Package 'rhte'

May 31, 2015

Type Package

Title Robust Horvitz-Thompson Estimation

Version 0.1-2

Date 2015-05-30

Author Beat Hulliger and Tobias Schoch

Maintainer Beat Hulliger <beat.hulliger@fhnw.ch>

Depends survey, KernSmooth

Suggests sampling

Description Robust mean and ratio estimation in complex sampling based on M-estimation, trimming, and winsorization.

License GPL (>=2)

LazyLoad yes

NeedsCompilation no

R topics documented:

rhte-package *Robust Mean and Ratio Estimation for Complex Samples*

Description

Robust mean and ratio estimation in complex samples based on M-estimation, trimming, and winsorization.

Details

Package:	rhte
Type:	Package
Version:	0.1-1
Date:	2011-11-04
License:	GPL (>=2)
LazyLoad:	yes

Implemented methods:

- Horvitz–Thompson M-estimator (total and mean)
- Ratio M-estimator
- Trimmed and winsorized estimation of the mean

Dependency: The package is based on the **survey** package (Lumley, 2004, 2011).

Author(s)

Beat Hulliger and Tobias Schoch

Maintainer: Tobias Schoch <tobias.schoch@fhnw.ch>

References

Thomas Lumley (2011). *survey: Analysis of complex survey samples*. R package version 3.26

Thomas Lumley (2004). Analysis of complex survey samples. *Journal of Statistical Software* 9(1), 1–19.

See Also

`survey`

`mer` *Minimum Estimated Risk M-Estimation*

Description

`mer` searches for the robustness-tuning parameter `k` (for M-estimation) that minimizes the (inverse-probability weighted) mean square error (MSE). Thus, MER-estimation is a strategy of adaptively choosing the optimal robustness tuning.

Usage

```
mer(object, init = 0.1, box.lo = 1e-04, tol = 1e-04)
```

Arguments

`object`	an object of the class `"svystat.rob"` (i.e. an estimate of `msvymean` with a first guess of the robustness tuning parameter `k`)
`init`	an initial value of the parameter `k` to start the search for an optimum (default 0.1)
`box.lo`	lower bound (box-constraint) on the variables for the `L-BFGS-B` method (default 1e-4; must be `>0`)
`tol`	numerical tolerance criterion (delivered to the IRLWS algorithm)

Details

mer searches for the robustness tuning parameter k (for a M-estimator) that minimizes the MSE. The function mer calls optim (in the **stats** package) to search for an optimal tuning constant k that minimizes the estimated risk function. Minimization is computed by means of the L-BFGS-B method (Byrd et al., 1995; Nocedal and Wright, 2006), i.e. a limited-memory modification of the BFGS quasi-Newton method. By default, the following box-constraints are used: lower=1e-4, upper=inf. Note that in typical applications, neither the box-constraints nor the initial value for the parameters to be optimized over, need to be adapted. The algorithm usually converges in a couple of iterations, since it capitalizes (by means of a finite-difference approximation of the gradient) on the almost quadratic shape (at least for symmetric distributions) of the MSE.

Important notice: In case of asymmetric distributions, mer-estimation tends to choose optimal tuning constants k that are far too large. Sometimes the global minimum of the MSE is at zero. In such a case, smaller k (i.e. downweighting a larger amount of observations) will always reduce the MSE and the optimal M-estimator may be, e.g., the median.

Failure of convergence: If the algorithm failed to converged, set the initial value (init) of k near the 'true' k. In addition, you may modify the numeric convergence criterion, tol.

Value

Object of the class(es) "svystat.rob" and "mer".

The following (S3) methods are defined for objects of the class "svystat.rob":

- print method,
- summary method,
- coef method,
- vcov method,
- residuals method,
- robweights method.

Author(s)

Beat Hulliger and Tobias Schoch

References

Byrd, R. H., Lu, P., Nocedal, J. and Zhu, C. (1995) A limited memory algorithm for bound constrained optimization. *SIAM J. Scientific Computing*, 16, 1190–1208.

Hulliger, B. (1995): Outlier robust Horvitz-Thompson estimators, *Survey Methodology* 21 (1), pp. 79-87.

Hulliger, B. (1999): Simple and robust estimators for sampling, *Proceedings of the Survey Research Methods Section*, American Statistical Association, 1999, pp. 54-63.

Nocedal, J. and Wright, S. J. (2006) *Numerical Optimization*, 2nd. ed. Springer.

