Package ‘forestinventory’

January 8, 2021

Type Package

Title Design-Based Global and Small-Area Estimations for Multiphase Forest Inventories

Version 1.0.0

Date 2021-01-08

Maintainer Andreas Hill `<andreas.hill@wald-rlp.de>`

Description Extensive global and small-area estimation procedures for multiphase forest inventories under the design-based Monte-Carlo approach are provided. The implementation has been published in the Journal of Statistical Software (<doi:10.18637/jss.v097.i04>) and includes estimators for simple and cluster sampling published by Daniel Mandallaz in 2007 (<doi:10.1201/9781584889779>), 2013 (<doi:10.1139/cjfr-2012-0381>, <doi:10.1139/cjfr-2013-0181>, <doi:10.1139/cjfr-2013-0449>, <doi:10.3929/ethz-a-009990020>) and 2016 (<doi:10.3929/ethz-a-010579388>). It provides point estimates, their external- and design-based variances and confidence intervals, as well as a set of functions to analyze and visualize the produced estimates. The procedures have also been optimized for the use of remote sensing data as auxiliary information, as demonstrated in 2018 by Hill et al. (<doi:10.3390/rs10071052>).

License GPL (>= 2)

LazyData TRUE

Imports methods, stats, utils, plyr (>= 1.8.6), tidyr, ggplot2

Suggests R.rsp

VignetteBuilder R.rsp

RoxygenNote 7.1.1

NeedsCompilation no

Author Andreas Hill [aut, cre],
Alexander Massey [aut],
Daniel Mandallaz [ctb]

Repository CRAN

Date/Publication 2021-01-08 11:00:05 UTC

R topics documented:

`confint` *Calculates Confidence Intervals for Global and Small-Area Estimations*

Description

Calculates Confidence Intervals for Global and Small-Area Estimations

Usage

```
## S3 method for class 'onephase'
confint(object, parm, level = 0.95, adjust.method = "none", ...)

## S3 method for class 'twophase'
confint(object, parm, level = 0.95, adjust.method = "none", ...)

## S3 method for class 'threephase'
confint(object, parm, level = 0.95, adjust.method = "none", ...)
```

Arguments

`object`	object of class `onephase`, `twophase` or `threephase`, containing estimation results of the respective estimation method.
`parm`	ignored.
`level`	the confidence level required.
`adjust.method`	correction method to obtain simultaneous confidence intervals for a set of estimates (thus restricted to objects of class `"onephase"`, `c("smallarea","twophase")` and `c("smallarea","threephase")`). Available correction methods are `c("none","bonferroni")`. Defaults to `"none"`.
`...`	additional arguments, so far ignored.

Details

Depending on the estimation method specified, `confint()` computes confidence intervals as follows:

`onephase`:

Two-sided confidence intervals are computed based on the t-distribution with `n2 -1` *degrees of freedom*, where `n2` is the number of terrestrial data in the respective inventory domain.

`twophase`:

The calculation of the two-sided confidence intervals for *global* twophase estimates (objects of class `global`) are calculated based on the quantiles of the *t*-distribution with `n2 -p` *degrees of freedom*, where `p` is the number of parameters used in the regression model, and `n2` is the number of terrestrial observations (i.e. *local densities*) in the inventory domain.

The calculation of the two-sided confidence intervals for *smallarea* twophase estimates (objects of class `smallarea`) are calculated based on the quantiles of the *t*-distribution with `n2G -1` *degrees of freedom*, where `n2G` is the number of terrestrial observations (i.e. *local densities*) in the smallarea.

`threephase`:

The calculation of the two-sided confidence intervals for *global* threephase estimates (objects of class `global`) are calculated based on the quantiles of the *t*-distribution with `n2 -p` *degrees of freedom*, where `p` is the number of parameters used in the **full** regression model, and `n2` is the number of terrestrial observations (i.e. *local densities*) in the inventory domain (note: in notation used here n0, n1 and n2 correspond to the zero, first and second phase sample sizes respectively).

The calculation of the two-sided confidence intervals for *smallarea* theephase estimates (objects of class `smallarea`) are calculated based on the quantiles of the *t*-distribution with `n2G -1` *degrees of freedom*, where `n2G` is the number of terrestrial observations (i.e. *local densities*) in the smallarea.

Value

`confint` returns a list of the following 3 components:

`ci` a `data.frame` containing the columns:

- `area` the domain, i.e. small area
- `ci_lower_ext` the lower confidence limit based on the external variance
- `ci_upper_ext` the upper confidence limit based on the external variance
- `ci_lower_g` the lower confidence limit based on the g-weight variance
- `ci_upper_g` the upper confidence limit based on the g-weight variance

`level` the applied confidence level

`adjust.method` the adjustment method applied to retrieve simultaneous confidence intervals

Note

In the special case of *synthetic* smallarea estimations, the two-sided confidence intervals are calculated based on the quantiles of the *t*-distribution with `n2 -p` *degrees of freedom*, i.e. based on the global sample size.

The confidence intervals for *synthetic* smallarea estimations do not account for the potential bias of a linear model that was fit in a large forest area and applied to a small area. Thus, the coverage rates for confidence intervals produced by synthetic estimators may be less than the nominal level of confidence.

In case of cluster-sampling, `n2G` is the number of terrestrial clusters (a cluster constitutes the sample unit). This is automatically considered by `confint`.

The adjustment methods passed to `adjust.method` are designed to achieve *simultaneous* confidence intervals by correcting the confidence level given by `level`. The use of this option is recommended if a set of estimates contained in a `onephase`- or `smallarea`-object should be compared by their confidence intervals. It ensures that the percentage of confidence intervals containing the true value will correspond to the nominal confidence level.

References

Hill, A., Massey, A. F. (2021). *The R Package forestinventory: Design-Based Global and Small Area Estimations for Multiphase Forest Inventories.* Journal of Statistical Software, 97(4), 1-40.

Mandallaz, D. (2013). *Design-based properties of some small-area estimators in forest inventory with two-phase sampling.* Canadian Journal of Forest Research, 43(5), 441-449.

Mandallaz, D., Breschan, J., & Hill, A. (2013). *New regression estimators in forest inventories with two-phase sampling and partially exhaustive information: a design-based monte carlo approach with applications to small-area estimation.* Canadian Journal of Forest Research, 43(11), 1023-1031.

Mandallaz, D. (2013). *A three-phase sampling extension of the generalized regression estimator with partially exhaustive information.* Canadian Journal of Forest Research, 44(4), 383-388.

Benjamini, Y., and Hochberg, Y. (1995). *Controlling the false discovery rate: a practical and powerful approach to multiple testing.* Journal of the Royal Statistical Society Series B 57, 289-300.

Examples

```
## Calculate twophase estimations by extended pseudosynthetic estimator
#  for 4 small areas ("A", "B", "C", "D") using the grisons-dataset:
sae.est <- twophase(formula = tvol ~ mean + stddev + max + q75,
                    data = grisons,
                    phase_id = list(phase.col = "phase_id_2p", terrgrid.id = 2),
                    small_area = list(sa.col = "smallarea",
                                      areas = c("A", "B","C", "D"),
                                      unbiased = TRUE))

## calculate 95%-confidence intervals for each small area:
confint(sae.est)

## calculate simultaneous 95%-confidence intervals using 'bonferroni'-method:
confint(sae.est, adjust.method = "bonferroni")
```

estTable *estTable*

Description

estTable can be used to compare the results of onephase to multiphase estimations (twophase, threephase). It restructures the estimation results into a table that can be used to plot the estimation results and provides the basis for further analysis.

