

Resampling variance estimation for complex survey data

Stanislav Kolenikov
University of Missouri
Columbia, MO/USA
kolenikovs@missouri.edu

Abstract. We discuss the main approaches to resampling variance estimation in complex survey data: balanced repeated replication, the jackknife and the bootstrap. Balanced repeated replication and the jackknife are implemented in Stata `svy` suite. The bootstrap for complex survey data is implemented by package `bsweights`. This package is described, and working examples are provided.

Keywords: `st0001`, balanced repeated replication, balanced bootstrap, bootstrap, BRR, `bsweights`, complex survey data, Hadamard matrix, half-samples, jackknife, resampling, weighted bootstrap

1 Complex survey data

Researchers who study large-scale health, behavioral and economic processes often have to deal with data sets collected via complex survey designs. To achieve balance between costs and statistical accuracy, data collection in large-scale surveys is organized using specialized sampling techniques: stratification, clustering, multiple stages of selection, unequal probabilities of selection, and sampling with or without replacement, to name a few. The estimation procedures must be adapted to these complex survey design features.

In *stratified samples*, the population is split into non-overlapping parts, called strata, before any of the sampling steps are taken. Stratification criteria might be geography and urbanicity in areal samples, land use in natural resource surveys, or industry and size of the firm in establishment surveys. Survey sample designers use stratification to improve efficiency, protect against badly unbalanced samples and optimize the total cost of the survey. Stratification also allows to conduct straightforward statistical analysis within strata and implement different sampling techniques in different strata.

Cluster samples allow to reduce costs in situations where it is impossible or impractical to obtain the complete list of ultimate observation units. Instead, the sample designer obtains a much shorter list of clusters, or *primary sampling units* (PSUs). Once the required number of PSUs is sampled, more detailed lists are collected for these PSUs. The procedure of taking clusters of units may be repeated at several levels, resulting in *multiple stages of selection*. Large scale human population surveys usually feature between two and four stages of selection.

Sampling weights are necessary to ensure unbiased estimation. In their basic form,

sampling weights are inverse probabilities of selection. Additional modifications of sampling weights are often performed. *Poststratification* uses the existing census or population information on variables such as demographics (the number of people in the age by gender by race groups). The weights are modified so that the weighted totals of poststratification variables match the population totals. To compute *non-response adjustments*, the sample is broken into non-response adjustment cells within which the non-response can be assumed completely random. The weights of the responding units are then increased to match the total weight of all sampled units.

When sampling is done without replacement, there are efficiency gains summarized by *finite population corrections* (fpc). For simple random samples, the fpc is $1 - f$, where $f = n/N$ is the sampling fraction, n is the sample size, and N is the population size. Finite population corrections are rarely available in large scale public use data sets either because the population sizes are not known, or because of privacy and confidentiality considerations.

Additional features of complex surveys include longitudinal and rotating designs, in which the same units provide information at several time points. In longitudinal designs, the sample is taken once before the first round of data collection. In rotating designs, new units are sampled each round, while older units are retired from the sample to reduce the response burden.

What happens if the complex survey design features are ignored in statistical analysis? If stratification and/or finite population corrections are ignored, the standard errors will be conservative (too large), the confidence intervals will be too long, and their coverage will exceed the nominal 95% level. While the positive bias of the standard errors leads to a loss of power, it can generally be considered acceptable. If clustering is ignored, the standard errors will be too small, and reported results will be claimed significant too often. If sampling weights are ignored, then the sampling distributions of unweighted statistics will be shifted towards more probable values, leading to biased estimates. Both effects are certainly undesirable, if not unacceptable.

For a very complex survey design, exact accounting for all its features is extremely cumbersome. At data analysis stage, approximations are often made to yield usable estimation formulae. The most common approximate design is stratified two-stage sample with replacement (S2SWR). In this design, the population is divided into strata. From each stratum, a sample of PSUs is taken with replacement, and from each PSU, samples of ultimate units are taken. If the same PSU were sampled more than once, independent samples within this PSU are taken and marked in the data set as distinct. Unequal probabilities of selection may be used. The S2SWR approximation allows for relatively simple computations of point estimates (using weights only) and variances (using weights, stratification, and PSU information only).

An example of the S2SWR design is NHANES II data from Stata [SVY] manual:

(Continued on next page)

```

. webuse nhanes2
. svyset

      pweight: finalwgt
      VCE: linearized
Single unit: missing
Strata 1: stratas
SU 1: psu
FPC 1: <zero>

```

Here, the design is specified as two stages. In the first stage, the stratification variable is `strata`, and the PSU/cluster level variable is `PSU`. The second stage units are assumed to be individual observations. The sampling weight variable is `finalwgt`. There are no finite population corrections available for this data set. By default, the variances of the parameter estimates will be computed using linearization method (`VCE: linearized`). The statement `Single unit: missing` specifies that the variances will be reported as missing when only one PSU is available in some strata. In this data set, PSUs are numbered 1 and 2 in each stratum, and in certain data manipulations, it is crucial to keep track of both strata and PSU identifiers.

Let us run a benchmark analysis of high blood pressure with individual covariates. This is Example 2 of [SVY] **svy estimation**.

```

. svy: logistic highbp height weight age female
(running logistic on estimation sample)

Survey: Logistic regression
Number of strata =      31          Number of obs   =    10351
Number of PSUs  =      62          Population size = 117157513
                                          Design df      =      31
                                          F( 4, 28)      =    178.69
                                          Prob > F       =    0.0000

```

highbp	Linearized		t	P> t	[95% Conf. Interval]	
	Odds Ratio	Std. Err.				
height	.9688567	.0056822	-5.39	0.000	.9573369	.9805151
weight	1.052489	.0032829	16.40	0.000	1.045814	1.059205
age	1.050473	.0024816	20.84	0.000	1.045424	1.055547
female	.7250086	.0641188	-3.64	0.001	.6053529	.8683157

Compared to typical output of non-svy commands, basic design information is added. In the upper left block, the number of PSUs and the number of strata are shown. The difference of the two is the design degrees of freedom, which is the greatest number of explanatory variables that can be used in a regression model. The design degrees of freedom are reported in the upper right block. The population size is estimated by the sum of weights. The column of the standard errors has a heading indicating the type of the standard errors used (linearized standard errors). Other components of the output are the same as in non-survey estimation.

To conclude this section, let us mention a few references on survey statistics. A popular introductory textbook is Lohr (2009). More formal treatment is given in classic

monographs of Kish (1995) and Cochran (1977). The utmost level of mathematical details can be found in Särndal et al. (1992) and Thompson (1997). Great intermediate books with extensive conceptual explanations are Korn and Graubard (1995) and Lehtonen and Pahkinen (2004). Advanced topics are discussed in collected volumes Skinner et al. (1989) and Chambers and Skinner (2003). For reviews of resampling methods, see Rust and Rao (1996) and Shao (1996, 2003).

2 Variance estimation in complex surveys

Variance estimation in complex surveys serves two goals. First, applied researchers need standard errors to test their hypotheses of substantive interest and construct (Wald) tests and confidence intervals. Second, sample designers use variance estimates to gauge performance of existing designs and choose design parameters for future surveys of similar populations.

There are several variance estimation methods commonly used with complex survey data. We shall talk about direct variance estimation and linearization in this section, turning to resampling methods in the next section.

Consider the following estimate of the total for stratified samples:

$$t_{str}[x] = \sum_{h=1}^L W_h \sum_i t_{hi}, \quad (1)$$

where

$$t_{hi} = \sum_{j \in \text{PSU}_{hi}} w_{hij} x_{hij}$$

is the estimate of the total based on PSU h, i , and other symbols are defined in Appendix. In the S2SWR design, the variance of $t_{str}[x]$ can be directly estimated by

$$v_{str}[t_{str}[x]] = \sum_{h=1}^L \frac{n_h}{n_h - 1} \sum_{i=1}^{n_h} (t_{hi} - \bar{t}_h)^2, \quad (2)$$

where

$$\bar{t}_h = \frac{1}{n_h} \sum_{i=1}^{n_h} t_{hi}.$$

This is analogous to equation (1) from [SVY] **variance estimation**, except that finite population corrections are ignored since sampling is assumed to be with replacement in the S2SWR design.

When the statistic of interest is a function of moments (e.g., means, ratios, correlation or regression coefficients), the customary method of variance estimation is Taylor series expansion, or linearization, also known in theoretical statistics and econometrics as the delta method. Let $\theta = f(T_1, \dots, T_K)$ be a smooth function of the totals

T_1, \dots, T_K and $\hat{\theta} = f(t_1, \dots, t_K)$ be an estimate of θ , where t_1, \dots, t_K are sample-based estimates of the corresponding totals. Then the linearization variance estimator is

$$v_L[\hat{\theta}] \approx \widehat{\text{MSE}}[\hat{\theta}] \approx v \left[\sum_k \frac{\partial f}{\partial t_k} t_k \right]. \quad (3)$$

An important case is the sample mean

$$\bar{x}_{str} = \sum_h W_h \frac{\sum_{ij} w_{hij} x_{hij}}{\sum_{ij} w_{hij}}.$$

If sample size is not fixed by design, the mean is a ratio estimator, and its variance is estimated by

$$v_{str}[\bar{x}_{str}] = \sum_h \frac{n_h}{n_h - 1} \sum_i (d_{hi} - \bar{d}_h)^2, \quad (4)$$

where

$$d_{hi} = \sum_j w_{hij} (x_{hi} - \bar{x}_{str}), \quad \bar{d}_h = \frac{1}{n_h} \sum_i d_{hi}.$$

Linearization method is also applicable when $\hat{\theta}$ is derived as a solution to a set of estimating equations

$$t[\psi(x, \theta)] = \sum_{h,i,j} w_{hij} \psi(x_{hij}, \theta) = 0. \quad (5)$$

The most common examples are generalized linear models (Binder 1983) and other models based on quasi-likelihoods (Skinner 1989), with $\psi(x, \theta)$ being the score equations. The linearization variance estimator has a sandwich structure:

$$v_L[\hat{\theta}] \approx (\nabla_{\theta} t[\psi(x, \theta)])^{-1} v[t[\psi(x, \theta)]] (\nabla_{\theta} t[\psi(x, \theta)])^{-1T}. \quad (6)$$

This expression should be contrasted with the traditional variance estimator in fully parametric likelihood inference: the inverse of the Hessian matrix, $-(\nabla_{\theta} t[\psi(x, \theta)])^{-1}$. In complex survey data analysis, the latter is not a consistent estimator of the variance of $\hat{\theta}$.

Variance estimation based on linearization is the default estimation method for survey data in Stata. The required derivatives are computed analytically for a number of commonly used models, and numerically for other models using `_robust`.

For parameters and statistics that are not smooth functions of the underlying distributions, such as quantiles, extreme order statistics or certain poverty indicators, the linearization variance estimator cannot be computed.

