DACSEIS

IST-2000-26057

Workpackage 5

Resampling Methods for Variance Estimation

Deliverable 5.2

2004

List of contributors:

Anthony C. Davison and Sylvain Sardy, EPFL.

Main responsibility:

Sylvain Sardy, EPFL.

IST–2000–26057–DACSEIS

The DACSEIS research project is financially supported within the IST programme of the European Commission. Research activities take place in close collaboration with Eurostat.

`http://europa.eu.int/comm/eurostat/research/`

`http://www.cordis.lu/ist/`

`http://www.dacseis.de/`

Preface

This report outlines the setup chosen and the methods that underlie the deliverables in the Appendices, namely pseudo-code written in the `R` language for applying resampling methods to sample surveys as required for the DACSEIS project. We review some standard techniques, give key formulae and references, and provide the corresponding pseudo code.

Sylvain Sardy Lausanne, March 2004

Contents

List of Figures

Chapter 1

Introduction

This report accompanies the summer 2002 deliverables for workpackage 5 of the DACSEIS project, on the use of resampling methods for variance estimation in complex surveys. As the pseudo-code for resampling methods that comprises the deliverable would be incomprehensible without first detailing the arguments on which it is based, the report briefly outlines the methods chosen for implementation, and then gives the pseudo-code in appendices. The pseudo-code is written in the `R` language and has been tested using simulated data, as outlined later in the report.

The resampling methods chosen for inclusion on the basis of existing work Canty and Davison (1999) include the bootstrap, the jackknife and its variants, balanced repeated replications, and jackknife linearization, which have been tested for a calibrated Horvitz–Thompson estimator of a total with missing responses imputed using a linear regression model. It would be straightforward to extend this to calibrated ratio estimators and to certain other imputation schemes.

We follow the notation suggested by Davison and Skinner (2001).

1.1 Estimation of a total

We first consider the case of complete response for a stratified random sampling design. Let n_h be the number of units sampled from the N_h population units in stratum h of H strata, and let π_{hi} be the inclusion probability of unit i in stratum h. In household surveys this unit might be a cluster of individuals, in which case we suppose that the unit response of interest is accumulated over the cluster. The total number of population units is $N = \sum_{h=1}^{H} N_h$ and the total number of sampled units is $n = \sum_{h=1}^{H} n_h$. Let x and y be the variables that have been measured on each unit, where x is a vector of q auxiliary variables, which may be continuous or categorical, and y is the scalar response of interest. We are interested in estimating the population total

$$\tau = \sum_{h=1}^{H} \sum_{i=1}^{N_h} y_{hi},$$

using the Horvitz–Thompson estimator

$$\widehat{\tau} = \sum_{h=1}^{H} \sum_{i=1}^{n_h} \omega_{hi} y_{hi} = \omega^{\mathrm{T}} y, \tag{1.1}$$

where y is the vector of responses and $\omega_{hi} = 1/\pi_{hi}$ are the weights associated to the units, the inverse inclusion probabilities. Below we let Ω denote the diagonal matrix whose elements are the ω_{hi}.

1.2 Calibration

In many cases population values are known for some of the auxiliary variables x, and can be used to increase precision of estimation, for example by making some allowance for unit non-response. Suppose that q_C marginals of the q auxiliary variables are known, with $q_C \leq q$. Let c be the vector of the q_C known marginals. Correspondingly, let X_C denote the matrix of auxiliary variables whose marginals are presumed to be known to equal the $q_C \times 1$ vector c. In order to improve the quality of the Horvitz–Thompson estimator, the weights w_{hi} can be calibrated (DEVILLE and SÄRNDAL, 1992) to be as close as possible to the original weights ω in some metric, subject to the constraint that the weighted auxiliary variables match the marginals, that is, $\sum_{h=1}^{H} \sum_{i=1}^{n_h} w_{hi} x_{Chi} = c$. If the ℓ_2 or squared error metric is used, then the calibrated weights equal

$$w = \omega + \Omega X_C (X_C^{\mathrm{T}} \Omega X_C)^{-1} (c - X_C^{\mathrm{T}} \omega), \tag{1.2}$$

and the calibrated Horvitz–Thompson estimator is

$$\begin{aligned} \widehat{\tau} &= w^{\mathrm{T}} y \\ &= \omega^{\mathrm{T}} y + (c - X_C^{\mathrm{T}} \omega)^{\mathrm{T}} (X_C^{\mathrm{T}} \Omega X_C)^{-1} X_C^{\mathrm{T}} \Omega y \\ &= \omega^{\mathrm{T}} y + (c - X_C^{\mathrm{T}} \omega)^{\mathrm{T}} \widehat{\gamma}, \end{aligned}$$

say, $\widehat{\gamma}$ being the regression estimator when y is regressed on X_C with weight matrix Ω. The second term of $\widehat{\tau}$ is an adjustment to the weights ω that accounts for the difference between $X_C^{\mathrm{T}} \omega$ and the target population value c of the margins.

A wide range of other measures of the distance between w and ω can be envisaged, constructed for example to ensure that $w \geq 0$, but they give results asymptotically equivalent to those above, which involve only a regression computation.

1.3 Imputation

When some responses are missing, the simplest approach is to drop the corresponding units from the analysis and to adjust the inclusion probabilities of the respondents accordingly. Though simple, there is an obvious potential loss of information when this idea is applied but the auxiliary variables for the nonrespondents are available. The imputation idea is to use the auxiliary variables to predict the missing responses through a model based on

the respondents only. Here and in the calculations below we assume that a linear model is used for imputation of missing responses, based on the corresponding vectors x_{hi} of auxiliary variables. That is, we assume either a common linear model across strata,

$$Y_{hi} = x_{hi}^{\mathrm{T}}\beta + \epsilon_{hi},$$

or a different linear model for each stratum,

$$Y_{hi} = x_{hi}^{\mathrm{T}}\beta_h + \epsilon_{hi}, \quad h = 1, \ldots, H,$$

fit this model to those individuals for which both x_{hi} and y_{hi} are available, and then use the fitted linear model to impute predicted responses for individuals for whom y_{hi} is missing. Below we shall suppose that the estimated coefficients $\widehat{\beta}$ are obtained by least squares estimation, but only minor modifications are needed to use other approaches such as robust M-estimation. The missing response for an individual with explanatory variable x can then be predicted by $x^{\mathrm{T}}\widehat{\beta}$.

