The Stata Journal (2010)
10, Number 2, pp. 165–199

Resampling variance estimation for complex survey data

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

Abstract. In this article, I 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 the Stata `svy` suite. The bootstrap for complex survey data is implemented by the `bsweights` command. I describe this command and provide working examples.

Editors' note. This article was submitted and accepted before the new `svy bootstrap` prefix was made available in the Stata 11.1 update. The variance estimation method implemented in the new `svy bootstrap` prefix is equivalent to the one in `bs4rw`. The only real difference is syntax. For example,

```
. bs4rw, rw(bw*): logistic highbp height weight age female [pw=finalwgt]
```

is equivalent to

```
. svyset [pw=finalwgt], vce(bootstrap) bsrweight(bw*)
. svy: logistic highbp height weight age female
```

Similarly, the example using mean bootstrap replicate weights,

```
. local mean2fay = 1-sqrt(1/10)
. svyset [pw=finalwgt], vce(brr) brrweight(bw*) fay(`mean2fay´)
. svy: logistic highbp height weight age female
```

is equivalent to

```
. svyset [pw=finalwgt], vce(bootstrap) bsrweight(bw*) bsn(10)
. svy: logistic highbp height weight age female
```

The weights created by the `bsweights` command discussed in this article are equally applicable with the `bs4rw` command and with the new `vce(bootstrap)` and `bsrweight()` options of `svy` and `svyset`.

Keywords: st0187, bsweights, balanced repeated replication, balanced bootstrap, bootstrap, complex survey data, Hadamard matrix, half-samples, jackknife, resampling, weighted bootstrap, mean bootstrap

1 Complex survey data

Researchers who study large-scale health, behavioral, and economic processes often have to deal with datasets 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 nonoverlapping 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 the user to conduct straightforward statistical analysis within strata and implement different sampling techniques in different strata.

Cluster samples allow the user 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, but additional modifications of sampling weights are often performed. Poststratification uses the existing census or population information, such as the number of people in a group defined by age category, gender, and race. The weights are modified so that the weighted totals of poststratification variables match the population totals. To compute nonresponse adjustments, the weight is further increased by the inverse probability of response. This probability can be estimated 1) by the ratio of the weighted totals of the sampled and responded units within nonresponse adjustment cells or 2) by response propensity obtained from logistic regression of the response indicator on the available unit characteristics.

When simple random sampling is performed without replacement, there are efficiency gains summarized by finite population corrections: FPC $= 1 - f$, where $f = n/N$ is the sampling fraction, n is the sample size, and N is the population size. If units are selected with varying probabilities, joint selection probabilities need to be used to compute standard errors of (design-consistent) estimators. Hence, the effects of nonnegligible sampling fractions and associated efficiency gains, if any, are reflected in the joint selection probabilities. Because of their special relevance to simple random sampling designs only, and also because the population sizes are either not known or protected for privacy reasons, finite population corrections are rarely available in large-scale public-use datasets.

What happens if the complex survey design features are ignored in statistical analysis? If stratification 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, then 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 underrepresent the values of the random variables associated with low selection probabilities and overrepresent the values associated with high selection probabilities. As a result, unweighted statistics are biased for population parameters they estimate. The effects of clustering and unequal weights are detrimental for statistical inference and so analysts and researchers need to account for them.

For a very complex survey design, exact accounting for all its features is extremely cumbersome. At the data analysis stage, approximations are often made to yield a usable estimation formula. The most common approximate design is stratified two-stage sampling 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 was sampled more than once, then independent samples within this PSU are taken and marked in the dataset 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 the Second National Health and Nutrition Examination Survey (NHANES II) data, which is used throughout the *Survey Data Reference Manual* ([SVY]).

```
. use http://www.stata-press.com/data/r11/nhanes2
. svyset
      pweight: finalwgt
          VCE: linearized
  Single unit: missing
     Strata 1: strata
         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 dataset. By default, the variances of the parameter estimates will be computed using the 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 dataset, 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 example is also provided as example 2 in [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

------------------------------------------------------------------------------
             |             Linearized
      highbp | Odds Ratio   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      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 |   .7250087   .0641188    -3.64   0.001      .605353    .8683158
------------------------------------------------------------------------------
```

Compared with 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 is reported in the upper right block. The population size is estimated by the sum of weights. The column of standard errors has a heading indicating the type of standard errors used (linearized). Other components of the output are the same as in nonsurvey estimation. See [U] **20 Estimation and postestimation commands**.

To conclude this section, I will mention a few references on survey statistics. A popular introductory textbook is Lohr (2010). More formal treatment is given in classic monographs of Kish (1995) and Cochran (1977). The utmost level of mathematical detail can be found in Särndal, Swensson, and Wretman (1992) and Thompson (1997). Great intermediate literature with extensive conceptual explanations are Korn and Graubard (1995) and Lehtonen and Pahkinen (2004). Advanced topics are discussed in the collected volumes of Skinner, Holt, and Smith (1989) and Chambers and Skinner (2003). References that combine conceptual explanations with detailed software examples include Lumley (2010), which covers R, and Heeringa, West, and Berglund (2010), which covers Stata.

2 Variance estimation in complex surveys

Survey statistics has developed inference paradigms that are quite different from the mainstream "let the data be independent and identically distributed (i.i.d.) from a distribution, $f_\theta(x)$, characterized by a vector of parameters, θ." In its purest form of design-based inference, the units in the population, as well as their characteristics (measured variables), are assumed fixed. The randomness in the sample-based statistics (e.g., totals, ratios, means, regression coefficients) comes only from randomization performed at the sample selection stage. The design-based distributions are obtained by enumerating all samples possible under a given design scheme and associating the numeric values of the statistics of interest with the probabilities of the samples they are based on. Other paradigms, such as model-based (Binder and Roberts 2003) and model-assisted (Särndal, Swensson, and Wretman 1992) inference, work with estimators that have good properties under a strict design-based paradigm and under a working model assumed by the survey statistician.

In this article, I discuss estimation of the design variances, i.e., the variances of the distributions generated by repeated sampling using a given design. In many practical situations, the design-based variance estimates are doubly robust: they are consistent when either the design is correctly described (typically, with replacement at the first stage) or the model is approximately correct (e.g., it is a linear or generalized linear model with a correctly specified mean structure). They are also robust to different specifications of the variance parameters in a model.

