Package: robcp (via r-universe)

October 18, 2026

Title Robust Change-Point Tests

Version 0.3.10

Description Provides robust methods to detect change-points in uni- or multivariate time series. They can cope with corrupted data and heavy tails. Focus is on the detection of abrupt changes in location, but changes in the scale or dependence structure can be detected as well. This package provides tests for change detection in uni- and multivariate time series based on Huberized versions of CUSUM tests proposed in Duerre and Fried (2019) <DOI:10.48550/arXiv.1905.06201>, and tests for change detection in univariate time series based on 2-sample U-statistics or 2-sample U-quantiles as proposed by Dehling et al. (2015) <DOI:10.1007/978-1-4939-3076-0_12> and Dehling, Fried and Wendler (2020) <DOI:10.1093/biomet/asaa004>. Furthermore, the packages provides tests on changes in the scale or the correlation as proposed in Gerstenberger, Vogel and Wendler (2020) <DOI:10.1080/01621459.2019.1629938>, Dehling et al. (2017) <DOI:10.1017/S026646661600044X>, and Wied et al. (2014) <DOI:10.1016/j.csda.2013.03.005>.

Depends R (>= 3.3.1)

License GPL-3

Encoding UTF-8

LazyData true

NeedsCompilation yes

Author Sheila Goerz [aut, cre], Alexander Duerre [aut], Roland Fried [ctb, ths]

Maintainer Sheila Goerz <sheila.goerz@tu-dortmund.de>

Imports methods, Rcpp, MASS, mvtnorm, pracma

LinkingTo Rcpp

RoxygenNote 7.3.2

Suggests testthat

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

Date/Publication 2026-01-09 13:04:32 UTC

RemoteUrl https://github.com/sgoerz/robcp

RemoteRef HEAD

RemoteSha 98f5678b2b3cfcd883b1068f04e133fb5ed1572e

Contents

robcp-package . . . 2
cor_cusum . . . 3
cor_stat . . . 5
CUSUM . . . 7
hl_test . . . 9
HodgesLehmann . . . 10
huber_cusum . . . 12
kthPair . . . 14
lrv . . . 15
medianDiff . . . 20
modifChol . . . 21
pKSdist . . . 22
plot.cpStat . . . 24
print.cpStat . . . 25
psi . . . 25
psi_cumsum . . . 27
Qalpha . . . 28
scale_cusum . . . 29
scale_stat . . . 31
weightedMedian . . . 33
wilcox_stat . . . 34
wmw_test . . . 35
zeros . . . 37

Index **38**

`robcp-package` *robcp: Robust Change-Point Tests*

Description

The package includes robust change point tests designed to identify structural changes in time series. As external influences might affect the characteristics of a time series, such as location, variability, and correlation structure, changes caused by them and the corresponding time point need to be detected reliably. Standard methods can struggle when dealing with heavy-tailed data or outliers; therefore, this package comprises robust methods that effectively manage extreme values by either ignoring them or assigning them less weight. Examples of these robust techniques include the Median, Huber M-estimator, mean deviation, Gini's mean difference, and Q^{α}.

The package contains the following tests and test statistics:

Tests on changes in the location

- Huberized CUSUM test: `huber_cusum` **(test),** `CUSUM` **(CUSUM test statistic),** `psi` **(transformation function).**

- Hodges-Lehmann test: `hl_test` **(test),** `HodgesLehmann` **(test statistic).**

- Wilcoxon-Mann-Whitney change point test: `wmw_test` **(test),** `wilcox_stat` **(test statistic).**

Tests on changes in the variability

Estimating the variability using the ordinary variance, mean deviation, Gini's mean difference and Q^α:

`scale_cusum` **(test),** `scale_stat` **(test statistic).**

Tests on changes in the correlation

Estimating the correlation by a sample version of Kendall's tau or Spearman's rho:

`cor_cusum` **(test),** `cor_stat` **(test statistic).**

Author(s)

Maintainer: Sheila Görz <sheila.goerz@tu-dortmund.de>

Author: Alexander Dürre <a.m.durre@math.leidenuniv.nl>

Thesis Advisor: Roland Fried <msnat@statistik.tu-dortmund.de>

`cor_cusum` *A CUSUM-type test to detect changes in the correlation.*

Description

Performs a CUSUM-based test on changes in Spearman's rho or Kendall's tau.

Usage

```
cor_cusum(x, version = c("tau", "rho"), method = "kernel", control = list(),
          fpc = TRUE, tol = 1e-08, plot = FALSE)
```

Arguments

`x`	time series (matrix or ts object with numeric/integer values).
`version`	version of the test. Either `rho` or `tau`.
`method`	method for estimating the long run variance.
`control`	a list of control parameters.
`fpc`	finite population correction (boolean).
`tol`	tolerance of the distribution function (numeric), which is used do compute p-values.
`plot`	should the test statistic be plotted (cf. `plot.cpStat`)? Boolean.

Details

The function perform a CUSUM-type test on changes in the correlation of a time series x. Formally, the hypothesis pair can be written as

$$H_0 : \xi_1 = ... = \xi_n$$

$$vs.$$

$$H_1 : \exists k \in \{1, ..., n-1\} : \xi_k \neq \xi_{k+1}$$

where ξ_i is a fluctuation measure (either Spearman's rho or Kendall's tau) for the i-th observations and n is the length of the time series. k is called a 'change point'.

The test statistic is computed using `cor_stat` and asymptotically follows a Kolmogorov distribution. To derive the p-value, the funtion `pKSdist` is used.

Value

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

`statistic`	return value of the function `cor_stat`.
`p.value`	p-value (numeric).
`alternative`	alternative hypothesis (character string).
`method`	name of the performed test (character string).
`cp.location`	index of the estimated change point location (integer).
`data.name`	name of the data (character string).
`lrv`	list containing the compontents `method`, `param` and `value`.

Author(s)

Sheila Görz

References

Wied, D., Dehling, H., Van Kampen, M., and Vogel, D. (2014). A fluctuation test for constant Spearman's rho with nuisance-free limit distribution. Computational Statistics & Data Analysis, 76, 723-736.

Dürre, A. (2022+). "Finite sample correction for cusum tests", *unpublished manuscript*

See Also

`cor_stat`, `lrv`, `pKSdist`

Examples

```
### first: generate a time series with a burn-in period of m and a change point k = n/2
require(mvtnorm)
n <- 500
m <- 100
N <- n + m
k <- m + floor(n * 0.5)
n1 <- N - k

## Spearman's rho:
rho <- c(0.4, -0.9)

# serial dependence:
theta1 <- 0.3
theta2 <- 0.2
theta <- cbind(c(theta1, 0), c(0, theta2))
q <- rho * sqrt( (theta1^2 + 1) * (theta2^2 + 1) / (theta1 * theta2 + 1))
# shape matrices of the innovations:
S0 <- cbind(c(1, q[1]), c(q[1], 1))
S1 <- cbind(c(1, q[2]), c(q[2], 1))

e0 <- rmvt(k, S0, 5)
e1 <- rmvt(n1, S1, 5)
e <- rbind(e0, e1)
# generate the data:
x <- matrix(numeric(N * 2), ncol = 2)
x[1, ] <- e[1, ]
invisible(sapply(2:N, function(i) x[i, ] <<- e[i, ] + theta %*% e[i-1, ]))
x <- x[-(1:m), ]

cor_cusum(x, "rho")


## Kendall's tau
S0 <- cbind(c(1, rho[1]), c(rho[1], 1))
S1 <- cbind(c(1, rho[2]), c(rho[2], 1))
e0 <- rmvt(k, S0, 5)
e1 <- rmvt(n1, S1, 5)
e <- rbind(e0, e1)
x <- matrix(numeric(N * 2), ncol = 2)
x[1, ] <- e[1, ]
# AR(1):
invisible(sapply(2:N, function(i) x[i, ] <<- 0.8 * x[i-1, ] + e[i, ]))
x <- x[-(1:m), ]

cor_cusum(x, version = "tau")
```

---

cor_stat *Test statistic to detect Correlation Changes*

Description

Computes the test statistic for a CUSUM-based tests on changes in Spearman's rho or Kendall's tau.