When the survey data are released for public use, confidentiality of the respondents must be protected. Geographic information is provided in a coarse form, incomes are top-coded, small racial groups are conglomerated, etc. Variance estimation with equations (2) or (4) require that stratum and PSU identifiers h and i are known for each

observation. In (6), stratum and PSU identifiers are needed to compute $v[t(\psi)]$, which is done analogously to (2) or (4). If the data provider decides that releasing strata and PSU information poses the threat that individual subjects could be identified, alternative variance estimation methods must be used.

3 Resampling methods

The three major resampling, or replication, methods used in complex survey inference are balanced repeated replication, the jackknife and the bootstrap. In each of these methods, multiple replicates of the original data are created. In the r -th replicate, some PSUs are omitted, and some are included (may be multiple times, as in the bootstrap). The parameter estimate of interest $\hat{\theta}_m^{(r)}$ is computed using the same estimation procedure as for the original data. Subindex m stands for the estimation method. The resulting estimator of variance is generally defined by

$$v_m[\hat{\theta}] = \frac{A}{R} \sum_{r=1}^R (\hat{\theta}_m^{(r)} - \tilde{\theta})^2. \quad (7)$$

Here, R is the number of replicates and A is a scaling constant chosen to ensure that in the linear case v_m coincides with the known estimator (2). The measure of the central tendency $\tilde{\theta}$ can be the mean of resampled values

$$\tilde{\theta} = \frac{1}{R} \sum_{r=1}^R \hat{\theta}_m^{(r)}, \quad (8)$$

resulting in the variance version of the estimator, or the original estimate

$$\tilde{\theta} = \hat{\theta}, \quad (9)$$

resulting in the MSE version of the estimator.

In most cases, $v_m[\hat{\theta}]/v_L[\hat{\theta}] \rightarrow 1$ in probability as $n = \sum_h n_h \rightarrow \infty$. In other words, in large samples all replication estimators are close to one another and to the linearization estimator. Shao (1996) suggests that the choice of the estimator should be based on computational rather than statistical grounds.

While formal interpretation of resampling methods is that the sample is re-created for each replicate r , a more practical implementation is achieved by varying the sampling weights. For instance, if a sampling unit is removed in a given replicate, it can simply be given a weight of zero. The weights of other units in the same stratum need to be increased to ensure that the totals are unbiased for each replicate. A set of replicate weights $w_{hij}^{(r)}$, $r = 1, \dots, R$, is created and distributed with the complex survey data set. Variance estimation proceeds by running the same command $1 + R$ times (where the first run is to obtain the point estimates based on the original weights), substituting the replicate weights in place of the original ones, computing the estimates of interest, and combining the results using (7).

For the methods below, we describe the mechanics, the implied replicate weights and the number of replicates each method requires. Comparison of the properties of variance estimation procedures is given in Section 3.5.

3.1 Balanced repeated replication

Balanced repeated replication (BRR) was introduced by McCarthy (1969) for the class of designs in which $n_h = 2$ for all strata. In each replicate, one of the two PSUs is omitted, the other one is retained and replicated twice to ensure the totals are on the right scale. Since exactly half of the PSUs are used, the replicates are also referred to as half-samples. The replicate weights are

$$w_{hij}^{(r)} = \begin{cases} 2w_{hij}, & \text{PSU } h, i \text{ is retained} \\ 0, & \text{PSU } h, i \text{ is omitted.} \end{cases} \quad (10)$$

These weights are used to compute $\hat{\theta}_{BRR}^{(r)}$. The BRR variance estimator is obtained from (7) with $A = 1$:

$$v_{BRR}[\hat{\theta}] = \frac{1}{R} \sum_{r=1}^R (\hat{\theta}_{BRR}^{(r)} - \hat{\theta})^2. \quad (11)$$

The number of all possible half-samples is 2^L . If all half-samples are used, $v_{BRR} = v_L = v_{str}$ in the linear case. McCarthy (1969) showed that this equality holds with much smaller number of replicates, $L \leq R \leq L + 3$, for resampling designs that satisfy certain balance conditions. To wit, R must be a multiple of 4; each PSU must be used $R/2$ times; and each pair of units from different strata must be used $R/4$ times. Efficient BRR designs are based on Hadamard matrices, which are square matrices with entries ± 1 and rows that are mutually orthogonal. It has been conjectured that an Hadamard matrix of order $4k$ exists for every positive integer k . A number of matrices is given in Sloane (2004), and the smallest order for which no Hadamard matrix is known is $4k = 668$. BRR designs can be generated from $R \times R$ Hadamard matrix H as follows. If the (h, r) -th entry H_{hr} of the matrix is $+1$, use the first PSU from stratum h for replicate r ; otherwise, use the second PSU:

$$w_{h1j}^{(r)} = (1 + H_{hr})w_{hij}, \quad w_{h2j}^{(r)} = (1 - H_{hr})w_{hij}.$$

There are several modifications of the BRR method. For a given pattern of included and excluded PSUs in the r -th replicate, a complementary half-sample is obtained by reversing the doubled and excluded units:

$$w_{h1j}^{(rc)} = (1 - H_{hr})w_{hij}, \quad w_{h2j}^{(rc)} = (1 + H_{hr})w_{hij}.$$

These weights are used to compute the complementary half-sample estimate $\hat{\theta}_{BRR}^{(rc)}$. Then additional variance estimates are

$$v_{BRR2}[\hat{\theta}] \equiv v_{BRR-D}[\hat{\theta}] = \frac{1}{4R} \sum_{r=1}^R (\hat{\theta}_{BRR}^{(r)} - \hat{\theta}_{BRR}^{(rc)})^2$$

and

$$v_{BRR3}[\hat{\theta}] \equiv v_{BRR-S}[\hat{\theta}] = \frac{1}{2R} \sum_{r=1}^R (\hat{\theta}_{BRR}^{(r)} - \tilde{\theta})^2 + (\hat{\theta}_{BRR}^{(rc)} - \tilde{\theta})^2,$$

where subindices BRR-D and BRR-S stand for the difference and the sum, respectively.

Fay's modification of BRR (Judkins 1990) is to increase the weight of one PSU by a factor of $2 - k$ and decrease the weight of the other PSU by a factor of k for some $0 \leq k < 1$:

$$w_{hij}^{(r)} = \begin{cases} (2 - k)w_{hij}, & \text{PSU } h, i \text{ is retained} \\ kw_{hij}, & \text{PSU } h, i \text{ is omitted.} \end{cases} \quad (12)$$

The value of $k = 0$ gives the original BRR procedure. The correct scaling factor is $A = 1/(1 - k)^2$.

BRR can be used to correct for small sample biases (Rao and Wu 1985). A bias-corrected estimate is

$$\hat{\theta}_{BRR} = 2\hat{\theta} - \frac{1}{R} \sum_r \hat{\theta}^{(r)}, \quad (13)$$

or, if complementary half-samples are used,

$$\hat{\theta}_{BRRc} = 2\hat{\theta} - \frac{1}{2R} \sum_r (\hat{\theta}^{(r)} + \hat{\theta}^{(rc)}). \quad (14)$$

Stata implements the original v_{BRR} (11) with `svy`, `brr`. By default, the variance formulation (8) is used, and the MSE formulation (9) can be requested with `svy brr, mse` option. Fay's modification is available with `fay(#)` option. The BRR replicate weights can be specified via `svyset`, `brrweights(varlist)`. If no BRR replicate weights are given, the user must provide an Hadamard matrix with `hadamard(matrix)`.

An example of a data set with replicate weights is provided with Stata [SVY] `svy brr` manual data set:

```
. webuse nhanes2brr
. svyset
    pweight: finalwgt
      VCE: brr
      MSE: off
    brrweight: brr_1 brr_2 brr_3 brr_4 brr_5 brr_6 brr_7 brr_8 brr_9 brr_10
              brr_11 brr_12 brr_13 brr_14 brr_15 brr_16 brr_17 brr_18 brr_19
              brr_20 brr_21 brr_22 brr_23 brr_24 brr_25 brr_26 brr_27 brr_28
              brr_29 brr_30 brr_31 brr_32
  Single unit: missing
    Strata 1: <one>
      SU 1: <observations>
    FPC 1: <zero>
```

Note that the default estimation method is `VCE: brr`. Hence, typing `svy: command` will invoke variance estimation by BRR:

```
. svy: logistic highbp height weight age female
(running logistic on estimation sample)
BRR replications (32)
-----|-----|-----|-----|-----|-----|
      1      2      3      4      5
.....
Survey: Logistic regression
                                     Number of obs   =    10351
                                     Population size  = 117157513
                                     Replications     =         32
                                     Design df       =         31
                                     F( 4, 28)        =    174.52
                                     Prob > F        =     0.0000
```

highbp	BRR				
	Odds Ratio	Std. Err.	t	P> t	[95% Conf. Interval]
height	.9688567	.0056915	-5.39	0.000	.9573181 .9805344
weight	1.052489	.0032886	16.37	0.000	1.045803 1.059217
age	1.050473	.0024619	21.01	0.000	1.045464 1.055506
female	.7250086	.0650444	-3.58	0.001	.6037787 .8705796

Most of the design information has been stripped from data, which is often desirable for confidentiality protection. The estimation output only shows the design degrees of freedom and number of replicates. The point estimates are the same as before, whereas the standard errors and the corresponding confidence intervals are slightly different.

Generalizations of BRR to designs with more than two PSUs per stratum have been proposed. Using Galois field theory, Gurney and Jewett (1975) extended BRR to designs with $n_h = p$ in all strata h for a prime number p . Gupta and Nigam (1987) and Wu (1991) discussed the use of mixed orthogonal arrays of strength 2. Sitter (1993) extended this work to orthogonal multi-arrays in which each entry is a q -tuple of distinct numbers between 1 and n_h . As a result, resampling schemes with q PSUs used from each stratum are feasible. All these extensions are quite cumbersome. They require special matrices or abstract algebra constructions, which may not exist for a given design. For this reason, applications of these extensions has been limited.

3.2 The jackknife

While BRR is a replication method unique to complex surveys, the jackknife has been widely used in mainstream statistics (Shao and Tu 1995). In its simplest form for i.i.d. sample of size n , the r -th replicate is obtained by removing the r -th observation, and hence the number of replicates is $R = n$. The appropriate scaling factor in (7) is $A = n - 1$.

