

Package ‘zyp’

January 20, 2025

Version 0.11-1

Date 2023-03-22

Title Zhang + Yue-Pilon Trends Package

Depends R (>= 2.4.0), Kendall

Description An efficient implementation of the slope method described by Sen (1968) <[doi:10.1080/01621459.1968.10480934](https://doi.org/10.1080/01621459.1968.10480934)> plus implementation of prewhitening approaches to determining trends in climate data described by Zhang, Vincent, Hogg, and Niitsoo (2000) <[doi:10.1080/07055900.2000.9649654](https://doi.org/10.1080/07055900.2000.9649654)> and Yue, Pilon, Phinney, and Cavadas (2002) <[doi:10.1002/hyp.1095](https://doi.org/10.1002/hyp.1095)>.

License LGPL-2.1

URL <https://www.r-project.org>

NeedsCompilation no

Author David Bronaugh [aut],
Arelia Schoeneberg [aut],
Lee Zeman [cre]

Maintainer Lee Zeman <lzeman@uvic.ca>

Repository CRAN

Date/Publication 2023-03-22 19:20:02 UTC

Contents

| | |
|----------------------------|----------|
| confint.zyp | 2 |
| zyp | 3 |
| zyp.sen | 4 |
| zyp.trend.csv | 5 |
| zyp.trend.vector | 6 |
| Index | 9 |

`confint.zyp``confint.zyp`

Description

Computes a confidence interval on a result from `zyp.sen`.

Usage

```
## S3 method for class 'zyp'  
confint(object, parm, level = 0.95, ...)
```

Arguments

| | |
|---------------------|--|
| <code>object</code> | an object of class <code>zyp</code> , returned by <code>zyp.sen</code> . |
| <code>parm</code> | unused; for compatibility with <code>confint</code> . |
| <code>level</code> | the confidence level to compute a confidence interval at. |
| <code>...</code> | additional unused arguments. |

Details

This routine computes a confidence interval on the slope and intercept of the result returned by `zyp.sen`. The confidence interval on the slope is calculated using the method defined in (Sen, 1968).

The confidence interval on the intercept is computed by taking the standard deviation of the intercepts (`sd.i`), the z statistic for the given confidence level (`z`), and the mean (`m`). The confidence level is then:

```
c(m - z * sd.i, m + z * sd.i)
```

Value

A matrix containing the upper and lower bounds of the intercept and slope, respectively.

See Also

[zyp.trend.vector](#), [zyp-package](#).

Examples

```
x <- c(0, 1, 2, 4, 5)  
y <- c(6, 4, 1, 8, 7)  
slope <- zyp.sen(y~x)  
ci <- confint.zyp(slope)
```

Description

Update April 1 2022

At the time of writing, the application of the Yue Pilon (Yue et al. 2002) is actively discouraged in the hydrologic community. This is because the trend free whitening procedure (TFWP) applied in the Yue Pilon method shows very high Type-I error rates with increasing autocorrelation and hence the rate of false trend detection with this method is unacceptable (Buerger 2017; Zhang and Zwiers 2004). This version of zyp has been modified to apply Zhang et al. 2000 trend detection methods as default.

Additionally, `preserve.range.for.sig.test` is set to TRUE throughout to make sure significance is re-inflated as per the Yue and Pilon paper exactly.

This function includes two approaches to analyze for trend, the Zhang method and Yue and Pilon method. These differ in their approach to pre-whitening to removing lag-1 autocorrelation. The magnitude of the trend is computed using the Theil-Sen approach (TSA).

In the Zhang method, trend is removed from the series if it is significant and the autocorrelation is computed. This process is continued until the differences in the estimates of the slope and the AR(1) in two consecutive iterations are smaller than 1 percent. The Mann-Kendall test for trend is then run on the resulting time series and TSA is used to compute the slope of the trend.

In the Yue and Pilon method, the slopes is estimated with the TSA, if almost equal to zero, then its is not necessary to conduct the trend analysis. If it differs from zero, then it is assumed to be linear and the data is detrended by the slope and the AR(1) is computed for the detrended series. This is referred to as the Trend Free Pre-whitening (TFPW) procedure. The residuals should be an independent series. The trend and residuals are then blended together. The Mann-Kendall test is the applied to the blended series to assess the significance of the trend.

One variance of this package, at least from the Yue and Pilon method, is that by default the values used to compute significance are re-inflated by dividing by $(1 - \text{AR}(1))$. Empirically this provides better results. However, if you do not desire this behaviour, it can be controlled using the parameter 'preserve.range.for.sig.test'; setting this to 'FALSE' should give results which follow the Yue and Pilon paper exactly.

