Package: tsfeatures (via r-universe)

October 21, 2024

Title Time Series Feature Extraction

Version 1.1.1.9000

Description Methods for extracting various features from time series data. The features provided are those from Hyndman, Wang and Laptev (2013) <doi:10.1109/ICDMW.2015.104>, Kang, Hyndman and Smith-Miles (2017) <doi:10.1016/j.ijforecast.2016.09.004> and from Fulcher, Little and Jones (2013) <doi:10.1098/rsif.2013.0048>. Features include spectral entropy, autocorrelations, measures of the strength of seasonality and trend, and so on. Users can also define their own feature functions.

Depends R (>= 3.6.0)

Imports fracdiff, forecast (>= 8.3), purrr, RcppRoll (>= 0.2.2), stats, tibble, tseries, urca, future, furrr

Suggests testthat, knitr, rmarkdown, ggplot2, tidyr, dplyr, Mcomp, GGally

License GPL-3

ByteCompile true

URL `https://pkg.robjhyndman.com/tsfeatures/`, `https://github.com/robjhyndman/tsfeatures`

BugReports `https://github.com/robjhyndman/tsfeatures/issues`

RoxygenNote 7.2.3

Roxygen list(markdown = TRUE, roclets=c('rd', 'collate', 'namespace'))

VignetteBuilder knitr

Encoding UTF-8

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

RemoteUrl https://github.com/robjhyndman/tsfeatures

RemoteRef HEAD

RemoteSha dfaff350918c4455cd31edd9e57a4429a9f791fc

Contents

`acf_features` *Autocorrelation-based features*

Description

Computes various measures based on autocorrelation coefficients of the original series, first-differenced series and second-differenced series

Usage

```
acf_features(x)
```

Arguments

`x`	a univariate time series

Value

A vector of 6 values: first autocorrelation coefficient and sum of squared of first ten autocorrelation coefficients of original series, first-differenced series, and twice-differenced series. For seasonal data, the autocorrelation coefficient at the first seasonal lag is also returned.

Author(s)

Thiyanga Talagala

`ac_9` *Autocorrelation at lag 9. Included for completion and consistency.*

Description

Autocorrelation at lag 9. Included for completion and consistency.

Usage

```
ac_9(y, acfv = stats::acf(y, 9, plot = FALSE, na.action = na.pass))
```

Arguments

`y`	the input time series
`acfv`	vector of autocorrelation, if exist, used to avoid repeated computation.

Value

autocorrelation at lag 9

Author(s)

Yangzhuoran Yang

References

B.D. Fulcher and N.S. Jones. hctsa: A computational framework for automated time-series phenotyping using massive feature extraction. Cell Systems 5, 527 (2017).

B.D. Fulcher, M.A. Little, N.S. Jones Highly comparative time-series analysis: the empirical structure of time series and their methods. J. Roy. Soc. Interface 10, 83 (2013).

`arch_stat` *ARCH LM Statistic*

Description

Computes a statistic based on the Lagrange Multiplier (LM) test of Engle (1982) for autoregressive conditional heteroscedasticity (ARCH). The statistic returned is the R^2 value of an autoregressive model of order `lags` applied to x^2.

Usage

```
arch_stat(x, lags = 12, demean = TRUE)
```

Arguments

`x`	a univariate time series
`lags`	Number of lags to use in the test
`demean`	Should data have mean removed before test applied?

Value

A numeric value.

Author(s)

Yanfei Kang

`as.list.mts` *Convert mts object to list of time series*

Description

An mts object contains a multivariate time series in a matrix, with time on rows. This is converted into a list of univariate time series.

Usage

```
## S3 method for class 'mts'
as.list(x, ...)
```

Arguments

`x`	multivariate time series of class mts.
`...`	other arguments are ignored.

Value

A list of ts objects.

Author(s)

Rob J Hyndman

`autocorr_features` *The autocorrelation feature set from software package* `hctsa`

Description

Calculate the features that grouped as autocorrelation set, which have been used in CompEngine database, using method introduced in package `hctsa`.

Usage

```
autocorr_features(x)
```

Arguments

`x`	the input time series

Details

Features in this set are `embed2_incircle_1`, `embed2_incircle_2`, `ac_9`, `firstmin_ac`, `trev_num`, `motiftwo_entro3`, and `walker_propcross`.