Usage

```
estTable(
  est.list,
  sae = FALSE,
  add.ci = TRUE,
  vartypes = c("variance", "ext_variance", "g_variance")
)
```

Arguments

est.list	a list object containing at least one multiphase estimation object created by the twophase or threephase function and the respective onephase estimation object.
sae	an object of type logical. Has to be set to TRUE if results of small area estimations are passed to estTable. Defaults to FALSE.
add.ci	logical: Should confidence intervals be added? Defaults to TRUE.
vartypes	Specifying the variances that should be included in the estimation table. Has to be specified as a character vector. The full set contains "variance", "ext_variance" and "g_variance".

Value

estTable returns a list of the following components:

- area: in case of small area estimations: the name of the small area
- estimate: the point estimates
- vartype: the type of variance
- variance: the variance values
- std: the standard errors (square root of variance values)
- error: the estimation errors defined as the ratio between *standard error* and *point estimate*
- domain: indicating if current row belongs to a smallarea or global estimation
- estimator: the estimator that that was applied
- method: the estimation method that was applied
- n2: terrestrial sample size in entire inventory area

- `n1`: first phase sample size in entire inventory area
- `n0`: in case of `threephase` estimations: zero phase sample size in entire inventory area
- `n2G`: terrestrial sample size in small area
- `n1G`: first phase sample size in small area
- `n0G`: in case of `threephase` estimations: zero phase sample size in small area
- `r.squared`: coefficient of determination of regression model
- `r.squared_reduced`: in case of `threephase` estimations: coefficient of determination of reduced regression model
- `r.squared_full`: in case of `threephase` estimations: coefficient of determination of full regression model
- `ci_lower`: if `add.ci=TRUE`: lower confidence limit
- `ci_upper`: if `add.ci=TRUE`: upper confidence limit

Note

An estimation object of class onephase as input is mandatory

Examples

```
## run onephase estimation:
op.a <- onephase(formula = tvol~1,
                 data = grisons,
                 phase_id = list(phase.col = "phase_id_2p", terrgrid.id = 2),
                 area = list(sa.col = "smallarea", areas = c("A", "B", "C", "D")))

## run small area twophase estimation:
sae.2p.est <- twophase(formula = tvol ~ mean + stddev + max + q75,
                       data = grisons,
                       phase_id = list(phase.col = "phase_id_2p", terrgrid.id = 2),
                    small_area = list(sa.col = "smallarea", areas = c("A", "B","C", "D"),
                                      unbiased = TRUE))

## run small area threephase estimation:
sae.3p.est <- threephase(formula.s0 =  tvol ~ mean,
                         formula.s1 = tvol ~ mean + stddev + max + q75,
                         data = grisons,
                   phase_id = list(phase.col = "phase_id_3p", s1.id = 1, terrgrid.id = 2),
                     small_area=list(sa.col = "smallarea", areas = c("A", "B", "C", "D"),
                                      unbiased = TRUE))

## create estimation table with confidence intervals:
sae.table<- estTable(est.list = list(op.a, sae.2p.est, sae.3p.est), add.ci=TRUE,
                    sae = TRUE, vartypes = c("variance", "g_variance",  "ext_variance"))

sae.table.df<- as.data.frame(sae.table)
```

`forestinventory` *forestinventory: A package for computing design-based estimates for multiphase forest inventories*

Description

The package provides ***global***- and ***smallarea estimators*** for ***twophase*** and ***threephase*** forest inventories under simple and cluster sampling, which have been developed by Daniel Mandallaz at ETH Zurich. The implemented methods have been published and applied in various studies (see `References`) and can be used for ***double- and triple sampling for stratification***, ***double- and triple sampling for regression*** and ***double- and triple sampling for regression within strata***.

Functions

The package provides three main functions to apply the various estimators for *twophase* and *threephase* forest inventories:

- `twophase` Function to apply global- and various smallarea estimation techniques for twophase inventories
- `threephase` Function to apply global- and various smallarea estimation techniques for threephase inventories
- `onephase` Function to apply estimations for onephase inventories, mainly for comparison with `two`-and `threephase`

Motivation

The Motivation of writing this package was to provide an extensive and consistent collection of state-of-the-art *design-based* estimation techniques for forest inventories. It was especially designed to facilitate the application of the available estimators in forest practice as well as in scientifically related studies. The work on this package was also the trigger to complete the range of the already published estimators, especially in the framework of three-phase smallarea estimators.

Selected references

Hill, A., Massey, A. F. (2021). *The R Package forestinventory: Design-Based Global and Small Area Estimations for Multiphase Forest Inventories.* Journal of Statistical Software, 97(4), 1-40.

Massey, A. F. (2015). *Multiphase estimation procedures for forest inventories under the design-based Monte Carlo approach* (Doctoral dissertation, Diss., ETH Zurich, Nr. 23025).

Mandallaz, D. (2013). *Design-based properties of some small-area estimators in forest inventory with two-phase sampling.* Canadian Journal of Forest Research, 43(5), 441-449.

Mandallaz, D., Breschan, J., & Hill, A. (2013). *New regression estimators in forest inventories with two-phase sampling and partially exhaustive information: a design-based monte carlo approach with applications to small-area estimation.* Canadian Journal of Forest Research, 43(11), 1023-1031.

Mandallaz, D. (2013). *A three-phase sampling extension of the generalized regression estimator with partially exhaustive information.* Canadian Journal of Forest Research, 44(4), 383-388.

`grisons` *Data from a multiphase forest inventory in the canton of Grisons, Switzerland*

Description

A dataset containing observations of 306 systematically arranged sample plots. Auxiliary information for all 306 plots is provided in the form of LiDAR canopy height metrics. For a systematic subsample of 67 out of the 306 plots, terrestrial information of the timber volume is provided from a terrestrial survey in the year 2007. Originally the inventory was carried out as a twophase inventory and has been artificially extended to a threephase inventory for demonstration purposes.

Usage

```
grisons
```

Format

data frame with 306 rows and 14 columns

Details

- `phase_id_2p` phase-membership of each observation for the twophase inventory. The large phase is indicated by `1`, the terrestrial phase by `2`.
- `phase_id_3p` phase-membership of each observation for the threephase inventory, i.e. the largest phase (`0`), the large phase (`1`) and terrestrial phase (`2`). *Note:* The threephase sample scheme was artificially created for demonstration purposes of the `threephase`-functions.
- `boundary_weights` proportion of analysis-window for auxiliary information lying within the forest.
- `mean` mean canopy height at the sample location based on the LiDAR canopy height model.
- `stddev` standard deviation of the LiDAR canopy height model at the sample location.
- `max` maximum value of the LiDAR canopy height model at the sample location.
- `q75` 75%-Quantile of the LiDAR canopy height model at the sample location.
- `smallarea` smallarea-indicator for each observation.
- `tvol` terrestrial timber volume from field survey. Use for `twophase`-inventory.
- `tvol.3p` terrestrial timber volume from field survey. Use for `threephase`-inventory.

