Package ‘rtrend’

April 3, 2022

Title  Trend Estimating Tools

Description  The traditional linear regression trend, Modified Mann-Kendall (MK) non-parameter trend and bootstrap trend are included in this package. Linear regression trend is rewritten by ‘.lm.fit’. MK trend is rewritten by ‘Rcpp’. Finally, those functions are about 10 times faster than previous version in R.


Version  0.1.3

License  MIT + file LICENSE

Encoding  UTF-8

RoxygenNote  7.1.2

LinkingTo  Rcpp, RcppArmadillo

Imports  Rcpp, fftwtools, boot, magrittr, matrixStats, terra, plyr, ggplot2

Suggests  covr, testthat (>= 3.0.0)

URL  https://github.com/rpkgs/rtrend

BugReports  https://github.com/rpkgs/rtrend/issues

Config/testthat/edition  3

NeedsCompilation  yes

Author  Dongdong Kong [aut, cre] (<https://orcid.org/0000-0003-1836-8172>), Heyang Song [aut] (<https://orcid.org/0000-0002-4192-5603>)

Maintainer  Dongdong Kong <kongdd.sysu@gmail.com>

Repository  CRAN

Date/Publication  2022-04-03 19:20:02 UTC
**R topics documented:**

- `mkTrend_r` ......................................................... 2
- `movmean` .......................................................... 3
- `slope_sen` .......................................................... 4
- `smooth_wSG` ....................................................... 5
- `summary_lm` ......................................................... 6

**Description**

If valid observations <= 5, NA will be returned.

**Usage**

```r
mkTrend_r(y, ci = 0.95, IsPlot = FALSE)
mkTrend(y, x = seq_along(y), ci = 0.95, IsPlot = FALSE)
```

**Arguments**

- `y` numeric vector
- `ci` critical value of autocorrelation
- `IsPlot` boolean
- `x` (optional) numeric vector

**Details**

`mkTrend` is 4-fold faster with `.lm.fit`.

**Value**

- `Z0` : The original (non corrected) Mann-Kendall test Z statistic.
- `pval0` : The original (non corrected) Mann-Kendall test p-value
- `Z` : The new Z statistic after applying the correction
- `pval` : Corrected p-value after accounting for serial autocorrelation $N/n*s$ Value of the correction factor, representing the quotient of the number of samples N divided by the effective sample size $n*s$
- `slp` : Sen slope, The slope of the (linear) trend according to Sen test

**Note**

`slp` is significant, if `pval < alpha`.
Author(s)

Dongdong Kong

References


See Also

fume::mktrend and trend::mk.test

Examples

```r
x <- c(4.81, 4.17, 4.41, 3.59, 5.87, 3.83, 6.03, 4.89, 4.32, 4.69)
 r <- mkTrend(x)
 r_cpp <- mkTrend(x, IsPlot = TRUE)
```

Description

NA and Inf values in the y will be ignored automatically.

Usage

```r
movmean(y, halfwin = 1L, SG_style = FALSE, w = NULL)
movmean2(y, win_left = 1L, win_right = 0L, w = NULL)
movmean_2d(mat, win_left = 3L, win_right = 0L)
```

Arguments

- **y** A numeric vector.
- **halfwin** Integer, half of moving window size
- **SG_style** If true, head and tail values will be in the style of SG (more weights on the center point), else traditional moving mean style.
- **w** Corresponding weights of y, with the same length.
- **win_left, win_right** windows size in the left and right
- **mat** numeric matrix
Examples

```r
x <- 1:100
x[50] <- NA; x[80] <- Inf
s1 <- movmean(x, 2, SG_style = TRUE)
s2 <- movmean(x, 2, SG_style = FALSE)
movmean2(c(4, 8, 6, -1, -2, -3, -1), 2, 0)
movmean2(c(4, 8, NA, -1, -2, Inf, -1), 2, 0)
```

Description

- `slope`: linear regression slope
- `slope_p`: linear regression slope and p-value
- `slope_mk`: mann kendall Sen’s slope and p-value
- `slope_sen`: same as `slope_mk`, but with no p-value
- `slope_boot`: bootstrap slope and p-value

Usage

```r
slope_sen(y, x = NULL)
slope(y, x, ...)
slope_p(y, x, fast = TRUE)
slope_sen_r(y, x = seq_along(y), ...)
slope_mk(y, x = NULL, ...)
slope_boot(y, x = NULL, slope_FUN = slope, times = 100, alpha = 0.1, seed, ...)
```

Arguments

- `y` vector of observations of length n, or a matrix with n rows.
- `x` vector of predictor of length n, or a matrix with n rows.
- `...` ignored.
- `fast` Boolean. If true, `stats::.lm.fit()` will be used, which is 10x faster than `stats::.lm()`.
- `slope_FUN` one of `slope()`, `slope_p()`, `slope_mk()`
- `times` The number of bootstrap replicates.
- `alpha` significant level, default 0.1
- `seed` a single value, interpreted as an integer, or NULL (see ‘Details’).
smooth_wSG

Value

• slope: linear regression coefficient
• pvalue: p-value <= 0.05 means that corresponding slope is significant.
• sd: Std. Error

For slope_boot, slope is estimated in many times. The lower, mean, upper and standard deviation (sd) are returned.

Examples

```r
y <- c(4.81, 4.17, 4.41, 3.59, 5.87, 3.83, 6.03, 4.89, 4.32, 4.69)
r <- slope(y)
r_p <- slope_p(y)
r_mk <- slope_mk(y)
r_boot <- slope_boot(y)
```

smooth_wSG

Weighted Savitzky-Golay

Description

NA and Inf values in the y has been ignored automatically.

Usage

```
smooth_wSG(y, halfwin = 1L, d = 1L, w = NULL)
smooth_SG(y, halfwin = 1L, d = 1L)
```

Arguments

- **y**: colvec
- **halfwin**: halfwin of Savitzky-Golay
- **d**: polynomial of degree. When d = 1, it becomes moving average.
- **w**: colvec of weight

Examples

```
y <- c(1, 3, 2, 5, 6, 8, 10, 1)
w <- seq_along(y)/length(y)

halfwin = 2
d = 2
s1 <- smooth_wSG(y, halfwin, d, w)
s2 <- smooth_SG(y, halfwin, d)
```
**Description**

summary method for class ".lm.fit". It’s 200 times faster than traditional `lm`.

**Usage**

`summary_lm(obj, ...)`

**Arguments**

- `obj` Object returned by `.lm.fit`.
- `...` ignored

**Value**

a p x 4 matrix with columns for the estimated coefficient, its standard error, t-statistic and corresponding (two-sided) p-value. Aliased coefficients are omitted.

**Examples**

```r
set.seed(129)
n <- 100
p <- 2
X <- matrix(rnorm(n * p), n, p) # no intercept!
y <- rnorm(n)

obj <- .lm.fit(x = cbind(1, X), y = y)
info <- summary_lm(obj)
```
Index

.lm.fit, 6
mkTrend (mkTrend_r), 2
mkTrend_r, 2
movmean, 3
movmean2 (movmean), 3
movmean_2d (movmean), 3
slope (slope_sen), 4
slope(), 4
slope_boot (slope_sen), 4
slope_mk (slope_sen), 4
slope_mk(), 4
slope_p (slope_sen), 4
slope_p(), 4
slope_sen, 4
slope_sen_r (slope_sen), 4
smooth_SG (smooth_wSG), 5
smooth_wSG, 5
stats::.lm.fit(), 4
stats::lm(), 4
summary_lmm, 6