Value

a vector with autocorrelation features

Author(s)

Yangzhuoran Yang

References

B.D. Fulcher and N.S. Jones. hctsa: A computational framework for automated time-series phenotyping using massive feature extraction. Cell Systems 5, 527 (2017).

B.D. Fulcher, M.A. Little, N.S. Jones Highly comparative time-series analysis: the empirical structure of time series and their methods. J. Roy. Soc. Interface 10, 83 (2013).

See Also

`embed2_incircle`

`ac_9`

`firstmin_ac`

`trev_num`

`motiftwo_entro3`

`walker_propcross`

`binarize_mean` *Converts an input vector into a binarized version from software package* `hctsa`

Description

Converts an input vector into a binarized version from software package `hctsa`

Usage

```
binarize_mean(y)
```

Arguments

`y`	the input time series

Value

Time-series values above its mean are given 1, and those below the mean are 0.

Author(s)

Yangzhuoran Yang

References

B.D. Fulcher and N.S. Jones. hctsa: A computational framework for automated time-series phenotyping using massive feature extraction. Cell Systems 5, 527 (2017).

B.D. Fulcher, M.A. Little, N.S. Jones Highly comparative time-series analysis: the empirical structure of time series and their methods. J. Roy. Soc. Interface 10, 83 (2013).

`compengine` *CompEngine feature set*

Description

Calculate the features that have been used in CompEngine database, using method introduced in package `hctsa`.

Usage

```
compengine(x)
```

Arguments

`x`	the input time series

Details

The features involved can be grouped as `autocorrelation`, `prediction`, `stationarity`, `distribution`, and `scaling`.

Value

a vector with CompEngine features

Author(s)

Yangzhuoran Yang

References

B.D. Fulcher and N.S. Jones. hctsa: A computational framework for automated time-series phenotyping using massive feature extraction. Cell Systems 5, 527 (2017).

B.D. Fulcher, M.A. Little, N.S. Jones Highly comparative time-series analysis: the empirical structure of time series and their methods. J. Roy. Soc. Interface 10, 83 (2013).

See Also

`autocorr_features`

`pred_features`

`station_features`

`dist_features`

`scal_features`

`crossing_points` *Number of crossing points*

Description

Computes the number of times a time series crosses the median.

Usage

```
crossing_points(x)
```

Arguments

`x`	a univariate time series

Value

A numeric value.

Author(s)

Earo Wang and Rob J Hyndman

`dist_features` *The distribution feature set from software package* `hctsa`

Description

Calculate the features that grouped as distribution set, which have been used in CompEngine database, using method introduced in package `hctsa`.

Usage

```
dist_features(x)
```

Arguments

`x`	the input time series

Details

Features in this set are `histogram_mode_10` and `outlierinclude_mdrmd`.

Value

a vector with distribution features

Author(s)

Yangzhuoran Yang

References

B.D. Fulcher and N.S. Jones. hctsa: A computational framework for automated time-series phenotyping using massive feature extraction. Cell Systems 5, 527 (2017).

B.D. Fulcher, M.A. Little, N.S. Jones Highly comparative time-series analysis: the empirical structure of time series and their methods. J. Roy. Soc. Interface 10, 83 (2013).

See Also

`histogram_mode`

`outlierinclude_mdrmd`

`embed2_incircle` *Points inside a given circular boundary in a 2-d embedding space from software package* `hctsa`

Description

The time lag is set to the first zero crossing of the autocorrelation function.

Usage

```
embed2_incircle(
  y,
  boundary = NULL,
  acfv = stats::acf(y, length(y) - 1, plot = FALSE, na.action = na.pass)
)
```

Arguments

`y`	the input time series
`boundary`	the given circular boundary, setting to 1 or 2 in CompEngine. Default to 1.
`acfv`	vector of autocorrelation, if exist, used to avoid repeated computation.

Value

the proportion of points inside a given circular boundary

Author(s)

Yangzhuoran Yang

References

B.D. Fulcher and N.S. Jones. hctsa: A computational framework for automated time-series phenotyping using massive feature extraction. Cell Systems 5, 527 (2017).

