Package ‘robcp’

June 4, 2019

Title  Robust Change-Point Tests
Version  0.2.4
Description  Provides robust methods to detect change-points in uni- or multivariate time series. They can cope with corrupted data and heavy tails. One can detect changes in location, scale and dependence structure of a possibly multivariate time series. Procedures are based on Huberized versions of CUSUM tests proposed in Duerre and Fried (2019) <arXiv:1905.06201>.

Depends  R (>= 3.3.1)
License  GPL-3
Encoding  UTF-8
LazyData  true
NeedsCompilation  yes
Author  Sheila Goerz [aut, cre],
        Alexander Duerre [ctb]
Maintainer  Sheila Goerz <sheila.goerz@tu-dortmund.de>
Repository  CRAN
Date/Publication  2019-06-03 23:50:14 UTC

R topics documented:

huber_cusum .................................................. 2
h_cumsum .................................................. 3
modifChol .................................................. 4
pKSdist .................................................... 5
psi ......................................................... 7
sigma2 ....................................................... 9
teststat ................................................... 10
zeros ...................................................... 11

Index  12
huber_cusum

**Huberized CUSUM test**

**Description**

Performs a CUSUM test on data transformed by \( \psi \). Depending in the chosen psi-function different types of changes can be detected.

**Usage**

```
huber_cusum(x, fun = "hlm", tol = 1e-8, b_n, k, constant)
```

**Arguments**

- `x`: numeric vector containing a single time series or a numeric matrix containing multiple time series (column-wise).
- `fun`: character string specifying the transformation function \( \psi \), see details.
- `tol`: tolerance of the distribution function (numeric), which is used to compute p-values.
- `b_n`: bandwidth, which is used to estimate the long run variance, see the help page of \( \sigma^2 \) for details.
- `k`: numeric bound used in \( \psi \).
- `constant`: scale factor of the MAD. Default is 1.4826.

**Details**

The function performs a Huberized CUSUM test. First the data is transformed by a suitable psi-function. To detect changes in location one can apply `fun = "hlm", "hlg", "vlm"` or `"vlg"`, for changes in scale `fun = "hcm"` is available and for changes in the dependence respectively covariance structure `fun = "hcm", "hcg", "vcm"` and `"vcg"` are possible. The actual definitions of the psi-functions can be found in the help page of `psi`. Denote \( Y_1, \ldots, Y_n \) the transformed time series. If \( Y_1 \) is one dimensional, then the test statistic

\[
V = \max_{k=1, \ldots, n} \frac{1}{\sqrt{n}\sigma} \left| \sum_{i=1}^{k} Y_i - \frac{k}{n} \sum_{i=1}^{n} Y_i \right|
\]

is calculated, where \( \sigma^2 \) is an estimator for the long run variance, see the help function of `sigma2` for details. \( V \) is asymptotically Kolmogorov-Smirnov distributed. We use a finite sample correction \( V + 0.58/\sqrt{n} \) to improve finite sample performance.

If \( Y[1] \) is multivariate, then the test statistic

\[
W = \max_{k=1, \ldots, n} \frac{1}{n} \left( \sum_{i=1}^{k} Y_i - \frac{k}{n} \sum_{i=1}^{n} Y_i \right)^T \Sigma^{-1} \left( \sum_{i=1}^{k} Y_i - \frac{k}{n} \sum_{i=1}^{n} Y_i \right)
\]

is computed, where \( \Sigma \) is the long run covariance, see also `sigma2` for details. \( W \) is asymptotically distributed like the maximum of a squared Bessel bridge. We use the identity derived in Kiefer to derive p-values. Like in the one dimensional case we use a finite sample correction \( (\sqrt{W} + 0.58/\sqrt{n})^2 \).
**Value**

A list of the class "htest" containing the following components:

- **statistic**: value of the test statistic (numeric).
- **p.value**: p-value (numeric).
- **alternative**: alternative hypothesis (character string).
- **method**: name of the performed test (character string).
- **data.name**: name of the data (character string).

