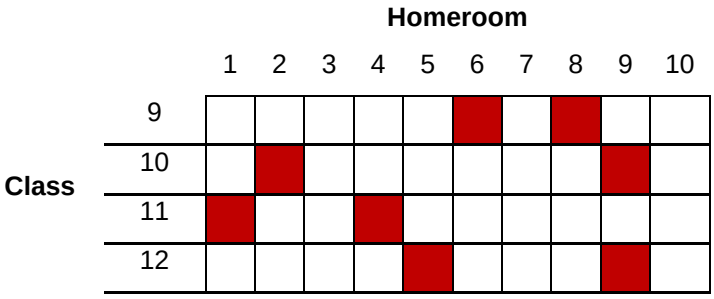

Figure 1: Visual Representation of the Stratified, Clustered Population and Sample for the Hypothetical Mathematics Aptitude Survey

Program 1 reads in data from this hypothetical survey effort into a temporary SAS data set named TEST. Features of the complex sample design are maintained by the following variables:

- CLASS – the stratification factor, the high school grade of the student, ranging from 9 to 12.
- HOMEROOM – the cluster identifier, a student's homeroom, coded as either a 1 or 2 that, in combination with codes of variable CLASS, defines the eight distinct PSUs.
- WEIGHT – the base weight, or inverse of the student's selection probability.

The numeric variables GRADE1 and GRADE2 contain the student's mathematics examination results at the start and end of the school year, respectively. The maximum grade on the examination is 100. The Y/N character variable TUTOR is an indicator of whether the student received mathematics tutoring of any kind during the school year.

Program 1: Reading in Data from a Hypothetical Mathematics Aptitude Survey

```
data test;
  input class class_type $ homeroom grade1 tutor $ grade2 weight;
datalines;
12 upper 1 87 Y 94 49.5
12 upper 1 89 N 89 49.5
12 upper 2 91 Y 94 48
12 upper 2 84 Y 92 48
11 upper 1 82 N 84 47.5
11 upper 1 94 N 95 47.5
11 upper 2 93 N 95 48
11 upper 2 94 Y 97 48
10 under 1 78 N 81 39
10 under 1 84 N 84 39
10 under 2 90 N 87 37.5
10 under 2 82 N 85 37.5
 9 under 1 88 N 88 40
 9 under 1 93 Y 91 40
 9 under 2 77 Y 85 48
 9 under 2 81 N 84 48
;
run;
```

3. BALANCED REPEATED REPLICATION

The first replication technique we will consider is *balanced repeated replication* (BRR) (McCarthy, 1966; Ch. 3 of Wolter, 2007), which was first conceived as a method strictly to estimate variances in two-PSU-per-stratum sample designs. In BRR, we select one of the two PSUs from each of *H* strata and double the weights of all units therein while setting the weights of all units in the unselected PSU to 0. This new weight is typically referred to as a *replicate weight*. The idea is to perform this process a number of times, appending a set of replicate weights to the analysis file for use in calculating a set of replicate-specific estimates. The variability amongst estimates calculated using each distinct replicate weight serves as the estimate of point estimate variability in the full sample. As will be shown shortly, there are built-in routines to do this when the VARMETHOD=BRR option is specified in the PROC statement of any SURVEY procedure in SAS/STAT®.

Before we continue, two questions immediately arise: (1) How many replicates (or replicate weights) are sufficient? (2) How do we know which particular PSUs to select? To achieve a desirable property called *full orthogonal balance*, the number of replicates necessary is the first multiple of four strictly greater than the number of strata. That is, if we denote R to be the number of replicates (or replicate weights), we require $H < R \leq H + 4$. With regard to the second question, we make use of Hadamard matrices from the experimental design field. A Hadamard matrix, typically denoted $\mathbf{H}$, is a square matrix of +1s and -1s where columns correspond to strata and rows replicates. (Columns and rows are technically interchangeable.) After randomly numbering the two PSUs (i.e., homerooms) within a stratum, let a +1 indicates selecting the first PSU and a -1 indicates selecting the second PSU. The "orthogonal" qualifier comes from the fact that the sum of the product of entries along any two rows or columns is 0. Below is one example Hadamard matrix that could be used for the hypothetical mathematics aptitude survey data set:

$$\mathbf{H} = \begin{bmatrix} +1 & +1 & +1 & +1 & +1 & +1 & +1 & +1 \\ +1 & -1 & +1 & +1 & -1 & -1 & -1 & +1 \\ +1 & +1 & -1 & -1 & -1 & -1 & +1 & +1 \\ +1 & -1 & -1 & -1 & +1 & +1 & -1 & +1 \\ +1 & +1 & -1 & -1 & +1 & -1 & -1 & -1 \\ +1 & -1 & -1 & +1 & -1 & +1 & +1 & -1 \\ +1 & +1 & +1 & -1 & -1 & +1 & -1 & -1 \\ +1 & -1 & +1 & -1 & +1 & -1 & +1 & -1 \end{bmatrix} \quad (1)$$

There are eight columns, but we only need four, and any will suffice, so we can take the first four. We can think of the leftmost column as representing the ninth grade stratum and the fourth column as representing the twelfth grade stratum. The first replicate will consist of the first homeroom from each stratum, while the second replicate will consist of the first homeroom for the ninth, eleventh, and twelfth grade strata, but the second homeroom from the tenth stratum, etc. SAS has Hadamard matrices stored internally to accommodate however many strata are detected in the input data set, but you can input your own with the VARMETHOD=BRR(HADAMARD=*data-set-name*) option in the PROC statement if desired. See the documentation for more details.