In complex survey data, the removed units are PSUs, and the number of replicates is the total number of PSUs, $R = n = n_1 + \dots + n_L$. If PSU k in stratum g is removed

in the r -th replicate, the replicate weights are

$$w_{hij}^{(gk)} = \begin{cases} 0, & h = g, i = k, \\ \frac{n_g}{n_g - 1} w_{hij}, & h = g, i \neq k, \\ w_{hij}, & h \neq g. \end{cases} \quad (15)$$

The jackknife variance estimators can be defined in a number of ways. Let $\hat{\theta}^{(hi)}$ be the estimate obtained with unit h, i removed. Let $\hat{\theta}^h = \sum_i \hat{\theta}^{(hi)} / n_h$ be the stratum average, and $\tilde{\theta}^{(hi)} = n_h \hat{\theta}^h - (n_h - 1) \hat{\theta}^{(hi)}$ be the pseudo-values. Then six jackknife variance estimators are defined as follows:

$$\begin{aligned} v_{J1} &= \sum_h \frac{n_h - 1}{n_h} \sum_i (\hat{\theta}^{(hi)} - \hat{\theta}^h)^2, \\ v_{J2} &= \sum_h \frac{n_h - 1}{n_h} \sum_i (\hat{\theta}^{(hi)} - \hat{\theta})^2, \\ v_{J3} &= \sum_h \frac{n_h - 1}{n_h} \sum_i (\hat{\theta}^{(hi)} - \sum_g \sum_k \hat{\theta}^{(gk)} / n)^2, \\ v_{J4} &= \sum_h \frac{n_h - 1}{n_h} \sum_i (\hat{\theta}^{(hi)} - \sum_h \hat{\theta}^h / L)^2, \\ v_{J5} &= \sum_h \frac{1}{(n_h - 1)n_h} \sum_i (\tilde{\theta}^{(hi)} - \sum_g \sum_k \tilde{\theta}^{(gk)} / n)^2, \\ v_{J6} &= \sum_h \frac{1}{(n_h - 1)n_h} \sum_i (\tilde{\theta}^{(hi)} - 1/L \sum_g 1/n_g \sum_k \tilde{\theta}^{(gk)})^2. \end{aligned} \quad (16)$$

The scaling factor needs to be applied within each stratum to produce correct totals and consistent variance estimates.

Like BRR, the jackknife can also be used to correct for small sample biases of $\hat{\theta}$ with a bias-corrected estimate

$$\hat{\theta}_J = (n + 1 - L) \hat{\theta} - \sum_h (n_h - 1) \hat{\theta}^h. \quad (17)$$

Stata implements the jackknife variance estimation with `svy`, `jackknife`. Either the original design information (strata and PSU) or resampling weights (specified via `svyset`, `jkrweight(varlist)`) should be present in the data set. The default estimator is v_{J1} , and `mse` option invokes estimator v_{J2} .

```
. webuse nhanes2, clear
. svy, vce(jackknife) : logistic highbp height weight age female
(running logistic on estimation sample)
```

(Continued on next page)

```

Jackknife replications (62)
-----|-----|-----|-----|-----|-----|-----
      1      2      3      4      5
.....
.....
Survey: Logistic regression
Number of strata =      31      Number of obs      =      10351
Number of PSUs  =      62      Population size   =      117157513
                                   Replications        =      62
                                   Design df            =      31
                                   F( 4, 28)           =      178.59
                                   Prob > F            =      0.0000
    
```

	Odds Ratio	Jackknife Std. Err.	t	P> t	[95% Conf. Interval]	
highbp						
height	.9688567	.005684	-5.39	0.000	.9573332	.9805189
weight	1.052489	.0032837	16.40	0.000	1.045813	1.059207
age	1.050473	.0024823	20.84	0.000	1.045422	1.055548
female	.7250086	.0641365	-3.64	0.001	.6053227	.8683589

Like linearization, the jackknife is inconsistent for non-smooth statistics. This problem can be ameliorated in designs with large number of PSUs per stratum. For a carefully chosen $k > 1$, *delete-k jackknife* removes k PSUs from the same stratum to form the jackknife replicates. Performance of this method is a complicated interplay between smoothness of the statistics of interest and parameter k . Delete- k jackknife requires $R = \sum_h \binom{n_h}{k} \sim O((n/L)^k)$ replications, notably increasing the computational burden.

3.3 The bootstrap

Inference in parametric statistical procedures is based on sampling distributions of parameter estimates and test statistics. These distributions can often be derived by transformations of the underlying random variables or by asymptotic arguments. The bootstrap provides an alternative paradigm: it mimics the original sampling procedure to obtain approximations to the sampling distributions of the statistics of interest. The bootstrap samples are taken from a distribution that is close, in some suitable sense, to the unknown population distribution. Usually the distribution from which the bootstrap samples are taken is the the empirical distribution of the data.

Let the sample data x_1, \dots, x_n be i.i.d. from distribution F characterized by parameter $\theta = T(F)$. The empirical distribution function of the data is F_n , and the associated parameter estimate is $\hat{\theta}_n = T(F_n)$. The bootstrap takes a simple random sample with replacement (x_1^*, \dots, x_m^*) of size m from x_1, \dots, x_n . The empirical distribution function of the bootstrap sample is F_m^* , and the associated parameter estimate is $\hat{\theta}_m^* = T(F_m^*)$. The bootstrap distribution of $\hat{\theta}_m^*$ is obtained by taking different bootstrap samples and computing $\hat{\theta}_m^*$ for each of them.

The plug-in principle of the bootstrap, illustrated in Fig. 1, states that relation of

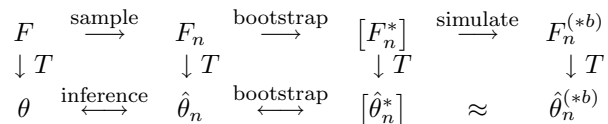


Figure 1: The bootstrap principle.

the bootstrap values $\hat{\theta}_m^*$ to $\hat{\theta}_n$ is approximately the same as that of $\hat{\theta}_n$ to the unknown parameter θ . Typically, but not necessarily, $m = n$. If this is the case, the bootstrap estimates of the moments and the distribution function of $\hat{\theta}_n$ are

$$\begin{aligned}
\text{Bias}[\hat{\theta}_n] &= \mathbb{E}[\hat{\theta}_n - \theta] \doteq \mathbb{E}^*[\hat{\theta}_n^* - \hat{\theta}_n], \\
\mathbb{V}[\hat{\theta}_n] &= \mathbb{E}[(\hat{\theta}_n - \mathbb{E}[\hat{\theta}_n])^2] \doteq \mathbb{E}^*[(\hat{\theta}_n^* - \mathbb{E}[\hat{\theta}_n^*])^2], \\
\text{MSE}[\hat{\theta}_n] &= \mathbb{E}[(\hat{\theta}_n - \theta)^2] \doteq \mathbb{E}^*[(\hat{\theta}_n^* - \theta_n)^2], \\
\text{cdf}_{\hat{\theta}_n}(x) &= \text{Prob}[\hat{\theta}_n - \theta < x] \doteq \text{Prob}^*[\hat{\theta}_n^* - \hat{\theta}_n < x],
\end{aligned} \tag{18}$$

where the starred quantities are taken with respect to the bootstrap distribution. In sufficiently simple situations (e.g., when the estimates are asymptotically normal), the bootstrap approximations (18) work well, and the ad-hoc equivalences denoted by \doteq correspond to consistent estimates. The particular strength of the bootstrap is the last equation of (18). The bootstrap accounts for asymmetry of the sampling distributions and gives better one-sided confidence interval coverage than the confidence intervals based on asymptotic normality (Efron and Tibshirani 1994; Shao and Tu 1995).

The theory of the bootstrap is based on the complete enumeration of all possible samples of size m from the distribution F_n . As the complete bootstrap requires n^m samples, it can only be performed for very small samples. For instance, even with $n = 5$, the complete listing gives $5^5 = 3125$ samples. In practice, instead of the complete bootstrap, a large number R of random samples with replacement from the original data are taken, the statistics of interest are computed for each bootstrap sample, and Monte Carlo distributions of the resulting statistics are used to conduct inference. Two approximations are thus taken. The sampling distributions of the statistics of interest are approximated by the complete bootstrap distributions, and the complete bootstrap distributions are in turn approximated by simulation. Conceptually, the number of samples R should be large enough so that the second error due to simulation is much smaller than the first error due to the bootstrap method (Efron and Tibshirani 1994, Sec. 6.4). In practice, the number of the bootstrap replicates R is often restricted by computational burden, and is usually taken to be between 100 and 1000.

Special bootstrap schemes exist to control the error induced by simulation. The package `bsweights` (Section 4) implements one of such schemes, the balanced bootstrap. The motivation for the balanced bootstrap is that for statistics with known means and variance estimates (such as the sample mean), an attempt should be made to match the moments of the bootstrap distribution with the known ones. This is achieved by carefully tracking the number of times the units appear in the bootstrap samples. The first order balance is achieved when each unit is included into the bootstrap samples the

same number of times (Davison et al. 1986). The first order balance removes simulation noise from the mean of the bootstrap distribution, and hence from the estimates of bias. The second order balance is achieved when each pair of units is included the same number of times (Graham et al. 1990). The second order balance removes simulation noise from the estimates of variance.

Unfortunately, the bootstrap does not solve each and every problem in statistical inference. Examples where the bootstrap fails are abundant (Canty et al. 2006; Shao and Tu 1995, Sec. 3.6.). Complex surveys data is one of them. More sophisticated resampling schemes and/or estimation procedures have to be employed in such situations.

We start developing the bootstrap for complex survey data by considering the following naïve bootstrap (NBS) scheme. To construct the r -th replicate, take a simple random sample with replacement of n_h units from the original data in stratum h ; repeat independently across strata; estimate the parameter of interest $\hat{\theta}^{(*r)}$; repeat R times and estimate the variance using (7). Rao and Wu (1988) demonstrated that even in the simple case of the stratified mean (4), the variance of the complete NBS distribution is

$$\mathbb{V}_{NBS}^*[\bar{x}^*] = \sum_h \frac{W_h^2}{n_h} \frac{n_h - 1}{n_h} s_h^2,$$

instead of

$$v_{str}[\bar{x}] = \sum_h \frac{W_h^2}{n_h} s_h^2.$$

If the number of PSUs per stratum is small (and this number is often as low as $n_h = 2$), the bootstrap estimator is biased and inconsistent. The issue also exists in the bootstrap for i.i.d. data, but is of a lesser importance since the bias disappears as $n \rightarrow \infty$.

How can the bootstrap variance be rectified? First, if all strata have the same number of units, $n_h = n_0$ for all h , then the rescaled variance $(n_0/(n_0 - 1)) \mathbb{V}^*[\bar{x}^*]$ will be unbiased and consistent. For unequal strata sizes, Rao and Wu (1988) proposed the following rescaling bootstrap (RBS) procedure. A simple random sample with replacement of m_h out of n_h units is taken, and internally scaled pseudo-values

$$\tilde{x}_{hi}^{(r)} = \bar{x}_h + m_h^{1/2} (n_h - 1)^{-1/2} (x_{hi}^{(*r)} - \bar{x}_h) \quad (19)$$

are used in computing the variances of the moments and their functions. Here, $\{x_{hi}^{(*r)}, i = 1, \dots, m_h\}$ is the r -th sample taken from the h -th stratum, and $\bar{x}_h = \sum_i x_{hi}/n_h$ is the estimated stratum mean. Rao et al. (1992) extended RBS method to cover estimating equations (5). They considered replicate weights implied by RBS and demonstrated that the necessary internal scaling can be achieved by the replicate weights

$$w_{hij}^{(*r)} = \left(1 - m_h^{1/2} (n_h - 1)^{-1/2} + m_h^{1/2} (n_h - 1)^{-1/2} \frac{n_h}{m_h} m_{hi}^{(*r)} \right) w_{hij}, \quad (20)$$

where $m_{hi}^{(*r)}$ is the bootstrap frequency of unit h, i , that is, the number of times PSU h, i was used in forming the r -th bootstrap replicate. This method of internal scaling is implemented in `bsweights` package described below in Section 4.

