Package ‘bbw’

January 16, 2025

Type Package

Title Blocked Weighted Bootstrap

Version 0.3.0

Description The blocked weighted bootstrap (BBW) is an estimation technique for use with data from two-stage cluster sampled surveys in which either prior weighting (e.g. population-proportional sampling or PPS as used in Standardized Monitoring and Assessment of Relief and Transitions or SMART surveys) or posterior weighting (e.g. as used in rapid assessment method or RAM and simple spatial sampling method or S3M surveys) is implemented. See Cameron et al (2008) <doi:10.1162/rest.90.3.414> for application of bootstrap to cluster samples. See Aaron et al (2016) <doi:10.1371/journal.pone.0163176> and Aaron et al (2016) <doi:10.1371/journal.pone.0162462> for application of the blocked weighted bootstrap to estimate indicators from two-stage cluster sampled surveys.

License GPL-3

Depends R (>= 4.1.0)

Imports car, cli, doParallel, foreach, methods, parallel, parallelly, stats, stringr, withr

Suggests knitr, rmarkdown, testthat, spelling, covr

Encoding UTF-8

Language en-GB

LazyData true

RoxygenNote 7.3.2

URL `https://github.com/rapidsurveys/bbw`, `https://rapidsurveys.io/bbw/`

BugReports `https://github.com/rapidsurveys/bbw/issues`

VignetteBuilder knitr

NeedsCompilation no

Author Mark Myatt [aut, cph],
Ernest Guevarra [aut, cre, cph]
(<`https://orcid.org/0000-0002-4887-4415`>)

Maintainer Ernest Guevarra <ernestgmd@gmail.com>

Repository CRAN

Date/Publication 2025-01-16 09:00:06 UTC

Contents

`bootBW` *Blocked Weighted Bootstrap*

Description

The **blocked weighted bootstrap (BBW)** is an estimation technique for use with data from two-stage cluster sampled surveys in which either prior weighting (e.g. **population proportional sampling** or **PPS** as used in **SMART** surveys) or posterior weighting (e.g. as used in **RAM** and **S3M** surveys).

Usage

```
bootBW(x, w, statistic, params, outputColumns = params, replicates = 400)
```

Arguments

`x`	A `data.frame()` with primary sampling unit (PSU) in variable named `psu` and at least one other variable containing data for estimation.
`w`	A `data.frame()` with primary sampling unit (PSU) in variable named `psu` and survey weights (i.e. PSU population) in variable named `pop`.
`statistic`	Am estimator function operating on variables in `x` containing data for estimation. The functions `bootClassic()` and `bootPROBIT()` are examples.
`params`	Parameters specified as names of columns in `x` that are to be passed to the function specified in `statistic`.

`outputColumns` Names to be used for columns in output `data.frame()`. Default to names specified in `params`.

`replicates` Number of bootstrap replicates to be performed. Default is 400.

Value

A `data.frame()` with:

- number of columns equal to length of `outputColumns`;
- number of rows equal to number of `replicates`; and,‘
- names equal to `outputColumns`.‘

Examples

```
# Example call to bootBW function using RAM-OP test data:

bootBW(
  x = indicatorsHH, w = villageData, statistic = bootClassic,
  params = "anc1", outputColumns = "anc1", replicates = 9
)

# Example estimate with 95% CI:
#quantile(bootP, probs = c(0.500, 0.025, 0.975), na.rm = TRUE)
```

`bootClassic` *Simple proportion statistics function for bootstrap estimation*

Description

Simple proportion statistics function for bootstrap estimation

Usage

```
bootClassic(x, params)
```

Arguments

`x` A data frame with **primary sampling unit (PSU)** in column named `psu` and with data column/s containing the binary variable/s (0/1) of interest with column names corresponding to `params` values

`params` A vector of column names corresponding to the binary variables of interest contained in `x`

Value

A numeric vector of the mean of each binary variable of interest with length equal to `length(params)`

Examples

```
# Example call to bootClassic function
sampled_clusters <- boot_bw_sample_clusters(
  x = indicatorsHH, w = boot_bw_weight(villageData)
)

boot <- boot_bw_sample_within_clusters(sampled_clusters)

bootClassic(boot, "anc1")
```

---

bootPROBIT *PROBIT statistics function for bootstrap estimation*

Description

PROBIT statistics function for bootstrap estimation

Usage

```
bootPROBIT(x, params, threshold = THRESHOLD)
```

Arguments

x	A data frame with **primary sampling unit (PSU)** in column named psu and with data column/s containing the continuous variable/s of interest with column names corresponding to params values
params	A vector of column names corresponding to the continuous variables of interest contained in x
threshold	cut-off value for continuous variable to differentiate case and non-case

Value

A numeric vector of the PROBIT estimate of each continuous variable of interest with length equal to length(params)

Examples

```
# Example call to bootBW function:
sampled_clusters <- boot_bw_sample_clusters(
  x = indicatorsCH1, w = boot_bw_weight(villageData)
)

boot <- boot_bw_sample_within_clusters(sampled_clusters)

bootPROBIT(x = boot,
           params = "muac1",
           threshold = 115)
```

boot_bw *Blocked Weighted Bootstrap - vectorised and parallel*

Description

This set of functions is an alternative to the bootBW() function. This set attempts to make the blocked weighted bootstrap algorithm more efficient through vectorisation and use of parallelisation techniques. The function syntax has been kept consistent with bootBW() for ease of transition. A more in depth discussion of the efficiencies gained from this alternative function is discussed here.