Once the replicates have been selected, the replicate-specific estimates are computed, which we can denote by $\hat{\theta}_r$ (r = 1, …, R). If we symbolize the full-sample estimate $\hat{\theta}$, the BRR variance estimate is the mean squared error of the replicate-specific estimates about $\hat{\theta}$, or

$$\text{var}_{BRR}(\hat{\theta}) = \frac{1}{R}\sum_{r=1}^{R}(\hat{\theta}_r - \hat{\theta})^2 \quad (2)$$

where the general theta notation is used here to emphasize that this is a universal formula applicable to *any* quantity.

Program 2 requests an estimated BRR variance for the mean of GRADE2 in the TEST data set. We inform SAS of the stratification identifier (CLASS) via the STRATA statement, the cluster identifier (HOMEROOM) via the CLUSTER statement, and the weights (WEIGHT) via the WEIGHT statement. The option VARMETHOD=BRR specified in the PROC statement requests balanced repeated replication.

Program 2: BRR Variance Estimation

```
proc surveymeans data=test varmethod=BRR mean var stderr;
  stratum class;
  cluster homeroom;
  var grade2;
weight weight;
run;
```

The SURVEYMEANS Procedure

Data Summary	
Number of Strata	4
Number of Clusters	8
Number of Observations	16
Sum of Weights	715

Variance Estimation	
Method	BRR
Number of Replicates	8

Statistics			
Variable	Mean	Std Error of Mean	Var of Mean
grade2	89.317483	1.201095	1.442630

While SAS is certainly capable of proceeding through all steps of the BRR process independently for each call to the given SURVEY procedure, it is more efficient to append the replicate weights onto the analysis file for subsequent BRR variance estimation requests. We can do this by specifying OUTWEIGHTS=*data-set-name* in parentheses immediately after VARMETHOD=BRR. Program 3 shows how to do so, storing the result in a data set called TEST_BRR. Figure 2 illustrates how TEST_BRR contains all input data set variables and observations, and also the replicate weights. There are 8 of them, but only the first 4 are shown. SAS affixes a common label of "Replicate Weight," but the variables are actually named REPWT_1, REPWT_2, …, REPWT_8.

Program 3: Creating an Analysis Data Set with BRR Replicate Weights Appended

```
proc surveymeans data=test varmethod=BRR (outweights=test_BRR)
  mean var stderr;
  stratum class;
  cluster homeroom;
  var grade2;
weight weight;
run;
```

	class	homeroom	grade2	weight	Replicate Weight	Replicate Weight	Replicate Weight	Replicate Weight
1	12	1	94	49.5	99	99	99	99
2	12	1	89	49.5	99	99	99	99
3	12	2	94	48	0	0	0	0
4	12	2	92	48	0	0	0	0
5	11	1	84	47.5	95	0	0	95
6	11	1	95	47.5	95	0	0	95
7	11	2	95	48	0	96	96	0
8	11	2	97	48	0	96	96	0
9	10	1	81	39	78	78	0	0
10	10	1	84	39	78	78	0	0
11	10	2	87	37.5	0	0	75	75
12	10	2	85	37.5	0	0	75	75
13	9	1	88	40	80	0	80	0
14	9	1	91	40	80	0	80	0
15	9	2	85	48	0	96	0	96
16	9	2	84	48	0	96	0	96

Figure 2: Partial View of the BRR Replicate Weights on Data Set TEST_BRR Created in Program 3

Examining the replicate weights, we can deduce that the first replicate was formed by selecting the first PSU (HOMEROOM=1) from all strata, since weights for those cases have doubled, whereas the weights for students in the other homeroom (HOMEROOM=2) were all set to zero. A similar line of reasoning applies to the other three replicate weights shown. If needed, you can have SAS output the particular Hadamard matrix used behind the scenes by specifying the PRINTH option in parentheses immediately after VARMETHOD=BRR.

Another appealing feature of the replicate weights is that they provide all of the complex design information necessary to properly calculate a variance. This means you can use them in later calls to any SURVEY procedure via a REPWEIGHTS statement without the STRATUM or CLUSTER statements. In fact, if those statements are specified in combination with the REPWEIGHTS statement, they are ignored.

Program 4 uses the data set with BRR replicate weights appended to produce the same results as Program 3 (output suppressed). Note that we still must specify the VARMETHOD=BRR in the PROC statement. When the REPWEIGHTS statement appears without the VARMETHOD= option, the replicate weights are assumed to be derived from the jackknife replication technique, which we will discuss in Section 5. Be advised that variance estimates will generally be incorrect if the technique specified in the VARMETHOD= option is not that which was used to construct the replicate weights.

Program 4: BRR Variance Estimation Using a Data Set with Replicate Weights Appended

```
proc surveymeans data=test_BRR varmethod=BRR mean var stderr;
  var grade2;
weight weight;
repweights RepWt_1-RepWt_8;
run;
```

4. FAY'S VARIANT TO BALANCED REPEATED REPLICATION

Judkins (1990) discusses a variant to the traditional BRR method whereby instead of doubling the weights of units in the selected PSU, weights are inflated by a factor of $2 - \varepsilon$, where $0 \leq \varepsilon < 1$, and weights of units in the "unselected" PSU are inflated by a factor of ε. Named after its inventor, Robert Fay, this method is typically called *Fay's BRR* and the epsilon term *Fay's coefficient*. Note that when $\varepsilon = 0$, the method defaults to traditional BRR; otherwise, there is a slight modification to formula 3 in that we divide through by a factor of $(1 - \varepsilon)^2$ as follows:

$$\mathrm{var}_{BRR-Fay}(\hat{\theta}) = \frac{1}{R(1-\varepsilon)^2}\sum_{r=1}^{R}(\hat{\theta}_r - \hat{\theta})^2 \qquad (3)$$

An advantage of Fay's BRR is that each PSU is represented in every replicate, which is believed to foster more stability in variance estimates. This is particularly important for surveys with fewer degrees of freedom or analyses of rare population domains.