Usage

```
cor_stat(x, version = c("tau", "rho"), method = "kernel", control = list())
```

Arguments

`x`	time series (numeric or ts vector).
`version`	version of the test. Either `"rho"` or `"tau"`.
`method`	methods of long run variance estimation. Options are `"kernel"` and `"none"`.
`control`	a list of control parameters.

Details

Let n be the length of the time series, i.e. the number of rows in `x`. In general, the (scaled) CUSUM test statistic is defined as

$$\hat{T}_{\xi;n} = \max_{k=1,\dots,n} \frac{k}{2\sqrt{n}\hat{\sigma}} |\hat{\xi}_k - \hat{\xi}_n|,$$

where $\hat{\xi}$ is an estimator for the property on which to test, and $\hat{\sigma}$ is an estimator for the square root of the corresponding long run variance (cf. `lrv`).

If `version = "tau"`, the function tests if the correlation between x_i and x_i of the bivariate time series $(x_i, x_i)_{i=1,\dots,n}$ stays constant for all $i = 1, ..., n$ by considering Kendall's tau. Therefore, $\hat{\xi} = \hat{\tau}$ is the the sample version of Kendall's tau:

$$\hat{\tau}_k = \frac{2}{k(k-1)} \sum_{1 \le i < j \le k} sign\left((x_j - x_i)(y_j - y_i)\right).$$

The default bandwidth for the kernel-based long run variance estimation is $b_n = \lfloor 2n^{1/3} \rfloor$ and the default kernel function is the quatratic kernel.

If `version = "rho"`, the function tests if the correlation of a time series of an arbitrary dimension d (>= 2) stays constant by considering a multivariate version of Spearman's rho. Therefore, $\hat{\xi} = \hat{\rho}$ is the sample version of Spearman's rho:

$$\hat{\rho}_k = a(d) \left(\frac{2^d}{k} \sum_{j=1}^{k} \prod_{i=1}^{d} (1 - U_{i,j;n}) - 1 \right)$$

where $U_{i,j;n} = n^{-1}$ (rank of $x_{i,j}$ in $x_{i,1}, ..., x_{i,n}$) and $a(d) = (d+1)/(2^d - d - 1)$. Here it is essential to use $\hat{U}_{i,j;n}$ instead of $\hat{U}_{i,j;k}$. The default bandwidth for the kernel-based long run variance estimation is $\sqrt{n}$ and the default kernel function is the Bartlett kernel.

Value

Test statistic (numeric value) with the following attributes:

cp-location	indicating at which index a change point is most likely.
teststat	test process (before taking the maximum).
lrv-estimation	long run variance estimation method.
sigma	estimated long run variance.
param	parameter used for the lrv estimation.
kFun	kernel function used for the lrv estimation.

Is an S3 object of the class "cpStat".

Author(s)

Sheila Görz

References

Wied, D., Dehling, H., Van Kampen, M., and Vogel, D. (2014). A fluctuation test for constant Spearman's rho with nuisance-free limit distribution. Computational Statistics & Data Analysis, 76, 723-736.

See Also

lrv, cor_cusum

CUSUM *CUSUM Test Statistic*

Description

Computes the test statistic for the CUSUM change point test.

Usage

```
CUSUM(x, method = "kernel", control = list(), inverse = "Cholesky", ...)
```

Arguments

x	vector or matrix with each column representing a time series (numeric).
method	method of long run variance estimation. Options are "kernel", "subsampling", "bootstrap" and "none".
control	a list of control parameters for the estimation of the long run variance (cf. lrv).
inverse	character string specifying the method of inversion. Options are "Cholesky" for inverting over modifChol, "svd" for singular value decomposition and "generalized" for using ginv from the MASS package.
...	further arguments passed to the inverse-computing functions.

Details

Let n be the length of the time series $x = (x_1, ..., x_n)$.

In case of a univariate time series the test statistic can be written as

$$\max_{k=1,...,n} \frac{1}{\sqrt{n}\sigma} \left| \sum_{i=1}^{k} x_i - (k/n) \sum_{i=1}^{n} x_i \right|,$$

where σ is the square root of `lrv`. Default method is `"kernel"` and the default kernel function is `"TH"`. If no bandwidth value is supplied, first the time series x is corrected for the estimated change point and Spearman's autocorrelation to lag 1 (ρ) is computed. Then the default bandwidth follows as

$$b_n = \max \left\{ \left\lceil n^{0.45} \left(\frac{2\rho}{1-\rho^2} \right)^{0.4} \right\rceil, 1 \right\}.$$

In case of a multivariate time series the test statistic follows as

$$\max_{k=1,...,n} \frac{1}{n} \left(\sum_{i=1}^{k} X_i - \frac{k}{n} \sum_{i=1}^{n} X_i \right)^T \Sigma^{-1} \left(\sum_{i=1}^{k} X_i - \frac{k}{n} \sum_{i=1}^{n} X_i \right),$$

where X_i denotes the i-th row of x and Σ^{-1} is the inverse of `lrv`.

Value

Test statistic (numeric value) with the following attributes:

`cp-location`	indicating at which index a change point is most likely.
`teststat`	test process (before taking the maximum).
`lrv-estimation`	long run variance estimation method.
`sigma`	estimated long run variance.
`param`	parameter used for the lrv estimation.
`kFun`	kernel function used for the lrv estimation.

Is an S3 object of the class "cpStat".

Author(s)

Sheila Görz

See Also

`psi_cumsum`, `psi`

Examples

```
# time series with a location change at t = 20
ts <- c(rnorm(20, 0), rnorm(20, 2))

# Huberized CUSUM change point test statistic
CUSUM(psi(ts))
```

`hl_test` *Hodges-Lehmann Test for Change Points*

Description

Performs the two-sample Hodges-Lehmann change point test.

Usage

```
hl_test(x, b_u = "nrd0", method = "kernel", control = list(), tol = 1e-8,
        plot = FALSE)
```

Arguments

`x`	time series (numeric or ts vector).
`b_u`	bandwidth for `u_hat`. Either a numeric value or the name of a bandwidth selection function (c.f. `bw.nrd0`).
`method`	method for estimating the long run variance.
`control`	a list of control parameters (cf. `lrv`).
`tol`	tolerance of the distribution function (numeric), which is used to compute p-values.
`plot`	should the test statistic be plotted (cf. `plot.cpStat`)? Boolean.

Details

The function performs the two-sample Hodges-Lehmann change point test. It tests the hypothesis pair

$$H_0 : \mu_1 = ... = \mu_n$$

$$vs.$$

$$H_1 : \exists k \in \{1, ..., n-1\} : \mu_k \neq \mu_{k+1}$$

where $\mu_t = E(X_t)$ and n is the length of the time series. k is called a 'change point'.

The test statistic is computed using `HodgesLehmann` and asymptotically follows a Kolmogorov distribution. To derive the p-value, the function `pKSdist` is used.