Usage

```
boot_bw(
  x,
  w,
  statistic,
  params,
  outputColumns = params,
  replicates = 400,
  strata = NULL,
  parallel = FALSE,
  cores = parallelly::availableCores(omit = 1)
)

boot_bw_parallel(
  x,
  w,
  statistic,
  params,
  outputColumns = params,
  replicates = 400,
  strata = NULL,
  cores = parallelly::availableCores(omit = 1)
)

boot_bw_sequential(
  x,
  w,
  statistic,
  params,
  outputColumns = params,
  replicates = 400,
  strata = NULL
)

boot_bw_weight(w)
```

```
boot_bw_sample_clusters(x, w, index = FALSE)

boot_bw_sample_within_clusters(cluster_df)
```

Arguments

`x` A `data.frame()` with primary sampling unit (PSU) in variable named `psu` and at least one other variable containing data for estimation.

`w` A `data.frame()` with primary sampling unit (PSU) in variable named `psu` and survey weights (i.e. PSU population) in variable named pop.

`statistic` Am estimator function operating on variables in `x` containing data for estimation. The functions `bootClassic()` and `bootPROBIT()` are examples.

`params` Parameters specified as names of columns in `x` that are to be passed to the function specified in `statistic`.

`outputColumns` Names to be used for columns in output `data.frame()`. Default to names specified in `params`.

`replicates` Number of bootstrap replicates to be performed. Default is 400.

`strata` A character value for name of variable in `x` providing information on how `x` is grouped such that resampling is performed for each group. Default to NULL for no grouping and resampling is performed for full data.

`parallel` Logical. Should resampling be done in parallel? Default to FALSE.

`cores` The number of computer cores to use or number of child processes to be run simultaneously. Default to one less than the available number of cores on current machine.

`index` Logical. Should index values be returned or a list of `data.frame()`s. Default to FALSE.

`cluster_df` A list of `data.frame()`s for selected clusters.

Value

For `boot_bw()`, a `data.frame()` with number of columns equal to length of `outputColumns`; number of rows equal to number of `replicates`; and, names of variables equal to values of `outputColumns`. For `boot_bw_weight()`, A `data.frame()` based on `w` with two additional variables for `weight` and `cumWeight`. For `boot_bw_sample_clusters()`, either a vector of integers corresponding to the primary sampling unit (psu) identifier of the selected clusters (when `index = TRUE`) or a list of `data.frame()`s corresponding to the data for the selected clusters (when `index = FALSE`). For `boot_bw_sample_within_clusters()`, a matrix similar in structure to `x` of resampled data from each selected cluster.

Examples

```
boot_bw(
  x = indicatorsHH, w = villageData, statistic = bootClassic,
  params = "anc1", replicates = 9, parallel = TRUE
)
```

`boot_bw_estimate` *Estimate median and confidence intervals from bootstrap replicates*

Description

Estimate median and confidence intervals from bootstrap replicates

Usage

```
boot_bw_estimate(boot_df)
```

Arguments

`boot_df` A `data.frame()` or a list of `data.frame()`s of bootstrap replicates with columns for each indicator to estimate. This is produced by a call to `boot_bw()`.

Value

A `data.frame()` with rows equal to the number of columns of `boot_df` and 4 columns for **indicator**, **estimate**, **95% lower confidence limit**, and **95% upper confidence limit**.

Examples

```
boot_df <- boot_bw(
  x = indicatorsHH, w = villageData, statistic = bootClassic,
  params = "anc1", parallel = TRUE, replicates = 9
)

boot_bw_estimate(boot_df)
```

`estimate_total` *Post-stratification analysis*

Description

Post-stratification analysis

Usage

```
estimate_total(est_df, pop_df, strata)
```

Arguments

- est_df: A data.frame() of stratified indicator estimates to get overall estimates of. est_df should have a variable named est for the values of the indicator estimate, a variable named strata for information on the stratification or grouping of the estimates, and a variable named se for the standard errors for the values of the indicator estimate. This is usually produced via a call to boot_bw_estimate().
- pop_df: A data.frame() with at least two variables: strata for the stratification/grouping information that matches strata in est_df and pop for information on population for the given strata.
- strata: A character value of the variable name in est_df that corresponds to the strata values to match with values in pop_df

Value

A vector of values for the overall estimate, overall 95% lower confidence limit, and overall 95% upper confidence limit for each of the strata in est_df.