Fay's BRR can be implemented by specifying VARMETHOD=BRR(FAY<=*Fay coefficient*>) in the PROC statement of any SURVEY procedure. As we illustrated in Program 4, we can have SAS do the legwork to create and append the Fay BRR replicate weights or we can provide the replicate weights directly. Declaring the Fay coefficient is technically optional; if nothing is specified, $\varepsilon = 0.5$ is assigned by default. There is no universally optimal ε, although Lee and Forthofer (2006) cite a few simulation results suggesting $\varepsilon = 0.3$ yields some desirable properties.

Program 5 requests Fay's BRR with $\varepsilon = 0.3$ to estimate the variance of the mean of GRADE2. The OUTWEIGHTS=*data-set-name* code in parentheses attaches the replicate weights to the original input data set and saves as a new data set named TEST_BRR_FAY. The standard error (1.1956) is slightly smaller than the traditional BRR standard error from Program 2 (1.2011). Figure 3 is a partial view of the first four replicate weights appended to the output data set TEST_BRR_FAY. Note how no weights are explicitly set to zero. For example, in the first replicate, weights of observations in the first PSU in both strata are multiplied by 1.7 while the "unselected" PSU's observations have their weights multiplied by 0.3.

Program 5: Fay's BRR Method for Variance Estimation

```
proc surveymeans data=test varmethod=BRR (outweights=test_BRR_Fay fay=.3)
  mean stderr;
  stratum class;
  cluster homeroom;
  var grade2;
weight weight;
run;
```

The SURVEYMEANS Procedure

Data Summary	
Number of Strata	4
Number of Clusters	8
Number of Observations	16
Sum of Weights	715

Variance Estimation	
Method	BRR
Number of Replicates	8
Fay Coefficient	0.3

Statistics		
Variable	Mean	Std Error of Mean
grade2	89.317483	1.195644

	class	homeroom	grade2	weight	Replicate Weight	Replicate Weight	Replicate Weight	Replicate Weight
1	12	1	94	49.5	84.15	84.15	84.15	84.15
2	12	1	89	49.5	84.15	84.15	84.15	84.15
3	12	2	94	48	14.4	14.4	14.4	14.4
4	12	2	92	48	14.4	14.4	14.4	14.4
5	11	1	84	47.5	80.75	14.25	14.25	80.75
6	11	1	95	47.5	80.75	14.25	14.25	80.75
7	11	2	95	48	14.4	81.6	81.6	14.4
8	11	2	97	48	14.4	81.6	81.6	14.4
9	10	1	81	39	66.3	66.3	11.7	11.7
10	10	1	84	39	66.3	66.3	11.7	11.7
11	10	2	87	37.5	11.25	11.25	63.75	63.75
12	10	2	85	37.5	11.25	11.25	63.75	63.75
13	9	1	88	40	68	12	68	12
14	9	1	91	40	68	12	68	12
15	9	2	85	48	14.4	81.6	14.4	81.6
16	9	2	84	48	14.4	81.6	14.4	81.6

Figure 3: Partial View of the BRR Replicate Weights on Data Set TEST_BRR Created in Program 3

5. THE JACKKNIFE

Another popular replication procedure is the *jackknife* (Ch. 4 of Wolter, 2007), the origins of which can be traced to Quenouille (1949), Tukey (1958), and Durbin (1959). There are actually several closely related forms of the jackknife used in practice, but the one we will focus on in this section is the traditional method, what Valliant et al. (2008) refer to as the "delete-one" version. Specifically, we delete each PSU, in turn, and weight up the remaining PSUs to form a replicate-specific estimate. As with BRR, the variance of any full-sample estimate is found using a universal function of variation amongst the replicate-specific estimates.

Since each PSU is deleted in one replicate, the number of replicate weights equals the number of PSUs. Although there tends to be more replicate weights to deal with, the jackknife is not restricted to two-PSU-per-stratum design as is BRR. In general, the replicate weights are constructed as follows:

1. For units in the dropped PSU, set all weights to 0.
2. For units in the same stratum as the dropped PSU, what SAS refers to as the donor stratum, inflate the weights by a factor of $n_h/(n_h - 1)$, where n_h is the number of PSUs in the donor stratum.
3. For units outside the donor stratum, the replicate weight takes on the same value as the original weight. (If there is no stratification in the sample design, you can skip this step.)

After computing the full-sample point estimate, $\hat{\theta}$, and the like using all replicate weights, $\hat{\theta}_r$ (r = 1, …, R), the jackknife variance estimate is

$$\text{var}_{JK}(\hat{\theta}) = \sum_{r=1}^{R} \frac{n_h - 1}{n_h} (\hat{\theta}_r - \hat{\theta})^2 \qquad (4)$$

Observe how the formula is no longer the mean squared error of the replicate-specific estimates, but instead a multiplicative factor of $(n_h - 1)/n_h$ is incorporated into each replicate estimate's squared deviation from the full-sample estimate. This stratum-specific factor is called the *jackknife coefficient*, and we will need to keep track of these when providing the SURVEY procedure jackknife replicate weights.

