

Package ‘rtrend’

May 11, 2021

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.

Reference:

Hamed, K. H., & Rao, A. R. (1998). A modified Mann-Kendall trend test for autocorrelated data. *Journal of hydrology*, 204(1-4), 182-196.
<doi:10.1016/S0022-1694(97)00125-X>.

Version 0.1.0

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.1.1

LinkingTo Rcpp

Imports Rcpp, fftwtools, boot, magrittr, matrixStats, 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>>)

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

Repository CRAN

Date/Publication 2021-05-11 09:00:02 UTC

R topics documented:

mkTrend_r	2
slope	3
stat_mk	4
summary_lm	5

mkTrend_r	<i>Modified Mann Kendall</i>
-----------	------------------------------

Description

If valid observations ≤ 5 , NA will be returned.

Usage

```
mkTrend_r(x, ci = 0.95, IsPlot = FALSE)
```

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

Arguments

x	numeric vector
ci	critical value of autocorrelation
IsPlot	boolean
y	numeric vector

Details

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

Value

- Z_0 : The original (non corrected) Mann-Kendall test Z statistic.
- $pval_0$: 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

Hipel, K.W. and McLeod, A.I. (1994), *Time Series Modelling of Water Resources and Environmental Systems*. New York: Elsevier Science.

Libiseller, C. and Grimvall, A., (2002), Performance of partial Mann-Kendall tests for trend detection in the presence of covariates. *Environmetrics* 13, 71–84, doi: [10.1002/env.507](https://doi.org/10.1002/env.507).

See Also

fume::mktrend and trend::mk.test

Examples

```
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)
```

slope

slope

Description

- slope : linear regression slope
- slope_p : linear regression slope and p-value
- slope_mk : mann kendall Sen's slope and p-value
- slope_boot: bootstrap slope and p-value

Usage

```
slope(y, x)
```

```
slope_p(y, x, fast = TRUE)
```

```
slope_mk(x)
```

```
slope_boot(y, 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.
fast	Boolean. If true, <code>stats::lm.fit()</code> will be used, which is 10x faster than <code>stats::lm()</code> .
slope_FUN	one of <code>slope()</code> , <code>slope_p()</code> , <code>slope_mk()</code>
times	The number of bootstrap replicates.
alpha	significant level, default 0.1
seed	a single value, interpreted as an integer, or NULL (see 'Details').

Value

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

Examples

```
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)
```

stat_mk	<i>stat_mk</i>
---------	----------------

Description

stat_mk

Usage

```
stat_mk(
  mapping = NULL,
  data = NULL,
  geom = "line",
  position = "identity",
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE,
  ...
)
```

Arguments

mapping	Set of aesthetic mappings created by <code>aes()</code> or <code>aes_()</code> . If specified and <code>inherit.aes = TRUE</code> (the default), it is combined with the default mapping at the top level of the plot. You must supply mapping if there is no plot mapping.
data	<p>The data to be displayed in this layer. There are three options:</p> <p>If <code>NULL</code>, the default, the data is inherited from the plot data as specified in the call to <code>ggplot()</code>.</p> <p>A <code>data.frame</code>, or other object, will override the plot data. All objects will be fortified to produce a data frame. See <code>fortify()</code> for which variables will be created.</p> <p>A function will be called with a single argument, the plot data. The return value must be a <code>data.frame</code>, and will be used as the layer data. A function can be created from a formula (e.g. <code>~ head(.x, 10)</code>).</p>

<code>geom</code>	Use to override the default connection between <code>geom_smooth()</code> and <code>stat_smooth()</code> .
<code>position</code>	Position adjustment, either as a string, or the result of a call to a position adjustment function.
<code>na.rm</code>	If FALSE, the default, missing values are removed with a warning. If TRUE, missing values are silently removed.
<code>show.legend</code>	logical. Should this layer be included in the legends? NA, the default, includes if any aesthetics are mapped. FALSE never includes, and TRUE always includes. It can also be a named logical vector to finely select the aesthetics to display.
<code>inherit.aes</code>	If FALSE, overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn't inherit behaviour from the default plot specification, e.g. <code>borders()</code> .
<code>...</code>	Other arguments passed on to <code>layer()</code> . These are often aesthetics, used to set an aesthetic to a fixed value, like <code>colour = "red"</code> or <code>size = 3</code> . They may also be parameters to the paired geom/stat.

Value

No return. This function is used to calculate data for ggplot2 geom_*, just like `ggplot2::stat_smooth()`.

Examples

```
library(ggplot2)

ggplot(mpg, aes(displ, hwy, colour = drv)) +
  geom_point() +
  stat_mk()
```

summary_lm

*summary_lm***Description**

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

Usage

```
summary_lm(obj, ...)
```

Arguments

<code>obj</code>	Object returned by <code>.lm.fit</code> .
<code>...</code>	ignored

Value

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

Examples

```
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`, 5
`aes()`, 4
`aes_()`, 4
`borders()`, 5
`fortify()`, 4
`ggplot()`, 4
`ggplot2::stat_smooth()`, 5
`layer()`, 5
`mkTrend (mkTrend_r)`, 2
`mkTrend_r`, 2
`slope`, 3
`slope()`, 3
`slope_boot (slope)`, 3
`slope_mk (slope)`, 3
`slope_mk()`, 3
`slope_p (slope)`, 3
`slope_p()`, 3
`stat_mk`, 4
`stats::.lm.fit()`, 3
`stats::lm()`, 3
`summary_lm`, 5