B.D. Fulcher, M.A. Little, N.S. Jones Highly comparative time-series analysis: the empirical structure of time series and their methods. J. Roy. Soc. Interface 10, 83 (2013).

`entropy` *Spectral entropy of a time series*

Description

Computes spectral entropy from a univariate normalized spectral density, estimated using an AR model.

Usage

```
entropy(x)
```

Arguments

`x`	a univariate time series

Details

The *spectral entropy* equals the Shannon entropy of the spectral density $f_x(\lambda)$ of a stationary process x_t:

$$H_s(x_t) = -\int_{-\pi}^{\pi} f_x(\lambda) \log f_x(\lambda) d\lambda,$$

where the density is normalized such that $\int_{-\pi}^{\pi} f_x(\lambda) d\lambda = 1$. An estimate of $f(\lambda)$ can be obtained using `spec.ar` with the `burg` method.

Value

A non-negative real value for the spectral entropy $H_s(x_t)$.

Author(s)

Rob J Hyndman

References

Jerry D. Gibson and Jaewoo Jung (2006). "The Interpretation of Spectral Entropy Based Upon Rate Distortion Functions". IEEE International Symposium on Information Theory, pp. 277-281.

Goerg, G. M. (2013). "Forecastable Component Analysis". Proceedings of the 30th International Conference on Machine Learning (PMLR) 28 (2): 64-72, 2013. Available at https://proceedings.mlr.press/v28/goerg13.html.

See Also

spec.ar

Examples

```
entropy(rnorm(1000))
entropy(lynx)
entropy(sin(1:20))
```

---

firstmin_ac — *Time of first minimum in the autocorrelation function from software package* hctsa

---

Description

Time of first minimum in the autocorrelation function from software package hctsa

Usage

```
firstmin_ac(
  x,
  acfv = stats::acf(x, lag.max = N - 1, plot = FALSE, na.action = na.pass)
)
```

Arguments

x	the input time series
acfv	vector of autocorrelation, if exist, used to avoid repeated computation.

Value

The lag of the first minimum

Author(s)

Yangzhuoran Yang

References

B.D. Fulcher and N.S. Jones. hctsa: A computational framework for automated time-series phenotyping using massive feature extraction. Cell Systems 5, 527 (2017).

B.D. Fulcher, M.A. Little, N.S. Jones Highly comparative time-series analysis: the empirical structure of time series and their methods. J. Roy. Soc. Interface 10, 83 (2013).

Examples

```
firstmin_ac(WWWusage)
```

`firstzero_ac` *The first zero crossing of the autocorrelation function from software package* `hctsa`

Description

Search up to a maximum of the length of the time series

Usage

```
firstzero_ac(y, acfv = stats::acf(y, N - 1, plot = FALSE, na.action = na.pass))
```

Arguments

`y`	the input time series
`acfv`	vector of autocorrelation, if exist, used to avoid repeated computation.

Value

The first zero crossing of the autocorrelation function

Author(s)

Yangzhuoran Yang

References

B.D. Fulcher and N.S. Jones. hctsa: A computational framework for automated time-series phenotyping using massive feature extraction. Cell Systems 5, 527 (2017).

B.D. Fulcher, M.A. Little, N.S. Jones Highly comparative time-series analysis: the empirical structure of time series and their methods. J. Roy. Soc. Interface 10, 83 (2013).

flat_spots — *Longest flat spot*

Description

"Flat spots” are computed by dividing the sample space of a time series into ten equal-sized intervals, and computing the maximum run length within any single interval.

Usage

```
flat_spots(x)
```

Arguments

`x`	a univariate time series

Value

A numeric value.

Author(s)

Earo Wang and Rob J Hyndman

fluctanal_prop_r1 — *Implements fluctuation analysis from software package* `hctsa`

Description

Fits a polynomial of order 1 and then returns the range. The order of fluctuations is 2, corresponding to root mean square fluctuations.

Usage

```
fluctanal_prop_r1(x)
```

Arguments

`x`	the input time series (or any vector)

Author(s)

Yangzhuoran Yang

References

B.D. Fulcher and N.S. Jones. hctsa: A computational framework for automated time-series phenotyping using massive feature extraction. Cell Systems 5, 527 (2017).