Let $z_{hi} = I(y_{hi} \neq \texttt{NA})$ be the indicator random variable corresponding to observed response, let $Z = \mathrm{diag}(z)$ be the $n \times n$ diagonal matrix of these indicators, and let X be the $n \times q$ regression matrix for the linear model. This contains auxiliary variables corresponding to both respondents and nonrespondents. Also let $\widehat{y} = X\widehat{\beta}$ represent the $n \times 1$ vector of fitted values from the regression model used for imputation. Then (1.2) implies that the calibrated and imputed Horvitz–Thompson estimator may be written as

$$\begin{aligned}
\widehat{\tau} &= w^{\mathrm{T}}\{Zy + (I - Z)\widehat{y}\} \\
&= \omega^{\mathrm{T}} Z y + (c - X_C^{\mathrm{T}}\omega)^{\mathrm{T}}(X_C^{\mathrm{T}}\Omega X_C)^{-1} X_C^{\mathrm{T}}\Omega Z y \\
&\quad + \omega^{\mathrm{T}}(I - Z)X\widehat{\beta} + (c - X_C^{\mathrm{T}}\omega)^{\mathrm{T}}(X_C^{\mathrm{T}}\Omega X_C)^{-1} X_C^{\mathrm{T}}\Omega(I - Z)X\widehat{\beta}. \qquad (1.3)
\end{aligned}$$

This is a consistent estimator of τ if the imputation is consistent.

Computation of the variance of the calibrated and imputed Horvitz–Thompson estimator as if the imputed responses $\widehat{y}$ were true responses can lead to considerable underestimation of the true variance, so the variance estimation technique must reflect the variance inflation due to imputation. We discuss how this is performed below.

Chapter 2

Variance estimation methods

2.1 Jackknife

The jackknife computes estimates of the properties such as the bias or variance of an estimator by systematically deleting groups of units at a time, recomputing the statistic with each group deleted in turn, and then combining all these recalculated statistics. The original, "delete-one", jackknife is due to Quenouille (1949) and Tukey (1958). Miller (1974) is an excellent review of early work, and a recent more extended account is given by Shao and Tu (1995).

With stratified data, the delete-one jackknife involves deleting each unit in turn, giving the statistics $\widehat{\tau}_{h,-i}$ calculated without the ith unit of stratum h, for all $i = 1, \ldots, n_h$ and $h = 1, \ldots, H$. Then the delete-one jackknife estimate of variance is given by

$$v_J = \sum_{h=1}^{H} \frac{(1-f_h)(n_h-1)}{n_h} \sum_{i=1}^{n_h} (\widehat{\tau}_{h,-i} - \overline{\widehat{\tau}}_h)^2, \quad \overline{\widehat{\tau}}_h = n_h^{-1} \sum_{i=1}^{n_h} \widehat{\tau}_{h,-i},$$

where $f_h = n_h/N_h$ is the sampling fraction in the hth stratum. Although simple, the delete-one jackknife is computationally intensive because a total of n deletions is required. Moreover, the variance estimates it produces are inconsistent for non-smooth statistics such as the median or other statistics based on quantiles.

The "delete-d" jackknife Shao and Wu (1989) reduces the number of replications needed, by splitting stratum h into m_h disjoint blocks of d observations, so that $dm_h = n_h$, and then systematically recomputing the statistic with each of these blocks deleted. The total number of recomputations is thus $m = \sum_{h=1}^{H} m_h$ rather than n. Variants choose the block members randomly at each recomputation, with or without replacement; see Section 5.2.2. of Shao and Tu (1995). For any of these the variance estimate is

$$v_{J-d} = \sum_{h=1}^{H} \frac{(1-f_h)(n_h-d)}{dm_h} \sum_{i=1}^{m_h} (\widehat{\tau}_{h,-i} - \overline{\widehat{\tau}}_h)^2.$$

To match the computational complexity of the bootstrap-based variance estimation technique which typically uses $m = 100$ or $m = 200$ resampling calculations, we use the same number of block deletions in our simulations below.

The delete-d jackknife has the added benefit of extending the range of functions to which the jackknife may be applied, as it turns out that if $d, m_h \to \infty$ together then the block jackknife gives valid variance estimators (SHAO and WU, 1989). It is not clear, however, how to choose d and m_h when n_h is finite.

RAO and SHAO (1992) adjust the delete-1 jackknife variance estimator in the presence of nonresponse by accounting for the imputation mechanism. We slightly generalize their work by allowing block deletion. This computer intensive approach amounts to re-imputing each time a block is deleted using the respondents of the remaining data to estimate the regression model.

2.2 Balanced repeated replication

Balanced half–sampling MCCARTHY (1969) is the simplest form of balanced repeated replication. When a sample consists of two observations in each stratum, then a half–sample is formed by taking one observation from each of the H strata. The statistic can be recalculated 2^H times, and the results combined to estimate the variance of the original statistic. When H is large, balanced half–sampling is computationally intensive, but ideas from experimental design allow fewer recalculations, say $L < 2^H$, resulting in the same result for linear statistics as would computation of all 2^H replicates. The simplest approach when each stratum has more than two observations is to split the stratum randomly into two groups of nearly equal size.

SHAO *et al.* (1998) adjust balanced repeated replication to the presence of nonresponse, by taking into account a deterministic or random imputation mechanism.

2.3 Bootstrap

The bootstrap idea is to mimic how the original data were generated. Like the jackknife and balanced repeated replication, the bootstrap involves recomputing the statistic, now using resampling from an estimated population $\widehat{F}$ to obtain bootstrap samples that may be represented by $\widehat{F}^*$, giving corresponding statistics $\widehat{\theta}^* = t(\widehat{F}^*)$. Repeating this process m times, the bootstrap estimate of variance is given by

$$v_B = (m-1)^{-1} \sum_{i=1}^{m} (\widehat{\theta}_i^* - \overline{\widehat{\theta}^*})^2, \quad \overline{\widehat{\theta}^*} = m^{-1} \sum_{i=1}^{m} \widehat{\theta}_i^*.$$

For stratified data, the resampling is performed within each stratum independently. This bald description assumes that the sampling fractions are negligible, as is typically the case for labour force surveys. Numerous more sophisticated approaches have been suggested for use with non-negligible sampling fractions, of which most are too complicated for implementation in practice; see Section 3.7 of DAVISON and HINKLEY (1997). PRESNELL and BOOTH (1994) point out theoretical difficulties with many of the published approaches. The simplest and most satisfactory approach in the case of an unstratified sample is to create an artificial population using N/n replicates of the sample, and then

to sample from this artificial population without replacement. For stratified samples the hth stratum is replicated N_h/n_h times, and then stratified sampling without replacement is applied.