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, and then turn to resampling methods in the next section.

Consider the following estimate of the total for stratified samples:

$$t_{\mathrm{str}}(x) = \sum_{h=1}^{L} \sum_{i=1}^{n_h} t_{hi}$$

where

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

is the estimate of the total based on the ith PSU in the hth stratum, and other symbols are defined in the appendix. In the S2SWR design, the variance of $t_{\mathrm{str}}(x)$ can be directly estimated by

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

This is analogous to (1) in [SVY] **variance estimation**, except that finite population corrections are ignored because 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, correlations, 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 let $\widehat{\theta} = f(t_1, \dots, t_K)$ 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

$$v_L\left(\widehat{\theta}\right) \approx \widehat{\mathrm{MSE}}\left(\widehat{\theta}\right) \approx v\left\{\left(\sum_k \frac{\partial f}{\partial \theta_k}\Big|_{\theta_k=T_k}\right) t_k\right\} \approx v\left\{\left(\sum_k \frac{\partial f}{\partial \theta_k}\Big|_{\theta_k=t_k}\right) t_k\right\}$$

Here the first approximation is to ignore a possible bias of $\widehat{\theta}$; the second approximation is the linearization itself; and the third approximation is to evaluate the necessary derivatives at the sample values $(t_1, \ldots, t_K)$. An important special case is the sample mean

$$\overline{x}_r = \frac{\sum_{hij} w_{hij} x_{hij}}{\sum_{hij} w_{hij}}$$

Subindex r stands for ratio estimator: $\overline{x}_r$ is the ratio $t(x)/t(1)$. Its variance is estimated by

$$v_r(\overline{x}_r) = \sum_h \frac{n_h}{n_h - 1} \sum_i (d_{hi} - \overline{d}_h)^2 \tag{2}$$

where

$$d_{hi} = \frac{1}{N_h} \sum_j w_{hij}(x_{hi} - \overline{x}_r), \quad \overline{d}_h = \frac{1}{n_h} \sum_i d_{hi}$$

The linearization method is also applicable when $\widehat{\theta}$ is derived as a solution to a set of estimating equations

$$t\left\{\psi(x, \theta)\right\} = \sum_{hij} w_{hij} \psi(x_{hij}, \theta) = 0 \tag{3}$$

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

$$v_L\left(\widehat{\theta}\right) \approx \left[\nabla_\theta t\left\{\psi(x, \theta)\right\}\right]^{-1} v\left[t\left\{\psi(x, \theta)\right\}\right] \left[\nabla_\theta t\left\{\psi(x, \theta)\right\}\right]^{-1} \tag{4}$$

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

Variance estimation based on linearization is the default estimation method for survey data in Stata. The required derivatives are computed analytically for several commonly used models listed in [SVY] **svy estimation** and are computed numerically for other models using `_robust`. The use of the same mechanics for both survey inference and more traditional robust variance estimation, leading to the Huber–White (Huber 1967; White 1982) sandwich variance estimator, is not surprising. In fact, this is the reflection of the aforementioned double robustness property of the design variances and their estimators.

For parameters and statistics that are not smooth functions of the underlying distributions—such as quantiles, extreme order statistics, or certain poverty and income inequality indicators—the linearization variance estimator is not applicable.

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 (1) or (2) requires that stratum and PSU identifiers h and i are known for each observation. In (4), stratum and PSU identifiers are needed to compute $v\{t(\psi)\}$, which is done by (1) or (2). 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 (Rust and Rao 1996; Shao 1996, 2003) are balanced repeated replication (BRR), the jackknife, and the bootstrap. In each of these methods, multiple replicates of the original data are created. In the rth replicate, some PSUs are omitted (i.e., all sample elements in PSU$_{hi}$ are removed) and some are retained (and may be included multiple times, as in the bootstrap). The parameter estimate of interest, $\widehat{\theta}_m^{(r)}$, is computed using the same estimation procedure as that for the original data. Subindex m stands for the estimation method. The resulting estimator of variance is generally defined by

$$v_m\left(\widehat{\theta}\right) = \frac{A}{R}\sum_{r=1}^{R}\left\{\widehat{\theta}_m^{(r)} - \widetilde{\theta}\right\}^2 \tag{5}$$

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 (1). The measure of the central tendency $\widetilde{\theta}$ can be the mean of resampled values

$$\widetilde{\theta} = \frac{1}{R}\sum_{r=1}^{R}\widehat{\theta}_m^{(r)} \tag{6}$$

resulting in the variance version of the estimator or can be the original estimate

$$\widetilde{\theta} = \widehat{\theta} \tag{7}$$

resulting in the mean squared error (MSE) version of the estimator.

In most cases, $v_m\left(\widehat{\theta}\right)/v_L\left(\widehat{\theta}\right) \to 1$ in probability as $n = \sum_h n_h \to \infty$. 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, $w^{(r)}_{hij}$, $r = 1, \dots, R$, is created and distributed with the public release dataset. 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 (5).

For the methods below, I describe the mechanics, the implied replicate weights, and the number of replicates each method requires. I compare the properties of variance estimation procedures in section 3.4.

3.1 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, and the other one is retained and replicated twice to ensure that the totals are on the right scale. Because exactly half of the PSUs are used, the replicates are also referred to as half-samples. The replicate weights are

$$w^{(r)}_{hij} = \begin{cases} 2w_{hij}, & \text{PSU}_{hi} \text{ is retained} \\ 0, & \text{PSU}_{hi} \text{ is omitted} \end{cases} \tag{8}$$

These weights are used to compute $\widehat{\theta}^{(r)}_{\text{BRR}}$. The BRR variance estimator is obtained from (5) with $A = 1$:

$$v_{\text{BRR}}\left(\widehat{\theta}\right) = \frac{1}{R}\sum_{r=1}^{R}\left\{\widehat{\theta}^{(r)}_{\text{BRR}} - \widehat{\theta}\right\}^2 \tag{9}$$

The number of all possible half-samples is 2^L. If all half-samples are used, $v_{\text{BRR}} = v_L = v_r$ in the linear case. McCarthy (1969) showed that this equality holds with a much smaller number of replicates, $L \le R \le L + 3$, 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 (Hedayat, Sloane, and Stufken 1999), which are square matrices with entries ±1 and rows that are mutually orthogonal. It has been conjectured that a Hadamard matrix of order $4k$ exists for every positive integer k. Several matrices are 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:

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

There are several modifications of the BRR method. For a given pattern of included and excluded PSUs in the rth 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 $\widehat{\theta}_{\mathrm{BRR}}^{(rc)}$. Then additional variance estimates are

$$v_{\mathrm{BRR2}}\left(\widehat{\theta}\right) \equiv v_{\mathrm{BRR\text{-}D}}\left(\widehat{\theta}\right) = \frac{1}{4R}\sum_{r=1}^{R}\left\{\widehat{\theta}_{\mathrm{BRR}}^{(r)} - \widehat{\theta}_{\mathrm{BRR}}^{(rc)}\right\}^2$$

and

$$v_{\mathrm{BRR3}}\left(\widehat{\theta}\right) \equiv v_{\mathrm{BRR\text{-}S}}\left(\widehat{\theta}\right) = \frac{1}{2R}\sum_{r=1}^{R}\left\{\widehat{\theta}_{\mathrm{BRR}}^{(r)} - \widetilde{\theta}\right\}^2 + \left\{\widehat{\theta}_{\mathrm{BRR}}^{(rc)} - \widetilde{\theta}\right\}^2$$

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 $0 \leq k < 1$:

$$w_{hij}^{(r)} = \begin{cases} (2-k) w_{hij}, & \mathrm{PSU}_{hi} \text{ is retained} \\ k w_{hij}, & \mathrm{PSU}_{hi} \text{ is omitted} \end{cases}$$

The value of $k = 0$ gives the original BRR procedure. The correct scaling factor is $A = 1/(1-k)^2$. If Fay BRR weights are supplied in a public-release data file, the value of k (or A) must be provided to the data user.

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

$$\widehat{\theta}_{\mathrm{BRR}} = 2\widehat{\theta} - \frac{1}{R}\sum_{r}\widehat{\theta}^{(r)}$$

or, if complementary half-samples are used,

$$\widehat{\theta}_{\mathrm{BRRc}} = 2\widehat{\theta} - \frac{1}{2R}\sum_{r}\left\{\widehat{\theta}^{(r)} + \widehat{\theta}^{(rc)}\right\}$$

Stata implements the original v_{BRR} (9) with `svy brr`. By default, the variance formulation (6) is used, and the MSE formulation (7) may be requested with `svy brr, mse`. Fay's modification is available with the `fay(#)` option. The BRR replicate weights may be specified via `svyset, brrweight(`*varlist*`)`. If no BRR replicate weights are given, the user must provide a Hadamard matrix with `hadamard(`*matrix*`)`.

An example of a dataset with replicate weights is provided as example 1 of [SVY] **svy brr**:

```
. use http://www.stata-press.com/data/r11/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

──────────────┬──────────────────────────────────────────────────────────────
              │                 BRR
       highbp │ 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 │   .7250087   .0650444    -3.58   0.001     .6037789    .8705797
──────────────┴──────────────────────────────────────────────────────────────
```

Most of the design information has been stripped from the 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 (Gurney and Jewett 1975; Gupta and Nigam 1987; Wu 1991; Sitter 1993) but did not find widespread use in practice because of excessive mathematical complexity and limited availability of the orthogonal arrays necessary to construct the BRR schemes.

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 an i.i.d. sample of size n, the rth replicate is obtained by removing the rth observation, and hence the number of replicates is $R = n$. The appropriate scaling factor in (5) is $A = n - 1$.

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

$$w_{hij}^{(gk)} = \begin{cases} 0, & h = g, i = k \\ \frac{n_g}{n_g - 1} w_{hij}, & h = g, i \neq k \\ w_{hij}, & h \neq g \end{cases} \tag{10}$$

The jackknife variance estimators can be defined in a number of ways. Let $\widehat{\theta}^{(hi)}$ be the estimate obtained with unit h, i removed. Then two jackknife variance estimators are defined as follows:

$$v_{J1} = \sum_h \frac{n_h - 1}{n_h} \sum_i \left\{ \widehat{\theta}^{(hi)} - \widehat{\theta}^h \right\}^2$$

$$v_{J2} = \sum_h \frac{n_h - 1}{n_h} \sum_i \left\{ \widehat{\theta}^{(hi)} - \widehat{\theta} \right\}^2$$

The scaling factor needs to be applied within each stratum to produce correct totals and consistent variance estimates. Rao and Wu (1985) provide four additional jackknife estimators that have virtually the same properties and are rarely used in practice.

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

$$\widehat{\theta}_J = (n + 1 - L)\widehat{\theta} - \sum_h (n_h - 1)\widehat{\theta}^h$$

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 dataset. The default estimator is v_{J1}, and the `mse` option invokes estimator v_{J2}.

(*Continued on next page*)

```
. use http://www.stata-press.com/data/r11/nhanes2

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

Jackknife replications (62)
────┼─── 1 ───┼─── 2 ───┼─── 3 ───┼─── 4 ───┼─── 5
..................................................    50
............

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

──────────────┬────────────────────────────────────────────────────────────────
              │              Jackknife
       highbp │ Odds Ratio   Std. Err.      t    P>|t|     [95% Conf. Interval]
──────────────┼────────────────────────────────────────────────────────────────
       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 │   .7250087   .0641366    -3.64   0.001     .6053228    .8683592
──────────────┴────────────────────────────────────────────────────────────────
```

Like linearization, the jackknife is inconsistent for nonsmooth statistics. This problem can be ameliorated in designs with a large number of PSUs per stratum. For a carefully chosen $k > 1$, delete-k jackknife (Shao and Tu 1995; Wolter 2007) removes k PSUs from the same stratum to form the jackknife replicates. Consistency of the method is a complicated interplay between smoothness of the statistics of interest and parameter k. The complete delete-k jackknife requires $R = \sum_h \binom{n_h}{k} \sim O\left\{(n/L)^k\right\}$ replications, notably increasing the computational burden.