References

- Buerger, G., 2017. On trend detection. *Hydrological Processes* 31, 4039-4042.
- Wang, X.L. and Swail, V.R., 2001. Changes in extreme wave heights in northern hemisphere oceans and related atmospheric circulation regimes. *Journal of Climate*, 14: 2204-2221.
- Yue, S., P. Pilon, B. Phinney and G. Cavadias, 2002. The influence of autocorrelation on the ability to detect trend in hydrological series. *Hydrological Processes*, 16: 1807-1829.
- Zhang, X., Vincent, L.A., Hogg, W.D. and Niitsoo, A., 2000. Temperature and Precipitation Trends in Canada during the 20th Century. *Atmosphere-Ocean* 38(3): 395-429.
- Zhang, X., Zwiers, F.W., 2004. Comment on "Applicability of prewhitening to eliminate the influence of serial correlation on the Mann-Kendall test" by Sheng Yue and Chun Yuan Wang. *Water Resources Research* 40.

See Also

[zyp.trend.csv](#), [zyp.trend.vector](#).

`zyp.sen`

zyp.sen

Description

Computes a Thiel-Sen estimate of slope for a vector of data.

Usage

```
zyp.sen(formula, dataframe)
```

Arguments

| | |
|------------------------|---|
| <code>formula</code> | a formula of the form $y \sim x$. |
| <code>dataframe</code> | an optional data frame to use with the formula. |

Details

This routine computes Sen's estimate of slope for a vector of data.

The formula specified is of the form $y \sim x$, where y is the data and x is the accompanying dates of observation.

If a data frame is specified, the formula selects columns within that data frame; otherwise, the formula uses the specified variables.

Value

A vector containing the coefficients and associated data.

| | |
|---------------------------|---|
| <code>coefficients</code> | the estimated intercept and slope. |
| <code>slopes</code> | the list of slopes (used for confidence intervals). |
| <code>intercepts</code> | the list of intercepts (used for confidence intervals). |
| <code>rank</code> | the rank of the relation (2). |
| <code>residuals</code> | the residuals of the fitted slope. |
| <code>x</code> | the original x axis data. |
| <code>y</code> | the original y axis data. |

See Also

[zyp.trend.vector](#), [zyp-package](#), [confint.zyp](#).

Examples

```
x <- c(0, 1, 2, 4, 5)
y <- c(6, 4, 1, 8, 7)
slope <- zyp.sen(y~x)
```

| | |
|---------------|----------------------|
| zyp.trend.csv | <i>zyp.trend.csv</i> |
|---------------|----------------------|

Description

Computes prewhitened nonlinear trends on CSV files or data frames with 0 to n columns of metadata, with 1 row per location and each column containing data for a particular time (day, month, year). The zyp package allows you to use either Zhang's method, or the Yue Pilon method of computing nonlinear prewhitened trends.

Usage

```
zyp.trend.dataframe(indat, metadata.cols, method=c("zhang", "yuepilon"),
                    conf.intervals=TRUE, preserve.range.for.sig.test=FALSE)
zyp.trend.csv(filename, output.filename, metadata.cols,
              method=c("zhang", "yuepilon"), conf.intervals=TRUE,
              csv.header=TRUE, preserve.range.for.sig.test=FALSE)
```

Arguments

| | |
|-----------------------------|--|
| indat | the input data frame. |
| filename | the filename of the input CSV file. |
| output.filename | the filename to write output to. |
| metadata.cols | the number of columns of metadata. |
| method | the prewhitened trend method to use. |
| conf.intervals | whether to compute a 95 percent confidence interval based on all possible slopes. |
| preserve.range.for.sig.test | whether to re-inflate values by dividing by (1 - ac) following removal of auto-correlation prior to computation of significance. |
| csv.header | whether the input CSV file has a header. |

Details

These routines compute prewhitened nonlinear trends on either CSV files with or without a header or data frames with 0 to n columns of metadata (which is preserved in the output). Each row is expected to contain metadata followed by a timeseries, and all rows are expected to have the same length of timeseries. NA values are handled correctly, so if you have several timeseries of unequal length you can pad them with NA values to provide valid input.

The prewhitened trend computation methods used are either Zhang's method (described in Wang and Swail, 2001) or Yue and Pilon's method (described in Yue and Pilon, 2002).

Value

A data frame containing the trends, in the case of `zyp.trend.dataframe`. Columns of the output are as follows.

| | |
|-------------------------------|---|
| <code>lbound</code> | the lower bound of the trend's confidence interval. |
| <code>trend</code> | the Sen's slope (trend) per unit time. |
| <code>trendp</code> | the Sen's slope (trend) over the time period. |
| <code>ubound</code> | the upper bound of the trend's confidence interval. |
| <code>tau</code> | Kendall's tau statistic computed on the final detrended timeseries. |
| <code>sig</code> | Kendall's P-value computed for the final detrended timeseries. |
| <code>nruns</code> | the number of runs required to converge upon a trend. |
| <code>autocor</code> | the autocorrelation of the final detrended timeseries. |
| <code>valid_frac</code> | the fraction of the data which is valid (not NA) once autocorrelation is removed. |
| <code>linear</code> | the least squares fit trend on the same dat. |
| <code>intercept</code> | the intercept of the Sen's slope (trend). |
| <code>lbound_intercept</code> | the lower bound of the estimate of the intercept of the Sen's slope (trend). |
| <code>ubound_intercept</code> | the upper bound of the estimate of the intercept of the Sen's slope (trend). |

See Also

[zyp.trend.vector](#).

