

Package ‘sazedR’

September 16, 2019

Type Package

Title Parameter-Free Domain-Agnostic Season Length Detection in Time Series

Version 2.0.1

Description Spectral and Average Autocorrelation Zero Distance Density ('sazed') is a method for estimating the season length of a seasonal time series. 'sazed' is aimed at practitioners, as it employs only domain-agnostic preprocessing and does not depend on parameter tuning or empirical constants. The computation of 'sazed' relies on the efficient autocorrelation computation methods suggested by Thibault Nion (2012, URL: <http://www.tibonihoo.net/literate_musing/autocorrelations.html>) and by Bob Carpenter (2012, URL: <<https://lingpipe-blog.com/2012/06/08/autocorrelation-fft-kiss-eigen/>>).

License GPL-2

URL https://github.com/mtoller/autocorr_season_length_detection/

Encoding UTF-8

LazyData true

Imports bspec (>= 1.5), dplyr (>= 0.8.0.1), fftwtools (>= 0.9.8),
pracma (>= 2.1.4), zoo (>= 1.8-3)

RoxygenNote 6.1.1

NeedsCompilation no

Author Maximilian Toller [aut],
Tiago Santos [aut, cre],
Roman Kern [aut]

Maintainer Tiago Santos <teixeiradossantos@tugraz.at>

Repository CRAN

Date/Publication 2019-09-16 21:00:02 UTC

R topics documented:

aze 2

| | |
|------------------------|---|
| azed | 3 |
| computeAcf | 3 |
| downsample | 4 |
| preprocessTs | 4 |
| S | 5 |
| Sa | 5 |
| sazed | 6 |
| sazed.maj | 6 |
| sazedR | 7 |
| ze | 7 |
| zed | 8 |

| | |
|--------------|----------|
| Index | 9 |
|--------------|----------|

| | |
|-----|--|
| aze | <i>Compute the AZE component of the SAZED ensemble</i> |
|-----|--|

Description

aze estimates the season length of its argument from the mean autocorrelation zero distance

Usage

```
aze(y, preprocess = T)
```

Arguments

| | |
|------------|--|
| y | The input time series. |
| preprocess | If true, y is detrended and z-normalized before computation. |

Value

The AZE season length estimate of y.

Examples

```
season_length <- 26
y <- sin(1:400*2*pi/season_length)
aze(y)
aze(y, preprocess = FALSE)
```

| | |
|------|---|
| azed | <i>Compute the AZED component of the SAZED ensemble</i> |
|------|---|

Description

azed computes the autocorrelation of its argument, and then derives the season length from its the autocorrelations zero density.

Usage

```
azed(y, preprocess = T)
```

Arguments

| | |
|------------|--|
| y | The input time series. |
| preprocess | If true, y is detrended and z-normalized before computation. |

Value

The AZED season length estimate of y.

Examples

```
season_length <- 26  
y <- sin(1:400*2*pi/season_length)  
azed(y)  
azed(y, preprocess = FALSE)
```

| | |
|------------|--|
| computeAcf | <i>Compute and shorten autocorrelation</i> |
|------------|--|

Description

computeAcf computes the autocorrelation function of its argument and discards the zero lag and all lags greater than 2/3 of the argument's length

Usage

```
computeAcf(y)
```

Arguments

| | |
|---|------------------------|
| y | The input time series. |
|---|------------------------|

Value

The shortened autocorrelation

Examples

```
season_length <- 26
y <- sin(1:400*2*pi/season_length)
computeAcf(y)
```

| | |
|------------|-------------------------------|
| downsample | <i>Downsample Time Series</i> |
|------------|-------------------------------|

Description

downsample samples down a time series with a rolling mean.

Usage

```
downsample(data, window_size = 2)
```

Arguments

| | |
|-------------|---|
| data | The input time series. |
| window_size | The size of the rolling mean window used. |

Value

The downsampled time series.

| | |
|--------------|--|
| preprocessTs | <i>Preprocess Time Series for SAZED ensemble</i> |
|--------------|--|

Description

preprocessTs detrends and z-normalizes its argument.

Usage

```
preprocessTs(y)
```

Arguments

| | |
|---|------------------------|
| y | The input time series. |
|---|------------------------|

Value

The detrended and z-normalized time series.