Computational burden can also be excessive when the number of PSUs in the complex survey design is several hundreds or even thousands. Some of the list-based establishment surveys may have such a structure. To reduce the number of replications, PSUs are randomly grouped, with groups acting as quasi-PSUs in the new design. The method is known as the random-groups method (Wolter 2007) or delete-a-group jackknife (Kott 2001). The method has some potential pitfalls, including inconsistency of the variance estimates (Shao 1996) and odd dependencies on the number of groups used (Valliant, Brick, and Dever 2008).

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. A typical choice is the empirical distribution of the data.

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

The plug-in principle of the bootstrap, illustrated in figure 1 on the next page, states that relation of the bootstrap values $\widehat{\theta}_m^*$ to $\widehat{\theta}_n$ is approximately the same as that of $\widehat{\theta}_n$ 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 $\widehat{\theta}_n$ are

$$\begin{aligned}
\text{Bias}\left(\widehat{\theta}_n\right) &= E\left(\widehat{\theta}_n - \theta\right) \doteq E^*\left(\widehat{\theta}_n^* - \widehat{\theta}_n\right) \\
V\left(\widehat{\theta}_n\right) &= E\left[\left\{\widehat{\theta}_n - E\left(\widehat{\theta}_n\right)\right\}^2\right] \doteq E^*\left[\left\{\widehat{\theta}_n^* - E\left(\widehat{\theta}_n^*\right)\right\}^2\right] \\
\text{MSE}\left(\widehat{\theta}_n\right) &= E\left\{\left(\widehat{\theta}_n - \theta\right)^2\right\} \doteq E^*\left\{\left(\widehat{\theta}_n^* - \theta_n\right)^2\right\} \\
\text{cdf}_{\theta_n}(x) &= \text{Prob}\left(\widehat{\theta}_n - \theta < x\right) \doteq \text{Prob}^*\left(\widehat{\theta}_n^* - \widehat{\theta}_n < x\right) \qquad (11)
\end{aligned}$$

where the starred quantities are taken with respect to the bootstrap distribution. The particular strength of the bootstrap is the last equation of (11). 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 1993; 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. In practice, instead of the complete bootstrap, a large number R of random samples with replacement from the original data is 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 simulation error is small (Efron and Tibshirani 1993, 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 1,000.

$$\begin{tikzcd}
F \arrow[r, "\text{sample}"] \arrow[d, "T"] & F_n \arrow[r, "\text{bootstrap}"] \arrow[d, "T"] & (F_n^*) \arrow[r, "\text{simulate}"] \arrow[d, "T"] & F_n^{(*b)} \arrow[d, "T"] \\
\theta \arrow[r, leftrightarrow, "\text{inference}"] & \widehat{\theta}_n \arrow[r, leftrightarrow, "\text{bootstrap}"] & \left(\widehat{\theta}_n^*\right) \arrow[r, phantom, "\approx"] & \widehat{\theta}_n^{(*b)}
\end{tikzcd}$$

Figure 1. The bootstrap principle

Special bootstrap schemes exist to control the error induced by simulation. The package `bsweights` (section 4) implements one such scheme, 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, Hinkley, and Schechtman 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 every problem in statistical inference. Examples where the bootstrap fails are abundant (Canty et al. 2006; Shao and Tu 1995, sec. 3.6). Complex survey data is one such example. More sophisticated resampling schemes 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 scheme (NBS). To construct the rth 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, $\widehat{\theta}^{(*r)}$; repeat R times; and estimate the variance using (5). Rao and Wu (1988) demonstrated that even in the simple case of the stratified mean (2), the variance of the complete NBS distribution is

$$V^*_{\mathrm{NBS}}(\overline{x}^*) = \sum_h \frac{W_h^2}{n_h} \frac{n_h - 1}{n_h} s_h^2$$

instead of

$$v_r(\overline{x}) = \sum_h \frac{W_h^2}{n_h} s_h^2$$

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 lesser importance because the bias disappears as $n \to \infty$.

To rectify the bias of the NBS estimate, 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 pseudovalues

$$\widetilde{x}_{hi}^{(r)} = \overline{x}_h + m_h^{1/2}(n_h - 1)^{-1/2} \left\{ x_{hi}^{(*r)} - \overline{x}_h \right\} \qquad (12)$$

are used in computing the variances of the moments and their functions. Here $x_{hi}^{(*r)}, i = 1, \ldots, m_h$, is the rth sample taken from the hth stratum, and $\overline{x}_h = \sum_i x_{hi}/n_h$ is the estimated stratum mean. Rao, Wu, and Yue (1992) extended the RBS method to cover the case of estimating equations (3). They considered replicate weights implied by RBS and demonstrated that the necessary internal scaling can be achieved by the replicate weights

$$w_{hij}^{(*r)} = \left\{ 1 - m_h^{1/2}(n_h-1)^{-1/2} + m_h^{1/2}(n_h-1)^{-1/2}\frac{n_h}{m_h} m_{hi}^{(*r)} \right\} w_{hij} \tag{13}$$

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

How should the bootstrap sample size m_h be chosen? First, note that internal scaling is not needed if $m_h = n_h - 1$. McCarthy and Snowden (1985) proposed this choice under the name of the bootstrap with replacement (BWR). 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 matching of the third-order moments of the bootstrap distribution with those of the theoretical sampling distribution. Simulation evidence indicated that bootstrap estimators with $m_h = n_h - 3$ are unstable for moderate sample sizes $n_h = 5$ and unknown strata variances (Kovar, Rao, and Wu 1988). Finally, note that when PSU$_{hi}$ is omitted from the rth replicate, the replicate weight is $w_{hij}^{(*r)} = \left\{ 1 - m_h^{1/2}(n_h-1)^{-1/2} \right\} w_{hij}$. 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).

To increase the number of replicates, and hence the stability of the estimates, the mean bootstrap (Yung 1997; Yeo, Mantel, and Liu 1999) can be used. To compute the rth replicate weight variable, the bootstrap frequencies are averaged across a series of K subsequent replicates, and the average frequency

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

is used instead of $m_{hi}^{(*r)}$ to scale weights according to (13). The total number of bootstrap replications in this method is the product RK, while the number of replicate weight variables is R. The scaling factor A in (5) 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.