**Author(s)**

Sheila Görz

**References**


**See Also**

`sigma2`, `psi`, `h_cumsum`, `teststat`, `pKsdist`

**Examples**

```r
set.seed(1895)

# time series with a structural break at t = 20
Z <- c(rnorm(20, 0), rnorm(20, 2))
huber_cumsum(Z)

# two time series with a structural break at t = 20
timeSeries <- matrix(c(rnorm(20, 0), rnorm(20, 2), rnorm(20, 1), rnorm(20, 3), ncol = 2))

huber_cumsum(timeSeries)
```

---

**h_cumsum**  
*Cumulative sum of transformed vectors*

**Description**

Computes the cumulative sum of a transformed numeric vector or matrix. Default transformation is `psi`. 
modifChol

Usage

modifChol(x, tau = .Machine$double.eps^(1 / 3),
          tau_bar = .Machine$double.eps^(2 / 3), mu = 0.1)

modifChol

Revised Modified Cholesky Factorization

Description

Computes the revised modified Cholesky factorization described in Schnabel and Eskow (1999).

Usage

modifChol(x, tau = .Machine$double.eps^(1 / 3),
          tau_bar = .Machine$double.eps^(2 / 3), mu = 0.1)
pKSDist

Arguments

- **x**: a symmetric matrix.
- **tau**: (machine epsilon)^{(1/3)}.
- **tau_bar**: (machine epsilon^{(2/3)}).
- **mu**: numeric, 0 < µ ≤ 1.

Details

`modif.chol` computes the revised modified Cholesky Factorization of a symmetric, not necessarily positive definite matrix \( x + E \) such that \( LL' = x + E \) for \( E \geq 0 \).

Value

Lower triangular matrix \( L \) of the form \( LL' = x + E \). The attribute `swaps` is a vector of the length of dimension of \( x \). It contains the indices of the rows and columns that were swapped in \( x \) in order to compute the modified Cholesky factorization. For example if the i-th element of `swaps` is the number j, then the i-th and the j-th row and column were swapped. To reconstruct the original matrix `swaps` has to be read backwards.

Author(s)

Sheila Görz

References


Examples

```r
y <- matrix(runif(9), ncol = 3)
x <- psi(y)
modifChol(sigma2(x))
```

---

pKSDist  

Asymptotic cumulative distribution for the Huberized CUSUM Test statistic

Description

Computes the asymptotic cumulative distribution of the statistic of `teststat`.

Usage

```
pKSDist(tn, tol = 1e-8)
pbessel3(tn, h)
```
Arguments

- **tn**: vector of test statistics (numeric). For pbessel13 length of tn has to be 1.
- **h**: dimension of time series (integer). If h is equal to 1 pbessel13 uses pKS2 to compute the corresponding probability.
- **tol**: tolerance (numeric).

Details

For a single time series, the distribution is the same distribution as in the two sample Kolmogorov Smirnov Test, namely the distribution of the maximal value of the absolute values of a Brownian bridge. It is computed as follows (van Mulbregt, 2018):

For \( t_n(x) < 1 \):

\[
P(t_n(X) \leq t_n(x)) = \sqrt{2 \pi / t_n(x)} \times t(1 + t^8(1 + t^{16}(1 + t^{24}(1 + ...))))
\]

up to \( t^{8k_{max}}, k_{max} = \lfloor \sqrt{2 - \log(tol)} \rfloor \) where \( t = \exp(-\pi^2/(8 * x^2)) \)

else:

\[
P(t_n(X) \leq t_n(x)) = 2 \sum_{k=1}^{\infty} (-1)^{k-1} \times \exp(-2 * k^2 * x^2)
\]

until \( |2 \times (-1)^{k-1} \times \exp(-2 * k^2 * x^2) - 2 \times (-1)^{(k-1)-1} \times \exp(-2 * (k-1)^2 * x^2)| \leq tol \).

