

Package ‘bootsurv’

August 28, 2026

Title Bootstrap Methods for Complete Survey Data

Version 0.1.0

Depends R (>= 3.5.0)

Description Bootstrap resampling methods have been widely studied in the context of survey data. This package implements various bootstrap resampling techniques tailored for survey data, with a focus on stratified simple random sampling and stratified two-stage cluster sampling. It provides tools for precise and consistent bootstrap variance estimation for population totals, means, and quartiles. Additionally, it enables easy generation of bootstrap samples for in-depth analysis.

License GPL-3

Imports MASS, stats

Encoding UTF-8

LazyData true

Config/roxygen2/version 8.1.0

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author Zeinab Mashreghi [aut, cre] (ORCID:
<<https://orcid.org/0000-0003-0563-0792>>)

Maintainer Zeinab Mashreghi <z.mashreghi@uwinnipeg.ca>

Repository CRAN

Date/Publication 2026-08-28 07:01:39 UTC

Contents

boot.replicates	2
boot.twostage	5
boot.weights.stsrs	9
bootsurv.data	12
direct.boot.stsrs	13
pseudopop.boot.stsrs	15

boot.replicates	<i>Bootstrap Replicate Weights from bootsurv Output</i>
-----------------	---

Description

The function `boot.replicates` turns the output of supported bootstrap functions in the package into a matrix of bootstrap replicate weights, with one row per unit of the original sample and one column per bootstrap replicate. This is the form in which replicate weights are stored on public-use survey files and supplied to survey packages.

Usage

```
boot.replicates(object)
```

Arguments

object	The output of <code>boot.twostage()</code> , <code>direct.boot.stsrs()</code> , <code>boot.weights.stsrs()</code> or <code>pseudopop.boot.stsrs()</code>
--------	--

Details

With this matrix, bootstrap variances can be estimated for estimators beyond the totals, means and quartiles returned by the package. The estimator is computed once with each column of weights, and the resulting values are used to estimate its bootstrap variance. The rows of the matrix follow the order of the original sample, so the replicate weights can be applied directly to any variable defined on the original sample. This allows bootstrap variances to be estimated for estimators such as totals, ratios, calibrated estimators, and domain estimates.

For the bootstrap weight methods, the replicate weight is the bootstrap weight itself. For the methods that resample units, a unit's replicate weight is its design weight multiplied by the number of times it is selected in the bootstrap sample, and zero if it is not selected. In both cases, the bootstrap statistics returned by the package are reproduced exactly, so the estimator can be computed in the same way whether the units are resampled or only reweighted.

The method of Rao and Wu (1988) is a special case. It rescales the study-variable values toward the stratum mean. Under one-stage simple random sampling without replacement, this rescaling can still be represented by replicate weights, which reproduce estimators that are functions of totals. Under a one-stage stratified design, it cannot be represented by a per-unit replicate-weight matrix that can be applied to arbitrary variables, so the function stops. The bootstrap can instead be reproduced directly from `boot.sample` and `rescale.factor` by rescaling the values within each stratum, as shown in the examples.

When the method of Rao and Wu (1988) is used under the two-stage design of `boot.twostage()`, the function also stops. In this case, the rescaling is variable-dependent and cannot be converted into a fixed matrix of replicate weights. To reproduce the bootstrap for a new variable, the rescaling must be recomputed for that variable using its bootstrap cluster totals and the corresponding stratum means. This requires information from both the original sample and each bootstrap sample,

including the cluster sizes and stratum population sizes. The output does not contain all the information needed to perform this recomputation for an arbitrary new variable. The study variable is an exception: its rescaled values are already stored in `boot.sample`, so an estimator based on the study variable can be computed from the output.

Value

A numeric matrix of bootstrap replicate weights with one row per unit of the original sample, and one column per bootstrap replicate. For the pseudo-population methods the columns are ordered by pseudo-population, that is, all the bootstrap samples drawn from the first pseudo-population, followed by those drawn from the second, and so on.

References

Rao, J. N. K. and C. F. J. Wu (1988). Resampling inference with complex survey data. *Journal of the American Statistical Association* 83 (401), 231–241.

See Also

[boot.twostage\(\)](#), [direct.boot.stsrs\(\)](#), [boot.weights.stsrs\(\)](#) and [pseudopop.boot.stsrs\(\)](#), whose output is turned into bootstrap replicate weights by this function.