3.4 Comparison of estimators and relations between them

The linearization, jackknife, bootstrap, and (where applicable) BRR variance estimators are estimating the same quantity, the variance $V\left(\widehat{\theta}\right) = \sigma^2$ of statistic $\widehat{\theta}$. 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: $v_L = v_J = v_{\text{BRR}} = v_{\text{BOOT}}$, covering all versions of the jackknife and BRR estimators, and rescaling (but not naïve) bootstrap estimators.

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 a bounded number of PSUs per stratum and number of strata $L \to \infty$. 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 $E\left\{(v_m - \sigma^2)^2\right\}$, the ordering was reversed.

Rao and Wu (1985, 1988) demonstrated that different jackknife and BRR estimators are very close to one another and that the jackknife is closest to the linearization estimator, followed by the bootstrap and BRR estimators. Also, v_{BRR1} tends to be farther from v_L than v_{BRR2} or v_{BRR3}. There is no preferred estimator in terms of bias: 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, because the linearization estimator v_L is usually considered the "golden standard" (if it is applicable for a given estimation problem). Valliant (1996) also discussed several situations in which the jackknife estimator exhibited better model-based properties than did the linearization estimator in ratio estimation and in poststratification.

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 balance 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. Nigam and Rao (1996) proposed the second-order balanced bootstrap schemes for more general designs with $m_h = n_h$ equal to an even number or a prime power.

Overall, the jackknife and linearization methods tend to exhibit similar performance. They are more stable for smooth functions but inconsistent for nonsmooth functions. The method that is applicable for all statistics is the bootstrap (and its kin, BRR, for designs with $n_h = 2$). Additionally, the bootstrap can 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 lesser stability and longer confidence intervals.

4 The bsweights command

4.1 Syntax

`bsweights` *prefix*`,` `reps(#)` `n(#)` [`replace` `average(#)` `balanced` `dots` `calibrate(`*call_to_weight_calibration_routine*`)` `verbose` `nosvy` `seed(#)` `float` `double`]

The *call_to_weight_calibration_routine* is

[`do`] *calibration_program* [*arg1*] `@` [*arg2*]

4.2 Options

`reps(#)` specifies the number of bootstrap replications to be taken and the number of weight variables to be generated. `reps()` is required.

`n(#)` specifies how the number of PSUs m_h per stratum be handled. If a positive number is specified, it is interpreted as the number of units per stratum m_h that will be used. If a nonpositive number is specified, then $m_h = n_h - |\#|$. Specifying `n(0)` leads to the number of units resampled being equal to the original number of units n_h. `n()` is required.

`replace` requests that the weight variables 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, you should specify the `vfactor()` option of `bs4rw`. The total number of replications is the product of `reps()` and `average()`.

`balanced` requests the balanced bootstrap (Graham et al. 1990). See *Remarks* below.

`dots` provides additional output.

`calibrate(`*call_to_weight_calibration_routine*`)` allows a call to another program to adjust the weights for poststratification and nonresponse. See *Remarks* below.

`verbose` provides output from the weights calibration commands.

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

`seed(#)` specifies the seed for the random-number generator. See [D] **generate**.

`float` specifies that the weight variables have float type. See [D] **data types**.

`double` specifies that the weight variables have double type. See [D] **data types**.

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 the symbol `@`. For instance, if you specify

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

then `bsweights` will issue the 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 and 6 in section 5.1.

The weight adjustments are taking place after the internal scaling (13). It is the user's responsibility to provide correct treatment of the resampling weights in their calibration procedures. The `verbose` option provides output from calibration commands for debugging purposes.

For proper results, you must specify as inputs to `bsweights` the original probability weights rather than the final sampling weights. 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 with 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.

Using the bootstrap weights

There are two ways to use the bootstrap weights generated by `bsweights`. In the examples below, I use the `bs4rw` command 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`, type `findit bs4rw` and follow the instructions. Here are 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 part of its syntax. This rules out many important commands, such as `correlate` and `xtmixed`.

The first call `bs4rw` makes is to find the point estimates. Therefore, the command specification must contain the original weights; 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 bootstrap and BRR use the same replication variance estimation formula (5) with the same scaling factor A, you can trick Stata (or any other software that performs BRR estimation) into accepting the bootstrap weights as the BRR weights; see example 4 in section 5.1.

4.4 Saved results

Scalars

`r(balanced)`	`1` if the first-order balance was achieved, `0` otherwise

5 Examples

Here are several examples (using Stata example datasets) of how `bsweights` and `bs4rw` can be used.

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 pseudodesign with seven pseudostrata. 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.

```
. use http://www.stata-press.com/data/r11/nhanes2

. generate 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

------------------------------------------------------------------------------
             |             Linearized
      highbp | Odds Ratio   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      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 |   .7250087    .073301    -3.18   0.002      .592036    .8878474
------------------------------------------------------------------------------
```

▷ Example 1: BWR scheme

In the first example, we shall create bootstrap weights with arbitrarily chosen $R = 100$ replications and the bootstrap sample size $m_h = n_h - 1$ most commonly used in practice:

```
. bsweights bw, reps(100) n(-1) seed(10101) dots
Running bsample 100  times ....................................................
> ................................................
Rescaling weights
...............................................................................
> .....................

Warning: the first-order balance was not achieved
```

```
. bs4rw, rweights(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)       =    982.09
                                                Prob > chi2        =    0.0000
Log pseudolikelihood = -2961.5987               Pseudo R2          =    0.1527

──────────────────────────────────────────────────────────────────────────────
             │   Observed   Bootstrap                         Normal-based
      highbp │ Odds Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
─────────────┼────────────────────────────────────────────────────────────────
      height │   .9688567   .0065946    -4.65   0.000      .9560174    .9818685
      weight │   1.052489   .0035124    15.33   0.000      1.045627    1.059395
         age │   1.050473   .0023775    21.76   0.000      1.045824    1.055143
      female │   .7250086   .0780813    -2.99   0.003      .5870448    .8953958
──────────────────────────────────────────────────────────────────────────────
```

The standard errors are within 10% of the linearization-based ones, and the inference conclusions regarding significant variables are unchanged. The `dots` option provides additional output, including the warning about the lack of balance.