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).
`cp.location`	index of the estimated change point location (integer).
`data.name`	name of the data (character string).

Author(s)

Sheila Görz

References

Dehling, H., Fried, R., and Wendler, M. "A robust method for shift detection in time series." Biometrika 107.3 (2020): 647-660.

See Also

HodgesLehmann, medianDiff, lrv, pKSdist

Examples

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

hl_test(Z, control = list(overlapping = TRUE, b_n = 5, b_u = 0.05))
```

HodgesLehmann *Hodges Lehmann Test Statistic*

Description

Computes the test statistic for the Hodges-Lehmann change point test.

Usage

```
HodgesLehmann(x, b_u = "nrd0", method = "kernel", control = list())
u_hat(x, b_u = "nrd0")
```

Arguments

x	time series (numeric or ts vector).
b_u	bandwidth for u_hat. Either a numeric value or the name of a bandwidth selection function (c.f. bw.nrd0).
method	method of long run variance estimation. Options are "kernel", "subsampling", "bootstrap" and "none".
control	a list of control parameters for the estimation of the long run variance (cf. lrv).

Details

Let n be the length of the time series. The Hodges-Lehmann test statistic is then computed as

$$\frac{\sqrt{n}}{\hat{\sigma}_n} \max_{1 \leq k < n} \hat{u}_{k,n}(0) \frac{k}{n} \left(1 - \frac{k}{n}\right) |med\{(x_j - x_i); 1 \leq i \leq k; k+1 \leq j \leq n\}|,$$

where $\hat{\sigma}$ is the estimated long run variance, computed by the square root of `lrv`. By default the long run variance is estimated kernel-based with the following bandwidth: first the time series x is corrected for the estimated change point and Spearman's autocorrelation to lag 1 (ρ) is computed. Then the default block length follows as

$$l = \max \left\{ \left\lceil n^{1/3} \left(\frac{2\rho}{1-\rho^2} \right)^{0.9} \right\rceil, 1 \right\}.$$

$\hat{u}_{k,n}(0)$ is estimated by `u_hat` on data $\tilde{x}$, where $med\{(x_j - x_i); 1 \leq i \leq k; k+1 \leq j \leq n\}$ was subtracted from $x_{k+1}, ..., x_n$. Then `density` with the arguments `na.rm = TRUE`, `from = 0`, `to = 0`, `n = 1` and `bw = b_u` is applied to $(\tilde{x}_i - \tilde{x}_j)_{1 \leq i < j \leq n}$.

Value

`HodgesLehmann` returns a test statistic (numeric value) with the following attributes:

`cp-location`	indicating at which index a change point is most likely.
`teststat`	test process (before taking the maximum).
`lrv-estimation`	long run variance estimation method.
`sigma`	estimated long run variance.
`param`	parameter used for the lrv estimation.
`kFun`	kernel function used for the lrv estimation.

Is an S3 object of the class "cpStat".

`u_hat` returns a numeric value.

Author(s)

Sheila Görz

References

Dehling, H., Fried, R., and Wendler, M. "A robust method for shift detection in time series." Biometrika 107.3 (2020): 647-660.

See Also

`medianDiff`, `lrv`

Examples

```
# time series with a location change at t = 20
x <- c(rnorm(20, 0), rnorm(20, 2))

# Hodges-Lehmann change point test statistic
HodgesLehmann(x, b_u = 0.01)
```

---

huber_cusum *Huberized CUSUM test*

Description

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

Usage

```
huber_cusum(x, fun = "HLm", k, constant = 1.4826, method = "kernel",
            control = list(), fpc = TRUE, tol = 1e-8, plot = FALSE, ...)
```

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 ψ, see details. For the ordinary CUSUM test, use fun = "none".
k	numeric bound used in psi.
constant	scale factor of the MAD. Default is 1.4826.
method	method for estimating the long run variance.
control	a list of control parameters for the estimation of the long run variance (cf. lrv).
fpc	finite population correction (boolean).
tol	tolerance of the distribution function (numeric), which is used to compute p-values.
plot	should the test statistic be plotted (cf. plot.cpStat). Boolean.
...	further arguments to be passed to CUSUM.

Details

The function performs a Huberized CUSUM test. It tests the null hypothesis H_0 : $\boldsymbol{\theta}$ does not change for x against the alternative of a change, where $\boldsymbol{\theta}$ is the parameter vector of interest. k is called a 'change point'. First the data is transformed by a suitable psi-function. To detect changes in location one can apply fun = "HLm", "HLg", "SLm" or "SLg" and the hypothesis pair is

$$H_0 : \mu_1 = ... = \mu_n$$

$$vs.$$

$$H_1 : \exists k \in \{1, ..., n-1\} : \mu_k \neq \mu_{k+1}$$

where $\mu_t = E(X_t)$ and n is the length of the time series. For changes in scale `fun = "HCm"` is available and for changes in the dependence respectively covariance structure `fun = "HCm"`, `"HCg"`, `"SCm"` and `"SCg"` are possible. The hypothesis pair is the same as in the location case, only with μ_i being replaced by Σ_i, $\Sigma_i = Cov(X_i)$. Exact definitions of the psi-functions can be found on the help page of `psi`.

Denote by $Y_1, \ldots, Y_n$ the transformed time series. If Y_1 is one-dimensional, then the test statistic

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

is calculated, where σ^2 is an estimator for the long run variance, see the help function of `lrv` for details. V is asymptotically Kolmogorov-Smirnov distributed. If `fpc` is `TRUE` we use a finite population correction $V + 0.58/\sqrt{n}$ to improve finite sample performance (Dürre, 2021+).
If Y_1 is multivariate, then the test statistic

$$W_n = \max_{k=1,\ldots,n} \frac{1}{n} \left(\sum_{i=1}^{k} Y_i - \frac{k}{n} \sum_{i=1}^{n} Y_i \right)' \Sigma^{-1} \left(\sum_{i=1}^{k} Y_i - \frac{k}{n} \sum_{i=1}^{n} Y_i \right)$$

is computed, where Σ is the long run covariance, see also `lrv` for details. W is asymptotically distributed like the maximum of a squared Bessel bridge. We use the identity derived by Kiefer (1959) to derive p-values. Like in the one dimensional case if `fpc` is `TRUE` we use a finite sample correction $(\sqrt{W} + 0.58/\sqrt{n})^2$.

The change point location is estimated as the time point k for which the CUSUM process takes its maximum.

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).
`cp.location`	index of the estimated change point location (integer).
`data.name`	name of the data (character string).

Author(s)

Sheila Görz

References

Hušková, M., & Marušiaková, M. (2012). M-procedures for detection of changes for dependent observations. Communications in Statistics-Simulation and Computation, 41(7), 1032-1050.

Dürre, A. and Fried, R. (2019). "Robust change point tests by bounded transformations", https://arxiv.org/abs/1905.06201

Dürre, A. (2021+). "Finite sample correction for cusum tests", *unpublished manuscript*

Kiefer, J. (1959). "K-sample analogues of the Kolmogorov-Smirnov and Cramer-V. Mises tests", *The Annals of Mathematical Statistics*, 420–447.

See Also

lrv, psi, psi_cumsum, CUSUM, pKSdist

Examples

```
set.seed(1895)

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

---

kthPair *K-th largest element in a sum of sets.*

Description

Selects the k-th largest element of X + Y, a sum of sets. X + Y denotes the set $\{x+y|x \in X, y \in Y\}$.