When responses are missing, the imputation mechanism must be applied to each resample $\widehat{F}^*$ (SHAO and SITTER, 1996). Thus we must re-impute repeatedly using the respondents of the bootstrapped sample to fit the imputation model and then impute the nonrespondents of the bootstrap sample. It is therefore rather computer intensive, but on the other hand gives consistent variance estimators for medians and other estimators based on quantiles.

2.4 Jackknife linearization

In problems with random samples the quantity θ we want to estimate can often be written as a functional $t(F)$ of the underlying distribution function F which generated the data. A simple estimator of $t(F)$ is then $t(\widehat{F})$, where $\widehat{F}$ is the empirical distribution function of the data. For the mean, for instance, $t(F) = \int y dF(y)$ and $t(\widehat{F}) = \overline{Y}$ is its empirical analogue. In this case, and assuming some differentiability properties for the functional $t(\cdot)$, the estimate $\widehat{\theta} = t(\widehat{F})$ can be expanded around $\theta = t(F)$ using Taylor series as

$$t(\widehat{F}) \doteq t(F) + n^{-1} \sum_{i=1}^{n} L_t(Y_i; F),$$

where

$$L_t(y; F) = \lim_{\epsilon \to 0} \frac{t\{(1-\epsilon)F + \epsilon\delta_y\} - t(F)}{\epsilon}$$

is the *influence function* for $t(\widehat{F})$, δ_y being the distribution function putting a point mass at y. This expansion can be used to establish that the estimator is asymptotically unbiased and Gaussian. Its variance $v_L(F) = n^{-1}\text{var}\{L_t(Y; F)\}$ can be estimated by

$$\widehat{v}_L = n^{-2} \sum_{i=1}^{n} l_i^2, \tag{2.1}$$

where $l_i = L_t(y_i; \widehat{F})$ are the *empirical influence values* for the statistical functional t evaluated at the observation y_i and the empirical distribution function $\widehat{F}$. Here l_i can be thought of as the derivative of t at $\widehat{F}$ in the direction of a distribution putting more mass on the ith observation. Section 2.11 of DAVISON and HINKLEY (1997) gives more details of these expansions and examples of these calculations.

For stratified sampling without replacement this variance estimate may be modified to

$$v_L = \sum_{h=1}^{H} (1 - f_h) \frac{1}{(n_h - 1)n_h} \sum_{i=1}^{n_h} l_{hi}^2, \tag{2.2}$$

where l_{hi} is the empirical influence value corresponding to the ith observation in stratum h. In simple situations (2.1) and (2.2) recover standard linearization formulae, but the use of influence functions gives a more general approach to variance estimation.

We now consider the Horvitz–Thompson estimator and give the formula for its empirical influence functions for stratified sampling in three situations of increasing complexity:

- the standard Horvitz–Thompson estimator given at (1.1) above yields

$$l_{hi} = n_h \omega_{hi} y_{hi} - \omega_h^{\mathrm{T}} y_h;$$

- the calibrated Horvitz–Thompson estimator (1.2) yields

$$\begin{aligned} l_{hi} &= (n_h \omega_{hi} y_{hi} - \omega_h^{\mathrm{T}} y_h) + (X_{Ch}^{\mathrm{T}} \omega_h - n_h \omega_{hi} x_{Chi})^{\mathrm{T}} \widehat{\gamma} \\ &\quad + n_h \omega_{hi} (c - X_C^{\mathrm{T}} \omega)^{\mathrm{T}} (X_C^{\mathrm{T}} \Omega X_C)^{-1} x_{Chi} (y_{hi} - x_{Chi}^{\mathrm{T}} \widehat{\gamma}), \end{aligned}$$

  where ω_h and y_h are $n_h \times 1$ vectors of the weights and responses for the hth stratum, X_{Ch} is the $n_h \times q_C$ matrix of calibration covariates for the hth stratum, and $\widehat{\gamma} = (X_C^{\mathrm{T}} \Omega X_C)^{-1} X_C^{\mathrm{T}} \Omega y$; and

- the calibrated Horvitz–Thompson estimator (1.3) with imputation of missing responses. Let

$$\widehat{\gamma}_M = (X_C^{\mathrm{T}} \Omega X_C)^{-1} X_C^{\mathrm{T}} \Omega (I - Z) \widehat{y}$$

  correspond to $\widehat{\gamma}$, but for those individuals with missing responses, and let $l_i(\widehat{\beta})$ be the elements of the $q \times 1$ vector of influence functions for the imputation regression coefficients, corresponding to differentiation with respect to the ith case in stratum h. Then

$$\begin{aligned} l_{hi} &= (n_h \omega_{hi} z_{hi} y_{hi} - \omega_h^{\mathrm{T}} Z_h y_h) + (X_{Ch}^{\mathrm{T}} \omega_h - n_h \omega_{hi} x_{Chi})^{\mathrm{T}} \widehat{\gamma} \\ &\quad + n_h \omega_{hi} (c - X_C^{\mathrm{T}} \omega)^{\mathrm{T}} (X_C^{\mathrm{T}} \Omega X_C)^{-1} x_{Chi} (z_{hi} y_{hi} - x_{Chi}^{\mathrm{T}} \widehat{\gamma}) \\ &\quad + \{n_h \omega_{hi} (1 - z_{hi}) \widehat{y}_{hi} - \omega_h^{\mathrm{T}} (I_h - Z_h) \widehat{y}_h\} + \omega^{\mathrm{T}} (I - Z) X l_i(\widehat{\beta}) \\ &\quad + (X_{Ch}^{\mathrm{T}} \omega_h - n_h \omega_{hi} x_{Chi})^{\mathrm{T}} \widehat{\gamma}_M \\ &\quad + n_h \omega_{hi} (c - X_C^{\mathrm{T}} \omega)^{\mathrm{T}} (X_C^{\mathrm{T}} \Omega X_C)^{-1} x_{Chi} \{(1 - z_{hi}) \tilde{y}_{hi} - x_{Chi}^{\mathrm{T}} \widehat{\gamma}_M\} \\ &\quad + (c - X_C^{\mathrm{T}} \omega)^{\mathrm{T}} (X_C^{\mathrm{T}} \Omega X_C)^{-1} X_{Ch}^{\mathrm{T}} \Omega_h (I_h - Z_h) X_h l_i(\widehat{\beta}). \end{aligned}$$

  For instance, *least squares deterministic regression mean* imputation has empirical influence values

$$l_i(\widehat{\beta}) = n_h z_i (X^{\mathrm{T}} Z X)^{-1} x_i (y_i - x_i^{\mathrm{T}} (X^{\mathrm{T}} Z X)^{-1} X^{\mathrm{T}} Z y),$$

  where X is the regression matrix. If the regression model is stratum dependent, then X, n, and $l_i(\widehat{y})$ are replaced in this expression by X_h, n_h, and $l_{hi}(\widehat{y})$.

