

Package ‘acrt’

December 27, 2016

Type Package

Title Autocorrelation Robust Testing

Version 1.0.1

Date 2016-12-23

Author David Preinerstorfer

Maintainer David Preinerstorfer <david.preinerstorfer@econ.au.dk>

Description Functions for testing affine hypotheses on the regression coefficient vector in regression models with autocorrelated errors.

License GPL-2

Imports methods, stats, sandwich, Rcpp

LinkingTo Rcpp, RcppEigen

NeedsCompilation yes

Repository CRAN

Date/Publication 2016-12-27 14:15:43

R topics documented:

acrt-package	2
critical.value	3
F.type.test.statistic	6
size	8

Index	13
--------------	-----------

Description

The package **acrt** provides functions for testing affine restrictions on the regression coefficient vector in linear models with autocorrelated errors. The methods implemented in **acrt** are based on the article Pötscher and Preinerstorfer (2016). In particular, the package can be used to compute various autocorrelation robust test statistics; to compute critical values that provide size control when the error process is Gaussian (or, more generally, elliptically symmetric) and autocorrelated; and to compute the size of a test that is obtained from an autocorrelation robust test statistic and a user-supplied critical value.

Details

acrt provides three functions:

1. The function `F.type.test.statistic` can be used to compute test statistics of the form T_w or $T_{E,W}$ as defined in Pötscher and Preinerstorfer (2016). The class of test statistics of the form T_w or $T_{E,W}$ includes F-type tests based on covariance estimators with data-independent bandwidth parameters and without prewhitening as considered in, e.g., Newey and West (1987), Andrews (1991), Kiefer and Vogelsang (2002, 2005), cf. also Preinerstorfer and Pötscher (2016).
2. The function `critical.value` provides an implementation of Algorithm 1 in Pötscher and Preinerstorfer (2016), and can be used to determine size-controlling critical values for test statistics of the form T_w or $T_{E,W}$.
3. The function `size` provides an implementation of Algorithm 2 in Pötscher and Preinerstorfer (2016), and can be used to determine the size of tests that are based on test statistics of the form T_w or $T_{E,W}$ and a user-supplied critical value (e.g., obtained from asymptotic theory).

We refer the user to Pötscher and Preinerstorfer (2016) for details concerning the framework, the test statistics, the algorithms, and the underlying theoretical results.

References

- Andrews, D. W. K. (1991). Heteroskedasticity and autocorrelation consistent covariance matrix estimation. *Econometrica*, **59** 817-858.
- Kiefer, N. M. and Vogelsang, T. J. (2002). Heteroskedasticity - autocorrelation robust standard errors using the Bartlett kernel without truncation. *Econometrica*, **70** 2093-2095.
- Kiefer, N. M. and Vogelsang, T. J. (2005). A new asymptotic theory for heteroskedasticity - autocorrelation robust tests. *Econometric Theory*, **21** 1130-1164.
- Newey, W. K. and West, K. D. (1987). A simple, positive semi-definite, heteroskedasticity and autocorrelation consistent covariance matrix. *Econometrica*, **55** 703-708.
- Pötscher, B.M. and Preinerstorfer, D. (2016). Controlling the size of autocorrelation robust tests. <https://arxiv.org/abs/1612.06127/>
- Preinerstorfer, D. and Pötscher, B. M. (2016). On size and power of heteroskedasticity and autocorrelation robust tests. *Econometric Theory*, **32** 261-358.

critical.value *Critical Values for Autocorrelation Robust Testing*

Description

This function provides an implementation of Algorithm 1 in Pötscher and Preinerstorfer (2016), which we here abbreviate as (A1). The user is referred to this article for definitions, a detailed description of the problem solved by (A1), and for a detailed description of the algorithm itself.