How should the bootstrap sample size m_h be chosen? First, note that there is no need for internal scaling if $m_h = n_h - 1$. Second, Rao and Wu (1988) provided theoretical arguments showing that when the strata variances are known, the choice $m_h = (n_h - 2)^2 / (n_h - 1) \approx n_h - 3$ for $n_h > 3$ allows to match the third order moments of the bootstrap distribution with those of the theoretical sampling distribution. Simulation evidence indicated that the bootstrap estimators with $m_h = n_h - 3$ are unstable for moderate sample sizes $n_h = 5$ and unknown strata variances (Kovar et al. 1988). Finally, note that when $m_{hi}^{(*r)} = 0$, the replicate weight is $w_{hij}^{(*r)} = (1 - m_h^{1/2} (n_h - 1)^{-1/2}) w_{hij}$. If $m_h > n_h - 1$, this weight will become negative, which may lead to violations of natural ranges for parameters such as quantiles, distribution functions, variances or correlations.

Given the above considerations, $m_h = n_h - 1$ appears to be a good choice that ensures efficiency of the bootstrap estimators without producing any artifacts like range restriction violations.

How should the number of replications R be chosen? When the data are i.i.d., we argued that this number should be chosen to make Monte Carlo variability of the bootstrap variance estimates sufficiently small. For complex surveys, it is also desirable that the number of replicates is at least as large as the design degrees of freedom, $n - L$. The design degrees of freedom is the largest possible rank of the covariance matrix of the coefficient estimates. The choice $R < n - L$ will not allow the bootstrap to provide this highest possible rank. The degrees of freedom may not be an issue if $n - L$ is a sufficiently large number (e.g., exceeds 100).

With these points in mind, let us outline the preferred bootstrap procedure.

1. From stratum h , take a simple random sample of $m_h = n_h - 1$ PSUs with replacement.
2. Compute the weights using simplified version of (20):

$$w_{hij}^{(*r)} = \frac{n_h}{n_h - 1} m_{hi}^{(*r)} w_{hij}.$$

3. Estimate $\hat{\theta}^{(*r)}$.
4. Repeat steps 1–3 R times, compute

$$v_{BWR}[\hat{\theta}] = \frac{1}{R} \sum_{r=1}^R (\hat{\theta}^{(*r)} - \bar{\hat{\theta}})^2. \quad (21)$$

Subindex BWR stands for the bootstrap with replacement, originally proposed by McCarthy and Snowden (1985). BWR is a special case of RBS with $m_h = n_h - 1$.

3.4 Extensions of the bootstrap methods for complex survey data

Alternative and complementary bootstrap procedures for complex survey data have been proposed in the literature.

McCarthy and Snowden (1985) and Sitter (1992a) proposed the bootstrap without replacement (BWO) procedure that first constructs a finite pseudopopulation, and then takes samples from it using a sampling design similar to the original one. Let $m_h = n_h - (1 - f_h)$ and $k_h = \frac{N_h}{n_h}(1 - \frac{1-f_h}{n_h})$. Create the pseudopopulation by replicating units $\{x_{hi}\}$ in the h -th stratum k_h times. To obtain the r -th replicate, take a simple random sample without replacement of m_h units from pseudopopulation stratum h , for all $h = 1, \dots, L$. Compute $\hat{\theta}^{(r)}$, repeat R times, and compute the variance estimate using (7). For non-integer m_h and k_h , randomization between the nearest integers can be used.

The mirror-match bootstrap (MMB) of Sitter (1992b) draws a simple random sample of $m_h < n_h$ PSUs from stratum h without replacement, repeating the process $k_h = \frac{n_h(1 - f_h^*)}{m_h(1 - f_h)}$ times to obtain one bootstrap replicate. Here, $f_h = n_h/N_h$ is the original sampling fraction, $f_h^* = m_h/n_h$ is the bootstrap sampling fraction, and the bootstrap sample size is $k_h m_h$. Note that when $m_h = 1$, we obtain the BWR scheme in which the bootstrap sample size $n_h - 1$ allows to avoid the issue of internal scaling. The choice of $f_h^* = f_h$ allows to match the third order moments. Like in the BWO method, randomization can be used if m_h is non-integer.

The mean bootstrap (Yeo et al. 1999; Yung 1997) processes the bootstrap samples in batches of size K . To compute the r -th replicate weight variable, the bootstrap frequencies are averaged across each batch, and the average frequency

$$\bar{m}_{hi}^{(*r)} = \frac{1}{K} \sum_{k=(r-1)K+1}^{rK} m_{hi}^{(*k)}$$

is used instead of $m_{hi}^{(*r)}$ to scale weights according to (20). The total number of the bootstrap replications in this method is the product RK , while the number of the replicate weight variables is R . The scaling factor A in (7) needs to be set equal to K to ensure consistency. There are some similarities between this scheme and Fay's modification of BRR. The motivation for both schemes is confidentiality protection: PSUs should never receive zero replicate weights. Also, the scaling factor A needs to be increased to compensate for smaller variability of the replicate weights.

Balanced versions of the bootstrap for stratified samples have been studied by Nigam and Rao (1996). While the first order balance can be easily achieved, the second order balanced bootstrap schemes are only known for some special cases. The second order balance conditions impose very tight restrictions on the number of times pairs of units (within and between strata) are used in the resampling scheme. There are designs for which these conditions cannot be satisfied. Nigam and Rao (1996) demonstrated how to construct the second-order balanced bootstrap schemes for designs with constant sample sizes across strata, $m_h = n_h = n_0$ for all h , when n_0 is an even number or a prime power.

Balanced bootstraps have a lot in common with BRR. If $n_h = 2$, the bootstrap scheme that avoids internal rescaling has the bootstrap sample size $m_h = n_h - 1 = 1$, that is, the bootstrap samples are random half-samples. The second order conditions

dictate that the number of times units from different strata are resampled together is the same for all pairs of units. These conditions are identical to the BRR balance conditions. Hence, BRR can be viewed as the second-order balanced bootstrap resampling scheme.

Special care needs to be taken of missing and imputed data in the bootstrap for complex survey data. The bootstrap variance estimates must take into account variability due to three processes: (i) the sampling process, (ii) the non-response process, and (iii) the imputation process. Suppose the sample data \mathbf{X} consist of available data on respondents \mathbf{X}_R and missing data on non-respondents, \mathbf{X}_M . Each data entry x_{hijk} on variable x_k for unit h, i, j has an associated flag a_{hijk} showing availability of x_{hijk} : $a_{hijk} = 1$ if x_{hijk} is observed and $a_{hijk} = 0$ if x_{hijk} is missing. Suppose the missing data are imputed using a deterministic (ratio or regression imputation) or a random (hot-deck, ratio or regression imputation with added noise) method: $\eta_{hijk} = \mathcal{J}(\mathbf{X}_R; h, i, j, k)$ if $a_{hijk} = 0$. Shao (1996) and Shao and Sitter (1996) show that to ensure consistency of the bootstrap variance estimates with respect to the sampling distribution, the bootstrap procedure needs to re-impute the missing data for each bootstrap sample r . Shao and Sitter (1996) propose the following procedure:

1. Draw a simple random sample $(x_{hij}^{(*r)}, a_{hij}^{(*r)})$ of size $n_h - 1$ with replacement from the (imputed) data (x_{hij}, a_{hij}) in stratum h . Combine the replicate data across strata.
2. Apply the original imputation procedure to the missing components of the resampled data: $\eta_{hijk}^* = \mathcal{J}(\mathbf{X}_R^*; h, i, j, k)$ if $a_{hijk}^* = 0$. That is, for all original missing data that were resampled, repeat the imputation using the resampled available data.
3. Obtain the estimate of interest $\hat{\theta}^{(*r)}$.

Steps 1–3 are repeated R times, and the resulting procedure accounts for both imputation and complex survey structure (as it utilizes design-consistent bootstrap with the bootstrap sample size $m_h = n_h - 1$).

3.5 Comparison of estimators and relations between them

The linearization, jackknife, bootstrap and (where applicable) BRR variance estimators are estimating the same quantity, the variance $\mathbb{V}[\hat{\theta}] = \sigma^2$ of statistic $\hat{\theta}$. Can we identify conditions under which some estimators perform better than others? (As noted by Eltinge (1996), different goals of variance estimation may lead to different estimators being preferred.) A number of comparisons, both theoretical and empirical (by simulation) have been made in the literature.

In the special case of linear statistics of moments such as totals, the variance estimators coincide: $v_L = v_J = v_{BRR} = v_{BOOT}$ covering all versions of the jackknife and BRR estimators, and bootstrap estimators RBS, MMB, BWR and BWO (except the naïve bootstrap).

Consistency of various versions of v_{BRR} and v_J , as well as v_L , was established by Krewski and Rao (1981) for smooth functions under the important setting of bounded number of PSUs per stratum and number of strata $L \rightarrow \infty^1$. They also found that in terms of two-sided confidence interval coverage, BRR was the best performing method, followed by the jackknife, and then by linearization. In terms of stability, i.e., the mean squared error $\mathbb{E}[(v_m - \sigma^2)^2]$, the ordering was reversed.

Second order analysis² of linearization, BRR and the jackknife was conducted by Rao and Wu (1985). Under general regularity conditions, the bias of nonlinear statistic $\hat{\theta}$ is $O(n^{-1})$. The BRR bias corrected estimates (13) and (14), as well as the jackknife bias corrected estimate (17), reduce bias to $O(n^{-2})$. In other words, these estimates are closer to the true values than $\hat{\theta}$ based on a single estimation with the original sampling weights.

For general smooth nonlinear functions, the six jackknife variance estimators (16) are equivalent to one another with the order $O_p(n^{-3})$, and asymptotically equivalent to the linearization estimator v_L :

$$v_J = v_L(1 + O_p(n^{-1})).$$

When $n_h = 2$, BRR method is applicable, with

$$v_{BRR2} = v_L(1 + O_p(n^{-1})), \quad v_{BRR3} = v_L(1 + O_p(n^{-1})),$$

while

$$v_{BRR1} = v_L(1 + O_p(n^{-1/2}))$$

for general nonlinear statistic $\hat{\theta}$. Performance of the jackknife estimators improves to

$$v_J = v_L(1 + O_p(n^{-2})).$$

No clear ordering of biases of v_m (i.e., the difference $\mathbb{E}[v_m] - \sigma^2$) can be established. Rao and Wu (1985) found conditions under which each of v_L , v_{J2} , and various versions of BRR estimators had the smallest biases. This is an interesting observation, since the linearization estimator v_L is usually considered the “golden standard” (if it is applicable for a given estimation problem). Also, Valliant (1996) discussed several situations in which the jackknife estimator exhibited better properties than the linearization estimator in model-based approach to survey inference, in ratio estimation, and in poststratification.