Examples

```
## load the data
data(api)
## define "survey.design" for stratified sampling
dstrat <- svydesign(id=~1,strata=~stype, weights=~pw, data=apistrat,
fpc=~fpc)
```

```
## compute the a robust Horvitz-Thompson mean
m1 <- msvymean(~api00, dstrat, type="rht", k=1.3)
## compute the minimum estimated risk (MER) estimator based on m1
m1.mer <- mer(m1)
summary(m1.mer)
```

`msvymean` *Robust M-estimation of the mean for complex samples*

Description

`msvymean` computes robust Horvitz-Thompson estimates of the mean or robust weighted mean estimates for complex samples based on using M-estimation.

Usage

```
msvymean(y, design, k, type = "rht", na.rm = FALSE,
         control = rht.control(...), ...)
```

Arguments

`y`	a formula object (only one variable)
`design`	a `survey.design` object
`k`	robustness tuning constant
`type`	either `"rht"` for robust Horvitz-Thompson estimator (default), or `"rwm"` for robust weighted mean estimator
`na.rm`	should cases with missing values be dropped? (default `FALSE`)
`control`	control object; see `rht.control`
`...`	(additional specifications which are delivered to `rht.control`)

Details

`msvymean` performs (inverse probability-) weighted M-estimation (Huber psi-function or asymmetric Huber psi-function; `asymmetric=TRUE`). The `msvymean` methods supports the following two methods (depending on the underlying sampling design)

- robust Horvitz-Thompson estimator (`type="rht"`),
- robust weighted mean estimator (`type="rwm"`).

If `y` is positively correlated with the inclusion probabilities, a "rht" type estimator should be used, and "rwm" otherwise. The initial value is a weighted median or a ratio of weighted medians. You may set `steps` equal to one in order to get a one-step estimator. Variance estimates are computed as first-order linearization using the design-based-estimation facilities in the **survey** package.

`msvymean` allows also the estimation for domains. Use the command `subset` and a *design subset expression* instead of the original `survey.design` object in `msvymean` (see examples for more details).

Users may set `exact=TRUE` to compute an "exact" linearization-variance estimate, which takes into account that the MAD has been used as preliminary scale estimate. However, the estimates may become very unstable.

Value

Object of class "svystat.rob".

The following (S3) methods are defined for objects of the class "svystat.rob":

- print method,
- summary method,
- coef method,
- vcov method,
- residuals method,
- robweights method.

Author(s)

Beat Hulliger and Tobias Schoch

References

Hulliger, B. (1995): Outlier robust Horvitz-Thompson estimators, *Survey Methodology* 21 (1), pp. 79-87.

Hulliger, B. (1999): Simple and robust estimators for sampling, *Proceedings of the Survey Research Methods Section*, American Statistical Association, 1999, pp. 54-63.

Hulliger, B. and T. Schoch (2011): Elementary robust estimators. In: *Robust methodology for Laeken indicators: AMELI Deliverable D4.2*, ed. by B. Hulliger, A. Alfons, P. Filzmoser, A. Meraner, T. Schoch and M. Templ. AMELI Project.

See Also

svymean

Examples

```
## load "api" data set from "survey" package (a description of the data
## set can be found there)
data(api)
## define "survey.design" for stratified sampling
dstrat <- svydesign(id=~1,strata=~stype, weights=~pw, data=apistrat,
fpc=~fpc)
## compute a robust Horvitz-Thompson estimate for the mean of the
## variable "api00" (Academic Performance Index in 2000)
rht1 <- msvymean(~api00, dstrat, type="rht", k=1.2)
# get a summary of the estimation
summary(rht1)
## robust Horvitz-Thompson estimates for a domain of the variable. Here
## we are interessted in the robust mean for api00 for
## (sch.wide == "Yes"). That is the average of the academic performance
## in 2000 only for the schools that met the school-wide growth target.
msvymean(~api00, subset(dstrat, sch.wide == "Yes"), type="rht", k=1.2)
## to extract the estimate from the object
coef(rht1)
## to extract the variance from the object
vcov(rht1)
```

`msvyratio` *Robust ratio M-estimation for complex samples*

Description

`msvyratio` computes a robust ratio estimate for complex samples by means of M-estimation ("rht" type).