Examples

```
data(data_samp_srs)
population_size<- 6000
y<- data_samp_srs$study.variable

# Showing that the bootstrap variance estimators from the
# boot.weights.stsrs and boot.replicates functions are equal:
boot.CP<- boot.weights.stsrs(data_samp_srs, population_size, R = 20,
                             bootstrap.method = "Chipperfield.Preston")

rep.weights<- boot.replicates(boot.CP)
dim(rep.weights)

# The replicate weights reproduce the bootstrap variance of the package:
total.star<- apply(rep.weights, 2, function(w) sum(w * y))

all.equal(var(total.star), boot.CP$boot.var)

# Variance of a ratio of two study variables:
u<- data_samp_srs$study.variable^2
ratio.star<- apply(rep.weights, 2, function(w) sum(w * y)/sum(w * u))
var(ratio.star)

# Variance of a total within a sub-group (domain):
domain<- data_samp_srs$study.variable > 50
domain.star<- apply(rep.weights, 2, function(w) sum(w[domain] * y[domain]))
var(domain.star)

# Several study variables are estimated from the same set of replicate weights:
```

```

Y<- cbind(y = y, u = u)
total.several<- crossprod(rep.weights, Y)
apply(total.several, 2, var)

# A direct method, where the units are resampled. The estimator is written in
# the same way, and the replicate weights reproduce the bootstrap statistics:
boot.Efron<- direct.boot.stsrs(data_samp_srs, population_size, R = 20,
                             bootstrap.method = "Efron")

rep.weights.Efron<- boot.replicates(boot.Efron)
total.Efron<- apply(rep.weights.Efron, 2, function(w) sum(w * y))
all.equal(total.Efron, as.vector(boot.Efron$boot.statistic))

# A pseudo-population method (Booth, Butler and Hall, 1994). The columns of
# the replicate-weight matrix are ordered by pseudo-population, with R.samp
# columns per pseudo-population:
boot.BBH<- pseudopop.boot.stsrs(data_samp_srs, population_size,
                                R.pop = 3, R.samp = 5)

rep.weights.BBH<- boot.replicates(boot.BBH)
total.BBH<- apply(rep.weights.BBH, 2, function(w) sum(w * y))
all.equal(total.BBH, as.vector(t(boot.BBH$boot.statistic)))

# For a pseudo-population method the bootstrap variance is the average of the
# within-pseudo-population variances, not the variance pooled over all the
# replicates. Grouping the columns by pseudo-population reproduces boot.var,
# and the same recipe gives the variance of any other estimator, such as a
# ratio:
pop<- rep(1:3, each = 5)          # R.pop groups of R.samp columns
mean(tapply(total.BBH, pop, var)) # equal to boot.BBH$boot.var

ratio.star<- apply(rep.weights.BBH, 2, function(w) sum(w * y) / sum(w * u))
mean(tapply(ratio.star, pop, var)) # bootstrap variance of a ratio

# The rescaling method of Rao and Wu (1988). Under simple random sampling the
# replicate weights are available directly:
boot.RW<- direct.boot.stsrs(data_samp_srs, population_size, R = 20,
                             bootstrap.method = "Rao.Wu")
rep.weights.RW<- boot.replicates(boot.RW)
all.equal(apply(rep.weights.RW, 2, function(w) sum(w * y)),
          as.vector(boot.RW$boot.statistic))

# Under a stratified design the rescaling is reproduced directly from the
# bootstrap sample: within each stratum a study variable is rescaled by the
# factor in 'rescale.factor', and any other variable is rescaled the
# same way.
data(data_samp_stsrs)
boot.RW.st<- direct.boot.stsrs(data_samp_stsrs, rep(3000, 5), R = 20,
                               stratum = "stratum", bootstrap.method = "Rao.Wu")

y.st<- data_samp_stsrs$study.variable
C<- boot.RW.st$rescale.factor      # one rescaling factor per stratum
strata<- sort(unique(data_samp_stsrs$stratum))

```

```

ybar<- tapply(y.st, data_samp_stsrs$stratum, mean) # stratum means

total.RW.st<- sapply(boot.RW.st$boot.sample, function(one.rep){
  h<- match(one.rep$stratum, strata)
  y.star<- ybar[h] + C[h] * (y.st[one.rep$row.id] - ybar[h])
  sum(one.rep$bootstrap.weight * y.star)
})
all.equal(total.RW.st, as.vector(boot.RW.st$boot.statistic))

```

boot.twostage

Bootstrap Methods for Two-Stage Sampling Designs

Description

The function `boot.twostage` applies one of the following bootstrap methods on complete (full response) survey data selected under stratified two-stage cluster sampling with simple random sampling without replacement at both stages (SRSWOR/SRSWOR): Rao and Wu (1988), Rao, Wu and Yue (1992), the modified version of Sitter (1992, CJS) (see Chen, Haziza and Mashreghi, 2022), Funaoka, Saigo, Sitter and Toida (2006), Chauvet (2007) and Preston (2009). This function also applies the method of Rao, Wu and Yue (1992) on complete survey data selected under stratified two-stage cluster sampling with inclusion-probability-proportional-to-size sampling without replacement at the first stage and SRSWOR at the second stage (IPPSWOR/SRSWOR), or the method of Chauvet (2007) under conditional Poisson sampling at the first stage and SRSWOR at the second stage (CPS/SRSWOR).