Note

There are additional columns in `grisons` to demonstrate the function-behaviors for special cases which might occur in a forest inventory

- `phase_id_3p_nG0` one of the smallareas does not contain any terrestrial observation.
- `phase_id_3p_nG1` one of the smallareas does contain only a single terrestrial observation.

- tvol.3p_nG0 Use as response variable to test phase_id_3p_nG0 for threephase-inventory.
- tvol.3p_nG1 Use as response variable to test phase_id_3p_nG1 for threephase-inventory.

We leave testing these special cases to the user.

Source

The terrestrial data are kindly provided by the forest service of the canton grisons.

The dataset was created and used within the framework of the publications listed under *References*.

References

Mandallaz, D., Breschan, J., & Hill, A. (2013). *New regression estimators in forest inventories with two-phase sampling and partially exhaustive information: a design-based monte carlo approach with applications to small-area estimation.* Canadian Journal of Forest Research, 43(11), 1023-1031.

Hill, A., Breschan, J., & Mandallaz, D. (2014). *Accuracy assessment of timber volume maps using forest inventory data and LiDAR canopy height models.* Forests, 5(9), 2253-2275.

---

mphase.gain *mphase.gain*

Description

mphase.gain takes as input an object created by the estTable function and returns a validation of which multiphase method and estimator performed best in comparison to the onephase estimation (baseline) in terms of estimation precision.

Usage

```
mphase.gain(esttable.obj, pref.vartype = "g_variance", exclude.synth = TRUE)
```

Arguments

esttable.obj	an object of class esttable created by the estTable function
pref.vartype	preferred type of multiphase variance that should be compared to the onephase variance, if more then one variance type has been calculated in the multiphase estimation object(s) stored in esttable. Valid input values are "g_variance" (default) and "ext_variance".
exclude.synth	logical. If set to TRUE (default), synthetic estimations are not considered in the validation.

Value

`mphase.gain` returns a `data.frame` containing the following components:

- `area`: in case of small area estimation: the name of the small area
- `var_onephase`: standard error of the `onephase` estimation
- `var_multiphase`: smallest variance among the (set of) multiphase estimations stored in `esttable.obj`
- `estimator`: multiphase estimator with the smallest variance
- `method`: estimation Method of the multiphase estimator with the smallest variance
- `gain`: the *gain* is the reduction (if value is positive) or possibly also the increase (if value is negative) in variance when applying the multiphase as alternative to the onephase estimation
- `rel.eff`: the *relative efficiency* defined as the ratio between the onephase variance and the multiphase variance

Note

The *gain* can be interpreted as: "The multiphase estimation procedure leads to a `gain` % reduction in variance compared to the onephase procedure".

The *relative efficiency* can be interpreted as: "Using the onephase estimation procedure, the terrestrial sample size would have to be `rel.eff` times larger in order to achieve the same precision (in terms of variance) as the mutiphase estimation procedure".

`onephase` *onephase*

Description

onephase is used to calculate estimations exclusively based on terrestrial observations of a forest inventory (i.e. the *local densities*). The estimation method is available for simple and cluster-sampling and provides point estimates of the sample mean and their variances.

Usage

```
onephase(
  formula,
  data,
  phase_id = list(phase.col = NA, terrgrid.id = NA),
  cluster = NA,
  area = list(sa.col = NA, areas = NA)
)
```

Arguments

formula
: an object of class "formula" that must be of the form `Y ~ 1`, where Y is the terrestrial response value of interest provided in `data`.

data
: a data frame or vector containing the response value Y. Specifications are given under 'Details'.

phase_id
: an object of class "list" containing two elements:

  - `phase.col`: the column name in `data` that specifies the phase membership of each observation
  - `terrgrid.id`: the indicator identifying the the terrestrial (a.k.a. "ground truth") phase for that column (must be of type "numeric")

  Note: Only has to be specified if `data` is of class `data.frame`.

cluster
: Specifies the column name in `data` containing the cluster identification. Only used in case of cluster sampling.

area
: (*Optional*) an object of class "list" containing two elements:

  - `sa.col`: the column name in `data` containing domain identification
  - `areas`: vector of desired domains for which the estimation should be computed. If estimations for multiple domains should be computed, the domains have to be defined within a `character` vector using `c()`

  Further details of the parameter-specifications are given under *'Details'*.

Details

`data` can either be a vector only containing the observations of the response variable Y, *or* a data frame containing a column for the response variable and a column for the sample-grid indication that has to be further specified by argument `phase_id`. Additional *optional* columns include a cluster identification in case of cluster sampling, as well as a column that specifies a domain (e.g. a forest district) the respective terrestrial observation falls into. The latter allows to compute onephase-estimations for multiple domains at a time (see *'Examples'*).

Value

`onephase` returns an object of class `"onephase"`.

The functions `summary` and `confint` can be used to obtain a summary of the estimation results (point estimations, variances and sample sizes) and the confidence intervals for the respective point estimates.

An object of class `"onephase"` returns a `list` of the following components:

input
: a `list` containing the function inputs

estimation
: a data frame containing the following components:

  - `area`: the domain (only present if argument `area` has been used)
  - `estimate`: the point estimate
  - `variance`: the variance of the point estimate
  - `n2`: the terrestrial sample size

samplesizes
: a named numeric vector giving the terrestrial samplesize

References

Hill, A., Massey, A. F. (2021). *The R Package forestinventory: Design-Based Global and Small Area Estimations for Multiphase Forest Inventories.* Journal of Statistical Software, 97(4), 1-40.

Mandallaz, D. (2007). *Sampling techniques for forest inventories.* Chapter 4. CRC Press.

Examples

```
# ----------- non-cluster sampling------------------#

## load grisons dataset:
data(grisons)

## 1) calculate onephase-estimation for entire dataset:
op <- onephase(formula = tvol~1 ,data = grisons,
              phase_id =list(phase.col = "phase_id_2p",terrgrid.id = 2))
summary(op)
confint(op)

## 2) calculate onephase-estimation for given domains (areas) in dataset:
op.a <- onephase(formula = tvol~1,
                 data = grisons,
                 phase_id = list(phase.col = "phase_id_2p", terrgrid.id = 2),
                 area = list(sa.col = "smallarea", areas = c("A", "B")))
summary(op.a)
confint(op.a)

# ----------- cluster sampling ------------------#

## load zurichberg dataset:
data(zberg)

## 1) calculate onephase-estimation for entire dataset:
op.clust <- onephase(formula = basal~1, data = zberg,
                     phase_id = list(phase.col = "phase_id_2p",terrgrid.id = 2),
                     cluster = "cluster")
summary(op.clust)
confint(op.clust)

## 2) calculate onephase-estimation for given areas in dataset:
op.clust.a <- onephase(formula = basal~1,
                       data = zberg,
                       phase_id = list(phase.col = "phase_id_2p", terrgrid.id = 2),
                       cluster = "cluster",
                       area = list(sa.col = "ismallg23", areas = c("2", "3")))
summary(op.clust.a)
confint(op.clust.a)
```

`plot.esttable` *Plotting Estimation Results*

Description

Function plots the estimation results of an object created by the estTable function. Provides the possibility to visualize and compare the point estimates and their estimation errors differentiated by the applied estimation method and estimator.