B.D. Fulcher, M.A. Little, N.S. Jones Highly comparative time-series analysis: the empirical structure of time series and their methods. J. Roy. Soc. Interface 10, 83 (2013).

`heterogeneity` *Heterogeneity coefficients*

Description

Computes various measures of heterogeneity of a time series. First the series is pre-whitened using an AR model to give a new series y. We fit a GARCH(1,1) model to y and obtain the residuals, e. Then the four measures of heterogeneity are: (1) the sum of squares of the first 12 autocorrelations of y^2; (2) the sum of squares of the first 12 autocorrelations of e^2; (3) the R^2 value of an AR model applied to y^2; (4) the R^2 value of an AR model applied to e^2. The statistics obtained from y^2 are the ARCH effects, while those from e^2 are the GARCH effects.

Usage

```
heterogeneity(x)
```

Arguments

`x`	a univariate time series

Value

A vector of numeric values.

Author(s)

Yanfei Kang and Rob J Hyndman

`histogram_mode` *Mode of a data vector from software package* `hctsa`

Description

Measures the mode of the data vector using histograms with a given number of bins as suggestion. The value calculated is different from `hctsa` and `CompEngine` as the histogram edges are calculated differently.

Usage

```
histogram_mode(y, numBins = 10)
```

Arguments

`y`	the input data vector
`numBins`	the number of bins to use in the histogram.

Value

the mode

Author(s)

Yangzhuoran Yang

References

B.D. Fulcher and N.S. Jones. hctsa: A computational framework for automated time-series phenotyping using massive feature extraction. Cell Systems 5, 527 (2017).

B.D. Fulcher, M.A. Little, N.S. Jones Highly comparative time-series analysis: the empirical structure of time series and their methods. J. Roy. Soc. Interface 10, 83 (2013).

`holt_parameters` *Parameter estimates of Holt's linear trend method*

Description

Estimate the smoothing parameter for the level-alpha and the smoothing parameter for the trend-beta. `hw_parameters` considers additive seasonal trend: ets(A,A,A) model.

Usage

```
holt_parameters(x)

hw_parameters(x)
```

Arguments

`x`	a univariate time series

Value

`holt_parameters` produces a vector of 2 values: alpha, beta.

`hw_parameters` produces a vector of 3 values: alpha, beta and gamma.

Author(s)

Thiyanga Talagala, Pablo Montero-Manso

`hurst`	*Hurst coefficient*

Description

Computes the Hurst coefficient indicating the level of fractional differencing of a time series.

Usage

```
hurst(x)
```

Arguments

`x`	a univariate time series. If missing values are present, the largest contiguous portion of the time series is used.

Value

A numeric value.

Author(s)

Rob J Hyndman

`localsimple_taures`	*The first zero crossing of the autocorrelation function of the residuals from Simple local time-series forecasting from software package* `hctsa`

Description

Simple predictors using the past trainLength values of the time series to predict its next value.

Usage

```
localsimple_taures(y, forecastMeth = c("mean", "lfit"), trainLength = NULL)
```

Arguments

`y`	the input time series
`forecastMeth`	the forecasting method, default to `mean`. `mean`: local mean prediction using the past trainLength time-series values. `lfit`: local linear prediction using the past trainLength time-series values.
`trainLength`	the number of time-series values to use to forecast the next value. Default to 1 when using method `mean` and 3 when using method `lfit`.

Value

The first zero crossing of the autocorrelation function of the residuals

---

lumpiness *Time series features based on tiled windows*

Description

Computes feature of a time series based on tiled (non-overlapping) windows. Means or variances are produced for all tiled windows. Then stability is the variance of the means, while lumpiness is the variance of the variances.

Usage

```
lumpiness(x, width = ifelse(frequency(x) > 1, frequency(x), 10))

stability(x, width = ifelse(frequency(x) > 1, frequency(x), 10))
```

Arguments

`x`	a univariate time series
`width`	size of sliding window

Value

A numeric vector of length 2 containing a measure of lumpiness and a measure of stability.