Usage

```

boot.twostage(
  data,
  no.cluster,
  cluster.size,
  R,
  study.variable = "study.variable",
  stratum = NULL,
  cluster = "cluster",
  Pi1 = NULL,
  parameter = "total",
  bootstrap.method = "Rao.Wu.Yue",
  survey.design = "SRSWOR",
  population.size = NULL,
  boot.sample.size = NULL
)

```

Arguments

<code>data</code>	A vector, matrix or data frame containing the sample. For a matrix or data frame, the study-variable column is identified through the <code>study.variable</code> argument (by default, the column named <code>study.variable</code>), and the cluster and, under stratification, stratum columns through the <code>cluster</code> and <code>stratum</code> arguments. If an IPPS or CPS design is used at the first stage, the column of first-stage inclusion probabilities is identified through the <code>Pi1</code> argument.
<code>no.cluster</code>	A vector of the number of clusters within strata
<code>cluster.size</code>	The number of elements within the selected clusters within each stratum. The length of this vector must be the same as the number of all selected clusters in all strata.
<code>R</code>	The number of bootstrap replicates. For the Chauvet (2007) method, <code>R</code> is a vector with two values: (<code>R.pop</code> , <code>R.samp</code>) representing the number of pseudo-populations and the number of bootstrap samples drawn from each pseudo-population.
<code>study.variable</code>	A character string giving the name of the column in data that holds the study-variable values. The default is <code>"study.variable"</code> .
<code>stratum</code>	A character string giving the name of the column in data that identifies the strata. The default is <code>NULL</code> , in which case a single stratum is assumed. For a stratified design, set <code>stratum</code> to the name of the column in data that identifies the strata.
<code>cluster</code>	A character string giving the name of the column in data that identifies the clusters (PSUs). The default is <code>"cluster"</code> . At least two clusters (PSUs) are required in each stratum.
<code>Pi1</code>	A character string giving the name of the column in data that holds the first-stage inclusion probabilities, required when the first stage uses IPPS sampling (<code>survey.design = "IPPS"</code>) or CPS (<code>survey.design = "CPS"</code>). The default is <code>NULL</code> .
<code>parameter</code>	One of the following population parameters can be specified: <code>"total"</code> (population total), <code>"mean"</code> (population mean), <code>"quartile.25"</code> (population 1st quartile), <code>"quartile.50"</code> or <code>"median"</code> (population median) or <code>"quartile.75"</code> (population 3rd quartile). If the parameter of interest is the population mean or total, the HT-estimator is applied. If the parameter of interest is a population quartile, the estimator in Särndal, Swensson, and Wretman (1992, Chapter 5) is applied. The default is the population total.
<code>bootstrap.method</code>	One of the following bootstrap methods can be applied: <code>"Rao.Wu"</code> (Rao and Wu, 1988), <code>"Rao.Wu.Yue"</code> (Rao, Wu and Yue, 1992), <code>"Modified.Sitter"</code> (the modified version of Sitter 1992 discussed in Chen, Haziza and Mashreghi, 2022), <code>"Funaoka.etal"</code> (Funaoka, Saigo, Sitter and Toida, 2006), <code>"Chauvet"</code> (Chauvet, 2007) or <code>"Preston"</code> (Preston, 2009). The default is <code>"Rao.Wu.Yue"</code> .
<code>survey.design</code>	It can be either <code>"IPPS"</code> only if the method of Rao, Wu and Yue (1992) is applied or <code>"CPS"</code> only if the method of Chauvet (2007) is applied or <code>"SRSWOR"</code> . The default is <code>"SRSWOR"</code> .

population.size

A vector of population sizes, giving the total number of elements within each stratum. It is required for the Rao and Wu (1988) method and when the population mean is requested. The default is NULL.

boot.sample.size

A vector of bootstrap sample sizes within strata. The bootstrap sample size is required only for the method of Rao, Wu and Yue (1992). If it is not specified, the bootstrap sample size will be $nh-1$ within each stratum, where nh is the original sample size within stratum h .

Details

One-stage cluster sampling, where all units in a selected cluster are enumerated, is handled as the special case in which the within-cluster sample equals the cluster size.

Value

boot.statistic A vector of bootstrap statistics of size R

boot.var The bootstrap variance estimator of the estimator of the parameter of interest

boot.mean The average of the bootstrap estimator of the parameter of interest

boot.sample A list of results for each bootstrap replicate. This includes the bootstrap values (**bootstrap.sample** for the methods that resample units, or **study.variable** for the bootstrap weight methods, where the original values are kept and only the weights change), the original-sample row id of each selected unit (**row.id**), a cluster identifier (**cluster**), a stratum identifier (**stratum**), and the bootstrap weight (**bootstrap.weight**). Some methods return additional columns, such as the number of times a cluster is selected in the bootstrap sample (**No.selected.psu**).

rescale.factor The rescaling factor within each stratum. It is returned by the method of Rao and Wu (1988) only.

sample.size The total number of sampled elements across clusters within each stratum