Program 6 requests a standard error for the mean of GRADE2 via the jackknife. Specifying VARMETHOD=JK is all that is needed to have SAS implement the three-step process sketched out above. The optional syntax in parentheses immediately thereafter tells SAS to append the jackknife replicate weights to the input data set and store the result in a data set named TEST_JK, and also to store the jackknife coefficients in a data set named TEST_JK_COEFS. As we will see in Program 7, we need to specify this supplemental data set as part of the REPWEIGHTS statement when VARMETHOD=JK. We find in the output the same point estimate is produced, but the standard error (1.1839) is slightly different from the techniques employed previously, but unbiased nonetheless.

Program 6: Jackknife Variance Estimation

```
proc surveymeans data=test varmethod=JK
 (outweights=test_JK outjkcoefs=test_JK_coefs) mean stderr;
  stratum class;
  cluster homeroom;
  var grade2;
weight weight;
run;
```

The SURVEYMEANS Procedure

Data Summary	
Number of Strata	4
Number of Clusters	8
Number of Observations	16
Sum of Weights	715

Variance Estimation	
Method	Jackknife
Number of Replicates	8

Statistics		
Variable	Mean	Std Error of Mean
grade2	89.317483	1.183876

Figures 4 and 5 offer a visualization of the output data sets housing the jackknife replicate weights and jackknife coefficients, respectively. There are 8 replicate weights generated, one for as many PSUs in the sample, although only the first four are shown in Figure 4. Underneath the label "Replicate Weight," they are named REPWT_1, REPWT_2, …, REPWT_8, the same nomenclature SAS uses when appending BRR replicate weights. SAS proceeds through the algorithm of dropping PSUs according to the sort order of their codes. This is why the PSU associated HOMEROOM=1 in CLASS=9 was dropped in the first replicate. We can also observe in the first replicate how units in the second PSU within the same donor stratum had their weights multiplied by $n_h/(n_h - 1) = 2$, whereas the weights of all units in other strata are unchanged. In Figure 5, we can observe that the data set TEST_JK_COEFS contains key variables REPLICATE and JKCOEFFICIENT that the SURVEY PROC will be looking for in subsequent analyses using these jackknife replicate weights.

	class	homeroom	grade2	weight	Replicate Weight	Replicate Weight	Replicate Weight	Replicate Weight
1	12	1	94	49.5	49.5	49.5	49.5	49.5
2	12	1	89	49.5	49.5	49.5	49.5	49.5
3	12	2	94	48	48	48	48	48
4	12	2	92	48	48	48	48	48
5	11	1	84	47.5	47.5	47.5	47.5	47.5
6	11	1	95	47.5	47.5	47.5	47.5	47.5
7	11	2	95	48	48	48	48	48
8	11	2	97	48	48	48	48	48
9	10	1	81	39	39	39	0	78
10	10	1	84	39	39	39	0	78
11	10	2	87	37.5	37.5	37.5	75	0
12	10	2	85	37.5	37.5	37.5	75	0
13	9	1	88	40	0	80	40	40
14	9	1	91	40	0	80	40	40
15	9	2	85	48	96	0	48	48
16	9	2	84	48	96	0	48	48

Figure 4: Partial View of the Jackknife Replicate Weights on Data Set TEST_JK Created in Program 6

	Replicate Number	Donor Stratum	Jackknife Coefficient
1	1	1	0.5
2	2	1	0.5
3	3	2	0.5
4	4	2	0.5
5	5	3	0.5
6	6	3	0.5
7	7	4	0.5
8	8	4	0.5

Figure 5: Data Set View of the TEST_JK_COEFS Data Set Containing the Jackknife Coefficients Created in Program 6

Program 7 repeats the analysis above by pointing PROC SURVEYMEANS to the data set containing the jackknife replicate weights. With the REPWEIGHTS statement and VARMETHOD=JK appearing in the PROC statement, we no longer need to include the STRATUM or CLUSTER statements. The JKCOEF=TEST_JK_COEFS option after the slash in the REPWEIGHTS statement points to the supplemental data set housing the jackknife coefficients, the data set shown in Figure 5. You can also specify a scalar if, as in the present case, all coefficients are the same. Hence, alternative syntax for this particular example would be JKCOEF=0.5.

Program 7: Jackknife Variance Estimation Using a Data Set with Replicate Weights Appended

```
proc surveymeans data=test_JK varmethod=JK mean stderr;
  var grade2;
weight weight;
repweights RepWt_1-RepWt_8 / jkcoefs=test_JK_coefs;
run;
```

The SURVEYMEANS Procedure

Data Summary	
Number of Observations	16
Sum of Weights	715

Variance Estimation	
Method	Jackknife
Replicate Weights	TEST_JK
Number of Replicates	8

Statistics		
Variable	Mean	Std Error of Mean
grade2	89.317483	1.183876

In the presence of two-PSU-per-stratum sample designs, a noteworthy simplification to the jackknife procedure can be employed (*cf*. Section 3.6.3.1 of Heeringa et al., 2010). Westat (2007, Appendix A) refers to this as the *JK2* approach. For linear estimators such as totals, it can be shown that retaining only one of the two replicates from each stratum and setting the jackknife coefficient to 1 is algebraically equivalent to the variance estimate obtained via the full jackknife procedure. For non-linear estimates such as weighted means, the variance is not always the same, yet still unbiased. Thus, we can cut our workload in half by randomly selecting one of the two replicates from all *H* strata and employing the following modified formula:

$$\text{var}_{JK2}(\hat{\theta}) = \sum_{\tilde{r}=1}^{R/2} (\hat{\theta}_{\tilde{r}} - \hat{\theta})^2 \qquad (5)$$

where $\tilde{r}$ indexes the replicate randomly chosen within each stratum.