Usage

```
## S3 method for class 'esttable'
plot(x, yvar = "error", ncol = 5, yscale.free = TRUE, ...)
```

Arguments

x	object of class "list" "esttable" created by the estTable function.
yvar	if set to "error" (default), the estimation error is plotted on the y-axis. If set to "estimate", point estimates with their confidence intervals are plotted.
ncol	number of columns to plot small area estimations.
yscale.free	logical: should y-axis scales be free (default) or fixed.
...	ignored.

Examples

```
## run onephase estimation:
op.a <- onephase(formula = tvol~1,
                 data = grisons,
                 phase_id = list(phase.col = "phase_id_2p", terrgrid.id = 2),
                 area = list(sa.col = "smallarea", areas = c("A", "B", "C", "D")))

## run small area twophase estimation:
sae.2p.est <- twophase(formula = tvol ~ mean + stddev + max + q75,
                       data = grisons,
                       phase_id = list(phase.col = "phase_id_2p", terrgrid.id = 2),
                    small_area = list(sa.col = "smallarea", areas = c("A", "B","C", "D"),
                                      unbiased = TRUE))

## run small area threephase estimation:
sae.3p.est <- threephase(formula.s0 =  tvol ~ mean,
                         formula.s1 = tvol ~ mean + stddev + max + q75,
                         data = grisons,
                   phase_id = list(phase.col = "phase_id_3p", s1.id = 1, terrgrid.id = 2),
                     small_area=list(sa.col = "smallarea", areas = c("A", "B", "C", "D"),
                                      unbiased = TRUE))

## create estimation table:
sae.table<- estTable(est.list = list(op.a, sae.2p.est, sae.3p.est), add.ci=TRUE,
                    sae = TRUE, vartypes = c("variance", "g_variance",  "ext_variance"))

## plot estimation errors:
plot(sae.table)
```

```
## plot point estimates and confidence intervals:
#  Hint: --> use ggplot2 functions to modify graphic:
library(ggplot2)
plot(sae.table, yvar = "estimate") +
   ylab("Timber Volume [m3/ha]")
```

---

summary *Summarizing Global and Small-Area Estimation Results*

Description

Summarizing Global and Small-Area Estimation Results

Usage

```
## S3 method for class 'onephase'
summary(object, coefs = FALSE, ...)

## S3 method for class 'twophase'
summary(object, coefs = FALSE, ...)

## S3 method for class 'threephase'
summary(object, coefs = FALSE, ...)
```

Arguments

`object`	object of class `onephase`, `twophase` or `threephase`, containing estimation results of the respective estimation method.
`coefs`	of type "`logical`". If set to `TRUE`, also gives the regression coefficients of `twophase` and `threephase` estimations. Defaults to `FALSE`.
`...`	additional arguments, so far ignored.

---

threephase *threephase*

Description

`threephase` is used to calculate estimations based on triple sampling under the *model-assisted Monte Carlo approach*. A *zero phase* of auxiliary information (e.g. taken from remote sensing data) is used to generate model predictions based on multiple linear regression using the method of ordinary least squares. A subsample of the zero phase comprises further auxiliary information that produces another set of model predictions. A further subsample produces a *second phase* based on terrestrial observations (i.e. the *local densities* of the ground truth) and is used to correct for bias in the design-based sense. The estimation method is available for *simple* and *cluster sampling* and includes the special case where the first phase is based on an *exhaustive* sample (i.e. a census). *Small-area applications* are supported for synthetic estimation as well as two varieties of bias-corrected estimators: the traditional small-area estimator and an asymptotically equivalent version derived under Mandallaz' extended model approach.

Usage

```
threephase(
  formula.s0,
  formula.s1,
  data,
  phase_id,
  cluster = NA,
  small_area = list(sa.col = NA, areas = NA, unbiased = TRUE),
  boundary_weights = NA,
  exhaustive = NA,
  progressbar = FALSE,
  psmall = FALSE
)
```

Arguments

`formula.s0` an object of class "`formula`" as would be used in the function `lm` that contains a reduced set of auxiliary variables available for all zero phase plots

`formula.s1` an object of class "`formula`" as would be used in the function `lm` that contains the predictors from `formula.s0` as well as further ancilliary predictors available for all first phase plots (i.e. `formula.s0` is **nested** in `formula.s1`)

`data` a data frame containing all variables contained in `formula` and a column indexing phase membership. Additional columns designating small-area membership, cluster ID and boundary weights should also be contained in the data frame if they are requested in the function.

`phase_id` an object of class "`list`" containing three elements:

- `phase.col`: the column name in `data` that specifies the phase membership of each observation
- `s1.id`: the indicator identifying the "second phase only" plots for that column (must be of type "`numeric`")
- `terrgrid.id`: the indicator identifying the terrestrial (a.k.a. "ground truth") phase for that column (must be of type "`numeric`")

`cluster` (*Optional*) Specifies the column name in `data` containing the cluster ID. Only used in case of cluster sampling.

`small_area` (*Optional*) a list that if containing three elements:

- `sa.col`: the column name in `data` containing domain identification
- `areas`: vector of desired small-area domain identifiers
- `unbiased`: an object of type "`logical`" that when FALSE designates that the estimator is allowed to be biased (i.e. the synthetic estimator) and when TRUE forces it to be design-unbiased. See *'Details'*.

Note: If `small_area` is left unchanged then `twophase` defaults to global estimation.

`boundary_weights`
(*Optional*) Specifies the column name in `data` containing the weights for boundary adjustment. See *'Details'*

exhaustive
: (*Optional*) For global estimation, a vector of true auxiliary means corresponding to an exhaustive first phase. The vector must be input in the same order that `lm` processes a `formula` object and include the intercept term. For small area estimation, `exhaustive` is a `data.frame` containing column names (`colnames`) for every variable appearing in the parameter `formula` including the variable "Intercept".Rownames (`row.names`) have to be used and must correspond to the names of the small areas. See *'Details'*.

progressbar
: (*Optional*) an object a type "`logical`" that when TRUE prints the progress of the calculation in the console (recommended for large amount of small areas). Defaults to FALSE.

psmall
: (*Optional*) an object a type "`logical`" used for small area estimations that only works when `unbiased` in the parameter `small_area` is set to TRUE. See *'Details'*.

Details

`s1.id` identifies "second phase only" plots because the terrestrial phase is known to be part of the second phase by the construction of the subsampling.

If estimations for multiple small-area domains should be computed, the domains have to be defined within a `character` vector using `c()`. Using `small_area(...,unbiased=FALSE)` calculates design-based estimates with the synthetic estimator and may be design-biased if the model is biased in that small area. The default, `small_area(...,unbiased=TRUE)`, allows for a residual correction by one of two asymptotically equivalent methods to create design-unbiased estimates:

- Mandallaz' extended model approach calculates the residual correction by extending the model formula with an indicator variable in the small area. It is the default method `psmall`=FALSE.
- the traditional small area estimator calculates the residual correction by taking the synthetic estimator and adding the mean residual observed in the small area. It is activated when `psmall`=TRUE.