References

- Chauvet, G. (2007). *Méthodes de bootstrap en population finie*. PhD thesis, École Nationale de Statistique et Analyse de l'Information, Bruz, France.
- Chen, S., Haziza, D. and Mashreghi, Z. (2022). A Comparison of Existing Bootstrap Algorithms for Multi-Stage Sampling Designs. *Stats* 5 (2), 521–537.
- Funaoka, F., Saigo, H., Sitter, R. R. and Toida, T. (2006). Bernoulli bootstrap for stratified multi-stage sampling. *Survey Methodology* 32 (2), 151–156.
- Rao, J. N. K. and C. F. J. Wu (1988). Resampling inference with complex survey data. *Journal of the American Statistical Association* 83 (401), 231–241.
- Rao, J. N. K., Wu, C. F. J. and Yue, K. (1992). Some recent work on resampling methods for complex surveys. *Survey Methodology* 18 (2), 209–217.
- Särndal, C.-E., Swensson, B. and Wretman, J. (1992). *Model-Assisted Survey Sampling*. New York: Springer.
- Sitter, R. R. (1992). Comparing three bootstrap methods for survey data. *The Canadian Journal of Statistics* 20 (2), 135–154.

Preston, J. (2009). Rescaled bootstrap for stratified multistage sampling. *Survey Methodology* 35 (2), 227–234.

Examples

```
R<- 20

data(data_samp_clust)
data(data_pop_clust)
no_cluster<- 200
cluster_size<- table(data_pop_clust$cluster)[
  unique(data_samp_clust$cluster)]

# The first stage sampling fraction is about 20% and the overall second
# stage sampling is about 15%.
# data_samp_clust is a sample taken from data_pop_clust available in
# the package.

boot.RWY<- boot.twostage(data_samp_clust, no_cluster, cluster_size, R)
boot.RWY$boot.var

boot.Pr<- boot.twostage(data_samp_clust, no_cluster, cluster_size, R,
  bootstrap.method="Preston")
boot.Pr$boot.var

boot.RWY.med<- boot.twostage(data_samp_clust, no_cluster, cluster_size,
  R, parameter="median")
boot.RWY.med$boot.var
boot.RWY.med$boot.sample[[5]]

boot.Ch<- boot.twostage(data_samp_clust, no_cluster, cluster_size,
  R=c(3, 10), bootstrap.method="Chauvet")
boot.Ch$boot.mean

data(data_samp_stclust)
data(data_pop_stclust)
# The first stage sampling fraction is about 20% and the overall second
# stage sampling is about 15%.
# data_samp_stclust is a sample taken from data_pop_stclust available
# in the package.

no_cluster_stclust<- c(100, 125, 65)
cluster_size_pop_st<- aggregate(data_pop_stclust$cluster,
  by=list(data_pop_stclust$stratum), table)[[2]]
L<- length(unique(data_samp_stclust$stratum))
cluster_size_st<- NULL
for(h in 1:L) cluster_size_st<- c(cluster_size_st,
  cluster_size_pop_st[[h]][unique(data_samp_stclust$cluster[
    data_samp_stclust$stratum==h])])

boot.RWY.st<- boot.twostage(data_samp_stclust, no_cluster_stclust,
  cluster_size_st, R, stratum="stratum")
boot.RWY.st$boot.statistic
```

```

# The output already stores the resampling (row.id and bootstrap.weight), so
# the bootstrap of another variable can be obtained by reusing it, without
# re-running the bootstrap and without boot.replicates. The new variable is
# defined on the original sample, and each replicate reuses the stored weights
# through row.id:
new.variable<- data_samp_clust$study.variable^2
total.new<- sapply(boot.RWY$boot.sample,
                   function(one.rep) sum(new.variable[one.rep$row.id] *
                                         one.rep$bootstrap.weight))
# Bootstrap variance of the new variable
var(total.new)

```

boot.weights.stsrs *Bootstrap Weight Methods for Survey Data*

Description

The function `boot.weights.stsrs` applies one of the following bootstrap weight methods on complete (full response) survey data selected under either simple random sampling without replacement (SRSWOR) or stratified simple random sampling without replacement (STSRSWOR): Rao, Wu and Yue (1992), Bertail and Combris (1997), Chipperfield and Preston (2007) and Beaumont and Patak (2012).

Usage

```

boot.weights.stsrs(
  data,
  population.size,
  R,
  study.variable = "study.variable",
  stratum = NULL,
  parameter = "total",
  bootstrap.method = "Rao.Wu.Yue",
  boot.sample.size = NULL,
  distribution.adjust = NULL,
  epsilon = NULL
)