Usage

```
kthPair(X, Y, k, k2 = NA)
```

Arguments

X	Numeric vector.
Y	Numeric vector.
k	Index of element to be selected. Must be an integer and between 1 and the product of the lengths of x and y.
k2	Optional second index. k and k2 must be consecutive. Useful, if the number of elements of X + Y is even and the median is to be calculated.

Details

A generalized version of the algorithm of Johnson and Mizoguchi (1978), where X and Y are allowed to be of different lengths. The optional argument `k2` allows the computation of the mean of two consecutive value without running the algorithm twice.

Value

K-th largest value (numeric).

Author(s)

Sheila Görz

References

Johnson, D. B., & Mizoguchi, T. (1978). Selecting the K-th Element in X+Y and X_1+X_2+ ... +X_m. SIAM Journal on Computing, 7(2), 147-153.

Examples

```
set.seed(1895)
x <- rnorm(100)
y <- runif(100)

kthPair(x, y, 5000)
kthPair(x, y, 5000, 5001)
```

---

lrv *Long Run Variance*

Description

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

Usage

```
lrv(x, method = c("kernel", "subsampling", "bootstrap", "none"), control = list())
```

Arguments

`x`	vector or matrix with each column representing a time series (numeric).
`method`	method of estimation. Options are `kernel`, `subsampling`, `bootstrap` and `none`.
`control`	a list of control parameters. See 'Details'.

Details

The long run variance equals the limit of n times the 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 tests concering the mean on dependent data.

If `method = "none"`, no long run variance estimation is performed and the value 1 is returned (i.e. it does not alterate the test statistic).

The `control` argument is a list that can supply any of the following components:

`kFun` Kernel function (character string). More in 'Notes'.

`b_n` Bandwidth (numeric > 0 and smaller than sample size).

`gamma0` Only use estimated variance if estimated long run variance is < 0? Boolean.

`l` Block length (numeric > 0 and smaller than sample size).

`overlapping` Overlapping subsampling estimation? Boolean.

`distr` Tranform observations by their empirical distribution function? Boolean. Default is `FALSE`.

`B` Bootstrap repetitions (integer).

`seed` RNG seed (numeric).

`version` What property does the CUSUM test test for? Character string, details below.

`loc` Estimated location corresponding to `version`. Numeric value, details below.

`scale` Estimated scale corresponding to `version`. Numeric value, details below.

Kernel-based estimation

The kernel-based long run variance estimation is available for various testing scenarios (set by `control$version`) and both for one- and multi-dimensional data. It uses the bandwidth $b_n =$ `control$b_n` and kernel function $k(x) =$ `control$kFun`. For tests on certain properties also a corresponding location `control$loc` (m_n) and scale `control$scale` (v_n) estimation needs to be supplied. Supported testing scenarios are:

- `"mean"`
  - 1-dim. data:

$$\hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2 + \frac{2}{n}\sum_{h=1}^{b_n}\sum_{i=1}^{n-h}(x_i - \bar{x})(x_{i+h} - \bar{x})k(h/b_n).$$

    If `control$distr = TRUE`, then the long run variance is estimated on the empirical distribution of x. The resulting value is then multiplied with $\sqrt{\pi}/2$.
    Default values: `b_n` = $0.9n^{1/3}$, `kFun = "bartlett"`.
  - multivariate time series: The k,l-element of Σ is estimated by

$$\hat{\Sigma}^{(k,l)} = \frac{1}{n}\sum_{i,j=1}^{n}(x_i^{(k)} - \bar{x}^{(k)})(x_j^{(l)} - \bar{x}^{(l)})k((i-j)/b_n),$$

    $k, l = 1, ..., m$.
    Default values: `b_n` = $\log_{1.8+m/40}(n/50)$, `kFun = "bartlett"`.

- `"empVar"` for tests on changes in the empirical variance.

$$\hat{\sigma}^2 = \sum_{h=-(n-1)}^{n-1} W\left(\frac{|h|}{b_n}\right) \frac{1}{n} \sum_{i=1}^{n-|h|} ((x_i - m_n)^2 - v_n)((x_{i+|h|} - m_n)^2 - v_n).$$

  Default values: $m_n =$ `mean(x)`, $v_n =$ `var(x)`.

- `"MD"` for tests on a change in the median deviation.

$$\hat{\sigma}^2 = \sum_{h=-(n-1)}^{n-1} W\left(\frac{|h|}{b_n}\right) \frac{1}{n} \sum_{i=1}^{n-|h|} (|x_i - m_n| - v_n)(|x_{i+|h|} - m_n| - v_n).$$

  Default values: $m_n =$ `median(x)`, $v_n = \frac{1}{n-1} \sum_{i=1}^{n} |x_i - m_n|$.

- `"GMD"` for tests on changes in Gini's mean difference.

$$\hat{\sigma}^2 = 4 \sum_{h=-(n-1)}^{n-1} W\left(\frac{|h|}{b_n}\right) \frac{1}{n} \sum_{i=1}^{n-|h|} \hat{\phi}_n(x_i)\hat{\phi}_n(x_{i+|h|})$$

  with $\hat{\phi}_n(x) = n^{-1} \sum_{i=1}^{n} |x - x_i| - v_n$.
  Default value: $v_n = \frac{2}{n(n-1)} \sum_{1 \le i < j \le n} |x_i - x_j|$.

- `"Qalpha"` for tests on changes in `Qalpha`.

$$\hat{\sigma}^2 = \frac{4}{\hat{u}(v_n)} \sum_{h=-(n-1)}^{n-1} W\left(\frac{|h|}{b_n}\right) \frac{1}{n} \sum_{i=1}^{n-|h|} \hat{\phi}_n(x_i)\hat{\phi}_n(x_{i+|h|}),$$

  where $\hat{\phi}_n(x) = n^{-1} \sum_{i=1}^{n} 1_{\{|x-x_i| \le v_n\}} - m_n$ and

$$\hat{u}(t) = \frac{2}{n(n-1)h_n} \sum_{1 \le i < j \le n} K\left(\frac{|x_i - x_j| - t}{h_n}\right)$$

  the kernel density estimation of the densitiy u corresponding to the distribution function $U(t) = P(|X - Y| \le t)$, $h_n =$ `IQR(x)`$n^{-\frac{1}{3}}$ and K is the quatratic kernel function.
  Default values: $m_n = \alpha = 0.5$, $v_n =$ `Qalpha(x, m_n)[n-1]`.

- `"tau"` for tests in changes in Kendall's tau.
  Only available for bivariate data: assume that the given data `x` has the format $(x_i, y_i)_{i=1,\ldots,n}$.

$$\hat{\sigma}^2 = \sum_{h=-(n-1)}^{n-1} W\left(\frac{|h|}{b_n}\right) \frac{1}{n} \sum_{i=1}^{n-|h|} \hat{\phi}_n((x_i, y_i))\hat{\phi}_n((x_{i+|h|}, y_{i+|h|}),$$

  where $\hat{\phi}_n(x) = 4F_n(x, y) - 2F_{X,n}(x)2 - F_{Y,n}(y) + 1 - v_n$ and F_n, $F_{X,n}$ and $F_{Y,n}$ are the empirical distribution functions of $((X_i, Y_i))_{i=1,\ldots,n}$, $(X_i)_{i=1,\ldots,n}$ and $(Y_i)_{i=1,\ldots,n}$.
  Default value: $v_n = \hat{\tau}_n = \frac{2}{n(n-1)} \sum_{1 \le i < j \le n} sign\left((x_j - x_i)(y_j - y_i)\right)$.