Missing values (`NA`) in the auxiliary variables (i.e. at least one auxiliary variable cannot be observed at an inventory location) are automatically removed from the dataset *before* the estimations are computed. Note that missingness in the auxiliary variables is only allowed if we assume that they are *missing at random*, since the unbiasedness of the estimates is based on the sampling design.

The boundary weight adjustment is pertinent for auxiliary information derived from remote sensing and is equal to the percentage of forested area (e.g. as defined by a forest mask) in the interpretation area.

Exhaustive estimation refers to when the true means of certain auxiliary variables are known at an exhaustive zero phase (i.e. a census). For global estimation, the vector must be input in the same order that `lm` processes a `formula` object including the intercept term whose true mean will always be one. For small area estimation, `exhaustive` is a `data.frame` containing column names for every variable appearing in the parameter `formula` including the variable "Intercept". The observations of the data.frame must represent the true auxiliary means in the same order as was presented in `areas` from the parameter `small_area`. See *'Examples'*.

Value

`threephase` returns an object of class `"threephase"`.

An object of class "threephase" returns a list of the following components:

input	a list containing the function's inputs
estimation	a data frame containing the following components: • area: the domain (only present if argument areas has been used) • estimate: the point estimate • ext_variance: the external variance of the point estimate that doesn't account for fitting the model from the current inventory • g_variance: the internal (g-weight) variance that accounts for fitting the model from the current inventory • n0 the zero phase sample size of plots • n1 the first phase sample size of plots • n2 the second phase (i.e. terrestrial) sample size of plots • n0G the zero phase sample size in the small area • n1G the first phase sample size in the small area • n2G the second phase (i.e. terrestrial) sample size in the small area • r.squared_reduced the R-squared of the linear model based on formula.s0 (i.e. the reduced model) • r.squared_full the R-squared of the linear model based on formula.s1 (i.e. the full model)
samplesizes	a data.frame summarizing all samplesizes: in case of cluster sampling both, the number of individual plots and the number of clusters is reported.
coefficients	the coefficients of the two linear models: • alpha: the reduced model coefficients • beta: the full model coefficients
cov_alpha_s2	the design-based covariance matrix of the reduced model coefficients
cov_beta_s2	the design-based covariance matrix of the full model coefficients
Z_bar_1_s0	the estimated auxiliary means of formula.s0 based on the zero phase. If the zero phase is exhaustive, these are the true auxiliary means specified in the input-argument exhaustive.
Z1_bar_s1	the estimated auxiliary means of formula.s0 based on the first phase
Z_bar_s1	the estimated auxiliary means of formula.s1 based on the first phase
cov_Z_bar_1_s0	the covariance matrix for Z_bar_1_s0
resid_reduced	the reduced model residuals at either the plot level or cluster level depending on the call
resid_full	the full model residuals at either the plot level or cluster level depending on the call
warn.messages	logical indicating if warning messages were issued

Note

In the special case of cluster sampling, the reported sample sizes in estimation are the number of clusters. The samplesize-object also provides the respective number of single plot units for cluster sampling. The reported r.squared_reduced and r.squared_full describe the model fit of the applied linear regression models (i.e. on *plot-level*, not on *cluster level*).

References

Hill, A., Massey, A. F. (2021). *The R Package forestinventory: Design-Based Global and Small Area Estimations for Multiphase Forest Inventories.* Journal of Statistical Software, 97(4), 1-40.

Mandallaz, D., Breschan, J., & Hill, A. (2013). *New regression estimators in forest inventories with two-phase sampling and partially exhaustive information: a design-based monte carlo approach with applications to small-area estimation.* Canadian Journal of Forest Research, 43(11), 1023-1031.

Mandallaz, D. (2014). *A three-phase sampling extension of the generalized regression estimator with partially exhaustive information.* Can. J. For. Res. 44: 383-388

Massey, A. and Mandallaz, D. and Lanz, A. (2014). *Integrating remote sensing and past inventory data under the new annual design of the Swiss National Forest Inventory using three-phase design-based regression estimation.* Can. J. For. Res. 44(10): 1177-1186

Mandallaz, D. (2013). *Regression estimators in forest inventories with three-phase sampling and two multivariate components of auxiliary information.* ETH Zurich, Department of Environmental Systems Science,Tech. rep. Available from doi: 10.3929/ethza009990020.