1. Additional regularity conditions include continuously differentiable functions of the sample moments with non-zero derivatives in the neighborhood of the population mean, and Lyapunov-type conditions on higher moments necessary to establish a suitable version of the Central Limit Theorem.

2. The first order analysis establishes consistency: $v_m/\sigma^2 \rightarrow 1$ in probability as $n \rightarrow \infty$, where σ^2 is the true variance. The second order analysis establishes biases of the form $n^\alpha(\mathbb{E} v_m - \sigma^2)$ and stability or mean squared errors of the estimators, $n^\beta \mathbb{E}[(v_m - \sigma^2)^2]$ for some $\alpha, \beta > 0$. For a sequence of random variables V_n , we shall write $V_n = O_p(n^{-\alpha})$ if V_n is bounded in probability: for all $\epsilon > 0$, there exist M, n_0 such that $\text{Prob}[n^\alpha |V_n| > M] < \epsilon$ for all $n \geq n_0$. For instance, if V_n satisfies the central limit theorem, then $V_n - \mathbb{E}[V_n] = O_p(n^{-1/2})$.

Rao and Wu (1988) demonstrated that the rescaling bootstrap variance estimator coincides with all other variance estimator in the linear case, and

$$v_{RBS} = v_L(1 + O_p(n^{-1}))$$

in nonlinear cases. Also, the distribution estimators

$$H_{RBS}(x) = \text{Prob}^*[\sqrt{n}(\hat{\theta}^* - \hat{\theta})]$$

and

$$G_{RBS}(x) = \text{Prob}^*[(\hat{\theta}^* - \hat{\theta})/\sqrt{v_L^*}]$$

are consistent for the distribution of $\hat{\theta}$ and t -statistic based on the linearization estimator, respectively.

Overall, the jackknife and linearization tend to exhibit similar performance. They are more stable for smooth functions, but inconsistent for non-smooth functions. The method that is applicable for all statistics is the bootstrap (and its kin BRR for designs with $n_h = 2$). The bootstrap can additionally provide more accurate one-sided confidence intervals and better balance of the tail probabilities of two-sided confidence intervals. However, this versatility comes at the price of a lesser stability and longer confidence intervals.

4 bsweights package

4.1 Syntax

```
bsweights prefix , reps(#) n( subsample size ) [ replace average(#)
  balanced calibrate(call to weight calibration routine) verbose nosvy
  seed(#) ]
```

The *call to weight calibration routine* is

calibration_program @ *additional arguments*

Returned values:

r(balanced) = 1 if the first order balance was achieved, 0 otherwise.

4.2 Options

reps(#) specifies the number of the replicate weight variables to be generated. This is a required option.

n(#) specifies the bootstrap sample size m_h . If a positive integer number # is specified, then $m_h = \#$. If a non-positive integer number # is specified, then $m_h = n_h - |\#|$. This is a required option.

`replace` requests that the weight variables will be created anew. Use with caution, it will **drop** the existing `prefix*` variables!

`average(#)` implements the mean bootstrap. The bootstrap frequency counts are averaged across the given number of replications. If the bootstrap weights are created using this option, the end user should specify `vfactor()` option of `bs4rw`. The total number of replications is the product of the numbers in `reps` and `average` options.

`balanced` requests the first order balancing of the bootstrap, if possible. See remarks below.

`dots` provides additional output.

`calibrate` allows to call another program to adjust the weights for post-stratification and non-response. See remarks below.

`verbose` provides output from the weights calibration call.

`nosvy` explicitly states that the data are not of a survey format.

`seed` specifies the seed for the random number generator. See [D] **generate**.

4.3 Remarks

Calibration

When rescaling the replicate weights, `bsweights` can make calls to *calibration_program* substituting the current replicate weight variable being processed for symbol `@`. For instance, if the user specifies

```
. bsweights bsw , ... calibrate( do adjust @ )
```

then `bsweights` will issue consecutive commands

```
do adjust bsw1
do adjust bsw2
...
```

In turn, the do-file `adjust.do` might contain

```
args weight_var
...
replace `weight_var' = ...
...
```

See Examples 5–6 below.

The weight adjustments are taking place after the internal scaling (20). It is the responsibility of the user to provide correct treatment of the resampling weights in their calibration procedures. Option `verbose` provides output from calibration commands for debugging purposes.

Note that for proper results, the survey agency must specify the original probability weights rather than the final sampling weights as inputs to `bsweights`. The same adjustment procedure that produces the publicly available weights from the probability weights should be applied to the bootstrap weights.

Balanced bootstraps

The balanced bootstrap in `bsweights` is implemented using permutation algorithm BB2 of Gleason (1988). Only the first order balance is achieved by this algorithm. For stratified samples, the balanced bootstrap is conducted in each stratum independently.

For the bootstrap scheme to be first order balanced, certain simple bookkeeping conditions must be satisfied. Namely, it is necessary that the total number of units recycled from stratum h across all bootstrap replicates be divisible by n_h for all strata h . It will be satisfied if R (or KR in the case of the mean bootstrap) is divisible by the least common multiple of n_1, \dots, n_L . The returned value `r(balanced)` shows whether the first order balance was achieved.

The use of the bootstrap weights

There are two ways to use the bootstrap weights generated by `bsweights`. In the examples below, we use `bs4rw` package written by Jeff Pitblado of StataCorp. This is an analogue of the official `bootstrap` command that uses the replicate weights instead of actually resampling the data in Stata memory. To install `bs4rw` package, type `findit bs4rw` and follow the instructions. Let us provide a few general comments about `bs4rw`.

The command called by `bs4rw` with different sets of weights must accept probability weights [`pweight=exp`] or importance weights [`iweight=exp`] as a part of its syntax. This rules out a number of important commands, such as `correlate` or `xtmixed`.

The first call `bs4rw` makes is to find the point estimates. Therefore, the command specification must contain the original weights or `svy` prefix, otherwise the point estimates reported in the output of `bs4rw` will be incorrect.

The bootstrap postestimation summaries (estimates of bias and various confidence intervals) are available with `estat bootstrap`, see [R] [bootstrap postestimation](#).

An alternative way to use the replicate weights is outlined by Phillips (2004). As long as both the bootstrap and BRR use the same replication variance estimation formula (7) with the same scaling factor $A = 1$, one can trick Stata (or any other software that performs BRR estimation) into accepting the bootstrap weights as the BRR weights. See Example 4 below.

5 Examples

Let us provide several examples of how `bsweights` and `bs4rw` can be used with Stata example data sets.

5.1 Complex survey data

The complex survey data examples will be based on the aforementioned NHANES II data.

To demonstrate the flexibility of `bsweights`, we shall collapse some of the strata, producing a pseudo-design with 7 pseudo-strata. The numbers of PSUs in these strata are 4, 4, 8, 8, 12, 10 and 16. We also need to recode the PSU variable to make it run from 1 to 62.

```
. webuse nhanes2
. gen cstrata = floor( sqrt( 2*strata-1) )
. egen upsu = group( strata psu )
. svyset upsu [pw=finalwgt], strata(cstrata)

      pweight: finalwgt
           VCE: linearized
Single unit: missing
Strata 1: cstrata
   SU 1: upsu
   FPC 1: <zero>

. svy: logistic highbp height weight age female
(running logistic on estimation sample)
Survey: Logistic regression
Number of strata   =          7          Number of obs       =       10351
Number of PSUs    =          62          Population size      =    117157513
                                           Design df           =          55
                                           F( 4, 52)           =       205.17
                                           Prob > F             =       0.0000
```

highbp	Linearized		t	P> t	[95% Conf. Interval]	
	Odds Ratio	Std. Err.				
height	.9688567	.0062847	-4.88	0.000	.9563433	.9815338
weight	1.052489	.0031645	17.01	0.000	1.046166	1.05885
age	1.050473	.0023319	22.18	0.000	1.04581	1.055157
female	.7250086	.073301	-3.18	0.002	.5920358	.8878473

Example 1: an arbitrary bootstrap scheme

In the first example, we shall create bootstrap weights with arbitrarily chosen $R = 100$ replications and the bootstrap sample size $m_h = 2$:

(Continued on next page)

```
. bsweights bw , reps(100) n(2) seed(10101) dots
Running bsample 100 times .....
> .....

Rescaling weights
.....
> .....

Warning: the first order balance was not achieved
. bs4rw , rw(bw*) : logistic highbp height weight age female [pw=finalwgt]
(running logistic on estimation sample)
BS4Rweights replications (100)
-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
      1      2      3      4      5
..... 50
..... 100

Logistic regression          Number of obs      =      10351
                             Replications          =         100
                             Wald chi2(4)           =       730.54
                             Prob > chi2            =         0.0000
                             Pseudo R2              =         0.1527

Log pseudolikelihood = -2961.5987
```

highbp	Observed	Bootstrap	z	P> z	Normal-based	
	Odds Ratio	Std. Err.			[95% Conf. Interval]	
height	.9688567	.0062701	-4.89	0.000	.9566451	.9812242
weight	1.052489	.0036141	14.90	0.000	1.045429	1.059596
age	1.050473	.0024032	21.52	0.000	1.045773	1.055194
female	.7250086	.0775258	-3.01	0.003	.587927	.8940523

The standard errors are within 5% of the linearization based ones. The option `dots` provides additional output, including the warning about the lack of balance.

Example 2: a balanced bootstrap scheme

In this example, the necessary conditions for the first order balance (Section 4.3) will be taken into account to set up a first-order balanced scheme. The least common multiple of the strata sizes is 240. The necessary condition for the first order balance is that the product $m_h R$ is divisible by 240 for all strata. We can choose the replication scheme with $R = 80$ and $m_h = 3 \leq n_h - 1$ for all h (the new option is underlined):

```
. bsweights bw , reps(80) n(3) seed(10101) balanced dots
Balancing within strata:
.....

Rescaling weights
.....

. bs4rw , rw(bw*) : logistic highbp height weight age female [pw=finalwgt]
(running logistic on estimation sample)
BS4Rweights replications (80)
-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
      1      2      3      4      5
..... 50
.....
```

(Continued on next page)

```

Logistic regression                Number of obs   =   10351
                                   Replications     =     80
                                   Wald chi2(4)       =   786.21
                                   Prob > chi2       =   0.0000
                                   Pseudo R2        =   0.1527

Log pseudolikelihood = -2961.5987
    
```

highbp	Observed	Bootstrap	z	P> z	Normal-based	
	Odds Ratio	Std. Err.			[95% Conf. Interval]	
height	.9688567	.0057047	-5.37	0.000	.9577399	.9801025
weight	1.052489	.003199	16.83	0.000	1.046237	1.058777
age	1.050473	.0021039	24.59	0.000	1.046358	1.054605
female	.7250086	.0716213	-3.26	0.001	.5973868	.8798946

Note that there is no warning about lack of balance in the output of `bsweights`. The standard errors are close to the ones obtained earlier.