```

Arguments

data	A vector, matrix or data frame containing the sample. For a matrix or data frame, the study-variable column is identified through the <code>study.variable</code> argument (by default, the column named <code>study.variable</code>), and, under STSRSWOR, the stratum column is identified through the <code>stratum</code> argument. When a vector is supplied, its values are taken as the study variable.
------	---

population.size	A vector of strata population sizes
R	The number of bootstrap replicates
study.variable	A character string giving the name of the column in data that holds the study-variable values. The default is "study.variable".
stratum	A character string giving the name of the column in data that identifies the strata under STSRSWOR. The default is NULL, in which case a single stratum is assumed (SRSWOR). For a stratified design, set stratum to the name of the column in data that identifies the strata.
parameter	One of the following population parameters can be specified: "total" (population total), "mean" (population mean), "quartile.25" (population 1st quartile), "quartile.50" or "median" (population median) or "quartile.75" (population 3rd quartile). If the parameter of interest is the population mean or total, the HT-estimator is applied. If the parameter of interest is a population quartile, the estimator in Särndal, Swensson, and Wretman (1992, Chapter 5) is applied. The default is the population total.
bootstrap.method	One of the following bootstrap methods can be applied: "Rao.Wu.Yue" (Rao, Wu and Yue, 1992), "Chipperfield.Preston" (Chipperfield and Preston, 2007), "Bertail.Combris" (Bertail and Combris, 1997) or "Beaumont.Patak" (Beaumont and Patak, 2012). The default is "Rao.Wu.Yue".
boot.sample.size	A vector of bootstrap sample sizes within strata only required for the method of Rao, Wu and Yue (1992). The length of this vector has to be the same as the number of strata. The default is NULL. If the method of Rao, Wu and Yue (1992) is applied and boot.sample.size is not specified, the bootstrap sample size will be $nh-1$ within each stratum, where nh is the original sample size within stratum h .
distribution.adjust	The default is NULL. A distribution should be specified for the method of Bertail and Combris (1997) and Beaumont and Patak (2012) to generate the bootstrap weight adjustments if epsilon is NULL. One of the following distributions can be used: "Normal", "Lognormal", "Exponential" or "Uniform".
epsilon	The default is NULL. If either Bertail and Combris (1997) or Beaumont and Patak (2012) is applied and distribution.adjust is NULL, a value must be given to epsilon so that Eq(5) in Beaumont and Patak (2012) can be used to generate the bootstrap weight adjustments.

Value

boot.statistic	A vector of bootstrap statistics
boot.var	The bootstrap variance estimator of the estimator of the parameter of interest
boot.mean	The average of the bootstrap estimator of the parameter of interest
boot.sample	A list of size R. Each element is a data frame containing the study-variable values (study.variable), the original-sample row id of each unit (row.id), a stratum identifier (stratum), the bootstrap weight (bootstrap.weight) and the bootstrap weight adjustment (weight.adjustment).
sample.size	The number of sampled units in each stratum.

References

- Beaumont, J.-F. and Patak, Z. (2012). On the generalized bootstrap for sample surveys with special attention to Poisson sampling. *International Statistical Review* 80 (1), 127–148.
- Bertail, P. and Combris, P. (1997). Bootstrap généralisé d'un sondage. *Annales d'économie et de statistique* 46, 49–83.
- Chipperfield, J. and Preston, J. (2007). Efficient bootstrap for business surveys. *Survey Methodology* 33 (2), 167–172.
- Rao, J. N. K., Wu, C. F. J. and Yue, K. (1992). Some recent work on resampling methods for complex surveys. *Survey Methodology* 18 (2), 209–217.
- Särndal, C.-E., Swensson, B. and Wretman, J. (1992). *Model-Assisted Survey Sampling*. New York: Springer.

Examples

```
R<- 20

data(data_samp_srs)
population_size<- 6000
# The sampling fraction is about 30%.
# data_samp_srs is a sample taken from data_pop available in the package.

boot.RWY<- boot.weights.stsrs(data_samp_srs, population_size, R)
boot.RWY$boot.var

boot.CP<- boot.weights.stsrs(data_samp_srs, population_size, R,
                             bootstrap.method="Chipperfield.Preston")
boot.CP$boot.var

boot.BP.med<- boot.weights.stsrs(data_samp_srs, population_size, R,
                                 parameter="median", bootstrap.method="Beaumont.Patak",
                                 distribution.adjust="Exponential")
boot.BP.med$boot.var
boot.BP.med$boot.sample[[5]]

data(data_samp_stsrs)
population_size_st<- c(4500, 6300, 3500, 2000, 1500)
# The overall sampling fraction is about 30%.
# data_samp_stsrs is a sample taken from data_pop_st available in the
# package.

boot.RWY.st<- boot.weights.stsrs(data_samp_stsrs, population_size_st, R,
                                 stratum="stratum")
boot.RWY.st$boot.var
boot.RWY.st$boot.statistic

# The output already stores the resampling (row.id and bootstrap.weight), so
# the bootstrap of another variable can be obtained by reusing it, without
# re-running the bootstrap and without boot.replicates(). The new variable is
# defined on the original sample, and each replicate reuses the stored weights
```

```
# through row.id:
new.variable<- data_samp_srs$study.variable^2
total.new<- sapply(boot.RWY$boot.sample,
  function(one.rep) sum(new.variable[one.rep$row.id] *
    one.rep$bootstrap.weight))
# Bootstrap variance of the new variable
var(total.new)
```

bootsurv.data

Simulated Populations and Samples in the bootsurv Package

Description

This package contains multiple datasets described below.