Author(s)

Earo Wang and Rob J Hyndman

---

max_level_shift *Time series features based on sliding windows*

Description

Computes feature of a time series based on sliding (overlapping) windows. `max_level_shift` finds the largest mean shift between two consecutive windows. `max_var_shift` finds the largest var shift between two consecutive windows. `max_kl_shift` finds the largest shift in Kulback-Leibler divergence between two consecutive windows.

Usage

```
max_level_shift(x, width = ifelse(frequency(x) > 1, frequency(x), 10))

max_var_shift(x, width = ifelse(frequency(x) > 1, frequency(x), 10))

max_kl_shift(x, width = ifelse(frequency(x) > 1, frequency(x), 10))
```

Arguments

`x`	a univariate time series
`width`	size of sliding window

Details

Computes the largest level shift and largest variance shift in sliding mean calculations

Value

A vector of 2 values: the size of the shift, and the time index of the shift.

Author(s)

Earo Wang and Rob J Hyndman

`motiftwo_entro3` *Local motifs in a binary symbolization of the time series from software package* `hctsa`

Description

Coarse-graining is performed. Time-series values above its mean are given 1, and those below the mean are 0.

Usage

```
motiftwo_entro3(y)
```

Arguments

`y`	the input time series

Value

Entropy of words in the binary alphabet of length 3.

Author(s)

Yangzhuoran Yang

References

B.D. Fulcher and N.S. Jones. hctsa: A computational framework for automated time-series phenotyping using massive feature extraction. Cell Systems 5, 527 (2017).

B.D. Fulcher, M.A. Little, N.S. Jones Highly comparative time-series analysis: the empirical structure of time series and their methods. J. Roy. Soc. Interface 10, 83 (2013).

Examples

```
motiftwo_entro3(WWWusage)
```

`nonlinearity` *Nonlinearity coefficient*

Description

Computes a nonlinearity statistic based on Lee, White & Granger's nonlinearity test of a time series. The statistic is $10X^2/T$ where X^2 is the Chi-squared statistic from Lee, White and Granger, and T is the length of the time series. This takes large values when the series is nonlinear, and values around 0 when the series is linear.

Usage

```
nonlinearity(x)
```

Arguments

`x`	a univariate time series

Value

A numeric value.

Author(s)

Yanfei Kang and Rob J Hyndman

References

Lee, T. H., White, H., & Granger, C. W. (1993). Testing for neglected nonlinearity in time series models: A comparison of neural network methods and alternative tests. *Journal of Econometrics*, 56(3), 269-290.

Teräsvirta, T., Lin, C.-F., & Granger, C. W. J. (1993). Power of the neural network linearity test. *Journal of Time Series Analysis*, 14(2), 209–220.

Examples

```
nonlinearity(lynx)
```

outlierinclude_mdrmd *How median depend on distributional outliers from software package* hctsa

Description

Measures median as more and more outliers are included in the calculation according to a specified rule, of outliers being furthest from the mean.

Usage

```
outlierinclude_mdrmd(y, zscored = TRUE)
```

Arguments

y	the input time series (ideally z-scored)
zscored	Should y be z-scored before computing the statistic. Default: TRUE

Details

The threshold for including time-series data points in the analysis increases from zero to the maximum deviation, in increments of 0.01*sigma (by default), where sigma is the standard deviation of the time series.

At each threshold, proportion of time series points included and median are calculated, and outputs from the algorithm measure how these statistical quantities change as more extreme points are included in the calculation.

Outliers are defined as furthest from the mean.

Value

median of the median of range indices

Author(s)

Yangzhuoran Yang

References

B.D. Fulcher and N.S. Jones. hctsa: A computational framework for automated time-series phenotyping using massive feature extraction. Cell Systems 5, 527 (2017).

B.D. Fulcher, M.A. Little, N.S. Jones Highly comparative time-series analysis: the empirical structure of time series and their methods. J. Roy. Soc. Interface 10, 83 (2013).

`pacf_features` *Partial autocorrelation-based features*

Description

Computes various measures based on partial autocorrelation coefficients of the original series, first-differenced series and second-differenced series

Usage

```
pacf_features(x)
```

Arguments

`x`	a univariate time series

Value

A vector of 3 values: Sum of squared of first 5 partial autocorrelation coefficients of the original series, first differenced series and twice-differenced series. For seasonal data, the partial autocorrelation coefficient at the first seasonal lag is also returned.