Chapter 3

Simulation study

This simulation is not intended as a realistic indication of the likely performance of the variance estimation methods, rather as a crude test on the correctness of the pseudo-code.

We generated a population of size $N = 10^5$ with three strata of size 25,000, 25,000, 50,000. The auxiliary variables are two discrete variables whose population totals are taken as known, and a three-level factor. Response variables were created using a linear model in the auxiliary variables with Gaussian noise.

We apply stratified sampling with $n_1 = 200$, $n_2 = 500$ and $n_3 = 300$ and uniform inclusion probability across each stratum, calculate the Horvitz-Thompson estimator, and its estimated standard deviation using the approaches described above. This sampling scheme was repeated 100 times with non-response levels of 0, 20%, 40% and 60%, yielding the results plotted in Figure 3.1. The bootstrap was applied with $m = 100$ replicates, and the jackknife used $m = 100$ and $d = 10$, giving 300 replicates in all.

Each of the lower panels shows an estimate of the true standard deviation, obtained from 10,000 replications of the sampling scheme, with boxplots of the variance estimators. The results are not surprising, and suggest that the linearization and bootstrap approaches work well, while balanced repeated replications works poorly in this context. This is hardly surprising, as when there are just 3 strata balanced repeated replications is based on very few replicates. It would be expected to become more reliable and more competitive with the other approaches when there are more strata.

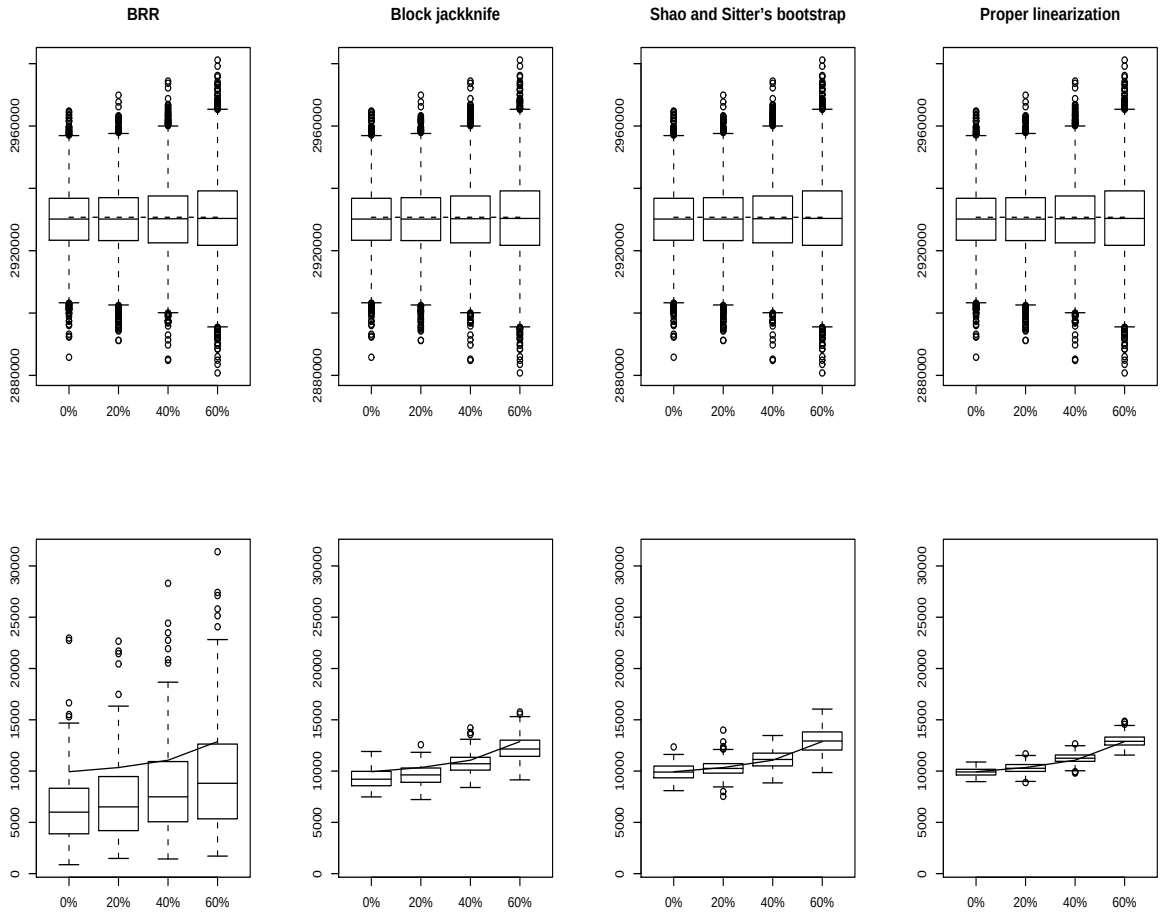

Figure 3.1: Simulation results. On top are the 100 Horvitz–Thompson estimates (boxplot) and the true population total (dotted line). At the bottom are the 100 estimated standard deviations (boxplot) and an estimate of the true standard deviation obtained from 10,000 replications.

Chapter 4

Discussion

All the resampling methods, that is, the jackknife, the balanced repeated replication and the bootstrap methods, extend straightforwardly to other statistics. As far as the jackknife linearization method is concerned, new formulae must be derived in a case by case basis for other statistics, a task which can be difficult for quantile estimators. The chain rule can be applied in certain cases, leading to simple derivations: for the calibrated ratio estimator $\widehat{\theta} = w^{\mathrm{T}} y_1 / w^{\mathrm{T}} y_2$ for instance, the empirical influence function is simply

$$l_{hi} = \frac{l_{hi}^{y_1} - \widehat{\theta} l_{hi}^{y_2}}{w^T y_2},$$

where $l_{hi}^{y_1}$ and $l_{hi}^{y_1}$ are the empirical influence values for the Horwitz–Thompson estimators $w^{\mathrm{T}} y_1$ and $w^{\mathrm{T}} y_2$, possibly with calibration and imputation, and the ith unit in stratum h. Formulae such as (2.2) are then applied directly using the l_{hi}.