Suppose we did this for our TEST_JK data set and selected the second, third, fifth, and eighth jackknife replicates. Although we want to set the jackknife coefficient to 1, SAS requires a value strictly between 0 and 1. (If we omit the JKCOEF= option altogether, the SURVEY procedure assumes it to be (*R* – 1)/*R*, where *R* is the number of replicate weights in the REPWEIGHTS statement.) The work-around is to specify a number that is inconsequentially less than 1, like 0.99999, as is done in Program 8. The corresponding output shows how the standard error (1.3195) is in the neighborhood of the standard error obtained from implementing the full jackknife procedure (1.1839).

Program 8: Illustrating the JK2 Variance Estimation Method, a Simplified Jackknife Procedure for Two-PSU-per-Stratum Designs

```
proc surveymeans data=test_JK varmethod=JK mean stderr;
  var grade2;
weight weight;
repweights RepWt_2 RepWt_3 RepWt_5 RepWt_8 / jkcoefs=0.99999;
run;
```

The SURVEYMEANS Procedure

Data Summary	
Number of Observations	16
Sum of Weights	715

Variance Estimation	
Method	Jackknife
Replicate Weights	TEST_JK
Number of Replicates	4

Statistics		
Variable	Mean	Std Error of Mean
grade2	89.317483	1.191791

6. THE BOOTSTRAP

The third replication technique we will consider is the *bootstrap* (Efron and Tibshirani, 1993) and its application to variance estimation of complex survey data (McCarthy and Snowden, 1985; Rao and Wu, 1988; Lahiri, 2003). Although the technique is not explicitly offered in the SURVEY procedures at the time of this writing (i.e., there is no VARMETHOD=BOOT), techniques shown in Lohr (2012) can be used as a work-around.

Perhaps the most popular form of the bootstrap in complex survey statistics is the *nonparametric bootstrap*. The first step is to independently select $n_h - 1$ PSUs with replacement from each stratum. If we let n^*_{hib} denote the number of times i^{th} PSU from the h^{th} stratum was selected in the b^{th} bootstrap sample, we can define the b^{th} set of replicate weights as $w_{hijb} = w_{hij}\left(\frac{n_h}{n_h - 1}\right)n^*_{hib}$, where j indexes units within the PSU. In words, this means we multiply the weights of all units in the i^{th} PSU from the h^{th} stratum by $n_h/(n_h - 1)$ times the number of times that PSU was selected. Using each replicate weight, we calculate $\hat{\theta}_b$, an estimate of the quantity of interest. The idea is to repeat the process independently B times, where B is often 200 or more, and then use the following formula to estimate the full-sample variance:

$$\text{var}_{boot}(\hat{\theta}) = \frac{1}{B-1}\sum_{b=1}^{B}(\hat{\theta}_b - \hat{\theta})^2 \qquad (6)$$

where, as before, $\hat{\theta}$ denotes the estimate calculated from the full-sample.

To use this technique with our example complex survey data set TEST, we must first create the bootstrap replicate weights. For this task, we can use the %BOOTWT macro introduced by Lohr (2012). The first macro parameter names the input data set and the second identifies the full-sample weight variable. The third and fourth identify the stratum and cluster identifiers, respectively. The fifth parameter specifies the number of bootstrap replicates requested. The sixth parameter names the output data set to consist of the input data set with replicate weights appended, and the seventh parameter is a prefix to be used in naming the replicate weights. The eighth and final parameter is a seed the user can specify to ensure the replicate weight creation process is reproducible—an addition to the original macro definition. The macro

call after the compilation step results in 200 bootstrap replicate weights named REPWT_1, REPWT_2, …, REPWT_200 being tacked onto the output data set called TEST_BOOT.

Program 9: Creating Bootstrap Replicate Weights Using the %BOOTWT Macro

```
%macro bootwt (fulldata,wt,stratvar,psuvar,numboot,fullrep,repwt,seed);
proc sort data=&fulldata out=fulldata;
  by &stratvar &psuvar;
run;

proc sql stimer;
  create table psulist as
  select distinct &stratvar, &psuvar
    from fulldata
   order by &stratvar, &psuvar;

  create table numpsu as
  select distinct &stratvar, count(*)-1 as _nsize_
    from psulist
   group by &stratvar
   order by &stratvar;
quit;

data fulldata (drop=_nsize_);
  merge fulldata (in=inf)
        numpsu   (in=inn);
  by &stratvar;
if inf & inn;
wtmult = (_nsize_ + 1) / _nsize_;
run;

proc surveyselect data=psulist method=urs sampsize=numpsu
                  out=repout outall reps=&numboot seed=&seed;
  strata &stratvar;
  id &psuvar;
run;

proc sort data=repout (keep=&stratvar &psuvar replicate numberhits)
          out=repout_sorted;
  by &stratvar &psuvar replicate;
run;

proc transpose data=repout_sorted (keep=&stratvar &psuvar replicate
                                        numberhits)
     out=repout_tr (keep=&stratvar &psuvar repmult:)
     prefix=repmult;
  by &stratvar &psuvar;
  id replicate;
  var numberhits;
run;

data &fullrep (drop=i repmult1-repmult&numboot wtmult);
array &repwt  (&numboot);
array repmult (&numboot);
  merge fulldata  (in=inf)
        repout_tr (in=inr);
  by &stratvar &psuvar;
  do i = 1 to &numboot;
```

```
    &repwt(i) = &wt * repmult(i) * wtmult;
  end;
run;

%mend bootwt;

%bootwt (test,weight,class,homeroom,200,test_boot,RepWt_,399448);
```