Usage

```
critical.value(alpha, ar.order.max, bandwidth, ker, R, X, N0, N1, N2, Mp, M1,
              M2, Eicker = FALSE,
              opt.method.1 = "Nelder-Mead",
              opt.method.2 = "Nelder-Mead",
              control.1 = list("reltol" = N1^(-.5), "maxit" = dim(X)[1]*20),
              control.2 = list("reltol" = N2^(-.5), "maxit" = dim(X)[1]*30),
              cores = 1, margin = rep(1, length = ar.order.max))
```

Arguments

alpha	Significance level. A real number in the interval (0, 1).
ar.order.max	Maximal order of the stationary autoregressive error process. A nonnegative integer. If ar.order.max is greater than sample size minus one, then ar.order.max is automatically replaced by sample size minus one. If ar.order.max equals sample size minus one, then the critical value approximated by (A1) controls size over the set of all stationary Gaussian error processes. If ar.order.max is set equal to 0, critical.value does not return the outcome of (A1), because no optimization is needed in that case. Instead, it draws a pseudorandom sample of size N2 from the distribution of the test statistic under the null hypothesis and returns the corresponding $1 - \alpha$ quantile.
bandwidth	Bandwidth parameter used in the construction of the test statistic. A positive real number.
ker	Kernel function used in the construction of the test statistic. ker can take one of the values "Bartlett", "Parzen", or "Quadratic Spectral". The <code>kweights</code> function is used to generate the weights. For the test statistic T_w this implies the weights used via $w(j, n) = \text{ker}(j/\text{bandwidth})$. For the test statistic $T_{E,W}$ this implies the weights matrix W via $W_{ij} = \text{ker}((i - j)/\text{bandwidth})$, cf. Pötscher and Preinerstorfer (2016) for definitions of T_w and $T_{E,W}$.
R	The restriction matrix. critical.value returns a critical value for the hypothesis $R\beta = r$. R needs to be of full row rank, and needs to have the same number of columns as X.
X	The design matrix. X needs to be of full column rank. The number of columns of X must be smaller than the number of rows of X.
N0	A positive integer. Corresponds to N_0 in the description of (A1) in Pötscher and Preinerstorfer (2016).

N1	A positive integer. Corresponds to N_1 in the description of (A1) in Pötscher and Preinerstorfer (2016). N1 should be greater than N0.
N2	A positive integer. Corresponds to N_2 in the description of (A1) in Pötscher and Preinerstorfer (2016). N2 should be greater than N1.
Mp	<p>A positive integer. Mp determines M_0 in (A1), i.e., the number of starting values chosen in Stage 0 of (A1). The way initial values are generated depends on <code>ar.order.max</code>:</p> <ol style="list-style-type: none"> 1. If <code>ar.order.max</code> is 0 the choice of Mp is irrelevant, see the description of <code>ar.order.max</code> above. 2. If <code>ar.order.max</code> is either 1 or 2, then the initial values are an i.i.d. pseudorandom sample of size Mp drawn from a distribution on the set of partial autocorrelation coefficients that induces a uniform distribution on the set of stationary autoregressive coefficients of order <code>ar.order.max</code> (cf. Jones (1987)). 3. If <code>ar.order.max</code> is greater than 2, then starting values are generated as follows: Let A denote the set of integer multiples of 5 that are greater than 2 and smaller than <code>ar.order.max</code>. For every number $l \in A \cup \{2, ar.order.max\}$ the algorithm in Jones (1987) is used to generate a pseudorandom sample of size Mp on the set of partial autocorrelation coefficients from a distribution that induces a uniform distribution on the set of stationary AR(l) coefficients. These samples are then used as initial values, cf. also the discussion in Pötscher and Preinerstorfer (2016).
M1	A positive integer. Corresponds to M_1 in the description of (A1) in Pötscher and Preinerstorfer (2016). M1 must not exceed Mp.
M2	A positive integer. Corresponds to M_2 in the description of (A1) in Pötscher and Preinerstorfer (2016). M2 must not exceed M1.
Eicker	Determines the test statistic used. If <code>Eicker = TRUE</code> , then critical values for $T_{E,W}$ (with $W_{ij} = ker((i-j)/bandwidth)$) are computed. If <code>Eicker = FALSE</code> , then critical values for T_w (with $w(j,n) = ker(j/bandwidth)$) are computed (cf. Pötscher and Preinerstorfer (2016) for a precise definition of these test statistics). Default is <code>Eicker = FALSE</code> .
opt.method.1	The optimization method chosen in Stage 1 of (A1). Any optimization routine implemented in <code>optim</code> apart from "Brent" can be supplied. The default is "Nelder-Mead".
opt.method.2	The optimization method chosen in Stage 2 of (A1). Any optimization routine implemented in <code>optim</code> apart from "Brent" can be supplied. The default is "Nelder-Mead".
control.1	Control parameters passed to the <code>optim</code> function in Stage 1 of (A1). Default is <code>control.1 = list("reltol" = N1^(-.5), "maxit" = dim(X)[1]*20)</code> .
control.2	Control parameters passed to the <code>optim</code> function in Stage 2 of (A1). Default is <code>control.2 = list("reltol" = N2^(-.5), "maxit" = dim(X)[1]*30)</code> .
cores	The number of CPU cores used in the (parallelized) computation of the Monte-Carlo approximations in (A1). Default is 1. Parallelized computation is enabled only if the compiler used to build <code>acrt</code> supports OpenMP.