Datasets

data_pop This is a population of size 6,000. This dataset contains a column of the generated study variable, labeled as `study.variable`.

data_pop_st This dataset represents a population of size 17,800, divided into 5 strata. It includes a column for the generated study variable, labeled as `study.variable`, and a column identifying the strata, labeled as `stratum`. The subpopulation sizes within each stratum are as follows: 4,500, 6,300, 3,500, 2,000, and 1,500, respectively.

data_pop_clust This dataset represents a population consisting of 10,048 units distributed across 200 clusters. The number of units within each cluster was generated using a Poisson distribution with a mean of 50. It includes columns for the generated study variable, labeled as `study.variable`, and cluster identification, denoted as `cluster`.

data_pop_stclust This dataset represents a population with 14,511 units distributed across three strata, consisting of 100, 125, and 65 clusters, respectively. The number of units within each cluster was generated using a Poisson distribution with a mean of 50. It includes columns of the generated study variable, labeled as `study.variable`, stratum identification, labeled as `stratum`, and cluster identification within each stratum, labeled as `cluster`.

data_samp_srs This dataset comprises a sample of size 1,850, obtained through simple random sampling without replacement from the `data_pop` dataset.

data_samp_stsrs This dataset represents a sample of size 5,350 obtained through stratified simple random sampling without replacement from the stratified population `data_pop_st`. The sample consists of subsample sizes of 1,350, 1,900, 1,050, 600, and 450.

data_samp_clust This sample was drawn using a two-stage cluster sampling method, with simple random sampling without replacement applied at each stage. The sample is drawn from the `data_pop_clust` dataset. In the first stage, approximately 20% of clusters were selected. Subsequently, within each selected cluster, approximately 15% of units were sampled.

`data_samp_stclust` A stratified two-stage cluster sampling method is applied to draw this sample from the `data_pop_stclust` dataset. In each stratum, simple random sampling without replacement is applied at each stage. The first stage sampling fraction is approximately 20%, and the overall second stage sampling is approximately 15%.

<code>direct.boot.stsrs</code>	<i>Direct Bootstrap Methods for Survey Data</i>
--------------------------------	---

Description

The function `direct.boot.stsrs` applies one of the following bootstrap methods on complete (full response) survey data selected under either simple random sampling without replacement (SRSWOR) or stratified simple random sampling without replacement (STSRSWOR): Efron (1979), McCarthy and Snowden (1985), Rao and Wu (1988) and Sitter (1992, JASA).

Usage

```
direct.boot.stsrs(
  data,
  population.size,
  R,
  study.variable = "study.variable",
  stratum = NULL,
  parameter = "total",
  bootstrap.method = "Rao.Wu",
  boot.sample.size = NULL
)
```

Arguments

<code>data</code>	A vector, matrix or data frame containing the sample. For a matrix or data frame, the study-variable column is identified through the <code>study.variable</code> argument (by default, the column named <code>study.variable</code>), and, under STSRSWOR, the stratum column is identified through the <code>stratum</code> argument. When a vector is supplied, its values are taken as the study variable.
<code>population.size</code>	A vector of strata population sizes
<code>R</code>	The number of bootstrap replicates
<code>study.variable</code>	A character string giving the name of the column in data that holds the study-variable values. The default is <code>"study.variable"</code> .
<code>stratum</code>	A character string giving the name of the column in data that identifies the strata under STSRSWOR. The default is <code>NULL</code> , in which case a single stratum is assumed (SRSWOR). For a stratified design, set <code>stratum</code> to the name of the column in data that identifies the strata.

parameter	One of the following population parameters can be specified: "total" (population total), "mean" (population mean), "quartile.25" (population 1st quartile), "quartile.50" or "median" (population median) or "quartile.75" (population 3rd quartile). If the parameter of interest is the population mean or total, the HT-estimator is applied. If the parameter of interest is a population quartile, the estimator in Särndal, Swensson, and Wretman (1992, Chapter 5) is applied. The default is the population total.
bootstrap.method	One of the following bootstrap methods can be applied: "Efron" (Efron, 1979), "McCarthy.Snowden" (McCarthy and Snowden, 1985), "Rao.Wu" (Rao and Wu, 1988) or "Sitter.BMM" (Sitter, 1992). The default is "Rao.Wu".
boot.sample.size	If the method of Rao and Wu (1988) is applied, a vector of bootstrap sample sizes for each stratum may be specified. The length of this vector must match the number of strata. By default, if 'boot.sample.size' is not specified, the bootstrap sample size within each stratum will be 'nh-3', where 'nh' is the original sample size in stratum 'h'.