Appendix A

Jackknife

Program A.1: Jackknife

```
varblockJK.HT.Calibrated.Total <-
  function(X,y,iip,stratareg,marginals,
           Xtype,fpc,impute=c("LS","across","deterministic"),
           strata.blockJK,d,m)
{
  ## Approximation to the block (delete-d) jackknife with random
     subsampling
  ## with replacement (See Shao, J. and Tu, D. (1995)
  ## "The Jackknife and Bootstrap", Springer, Section 5.2.2

  ## Notation
  ## n = nbr of sampled units
  ## q = nbr of auxiliary variables

  ## Input
  ## X = matrix of auxiliary variables organized columnwise (n rows, q
     colums)
  ## y = vector of responses values (some can be missing)
  ## iip = inverse inclusion probabilities (uncalibrated)
  ## stratareg = vector of length n with stratum indicator
  ##   used for the regression imputation
  ## marginals = numerical vector (of length q) giving the marginals of
  ##    the auxilairy variables (some can be missing, ie, = NA)
  ## Xtype = character vector (of size q) giving the type of
  ##    auxiliary variables: "c"=continuous, "f"=factor
  ## fpc = finite population correction vector for each stratum
  ## impute = vector of length 3 giving information about
  ##    the type of imputation:
  ##    o impute[1] = "LS" for standard least squares regression,
  ##      or "RH" for robust regression with the Huber loss
  ##    o impute[2] = "across" if the regression is "across" strata,
  ##      or "within" strata
  ##    o impute[3] = "deterministic" for mean imputation,
  ##      or "stochastic" for random imputation
```

```
  ## strata.blockJK = vector of length n with stratum indicator
  ##     used for the jackknife
  ## d = block size removed for delete-d jackknife (e.g., 10)
  ## m = nbr of repetitions overall >> d (e.g., 100)

  ## Output
  ## delete-d jackknife variance estimate

  ## Start
  strata.names <- unique(strata.blockJK)
  H <- length(strata.names)
  mps <- floor(m/H)
  n <- length(y)

  vartheta.hat <- 0
  res.h <- rep(NA, mps)
  for(h in 1:H)
    {
      strata.name.h <- strata.names[h]
      strata.blockJK.h <- (strata.blockJK==strata.name.h)
      nh <- sum(strata.blockJK.h)
      fpc.h <- fpc[h]
      for(j in 1:mps){
        ind.j <- sample(c(1:n),d,replace=F,prob=strata.blockJK.h)
        Xh.j <- X[-ind.j,]
        yh.j <- y[-ind.j]
        iip.j <- nh/(nh-d)*iip[-ind.j]
        stratareg.j <- stratareg[-ind.j]
        out.j <- HT.Calibrated.Total(Xh.j,yh.j,iip.j,stratareg.j,
                                     marginals,Xtype,fpc,impute,JKlin=F)
        res.h[j] <- out.j$theta.hat
      }
      vartheta.hat <- vartheta.hat +
        fpc.h*(nh-d)/d*mean((res.h-mean(res.h))^2)
    }

  ## Output
  sdtheta.hat <- sqrt(vartheta.hat)
  return(sdtheta.hat)
}
```

Appendix B

Balanced repeated replication

Program B.1: Balanced repeated replication

```
varBRRaverage.HT.Calibrated.Total <-
  function(nbr.repeat, X,y,iip,stratareg,marginals, BRRmatrix, Xtype,fpc,
           impute=c("LS","across","deterministic"),strataBRR)
{
  ## Averaged Balanced repeated replication

  ## Notation
  ## n = nbr of sampled units
  ## q = nbr of auxiliary variables

  ## Input
  ## X = matrix of auxiliary variables organized columnwise (n rows, q
      colums)
  ## y = vector of responses values (some can be missing)
  ## iip = inverse inclusion probabilities (uncalibrated)
  ## stratareg = vector of length n with stratum indicator
  ##     used for the regression imputation
  ## marginals = numerical vector (of length q) giving the marginals of
  ##     the auxilairy variables (some can be missing, ie, = NA)
  ## BRRmatrix = matrix used to balance the replications
  ## Xtype = character vector (of size q) giving the type of
  ##     auxiliary variables: "c"=continuous, "f"=factor
  ## fpc = finite population correction vector for each stratum
  ## impute = vector of length 3 giving information about
  ##     the type of imputation:
  ##       o impute[1] = "LS" for standard least squares regression,
  ##         or "RH" for robust regression with the Huber loss
  ##       o impute[2] = "across" if the regression is "across" strata,
  ##         or "within" strata
  ##       o impute[3] = "deterministic" for mean imputation,
  ##         or "stochastic" for random imputation
  ## strataBRR = vector of length n with stratum indicator

  ## Output
```

```
  ## Balanced repeated replication variance estimate averaged
  ## over nbr.repeat random BRRs

  ## Start
  var.hat <- 0
  for(i in 1:nbr.repeat){
    out.i <-
      varBRR.HT.Calibrated.Total(X,y,iip,stratareg,marginals, BRRmatrix,
          Xtype,fpc,
                                 impute=c("LS","across","deterministic"),
                                 strataBRR)
    var.hat <- var.hat + out.i$sdtheta.hat^2
  }
  sdtheta.hat <- sqrt(var.hat/nbr.repeat)

  return(sdtheta.hat)
}



varBRR.HT.Calibrated.Total <-
  function(X,y,iip,stratareg,marginals, BRRmatrix, Xtype,fpc,
           impute=c("LS","across","deterministic"),strataBRR)
{
  ## Balanced repeated replication

  ## Notation
  ## n = nbr of sampled units
  ## q = nbr of auxiliary variables

  ## Input
  ## X = matrix of auxiliary variables organized columnwise (n rows, q
      colums)
  ## y = vector of responses values (some can be missing)
  ## iip = inverse inclusion probabilities (uncalibrated)
  ## stratareg = vector of length n with stratum indicator
  ##     used for the regression imputation
  ## marginals = numerical vector (of length q) giving the marginals of
  ##     the auxilairy variables (some can be missing, ie, = NA)
  ## BRRmatrix = matrix used to balance the replications
  ## Xtype = character vector (of size q) giving the type of
  ##     auxiliary variables: "c"=continuous, "f"=factor
  ## fpc = finite population correction vector for each stratum
  ## impute = vector of length 3 giving information about
  ##     the type of imputation:
  ##       o impute[1] = "LS" for standard least squares regression,
  ##         or "RH" for robust regression with the Huber loss
  ##       o impute[2] = "across" if the regression is "across" strata,
  ##         or "within" strata
  ##       o impute[3] = "deterministic" for mean imputation,
  ##         or "stochastic" for random imputation
  ## strataBRR = vector of length n with stratum indicator

  ## Output
  ## theta.hat = Horvitz-Thompson estimate
  ## thetas = Balanced estimates
```

```
  ## sdtheta.hat = Balanced repeated replication variance estimate

  ## Start
  stratanames <- unique(strataBRR)
  H <- length(stratanames)
  R <- dim(BRRmatrix)[1]

  ##  mps <- floor(m/H)
  n <- length(y)

  ## Original estimate on complete sample
  outo <- HT.Calibrated.Total(X,y,iip,stratareg,
                              marginals,Xtype,fpc,impute,JKlin=F)
  theta.hato <- outo$theta.hat

  ## Random split of each stratum into two blocks
  indicmat <- matrix(0,n,H)
  for(h in 1:H){
    strataname.h <- stratanames[h]
    strataBRR.h <- (strataBRR==strataname.h)
    nh <- sum(strataBRR.h)
    index <- rep(F,n)
    index[sample(c(1:n),floor(nh/2),replace=F,prob=strataBRR.h)] <- T
    indicmat[,h] <- index
  }

  ## Balanced replication
  theta.hats <- rep(0, R)
  j <- 0
  for(i in 1:R){
    ind.i <- rep(0,n)
    for(h in 1:H){
      pm <- BRRmatrix[i,h]
      ind.ih <- indicmat[,h]
      strataname.h <- stratanames[h]
      strataBRR.h <- (strataBRR==strataname.h)
      if(pm==-1) ind.i <- ind.i+ind.ih
      if(pm==1) ind.i <- ind.i+((!ind.ih)&(strataBRR.h))
    }
    ind.i <- (!ind.i)
    X.i <- X[ind.i,]; y.i <- y[ind.i]; iip.i <- iip[ind.i]
    stratareg.i <- stratareg[ind.i]
    out.i <- HT.Calibrated.Total(X.i,y.i,2*iip.i,stratareg.i,
                                 marginals,Xtype,fpc,impute,JKlin=F)
    theta.hats[i] <- out.i$theta.hat
  }

  ## Output
  out <- NULL
  out$theta.hat <- theta.hato
  out$thetas <- theta.hats
  out$sdtheta.hat <- sqrt(mean((theta.hats-theta.hato)^2))

  return(out)
}
```

```
##  Example of input file for BRR for R=8
##
##    1 -1 -1  1 -1  1  1 -1
##    1  1 -1 -1  1 -1  1 -1
##    1  1  1 -1 -1  1 -1 -1
##   -1  1  1  1 -1 -1  1 -1
##    1 -1  1  1  1 -1 -1 -1
##   -1  1 -1  1  1  1 -1 -1
##   -1 -1  1 -1  1  1  1 -1
##   -1 -1 -1 -1 -1 -1 -1 -1
```