Usage

```
msvyratio(numerator, denominator, design, k, na.rm = FALSE,
          control = rht.control(...), ...)
```

Arguments

`numerator`	a formula object (only one variable)
`denominator`	a formula object (only one variable)
`design`	a `survey.design` object
`k`	robustness tuning constant
`na.rm`	should cases with missing values be dropped? (default `FALSE`)
`control`	control object; see `rht.control`
`...`	(additional specifications which are delivered to `rht.control`)

Details

`msvyratio` computes a robust ratio estimate for complex samples by means of M-estimation (type "rht"; see `msvymean` for more details). Variance estimates are computed as first-order linearization using the design-based estimation facilities in the **survey** package.

The initial value is a weighted median or a ratio of weighted medians. You may set `steps` equal to one in order to obtain an one-step estimator.

`msvyratio` allows also the estimation for domains. Use the command `subset` and a *design subset expression* instead of the original `survey.design` object in `msvyratio` (see examples for more details).

Value

Object of class `"svystat.rob"`

The following (S3) methods are defined for objects of the class `"svystat.rob"`:

- `print` method,
- `summary` method,
- `coef` method,
- `vcov` method,
- `residuals` method,
- `robweights` method.

Author(s)

Beat Hulliger and Tobias Schoch

References

Hulliger, B. (1995): Outlier robust Horvitz-Thompson estimators, *Survey Methodology* 21 (1), pp. 79-87.

Hulliger, B. (1999): Simple and robust estimators for sampling, *Proceedings of the Survey Research Methods Section*, American Statistical Association, 1999, pp. 54-63.

Hulliger, B. and T. Schoch (2011): Elementary robust estimators. In: *Robust methodology for Laeken indicators: AMELI Deliverable D4.2*, ed. by B. Hulliger, A. Alfons, P. Filzmoser, A. Meraner, T. Schoch and M. Templ. AMELI Project.

See Also

svyratio

Examples

```
## load "api" data set from "survey" package (a description of the data
## set can be found there)
data(api)
## define "survey.design" for stratified sampling
dstrat <- svydesign(id=~1,strata=~stype, weights=~pw, data=apistrat,
fpc=~fpc)
## compute a robust Horvitz-Thompson estimate for the mean of the
## variable api00 (Academic Performance Index in 2000)
ratio1 <- msvyratio(~api00, ~api99, dstrat, k=1.2, na.rm=TRUE)
## get a summary of the estimation
summary(ratio1)
```

---

msvytotal *Robust M-estimation of the total for complex samples*

Description

msvytotal computes robust Horvitz-Thompson estimates of the total or robust weighted total estimates for complex samples by means of M-estimation.

Usage

```
msvytotal(y, design, k, type = "rht", na.rm = FALSE,
         control = rht.control(...), ...)
```

Arguments

y	a formula object (only one variable)
design	a survey.design object
k	robustness tuning constant
type	either "rht" for robust Horvitz-Thompson estimator (default), or "rwt" for robust weighted total estimator

na.rm	should cases with missing values be dropped? (default FALSE)
control	control object; see rht.control
...	(additional specifications which are delivered to rht.control)

Details

msvytotal performs (inverse probability-) weighted M-estimation (Huber psi-function or asymmetric Huber psi-function; asymmetric=TRUE). The msvytotal methods supports the following two methods (depending on the underlying survey design)

- robust Horvitz-Thompson total estimator (type="rht"),
- robust weighted total estimator (type="rwt").

If y is positively correlated with the inclusion probabilities a "rht" type estimator should be used, and "rwm" otherwise. The initial value is a weighted median or a ratio of weighted medians. You may set steps equal to one in order to get a one-step estimation. Variance estimates are computed as first-order linearization using the design-based estimation facilities in the **survey** package.

msvytotal allows also the estimation for domains. Use the command subset and a *design subset expression* instead of the original survey.design object in msvytotal (see examples for more details).

Users may set exact=TRUE to compute an "exact" linearization-variance estimate, which takes into account that the MAD has been used as preliminary scale estimate. However, the estimates may become very unstable.

Value

Object of class "svystat.rob"

The following (S3) methods are defined for objects of the class "svystat.rob":

- print method,
- summary method,
- coef method,
- vcov method,
- residuals method,
- robweights method.

Author(s)

Beat Hulliger and Tobias Schoch

References

Hulliger, B. (1995): Outlier robust Horvitz-Thompson estimators, *Survey Methodology* 21 (1), pp. 79-87.

Hulliger, B. (1999): Simple and robust estimators for sampling, *Proceedings of the Survey Research Methods Section*, American Statistical Association, 1999, pp. 54-63.