Examples

```
## load datasets:
data(grisons)
data(zberg)


## define regression models for simple and cluster sampling:
formula.s0 <- tvol ~ mean # reduced model:
formula.s1 <- tvol ~ mean + stddev + max + q75 # full model
formula.clust.s0 <- basal ~ stade
formula.clust.s1 <- basal ~ stade + couver + melange


# -----------------------------------------------#
# ----------- GLOBAL ESTIMATION ------------------#

#----
## 1) -- Design-based estimation with non-exhaustive auxiliary information
#----

# 1.1) non-cluster-sampling (see eqns. [11], [14] and [16] in Mandallaz 2014):
summary(threephase(formula.s0, formula.s1, data = grisons,
                  phase_id = list(phase.col = "phase_id_3p", s1.id=1, terrgrid.id = 2)))


# 1.2) cluster-sampling (see eqns. [49] and [50] in Mandallaz 2013):
summary(threephase(formula.clust.s0, formula.clust.s1, data = zberg,
                   phase_id = list(phase.col="phase_id_3p", s1.id = 1, terrgrid.id = 2),
                   cluster = "cluster"))


# 1.3) example for boundary weight adjustment (non-cluster example):
```

```
summary(threephase(formula.s0, formula.s1, data = grisons,
                   phase_id = list(phase.col="phase_id_3p", s1.id = 1, terrgrid.id = 2),
                   boundary_weights = "boundary_weights"))

#----
## 2) -- Design-based estimation with exhaustive auxiliary information
#----

# 2.1) non-cluster-sampling (see eqns. [7], [9] and [10] in Mandallaz 2014):
summary(threephase(formula.s0, formula.s1, data = grisons,
                  phase_id = list(phase.col = "phase_id_3p", s1.id = 1, terrgrid.id = 2),
                   exhaustive = c(1,11.39)))


# 2.2) cluster-sampling:
summary(threephase(formula.clust.s0, formula.clust.s1, data = zberg,
                  phase_id = list(phase.col = "phase_id_3p", s1.id = 1, terrgrid.id = 2),
                   cluster = "cluster", exhaustive = c(1, 0.10, 0.7, 0.10)))


# ----------------------------------------------------#
# ----------- SMALL AREA ESTIMATION ------------------#

#----
## 1) --  Design-based estimation with non-exhaustive auxiliary information
#----

# 1.1) Mandallaz's extended pseudo small area estimator:
summary(threephase(formula.s0,
                   formula.s1,
                   data = grisons,
                  phase_id = list(phase.col = "phase_id_3p", s1.id = 1, terrgrid.id = 2),
                   small_area=list(sa.col = "smallarea", areas = c("A", "B", "C", "D"),
                                   unbiased = TRUE)))

summary(threephase(formula.clust.s0,
                   formula.clust.s1,
                   data = zberg,
                  phase_id = list(phase.col = "phase_id_3p", s1.id = 1, terrgrid.id = 2),
                   cluster = "cluster",
                small_area = list(sa.col = "ismallold", areas = c("1"), unbiased = TRUE)))


# 1.2) pseudo small area estimator:
summary(threephase(formula.s0,
                   formula.s1,
                   data = grisons,
                  phase_id = list(phase.col = "phase_id_3p", s1.id = 1, terrgrid.id = 2),
                  small_area = list(sa.col = "smallarea", areas = c("A", "B", "C", "D"),
                                   unbiased = TRUE),
                   psmall = TRUE))

summary(threephase(formula.clust.s0,
```

```
                   formula.clust.s1,
                   data=zberg,
                   phase_id=list(phase.col="phase_id_3p", s1.id=1, terrgrid.id=2),
                   cluster="cluster",
                   small_area=list(sa.col="ismallold", areas=c("1"), unbiased=TRUE),
                   psmall = TRUE))


# 1.3) pseudosynthetic small area estimator:
summary(threephase(formula.s0  = tvol ~ mean,
                   formula.s1 = tvol ~ mean + stddev + max + q75,
                   data = grisons,
                  phase_id = list(phase.col = "phase_id_3p", s1.id = 1, terrgrid.id = 2),
                  small_area = list(sa.col = "smallarea", areas = c("A", "B", "C", "D"),
                                   unbiased = FALSE)))

summary(threephase(formula.clust.s0,
                   formula.clust.s1,
                   data = zberg,
                  phase_id = list(phase.col = "phase_id_3p", s1.id = 1, terrgrid.id = 2),
                   cluster = "cluster",
               small_area = list(sa.col = "ismallold", areas = c("1"), unbiased = FALSE)))


#----
## 2) --  Design-based estimation with exhaustive auxiliary information
#----

# true auxiliary mean for variable "mean" taken from Mandallaz et al. (2013):
truemeans.G <- data.frame(Intercept = rep(1, 4),
                         mean = c(12.85, 12.21, 9.33, 10.45))
rownames(truemeans.G) <- c("A", "B", "C", "D")

# true auxiliary means taken from Mandallaz (1991):
truemeans.G.clust <- data.frame(Intercept = 1, stade400 = 0.175, stade500 = 0.429,
                               stade600 = 0.321)
rownames(truemeans.G.clust) <- c("1")


# 2.1) Mandallaz's extended small area estimator:
summary(threephase(formula.s0,
                   formula.s1,
                   data = grisons,
                  phase_id = list(phase.col = "phase_id_3p", s1.id = 1, terrgrid.id = 2),
                  small_area = list(sa.col = "smallarea", areas = c("A", "B", "C", "D"),
                                unbiased = TRUE),
                   exhaustive = truemeans.G))

summary(threephase(formula.clust.s0,
                   formula.clust.s1,
                   data = zberg,
                  phase_id = list(phase.col = "phase_id_3p", s1.id = 1, terrgrid.id = 2),
                   cluster = "cluster",
```

```
                small_area = list(sa.col = "ismallold", areas = c("1"), unbiased = TRUE),
                  exhaustive = truemeans.G.clust))


# 2.2) small area estimator:
summary(threephase(formula.s0,
                  formula.s1,
                  data = grisons,
                 phase_id = list(phase.col = "phase_id_3p", s1.id = 1, terrgrid.id = 2),
                  small_area = list(sa.col = "smallarea", areas = c("A", "B", "C", "D"),
                                unbiased = TRUE),
                  exhaustive = truemeans.G,
                  psmall = TRUE))

summary(threephase(formula.clust.s0,
                  formula.clust.s1,
                  data = zberg,
                 phase_id = list(phase.col = "phase_id_3p", s1.id = 1, terrgrid.id = 2),
                  cluster = "cluster",
                small_area = list(sa.col = "ismallold", areas = c("1"), unbiased = TRUE),
                  exhaustive = truemeans.G.clust,
                  psmall = TRUE))


# 2.3) synthetic small area estimator:
summary(threephase(formula.s0,
                  formula.s1,
                  data = grisons,
                  phase_id = list(phase.col="phase_id_3p", s1.id = 1, terrgrid.id = 2),
                  small_area = list(sa.col = "smallarea", areas = c("A", "B", "C", "D"),
                                unbiased = FALSE),
                  exhaustive = truemeans.G))

summary(threephase(formula.clust.s0,
                  formula.clust.s1,
                  data = zberg,
                 phase_id = list(phase.col = "phase_id_3p", s1.id = 1, terrgrid.id = 2),
                  cluster = "cluster",
               small_area = list(sa.col = "ismallold", areas = c("1"), unbiased = FALSE),
                  exhaustive = truemeans.G.clust))
```

---

twophase *twophase*

Description

twophase is used to calculate estimations based on double sampling under the *model-assisted Monte Carlo approach*. A *first phase* of auxiliary information (e.g. taken from remote sensing data) is used to generate model predictions based on multiple linear regression using the method of ordinary least squares. A subsample of the first phase comprises the *second phase* which contains terrestrial observations (i.e. the *local densities* of the ground truth) that is used to correct for bias in the design-based sense. The estimation method is available for *simple* and *cluster sampling* and includes the special case where the first phase is based on an *exhaustive* sample (i.e. a census). *Small-area applications* are supported for synthetic estimation as well as two varieties of bias-corrected estimators: the traditional small-area estimator and an asymptotically equivalent version derived under Mandallaz' extended model approach.

Usage

```
twophase(
  formula,
  data,
  phase_id,
  cluster = NA,
  small_area = list(sa.col = NA, areas = NA, unbiased = TRUE),
  boundary_weights = NA,
  exhaustive = NA,
  progressbar = FALSE,
  psmall = FALSE
)
```

Arguments

`formula` an object of class "`formula`" as would be used in the function `lm`

`data` a data frame containing all variables contained in `formula` and a column indexing phase membership. Additional columns designating small-area membership, cluster ID and boundary weights should also be contained in the data frame if they are requested in the function.

`phase_id` an object of class "`list`" containing two elements:

- `phase.col`: the column name in `data` that specifies the phase membership of each observation
- `terrgrid.id`: the indicator identifying the terrestrial (a.k.a. "ground truth") phase for that column (must be of type "`numeric`")

`cluster` (*Optional*) Specifies the column name in `data` containing the cluster ID. Only used in case of cluster sampling.

`small_area` (*Optional*) a list that if containing three elements:

- `sa.col`: the column name in `data` containing domain identification
- `areas`: vector of desired small-area domain identifiers
- `unbiased`: an object of type "`logical`" that when FALSE designates that the estimator is allowed to be biased (i.e. the synthetic estimator) and when TRUE forces it to be design-unbiased. See *'Details'*.

Note: If `small_area` is left unchanged then `twophase` defaults to global estimation.

`boundary_weights` (*Optional*) Specifies the column name in `data` containing the weights for boundary adjustment. See *'Details'* `exhaustive` (*Optional*) For global estimation, a vector of true auxiliary means corresponding to an exhaustive first phase. The vector must be input in the same order that `lm` processes a `formula` object and include the intercept term. For small area estimation, `exhaustive` is a `data.frame` containing column names (`colnames`) for every variable appearing in the parameter `formula` including the variable "Intercept".Rownames (`row.names`) have to be used and must correspond to the names of the small areas. See *'Details'*.

`progressbar` (*Optional*) an object a type "`logical`" that when TRUE prints the progress of the calculation in the console (recommended for large amount of small areas). Defaults to FALSE.