Author(s)

Thiyanga Talagala

`pred_features` *The prediction feature set from software package* `hctsa`

Description

Calculate the features that grouped as prediction set, which have been used in CompEngine database, using method introduced in package `hctsa`.

Usage

```
pred_features(x)
```

Arguments

`x`	the input time series

Details

Features in this set are `localsimple_mean1`, `localsimple_lfitac`, and `sampen_first`.

Value

a vector with prediction features

Author(s)

Yangzhuoran Yang

References

B.D. Fulcher and N.S. Jones. hctsa: A computational framework for automated time-series phenotyping using massive feature extraction. Cell Systems 5, 527 (2017).

B.D. Fulcher, M.A. Little, N.S. Jones Highly comparative time-series analysis: the empirical structure of time series and their methods. J. Roy. Soc. Interface 10, 83 (2013).

See Also

`localsimple_taures`

`sampen_first`

`sampenc` — *Second Sample Entropy from software package* `hctsa`

Description

Modified from the Ben Fulcher version of original code sampenc.m from http://physionet.org/physiotools/sampen/ http://www.physionet.org/physiotools/sampen/matlab/1.1/sampenc.m Code by DK Lake (dlake@virginia.edu), JR Moorman and Cao Hanqing.

Usage

```
sampenc(y, M = 6, r = 0.3)
```

Arguments

`y`	the input time series
`M`	embedding dimension
`r`	threshold

Author(s)

Yangzhuoran Yang

References

cf. "Physiological time-series analysis using approximate entropy and sample entropy", J. S. Richman and J. R. Moorman, Am. J. Physiol. Heart Circ. Physiol., 278(6) H2039 (2000)

B.D. Fulcher and N.S. Jones. hctsa: A computational framework for automated time-series phenotyping using massive feature extraction. Cell Systems 5, 527 (2017).

B.D. Fulcher, M.A. Little, N.S. Jones Highly comparative time-series analysis: the empirical structure of time series and their methods. J. Roy. Soc. Interface 10, 83 (2013).

`sampen_first` *Second Sample Entropy of a time series from software package* `hctsa`

Description

Modified from the Ben Fulcher's `EN_SampEn` which uses code from PhysioNet. The publicly-available PhysioNet Matlab code, sampenc (renamed here to RN_sampenc) is available from: http://www.physionet.org/physiotools/sampen/matlab/1.1/sampenc.m

Usage

```
sampen_first(y)
```

Arguments

`y`	the input time series

Details

Embedding dimension is set to 5. The threshold is set to 0.3.

Author(s)

Yangzhuoran Yang

References

cf. "Physiological time-series analysis using approximate entropy and sample entropy", J. S. Richman and J. R. Moorman, Am. J. Physiol. Heart Circ. Physiol., 278(6) H2039 (2000)

B.D. Fulcher and N.S. Jones. hctsa: A computational framework for automated time-series phenotyping using massive feature extraction. Cell Systems 5, 527 (2017).

B.D. Fulcher, M.A. Little, N.S. Jones Highly comparative time-series analysis: the empirical structure of time series and their methods. J. Roy. Soc. Interface 10, 83 (2013).

scal_features — *The scaling feature set from software package* hctsa

Description

Calculate the features that grouped as scaling set, which have been used in CompEngine database, using method introduced in package hctsa.

Usage

```
scal_features(x)
```

Arguments

x	the input time series

Details

Feature in this set is fluctanal_prop_r1.

Value

a vector with scaling features

Author(s)

Yangzhuoran Yang

References

B.D. Fulcher and N.S. Jones. hctsa: A computational framework for automated time-series phenotyping using massive feature extraction. Cell Systems 5, 527 (2017).

B.D. Fulcher, M.A. Little, N.S. Jones Highly comparative time-series analysis: the empirical structure of time series and their methods. J. Roy. Soc. Interface 10, 83 (2013).