◁

▷ Example 2: 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 or changed options are underlined):

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

(*Continued on next page*)

```
. bs4rw, rweights(bw*): logistic highbp height weight age female [pw=finalwgt]
(running logistic on estimation sample)

BS4Rweights replications (80)
───┼─── 1 ───┼─── 2 ───┼─── 3 ───┼─── 4 ───┼─── 5
..................................................    50
..............................

Logistic regression                               Number of obs      =     10351
                                                  Replications       =        80
                                                  Wald chi2(4)       =    786.21
                                                  Prob > chi2        =    0.0000
Log pseudolikelihood = -2961.5987                 Pseudo R2          =    0.1527

------------------------------------------------------------------------------
             |   Observed   Bootstrap                         Normal-based
      highbp | Odds Ratio   Std. Err.      z    P>|z|     [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
------------------------------------------------------------------------------
```

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, because only 3 out of 16 PSUs are resampled from the largest stratum. Better schemes would have fewer 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 the 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 the computational burden is to use the mean bootstrap. It creates R replicate weight variables from RK bootstrap replicates. The number of replicates that is averaged over by the mean bootstrap to create one replicate weight, K, must be carried over to `bs4rw` with the `vfactor(`K`)` option.

```
. bsweights bw, reps(120) average(10) n(-1) seed(10101) balanced dots replace
Balancing within strata:
.......
Rescaling weights
...............................................................................
> .........................................
Warning: the first-order balance was not achieved
```

```
. bs4rw, rweights(bw*) vfactor(10): logistic highbp height weight age female
> [pw=finalwgt]
(running logistic on estimation sample)

BS4Rweights replications (120)
────┼─── 1 ───┼─── 2 ───┼─── 3 ───┼─── 4 ───┼─── 5
..................................................    50
..................................................   100
....................

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

─────────────┬────────────────────────────────────────────────────────────────
             │   Observed   Bootstrap                         Normal-based
      highbp │ Odds Ratio   Std. Err.      z    P>|z|     [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      1.045677    1.055291
      female │   .7250086   .0725368    -3.21   0.001      .5959102    .8820749
─────────────┴────────────────────────────────────────────────────────────────
```

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

◁

▷ Example 4: 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 (Phillips 2004). The weight variables will be provided to `svyset` with the `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
  (output omitted)
               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>
```

```
. svy 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)    =    174.87
                                                Prob > F           =    0.0000

──────────────────────────────────────────────────────────────────────────────
             │                 BRR
      highbp │ Odds Ratio   Std. Err.      t    P>|t|     [95% Conf. Interval]
─────────────┼────────────────────────────────────────────────────────────────
      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     1.045648     1.05532
      female │   .7250087   .0722339    -3.23   0.002     .5952031    .8831231
──────────────────────────────────────────────────────────────────────────────
```

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

──────────────────────────────────────────────────────────
             │                 BRR
      highbp │      Coef.   Std. Err.        DEFF      DEFT
─────────────┼────────────────────────────────────────────
      height │  -.0316386   .0062328       1.2491   1.11763
      weight │   .0511574   .0032142       1.8184   1.34848
         age │   .0492406   .0023248      1.03103   1.01539
      female │  -.3215716   .0996318      .972727   .986269
       _cons │  -2.858968   1.085424      1.27086   1.12732
──────────────────────────────────────────────────────────
```

However, because `svy 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 use this 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 because it relies on asymptotic normality of the estimates. Thus the analysis in example 3 contained χ^2 instead of the F test, and the confidence intervals were based on the normal distribution and hence slightly shorter.

If the usual RBS rather than the mean bootstrap is used, Fay's correction can be omitted.

◁

▷ Example 5: Calibration

As discussed by Shao (1996), if the original probability weights were modified to account for poststratification and nonresponse, the bootstrap procedure must take the original probability weights, process the bootstrap frequencies $m_{hi}^{(*r)}$ according to (13), and then apply the same adjustments that were used with the original weights. **bsweights** allows the performance of weight adjustments using the **calibrate()** option.

Suppose that we want to calibrate the bootstrap weights so that the sums of weights for each gender are equal to the original sums of weights in the NHANES II data. Before we run **bsweights**, we need to store these sums:

```
. svyset upsu [pw=finalwgt], strata(cstrata)
. generate byte ones = 1
. quietly svy: total ones, over(sex)
. matrix Totals = e(b)
```

Next let us define the weight calibration program. It will compute the current sum of weights and scale it to match the existing gender totals.

```
. capture program drop CalGender
. program CalGender
  1.    args wvar
  2.    total ones [pw=`wvar´], over(sex)
  3.    replace `wvar´ = `wvar´*Totals[1,1]/_b[ones:Male] if sex==1
  4.    replace `wvar´ = `wvar´*Totals[1,2]/_b[ones:Female] if sex==2
  5. end
```

We now can create our replicate weights and run the estimation procedure:

```
. bsweights bw, n(-2) reps(120) dots balanced calibrate(CalGender @) seed(10101)
> replace
Balancing within strata:
.......
Rescaling weights
...............................................................................
> .........................................
```

(*Continued on next page*)

```
. bs4rw, rweights(bw*): logistic highbp height weight age female [pw=finalwgt]
(running logistic on estimation sample)

BS4Rweights replications (120)
────┼─── 1 ───┼─── 2 ───┼─── 3 ───┼─── 4 ───┼─── 5
..................................................    50
..................................................   100
....................

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

─────────────┬────────────────────────────────────────────────────────────────
             │   Observed   Bootstrap                         Normal-based
      highbp │ Odds Ratio   Std. Err.      z    P>|z|     [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
─────────────┴────────────────────────────────────────────────────────────────
```

Each dot under `Rescaling weights` includes both the internal scaling (13) 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 (West, Berglund, and Heeringa 2008) is a special case of calibration where the weights outside the domain are set to zero. Suppose that we want to conduct a separate analysis for females only, reproducing the subpopulation estimation of example 2 of [SVY] **svy estimation**. The variable `female` takes on the value 0 for males and 1 for females, so the analogue to using the `subpop(female)` option of `svy` will be achieved through the following calibration program:

```
. capture program drop CalSubpop

. program CalSubpop
  1.     args wvar
  2.     replace `wvar´ = `wvar´ * female
  3. end
```