`margin` The restrictions imposed on the partial autocorrelation coefficients. `margin` is an `ar.order.max`-dimensional vector of real numbers in $(0, 1]$. Default is `margin = rep(1, length = ar.order.max)`, which corresponds to no restriction.

Details

For details see the relevant sections in Pötscher and Preinerstorfer (2016), in particular the description of Algorithm 1 in the Appendix.

Value

The output of `critical.value` depends on `ar.order.max`. If `ar.order.max` is zero, the function `critical.value` returns a list consisting of:

`critical.value` The critical value obtained by drawing a pseudorandom sample of size `N2` from the distribution of the test statistic under the null hypothesis and by computing the corresponding $1 - \alpha$ quantile.

If `ar.order.max` is greater than zero, the function `critical.value` returns a list consisting of:

`starting.parameters`

The rows of this matrix are the initial values (partial autocorrelation coefficient vectors) that were used in Stage 1 of (A1), and which were chosen from the pool of randomly generated initial values in Stage 0. The rows correspond to $\rho_{1:M_0}, \dots, \rho_{M_1:M_0}$, respectively, in the description of (A1).

`starting.quantiles`

Monte Carlo approximations of the $1 - \alpha$ quantiles corresponding to the initial values used in Stage 1 of (A1). The coordinates of this vector correspond to $\bar{F}_{1:M_0}^{-1}(1 - \alpha), \dots, \bar{F}_{M_1:M_0}^{-1}(1 - \alpha)$ in the description of (A1).

`first.stage.parameters`

The rows of this matrix are the parameters (partial autocorrelation coefficients) that were obtained in Stage 1 of (A1). The rows correspond to $\rho_1^*, \dots, \rho_{M_1}^*$, respectively, in the description of (A1).

`first.stage.quantiles`

Monte Carlo approximations of the $1 - \alpha$ quantiles corresponding to the `first.stage.parameters`, i.e., $\bar{F}_{j,\rho_j^*}^{-1}(1 - \alpha)$ for $j = 1, \dots, M_1$ in the description of (A1).

`second.stage.parameters`

The rows of this matrix are the parameters (partial autocorrelation coefficients) that were obtained in Stage 2 of (A1). The rows correspond to $\rho_1^{**}, \dots, \rho_{M_2}^{**}$, respectively, in the description of (A1).

`second.stage.quantiles`

Monte Carlo approximations of the $1 - \alpha$ quantiles corresponding to the `second.stage.parameters`, i.e., $\bar{F}_{j,\rho_j^{**}}^{-1}(1 - \alpha)$ for $j = 1, \dots, M_2$ in the description of (A1).