Appendix C

Bootstrap

Program C.1: Bootstrap

```
varboot.HT.Calibrated.Total.r <-
  function(X,y,iip,stratareg,marginals,Xtype,fpc,
           impute=c("LS","across","deterministic"),
           strataboot,weights.boot,R)
{
  ## Notation
  ## n = nbr of sampled units
  ## q = nbr of auxiliary variables

  ## Input
  ## X = matrix of auxiliary variables organized columnwise (n rows, q
      colums)
  ## y = vector of responses values (some can be missing)
  ## iip = inverse inclusion probabilities (uncalibrated)
  ## stratareg = vector of length n with stratum indicator
  ##   used for the regression imputation
  ## marginals = numerical vector (of length q) giving the marginals of
  ##   the auxilairy variables (some can be missing, ie, = NA)
  ## Xtype = character vector (of size q) giving the type of
  ##   auxiliary variables: "c"=continuous, "f"=factor
  ## fpc = finite population correction vector for each stratum
  ## impute = vector of length 3 giving information about
  ##   the type of imputation:
  ##   o impute[1] = "LS" for standard least squares regression,
  ##     or "RH" for robust regression with the Huber loss
  ##   o impute[2] = "across" if the regression is "across" strata,
  ##     or "within" strata
  ##   o impute[3] = "deterministic" for mean imputation,
  ##     or "stochastic" for random imputation
  ## strataboot = vector of length n with stratum indicator
  ##     used for the bootstrap (arguement of the 'boot' function)
  ## weights.boot = vector of length n with probabilities used
  ##     for the boostrap (arguement of the 'boot' function)
  ## R = number of bootstrap replicates (arguement of the 'boot' function)
```

```
  ## Output
  ## output of the 'boot' function  with the 'boot.HT.Calibrated'
      statistic

  impute.results <- HT.Calibrated.Total(X,y,iip,stratareg,marginals,
                                        Xtype,fpc,impute,JKlin=F)
  yI <- impute.results$y.imputed
  data <- cbind(X,yI,iip,!is.na(y))
  outBoot <- boot(data,boot.HT.Calibrated.Total,R,
                  strata=strataboot,weights=weights.boot,
                  stratareg=stratareg,marginals=marginals,
                  Xtype=Xtype,fpc=pfc,impute=impute)

  return(outBoot)
}

boot.HT.Calibrated.Total <-
  function(data,i,stratareg,marginals,Xtype,fpc,
           impute=c("LS","across","deterministic"))
{
  ## Notation
  ## n = nbr of sampled units
  ## q = nbr of auxiliary variables

  ## Input
  ## X = matrix of auxiliary variables organized columnwise (n rows, q
      colums)
  ## y = vector of responses values (some can be missing)
  ## d = inverse inclusion probabilities (uncalibrated)
  ## stratareg = vector of length n with stratum indicator
  ## marginals = numerical vector (of length q) giving the marginals of
  ##   the auxilairy variables (some can be missing, ie, = NA)
  ## Xtype = character vector (of size q) giving the type of
  ##   auxiliary variables: "c"=continuous, "f"=factor
  ## fpc = finite population correction vector for each stratum
  ## impute = vector of length 3 giving information about
  ##   the type of imputation:
  ##   o impute[1] = "LS" for standard least squares regression,
  ##     or "RH" for robust regression with the Huber loss
  ##   o impute[2] = "across" if the regression is "across" strata,
  ##     or "within" strata
  ##   o impute[3] = "deterministic" for mean imputation,
  ##     or "stochastic" for random imputation

  ## Output
  ## theta.hat = calibrated Horvitz-Thomson estimator for population total

  ## Start
  ncol <- dim(data)[2]
  X <- data[i,-c(ncol-2,ncol-1,ncol)]
  y <- data[i,ncol-2]
  d <- data[i,ncol-1]
  z <- data[i,ncol]

  stratanames <- unique(stratareg)
  H <- length(stratanames)
```

```
  ## Calibration
  known.marginals <- (!is.na(marginals))
  marginals <- marginals[known.marginals]

  Xcal <- as.matrix(X[,known.marginals])
  Xcalt.d <- as.vector(t(Xcal) %*% d)
  Xcalt.D <- t(d*Xcal)
  vec1 <- marginals - Xcalt.d
  inv.d <- solve(Xcalt.D %*% Xcal)
  w <- d
  w <- d + as.vector(t(vec1) %*% inv.d %*% Xcalt.D)

  ## Imputation
  ## Right now, only deterministic imputation has been done
  if(sum(!z)>0)
    {
      RegTech <- impute[1]
      Imputype <- impute[3]
      if(impute[2] == "across")
        {
          out.impute <- RegressionImputation(X,y,z,Xtype,RegTech,Imputype)
          residuals <- out.impute$residuals
          y[!z] <- out.impute$yNA.imputed
        }
      else
        {
          for(h in 1:H)
            {
              strataname.h <- stratanames[h]
              stratareg.h <- (stratareg==strataname.h)
              Xh <- X[stratareg.h,]
              yh <- y[stratareg.h]
              zh <- z[stratareg.h]
              out.impute.h <- RegressionImputation(Xh,yh,zh,Xtype,RegTech,
                  Imputype)
              y[(!z)&(stratareg.h)] <- out.impute.h$yNA.imputed.h
            }
        }
    }

  ## End
  theta.hat <- sum(as.vector(w)*y)
  return(theta.hat)
}
```