The above balanced bootstrap scheme does not use information from larger strata very effectively, as only 3 out of 16 PSUs are resampled from the largest stratum. Better schemes would have few units omitted from every strata, keeping m_h close to n_h . Given the number of PSUs per stratum in the current data configuration, if $m_h = n_h - 1$, the bootstrap sample sizes are odd numbers ranging from 3 to 15, and hence $R = 240$ replicates need to be taken. If we want to reduce computational burden, we can set $m_h = n_h - 2$. Then the bootstrap sample sizes are even numbers ranging from 2 to 14, and the number of replicates can be reduced to 120.

Example 3: mean bootstrap

Another efficient way to reduce computational burden is to use the mean bootstrap. It creates R replicate weight variables from RK bootstrap replicates. The number of replicates that are averaged over by the mean bootstrap to create a single replicate weight, K , must be carried over to `bs4rw` with `vfactor(K)` option.

```

. bsweights bw , reps(120) average(10) n(-1) balanced seed(10101) dots
Balancing within strata:
.....
Rescaling weights
.....
> .....
. bs4rw , rw(bw*) vf(10) : logistic highbp height weight age female
> [pw=finalwgt]
(running logistic on estimation sample)
BS4Rweights replications (120)
-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
      1   2   3   4   5
..... 50
..... 100
.....
    
```

(Continued on next page)

```

Logistic regression          Number of obs   =   10351
                             Replications      =     120
                             Wald chi2(4)       =   711.62
                             Prob > chi2       =     0.0000
Log pseudolikelihood = -2961.5987          Pseudo R2      =     0.1527

```

highbp	Observed	Bootstrap	z	P> z	Normal-based	
	Odds Ratio	Std. Err.			[95% Conf. Interval]	
height	.9688567	.0060641	-5.05	0.000	.957044	.9808152
weight	1.052489	.0033971	15.85	0.000	1.045851	1.059168
age	1.050473	.0024524	21.09	0.000	1.045678	1.055291
female	.7250086	.0725368	-3.21	0.001	.5959102	.8820749

Since the effective number of the bootstrap replications $RK = 120 \cdot 10 = 1200$ is larger than in earlier examples, the standard errors in this scheme are more stable. Note the use of `vfactor()` option in the call to `bs4rw`.

Example 4: the bootstrap weights as BRR weights

As mentioned in Section 4.3, it is possible to use the existing `svy` commands with the bootstrap weights by specifying them as the BRR weights. The weight variables will be provided to `svyset` with `brrweight` option, and Fay's scaling correction is $1 - 1/\sqrt{K}$:

```

. local mean2fay = 1-sqrt(1/10)
. svyset [pw=finalwgt] , vce(brr) brrweight( bw* ) fay(`mean2fay`)
      pweight: finalwgt
              VCE: brr
              MSE: off
      brrweight: bw1 bw2 bw3 bw4 bw5 bw6 bw7 bw8 bw9 bw10 bw11 bw12 bw13 bw14
                bw15 bw16 bw17 bw18 bw19 bw20 bw21 bw22 bw23 bw24 bw25 bw26
                bw27 bw28 bw29 bw30 bw31 bw32 bw33 bw34 bw35 bw36 bw37 bw38
                bw39 bw40 bw41 bw42 bw43 bw44 bw45 bw46 bw47 bw48 bw49 bw50
                bw51 bw52 bw53 bw54 bw55 bw56 bw57 bw58 bw59 bw60 bw61 bw62
                bw63 bw64 bw65 bw66 bw67 bw68 bw69 bw70 bw71 bw72 bw73 bw74
                bw75 bw76 bw77 bw78 bw79 bw80 bw81 bw82 bw83 bw84 bw85 bw86
                bw87 bw88 bw89 bw90 bw91 bw92 bw93 bw94 bw95 bw96 bw97 bw98
                bw99 bw100 bw101 bw102 bw103 bw104 bw105 bw106 bw107 bw108
                bw109 bw110 bw111 bw112 bw113 bw114 bw115 bw116 bw117 bw118
                bw119 bw120
      fay: .68377223
Single unit: missing
Strata 1: <one>
SU 1: <observations>
FPC 1: <zero>

```

(Continued on next page)

```
. svy , vce(brr) : logistic highbp height weight age female
(running logistic on estimation sample)
BRR replications (120)
-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 1 | 2 | 3 | 4 | 5
..... 50
..... 100
.....
Survey: Logistic regression          Number of obs   =    10351
                                   Population size  =  117157513
                                   Replications    =     120
                                   Design df       =     119
                                   F( 4, 116)      =    174.88
                                   Prob > F       =     0.0000
```

highbp	BRR		t	P> t	[95% Conf. Interval]	
	Odds Ratio	Std. Err.				
height	.9688567	.0060387	-5.08	0.000	.9569729	.9808881
weight	1.052489	.0033829	15.92	0.000	1.045811	1.059208
age	1.050473	.0024421	21.18	0.000	1.045649	1.05532
female	.7250086	.0722339	-3.23	0.002	.595203	.883123

The advantage of using the bootstrap replicate weights as BRR weights is that Stata post-svy-estimation commands, such as the design effect estimation, can be invoked seamlessly:

```
. estat effect
```

highbp	BRR		DEFF	DEFT
	Coef.	Std. Err.		
height	-.0316386	.0062328	1.2491	1.11763
weight	.0511575	.0032142	1.81839	1.34848
age	.0492406	.0023248	1.03102	1.01539
female	-.3215718	.0996318	.972726	.986269
_cons	-2.858969	1.085424	1.27086	1.12732

However, since `svy, vce(brr)` makes additional assumptions about the design, some of the inferential statistics will be computed incorrectly. Most importantly, the design degrees of freedom will be assumed to be equal to the number of replicates R , which may be much greater than the degrees of freedom of the actual design. Furthermore, the inferential statistics that utilize these degrees of freedom will also be questionable. Among these inferential statistics are the overall F -test, t -tests of individual coefficients, and confidence intervals based on the t -distribution. Implementation of the bootstrap estimation with `bs4rw` is free of these problems since it relies on asymptotic normality of the estimates. Thus, the analysis in Example 3 contained χ^2 instead of F -test, and the confidence intervals that were based on the normal distribution, and hence slightly shorter.

If the usual rescaling bootstrap rather than the mean bootstrap is used, Fay's cor-


```

Logistic regression           Number of obs   =   10351
                             Replications       =     120
                             Wald chi2(4)        =   863.55
                             Prob > chi2        =   0.0000
                             Pseudo R2         =   0.1527
Log pseudolikelihood = -2961.5987

```

highbp	Observed	Bootstrap	z	P> z	Normal-based	
	Odds Ratio	Std. Err.			[95% Conf. Interval]	
height	.9688567	.0058832	-5.21	0.000	.9573943	.9804564
weight	1.052489	.0032217	16.71	0.000	1.046193	1.058822
age	1.050473	.0022441	23.05	0.000	1.046084	1.054881
female	.7250086	.0718354	-3.25	0.001	.5970412	.880404

Note that each dot under “Rescaling weights” include both the internal scaling (20) and a call to the calibration program.

Generally speaking, calibration conflicts with the mean bootstrap. Calibration needs to be performed for every bootstrap sample after the weights are rescaled. The mean bootstrap, however, takes averages across the bootstrap replicates, and then applies rescaling. An exception to this incompatibility is domain estimation described next.

Example 6: domain estimation

Domain estimation is a special case of calibration where the weights outside the domain are set to zero. Suppose we want to conduct a separate analysis for females only, reproducing subpopulation estimation of Example 2 of [SVY] **svy estimation**. The variable `female` takes values 0 and 1 for males and females, so the analogue to `subpop(female)` option of `svy` will be achieved by the following calibration program:

```

. capture program drop CalSubpop
. program define CalSubpop
1.   args wvar
2.   replace `wvar' = `wvar' * female
3. end

```

Now we can run `bsweights` calling `CalSubpop` for the calibration step:

```

. drop bw*
. bsweights bw , reps(120) average(10) n(-2) balanced dots calibrate(CalSubpop @)
> seed(10101) replace
Warning: combination of calibration with mean bootstrap can lead to incorrect
> results
Balancing within strata:
.....
Rescaling weights
.....
> .....

```

(Continued on next page)

which would slow down the estimation. The last line of the program calls the estimation command and leaves the vector of parameter estimates $e(b)$ in Stata memory.

```
. pro def myicereg
1.  syntax [if] [in] [pw iw /] , [*]
2.  * local macro `weight' contains the weight type
.  * local macro `exp' contains the weight variable
.
.  cap drop imp*lead
3.  uvis regress lnlead region1 region2 region3 rural black orace age age2
>  zinc copper vitaminc albumin tbc tresult tresult [pw=`exp'],
>  gen(implnlead)
4.  gen implead = exp( implnlead )
5.  logistic highbp height weight age female implead [pw=`exp']
6.  end
```

For greater precision, we shall use a larger number of replicates $R = 240$.

```
. bsweights bw , n(-1) reps(240) dots balanced seed(10101)
Balancing within strata:
.....
Rescaling weights
.....
> .....
> .....
> .....
. bs4rw , rw(bw*) : myicereg [pw=finalwgt]
(running myicereg on estimation sample)
BS4Rweights replications (240)
-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 1 | 2 | 3 | 4 | 5 |
..... 50
..... 100
..... 150
..... 200
.....
Logistic regression                               Number of obs   =   10351
                                                    Replications    =     240
                                                    Wald chi2(5)    =   593.42
                                                    Prob > chi2     =   0.0000
Log pseudolikelihood = -1991.3363                Pseudo R2       =   0.1641
```

highbp	Observed	Bootstrap	z	P> z	Normal-based	
	Odds Ratio	Std. Err.			[95% Conf. Interval]	
height	.9576811	.0078642	-5.27	0.000	.9423908	.9732194
weight	1.054551	.0040065	13.98	0.000	1.046728	1.062433
age	1.05124	.0031789	16.53	0.000	1.045028	1.057489
female	.6362455	.0903066	-3.19	0.001	.4817348	.8403137
implead	1.014953	.008871	1.70	0.089	.9977144	1.03249

Note that in the first call, `myicereg` produces a single imputation of the missing lead concentrations, and the point estimates suffer from imputation variability. To rectify this issue, multiple imputations can be taken and the point estimates produced by averaging across imputations can be used. Alternatively, we can utilize bootstrap postestimation capabilities of `bs4rw`. The postestimation command `estat bootstrap`

can produce normal, bias corrected and percentile confidence intervals. The first two employ the original point estimates, and hence will suffer from their instability. The percentile method, however, works with the simulated bootstrap estimates only and does not rely on the original point estimates. Hence, it is more suitable for reporting purposes:

```
. estat bootstrap, eform percent
Logistic regression                Number of obs   =   10351
                                   Replications      =    240
```

highbp	Observed exp(b)	Bias	Bootstrap Std. Err.	[95% Conf. Interval]	
height	.95768109	-.0002745	.00786424	.941625	.9736155 (P)
weight	1.0545513	.000285	.00400651	1.047232	1.063002 (P)
age	1.0512398	.0000774	.00317886	1.044921	1.057708 (P)
female	.63624552	-.0058496	.09030655	.4812487	.8271627 (P)
implead	1.0149532	-.0032705	.00887099	.9920401	1.02652 (P)

(P) percentile confidence interval

Also, more accurate point estimates can be obtained as the means of bootstrap values $\hat{\theta}^{(*r)}$.