See Also

fluctanal_prop_r1

spreadrandomlocal_meantaul — *Bootstrap-based stationarity measure from software package* hctsa

Description

100 time-series segments of length `l` are selected at random from the time series and the mean of the first zero-crossings of the autocorrelation function in each segment is calculated.

Usage

```
spreadrandomlocal_meantaul(y, l = 50)
```

Arguments

`y`	the input time series
`l`	the length of local time-series segments to analyse as a positive integer. Can also be a specified character string: "ac2": twice the first zero-crossing of the autocorrelation function

Value

mean of the first zero-crossings of the autocorrelation function

Author(s)

Yangzhuoran Yang

References

B.D. Fulcher and N.S. Jones. hctsa: A computational framework for automated time-series phenotyping using massive feature extraction. Cell Systems 5, 527 (2017).

B.D. Fulcher, M.A. Little, N.S. Jones Highly comparative time-series analysis: the empirical structure of time series and their methods. J. Roy. Soc. Interface 10, 83 (2013).

station_features — *The stationarity feature set from software package* hctsa

Description

Calculate the features that grouped as stationarity set, which have been used in CompEngine database, using method introduced in package `hctsa`.

Usage

```
station_features(x)
```

Arguments

x the input time series

Details

Features in this set are `std1st_der`, `spreadrandomlocal_meantaul_50`, and `spreadrandomlocal_meantaul_ac2`.

Value

a vector with stationarity features

Author(s)

Yangzhuoran Yang

References

B.D. Fulcher and N.S. Jones. hctsa: A computational framework for automated time-series phenotyping using massive feature extraction. Cell Systems 5, 527 (2017).

B.D. Fulcher, M.A. Little, N.S. Jones Highly comparative time-series analysis: the empirical structure of time series and their methods. J. Roy. Soc. Interface 10, 83 (2013).

See Also

`std1st_der`

`spreadrandomlocal_meantaul`

---

std1st_der *Standard deviation of the first derivative of the time series from software package* `hctsa`

Description

Modified from `SY_StdNthDer` in `hctsa`. Based on an idea by Vladimir Vassilevsky.

Usage

```
std1st_der(y)
```

Arguments

y the input time series. Missing values will be removed.

Value

Standard deviation of the first derivative of the time series.

Author(s)

Yangzhuoran Yang

References

cf. http://www.mathworks.de/matlabcentral/newsreader/view_thread/136539

B.D. Fulcher and N.S. Jones. hctsa: A computational framework for automated time-series phenotyping using massive feature extraction. Cell Systems 5, 527 (2017).

B.D. Fulcher, M.A. Little, N.S. Jones Highly comparative time-series analysis: the empirical structure of time series and their methods. J. Roy. Soc. Interface 10, 83 (2013).

`stl_features` *Strength of trend and seasonality of a time series*

Description

Computes various measures of trend and seasonality of a time series based on an STL decomposition. The number of seasonal periods, and the length of the seasonal periods are returned. Also, the strength of seasonality corresponding to each period is estimated. The `mstl` function is used to do the decomposition.

Usage

```
stl_features(x, ...)
```

Arguments

`x`	a univariate time series.
`...`	Other arguments are passed to `mstl`.

Value

A vector of numeric values.

Author(s)

Rob J Hyndman

trev_num — *Normalized nonlinear autocorrelation, the numerator of the trev function of a time series from software package* hctsa

Description

Calculates the numerator of the trev function, a normalized nonlinear autocorrelation, The time lag is set to 1.

Usage

```
trev_num(y)
```

Arguments

y	the input time series

Value

the numerator of the trev function of a time series

Author(s)

Yangzhuoran Yang

References

B.D. Fulcher and N.S. Jones. hctsa: A computational framework for automated time-series phenotyping using massive feature extraction. Cell Systems 5, 527 (2017).

B.D. Fulcher, M.A. Little, N.S. Jones Highly comparative time-series analysis: the empirical structure of time series and their methods. J. Roy. Soc. Interface 10, 83 (2013).