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

```
. capture 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
..............................................................................
> ..........................................
Warning: the first-order balance was not achieved
. bs4rw, rweights(bw*) vfactor(10): logistic highbp height weight age
> [pw=finalwgt*female]
(running logistic on estimation sample)
BS4Rweights replications (120)
───┼─── 1 ───┼─── 2 ───┼─── 3 ───┼─── 4 ───┼─── 5
..................................................    50
..................................................   100
....................
Logistic regression                             Number of obs      =     10351
                                                Replications       =       120
                                                Wald chi2(3)       =    514.79
                                                Prob > chi2        =    0.0000
Log pseudolikelihood = -1359.1291               Pseudo R2          =    0.1703

------------------------------------------------------------------------------
             |   Observed   Bootstrap                         Normal-based
      highbp | Odds Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      height |   .9765379   .0096453    -2.40   0.016     .9578152    .9956266
      weight |   1.047845   .0039144    12.51   0.000     1.040201    1.055545
         age |   1.058105   .0034868    17.14   0.000     1.051293    1.064961
------------------------------------------------------------------------------
```

`bsweights` issued a warning that calibration and the mean bootstrap are generally incompatible. Note how the original `pweight` was modified in the call to `bs4rw`. Instead of `[pw=finalwgt]`, which would be appropriate for estimation based on the full sample, the units outside the domain were zeroed out by `[pw=finalwgt*female]`. Without this modification, `bs4rw` would report the point estimates for the complete sample rather than the subpopulation only.

◁

5.2 Nonsurvey data

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

▷ Example 7: Balanced bootstrap for i.i.d. data

We start with the all-time favorite `auto.dta` and generate a set of balanced bootstrap weights:

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

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

Let us first address bootstrap estimation for an unbiased statistic:

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

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

. quietly bs4rw, rweights(bw*): mean price
. estat bootstrap
Mean estimation                                 Number of obs      =        74
                                                Replications       =       100

------------------------------------------------------------------------------
             |    Observed               Bootstrap
             |        Mean       Bias    Std. Err.  [95% Conf. Interval]
-------------+----------------------------------------------------------------
       price |   6165.2568   .0000108   298.10338    5570.475   6700.198  (BC)
------------------------------------------------------------------------------
(BC)   bias-corrected confidence interval
. quietly bstrap, rep(100): mean price
. estat bootstrap
Mean estimation                                 Number of obs      =        74
                                                Replications       =       100

------------------------------------------------------------------------------
             |    Observed               Bootstrap
             |        Mean       Bias    Std. Err.  [95% Conf. Interval]
-------------+----------------------------------------------------------------
       price |   6165.2568  -64.61972   333.47902    5507.595       6903  (BC)
------------------------------------------------------------------------------
(BC)   bias-corrected confidence interval
```

The regular bootstrap provided an estimate of bias that is nonnegligible, while the estimate of bias coming from the balanced bootstrap is a numeric zero.

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

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

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

```
. quietly bs4rw, rw(bw*): ratio price/mpg
. estat bootstrap
Ratio estimation                                Number of obs      =        74
                                                Replications       =       100

------------------------------------------------------------------------------
             |    Observed               Bootstrap
             |       Ratio       Bias    Std. Err.  [95% Conf. Interval]
-------------+----------------------------------------------------------------
    _ratio_1 |   289.48541   .4471263   19.208605    245.2041   322.1176  (BC)
------------------------------------------------------------------------------
(BC)   bias-corrected confidence interval
. quietly bstrap, rep(100): ratio price/mpg
. estat bootstrap
Ratio estimation                                Number of obs      =        74
                                                Replications       =       100