`psmall` (*Optional*) an object a type "`logical`" used for small area estimations that only works when `unbiased` in the parameter `small_area` is set to TRUE. See *'Details'*.

Details

If estimations for multiple small-area domains should be computed, the domains have to be defined within a `character` vector using `c()`. Using `small_area(...,unbiased=FALSE)` calculates design-based estimates with the synthetic estimator and may be design-biased if the model is biased in that small area. The default, `small_area(...,unbiased=TRUE)`, allows for a residual correction by one of two asymptotically equivalent methods to create design-unbiased estimates:

- Mandallaz' extended model approach calculates the residual correction by extending the model formula with an indicator variable in the small area. It is the default method `psmall`=FALSE.
- the traditional small area estimator calculates the residual correction by taking the synthetic estimator and adding the mean residual observed in the small area. It is activated when `psmall`=TRUE.

Missing values (`NA`) in the auxiliary variables (i.e. at least one auxiliary variable cannot be observed at an inventory location) are automatically removed from the dataset *before* the estimations are computed. Note that missingness in the auxiliary variables is only allowed if we assume that they are *missing at random*, since the unbiasedness of the estimates is based on the sampling design.

The boundary weight adjustment is pertinent for auxiliary information derived from remote sensing and is equal to the percentage of forested area (e.g. as defined by a forest mask) in the interpretation area.

Exhaustive estimation refers to when the true means of certain auxiliary variables are known and an exhaustive first phase (i.e. a census). For global estimation, the vector must be input in the same order that `lm` processes a `formula` object including the intercept term whose true mean will always be one. For small area estimation, `exhaustive` is a `data.frame` containing column names for every variable appearing in the parameter `formula` including the variable "Intercept". The observations of the data.frame must represent the true auxiliary means in the same order as was presented in `areas` from the parameter `small_area`. See *'Examples'*.

Value

`twophase` returns an object of class `"twophase"`.

An object of class `"twophase"` returns a `list` of the following components:

`input` a `list` containing the function's inputs

`estimation` a data frame containing the following components:

- `area`: the domain (only present if argument `areas` has been used)
- `estimate`: the point estimate
- `ext_variance`: the external variance of the point estimate that doesn't account for fitting the model from the current inventory
- `g_variance`: the internal (g-weight) variance that accounts for fitting the model from the current inventory
- `n1` the first phase sample size of plots
- `n2` the second phase (i.e. terrestrial) sample size of plots
- `n1G` the first phase sample size in the small area
- `n2G` the second phase (i.e. terrestrial) sample size in the small area
- `r.squared` the R squared of the linear model

`samplesizes` a `data.frame` summarizing all samplesizes: in case of cluster sampling both, the number of individual plots and the number of clusters is reported.

`coefficients` the linear model coefficients

`cov_coef` the design-based covariance matrix of the model coefficients

`Z_bar_1G` the estimated auxiliary means of `formula` based on the first phase. If the first phase is exhaustive, these are the true auxiliary means specified in the input-argument `exhaustive`.

`cov_Z_bar_1G` the covariance matrix of `Z_bar_1G`

`Rc_x_hat_G` the small-area residuals at either the plot level or cluster level depending on the call

`Rc_x_hat` the residuals at either the plot level or cluster level depending on the call

`Yx_s2G` the local densities in the small area

`Mx_s2G` the cluster weights in the small area

`mean_Rc_x_hat_G` the mean residual (weighted mean in the case of cluster sampling) in the small area

`mean_Rc_x_hat` the mean residual (weighted mean in the case of cluster sampling)

`warn.messages` logical indicating if warning messages were issued

Note

In the special case of cluster sampling, the reported sample sizes in `estimation` are the number of clusters. The `samplesize`-object also provides the respective number of single plot units for cluster sampling. The reported `r.squared` describe the model fit of the applied linear regression model (i.e. on *plot-level*, not on *cluster level*).

References

Hill, A., Massey, A. F. (2021). *The R Package forestinventory: Design-Based Global and Small Area Estimations for Multiphase Forest Inventories.* Journal of Statistical Software, 97(4), 1-40.

Mandallaz, D. (2007). *Sampling techniques for forest inventories.* Chapter 4. CRC Press.

Mandallaz, D. (2013). *Design-based properties of some small-area estimators in forest inventory with two-phase sampling.* Can. J. For. Res. 43: 441-449

Mandallaz, D. and Hill, A. and Massey, A. (2016). *Design-based properties of some small-area estimators in forest inventory with two-phase sampling.* ETH Zurich, Department of Environmental Systems Science,Tech. rep. Available from http://e-collection.library.ethz.ch.