Value

boot.statistic	A vector of bootstrap statistics
boot.var	The bootstrap variance estimator of the estimator of the parameter of interest
boot.mean	The average of the bootstrap estimator of the parameter of interest
boot.sample	A list of size R. Each element is a data frame with four columns: the bootstrap values (bootstrap.sample; rescaled for the Rao and Wu (1988) method), the original-sample row id of each selected unit (row.id), a stratum identifier (stratum), and the bootstrap weight (bootstrap.weight).
rescale.factor	The per-stratum rescaling factor applied to the study variable. It equals 1 for every method except Rao and Wu (1988). For Rao and Wu, it can be used together with row.id and the original sample to reproduce the bootstrap rescaling for another variable or estimator.
sample.size	The number of sampled units in each stratum.

References

- Efron, B. (1979). Bootstrap methods: another look at the jackknife. *The Annals of Statistics* 7 (1), 1–26.
- McCarthy, P. J. and C. B. Snowden (1985). The bootstrap and finite population sampling. *Vital and Health Statistics, Series 2, No. 95*. DHHS Publication No. (PHS) 85–1369. Public Health Service. Washington. U.S. Government Printing Office.
- Rao, J. N. K. and C. F. J. Wu (1988). Resampling inference with complex survey data. *Journal of the American Statistical Association* 83 (401), 231–241.
- Särndal, C.-E., Swensson, B. and Wretman, J. (1992). *Model-Assisted Survey Sampling*. New York: Springer.
- Sitter, R. R. (1992). A resampling procedure for complex survey data. *Journal of the American Statistical Association* 87 (419), 755–765.

Examples

```

R<- 20

data(data_samp_srs)
population_size<- 6000
# The sampling fraction is about 30%.
# data_samp_srs is a sample taken from data_pop available in the package.

boot.RW<- direct.boot.stsrs(data_samp_srs, population_size, R)
boot.RW$boot.var

boot.Efron<- direct.boot.stsrs(data_samp_srs, population_size, R,
                              parameter="total", bootstrap.method="Efron")
boot.Efron$boot.var

boot.RW.med<- direct.boot.stsrs(data_samp_srs, population_size, R,
                               parameter="median")
boot.RW.med$boot.var

data(data_samp_stsrs)
population_size_st<- c(4500, 6300, 3500, 2000, 1500)
# The overall sampling fraction is about 30%.
# data_samp_stsrs is a sample taken from data_pop_st available in the
# package.

boot.RW.st<- direct.boot.stsrs(data_samp_stsrs, population_size_st, R,
                              stratum="stratum", parameter="total",
                              bootstrap.method="Rao.Wu")
boot.RW.st$boot.statistic

# The output already stores the resampling (row.id and bootstrap.weight), so
# the bootstrap of another variable can be obtained by reusing it, without
# re-running the bootstrap and without boot.replicates(). The new variable is
# defined on the original sample, and each replicate reuses the stored weights
# through row.id:
new.variable<- data_samp_srs$study.variable^2
total.new<- sapply(boot.Efron$boot.sample,
                  function(one.rep) sum(new.variable[one.rep$row.id] *
                                         one.rep$bootstrap.weight))

# Bootstrap variance of the new variable:
var(total.new)

```

Description

The function `pseudopop.boot.stsrs` applies one of the following pseudo-population bootstrap methods on complete (full response) survey data selected under either simple random sampling without replacement (SRSWOR) or stratified simple random sampling without replacement (STSR-SWOR): Bickel and Freedman (1984), Chao and Lo (1985), Sitter (1992, CJS), Booth, Butler and Hall (1994) and Chao and Lo (1994).

Usage

```
pseudopop.boot.stsrs(
  data,
  population.size,
  R.pop,
  R.samp,
  study.variable = "study.variable",
  stratum = NULL,
  parameter = "total",
  bootstrap.method = "Booth.Butler.Hall"
)
```

Arguments

<code>data</code>	A vector, matrix or data frame containing the sample. For a matrix or data frame, the study-variable column is identified through the <code>study.variable</code> argument (by default, the column named <code>study.variable</code>), and, under STSR-SWOR, the stratum column is identified through the <code>stratum</code> argument. When a vector is supplied, its values are taken as the study variable.
<code>population.size</code>	A vector of strata population sizes
<code>R.pop</code>	The number of bootstrap replicates to create bootstrap pseudo-populations
<code>R.samp</code>	The number of bootstrap replicates to draw bootstrap samples from each bootstrap pseudo-population
<code>study.variable</code>	A character string giving the name of the column in data that holds the study-variable values. The default is <code>"study.variable"</code> .
<code>stratum</code>	A character string giving the name of the column in data that identifies the strata under STSR-SWOR. The default is <code>NULL</code> , in which case a single stratum is assumed (SRSWOR). For a stratified design, set <code>stratum</code> to the name of the column in data that identifies the strata.
<code>parameter</code>	One of the following population parameters can be specified: <code>"total"</code> (population total), <code>"mean"</code> (population mean), <code>"quartile.25"</code> (population 1st quartile), <code>"quartile.50"</code> or <code>"median"</code> (population median) or <code>"quartile.75"</code> (population 3rd quartile). If the parameter of interest is the population mean or total, the HT-estimator is applied. If the parameter of interest is a population quartile, the estimator in Särndal, Swensson, and Wretman (1992, Chapter 5) is applied. The default is the population total.

bootstrap.method

One of the following bootstrap methods can be applied: "Bickel.Freedman" (Bickel and Freedman, 1984), "Booth.Butler.Hall" (Booth, Butler and Hall, 1994), "Chao.Lo.1985" (Chao and Lo, 1985), "Sitter.BW0" (Sitter, 1992) or "Chao.Lo.1994" (Chao and Lo, 1994). The default is "Booth.Butler.Hall".