In case of multiple time series, the distribution equals that of the maximum of an \( h \) dimensional squared Bessel bridge. It can be computed by (Kiefer, 1959):

\[
P(t_n(X) \leq t_n(x)) = \frac{4}{\Gamma(h/2)} \times \sum_{i=1}^{\infty} \left( (\gamma(h-2)/2, n)^{h-2} \times \exp(-\gamma(h-2)/2, n)^2/(2t_n^2) \right) / J_{h/2}(\gamma(h-2)/2, n) \]

where \( J_h \) is the Bessel function of first kind and \( h \)-th order, \( \Gamma \) is the gamma function and \( \gamma_{h,n} \) denotes the n-th zero of \( J_h \).

Value

vector of \( P(t_n(X) \leq t_n[i]) \).

Author(s)

Sheila Görz, Alexander Dürre

References


psi

See Also

psi, teststat, h_cumsum, huber_cusum

Examples

# single time series
timeSeries <- c(rnorm(20, 0), rnorm(20, 2))
 tn <- teststat(timeSeries)

pKSDist(tn)

# two time series
timeSeries <- matrix(c(rnorm(20, 0), rnorm(20, 2), rnorm(20, 1), rnorm(20, 3),
 ncol = 2))
tn <- teststat(timeSeries)

pbessel3(tn, 2)

---

psi Transformation of time series

Description

Computation of values transformed by their median, MAD and a ψ function.

Usage

psi(y, fun = "HLM", k, constant = 1.4826)

Arguments

y vector or matrix with each column representing a time series (numeric).
fun character string specifying the transformation function ψ.
k numeric bound used for Huber type psi-functions which determines robustness
and efficiency of the test. Default for psi = "HgL" or "HCG" is \sqrt{qchisq(0.8, df = m)}
where m are the number of time series, and otherwise it is 1.5.
constant scale factor of the MAD.

Details

Let x = (y - Median(y))/MAD(y) be the standardized values of a single time series.

Available ψ functions are:

marginal Huber for location:
fun = "HLM".
\[ \psi_{HLm}(x) = k \ast 1_{x > k} + z \ast 1_{-k \leq x \leq k} - k \ast 1_{x < -k}. \]

global Huber for location:
\[ \text{fun} = "HLg". \]
\[ \psi_{HLg}(x) = x \ast 1_{0 \leq |x| \leq k} + k \ast x / |x| \ast 1_{|x| > k}. \]

marginal sign for location:
\[ \text{fun} = "Vlm". \]
\[ \psi_{Vlm}(x_i) = \text{sign}(x_i). \]

global sign for location:
\[ \text{fun} = "Vlg". \]
\[ \psi_{Vlg}(x) = x / |x| \ast 1_{|x| > 0}. \]

marginal Huber for covariance:
\[ \text{fun} = "Hcm". \]
\[ \psi_{Hcm}(x) = \psi_{HLm}(x) \psi_{HLm}(x)^T. \]

global Huber for covariance:
\[ \text{fun} = "Hcg". \]
\[ \psi_{Hcg}(x) = \psi_{HLg}(x) \psi_{HLg}(x)^T. \]

marginal sign covariance:
\[ \text{fun} = "Vcm". \]
\[ \psi_{Vcm}(x) = \psi_{Vlm}(x) \psi_{Vlm}(x)^T. \]

global sign covariance:
\[ \text{fun} = "Vcg". \]
\[ \psi_{Vcg}(x) = \psi_{Vcg}(x) \psi_{Vcg}(x)^T. \]

Note that for all covariances only the upper diagonal is used and turned into a vector. In case of the marginal sign covariance, the main diagonal is also left out. At the global sign covariance matrix the last element of the main diagonal is left out.

**Value**

Transformed numeric vector or matrix with the same number of rows as \( y \).

**Author(s)**

Sheila Görz

**See Also**

h_cumsum, teststat
**Examples**

```r
psi(rnorm(100))
```

---

**Description**

Estimates the long run variance respectively covariance matrix of the supplied time series.

**Usage**

```r
sigma2(x, b_n)
```

**Arguments**

- `x`: vector or matrix with each column representing a time series (numeric).
- `b_n`: Must be greater than 0. default is \( n^{1/3} \) with \( n \) being the number of observations.