Examples

```
## Not run:
zyp.trend.csv("in.csv", "out.csv", 2, "yuepilon", F)
trends <- zyp.trend.dataframe(indat, 2, "yuepilon")

## End(Not run)
```

`zyp.trend.vector` *zyp.trend.vector*

Description

Computes a prewhitened linear trend on a vector of data. The 'zyp' package allows you to use either Zhang's method, or the Yue Pilon method of computing nonlinear prewhitened trends.

Usage

```
zyp.trend.vector(y, x=1:length(y), method=c("zhang", "yuepilon"),
  conf.intervals=TRUE, preserve.range.for.sig.test=FALSE)
zyp.zhang(y, x=1:length(y), conf.intervals=TRUE, preserve.range.for.sig.test=FALSE)
zyp.yuepilon(y, x=1:length(y), conf.intervals=TRUE, preserve.range.for.sig.test=FALSE)
```

Arguments

| | |
|-----------------------------|---|
| y | vector of input data. |
| x | vector of time data (optional). |
| method | the prewhitened trend method to use. |
| conf.intervals | whether to compute a 95 percent confidence interval based on all possible slopes. |
| preserve.range.for.sig.test | whether to re-inflate values by dividing by (1 - ac) following removal of autocorrelation prior to computation of significance. |

Details

This routine computes a prewhitened nonlinear trend on a vector of data, using either Zhang's (described in Wang and Swail, 2001) or Yue Pilon's (describe in Yue Pilon, 2002) method of prewhitening and Sen's slope, and use a Kendall test for significance.

Value

A vector containing the trend and associated data.

| | |
|------------------|---|
| lbound | the lower bound of the trend's confidence interval. |
| trend | the Sen's slope (trend) per unit time. |
| trendp | the Sen's slope (trend) over the time period. |
| ubound | the upper bound of the trend's confidence interval. |
| tau | Kendall's tau statistic computed on the final detrended timeseries. |
| sig | Kendall's P-value computed for the final detrended timeseries. |
| nruns | the number of runs required to converge upon a trend. |
| autocor | the autocorrelation of the final detrended timeseries. |
| valid_frac | the fraction of the data which is valid (not NA) once autocorrelation is removed. |
| linear | the least squares fit trend on the same dat. |
| intercept | the intercept of the Sen's slope (trend). |
| lbound_intercept | the lower bound of the estimate of the intercept of the Sen's slope (trend). |
| ubound_intercept | the upper bound of the estimate of the intercept of the Sen's slope (trend). |

See Also

[zyp.trend.csv](#), [zyp-package](#), [confint.zyp](#), [zyp.sen](#).

Examples

```
# Without confidence intervals, using the wrapper routine
d <- zyp.trend.vector(c(0, 1, 3, 4, 2, 5), conf.intervals=FALSE)

# With confidence intervals, using the wrapper routine
d <- zyp.trend.vector(c(0, 1, 3, 4, 2, 5))

# With confidence intervals, not using the wrapper routine
d.zhang <- zyp.zhang(c(0, 1, 3, 4, 2, 5))
d.yuepilon <- zyp.yuepilon(c(0, 1, 3, 4, 2, 5))

# With confidence intervals, with time data.
t.dat <- c(0, 0.3, 1, 3, 3.4, 6)
d <- zyp.trend.vector(c(0, 1, 3, 4, 2, 5), t.dat, method="yuepilon")
d.zhang <- zyp.zhang(c(0, 1, 3, 4, 2, 5), t.dat)
d.yuepilon <- zyp.yuepilon(c(0, 1, 3, 4, 2, 5), t.dat)
```


Index

* **robust**

- confint.zyp, 2
- zyp, 3
- zyp.sen, 4
- zyp.trend.csv, 5
- zyp.trend.vector, 6

* **ts**

- confint.zyp, 2
- zyp, 3
- zyp.sen, 4
- zyp.trend.csv, 5
- zyp.trend.vector, 6

confint.zyp, 2, 4, 7

zyp, 3

zyp-package, 2, 4, 7

zyp-package (zyp), 3

zyp.sen, 4, 7

zyp.trend.csv, 4, 5, 7

zyp.trend.dataframe (zyp.trend.csv), 5

zyp.trend.vector, 2, 4, 6, 6

zyp.yuepilon (zyp.trend.vector), 6

zyp.zhang (zyp.trend.vector), 6