Appendix D

Jackknife linearization

Program D.1: Jackknife linearization

```
HT.Calibrated.Total <-
  function(X,y,iip,stratareg,marginals,Xtype,fpc,
           impute=c("LS","across","deterministic"),
           JKlin=F)
{
  ## Notation
  ## n = nbr of sampled units
  ## q = nbr of auxiliary variables

  ## Input
  ## X = matrix of auxiliary variables organized columnwise (n rows, q
      colums)
  ## y = vector of responses values (some can be missing)
  ## iip = inverse inclusion probabilities (uncalibrated)
  ## stratareg = vector of length n with stratum indicator
  ## marginals = numerical vector (of length q) giving the marginals of
  ##   the auxilairy variables (some can be missing, ie, = NA)
  ## Xtype = character vector (of size q) giving the type of
  ##   auxiliary variables: "c"=continuous, "f"=factor
  ## fpc = finite population correction vector for each stratum
  ## impute = vector of length 3 giving information about
  ##   the type of imputation:
  ##   o impute[1] = "LS" for standard least squares regression,
  ##     or "RH" for robust regression with the Huber loss
  ##   o impute[2] = "across" if the regression is "across" strata,
  ##     or "within" strata
  ##   o impute[3] = "deterministic" for mean imputation,
  ##     or "stochastic" for random imputation
  ## JKlin = T or F: if T, the Jackknife linearization formula
  ##   is evaluated to estimate the variance of the estimator for total

  ## Output
  ## w = calibrated weights
  ## theta.hat = calibrated Horvitz-Thomson estimator for population total
```

```
## eif = empirical influence function
## sdtheta.hat = estimate standard deviation of
##   the calibrated Horvitz-Thomson estimator for population total

## Start
out <- NULL
z <- (!is.na(y))
stratanames <- unique(stratareg)
H <- length(stratanames)
n <- length(z)

## Calibration
known.marginals <- (!is.na(marginals))
marginals <- marginals[known.marginals]

Xcal <- as.matrix(X[,known.marginals])
Xcalt.iip <- as.vector(t(Xcal) %*% iip)
Xcalt.IIP <- t(iip*Xcal)
vec1 <- marginals - Xcalt.iip
inv.iip <- solve(Xcalt.IIP %*% Xcal)
w <- iip
w <- iip + as.vector(t(vec1) %*% inv.iip %*% Xcalt.IIP)
out$w <- w

out$eif <- NULL
out$sdtheta.hat <- NULL
y[!z] <- 0
li <- rep(0, length(y))
if(JKlin){
  zy <- z*y
  beta.y <- inv.iip %*% Xcalt.IIP %*% zy
  Xcal.inv.iip <- Xcal %*% inv.iip
  Xcalbeta.y <- Xcal %*% beta.y
  Xcal.inv.iip.vec1 <- Xcal.inv.iip %*% vec1
  for(h in 1:H){
    strataname.h <- stratanames[h]
    stratareg.h <- (stratareg==strataname.h)
    n.h1 <- sum(z[stratareg.h])
    n.h2 <- sum(stratareg.h)
    n.h1 <- n.h2
    iip.h <- iip[stratareg.h]
    nd.h1 <- n.h1*iip.h
    nd.h2 <- n.h2*iip.h
    z.h <- z[stratareg.h]
    y.h <- y[stratareg.h]
    zy.h <- z.h*y.h
    Xcalbeta.y.h <- as.matrix(Xcalbeta.y[stratareg.h])
    Xcalt.iip.h <- as.vector(t(Xcal[stratareg.h,]) %*% iip.h)
    Xcal.inv.iip.h <- Xcal.inv.iip[stratareg.h,]
    Xcal.inv.iip.vec1.h <- as.matrix(Xcal.inv.iip.vec1[stratareg.h,])
    ## Empirical influence functions corresponding to respondents
    li.h <- nd.h1*zy.h - sum(iip.h*zy.h)
    li.h <- li.h + sum(Xcalt.iip.h * beta.y) - nd.h2*Xcalbeta.y.h
    li.h <- li.h + nd.h2*(zy.h-Xcalbeta.y.h)*Xcal.inv.iip.vec1.h
    li[stratareg.h] <- li.h
  }
}
```

```
## Imputation
## So far, only deterministic imputation has been done
fit.residuals <- rep(0,length(y))
if(sum(!z)>0)
  {
    RegTech <- impute[1]
    Imputype <- impute[3]
    if(impute[2] == "across")
      {
        out.impute <- RegressionImputation(X,y,z,Xtype,RegTech,Imputype)
        fit.residuals <- out.impute$residuals
        y[!z] <- out.impute$yNA.imputed
        if(JKlin){
          Ximpute <- out.impute$Ximpute
          Ximpute.R <- Ximpute[z,]
          inv.z <- solve(t(Ximpute.R) %*% Ximpute.R)
          zcy <- (1-z)*y
          beta.yhat <- inv.iip %*% Xcalt.IIP %*% zcy
          Xcalbeta.yhat <- Xcal %*% beta.yhat
          for(h in 1:H){
            strataname.h <- stratanames[h]
            stratareg.h <- (stratareg==strataname.h)
            n.h1 <- sum(!z[stratareg.h])
            n.h2 <- sum(stratareg.h)
            n.h1 <- n.h2
            iip.h <- iip[stratareg.h]
            nd.h1 <- n.h1*iip.h
            nd.h2 <- n.h2*iip.h
            z.h <- z[stratareg.h]
            y.h <- y[stratareg.h]
            res.h <- fit.residuals[stratareg.h]
            zy.h <- z.h*y.h
            zcy.h <- zcy[stratareg.h]
            Ximpute.h <- Ximpute[stratareg.h,]
            Xcal.h <- Xcal[stratareg.h,]
            temp1 <- inv.z %*% t(Ximpute) %*% ((1-z)*iip)
            li.h <- n.h2*z.h*res.h * (Ximpute.h %*% temp1)
            li.h <- li.h + nd.h1*(1-z.h)*y.h - sum(iip.h*(1-z.h)*y.h)
            ##
            Xcalbeta.yhat.h <- as.matrix(Xcalbeta.yhat[stratareg.h])
            Xcalt.iip.h <- as.vector(t(Xcal.h) %*% iip.h)
            Xcal.inv.iip.h <- Xcal.inv.iip[stratareg.h,]
            Xcal.inv.iip.vec1.h <- as.matrix(Xcal.inv.iip.vec1[stratareg
                .h,])
            li.h <- li.h + sum(Xcalt.iip.h*beta.yhat) - nd.h2*Xcalbeta.
                yhat.h
            ##
            li.h <- li.h + nd.h2*(zcy.h-Xcalbeta.yhat.h)*Xcal.inv.iip.
                vec1.h
            ##
            temp1 <- t((iip*(1-z))*Ximpute) %*% Xcal
            temp2 <- inv.z %*% temp1 %*% inv.iip %*% vec1
            li.h <- li.h + n.h2*z.h*res.h * (Ximpute.h %*% temp2)
            li[stratareg.h] <- li[stratareg.h] + li.h
          }
        }
      }
```

```
      else
        {
          fit.residuals <- rep(NA, sum(z))
          for(h in 1:H)
            {
              strataname.h <- stratanames[h]
              stratareg.h <- (stratareg==strataname.h)
              Xh <- X[stratareg.h,]
              yh <- y[stratareg.h]
              zh <- z[stratareg.h]
              out.impute.h <- RegressionImputation(Xh,yh,zh,Xtype,
                                                   RegTech,Imputype)
              y[(!z) & stratareg.h] <- out.impute.h$yNA.imputed.h
              fit.residuals[z & stratareg.h] <- out.impute.h$residuals
              ## Can be implemented stratumwise
            }
        }
    }

  ## End
  out$theta.hat <- sum(as.vector(w)*y)
  out$z <- z
  out$y.imputed <- y
  out$residuals <- fit.residuals
  if(JKlin){
    out$eif <- as.vector(li)
    var.JKlin <- 0
    for(h in 1:H){
      strataname.h <- stratanames[h]
      stratareg.h <- (stratareg==strataname.h)
      n.h <- sum(stratareg.h)
      fpc.h <- fpc[h]
      var.JKlin <- var.JKlin + sum(li[stratareg.h]^2)/n.h/(n.h-1)*fpc.h
    }
    out$sdtheta.hat <- sqrt(var.JKlin)
  }
  return(out)
}
```