	RepWt_1	RepWt_2	RepWt_3	RepWt_4	RepWt_5
1	0	80	0	80	0
2	0	80	0	80	0
3	96	0	96	0	96
4	96	0	96	0	96
5	78	78	78	0	0
6	78	78	78	0	0
7	0	0	0	75	75
8	0	0	0	75	75
9	0	0	95	95	95
10	0	0	95	95	95
11	96	96	0	0	0
12	96	96	0	0	0
13	99	99	99	99	0
14	99	99	99	99	0
15	0	0	0	0	96
16	0	0	0	0	96

Figure 6: Data Set View of the TEST_JK_COEFS Data Set Containing the Jackknife Coefficients Created in Program 6

The replicate weights shown in Figure 6 might strike you as closely resembling the BRR replicate weights shown in Figure 1. Indeed, BRR is sometimes characterized as a "smart" bootstrap requiring far fewer replications in the two-PSU-per-stratum setting. An advantage of the bootstrap technique, however, just like the jackknife, is that it is directly amenable to variable PSU counts per stratum.

Once the replicate weights have been appended to the file, we can utilize the REPWEIGHTS statement to do the heavy lifting in calculating all $B = 200$ bootstrap replicate estimates. Looking at equation 6, Lohr (2012) notes how you can move the term $1/(B - 1)$ inside the summation and arrive at essentially the same structure given by equation 4. This implies we could specify VARMETHOD=JK in the PROC statement and insert $1/(B - 1)$ as the JKCOEF= option. Since $1/(200 - 1) = 0.005025$ in the present case, Program 10 shows how this can be done for an estimate of the mean of GRADE2. From the output, we observe yet another valid standard error estimate (1.1561).

Program 10: Creating Bootstrap Replicate Weights Using the %BOOTWT Macro

```
proc surveymeans data=test_boot varmethod=JK mean stderr;
  var grade2;
weight weight;
repweights RepWt_1-RepWt_200 / jkcoef=0.005025;
run;
```

The SURVEYMEANS Procedure

Data Summary	
Number of Observations	16
Sum of Weights	715

Variance Estimation	
Method	Jackknife
Replicate Weights	TEST_BOOT
Number of Replicates	200

Statistics		
Variable	Mean	Std Error of Mean
grade2	89.317483	1.156067

7. REPLICATION WITH LINEAR MODELS

While the exposition above focused on a single descriptive statistic, the sample mean, replication techniques can also be used to estimate variances for multivariate statistics, such as a vector of estimated linear model parameters $\hat{\mathbf{B}}$. For example, the multivariate generalization to BRR is

$$\text{cov}_{BRR}(\hat{\mathbf{B}}) = \frac{1}{R}\sum_{r=1}^{R}(\hat{\mathbf{B}}_r - \hat{\mathbf{B}})^{\mathrm{T}}(\hat{\mathbf{B}}_r - \hat{\mathbf{B}}) \qquad (7)$$

where $\hat{\mathbf{B}}_r$ denotes the r^{th} replicate-specific estimate of the given model parameters and $\hat{\mathbf{B}}$ symbolizes the parameters estimated from the full sample.

Suppose we wanted to fit a simple linear regression model predicting GRADE2 based on a 0/1 indicator variable of whether the student regularly receives math tutoring. We can use the same VARMETHOD=BRR and REPWEIGHTS syntax on the data set TEST_BRR to estimate $\text{cov}_{BRR}(\hat{\mathbf{B}})$. In Program 11, we first create the indicator variable TUTOR_Y in a DATA step, and then run PROC SURVEYREG to fit the model of GRADE based on an intercept and TUTOR_Y. The COVB option is specified after the slash in the MODEL statement to output $\text{cov}_{BRR}(\hat{\mathbf{B}})$.

Program 11: BRR Variance Estimation for a Linear Model

```
data test_brr;
  set test_brr;
tutor_Y=(tutor='Y');
run;

proc surveyreg data=test_brr varmethod=BRR;
  model grade2 = tutor_Y / covb;
weight weight;
repweights RepWt_1-RepWt_8;
run;
```

Covariance of Estimated Regression Coefficients		
	Intercept	tutor_Y
Intercept	0.9928488886	-0.201326971
tutor_Y	-0.201326971	1.4889379508

To get a better handle on the underlying calculations prescribed by equation 7, let us briefly walk through how PROC SURVEYREG arrives at these numbers. Essentially, the weighted least squares parameters are estimated using the variable WEIGHT then for REPWT_1, REPWT_2, …, REPWT_8. The estimates of the intercept and slope using the variable WEIGHT are 87.4394 and 4.7701, respectively, and Table 1 summarizes the eight sets of estimates found by using each of the BRR replicate weights.

Replicate	$\hat{B}_0$	$\hat{B}_1$
1	87.0971	5.5621
2	86.9463	5.0743
3	89.1082	5.0663
4	87.4393	2.1300
5	86.6549	5.7568
6	86.3621	5.6379
7	89.1411	4.4676
8	87.0849	3.2485

Table 1: Summary of BRR Replicate-Specific Parameter Estimates for Simple Linear Regression Model Fitted in Program 11

The figures in Table 1 are coalesced to produce the following matrix generated by Program 11:

$$\mathrm{cov}_{BRR}(\hat{\mathbf{B}}) = \begin{bmatrix} \mathrm{var}_{BRR}(\hat{B}_0) & \mathrm{cov}_{BRR}(\hat{B}_0, \hat{B}_1) \\ \mathrm{cov}_{BRR}(\hat{B}_0, \hat{B}_1) & \mathrm{var}_{BRR}(\hat{B}_1) \end{bmatrix} \quad (8)$$

where, for example, $\mathrm{var}_{BRR}(\hat{B}_0) = \frac{1}{8}\left[(87.0971 - 87.4394)^2 + \ldots + (87.0849 - 87.4394)^2\right]$ and

$$\mathrm{cov}_{BRR}(\hat{B}_0, \hat{B}_1) = \frac{1}{8}\left[(87.0971 - 87.4394)(5.5621 - 4.7701) + \ldots + (87.0849 - 87.4394)(3.2485 - 4.7701)\right].$$