**Details**

The long run variance equals \( n \) times the asymptotic variance of the arithmetic mean of a short range dependent time series, where \( n \) is the length of the time series. It is used to standardize CUSUM Tests.

The long run variance is estimated by a kernel estimator using the bandwidth \( b_n = n^{1/3} \) and the flat top kernel

\[
k(x) = x \cdot 1_{|x| < 0.5} + (2 - |x|) \cdot 1_{0.5 < |x| < 1}
\]

. In the one dimensional case this results in:

\[
\hat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^{n} \left( x[i] - \text{mean}(x) \right)^2 + 2 \frac{1}{n} \sum_{h=1}^{n-h} \sum_{i=1}^{n-h} \left( x[i] - \text{mean}(x) \right) \left( x[i+h] - \text{mean}(x) \right) k(h/b_n).
\]

If \( x \) is a multivariate timeseries the \( k, l \)-element of \( \Sigma \) is estimated by

\[
\hat{\Sigma}^{(k,l)} = \frac{1}{n} \sum_{i,j=1}^{n} \left( x[i]^{(k)} - \text{mean}(x)^{(k)} \right) \left( x[j]^{(l)} - \text{mean}(x)^{(l)} \right) k((i-j)/b_n).
\]

**Value**

- long run variance \( \sigma^2 \) respectively \( \Sigma \) (numeric)

**Author(s)**

Sheila Görz
teststat

Test statistic for the Huberized CUSUM Test

Description
Computes the test statistic for a structural break test called 'Huberized CUSUM Test'.

Usage
teststat(y, fun = "hlm", b_n, k, constant)

Arguments
y vector or matrix with each column representing a time series (numeric).
fun character string specifying the transformation function \( \psi \).
b_n for \( \sigma_2 \).
k numeric bound used in \( \psi \).
constant scale factor of the MAD. Default is 1.4826.

Details
\( y \) is transformed by \( \text{fun} \). Let \( x \) be the resulting vector or matrix and \( n \) be the length of a time series.
In case of a vector the test statistic can be written as
\[
\max_{k=1,\ldots,n} \frac{1}{\sqrt{n} \hat{\sigma}} \left| \sum_{i=1}^{k} x_i - \left( \frac{k}{n} \right) \sum_{i=1}^{n} x_i \right|,
\]
where \( \sigma \) is the square root of \( \sigma_2 \).
In case of a matrix the test statistic follows as
\[
\max_{k=1,\ldots,n} \frac{1}{n} \left( \sum_{i=1}^{k} X_i - \left( \frac{k}{n} \right) \sum_{i=1}^{n} X_i \right)^T \Sigma^{-1} \left( \sum_{i=1}^{k} X_i - \left( \frac{k}{n} \right) \sum_{i=1}^{n} X_i \right),
\]
where \( X_i \) denotes the \( i \)th row of \( x \) and \( \Sigma^{-1} \) is the inverse of \( \sigma_2 \).

Value
test statistic (numeric value).

See Also
psi, h_cumsum, teststat, pKdist, huber_cusum

Examples
Z <- c(rnorm(20), rnorm(20, 2))
sigma2(Z)
## zeros

### Author(s)
Sheila Görz

### See Also
- `h_cumsum`
- `psi`

### Examples

```r
# time series with structural break at t = 20
ts <- c(rnorm(20, 0), rnorm(20, 2))
teststat(ts)
```

### Description
Contains the zeros of the Bessel function of first kind.

### Usage
```r
data("zeros")
```

### Format
A data frame where the ith column contains the first 50 zeros of the Bessel function of the first kind and ((i - 1) / 2)th order, i = 1, ..., 5001.

### Source
The zeros are computed by the mathematical software octave.

### References
Index

*Topic datasets
  zeros, 11

  h_cumsum, 3, 7, 8, 10, 11
  huber_cusum, 2, 7, 10

  modifChol, 4

  pbessel3 (pKSdist), 5
  pKSdist, 3, 5, 10
  psi, 2–4, 7, 7, 10, 11

  sigma2, 2, 3, 9

  teststat, 3, 7, 8, 10, 10

  zeros, 11