- "rho" for tests on changes in Spearman's rho.
  Only availabe for d-variate data with $d > 1$: assume that the given data x has the format $(x_{i,j}|i = 1, ..., n; j = 1, ..., d)$.

$$\hat{\sigma}^2 = a(d)^2 2^{2d} \left\{ \sum_{h=-(n-1)}^{n-1} K\left(\frac{|h|}{b_n}\right) \left(\sum_{i=1}^{n-|h|} n^{-1} \prod_{j=1}^{d} \hat{\phi}_n(x_i, x_j)\hat{\phi}_n(x_{i+|h|}, x_j) - M^2 \right) \right\},$$

  where $a(d) = (d+1)/(2^d - d - 1)$, $M = n^{-1} \sum_{i=1}^{n} \prod_{j=1}^{d} \hat{\phi}_n(x_i, x_j)$ and $\hat{\phi}_n(x, y) = 1 - \hat{U}_n(x, y)$, $\hat{U}_n(x, y) = n^{-1}$ (rank of $x_{i,j}$ in $x_{i,1}, ..., x_{i,n}$).

When control$gamma0 = TRUE (default) then negative estimates of the long run variance are replaced by the autocovariance at lag 0 (= ordinary variance of the data). The function will then throw a warning.

Subsampling estimation

For method = "subsampling" there are an overlapping and a non-overlapping version (parameter control$overlapping). Also it can be specified if the observations x were transformed by their empirical distribution function $\tilde{F}_n$ (parameter control$distr). Via control$l the block length l can be controlled.

If control$overlapping = TRUE and control$distr = TRUE:

$$\hat{\sigma}_n = \frac{\sqrt{\pi}}{\sqrt{2l}(n-l+1)} \sum_{i=0}^{n-l} \left| \sum_{j=i+1}^{i+l} (F_n(x_j) - 0.5) \right|.$$

Otherwise, if control$distr = FALSE, the estimator is

$$\hat{\sigma}^2 = \frac{1}{l(n-l+1)} \sum_{i=0}^{n-l} \left(\sum_{j=i+1}^{i+l} x_j - \frac{l}{n} \sum_{j=1}^{n} x_j \right)^2.$$

If control$overlapping = FALSE and control$distr = TRUE:

$$\hat{\sigma} = \frac{1}{n/l} \sqrt{\pi/2} \sum_{i=1} n/l \frac{1}{\sqrt{l}} \left| \sum_{j=(i-1)l+1}^{il} F_n(x_j) - \frac{l}{n} \sum_{j=1}^{n} F_n(x_j) \right|.$$

Otherwise, if control$distr = FALSE, the estimator is

$$\hat{\sigma}^2 = \frac{1}{n/l} \sum_{i=1}^{n/l} \frac{1}{l} \left(\sum_{j=(i-1)l+1}^{il} x_j - \frac{l}{n} \sum_{j=1}^{n} x_j \right)^2.$$

Default values: overlapping = TRUE, the block length is chosen adaptively:

$$l_n = \max \left\{ \left\lceil n^{1/3} \left(\frac{2\rho}{1-\rho^2} \right)^{(2/3)} \right\rceil, 1 \right\}$$

where ρ is the Spearman autocorrelation at lag 1.

Bootstrap estimation

If `method = "bootstrap"` a dependent wild bootstrap with the parameters $B =$ `control$B`, $l =$ `control$l` and $k(x) =$ `control$kFun` is performed:

$$\hat{\sigma}^2 = \sqrt{n} Var(\bar{x}_k^* - \bar{x}), k = 1, ..., B$$

A single x_{ik}^* is generated by $x_i^* = \bar{x} + (x_i - \bar{x})a_i$ where a_i are independent from the data `x` and are generated from a multivariate normal distribution with $E(A_i) = 0$, $Var(A_i) = 1$ and $Cov(A_i, A_j) = k\left(\frac{i-j}{l}\right), i = 1, ..., n; j \neq i$. Via `control$seed` a seed can optionally be specified (cf. `set.seed`). Only `"bartlett"`, `"parzen"` and `"QS"` are supported as kernel functions. Uses the function `sqrtm` from package `pracma`.

Default values: `B` = 1000, `kFun` = `"bartlett"`, `l` is the same as for subsampling.

Value

long run variance σ^2 (numeric) resp. Σ (numeric matrix)

Note

Kernel functions

`bartlett`:

$$k(x) = (1 - |x|) * 1_{\{|x|<1\}}$$

`FT`:

$$k(x) = 1 * 1_{\{|x|\leq 0.5\}} + (2 - 2 * |x|) * 1_{\{0.5<|x|<1\}}$$

`parzen`:

$$k(x) = (1 - 6x^2 + 6|x|^3) * 1_{\{0\leq|x|\leq 0.5\}} + 2(1 - |x|)^3 * 1_{\{0.5<|x|\leq 1\}}$$

`QS`:

$$k(x) = \frac{25}{12\pi^2 x^2}\left(\frac{\sin(6\pi x/5)}{6\pi x/5} - \cos(6\pi x/5)\right)$$

`TH`:

$$k(x) = (1 + \cos(\pi x))/2 * 1_{\{|x|<1\}}$$

`truncated`:

$$k(x) = 1_{\{|x|<1\}}$$

`SFT`:

$$k(x) = (1 - 4(|x| - 0.5)^2)^2 * 1_{\{|x|<1\}}$$

`Epanechnikov`:

$$k(x) = 3\frac{1 - x^2}{4} * 1_{\{|x|<1\}}$$

`quatratic`:

$$k(x) = (1 - x^2)^2 * 1_{\{|x|<1\}}$$

Author(s)

Sheila Görz

References

Andrews, D.W. "Heteroskedasticity and autocorrelation consistent covariance matrix estimation." Econometrica: Journal of the Econometric Society (1991): 817-858.

Dehling, H., et al. "Change-point detection under dependence based on two-sample U-statistics." Asymptotic laws and methods in stochastics. Springer, New York, NY, (2015). 195-220.

Dehling, H., Fried, R., and Wendler, M. "A robust method for shift detection in time series." Biometrika 107.3 (2020): 647-660.

Parzen, E. "On consistent estimates of the spectrum of a stationary time series." The Annals of Mathematical Statistics (1957): 329-348.

Shao, X. "The dependent wild bootstrap." Journal of the American Statistical Association 105.489 (2010): 218-235.

See Also

CUSUM, HodgesLehmann, wilcox_stat

Examples

```
Z <- c(rnorm(20), rnorm(20, 2))

## kernel density estimation
lrv(Z)

## overlapping subsampling
lrv(Z, method = "subsampling", control = list(overlapping = FALSE, distr = TRUE, l_n = 5))

## dependent wild bootstrap estimation
lrv(Z, method = "bootstrap", control = list(l_n = 5, kFun = "parzen"))
```

---

medianDiff *Median of the set X - Y*

Description

Computes the median of the set X - Y. X - Y denotes the set $\{x - y | x \in X, y \in Y\}$.

Usage

```
medianDiff(x, y)
```

Arguments

x, y	Numeric vectors

Details

Special case of the function `kthPair`.

Value

The median element of X - Y

Author(s)

Sheila Görz

References

Johnson, D. B., & Mizoguchi, T. (1978). Selecting the K-th Element in X+Y and X_1+X_2+ ... +X_m. SIAM Journal on Computing, 7(2), 147-153.