Other replication techniques can be employed in an analogous manner. Be advised, however, that there is not yet a SURVEY procedure to accommodate all possible linear models. One example is Poisson regression models, which are applicable when the outcome variable is a count of some kind. One plausible work-around would be to use a SURVEY procedure or the %BOOTWT macro to append replicate weights to the analysis file, and then tweak code in one of the user-written macros developed to perform replication for linear models—for instance, Bienias (2001) or Berglund (2004). Another potential work-around is to use the %SASMOD module of IVEware, a free set of SAS-callable macros developed by researchers at the University of Michigan (Raghunathan et al., 2002). We will not walk through an example here.

8. REPLICATION WITH LINEAR MODELS

Examples thus far have dealt largely with utilizing replication to approximate one or more measures of sampling variability. If you wish to have a SURVEY procedure deploy these measures as part of a significance test or to construct confidence intervals, it is important to remain cognizant of SAS' rules for assigning degrees of freedom to the underlying reference distributions. When you specify a replication technique in the VARMETHOD= option in the PROC statement without the REPWEIGHTS statement—but with a STRATA and/or CLUSTER statement, if applicable—SAS calculates the degrees of freedom # PSUs – # strata, which coincides with the recommendations in Figure 4-20 of Westat (2007). If a

REPWEIGHTS statement is present, however, SAS assigns the degrees of freedom to be the number of replicate weight variables. If this is not appropriate, you can reassign the value using the DF=*number* option after the slash in the REPWEIGHTS statement.

Let us revisit using the bootstrap replicate weights on the TEST_BOOT data set for inferences on the mean of the variable GRADE. Program 12 builds on syntax shown previously in Program 10 by adding the CLM and DF options in the PROC statement to output a 95% confidence interval and the degrees of freedom used for the underlying t distribution. The first SURVEYMEANS run uses the default degrees of freedom, 200, which is vastly overstated. Following Lohr (2012, p. 12), the second run overrides the default with a more appropriate number, DF=4, the # PSUs – # strata in the underlying sample design. Note from the output how the confidence interval is narrower in the first run as compared to the second. Hence, failing to properly adjust for the degrees of freedom would have led us to overstate the precision.

Program 12: Adjusting the Degrees of Freedom for the Sample Mean of a Sparse Domain When Using Replication for Variance Estimation

```
title '1) Bootstrap CI Using Default DF Calculation';
proc surveymeans data=test_boot varmethod=JK mean stderr clm df;
  var grade2;
weight weight;
repweights RepWt_1-RepWt_200 / jkcoef=0.005025;
run;

title '2) Bootstrap CI Overriding Default DF Calculation';
proc surveymeans data=test_boot varmethod=JK mean stderr clm df;
  var grade2;
weight weight;
repweights RepWt_1-RepWt_200 / jkcoef=0.005025 DF=4;
run;
```

1) Bootstrap CI Using Default DF Calculation

The SURVEYMEANS Procedure

Data Summary	
Number of Observations	16
Sum of Weights	715

Variance Estimation	
Method	Jackknife
Replicate Weights	TEST_BOOT
Number of Replicates	200

Statistics					
Variable	DF	Mean	Std Error of Mean	95% CL for Mean	
grade2	200	89.317483	1.156067	87.0378387	91.5971263

2) Bootstrap CI Overriding Default DF Calculation

The SURVEYMEANS Procedure

Data Summary	
Number of Observations	16
Sum of Weights	715

Variance Estimation	
Method	Jackknife
Replicate Weights	TEST_BOOT
Number of Replicates	200

Statistics					
Variable	DF	Mean	Std Error of Mean	95% CL for Mean	
grade2	4	89.317483	1.156067	86.1077265	92.5272386

Similar logic applies if we were to flesh out Programs 4 and 7; in both of those instances, 8 degrees of freedom are assumed, which is still too many.

9. SUMMARY

Replication techniques are flexible alternatives that complex survey data analysts can utilize to estimate variances in lieu of Taylor series linearization, the default method used by SAS and most other software. Instead of requiring a unique formula for each point estimate, replication techniques generally employ a universal formula regardless of the quantity. We did not consider all replication techniques used in practice. For a more comprehensive treatment of the subject, see Wolter (2007). Replication techniques or subtle variants thereof are sometimes proposed as a way to account for the uncertainty of missing data adjustments. For instance, many researchers (e.g., Valliant, 2004) argue that applying the nonresponse adjustment procedure independently on each replicate weight is needed to capture the uncertainty attributable to techniques compensating for unit nonresponse. As another example, Efron (1994) proposes a bootstrap approach to account for the uncertainty when imputing missing data for item nonresponse.

There may be circumstances where a particular replication technique as prescribed in this chapter is difficult or impossible to implement. For example, we cannot use BRR or Fay's BRR directly unless there are exactly 2 PSUs in all strata. Moreover, for designs with a large number of PSUs, the number of replicate weights required can become cumbersome, even with modern computing power. The typical work-around is to group PSUs into pseudo-PSUs, collapse strata into pseudo-strata, or perhaps a combination of both. Some general references are Rust (1986), Kott (2001), Appendix D of Westat (2007), and the references cited in the appendix of Mukhopadhyay et al. (2008). These grouping procedures do not induce bias into the variance estimates, but they sacrifice some degrees of freedom.