Appendix E

Other R function

Program E.1: Other R functions

```
RegressionImputation <- function(X,y,z,Xtype,RegTech,Imputype) {
## Notation
  ## n = nbr of sampled units
  ## q = nbr of auxiliary variables

  ## Input
  ## X = matrix of auxiliary variables organized columnwise (n rows, q
      colums)
  ## y = vector of responses values
  ## z = vector of logical indicator for respondent (z=FALSE if
      nonrespondent)
  ## Xtype = character vector (of size q) giving the type of
  ##     auxiliary variables: "c"=continuous, "f"=factor
  ## RegTech = imputation tehnique: "LS" for standard least squares
      regression,
  ##     "RH" for robust regression with the Huber loss
  ## Imputype = "deterministic" or "stochastic"

  ## Output
  ## yNA.imputed = imputed values
  ## Ximpute = matrix used for imputation by linear regression
  ## residuals = vector of regression residuals

  ## Start
  y[!z] <- 0
  out <- NULL

  if(RegTech=="LS")
    {
      outfit <- lm(y ~ ., data=X, weights=as.numeric(z),x=T)
    }
    else
      {
        outfit <- rlm(y ~ ., data=X, weights=as.numeric(z),x.ret=T)
```

```
      }
  yNA.imputed <- outfit$fitted.values[!z]
  if(Imputype == "stochastic"){
    yNA.imputed <- yNA.imputed + sample(outfit$residuals[!!z], length(yNA.
        imputed),
                                        replace=T)
  }

  ## End
  out$yNA.imputed <- yNA.imputed
  out$Ximpute <- outfit$x
  out$residuals <- outfit$residuals
  out$coefficients <- outfit$coefficients
  out$fit <- outfit
  return(out)
}
```

References

Canty, A. J. and Davison, A. C. (1999): Resampling-based variance estimation for labour force surveys. *The Statistician* **48**, 379–391.

Davison, A. C. and Hinkley, D. V. (1997): *Bootstrap Methods and Their Application*. Cambridge: Cambridge University Press.

Davison, A. C. and Skinner, C. J. (2001): Proposed notation for DACSEIS reports. Technical report, DACSEIS. Technical report.

Deville, J. C. and Särndal, C. E. (1992): Calibration estimators in survey sampling. *Journal of the American Statistical Association* **87**, 376–382.

McCarthy, P. J. (1969): Pseudo-replication: Half samples. *Review of the Interational Statistics Institute* **37**, 239–264.

Miller, R. G. (1974): The jackknife - A review. *Biometrika* **61**, 1–15.

Presnell, B. and Booth, J. G. (1994): Resampling methods for sample surveys. Technical Report 470, Department of Statistics, University of Florida, Gainesville.

Quenouille, M. H. (1949): Approximate tests of correlation in time-series. *Journal of the Royal Statistical Society, Series B* **11**, 68–84.

Rao, J. N. K. and Shao, J. (1992): Jackknife variance estimation with survey data under hot deck imputation. *Biometrika* **79**, 811–822.

Shao, J., Chen, Y. and Chen, Y. (1998): Balanced repeated replication for stratified multistage survey data under imputation. *Journal of the American Statistical Association* **93**, 819–831.

Shao, J. and Sitter, R. R. (1996): Bootstrap for imputed survey data. *Journal of the American Statistical Association* **91**, 1278–1288.

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

Shao, J. and Wu, C. F. J. (1989): A general theory for jackknife variance estimation. *Annals of Statistics* **17**, 1176–1197.

Tukey, J. W. (1958): Bias and confidence in not quite large samples (abstract). *Annals of Mathematical Statistics* **29**, 614.