Examples

```
x <- rnorm(100)
y <- runif(100)
medianDiff(x, y)
```

---

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)
```

Arguments

`x`	a symmetric matrix.
`tau`	(machine epsilon)^(1/3).
`tau_bar`	(machine epsilon^(2/3)).
`mu`	numeric, $0 < \mu \leq 1$.

Details

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

Value

Upper triangular matrix L of the form $L'L = x + E$.

Author(s)

Sheila Görz

References

Schnabel, R. B., & Eskow, E. (1999). "A revised modified Cholesky factorization algorithm" SIAM Journal on optimization, 9(4), 1135-1148.

Examples

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

`pKSdist` *Asymptotic cumulative distribution for the CUSUM Test statistic*

Description

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

Usage

```
pKSdist(tn, tol = 1e-8)
pBessel(tn, p)
```

Arguments

`tn`	vector of test statistics (numeric). For `pBessel` length of `tn` has to be 1.
`p`	dimension of time series (integer). If `p` is equal to 1 `pBessel` uses `pKSdist` 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 computated as follows (Durbin, 1973 and van Mulbregt, 2018):

For $t_n(x) < 1$:

$$P(t_n(X) \leq t_n(x)) = \frac{\sqrt{2\pi}}{t_n(x)} t(1 + t^8(1 + t^{16}(1 + t^{24}(1 + ...))))$$

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

else:

$$P(t_n(X) \leq t_n(x)) = 2\sum_{k=1}^{\infty}(-1)^{k-1}\exp(-2k^2x^2)$$

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

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

$$P(t_n(X) \leq t_n(x)) = \frac{4}{\Gamma(p/2)2^{p/2}t_n^p}\sum_{i=1}^{\infty}\frac{(\gamma_{(p-2)/2,n})^{p-2}\exp(-(\gamma_{(p-2)/2,n})^2/(2t_n^2))}{J_{p/2}(\gamma_{(p-2)/2,n})^2},$$

where J_p is the Bessel function of first kind and p-th order, Γ is the gamma function and $\gamma_{p,n}$ denotes the n-th zero of J_p.

Value

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

Author(s)

Sheila Görz, Alexander Dürre

References

Durbin, James. (1973) "Distribution theory for tests based on the sample distribution function." *Society for Industrial and Applied Mathematics*.

van Mulbregt, P. (2018) "Computing the Cumulative Distribution Function and Quantiles of the limit of the Two-sided Kolmogorov-Smirnov Statistic." arXiv preprint arXiv:1803.00426.

/src/library/stats/src/ks.c rev60573

Kiefer, J. (1959). "K-sample analogues of the Kolmogorov-Smirnov and Cramer-V. Mises tests", *The Annals of Mathematical Statistics*, 420–447.

See Also

psi, CUSUM, psi_cumsum, huber_cusum

Examples

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

pKSdist(tn)

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

```
tn <- CUSUM(timeSeries)

pBessel(tn, 2)
```

`plot.cpStat` *Plot method for change point statistics*

Description

Plots the trajectory of the test statistic process together with a red line indicating the critical value (alpha = 0.05) and a blue line indicating the most probable change point location

Usage

```
## S3 method for class 'cpStat'
plot(x, ylim, xaxt, crit.val, ...)
```

Arguments

`x`	object of the class 'cpStat'.
`ylim`	the y limits of the plot.
`xaxt`	a character which specifies the x axis type (see `par`).
`crit.val`	critical value of the test. Default: 1.358.
`...`	other graphical parameters (see `par`).

Details

- Default for `ylim` is `c(min(c(data, 1.358)), max(c(data, 1.358)))`.
- Default for `xaxt` is the simliar to the option `"s"`, only that there is a red labelled tick at the most probable change point location. Ticks too close to this will be suppressed.

Value

No return value; called for side effects.

Author(s)

Sheila Görz

See Also

`plot`, `par`, `CUSUM`, `HodgesLehmann`, `wilcox_stat`

print.cpStat *Print method for change point statistics*

Description

Prints the value of the test statistic and adds the most likely change point location.

Usage

```
## S3 method for class 'cpStat'
print(x, ...)
```

Arguments

x	object of the class 'cpStat'.
...	further arguments passed to or from other methods.

Value

Object x is return; function is called for its side effect.

Author(s)

Sheila Görz

See Also

print, wilcox_stat, CUSUM, HodgesLehmann,

psi *Transformation of time series*

Description

Standardizes (multivariate) time series by their median, MAD and transforms the standardized time series by a ψ function.

Usage

```
psi(y, fun = c("HLm", "HLg", "SLm", "SLg", "HCm", "HCg", "SCm", "SCg"), 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 (more in Details). If `fun = "none"`, no transformation is performed.
k	numeric bound used for Huber type psi-functions which determines robustness and efficiency of the test. Default for `psi = "HLg"` 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 $x = \frac{y - med(y)}{MAD(y)}$ be the standardized values of a univariate time series.

Available ψ functions are:

marginal Huber for location:
`fun = "HLm"`.
$\psi_{HLm}(x) = k * 1_{\{x>k\}} + x * 1_{\{-k \leq x \leq k\}} - k * 1_{\{x<-k\}}$.

global Huber for location:
`fun = "HLg"`.
$\psi_{HLg}(x) = x * 1_{\{0<|x| \leq k\}} + \frac{kx}{|x|} * 1_{\{|x|>k\}}$.

marginal sign for location:
`fun = "SLm"`.
$\psi_{SLm}(x_i) = sign(x_i)$.

global sign for location:
`fun = "SLg"`.
$\psi_{SLg}(x) = x/|x| * 1_{\{|x|>0\}}$.

marginal Huber for covariance:
`fun = "HCm"`.
$\psi_{HCm}(x) = \psi_{HLm}(x)\psi_{HLm}(x)^T$.

global Huber for covariance:
`fun = "HCg"`.
$\psi_{HCg}(x) = \psi_{HLg}(x)\psi_{HLg}(x)^T$.

marginal sign covariance:
`fun = "SCm"`.
$\psi_{SCm}(x) = \psi_{SLm}(x)\psi_{SLm}(x)^T$.

gloabl sign covariance:
`fun = "SCg"`.

$\psi_{SCg}(x) = \psi_{SCg}(x)\psi_{SCg}(x)^T$.

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. For 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

`psi_cumsum`, `CUSUM`

Examples

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

`psi_cumsum` *Cumulative sum of transformed vectors*

Description

Computes the cumulative sum of a transformed numeric vector or matrix. Transformation function is `psi`.

Usage

```
psi_cumsum(y, fun = "HLm", k, constant)
```

Arguments

`y`	numeric vector containing a single time series or a numeric matrix containing multiple time series (column-wise).
`fun`	character string specifying the transformation function ψ.
`k`	numeric bound used in `psi`.
`constant`	scale factor of the MAD. Default is 1.4826.

Details

Prior to computing the sums, y is being transformed by the function `fun`.

Value

Numeric vector or matrix containing the cumulative sums of the transformed values. In case of a matrix, cumulative sums are being computed for each time series (column) independently.

Author(s)

Sheila Görz

See Also

psi.

Examples

```
psi_cumsum(rnorm(100))
```

Qalpha $Q\hat{}\alpha$

Description

Estimates Q-alpha using the first k elements of a time series.

Usage

```
Qalpha(x, alpha = 0.8)
```

Arguments

x	numeric vector.
alpha	quantile. Numeric value in (0, 1]

Details

$$\hat{Q}^{\alpha}_{1:k} = U^{-1}_{1:k}(\alpha) = \inf\{x|\alpha \leq U_{1:k}(x)\},$$

where $U_{1:k}$ is the empirical distribtion function of $|x_i - x_j|$, $1 \leq i < j \leq k$.