Let us now compare these results with the appropriate specification of weights (although we would still use the collapsed pseudo-strata to make sure the variances are being compared across the same design specifications):

```
. svyset upsu [pw=leadwt] , strata( cstrata )
      pweight: leadwt
      VCE: linearized
Single unit: missing
Strata 1: cstrata
SU 1: psu
FPC 1: <zero>
```

```
. svy : logistic highbp height weight age female lead
(running logistic on estimation sample)
```

Survey: Logistic regression

Number of strata	=	7	Number of obs	=	10191
Number of PSUs	=	62	Population size	=	112915016
			Design df	=	55
			F(5, 51)	=	70.60
			Prob > F	=	0.0000

highbp	Odds Ratio	Linearized Std. Err.	t	P> t	[95% Conf. Interval]	
height	.966196	.0080217	-4.14	0.000	.9502533	.9824063
weight	1.054808	.0045895	12.26	0.000	1.04565	1.064046
age	1.053083	.0040617	13.41	0.000	1.044975	1.061255
female	.7837347	.1105438	-1.73	0.090	.5907572	1.03975
lead	1.017569	.0089613	1.98	0.053	.9997674	1.035687

The estimation samples are slightly different. Standard errors are greater, and con-

fidence intervals are wider in the second specification for all parameters. It might be possible that the bootstrap procedure has not accounted enough for the imputation variability, or the imputation model has not provided enough variability in the imputed values. The two formulations lead to different conclusions. The bootstrap with imputed data rejects the hypothesis of no difference between males and females, while the re-weighted estimation results in weaker p -value of 9%. Both models give relatively weak support to the effect of lead on blood pressure, with the odds ratio of 1 at the edge of either CI.

The same model could have been estimated using multiple imputation package `ice` (Royston 2007). In multiple imputation, the missing values are imputed M times. For the k -th imputed complete data set, estimation is performed, and both resulting estimates $\hat{\theta}^{(k)}$ and the associated variance estimates $v^{(k)}[\hat{\theta}^{(k)}]$ are recorded. The point estimates are obtained as the average across imputations, $\hat{\theta}_{MI} = \sum_k \hat{\theta}^{(k)} / M$, and variances are estimated by

$$\begin{aligned} \hat{V}_{MI}[\hat{\theta}] &= W + \left(1 + \frac{1}{M}\right)B \\ &= \frac{1}{M} \sum_{k=1}^M v^{(k)}[\hat{\theta}^{(k)}] + \left(1 + \frac{1}{M}\right) \frac{1}{M-1} \sum_{k=1}^M (\hat{\theta}^{(k)} - \hat{\theta}_{MI})(\hat{\theta}^{(k)} - \hat{\theta}_{MI})', \end{aligned} \quad (22)$$

where W is the within-imputation component of variance, and B is the between-imputation component. However, if the imputation model uses the data across all strata, then the assumption of independence of observations across strata is no longer valid. The within component of (22) is biased down, and hence the whole estimator is biased down. The magnitude of this effect may be small if efficiency gains from stratification are small. Conservative variance estimates can be obtained by ignoring stratification. They will likely be acceptable in construction of confidence intervals, but their accuracy in small domain estimation or in design work is questionable, as such applications require consistency to the true design variance.

Example 7 is only intended to provide the typical syntax to implement variance estimation with missing data. For this procedure to be valid, the analyst must have access to the original probability weights. Post-stratification and non-response adjustments to the weights need to be performed at the calibration stage of the bootstrap replicate weight computation. Of course this assumption is not satisfied for NHANES data at hand where `finalwgt` variable already contains these adjustments. Also, imputation must be performed in each bootstrap replicate, which rules out the mean bootstrap where the replicate weights are averaged across several bootstrap replications.

5.2 Non-survey data

As a final example, let us consider the use of `bsweights` and `bs4rw` outside of the survey context where the two packages can be used to provide the first order balanced bootstrap simulations.

Example 8: balanced bootstrap for i.i.d. data

We start with all-time favorite automobile data set and generate a set of balanced bootstrap weights:

```
. sysuse auto, clear
(1978 Automobile Data)
. bsweights bw , nosvy rep(100) n(37) balanced seed(2083)
```

Note that the bootstrap sample size is set to 37, exactly the half of the data set size, and the number of replications is even, so that the total number of resampled units in all replicates is the multiple of the data set size, $n = 74$.

Let us first address bootstrap estimation for an unbiased statistic:

```
. bs4rw , rw(bw*) : mean price
(running mean on estimation sample)
BS4Rweights replications (100)
-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
      1   2   3   4   5
..... 50
..... 100
Mean estimation                Number of obs   =       74
                               Replications    =       100
```

	Observed Mean	Bootstrap Std. Err.	Normal-based [95% Conf. Interval]	
price	6165.257	298.1034	5580.985	6749.529

```
. estat bootstrap
Mean estimation                Number of obs   =       74
                               Replications    =       100
```

	Observed Mean	Bias	Bootstrap Std. Err.	[95% Conf. Interval]	
price	6165.2568	1.06e-06	298.10339	5570.475	6700.198 (BC)

(BC) bias-corrected confidence interval

```
. bstrap , rep(100) : mean price
(running mean on estimation sample)
Bootstrap replications (100)
-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
      1   2   3   4   5
..... 50
..... 100
Mean estimation                Number of obs   =       74
                               Replications    =       100
```

	Observed Mean	Bootstrap Std. Err.	Normal-based [95% Conf. Interval]	
price	6165.257	333.479	5511.65	6818.864

(Continued on next page)

```
. estat bootstrap
Mean estimation                Number of obs   =       74
                             Replications         =       100
```

	Observed Mean	Bias	Bootstrap Std. Err.	[95% Conf. Interval]	
price	6165.2568	-64.61972	333.47902	5507.595	6903 (BC)

(BC) bias-corrected confidence interval

```
. mean price
Mean estimation                Number of obs   =       74
```

	Mean	Std. Err.	[95% Conf. Interval]	
price	6165.257	342.8719	5481.914	6848.6

The regular bootstrap provided an estimate of bias that is non-negligible, while the estimate of bias coming from the balanced bootstrap is within the numeric accuracy of the float type.

What happens when the statistic is indeed biased in small samples? Let us consider the bootstrap estimation procedures for a ratio:

```
. bs4rw , rw(bw*) : ratio price / mpg
(running ratio on estimation sample)
BS4Rweights replications (100)
-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 1 | 2 | 3 | 4 | 5 |
..... 50
..... 100
Ratio estimation                Number of obs   =       74
                             Replications         =       100
```

```
_ratio_1: price/mpg
```

	Observed Ratio	Bootstrap Std. Err.	Normal-based [95% Conf. Interval]	
_ratio_1	289.4854	19.2086	251.8372	327.1336

```
. estat bootstrap
Ratio estimation                Number of obs   =       74
                             Replications         =       100
```

	Observed Ratio	Bias	Bootstrap Std. Err.	[95% Conf. Interval]	
_ratio_1	289.48541	.4471263	19.208605	245.2041	322.1176 (BC)

(BC) bias-corrected confidence interval

(Continued on next page)

```
. bstrap, rep(100) : ratio price / mpg
(running ratio on estimation sample)
Bootstrap replications (100)
-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 1 | 2 | 3 | 4 | 5 |
..... 50
..... 100
Ratio estimation          Number of obs   =    74
                          Replications    =   100
```

```
_ratio_1: price/mpg
```

	Observed Ratio	Bootstrap Std. Err.	Normal-based [95% Conf. Interval]	
_ratio_1	289.4854	21.9389	246.486	332.4849

```
. estat bootstrap
Ratio estimation          Number of obs   =    74
                          Replications    =   100
```

	Observed Ratio	Bias	Bootstrap Std. Err.	[95% Conf. Interval]	
_ratio_1	289.48541	-1.924227	21.9389	257.3014	342.1601 (BC)

(BC) bias-corrected confidence interval

```
. ratio price / mpg
Ratio estimation          Number of obs   =    74
                          Replications    =   100
_ratio_1: price/mpg
```

	Ratio	Linearized Std. Err.	[95% Conf. Interval]	
_ratio_1	289.4854	21.92466	245.7896	333.1812

Now, we see that a non-trivial amount of bias is reported by both methods. However, the estimate coming from the balanced bootstrap is much more trustworthy, as the simulation noise has been removed from it. An interested reader is encouraged to vary `seed(#)` and observe changes in the reported results for both balanced and unbalanced bootstraps.

6 Further remarks

Let us first revisit the scaling issues related to replication methods. First, if there are too few units resampled from stratum h , the weights of these units need to be increased so that the totals can be estimated without bias. Examples of the explicit expressions are given by the BRR and the jackknife replicate weights (10) and (15). Second, the scale of the deviations between $\hat{\theta}^{(r)}$ and $\hat{\theta}$ needs to be calibrated so that the resulting estimator $v_m[\hat{\theta}]$ from (7) matches a known estimator such as v_{str} . In BRR and the bootstraps

such as MMB or BWO, these deviations are on the correct scale, so the scaling factor is $A = 1$. On the other hand, these deviations are too small in the jackknife, Fay’s modification of BRR and the mean bootstrap, so these methods need to apply scaling factors $A > 1$ (and the jackknife needs the scaling to be performed within strata). The third scaling issue is that of internal scaling for the bootstrap procedures where samples are taken from small populations of size n_h . The differences between $\hat{\theta}^{(r)}$ and $\tilde{\theta}$ are on the wrong scale if $m_h \neq n_h - 1$, so modifications like (19) or (20) need to be taken.

The complex survey data features that `bsweights` can currently handle are stratification at the first stage of sample selection, cluster samples within strata and unequal sampling weights. The issues of sampling at higher stages and finite population corrections have been ignored. If the finite population corrections are known, they can be incorporated in the bootstrap replicate weights as

$$w_{hij}^{(*r)} = \left(1 - ((1 - f_h)m_h)^{1/2}(n_h - 1)^{-1/2} + ((1 - f_h)m_h)^{1/2}(n_h - 1)^{-1/2} \frac{n_h}{m_h} m_{hi}^{(*r)} \right) w_{hij}. \quad (23)$$

While `bsweights` provides functionality to create the bootstrap replicate weights on the spot, a better practice is to create a fixed set of weights and run different analyses using the same weights. That way, reproducibility of results between different runs and between different researchers is guaranteed.