Examples

```
trev_num(WWWusage)
```

tsfeatures *Time series feature matrix*

Description

`tsfeatures` computes a matrix of time series features from a list of time series

Usage

```
tsfeatures(
  tslist,
  features = c("frequency", "stl_features", "entropy", "acf_features"),
  scale = TRUE,
  trim = FALSE,
  trim_amount = 0.1,
  parallel = FALSE,
  multiprocess = future::multisession,
  na.action = na.pass,
  ...
)
```

Arguments

`tslist`	a list of univariate time series, each of class `ts` or a numeric vector. Alternatively, an object of class `mts` may be used.
`features`	a vector of function names which return numeric vectors of features. All features returned by these functions must be named if they return more than one feature. Existing functions from installed packages may be used, but the package must be loaded first. Functions must return a result for all time series, even if it is just NA.
`scale`	if `TRUE`, time series are scaled to mean 0 and sd 1 before features are computed.
`trim`	if `TRUE`, time series are trimmed by `trim_amount` before features are computed. Values larger than `trim_amount` in absolute value are set to `NA`.
`trim_amount`	Default level of trimming if `trim==TRUE`.
`parallel`	If TRUE, multiple cores (or multiple sessions) will be used. This only speeds things up when there are a large number of time series.
`multiprocess`	The function from the `future` package to use for parallel processing. Either `multisession` or `multicore`. The latter is preferred for Linux and MacOS.
`na.action`	A function to handle missing values. Use `na.interp` to estimate missing values.
`...`	Other arguments get passed to the feature functions.

Value

A feature matrix (in the form of a tibble) with each row corresponding to one time series from tslist, and each column being a feature.

Author(s)

Rob J Hyndman

Examples

```
mylist <- list(sunspot.year, WWWusage, AirPassengers, USAccDeaths)
tsfeatures(mylist)
```

---

unitroot_kpss *Unit Root Test Statistics*

Description

`unitroot_kpss` computes the statistic for the Kwiatkowski et al. unit root test using the default settings for the `ur.kpss` function. `unitroot_pp` computes the statistic for the Phillips-Perron unit root test using the default settings for the `ur.pp` function.

Usage

```
unitroot_kpss(x, ...)

unitroot_pp(x, ...)
```

Arguments

`x`	a univariate time series.
`...`	Other arguments are passed to the `ur.kpss` or `ur.kpss` functions.

Value

A numeric value

Author(s)

Pablo Montero-Manso

`walker_propcross` *Simulates a hypothetical walker moving through the time domain from software package* `hctsa`

Description

The hypothetical particle (or 'walker') moves in response to values of the time series at each point. The walker narrows the gap between its value and that of the time series by 10%.

Usage

```
walker_propcross(y)
```

Arguments

`y`	the input time series

Value

fraction of time series length that walker crosses time series

Author(s)

Yangzhuoran Yang

References

B.D. Fulcher and N.S. Jones. hctsa: A computational framework for automated time-series phenotyping using massive feature extraction. Cell Systems 5, 527 (2017).

B.D. Fulcher, M.A. Little, N.S. Jones Highly comparative time-series analysis: the empirical structure of time series and their methods. J. Roy. Soc. Interface 10, 83 (2013).

`yahoo_data` *Yahoo server metrics*

Description

Yahoo server metrics

Usage

```
yahoo_data(...)
```

Arguments

...	Additional arguments passed to `download.file` Downloads and returns aggregated and anonymized datasets from Yahoo representing server metrics of Yahoo services.

Value

A matrix of time series with 1437 rows of hourly data, and 1748 columns representing different servers.

Author(s)

Rob Hyndman, Earo Wang, Nikolay Laptev, Mitchell O'Hara-Wild

References

Hyndman, R.J., Wang, E., Laptev, N. (2015) Large-scale unusual time series detection. In: *Proceedings of the IEEE International Conference on Data Mining*. Atlantic City, NJ, USA. 14–17 November 2015. https://robjhyndman.com/publications/icdm2015/

Examples

```
yahoo <- yahoo_data()
plot(yahoo[,1:10])
plot(yahoo[,1:44], plot.type='single', col=1:44)
```

`zero_proportion` *Proportion of zeros*

Description

Computes proportion of zeros in a time series

Usage

```
zero_proportion(x, tol = 1e-08)
```

Arguments

x	a univariate time series
tol	tolerance level. Absolute values below this are considered zeros.

Value

A numeric value.

Author(s)

Thiyanga Talagala

Index