Examples

```
est_df <- boot_bw(
  x = indicatorsHH, w = villageData, statistic = bootClassic,
  params = "anc1", strata = "region", replicates = 9, parallel = TRUE
) |>
  boot_bw_estimate()

## Add population ----
pop_df <- somalia_population |>
  subset(select = c(region, total))

names(pop_df) <- c("strata", "pop")

estimate_total(est_df, pop_df, strata = "region")
```

indicatorsCH1 *Child Morbidity, Health Service Coverage, Anthropometry*

Description

Child indicators on morbidity, health service coverage and anthropometry calculated from survey data collected in survey conducted in 4 districts from 3 regions in Somalia.

Usage

```
indicatorsCH1
```

Format

A data frame with 16 columns and 3090 rows.

Variable	Description
`region`	Region in Somalia from which the cluster belongs to
`district`	District in Somalia from which the cluster belongs to
`psu`	The PSU identifier. This must use the same coding system used to identify the PSUs that is used in the indicators
`mID`	The mother identifier
`cID`	The child identifier
`ch1`	Diarrhoea in the past 2 weeks (0/1)
`ch2`	Fever in the past 2 weeks (0/1)
`ch3`	Cough in the past 2 weeks (0/1)
`ch4`	Immunisation card (0/1)
`ch5`	BCG immunisation (0/1)
`ch6`	Vitamin A coverage in the past month (0/1)
`ch7`	Anti-helminth coverage in the past month (0/1)
`sex`	Sex of child
`muac1`	Mid-upper arm circumference in mm
`muac2`	Mid-upper arm circumference in mm
`oedema`	Oedema (0/1)

Source

Mother and child health and nutrition survey in 3 regions of Somalia

Examples

```
indicatorsCH1
```

`indicatorsCH2` *Infant and Child Feeding Index*

Description

Infant and young child feeding indicators using the infant and child feeding index (ICFI) by Arimond and Ruel. Calculated from survey data collected in survey conducted in 4 districts from 3 regions in Somalia.

Usage

```
indicatorsCH2
```

Format

A data frame with 15 columns and 2083 rows.

Variable	Description
`region`	Region in Somalia from which the cluster belongs to
`district`	District in Somalia from which the cluster belongs to
`psu`	The PSU identifier. This must use the same coding system used to identify the PSUs that is used in the indicators
`mID`	The mother identifier
`cID`	The child identifier
`ebf`	Exclusive breastfeeding (0/1)
`cbf`	Continued breastfeeding (0/1)
`ddd`	Dietary diversity (0/1)
`mfd`	Meal frequency (0/1)
`icfi`	Infant and child feeding index (from 0 to 6)
`iycf`	Good IYCF
`icfiProp`	Good ICFI
`age`	Child's age
`bf`	Child is breastfeeding (0/1)
`bfStop`	Age in months child stopped breastfeeding

Source

Mother and child health and nutrition survey in 3 regions of Somalia

Examples

```
indicatorsCH2
```

`indicatorsHH` *Mother Indicators Dataset*

Description

Mother indicators for health and nutrition calculated from survey data collected in survey conducted in 4 districts from 3 regions in Somalia.

Usage

```
indicatorsHH
```

Format

A data frame with 26 columns and 2136 rows:

Variable	Description
`region`	Region in Somalia from which the cluster belongs to
`district`	District in Somalia from which the cluster belongs to
`psu`	The PSU identifier. This must use the same coding system used to identify the PSUs that is used in the indicators
`mID`	The mother identifier
`mMUAC`	Mothers with mid-upper arm circumference < 230 mm (0/1)
`anc1`	At least 1 antenatal care visit with a trained health professional (0/1)
`anc2`	At least 4 antenatal care visits with any service provider (0/1)
`anc3`	FeFol coverage (0/1)
`anc4`	Vitamin A coverage (0/1)
`wash1`	Improved sources of drinking water (0/1)
`wash2`	Improved sources of other water (0/1)
`wash3`	Probable safe drinking water (0/1)
`wash4`	Number of litres of water collected in a day
`wash5`	Improved toilet facilities (0/1)
`wash6`	Human waste disposal practices / behaviour (0/1)
`wash7a`	Handwashing score (from 0 to 5)
`wash7b`	Handwashing score of 5 (0/1)
`hhs1`	Household hunger score (from 0 to 6)
`hhs2`	Little or no hunger (0/1)
`hhs3`	Moderate hunger (0/1)
`hhs4`	Severe hunger (0/1)
`mfg`	Mother's dietary diversity score
`pVitA`	Plant-based vitamin A-rich foods (0/1)
`aVitA`	Animal-based vitamin A-rich foods (0/1)
`xVitA`	Any vitamin A-rich foods (0/1)
`iron`	Iron-rich foods (0/1)

Source

Mother and child health and nutrition survey in 3 regions of Somalia

Examples

```
indicatorsHH
```

`recode` *Recode*

Description

Utility function that recodes variables based on user recode specifications. Handles both numeric or factor variables.