`convergence`

Convergence codes returned from `optim` in Stage 2 of (A1) for each initial value.

`critical.value` The critical value obtained by (A1), i.e., the maximum of $\bar{F}_{j,\rho_j^{**}}^{-1}(1 - \alpha)$ for $j = 1, \dots, M_2$.

References

Jones, M. C. (1987). Randomly choosing parameters from the stationarity and invertibility region of autoregressive-moving average models. *Applied Statistics*, **36** 134-138.

Pötscher, B.M. and Preinerstorfer, D. (2016). Controlling the size of autocorrelation robust tests. <https://arxiv.org/abs/1612.06127/>

See Also

[optim,kweights](#).

Examples

```
## Not run:
n <- 100
alpha <- .05
ar.order.max <- n-1
bandwidth <- 10
ker <- "Bartlett"
R <- matrix(c(0, 1), nrow = 1, ncol = 2)
X <- cbind(rep(1, length = n), rnorm(n))
N0 <- 1000
N1 <- 10000
N2 <- 50000
Mp <- 5000
M1 <- 10
M2 <- 2

critical.value(alpha, ar.order.max, bandwidth, ker, R, X, N0, N1, N2, Mp, M1,
              M2)

## End(Not run)
```

F.type.test.statistic *Computation of F-type test statistics*

Description

This function computes test statistics of the form T_w and $T_{E,W}$ as defined in Pötscher and Preinerstorfer (2016). The weights for T_w and for $T_{E,W}$ are obtained from a kernel function (Bartlett, Parzen, or Quadratic Spectral kernel, which provide nonnegative definite covariance estimators) and a bandwidth parameter. See also the description of the argument `ker` below for further details concerning the weights. The class of test statistics of the form T_w or $T_{E,W}$ includes F-type tests based on covariance estimators with data-independent bandwidth parameters and without prewhitening as considered in, e.g., Newey and West (1987), Andrews (1991), Kiefer and Vogelsang (2002, 2005), cf. also Preinerstorfer and Pötscher (2016).

Usage

```
F.type.test.statistic(y, R, r, X, bandwidth, ker, Eicker = FALSE, cores = 1)
```

Arguments

y	Either an observation vector, or a matrix the columns of which are observation vectors. The number of rows of an observation vector must coincide with the number of rows of the design matrix X.
R	The restriction matrix. <code>F.type.test.statistic</code> computes a test statistic for the hypothesis $R\beta = r$. R needs to be of full row rank, and needs to have the same number of columns as X.
r	The restriction vector. <code>F.type.test.statistic</code> computes a test statistic for the hypothesis $R\beta = r$. r needs to be a vector with the same number of coordinates as the number of rows of R.
X	The design matrix. X needs to be of full column rank. The number of columns of X must be smaller than the number of rows of X.
bandwidth	Bandwidth parameter used in the construction of the test statistic. A positive real number.
ker	Kernel function used in the construction of the test statistic. ker can take one of the values "Bartlett", "Parzen", or "Quadratic Spectral". The <code>kweights</code> function is used to generate the weights. For the test statistic T_w this implies the weights used via $w(j, n) = \text{ker}(j/\text{bandwidth})$. For the test statistic $T_{E,W}$ this implies the weights matrix W via $W_{ij} = \text{ker}((i - j)/\text{bandwidth})$, cf. Pötscher and Preinerstorfer (2016) for definitions of T_w and $T_{E,W}$.
Eicker	Determines the test statistic computed. If <code>Eicker = TRUE</code> , then $T_{E,W}$ (with $W_{ij} = \text{ker}((i - j)/\text{bandwidth})$) is computed on the input observation vector(s) y. If <code>Eicker = FALSE</code> , then T_w (with $w(j, n) = \text{ker}(j/\text{bandwidth})$) is computed on the input observation vector(s) y (cf. Pötscher and Preinerstorfer (2016) for a precise definition of these test statistics). Default is <code>Eicker = FALSE</code> .
cores	The number of CPU cores used in the (parallelized) computation of the test statistics. Default is 1. This can be used to speed up the computation in case y is a matrix with many columns. Parallelized computation is enabled only if the compiler used to build acrt supports OpenMP.

