Package: wbsd (via r-universe)

October 21, 2026

Type Package

Title Wild Bootstrap Size Diagnostics

Version 1.0.1

Date 2026-05-15

Maintainer David Preinerstorfer <david.preinerstorfer@wu.ac.at>

Description Implements the diagnostic ``theta" developed in Poetscher and Preinerstorfer (2020) ``How Reliable are Bootstrap-based Heteroskedasticity Robust Tests?" <doi:10.48550/arXiv.2005.04089>, which appeared as <doi:10.1017/S0266466622000184> in Econometric Theory , Volume 39 , Issue 4 , August 2023 , pp. 789 - 847. The diagnostic ``theta" can be used to detect and weed out bootstrap-based procedures that provably have size equal to one for a given testing problem. The implementation covers a large variety of bootstrap-based procedures, cf. the above mentioned article for details. A function for computing bootstrap p-values is provided.

License GPL-2

Imports methods, stats, Rcpp

LinkingTo Rcpp, RcppEigen

NeedsCompilation yes

Author David Preinerstorfer [aut, cre]

Repository https://cran.r-universe.dev

Date/Publication 2026-05-15 10:41:40 UTC

RemoteUrl https://github.com/cran/wbsd

RemoteRef HEAD

RemoteSha 781e77fb040f34abc1de66b3b04a47f3dc5f3476

Contents

wbsd-package 2

boot.pval 2
theta 4

Index **7**

`wbsd-package` *Wild Bootstrap Size Diagnostics*

Description

The package **wbsd** provides functions for testing affine hypotheses on the regression coefficient vector in linear regression models with heteroskedastic errors based on the wild bootstrap. In particular, it provides (i) the function `theta` to compute the value of the diagnostic "theta" suggested in the article Pötscher and Preinerstorfer (2020); and it provides (ii) the function `boot.pval` to obtain bootstrap p-values. Various test statistics (null-restricted/unrestricted covariance estimator; classical F-test and F-tests based on HC0 - HC4 covariance estimators) are included. The functions also cover various bootstrap schemes. We refer the user to Pötscher and Preinerstorfer (2020) for: details concerning the framework, test statistics, bootstrap schemes, the definition of theta (depending on the test statistic and scheme used), the underlying theoretical results, and for further references. The paper Pötscher and Preinerstorfer (2020) was published in Econometric Theory , Volume 39 , Issue 4 , August 2023 , pp. 789 - 847.

References

Pötscher, B. M. and Preinerstorfer, D. (2020). How Reliable are Bootstrap-based Heteroskedasticity Robust Tests? <arXiv:2005.04089>

`boot.pval` *Bootstrap p-value*

Description

This function computes a bootstrap p-value as discussed in the article Pötscher and Preinerstorfer (2020).

Usage

```
boot.pval(y, X, R, r, hcmethod, restr.cov, wilddist, wildmult,
wildmult.restr, boot.res.restr, boot.center.restr,
tol = 1e-07, comp.meth = "exact", Boot.supp = NULL,
checks = TRUE, cores = 1)
```

Arguments

y	Matrix (n rows)
X	Matrix (n times k, rank k).
R	Matrix (q times k, rank q).
r	Vector (q dimensional).
hcmethod	Integer in [-1, 4]. Determines the method used in the construction of the covariance estimator used in the test statistic. The value -1 corresponds to uncorrected F statistic without df adjustment; the value 0 corresponds to the HC0 estimator; ...; the value 4 corresponds to the HC4 estimator. Cf. Pötscher and Preinerstorfer (2020) and the references there for details.
restr.cov	TRUE or FALSE. Covariance matrix estimator based on null-restricted (TRUE) or unrestricted (FALSE) residuals.
wilddist	Auxiliary distribution used to generate the bootstrap sample. Either "rademacher" or "mammen". This is only used in case comp.meth = exact.
wildmult	Integer in [0, 4]. Determines the wild-bootstrap multiplier weights used. The value 0 corresponds to constant multipliers of 1; the value 1 corresponds to multipliers that are obtained by taking the square of the HC1 weights; ...; the value 4 corresonds to multipliers that are obtained by taking the square of the HC4 weights.
wildmult.restr	TRUE or FALSE. Bootstrap multiplier weights computed on null-restricted (TRUE) or unrestricted (FALSE) projection matrix.
boot.res.restr	TRUE or FALSE. Bootstrap residuals based on null-restricted sample residuals (TRUE) or unrestricted sample residuals (FALSE)
boot.center.restr	TRUE or FALSE. Bootstrap sample centered at null-restricted predictor (TRUE) or at the unrestricted predictor (FALSE).
tol	Real number. Tolerance parameter used in checking invertibility of the covariance matrix in the test statistic. Default is 1e-07. A negativ input value will be converted to the machine epsilon. To check the stability of the numerical computations, it is recommended to try different values of tol and compare the obtained value of theta. Larger values of tol will lead to larger theta.
comp.meth	"exact" or "approximation". Determines whether probabilities over the bootstrap distribution are computed exactly or approximately (the latter requires input Boot.supp). Exact computations are only feasible for small sample sizes.
Boot.supp	Bootstrap approximation distribution (e.g., obtained by drawing vectors each with n independent entries distributed according to a Rademacher or Mammen distribution); (matrix, n rows, columns = number of bootstrap samples). This is only used if comp.meth = "approximation"; note that these vectors are further multiplied by wildmult weights.
checks	TRUE (default) or FALSE. Determines whether input checks are conducted (TRUE) or not (FALSE).
cores	Positive integer. Maximal number of cores used in the computations.

Details

For details see the relevant sections in Pötscher and Preinerstorfer (2020).