Value

Numeric vector of Q^{α}-s estimated using x[1], ..., x[k], $k = 1, ..., n$, n being the length of x.

Examples

```
x <- rnorm(10)
Qalpha(x, 0.5)
```

scale_cusum *Tests for Scale Changes Based on Pairwise Differences*

Description

Performs the CUSUM-based test on changes in the scale.

Usage

```
scale_cusum(x, version =  c("empVar", "MD", "GMD", "Qalpha"), method = "kernel",
            control = list(), alpha = 0.8, fpc = TRUE, tol,
            plot = FALSE, level = 0.05)
```

Arguments

x	time series (numeric or ts vector).
version	variance estimation method (see scale_stat).
method	method for estimating the long run variance.
control	a list of control parameters.
alpha	quantile of the distribution function of all absolute pairwise differences used in version = "Qalpha".
fpc	finite population correction (boolean).
tol	tolerance for the computation of the p-value (numeric). Default for kernel-based long run variance estimation: 1e-8. Default for bootstrap: 1e-3.
plot	should the test statistic be plotted (cf. plot.cpStat)? Boolean.
level	significance level of the test (numeric between 0 and 1).

Details

The function performs a CUSUM-type test on changes in the scale. Formally, the hypothesis pair is

$$H_0 : \sigma_2^2 = ... = \sigma_n^2$$

$$vs.$$

$$H_1 : \exists k \in \{2, ..., n-1\} : \sigma_k^2 \neq \sigma_{k+1}^2$$

where $\sigma_t^2 = Var(X_t)$ and n is the length of the time series. k is called a 'change point'. The hypotheses do not include σ_1^2 since then the variance of one single observation would need to be estimated.

The test statistic is computed using scale_stat and in the case of method = "kernel" asymptotically follows a Kolmogorov distribution. To derive the p-value, the function pKSdist is used.

If method = "bootstrap", a dependent block bootstrap with parameters $B = 1/$tol and $l =$ control$l is performed in order to derive the p-value of the test. First, select a boostrap sample $x_1^*, ..., x_{kl}^*$, $k = \lfloor n/l \rfloor$, the following way: Uniformly draw a random iid sample $J_1, ..., J_k$ from $\{1, ..., n-l+1\}$ and concatenate the blocks $x_{J_i}, ..., x_{J_i+l-1}$ for $i = 1, ..., k$. Then apply the test statistic $\hat{T}_s$ to the bootstrap sample. Repeat the procedure B times. The p-value is can be obtained as the proportion of the bootstrapped test statistics which is larger than the test statistic on the full sample.

Value

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

`statistic`	return value of the function `scale_stat`.
`p.value`	p-value (numeric).
`alternative`	alternative hypothesis (character string).
`method`	name of the performed test (character string).
`cp.location`	index of the estimated change point location (integer).
`data.name`	name of the data (character string).

Plus if `method = "kernel"`:

`lrv`	list containing the compontents `method`, `param` and `value`.

else if `method = "bootstrap"`:

`bootstrap`	list containing the compontents `param` (= block length) and `crit.value`.

Author(s)

Sheila Görz

References

Gerstenberger, C., Vogel, D., and Wendler, M. (2020). Tests for scale changes based on pairwise differences. Journal of the American Statistical Association, 115(531), 1336-1348.

Dürre, A. (2022+). "Finite sample correction for cusum tests", *unpublished manuscript*

See Also

`scale_stat`, `lrv`, `pKSdist`, `Qalpha`

Examples

```
x <- arima.sim(list(ar = 0.5), 100)

# under H_0:
scale_cusum(x)
scale_cusum(x, "MD")

# under the alternative:
x[51:100] <- x[51:100] * 3
scale_cusum(x)
scale_cusum(x, "MD")
```

scale_stat *Test statistic to detect Scale Changes*

Description

Computes the test statistic for CUSUM-based tests on scale changes.

Usage

```
scale_stat(x, version =  c("empVar", "MD", "GMD", "Qalpha"), method = "kernel",
           control = list(), alpha = 0.8)
```

Arguments

x	time series (numeric or ts vector).
version	variance estimation method.
method	either "kernel" for performing a kernel-based long run variance estimation, or "bootstrap" for performing a dependent wild bootstrap. See 'Details' below.
control	a list of control parameters.
alpha	quantile of the distribution function of all absolute pairwise differences used in version = "Qalpha".

Details

Let n be the length of the time series. The CUSUM test statistic for testing on a change in the scale is then defined as

$$\hat{T}_s = \max_{1<k\leq n} \frac{k}{\sqrt{n}} |\hat{s}_{1:k} - \hat{s}_{1:n}|,$$

where $\hat{s}_{1:k}$ is a scale estimator computed using only the first k elements of the time series x.

If method = "kernel", the test statistic $\hat{T}_s$ is divided by the estimated long run variance $\hat{D}_s$ so that it asymptotically follows a Kolmogorov distribution. $\hat{D}_s$ is computed by the function lrv using kernel-based estimation.

For the scale estimator $\hat{s}_{1:k}$, there are five different options which can be set via the version parameter:

Empirical variance (empVar)

$$\hat{\sigma}^2_{1:k} = \frac{1}{k-1} \sum_{i=1}^{k} (x_i - \bar{x}_{1:k})^2; \ \bar{x}_{1:k} = k^{-1} \sum_{i=1}^{k} x_i.$$

Mean deviation (MD)

$$\hat{d}_{1:k} = \frac{1}{k-1} \sum_{i=1}^{k} |x_i - med_{1:k}|,$$

where $med_{1:k}$ is the median of $x_1, ..., x_k$.

Gini's mean difference (GMD)

$$\hat{g}_{1:k} = \frac{2}{k(k-1)} \sum_{1 \le i < j \le k} (x_i - x_j).$$

Q^α (Qalpha)

$$\hat{Q}^\alpha_{1:k} = U^{-1}_{1:k}(\alpha) = \inf\{x | \alpha \le U_{1:k}(x)\},$$

where $U_{1:k}$ is the empirical distribtion function of $|x_i - x_j|$, $1 \le i < j \le k$ (cf. Qalpha).

For the kernel-based long run variance estimation, the default bandwidth b_n is determined as follows:

If $\hat{\rho}_j$ is the estimated autocorrelation to lag j, a maximal lag l is selected to be the smallest integer k so that

$$\max\{|\hat{\rho}_k|, ..., |\hat{\rho}_{k+\kappa_n}|\} \le 2\sqrt{(}\log_{10}(n)/n),$$

$\kappa_n = \max\{5, \sqrt{\log_{10}(n)}\}$. This l is determined for both the original data x and the squared data x^2 and the maximum l_{max} is taken. Then the bandwidth b_n is the minimum of l_{max} and $n^{1/3}$.

Value

Test statistic (numeric value) with the following attributes:

`cp-location`	indicating at which index a change point is most likely.
`teststat`	test process (before taking the maximum).
`lrv-estimation`	long run variance estimation method.

If `method = "kernel"` the following attributes are also included:

`sigma`	estimated long run variance.
`param`	parameter used for the lrv estimation.
`kFun`	kernel function used for the lrv estimation.

Is an S3 object of the class "cpStat".