Details

For details concerning the test statistics please see the relevant sections in Pötscher and Preinerstorfer (2016) .

Value

The function returns a list consisting of:

test.val	Either a vector the entries of which correspond to the values of the test statistic evaluated at each column of the input matrix y, or, if y is a vector, the test statistic evaluated at y.
----------	--

References

- Andrews, D. W. K. (1991). Heteroskedasticity and autocorrelation consistent covariance matrix estimation. *Econometrica*, **59** 817-858.
- Kiefer, N. M. and Vogelsang, T. J. (2002). Heteroskedasticity - autocorrelation robust standard errors using the Bartlett kernel without truncation. *Econometrica*, **70** 2093-2095.
- Kiefer, N. M. and Vogelsang, T. J. (2005). A new asymptotic theory for heteroskedasticity - autocorrelation robust tests. *Econometric Theory*, **21** 1130-1164.
- Newey, W. K. and West, K. D. (1987). A simple, positive semi-definite, heteroskedasticity and autocorrelation consistent covariance matrix. *Econometrica*, **55** 703-708.
- Pötscher, B.M. and Preinerstorfer, D. (2016). Controlling the size of autocorrelation robust tests. <https://arxiv.org/abs/1612.06127/>
- Preinerstorfer, D. and Pötscher, B. M. (2016). On size and power of heteroskedasticity and autocorrelation robust tests. *Econometric Theory*, **32** 261-358.

See Also

[kweights](#).

Examples

```
n <- 100
y <- rnorm(n)
X <- cbind(rep(1, length = n), rnorm(n))
R <- matrix(c(0, 1), nrow = 1, ncol = 2)
r <- 0
bandwidth <- n/10
ker <- "Bartlett"
F.type.test.statistic(y, R, r, X, bandwidth, ker)
```

size

Computing the Size of Autocorrelation Robust Tests

Description

This function provides an implementation of Algorithm 2 in Pötscher and Preinerstorfer (2016), which we here abbreviate as (A2). The user is referred to this article for definitions, a detailed description of the problem solved by (A2), and for a detailed description of the algorithm itself.

Usage

```
size(C, ar.order.max, bandwidth, ker, R, X, N0, N1, N2, Mp, M1,
     M2, Eicker = FALSE,
     opt.method.1 = "Nelder-Mead",
     opt.method.2 = "Nelder-Mead",
     control.1 = list("reltol" = N1^(-.5), "maxit" = dim(X)[1]*20),
     control.2 = list("reltol" = N2^(-.5), "maxit" = dim(X)[1]*30),
     cores = 1, margin = rep(1, length = ar.order.max))
```