Hulliger, B. and T. Schoch (2011): Elementary robust estimators. In: *Robust methodology for Laeken indicators: AMELI Deliverable D4.2*, ed. by B. Hulliger, A. Alfons, P. Filzmoser, A. Meraner, T. Schoch and M. Templ. AMELI Project.

See Also

svytotal

Examples

```
## load "api" data set from "survey" package (a description of the data
## set can be found there)
data(api)
## define "survey.design" for stratified sampling
dstrat <- svydesign(id=~1,strata=~stype, weights=~pw, data=apistrat,
fpc=~fpc)
## compute a robust Horvitz-Thompson estimate for the total of the
## variable "api00" (Academic Performance Index in 2000)
rht1 <- msvytotal(~api00, dstrat, k=1.2)
# get a summary of the estimation
summary(rht1)
## robust Horvitz-Thompson estimates for a domain of the variable. Here
## we are interessted in the robust total for api00 for
## (sch.wide == "Yes"). That is the average of the academic performance
## in 2000 only for the schools that met the school-wide growth target.
msvytotal(~api00, subset(dstrat, sch.wide == "Yes"), k=1.2)
## to extract the estimate from the object
coef(rht1)
## to extract the variance from the object
vcov(rht1)
```

---

pmu284 *The MU284 Population*

Description

The MU284 population comes from Sarndal, Swensson, and Wretman (1992, Appendix B) and can be downloaded from Stablib. The data consists of readings from 284 Swedish municipalites. Typically, a Swedish municipality consists of a town and the surrounding area.

Usage

```
data(pmu284)
```

Format

A data frame with 284 observations on the following 11 variables.

`LABEL` municipality identifier, running from 1 to 284

`P85` 1985 population in thousands

`P75` 1975 population on thousands

`RMT85` revenues from the 1985 municipal taxation (in millions of kronor)

`CS82` number of Conservative seats in municipal council

`SS82` number of Social-Democratic seats in municipal council

`S82` total number of seats in municipal council

`ME84` number of municipal employees in 1984

REV84 real estate values according to 1984 assessement (in millions of kronor)

REG geographic region indicator

CL cluster indicator (a cluster consists of a set of neighboring municipalities)

Details

See Sarndal, Swensson, and Wretman (1991, Appendix B) for more details

Source

Carl Erik Sarndal, Bengt Swensson, and Jan Wretman (1991). *Model Assisted Survey Sampling*. New York: Springer.

References

Full MU284 population at http://lib.stat.cmu.edu/datasets/mu284

---

rht.control *Control function for rht*

Description

This function specifies the parameters needed to compute robust estimators

Usage

```
rht.control(acc = 1e-06, steps = 50, asymmetric = FALSE,
            quietly = FALSE, exact = FALSE, ddigits = 3, ...)
```

Arguments

acc	scalar; determines the precision in the termination rule of the IRWLS algorithm
steps	scalar (integer); determines the maximal number of IRWLS iterations
asymmetric	logical; FALSE gives symmetric (standard) Huber psi-function (default); TRUE gives the asymmetric Huber psi-function
quietly	logical; default is FALSE; if quietly is TRUE, the text/message output (stdout) of all algorithms is suppressed (this is useful in simulation studies)
exact	locial; if TRUE the exact asymptotic variance of the M-estimator is computed, where exact means that the variance of the scale estimate is considered; if FALSE, the contribution of the scale variance is neglected (default).
ddigits	scalar (integer); determines the number of relevant decimal digits to be reported (default=3)
...	additional specifications

Details

The user may specify parameters differently if needed. Notably, if the algorithm fails to converge, one may consider modifying the number of iterations (steps) and/or the precision (acc).

Value

A list consisting of the parameter values.

Author(s)

Tobias Schoch

`rhte-utils` *rhte utility functions*

Description

The `rhte` package contains some useful utility functions to extract relevant information from objects of the class `"svystat.rob"`.