Examples

```
## load datasets:
data(grisons)
data(zberg)

# -----------------------------------------------#
# ----------- GLOBAL ESTIMATION -----------------#

#----
## 1) -- Design-based estimation with non-exhaustive auxiliary information
#----

# 1.1) non-cluster-sampling:
summary(twophase(formula = tvol ~mean + stddev + max + q75,
                 data = grisons,
                 phase_id = list(phase.col = "phase_id_2p", terrgrid.id = 2)))

# 1.2) cluster-sampling (see eqns. [57] and [58] in Mandallaz, Hill, Massey 2016):
summary(twophase(formula = basal ~ stade + couver + melange,
                data = zberg,
                phase_id = list(phase.col = "phase_id_2p", terrgrid.id = 2),
                cluster = "cluster"))

# 1.3) example for boundary weight adjustment (non-cluster example):
summary(twophase(formula=tvol ~ mean + stddev + max + q75,
                 data=grisons,
                 phase_id=list(phase.col = "phase_id_2p", terrgrid.id = 2),
                 boundary_weights = "boundary_weights"))

#----
## 2) -- Design-based estimation with exhaustive auxiliary information
#----

# establish order for vector of true auxiliary means:
colnames(lm(formula = tvol ~ mean + stddev + max + q75, data = grisons, x = TRUE)$x)
true.means <- c(1, 11.39, 8.84, 32.68, 18.03)

# 2.1) non-cluster-sampling:
```

```
summary(twophase(formula = tvol ~ mean + stddev + max + q75,
                 data = grisons,
                 phase_id = list(phase.col = "phase_id_2p", terrgrid.id = 2),
                 exhaustive = true.means))

# 2.2) cluster-sampling:
summary(twophase(formula = stem ~ stade + couver + melange,
                 data = zberg,
                 phase_id = list(phase.col = "phase_id_2p", terrgrid.id = 2),
                 cluster = "cluster",
                 exhaustive = c(1, 0.10, 0.7, 0.10, 0.6, 0.8)))


# ----------------------------------------------------#
# ----------- SMALL AREA ESTIMATION ------------------#

#----
## 1) -- Design-based estimation with non-exhaustive auxiliary information
#----

# 1.1) Mandallaz's extended pseudo small area estimator (see eqns. [35] and [36] in Mandallaz 2013):
summary(twophase(formula = tvol ~ mean + stddev + max + q75, data = grisons,
                 phase_id = list(phase.col = "phase_id_2p", terrgrid.id = 2),
                 small_area = list(sa.col = "smallarea", areas = c("A", "B","C", "D"),
                                   unbiased = TRUE)))

summary(twophase(formula = basal ~ stade + couver + melange, data=zberg,
                 phase_id = list(phase.col = "phase_id_2p", terrgrid.id = 2),
                 cluster = "cluster",
                 small_area = list(sa.col = "ismallg23", areas = c("2", "3"),
                                   unbiased = TRUE)))


# 1.2) pseudo small area estimator (see eqns. [25] and [26] in Mandallaz 2013):
summary(twophase(formula = tvol ~ mean + stddev + max + q75, data = grisons,
                 phase_id = list(phase.col = "phase_id_2p", terrgrid.id = 2),
                 small_area = list(sa.col = "smallarea", areas = c("A", "B"),
                                   unbiased = TRUE),
                 psmall = TRUE))

summary(twophase(formula = basal ~ stade + couver + melange, data=zberg,
                 phase_id = list(phase.col = "phase_id_2p", terrgrid.id = 2),
                 cluster = "cluster",
                 small_area = list(sa.col = "ismallg23", areas = c("2", "3"),
                                   unbiased = TRUE),
                 psmall = TRUE))


# 1.3) pseudosynthetic small area estimator (see eqns. [35] and [36] in Mandallaz 2013):
summary(twophase(formula = tvol ~ mean + stddev + max + q75, data=grisons,
                 phase_id = list(phase.col = "phase_id_2p", terrgrid.id = 2),
                 small_area = list(sa.col = "smallarea", areas = c("B", "A"),
                                   unbiased = FALSE)))
```

```
summary(twophase(formula = basal ~ stade + couver + melange, data=zberg,
                 phase_id = list(phase.col = "phase_id_2p", terrgrid.id = 2),
                 cluster = "cluster",
                 small_area = list(sa.col = "ismallg23", areas = c("2", "3"),
                                   unbiased = FALSE)))


#----
## 2) -- Design-based estimation with exhaustive auxiliary information
#----

# establish order for vector of true auxiliary means:
colnames(lm(formula = tvol ~ mean + stddev + max + q75, data = grisons, x = TRUE)$x)
colnames(lm(formula = basal ~ stade + couver + melange, data = zberg, x = TRUE)$x)

# true auxiliary means taken from Mandallaz et al. (2013):
truemeans.G <- data.frame(Intercept = rep(1, 4),
                         mean = c(12.85, 12.21, 9.33, 10.45),
                         stddev = c(9.31, 9.47, 7.90, 8.36),
                         max = c(34.92, 35.36, 28.81, 30.22),
                         q75 = c(19.77, 19.16, 15.40, 16.91))
rownames(truemeans.G) <- c("A", "B", "C", "D")

# true auxiliary means taken from Mandallaz (1991):
truemeans.G.clust <- data.frame(Intercept = 1,
                               stade400 = 0.175,
                               stade500 = 0.429,
                               stade600 = 0.321,
                               couver2 = 0.791,
                               melange2 = 0.809)
rownames(truemeans.G.clust) <- c("1")


# 2.1) Mandallaz's extended small area estimator (see eqns. [31] and [33] in Mandallaz 2013):
summary(twophase(formula = tvol ~ mean + stddev + max + q75, data = grisons,
                 phase_id = list(phase.col = "phase_id_2p", terrgrid.id = 2),
                 small_area = list(sa.col ="smallarea", areas = c("A", "B"),
                                   unbiased = TRUE),
                 exhaustive = truemeans.G))

summary(twophase(formula = basal ~ stade + couver + melange, data=zberg,
                 phase_id = list(phase.col = "phase_id_2p", terrgrid.id = 2),
                 cluster = "cluster",
                 small_area = list(sa.col = "ismallold", areas = c("1"),
                                   unbiased = TRUE),
                 exhaustive = truemeans.G.clust))


# 2.2) small area estimator (see eqns. [20] and [21] in Mandallaz 2013):
summary(twophase(formula = tvol ~ mean + stddev + max + q75, data = grisons,
                 phase_id = list(phase.col = "phase_id_2p", terrgrid.id = 2),
                 small_area = list(sa.col = "smallarea", areas = c("A"),
```

```
                              unbiased = TRUE),
                 exhaustive = truemeans.G, psmall = TRUE))

summary(twophase(formula = basal ~ stade + couver + melange, data = zberg,
                 phase_id = list(phase.col = "phase_id_2p", terrgrid.id = 2),
                 cluster = "cluster",
                 small_area = list(sa.col ="ismallold", areas = c("1"),
                                   unbiased = TRUE),
                 psmall = TRUE,
                 exhaustive = truemeans.G.clust))


# 2.3) synthetic small area estimator (see eqns. [18] and [19] in Mandallaz 2013):
summary(twophase(formula=tvol ~ mean + stddev + max + q75, data=grisons,
                 phase_id=list(phase.col = "phase_id_2p", terrgrid.id = 2),
                 small_area=list(sa.col = "smallarea", areas = c("A", "B"),
                                 unbiased = FALSE),
                 exhaustive = truemeans.G))

summary(twophase(formula = basal ~ stade + couver + melange, data = zberg,
                 phase_id = list(phase.col = "phase_id_2p", terrgrid.id = 2),
                 cluster = "cluster",
                 small_area = list(sa.col = "ismallold", areas = c("1"),
                                   unbiased = FALSE),
                 exhaustive = truemeans.G.clust))
```

---

zberg *Data from a multiphase forest inventory at the zurichberg (zurich), switzerland*

Description

A dataset from 1991 containing 1203 sample plots observations from a forest inventory using cluster-sampling. The large phase comprises 298 clusters. Terrestrial information of the stem number as well as the basal area is available for a systematic subsample of 73 clusters. Auxiliary information at all 2103 sample plots were derived by stand maps. Originally the inventory was carried out as a twophase inventory and has been artificially extended to a threephase inventory for demonstration purposes.

Usage

```
zberg
```

Format

data frame with 1203 rows and 12 columns

Details

- cluster cluster identification. Maximum number of sample plots per cluster is 5.
- phase_id_2p phase-membership of each observation for the twophase inventory. The first phase is indicated by 1, the second (i.e. terrestrial) phase by 2.
- phase_id_3p the phase-membership of each observation for the threephase inventory, i.e. the first phase (0), the second phase (1) and third (terrestrial) phase (2). *Note:* The threephase sample scheme was artificially created for demonstration purposes of the threephase-functions.
- stade development stage at sample plot location based on the stand map. Categorical variable of class factor with 4 levels.
- melange degree of mixture at sample plot location based on the stand map. Categorical variable of class factor with 2 levels.
- couver crown-coverage at sample plot location based on the stand map. Categorical variable of class factor with 2 levels.
- stem stem number derived at field survey.
- basal basal area derived at field survey.
- ismallg23 indicator for small area 2 and 3 for each observation.
- ismallold indicator for small area 1 for each observation.

Source

Data provided by D.Mandallaz

References

Mandallaz, Daniel (1991). *A unified approach to sampling theory for forest inventory based on infinite population and superpopulation models.* http://dx.doi.org/10.3929/ethz-a-000585900

Mandallaz, Daniel (1993). *Geostatistical methods for double sampling schemes. application to combined forest inventories.* Chair of Forest Inventory and Planning, Swiss Federal Institute of Technology (ETH). http://dx.doi.org/10.3929/ethz-a-000943897

Index