------------------------------------------------------------------------------
             |    Observed               Bootstrap
             |       Ratio       Bias    Std. Err.  [95% Conf. Interval]
-------------+----------------------------------------------------------------
    _ratio_1 |   289.48541  -1.924227     21.9389    257.3014   342.1601  (BC)
------------------------------------------------------------------------------
(BC)   bias-corrected confidence interval
```

Now we see that a nontrivial amount of bias is reported by both methods. However, the estimate coming from the balanced bootstrap is much more trustworthy, because 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

There are several scaling issues associated with replication methods. The first issue is scaling of the point estimates. 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 (8) and (10). The second issue is the scale of the deviations between $\widehat{\theta}^{(r)}$ and $\widehat{\theta}$. In BRR and BWR, these deviations are on the correct scale, so the scaling factor for the resulting estimator $v_m\left(\widehat{\theta}\right)$ to match the known v_r 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 the 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 $\widetilde{\theta}^{(r)}$ and $\widetilde{\theta}$ are on the wrong scale if $m_h \neq n_h - 1$, so modifications like (12) or (13) need to be taken.

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. This guarantees reproducibility of results between different runs and different researchers.

Because designs with two PSUs per stratum are widely used in practice, BRR is the most popular replication variance estimation procedure. Examples of datasets supplied with BRR weights include U.S. datasets from NHANES and the National Education Longitudinal Survey. Given the popularity and simplicity of BRR estimation, survey organizations often approximate their actual designs with stratified PSUs/stratum designs and provide quasi-BRR weights. The modifications to an original design that could make it "BRR-able" include collapsing of strata, reallocating 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, as discussed in section 3.2.

While the U.S. 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. The particular version of the bootstrap favored by Statistics Canada is the Rao–Wu rescaling bootstrap with $m_h = n_h - 1$.

Difficulties may arise in replication variance estimation for domains. Because some units are removed when replicates are constructed, the number of available observations in the domain decreases. Some strata or even the complete replicate dataset 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 using replication methods that lead to nonzero weights for all units, such as Fay's modification of BRR and the mean bootstrap. The latter, however, is not compatible with poststratification and nonresponse adjustments.

7 Acknowledgments

I would like to thank Jeff Pitblado for helpful discussions and code suggestions. I would also like to thank a referee for a very informative review of the article with multiple clarifications. Many of the referee's ideas and suggestions have been incorporated. All remaining errors, omissions, and bugs are my responsibility.

8 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, 29–48. New York: Wiley.

Canty, A. J., A. C. Davison, D. V. Hinkley, and V. Ventura. 2006. Bootstrap diagnostics and remedies. *Canadian Journal of Statistics* 34: 5–27.

Chambers, R. L., and C. J. Skinner, ed. 2003. *Analysis of Survey Data*. New York: Wiley.

Cochran, W. G. 1977. *Sampling Techniques*. 3rd ed. New York: Wiley.

Davison, A. C., D. V. Hinkley, and E. Schechtman. 1986. Efficient bootstrap simulation. *Biometrika* 73: 555–566.

Efron, B., and R. J. Tibshirani. 1993. *An Introduction to the Bootstrap*. New York: 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. *American Statistician* 42: 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, Series B* 52: 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: 735–742.

Gurney, M., and R. S. Jewett. 1975. Constructing orthogonal replications for variance estimation. *Journal of the American Statistical Association* 70: 819–821.

Hedayat, A. S., N. J. A. Sloane, and J. Stufken. 1999. *Orthogonal Arrays: Theory and Applications*. New York: Springer.

Heeringa, S. G., B. T. West, and P. A. Berglund. 2010. *Applied Survey Data Analysis*. Boca Raton, FL: Chapman & Hall/CRC.

Huber, P. J. 1967. The behavior of maximum likelihood estimates under nonstandard conditions. In Vol. 1 of *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 221–233. Berkeley: University of California Press.

Judkins, D. R. 1990. Fay’s method for variance estimation. *Journal of Official Statistics* 6: 223–239.

Kish, L. 1995. *Survey Sampling*. 3rd ed. New York: Wiley.

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: 263–295.

Kott, P. S. 2001. The delete-a-group jackknife. *Journal of Official Statistics* 17: 521–526.

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 (Suppl.): 25–45.

Krewski, D., and J. N. K. Rao. 1981. Inference from stratified samples: Properties of the linearization, jackknife, and balanced repeated replication methods. *Annals of Statistics* 9: 1010–1019.

Lehtonen, R., and E. Pahkinen. 2004. *Practical Methods for Design and Analysis of Complex Surveys*. 2nd ed. New York: Wiley.

Lohr, S. L. 2010. *Sampling: Design and Analysis*. 2nd ed. Pacific Grove, CA: Duxbury.

Lumley, T. S. 2010. *Complex Surveys: A Guide to Analysis Using R*. Hoboken, NJ: Wiley.

McCarthy, P. J. 1969. Pseudo-replication: Half samples. *Review of the International Statistical Institute* 37: 239–264.

McCarthy, P. J., and C. B. Snowden. 1985. The bootstrap and finite population sampling. In *Vital and Health Statistics*, 1–23. Washington, DC: U.S. Government Printing Office.

Nigam, A. K., and J. N. K. Rao. 1996. On balanced bootstrap for stratified multistage samples. *Statistica Sinica* 6: 199–214.

Phillips, O. 2004. Using bootstrap weights with WesVar and SUDAAN. *Research Data Centres Information and Technical Bulletin* 1: 6–15.

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: 620–630.

———. 1988. Resampling inference with complex survey data. *Journal of the American Statistical Association* 83: 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: 209–217.

Rust, K. F., and J. N. K. Rao. 1996. Variance estimation for complex surveys using replication techniques. *Statistical Methods in Medical Research* 5: 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.

———. 2003. Impact of the bootstrap on sample surveys. *Statistical Science* 18: 191–198.

Shao, J., and D. Tu. 1995. *The Jackknife and Bootstrap*. New York: Springer.

Sitter, R. R. 1993. Balanced repeated replications based on orthogonal multi-arrays. *Biometrika* 80: 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. F. Smith, 59–88. New York: Wiley.

Skinner, C. J., D. Holt, and T. M. F. Smith. 1989. *Analysis of Complex Surveys*. New York: Wiley.

Sloane, N. J. A. 2004. A Library of Hadamard Matrices.
http://www2.research.att.com/~njas/hadamard/.

Thompson, M. E. 1997. *Theory of Sample Surveys*. London: Chapman & Hall.

Valliant, R. 1996. Discussion of "Resampling methods in sample surveys" by J. Shao. *Statistics* 27: 247–251.

Valliant, R., J. M. Brick, and J. A. Dever. 2008. Weight adjustments for the grouped jackknife variance estimator. *Journal of Official Statistics* 24: 469–488.

West, B. T., P. Berglund, and S. G. Heeringa. 2008. A closer examination of subpopulation analysis of complex-sample survey data. *Stata Journal* 8: 520–531.

White, H. 1982. Maximum likelihood estimation of misspecified models. *Econometrica* 50: 1–26.

Wolter, K. M. 2007. *Introduction to Variance Estimation*. 2nd ed. New York: Springer.

Wu, C. F. J. 1991. Balanced repeated replications based on mixed orthogonal arrays. *Biometrika* 78: 181–188.

Yeo, D., H. Mantel, and T.-P. Liu. 1999. Bootstrap variance estimation for the National Population Health Survey. In *Proceedings of the Survey Research Methods Section*, 778–785. American Statistical Association.

Yung, W. 1997. Variance estimation for public use files under confidentiality constraints. In *Proceedings of the Survey Research Methods Section*, 434–439. American Statistical Association.

About the author

Stanislav (Stas) Kolenikov is a statistical consultant and an adjunct assistant professor in the Department of Statistics at the University of Missouri in Columbia, MO. His research interests include statistical methods in social sciences, with a focus on structural equation models, microeconometrics, and analysis of complex survey data. He started using Stata in 1998 with version 5.

Appendix. Commonly used notation

The generic datum x_{hij} denotes the measurement on variable x taken on the jth observation in the ith PSU in the hth stratum.

f	sampling fraction: $f = n/N$
f_h	sampling fraction in stratum h: $f_h = n_h/N_h$
$h = 1, \ldots, L$	stratum index
$i = 1, \ldots, n_h$	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 hi is sampled in the rth 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 stratum h; 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 $T(x)$
$v\left(\widehat{\theta}\right)$	estimator of variance $V\left(\widehat{\theta}\right)$
$v_m\left(\widehat{\theta}\right)$	estimator of variance $V\left(\widehat{\theta}\right)$ obtained by method m; the methods include linearization L, the jackknife J, BRR, or the rescaling bootstrap RBS
$V\left(\widehat{\theta}\right)$	(design) variance of the estimate $\widehat{\theta}$ 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 hij
$w_{hij}^{(r)}$	replicate weight of unit hij in the rth replicate
θ	population parameter, such as total, mean, ratio, or regression coefficient
$\widehat{\theta}$	parameter estimate obtained from survey data
$\widehat{\theta}^{(r)}$	parameter estimate obtained in the rth replicate