Value

`boot.statistic` A vector of bootstrap statistics

boot.parameter A vector of bootstrap parameters computed from each generated bootstrap pseudo-population

boot.var The bootstrap variance estimator of the estimator of the parameter of interest

boot.mean The average of the bootstrap estimator of the parameter of interest

boot.sample A list of size R.pop. Each element contains a list of results for the bootstrap samples from the corresponding generated bootstrap pseudo-population. Each bootstrap sample includes four columns: the bootstrap values (bootstrap.sample), the original-sample row id of each selected unit (row.id), a stratum identifier (stratum), and the bootstrap weight (bootstrap.weight).

`sample.size` The number of sampled units in each stratum.

References

- Bickel, P. J. and Freedman, D. A. (1984). Asymptotic normality and the bootstrap in stratified sampling. *The Annals of Statistics* 12, 470–82.
- Booth, J. G., Butler, R. W. and Hall, P. (1994). Bootstrap methods for finite populations. *Journal of the American Statistical Association* 89 (428), 1282–1289.
- Chao, M. T. and Lo, S.-H. (1985). A bootstrap method for finite population. *Sankhya: The Indian Journal of Statistics, Series A* 47, 399–405.
- Chao, M. T. and Lo, S.-H. (1994). Maximum likelihood summary and the bootstrap method in structured finite populations. *Statistica Sinica* 4 (2), 389–406.
- Särndal, C.-E., Swensson, B. and Wretman, J. (1992). *Model-Assisted Survey Sampling*. New York: Springer.
- Sitter, R. R. (1992). Comparing three bootstrap methods for survey data. *The Canadian Journal of Statistics* 20 (2), 135–154.

Examples

[illegible]

```

boot.BF<- pseudopop.boot.stsrs(data_samp_srs, population_size, R.pop, R.samp,
                               bootstrap.method="Bickel.Freedman")
boot.BF$boot.var

boot.Sitter.med<- pseudopop.boot.stsrs(data_samp_srs, population_size, R.pop,
                                       R.samp, parameter="median", bootstrap.method="Sitter.BW0")
boot.Sitter.med$boot.var
boot.Sitter.med$boot.sample[[2]][[5]]

data(data_samp_stsrs)
population_size_st<- c(4500, 6300, 3500, 2000, 1500)
# The overall sampling fraction is about 30%.
# data_samp_stsrs is a sample taken from data_pop_st available in the
# package.

boot.Booth.st<- pseudopop.boot.stsrs(data_samp_stsrs, population_size_st,
                                     R.pop, R.samp, stratum="stratum")
boot.Booth.st$boot.statistic

# The output already stores the resampling (row.id and bootstrap.weight), so
# the bootstrap of another variable can be obtained by reusing it, without
# re-running the bootstrap and without boot.replicates(). For a pseudo-population
# method the variance is the average of the within-pseudo-population variances:
new.variable<- data_samp_srs$study.variable^2
total.new<- lapply(boot.Booth$boot.sample,
                  function(one.pop) sapply(one.pop,
                                             function(one.rep) sum(new.variable[one.rep$row.id] *
                                                                    one.rep$bootstrap.weight)))

# Bootstrap variance of the new variable:
mean(sapply(total.new, var))

```

Index

`boot.replicates`, [2](#)
`boot.twostage`, [5](#)
`boot.twostage()`, [2](#), [3](#)
`boot.weights.stsrs`, [9](#)
`boot.weights.stsrs()`, [2](#), [3](#)
`bootsurv.data`, [12](#)

`data_pop (bootsurv.data)`, [12](#)
`data_pop_clust (bootsurv.data)`, [12](#)
`data_pop_st (bootsurv.data)`, [12](#)
`data_pop_stclust (bootsurv.data)`, [12](#)
`data_samp_clust (bootsurv.data)`, [12](#)
`data_samp_srs (bootsurv.data)`, [12](#)
`data_samp_stclust (bootsurv.data)`, [12](#)
`data_samp_stsrs (bootsurv.data)`, [12](#)
`direct.boot.stsrs`, [13](#)
`direct.boot.stsrs()`, [2](#), [3](#)

`pseudopop.boot.stsrs`, [15](#)
`pseudopop.boot.stsrs()`, [2](#), [3](#)