Usage

```
robweights(object)
## S3 method for class 'svystat.rob'
print(x, ...)
## S3 method for class 'svystat.rob'
summary(object, ...)
## S3 method for class 'svystat.rob'
coef(object, ...)
## S3 method for class 'svystat.rob'
vcov(object, ...)
## S3 method for class 'svystat.rob'
residuals(object, ...)
```

Arguments

`x`	(only used by the `print` method)
`object`	(only used by the `summary`, `coef`, and `vcov` method)
`...`	additional specifications

Details

- `summary` is a method to summarize the object
- `coef` is a method to extract the estimates (coefficients) from a `svystat.rob` object
- `vcov` is a method to extract the variance from a `svystat.rob` object
- `print` is a print method
- `residuals` extracts the residuals from a `svystat.rob` object
- `robweights` extracts the robustness weights from an M-estimate

Author(s)

Beat Hulliger and Tobias Schoch

Examples

```
## load "api" data set from "survey" package (a description of the data
## set can be found there)
data(api)
## define "survey.design" for stratified sampling
dstrat <- svydesign(id=~1,strata=~stype, weights=~pw, data=apistrat,
fpc=~fpc)
## compute a robust Horvitz-Thompson estimate for the mean of the
## variable api00 (Academic Performance Index in 2000)
rht1 <- msvymean(~api00, dstrat, type="rht", k=4)
# get a summary of the estimation
summary(rht1)
## robust Horvitz-Thompson estimates for a domain of the variable. Here
## we are interessted in the robust mean for api00 in case of
## (sch.wide == "Yes"). That is the average of the academic performance
## in 2000 only for the schools that met the school-wide growth target.
msvymean(~api00, subset(dstrat, sch.wide == "Yes"), k=4, type="rht")
## to extract the estimate from the object
coef(rht1)
## to extract the variance from the object
vcov(rht1)
```

`tsvymean` *Trimmed and winsorized weighted mean for complex samples*

Description

`tsvymean` computes either the trimmed or winsorized weighted mean for complex samples.

Usage

```
tsvymean(y, design, trim = c(0, 0), type = c("trim", "win"),
         na.rm=FALSE, control = rht.control(...), ...)
```

Arguments

`y`	a formula object (only one variable)
`design`	a `survey.design` object
`trim`	vector of size two consisting of the lower and upper amount of trimming (`[lo, hi]`). The fraction `lo` of observations is trimmed from the lower end and the fraction `hi` is trimmed from the upper end. The symmetrically 5%-trimmed mean is obtained with `trim=c(0.05, 0.05)`. Alternatively, one may specify the amount of trimming in terms of integer values; e.g., `trim=c(0, 6)` trims only the largest 6 observations.
`type`	either `"trim"` for trimming (default), or `"win"` for winsorization
`na.rm`	should cases with missing values be dropped? (default `FALSE`)
`control`	control object; see `rht.control`
`...`	additional specifications

Details

By default `trim` equals `c(0,0)` and the regular weighted mean is computed. The variance estimators are based on first-order linearizations using the design-based-estimation facilities of the **survey** package. For reasons of numerical stability, the variance of the winsorized weighted mean is computed using the variance estimator of the trimmed mean.

`tsvymean` allows also the estimation for domains. Use the command `subset` and a *design subset expression* instead of the original `survey.design` object in `tsvymean` (see examples for more details).

Value

Object of class `"svystat.rob"`

The following (S3) methods are defined for objects of the class `"svystat.rob"`:

- `print` method,
- `summary` method,
- `coef` method,
- `vcov` method.

Author(s)

Tobias Schoch

References

Hulliger, B. (1999): Simple and robust estimators for sampling, *Proceedings of the Survey Research Methods Section*, American Statistical Association, 1999, pp. 54-63.

Hulliger, B. and T. Schoch (2011): Elementary robust estimators. In: *Robust methodology for Laeken indicators: AMELI Deliverable D4.2*, ed. by B. Hulliger, A. Alfons, P. Filzmoser, A. Meraner, T. Schoch and M. Templ. AMELI Project.

See Also

`svymean`

Examples

```
## load "api" data set from "survey" package (a description of the data
## set can be found there)
data(api)
## define "survey.design" for stratified sampling
dstrat <- svydesign(id=~1,strata=~stype, weights=~pw, data=apistrat,
fpc=~fpc)
## compute a robust Horvitz-Thompson estimate for the mean of the
## variable "api00" (Academic Performance Index in 2000)
tm1 <- tsvymean(~api00, dstrat, trim=c(0.01, 0.09), type="trim")
# get a summary of the estimation
summary(tm1)
## robust estimates for a domain of the variable. Here we are
## interessted in the trimmed mean for api00 in case of
## (sch.wide == "Yes"). That is the average of the academic performance
## in 2000 only for the schools that met the school-wide growth target.
tsvymean(~api00, subset(dstrat, sch.wide == "Yes"), trim=c(0.01, 0.09),
```

```
type="trim")
## to extract the estimate from the object use
coef(tm1)
## to extract the variance from the object use
vcov(tm1)
```

Index