Usage

```
recode(var, recodes, afr, anr = TRUE, levels)
```

Arguments

`var` Variable to recode

`recodes` Character string of recode specifications:

- Recode specifications in a character string separated by semicolons of the form `input=output` as in: `"1=1;2=1;3:6=2;else=NA"`

```
\item If an input value satisfies more than one specification, then the
first (reading from left to right) is applied

\item If no specification is satisfied, then the input value is carried
over to the result unchanged

\item \code{NA} is allowed on both input and output

\item The following recode specifications are supported:

    \tabular{lll}{
    \strong{Specification} \tab \strong{Example}       \tab \strong{Notes}
    Single values       \tab \code{9=NA}           \tab
    Set of values       \tab \code{c(1,2,5)=1}      \tab The left-hand-side is any
                    \tab \code{seq(1,9,2)='odd'}  \tab
                    \tab \code{1:10=1}         \tab
    Range of values      \tab \code{7:9=3}          \tab Special values \code{lo} ar
                    \tab \code{lo:115=1}       \tab
     Other values          \tab \code{else=NA}           \tab
    }

\item Character values are quoted as in :

    \code{recodes = "c(1,2,5)='sanitary' else='unsanitary'"}

\item The output may be the (scalar) result of a function call as in:

     \code{recodes = "999=median(var, na.rm = TRUE)"}

\item Users are advised to carefully check the results of \code{recode()} calls
with any outputs that are the results of a function call.

\item The output may be the (scalar) value of a variable as in:

     \code{recodes = "999=scalarVariable"}

\item If all of the output values are numeric, and if \code{'afr'} is \code{FALSE},
then a numeric result is returned; if \code{var} is a factor then
(by default) so is the result.
```

`afr` Return a factor. Default is TRUE if `var` is a factor and is FALSE otherwise

`anr` Coerce result to numeric (default is TRUE)

`levels` Order of the levels in the returned factor; the default is to use the sort order of the level names.

Value

Recoded variable

Examples

```
# Recode values from 1 to 9 to various specifications
var <- sample(x = 1:9, size = 100, replace = TRUE)

# Recode single values
recode(var = var, recodes = "9=NA")

# Recode set of values
recode(var = var, recodes = "c(1,2,5)=1")

# Recode range of values
recode(var = var, recodes = "1:3=1;4:6=2;7:9=3")

# Recode other values
recode(var = var, recodes = "c(1,2,5)=1;else=NA")
```

`somalia_population` *Somalia regional population in 2022*

Description

A data.frame with 19 rows and 18 columns:

Usage

```
somalia_population
```

Format

An object of class `data.frame` with 19 rows and 18 columns.

Details

Variable	Description
`region`	Region name
`total`	Total population
`urban`	Total urban population
`rural`	Total rural population
`idp`	Total IDP population
`urban_stressed`	Total urban population - stressed
`rural_stressed`	Total rural population - stressed
`idp_stressed`	Total IDP population - stressed
`urban_crisis`	Total urban population - crisis
`rural_crisis`	Total rural population - crisis
`idp_crisis`	Total IDP population - crisis
`urban_emergency`	Total urban population - emergency
`rural_emergency`	Total rural population - emergency
`idp_emergency`	Total IDP population - emergency
`urban_catastrophe`	Total urban population - catastrophe
`rural_catastrophe`	Total rural population - catastrophe
`idp_catastrophe`	Total IDP population - catastrophe
`percent_at_least_crisis`	Percentage of population that are at least in crisis

Source

https://fsnau.org/downloads/2022-Gu-IPC-Population-Tables-Current.pdf

---

`villageData` *Cluster Population Weights Dataset*

Description

Dataset containing cluster population weights for use in performing posterior weighting with the blocked weighted bootstrap approach. This dataset is from a mother and child health and nutrition survey conducted in 4 districts from 3 regions in Somalia.

Usage

```
villageData
```

Format

A data frame with 6 columns and 117 rows:

Variable	**Description**
`region`	Region in Somalia from which the cluster belongs to
`district`	District in Somalia from which the cluster belongs to
`psu`	The PSU identifier. This must use the same coding system used to identify the PSUs that is used in the indicators
`lon`	Longitude coordinate of the cluster
`lat`	Latitude coordinate of the cluster
`pop`	Population size of the cluster

Source

Mother and child health and nutrition survey in 3 regions of Somalia

Examples

```
villageData
```

Index