Since designs with two PSUs per stratum are widely used in practice, BRR is the most popular replication variance estimation procedure. Many US data collection agencies, including National Center for Health Statistics (NCHS) and National Center for Educational Statistics (NCES), release public data files with BRR replicate weights. Examples include National Health and Nutrition Examination Survey (NHANES) and National Education Longitudinal Survey (NELS). Given the popularity and simplicity of BRR estimation, survey organizations often approximate their actual designs with stratified 2 PSUs/stratum designs and provide quasi-BRR weights. The modifications to an original design that could make it “BRR-able” include collapsing of strata, re-locating PSUs to similar strata, or merging PSUs in a stratum to obtain two groups of PSUs, so that grouped BRR can be applied to these groups. In some situations, this can cause problems: Shao (1996) gives an example when grouped BRR is inconsistent even for linear statistics.

While the US agencies tend to favor BRR estimation, Statistics Canada extensively uses the bootstrap procedures. Researchers in Canadian universities have access to Statistics Canada complex survey data through the network of Research Data Centers. The bootstrap procedures based on replicate weights are run on Statistics Canada servers to process researchers’ data analysis requests.

Difficulties may arise in replication variance estimation for domains. Since some units are removed when replicates are constructed, the number of available observations in the domain decreases. Some strata or even the complete replicate data set may be left with no observations in the domain, and estimation will result in missing parameter estimates. In such situations, Stata will print a red `e` or `x` instead of a dot in the `bs4rw`

output.

A similar issue may occur in logistic regression and some other limited dependent variable models where insufficient variability in the replicate data may lead to perfect prediction. In this case Stata drops the perfect predictor, and the estimation results become invalid for use by `bs4rw`.

A possible remedy for both problems is utilization of replication methods that lead to non-zero weights for all units, such as Fay's modification of BRR and the mean bootstrap. The latter however is not compatible with post-stratification and non-response adjustments.

Acknowledgements

All remaining errors, omissions and bugs are the author's responsibility.

7 References

- Binder, D. A. 1983. On the Variances of Asymptotically Normal Estimators from Complex Surveys. *International Statistical Review* 51: 279–292.
- Binder, D. A., and G. R. Roberts. 2003. Design-based and Model-based Methods for Estimating Model Parameters. In *Analysis of Survey Data*, ed. R. L. Chambers and C. J. Skinner, chap. 3. New York: John Wiley & Sons.
- Canty, A. J., A. C. Davison, D. V. Hinkley, and V. Ventura. 2006. Bootstrap diagnostics and remedies. *The Canadian Journal of Statistics* 34(1): 5–27.
- Chambers, R. L., and C. J. Skinner, ed. 2003. *Analysis of Survey Data*. Wiley series in survey methodology, New York: Wiley.
- Cochran, W. G. 1977. *Sampling Techniques*. 3rd ed. New York: John Wiley and Sons.
- Davison, A. C., D. V. Hinkley, and E. Schechtman. 1986. Efficient bootstrap simulation. *Biometrika* 73(3): 555–566.
- Efron, B., and R. J. Tibshirani. 1994. *An Introduction to the Bootstrap*. Chapman & Hall/CRC.
- Eltinge, J. 1996. Discussion of “Resampling Methods in Sample Surveys” by J. Shao. *Statistics* 27: 241–244.
- Gleason, J. R. 1988. Algorithms for Balanced Bootstrap Simulations. *The American Statistician* 42(4): 263–266.
- Graham, R. L., D. V. Hinkley, P. W. M. John, and S. Shi. 1990. Balanced Design of Bootstrap Simulations. *Journal of the Royal Statistical Society* 52(1): 185–202.

- Gupta, V. K., and A. K. Nigam. 1987. Mixed Orthogonal Arrays for Variance Estimation with Unequal Numbers of Primary Selections per Stratum. *Biometrika* 74(4): 735–742.
- Gurney, M., and R. S. Jewett. 1975. Constructing Orthogonal Replications for Variance Estimation. *Journal of the American Statistical Association* 70(352): 819–821.
- Judkins, D. R. 1990. Fay's Method for Variance Estimation. *Journal of Official Statistics* 6(3): 223–239.
- Kish, L. 1995. *Survey Sampling*. 3rd ed. New York: John Wiley and Sons.
- Korn, E. L., and B. I. Graubard. 1995. Analysis of Large Health Surveys: Accounting for the Sampling Design. *Journal of the Royal Statistical Society, Series A* 158(2): 263–295.
- Kovar, J. G., J. N. K. Rao, and C. F. J. Wu. 1988. Bootstrap and other methods to measure errors in survey estimates. *Canadian Journal of Statistics* 16: 25–45.
- Krewski, D., and J. N. K. Rao. 1981. Inference From Stratified Samples: Properties of the Linearization, Jackknife and Balanced Repeated Replication Methods. *The Annals of Statistics* 9(5): 1010–1019.
- Lehtonen, R., and E. Pahkinen. 2004. *Practical Methods for Design and Analysis of Complex Surveys*. 2nd ed. Statistics in Practice, New York: John Wiley & Sons.
- Lohr, S. L. 2009. *Sampling: Design and Analysis*. 2nd ed. Duxbury Press.
- McCarthy, P. J. 1969. Pseudo-Replication: Half Samples. *Review of the International Statistical Institute* 37(3): 239–264.
- McCarthy, P. J., and C. B. Snowden. 1985. The bootstrap and finite population sampling. In *Vital and Health Statistics*, 2–95. 85-1369, Washington, DC.
- Nigam, A. K., and J. N. K. Rao. 1996. On balanced bootstrap for stratified multistage samples. *Statistica Sinica* 6(1): 199–214.
- Phillips, O. 2004. Using bootstrap weights with WesVar and SUDAAN. Technical Report 2, Statistics Canada.
- Rao, J. N. K., and C. F. J. Wu. 1985. Inference From Stratified Samples: Second-Order Analysis of Three Methods for Nonlinear Statistics. *Journal of the American Statistical Association* 80(391): 620–630.
- . 1988. Resampling Inference With Complex Survey Data. *Journal of the American Statistical Association* 83(401): 231–241.
- Rao, J. N. K., C. F. J. Wu, and K. Yue. 1992. Some recent work on resampling methods for complex surveys. *Survey Methodology* 18(2): 209–217.

- Royston, P. 2007. Multiple imputation of missing values: further update of *ice*, with an emphasis on interval censoring. *Stata Journal* 7(4): 445–464.
- Rust, K. F., and J. N. Rao. 1996. Variance estimation for complex surveys using replication techniques. *Statistical Methods in Medical Research* 5(3): 283–310.
- Särndal, C.-E., B. Swensson, and J. Wretman. 1992. *Model Assisted Survey Sampling*. New York: Springer.
- Shao, J. 1996. Resampling Methods in Sample Surveys (with discussion). *Statistics* 27: 203–254.
- Shao, J., and R. R. Sitter. 1996. Bootstrap for Imputed Survey Data. *Journal of the American Statistical Association* 91(435): 1278–1288.
- Shao, J., and D. Tu. 1995. *The Jackknife and Bootstrap*. New York: Springer.
- Sitter, R. R. 1992a. Comparing Three Bootstrap Methods for Survey Data. *The Canadian Journal of Statistics* 20(2): 135–154.
- . 1992b. A Resampling Procedure for Complex Survey Data. *Journal of the American Statistical Association* 87(419): 755–765.
- . 1993. Balanced repeated replications based on orthogonal multi-arrays. *Biometrika* 80(1): 211–221.
- Skinner, C. J. 1989. Domain Means, Regression and Multivariate Analysis. In *Analysis of Complex Surveys*, ed. C. J. Skinner, D. Holt, and T. M. Smith, chap. 3, 59–88. New York: Wiley.
- Skinner, C. J., D. Holt, and T. M. Smith, ed. 1989. *Analysis of Complex Surveys*. New York: Wiley.
- Sloane, N. J. A. 2004. A Library of Hadamard Matrices. <http://research.att.com/njas/hadamard/>.
- Thompson, M. E. 1997. *Theory of Sample Surveys*, vol. 74 of *Monographs on Statistics and Applied Probability*. New York: Chapman & Hall/CRC.
- Valliant, R. 1996. Discussion of “Resampling Methods in Sample Surveys” by J. Shao. *Statistics* 27: 247–251.
- Wu, C. F. J. 1991. Balanced repeated replications based on mixed orthogonal arrays. *Biometrika* 78(1): 181–188.
- Yeo, D., H. Mantel, and T.-P. Liu. 1999. Bootstrap variance estimation for the National Population Health Survey. In *Proceedings of Survey Research Methods Section*, 778–785. The American Statistical Association.
- Yung, W. 1997. Variance Estimation for Public Use Files under Confidentiality Constraints. In *Proceedings of Statistics Canada Symposium*, 434–439. Statistics Canada.

About the author

Stanislav Kolenikov is an Assistant Professor at Department of Statistics, University of Missouri, Columbia, MO, USA. His research interests include statistical methods in social sciences, with focus on structural equation models, microeconometrics and analysis of complex survey data. The first version of Stata he worked with was Stata 5 in 1998.

Appendix: Commonly used notation

The generic datum x_{hij} denotes the measurement on variable y taken on the j -th observation in the i -th PSU in stratum h .

f	sampling fraction: $f = n/N$
f_h	sampling fraction in stratum h : $f_h = n_h/N_h$
$h = 1, \dots, L$	stratum index
$i = 1, \dots, n_h$	PSU index within strata
j	observation index within PSU
L	number of strata
m_h	bootstrap sample size; the number of PSUs taken from stratum h to form a bootstrap replicate
$m_{hi}^{(r)}$	bootstrap frequency; the number of times unit h, i is sampled in the r -th replicate
n	total sample size; in complex surveys, the total number of PSUs in the sample: $n = \sum_{h=1}^L n_h$
N	population size; in complex surveys, the total number of PSUs in the population: $N = \sum_{h=1}^L N_h$
n_h	sample size in stratum h ; in complex surveys, the number of PSUs taken from stratum h
N_h	population size; in complex surveys, the number of PSUs in stratum h
R	the number of replicates; the number of replicate weights for the mean bootstrap
$T[x]$	population total: $T[x] = \sum_h \sum_i \sum_j x_{hij}$
$t[x]$	estimate of the population total $T[x]$
$v[\hat{\theta}]$	estimator of variance $\mathbb{V}[\hat{\theta}]$
$v_m[\hat{\theta}]$	estimator of variance $\mathbb{V}[\hat{\theta}]$ obtained by method m ; the methods include linearization L , the jackknife J , BRR, or bootstrap schemes RBS, MMB, BWO and BWR
$\mathbb{V}[\hat{\theta}]$	(design) variance of the estimate $\hat{\theta}$ with respect to the sampling distribution
W_h	fraction of stratum h in population: $W_h = N_h/N$
w_{hij}	sampling weight of unit h, i, j
$w_{hij}^{(r)}$	replicate weight of unit h, i, j in the r -th replicate
θ	population parameter, such as total, mean, ratio, regression coefficient
$\hat{\theta}$	parameter estimate obtained from survey data
$\hat{\theta}^{(r)}$	parameter estimate obtained in the r -th replicate