Arguments

C	Critical value. A positive real number (for negative critical values the size of the test equals 1).
ar.order.max	Maximal order of the stationary autoregressive error process. A nonnegative integer. If ar.order.max is greater than sample size minus one, then ar.order.max is automatically replaced by sample size minus one. If ar.order.max equals sample size minus one, then the size approximated by (A2) is with respect to the set of all stationary Gaussian error processes. If ar.order.max is set equal to 0, size does not return the outcome of (A2), because no optimization is needed in that case. Instead, it draws a pseudorandom sample of size N_2 from the distribution of the test statistic under the null hypothesis and returns the fraction of observations that give rise to a test statistic not smaller than the critical value C.
bandwidth	Bandwidth parameter used in the construction of the test statistic. A positive real number.
ker	Kernel function used in the construction of the test statistic. ker can take one of the values "Bartlett", "Parzen", or "Quadratic Spectral". The <code>kweights</code> function is used to generate the weights. For the test statistic T_w this implies the weights used via $w(j, n) = \text{ker}(j/\text{bandwidth})$. For the test statistic $T_{E,W}$ this implies the weights matrix W via $W_{ij} = \text{ker}((i - j)/\text{bandwidth})$, cf. Pötscher and Preinerstorfer (2016) for definitions of T_w and $T_{E,W}$.
R	The restriction matrix. size computes the size of a test for the hypothesis $R\beta = r$. R needs to be of full row rank, and needs to have the same number of columns as X.
X	The design matrix. X needs to be of full column rank. The number of columns of X must be smaller than the number of rows of X.
N0	A positive integer. Corresponds to N_0 in the description of (A2) in Pötscher and Preinerstorfer (2016).
N1	A positive integer. Corresponds to N_1 in the description of (A2) in Pötscher and Preinerstorfer (2016). N1 should be greater than N0.
N2	A positive integer. Corresponds to N_2 in the description of (A2) in Pötscher and Preinerstorfer (2016). N2 should be greater than N1.
Mp	A positive integer. Mp determines M_0 in (A2), i.e., the number of starting values chosen in Stage 0 of (A2). The way initial values are generated depends on ar.order.max: <ol style="list-style-type: none"> 1. If ar.order.max is 0 the choice of Mp is irrelevant, see the description of ar.order.max above. 2. If ar.order.max is either 1 or 2, then the initial values are an i.i.d. pseudorandom sample of size Mp drawn from a distribution on the set of partial autocorrelation coefficients that induces a uniform distribution on the set of stationary autoregressive coefficients of order ar.order.max (cf. Jones (1987)). 3. If ar.order.max is greater than 2, then starting values are generated as follows: Let A denote the set of integer multiples of 5 that are greater than 2 and smaller than ar.order.max. For every number

$$l \in A \cup \{2, \text{ar.order.max}\}$$

the algorithm in Jones (1987) is used to generate a pseudorandom sample of size M_p on the set of partial autocorrelation coefficients from a distribution that induces a uniform distribution on the set of stationary $AR(l)$ coefficients. These samples are then used as initial values, cf. also the discussion in Pötscher and Preinerstorfer (2016).

M1	A positive integer. Corresponds to M_1 in the description of (A2) in Pötscher and Preinerstorfer (2016). M1 must not exceed M_p .
M2	A positive integer. Corresponds to M_2 in the description of (A2) in Pötscher and Preinerstorfer (2016). M2 must not exceed M1.
Eicker	Determines the test for which the size is computed. If <code>Eicker = TRUE</code> , then the size of the test that rejects if $T_{E,W}$ is not smaller than C (with $W_{ij} = \ker((i - j)/bandwidth)$) is computed. If <code>Eicker = FALSE</code> , then the size of the test that rejects if T_w is not smaller than C (with $w(j, n) = \ker(j/bandwidth)$) is computed (cf. Pötscher and Preinerstorfer (2016) for a precise definition of these test statistics). Default is <code>Eicker = FALSE</code> .
opt.method.1	The optimization method chosen in Stage 1 of (A2). Any optimization routine implemented in <code>optim</code> apart from "Brent" can be supplied. The default is "Nelder-Mead".
opt.method.2	The optimization method chosen in Stage 2 of (A2). Any optimization routine implemented in <code>optim</code> apart from "Brent" can be supplied. The default is "Nelder-Mead".
control.1	Control parameters passed to the <code>optim</code> function in Stage 1 of (A2). Default is <code>control.1 = list("reltol" = N1^(-.5), "maxit" = dim(X)[1]*20)</code> .
control.2	Control parameters passed to the <code>optim</code> function in Stage 2 of (A2). Default is <code>control.2 = list("reltol" = N2^(-.5), "maxit" = dim(X)[1]*30)</code> .
cores	The number of CPU cores used in the (parallelized) computation of the Monte-Carlo approximations in (A2). Default is 1. Parallelized computation is enabled only if the compiler used to build <code>acrt</code> supports OpenMP.
margin	The restrictions imposed on the partial autocorrelation coefficients. <code>margin</code> is an <code>ar.order.max</code> -dimensional vector of real numbers in $(0, 1]$. Default is <code>margin = rep(1, length = ar.order.max)</code> , which corresponds to no restriction.