REFERENCES

Berglund, P. (2004). "Analysis of Complex Sample Survey Data Using the SURVEY PROCEDURES and Macro Coding," *Proceedings of the Annual Conference of the MidWest SAS Users Group*. Available on-line at: http://www.lexjansen.com/mwsug/2004/Statistics/S7_Berglund.pdf

Bienias, J. (2001). “Replicate-Based Variance Estimation in a SAS® Macro,” *Proceedings of the Annual Conference of the NorthEast SAS Users Group (NESUG)*. Available on-line at: http://nesug.org/proceedings/nesug01/st/st9001.pdf

Durbin, J. (1959). “A Note on the Application of Quenouille's Method of Bias Reduction to the Estimation of Ratios,” *Biometrika*, **46**, pp. 477 – 480.

Efron, B., and Tibshirani, R. (1993). *An Introduction to the Bootstrap*. New York, NY: Chapman and Hall.

Efron, B. (1994). “Missing Data, Imputation, and the Bootstrap,” *Journal of the American Statistical Association*, **89**, pp. 463 – 479.

Hawkes, R. (1997). “Implementing Balanced Replicated Subsampling Designs in SAS® Software,” *Proceedings of the SAS Users Group International (SUGI) Conference*. Available on-line at: http://www2.sas.com/proceedings/sugi22/STATS/PAPER279.PDF

Heeringa, S., West, B., and Berglund, P. (2010). *Applied Survey Data Analysis*. Boca Raton, FL: Chapman and Hall/CRC Press.

Judkins, D. (1990). “Fay’s Method for Variance Estimation,” *Journal of Official Statistics*, **6**, pp. 223 – 240.

Lahiri, P. (2003). “On the Impact of Bootstrap in Survey Sampling and Small-Area Estimation,” *Statistical Science*, **18**, pp. 199 – 210.

Lee, E., and Forthofer, R. (2006). *Analyzing Complex Survey Data. Second Edition.* Thousand Oaks, CA: Sage.

Lewis, T. (2015). “An Introductory Overview of the Features of Complex Survey Data,” *Proceedings of the SAS Global Forum*. Dallas, TX.

Lohr, S. (2012). “Using SAS® for the Design, Analysis, and Visualization of Complex Surveys,” *Proceedings of the SAS Global Forum*. Available on-line at: http://support.sas.com/resources/papers/proceedings12/343-2012.pdf

McCarthy, P. (1966). “Replication: An Approach to the Analysis of Data from Complex Surveys,” *Vital and Health Statistics*, *Series 2*, **14**. Washington, DC: U.S. Government Printing Office.

McCarthy, P., and Snowden, C. (1985). “The Bootstrap and Finite Population Sampling,” *Vital and Health Statistics*, *Series 2*, **95**. Washington, DC: U.S. Government Printing Office.

Mukhopadhyay, P., An, A., Tobias, R., and Watts, D. (2008). “Try, Try Again: Replication-Based Variance Estimation Methods for Survey Data Analysis in SAS® 9.2,” *Proceedings of the SAS Global Forum*. Available on-line at: http://www2.sas.com/proceedings/forum2008/367-2008.pdf

Quenouille, M., (1949). “Approximate Tests of Correlation in Time Series,” *Journal of the Royal Statistical Society*, *Series B*, **11**, pp. 68 – 84.

Raghunathan, T., Solenberger, P., and Van Hoewyk. J. (2002). *IVEware: Imputation and Variance Estimation Software: User Guide.* Ann Arbor, MI: Institute for Social Research, University of Michigan. Available on-line at: ftp://ftp.isr.umich.edu/pub/src/smp/ive/ive_user.pdf.

Rao, J.N.K., and Wu, C. (1988). “Resampling Inference with Complex Survey Data,” *Journal of the American Statistical Association*, **83**, pp. 231 – 241.

Rust, K. (1985). “Variance Estimation for Complex Estimators in Sample Surveys,” *Journal of Official Statistics*, **1**, pp. 381 – 397.

Rust, K. (1986). “Efficient Formation of Replicates for Replicated Variance Estimation,” *Proceedings of the Survey Research Methods Section of the American Statistical Association*. Alexandria, VA: American Statistical Association.

Rust, K., and Rao, J.N.K. (1996). “Replication Methods for Analyzing Complex Survey Data,” *Statistical Methods in Medical Research: Special Issue on the Analysis of Complex Surveys*, **5**, pp. 283 – 310.

Tukey, J. (1958). “Bias and Confidence in Not-Quite Large Samples,” (Abstract) *Annals of Mathematical Statistics*, **29**, p. 614.

Valliant, R. (2004). “The Effect of Multiple Weight Adjustments on Variance Estimation,” *Journal of Official Statistics*, **20**, pp. 1 – 18.

Valliant, R., Brick, J.M., and Dever, J. (2008). “Weight Adjustments for the Grouped Jackknife Variance Estimator,” *Journal of Official Statistics*, **24**, pp. 469 – 488.

Westat. (2007). *WesVar® 4.3 User’s Guide*. Available on-line at: http://www.westat.com/Westat/pdf/wesvar/WV_4-3_Manual.pdf

Wolter, K. (2007). *Introduction to Variance Estimation. Second Edition*. New York, NY: Springer.

RECOMMENDED READING

Complex Survey Data Analysis in SAS® (forthcoming) by Taylor Lewis

CONTACT INFORMATION

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

Taylor Lewis
PhD Graduate
Joint Program in Survey Methodology
University of Maryland, College Park
tlewis9@umd.edu