Value

p	Bootstrap p-value.

References

Pötscher, B. M. and Preinerstorfer, D. (2020). How Reliable are Bootstrap-based Heteroskedasticity Robust Tests? <arXiv:2005.04089>

Examples

```
y <- matrix(rnorm(10), ncol = 1)
X <- cbind(rep(1, 10), rnorm(10))
R <- matrix(c(0, 1), nrow = 1, ncol = 2)
r <- 0

boot.pval(y, X, R, r, 3, FALSE, "rademacher", 3, FALSE, TRUE, TRUE)
```

---

theta *theta*

Description

This function computes theta as discussed in the article Pötscher and Preinerstorfer (2020).

Usage

```
theta(
X, R, r, hcmethod, restr.cov, wilddist, wildmult,
wildmult.restr, boot.res.restr,  boot.center.restr,
tol = 1e-07, as.tol = 1e-07, in.tol = 1e-05, comp.meth = "exact",
Boot.supp = NULL, checks = TRUE, cores = 1)
```

Arguments

X	Matrix (n times k, rank k).
R	Matrix (q times k, rank q).
r	Vector (q dimensional).
hcmethod	Integer in [-1, 4]. Determines the method used in the construction of the covariance estimator used in the test statistic. The value -1 corresponds to uncorrected F statistic without df adjustment; the value 0 corresponds to the HC0 estimator; ...; the value 4 corresponds to the HC4 estimator. Cf. Pötscher and Preinerstorfer (2020) and the references there for details.

`restr.cov` TRUE or FALSE. Covariance matrix estimator based on null-restricted (TRUE) or unrestricted (FALSE) residuals.

`wilddist` Distribution used to generate the bootstrap sample. Either "rademacher" or "mammen". This is only used in case comp.meth = exact.

`wildmult` Integer in [0, 4]. Determines the wild-bootstrap multiplier weights used. The value 0 corresponds to constant multiplieres of 1; the value 1 corresponds to multipliers that are obtained by taking the square of the HC1 weights; ...; the value 4 corresonds to multipliers that are obtained by taking the square of the HC4 weights.

`wildmult.restr` TRUE or FALSE. Bootstrap multiplier weights computed on null-restricted (TRUE) or unrestricted (FALSE) projection matrix.

`boot.res.restr` TRUE or FALSE. Bootstrap residuals based on null-restricted sample residuals (TRUE) or unrestricted sample residuals (FALSE)

`boot.center.restr`
TRUE or FALSE. Bootstrap sample centered at null-restricted predictor (TRUE; or FALSE)

`tol` Real number. Tolerance parameter used in checking invertibility of the covariance matrix in the test statistic. This tolerance parameter is also used in checking $R\hat{\beta}(e_i) \neq 0$ in the computation of $\vartheta_{2,Het}$ and $\vartheta_{2,uc}$ (it is checked whether $\|R\hat{\beta}(e_i)\|_\infty > tol$. Default is 1e-07. A negativ input value will be converted to the machine epsilon. To check the stability of the numerical computations, it is recommended to try different values of tol and compare the obtained value of theta. Larger values of tol will lead to larger theta.

`as.tol` Real number. Tolerance parameter used in checking Assumptions 1 or 2. Default is 1e-07. A negativ input value will be converted to the machine epsilon. To check the stability of the numerical computations, it is recommended to try different values of as.tol and compare the obtained value of theta. Larger values of as.tol will lead to larger theta.

`in.tol` Real number. Tolerance parameter used in checking a strict inequality in the theta computation. Default is 1e-05. A negativ input value will be converted to the machine epsilon. To check the stability of the numerical computations, it is recommended to try different values of in.tol and compare the obtained value of theta. Larger values of in.tol will lead to larger theta.

`comp.meth` "exact" or "approximation". Determines whether probabilities over the bootstrap distribution are computed exactly or approximately. Exact computations are only feasible for small sample sizes.

`Boot.supp` Bootstrap approximation distribution (e.g., obtained by drawing vectors each with n independent entries distributed according to a Rademacher distribution, or a Mammen distribution); (matrix, n rows, columns = number of bootstrap samples). This is only used if comp.meth = "approximation"; note that these vectors are further multiplied by wildmult weights.

`checks` TRUE (default) or FALSE. Determines whether input checks are conducted (TRUE) or not (FALSE).

`cores` Positive integer. Maximal number of cores used in the computations.

Details

For details see the relevant sections in Pötscher and Preinerstorfer (2020).

Value

`theta`	The value of theta; in case Assumptions are not satisfied NA is returned.
`Aspt.sat`	TRUE if checks passed, FALSE if checks did not pass.
`Max.ind`	The index that led to the minimum in the computation of theta. If Astp.sat = FALSE, then Max.ind = NA. Furthermore, Max.ind = NA in case theta was determined by the sufficient condition (k = q, boot.res.restr = boot.center.restr = TRUE) for theta = 1 discussed in the paper

References

Pötscher, B. M. and Preinerstorfer, D. (2020). How Reliable are Bootstrap-based Heteroskedastiity Robust Tests? <arXiv:2005.04089>

Examples

```
v1 <- rnorm(10)
X <- cbind(rep(1, 10), rnorm(10), rnorm(10), sign(v1)*exp(v1))
R <- rbind(c(0, 0, 1, 0), c(0, 0, 0, 1))
r <- c(0, 0)

theta(X, R, r, 3, FALSE, "rademacher", 3, FALSE, TRUE, TRUE)
```

Index

boot.pval, 2

theta, 4

wbsd *(*wbsd-package*)*, 2
wbsd-package, 2