Details

For details see the relevant sections in Pötscher and Preinerstorfer (2016), in particular the description of Algorithm 2 in the Appendix.

Value

The output of `size` depends on `ar.order.max`. If `ar.order.max` is zero, the function `size` returns a list consisting of:

size	The size obtained by drawing a pseudorandom sample of size N_2 from the distribution of the test statistic under the null hypothesis and by computing the fraction of observations giving rise to a test statistic not smaller than the critical value C .
------	--

If `ar.order.max` is greater than zero, the function `size` returns a list consisting of:

`starting.parameters`

The rows of this matrix are the initial values (partial autocorrelation coefficient vectors) that were used in Stage 1 of (A2), and which were chosen from the pool of randomly generated initial values in Stage 0. The rows correspond to $\rho_{1:M_0}, \dots, \rho_{M_1:M_0}$, respectively, in the description of (A2).

`starting.rejection.probs`

Monte Carlo approximations of the null-rejection probabilities corresponding to the initial values used in Stage 1 of (A2). The coordinates of this vector correspond to $\tilde{p}_{1:M_0}, \dots, \tilde{p}_{M_1:M_0}$ in the description of (A2).

`first.stage.parameters`

The rows of this matrix are the parameters (partial autocorrelation coefficients) that were obtained in Stage 1 of (A2). The rows correspond to $\rho_1^*, \dots, \rho_{M_1}^*$, respectively, in the description of (A2).

`first.stage.rejection.probs`

Monte Carlo approximations of the null-rejection probabilities corresponding to the `first.stage.parameters`, i.e., \bar{p}_{j,ρ_j^*} for $j = 1, \dots, M_1$ in the description of (A2).

`second.stage.parameters`

The rows of this matrix are the parameters (partial autocorrelation coefficients) that were obtained in Stage 2 of (A2). The rows correspond to $\rho_1^{**}, \dots, \rho_{M_2}^{**}$, respectively, in the description of (A2).

`second.stage.rejection.probs`

Monte Carlo approximations of the null rejection probabilities corresponding to the `second.stage.parameters`, i.e., $\bar{p}_{j,\rho_j^{**}}$ for $j = 1, \dots, M_2$ in the description of (A2).

`convergence`

Convergence codes returned from `optim` in Stage 2 of (A2) for each initial value.

`size`

The size computed by (A2), i.e., the maximum of $\bar{p}_{j,\rho_j^{**}}$ for $j = 1, \dots, M_2$.

References

Jones, M. C. (1987). Randomly choosing parameters from the stationarity and invertibility region of autoregressive-moving average models. *Applied Statistics*, **36** 134-138

Pötscher, B.M. and Preinerstorfer, D. (2016). Controlling the size of autocorrelation robust tests. <https://arxiv.org/abs/1612.06127/>

See Also

[optim](#), [kweights](#).

Examples

```
## Not run:
n <- 100
C <- 2.260568^2
ar.order.max <- n-1
```

```
bandwidth <- n/10
ker <- "Bartlett"
R <- matrix(c(0, 1), nrow = 1, ncol = 2)
X <- cbind(rep(1, length = n), rnorm(n))
N0 <- 1000
N1 <- 10000
N2 <- 50000
Mp <- 5000
M1 <- 10
M2 <- 2

size(C, ar.order.max, bandwidth, ker, R, X, N0, N1, N2, Mp, M1, M2)

## End(Not run)
```

Index

acrt (acrt-package), [2](#)
acrt-package, [2](#)
critical.value, [3](#)
F.type.test.statistic, [6](#)
kweights, [3](#), [6-9](#), [11](#)
optim, [4-6](#), [10](#), [11](#)
size, [8](#)