Author(s)

Sheila Görz

References

Gerstenberger, C., Vogel, D., and Wendler, M. (2020). Tests for scale changes based on pairwise differences. Journal of the American Statistical Association, 115(531), 1336-1348.

See Also

lrv, Qalpha

Examples

```
x <- arima.sim(list(ar = 0.5), 100)

# under H_0:
scale_stat(x, "GMD")
scale_stat(x, "Qalpha", method = "bootstrap")

# under the alternative:
x[51:100] <- x[51:100] * 3
scale_stat(x)
scale_stat(x, "Qalpha", method = "bootstrap")
```

---

weightedMedian *Weighted Median*

Description

Computes the weighted median of a numeric vector.

Usage

```
weightedMedian(x, w)
```

Arguments

x	Numeric vector.
w	Integer vector of weights.

Details

Here, the median of an even length n of x is defined as $x_{(n/2+1)}$ if $x_{(i)}$ is the i-th largest element in x, i.e. the larger value is taken.

Value

Weighted median of x with respect to w.

Examples

```
x <- c(1, 4, 9)
w <- c(5, 1, 1)
weightedMedian(x, w)
```

`wilcox_stat` *Wilcoxon-Mann-Whitney Test Statistic for Change Points*

Description

Computes the test statistic for the Wilcoxon-Mann-Whitney change point test

Usage

```
wilcox_stat(x, h = 1L, method = "kernel", control = list())
```

Arguments

`x`	Time series (numeric or ts vector).
`h`	Kernel function of the U statistic (1L or 2L, or a function with two parameters).
`method`	Method for estimating the long run variance. Options are `"kernel"`, `"subsampling"`, `"bootstrap"` and `"none"`.
`control`	A list of control parameters for the estimation of the long run variance (cf. `lrv`).

Details

Let n be the length of `x`, i.e. the number of observations.

`h = 1L`:

$$T_n = \frac{1}{\hat{\sigma}} \max_{1 \le k \le n} \left| \frac{1}{n^{3/2}} \sum_{i=1}^{k} \sum_{j=k+1}^{n} (1_{\{x_i < x_j\}} - 0.5) \right|$$

`h = 2L`:

$$T_n = \frac{1}{\hat{\sigma}} \max_{1 \le k \le n} \left| \frac{1}{n^{3/2}} \sum_{i=1}^{k} \sum_{j=k+1}^{n} (x_i - x_j) \right|$$

$\hat{\sigma}$ is estimated by the square root of `lrv`. The denominator corresponds to that in the ordinary CUSUM change point test.

By default, kernel-based estimation is used.

If `h = 1L`, the default for `distr` is `TRUE`. If no block length is supplied, first the time series x is corrected for the estimated change point and Spearman's autocorrelation to lag 1 (ρ) is computed. Then the default bandwidth follows as

$$b_n = \max \left\{ \left\lceil n^{0.25} \left(\frac{2\rho}{1-\rho^2} \right)^{0.8} \right\rceil, 1 \right\}.$$

Otherwise, the default for `distr` is `FALSE` and the default bandwidth is

$$b_n = \max \left\{ \left\lceil n^{0.4} \left(\frac{2\rho}{1-\rho^2} \right)^{1/3} \right\rceil, 1 \right\}.$$

Value

Test statistic (numeric value) with the following attributes:

`cp-location`	indicating at which index a change point is most likely.
`teststat`	test process (before taking the maximum).
`lrv-estimation`	long run variance estimation method.
`sigma`	estimated long run variance.
`param`	parameter used for the lrv estimation.
`kFun`	kernel function used for the lrv estimation.

Is an S3 object of the class "cpStat".

Author(s)

Sheila Görz

References

Dehling, H., et al. "Change-point detection under dependence based on two-sample U-statistics." Asymptotic laws and methods in stochastics. Springer, New York, NY, 2015. 195-220.

See Also

`lrv`

Examples

```
# time series with a location change at t = 20
x <- c(rnorm(20, 0), rnorm(20, 2))

# Wilcoxon-Mann-Whitney change point test statistic
wilcox_stat(x, h = 1L, control = list(b_n = length(x)^(1/3)))
```

---

wmw_test *Wilocxon-Mann-Whitney Test for Change Points*

Description

Performs the Wilcoxon-Mann-Whitney change point test.

Usage

```
wmw_test(x, h = 1L, method = "kernel", control = list(), tol = 1e-8,
         plot = FALSE)
```

Arguments

x	time series (numeric or ts vector).
h	version of the test (integer, 1L or 2L)
method	method for estimating the long run variance.
control	a list of control parameters (cf. lrv).
tol	tolerance of the distribution function (numeric), which is used to compute p-values.
plot	should the test statistic be plotted (cf. plot.cpStat). Boolean.

Details

The function performs a Wilcoxon-Mann-Whitney change point test. It tests the hypothesis pair

$$H_0 : \mu_1 = ... = \mu_n$$

$$vs.$$

$$H_1 : \exists k \in \{1, ..., n-1\} : \mu_k \neq \mu_{k+1}$$

where $\mu_t = E(X_t)$ and n is the length of the time series. k is called a 'change point'.

The test statistic is computed using wilcox_stat and asymptotically follows a Kolmogorov distribution. To derive the p-value, the function pKSdist is used.

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).
cp.location	index of the estimated change point location (integer).
data.name	name of the data (character string).

Author(s)

Sheila Görz

References

Dehling, H., et al. "Change-point detection under dependence based on two-sample U-statistics." Asymptotic laws and methods in stochastics. Springer, New York, NY, 2015. 195-220.

See Also

wilcox_stat, lrv, pKSdist

Examples

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

wmw_test(Z, h = 1L, control = list(overlapping = TRUE, b_n = 5))
```

`zeros` *Zero of the Bessel function of first kind*

Description

Contains the zeros of the Bessel function of first kind.

Usage

```
data("zeros")
```

Format

A data frame where the i-th 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

Eaton, J., Bateman, D., Hauberg, S., Wehbring, R. (2015). "GNU Octave version 4.0.0 manual: a high-level interactive language for numerical computations".

Index

∗ **datasets**
    zeros, 37

bw.nrd0, *9, 10*

cor_cusum, 3, *3*, *7*
cor_stat, *3, 4*, 5
CUSUM, *3*, 7, *12*, *14*, *20*, *22–25*, *27*

density, *11*

ginv, *7*

hl_test, *3*, 9
HodgesLehmann, *3*, *9*, 10, *10*, *20*, *24, 25*
huber_cusum, *3*, 12, *23*

kthPair, 14, *21*

lrv, *4*, *6–14*, 15, *30–32*, *34–36*

medianDiff, *10, 11*, 20
modifChol, *7*, 21

par, *24*
pBessel *(*pKSdist*)*, 22
pKSdist, *4*, *9, 10*, *14*, 22, *29, 30*, *36*
plot, *24*
plot.cpStat, *3*, *9*, *12*, 24, *29*, *36*
print, *25*
print.cpStat, 25
psi, *3*, *8*, *12–14*, *23*, 25, *27, 28*
psi_cumsum, *8*, *14*, *23*, 27, *27*

Qalpha, *17*, 28, *30*, *32*

robcp *(*robcp-package*)*, 2
robcp-package, 2

scale_cusum, *3*, 29
scale_stat, *3*, *29, 30*, 31
set.seed, *19*
sqrtm, *19*

u_hat, *9, 10*
u_hat *(*HodgesLehmann*)*, 10

weightedMedian, 33
wilcox_stat, *3*, *20*, *24, 25*, 34, *36*
wmw_test, *3*, 35

zeros, 37