Examples

```
season_length <- 26
y <- sin(1:400*2*pi/season_length)
preprocessTs(y)
```

S *Compute the S component of the SAZED ensemble*

Description

S computes the spectral density of its argument, and then derives the season length from it.

Usage

```
S(y, preprocess = T)
```

Arguments

y The input time series.
preprocess If true, y is detrended and z-normalized before computation.

Value

The S season length estimate of y.

Examples

```
season_length <- 26  
y <- sin(1:400*2*pi/season_length)  
S(y)  
S(y, preprocess = FALSE)
```

Sa *Compute the SA component of the SAZED ensemble*

Description

Sa computes the autocorrelation of its argument, and then derives the season length from its spectral density.

Usage

```
Sa(y, preprocess = T)
```

Arguments

y The input time series.
preprocess If true, y is detrended and z-normalized before computation.

Value

The SA season length estimate of y.

Examples

```

season_length <- 26
y <- sin(1:400*2*pi/season_length)
Sa(y)
Sa(y, preprocess = FALSE)

```

| | |
|-------|---------------------------------|
| sazed | <i>SAZED Ensemble (Optimum)</i> |
|-------|---------------------------------|

Description

sazed estimates a time series' season length by combining 3 different estimates computed on an input time series and its 10-fold self-composed autocorrelation.

Usage

```
sazed(y)
```

Arguments

y The input time series.

Value

The season length of the input time series.

Examples

```

season_length <- 26
y <- sin(1:400*2*pi/season_length)
sazed(y)

```

| | |
|-----------|----------------------------------|
| sazed.maj | <i>SAZED Ensemble (Majority)</i> |
|-----------|----------------------------------|

Description

sazed.maj estimates a time series' season length by computing 6 different estimates and taking a majority vote.

Usage

```
sazed.maj(y, iter = 0, method = "down", preprocess = T)
```

Arguments

| | |
|-------------------------|---|
| <code>y</code> | The input time series. |
| <code>iter</code> | The recursion depth. |
| <code>method</code> | The method used for breaking ties. One of <code>c("alt", "diff", "down")</code> . |
| <code>preprocess</code> | If true, <code>y</code> is detrended and z-normalized before computation. |

Value

The season length of the input time series.

Examples

```
season_length <- 26
y <- sin(1:400*2*pi/season_length)
sazed.maj(y)
```

| | |
|---------------------|--|
| <code>sazedR</code> | <i>sazedR: A package for for estimating the season length of a seasonal time series.</i> |
|---------------------|--|

Description

The `sazedR` package provides the main function to compute season length, `sazed`, which is an ensemble of many season length estimation methods, also included in this package.

| | |
|-----------------|---|
| <code>ze</code> | <i>Compute the ZE component of the SAZED ensemble</i> |
|-----------------|---|

Description

`ze` estimates the season length of its argument from the mean zero distance

Usage

```
ze(y, preprocess = T)
```

Arguments

| | |
|-------------------------|---|
| <code>y</code> | The input time series. |
| <code>preprocess</code> | If true, <code>y</code> is detrended and z-normalized before computation. |

Value

The ZE season length estimate of `y`.

Examples

```
season_length <- 26
y <- sin(1:400*2*pi/season_length)
ze(y)
zed(y, preprocess = FALSE)
```

zed*Compute the ZED component of the SAZED ensemble*

Description

zed computes the zero density of its argument, and then derives the season length from it.

Usage

```
zed(y, preprocess = T)
```

Arguments

| | |
|------------|--|
| y | The input time series. |
| preprocess | If true, y is detrended and z-normalized before computation. |

Value

The ZED season length estimate of y.

Examples

```
season_length <- 26
y <- sin(1:400*2*pi/season_length)
zed(y)
zed(y, preprocess = FALSE)
```


Index

aze, [2](#)
azed, [3](#)

computeAcf, [3](#)

downsample, [4](#)

preprocessTs, [4](#)

S, [5](#)

Sa, [5](#)

sazed, [6](#)

sazed.maj, [6](#)

sazedR, [7](#)

sazedR-package (sazedR), [7](#)

ze, [7](#)

zed, [8](#)