Package ‘runstats’

May 9, 2026

Type Package

Title Fast Computation of Running Statistics for Time Series

Version 1.1.0

Description Provides methods for fast computation of running sample statistics for time series. These include: (1) mean, (2) standard deviation, and (3) variance over a fixed-length window of time-series, (4) correlation, (5) covariance, and (6) Euclidean distance (L2 norm) between short-time pattern and time-series. Implemented methods utilize Convolution Theorem to compute convolutions via Fast Fourier Transform (FFT).

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

URL https://github.com/martakarass/runstats

BugReports https://github.com/martakarass/runstats/issues

Imports fftwtools

Suggests covr, testthat, ggplot2, knitr, rmarkdown, sessioninfo, rbenchmark, cowplot, spelling

VignetteBuilder knitr

Language en-US

NeedsCompilation no

Author Marta Karas [aut, cre] (ORCID: <https://orcid.org/0000-0001-5889-3970>), Jacek Urbanek [aut] (ORCID: <https://orcid.org/0000-0002-1890-8899>), John Muschelli [ctb] (ORCID: <https://orcid.org/0000-0001-6469-1750>), Lacey Etzkorn [ctb]

Maintainer Marta Karas <marta.karass@gmail.com>

Repository CRAN

Date/Publication 2019-11-14 20:30:02 UTC

Contents

`RunningCor` *Fast Running Correlation Computation*

Description

Computes running correlation between time-series `x` and short-time pattern `y`.

Usage

```
RunningCor(x, y, circular = FALSE)
```

Arguments

`x`	A numeric vector.
`y`	A numeric vector, of equal or shorter length than `x`.
`circular`	logical; whether running correlation is computed assuming circular nature of `x` time-series (see Details).

Details

Computes running correlation between time-series `x` and short-time pattern `y`. The length of output vector equals the length of `x`. Parameter `circular` determines whether `x` time-series is assumed to have a circular nature. Assume l_x is the length of time-series `x`, l_y is the length of short-time pattern `y`.

If `circular` equals `TRUE` then

- first element of the output vector corresponds to sample correlation between `x[1:l_y]` and `y`,
- last element of the output vector corresponds to sample correlation between `c(x[l_x], x[1:(l_y - 1)])` and `y`.

If `circular` equals `FALSE` then

- first element of the output vector corresponds to sample correlation between `x[1:l_y]` and `y`,
- the $l_x - W + 1$-th element of the output vector corresponds to sample correlation between `x[(l_x - l_y + 1):l_x]`,
- last `W-1` elements of the output vector are filled with `NA`.

See `runstats.demo(func.name = "RunningCor")` for a detailed presentation.

Value

A numeric vector.

Examples

```
x <- sin(seq(0, 1, length.out = 1000) * 2 * pi * 6)
y <- x[1:100]
out1 <- RunningCor(x, y, circular = TRUE)
out2 <- RunningCor(x, y, circular = FALSE)
plot(out1, type = "l"); points(out2, col = "red")
```

RunningCov *Fast Running Covariance Computation*

Description

Computes running covariance between time-series x and short-time pattern y.

Usage

```
RunningCov(x, y, circular = FALSE)
```

Arguments

x	A numeric vector.
y	A numeric vector, of equal or shorter length than x.
circular	Logical; whether running variance is computed assuming circular nature of x time-series (see Details).

Details

Computes running covariance between time-series x and short-time pattern y.

The length of output vector equals the length of x. Parameter circular determines whether x time-series is assumed to have a circular nature. Assume l_x is the length of time-series x, l_y is the length of short-time pattern y.

If circular equals TRUE then

- first element of the output vector corresponds to sample covariance between x[1:l_y] and y,
- last element of the output vector corresponds to sample covariance between c(x[l_x], x[1:(l_y - 1)]) and y.

If circular equals FALSE then

- first element of the output vector corresponds to sample covariance between x[1:l_y] and y,
- the $l_x - W + 1$-th last element of the output vector corresponds to sample covariance between x[(l_x - l_y + 1):l_x],
- last W-1 elements of the output vector are filled with NA.

See runstats.demo(func.name = "RunningCov") for a detailed presentation.

Value

A numeric vector.

Examples

```
x <- sin(seq(0, 1, length.out = 1000) * 2 * pi * 6)
y <- x[1:100]
out1 <- RunningCov(x, y, circular = TRUE)
out2 <- RunningCov(x, y, circular = FALSE)
plot(out1, type = "l"); points(out2, col = "red")
```

RunningL2Norm *Fast Running L2 Norm Computation*

Description

Computes running L2 norm between between time-series `x` and short-time pattern `y`.

Usage

```
RunningL2Norm(x, y, circular = FALSE)
```

Arguments

`x`	A numeric vector.
`y`	A numeric vector, of equal or shorter length than `x`.
`circular`	logical; whether running L2 norm is computed assuming circular nature of `x` time-series (see Details).

Details

Computes running L2 norm between between time-series `x` and short-time pattern `y`. The length of output vector equals the length of `x`. Parameter `circular` determines whether `x` time-series is assumed to have a circular nature. Assume l_x is the length of time-series `x`, l_y is the length of short-time pattern `y`.

If `circular` equals `TRUE` then

- first element of the output vector corresponds to sample L2 norm between `x[1:l_y]` and `y`,
- last element of the output vector corresponds to sample L2 norm between `c(x[l_x], x[1:(l_y - 1)])` and `y`.

If `circular` equals `FALSE` then

- first element of the output vector corresponds to sample L2 norm between `x[1:l_y]` and `y`,
- the $l_x - W + 1$-th element of the output vector corresponds to sample L2 norm between `x[(l_x - l_y + 1):l_x]`,
- last `W-1` elements of the output vector are filled with `NA`.

See `runstats.demo(func.name = "RunningL2Norm")` for a detailed presentation.

Value

A numeric vector.

Examples

```
## Ex.1.
x <- sin(seq(0, 1, length.out = 1000) * 2 * pi * 6)
y1 <- x[1:100] + rnorm(100)
y2 <- rnorm(100)
out1 <- RunningL2Norm(x, y1)
out2 <- RunningL2Norm(x, y2)
plot(out1, type = "l"); points(out2, col = "blue")
## Ex.2.
x <- sin(seq(0, 1, length.out = 1000) * 2 * pi * 6)
y <- x[1:100] + rnorm(100)
out1 <- RunningL2Norm(x, y, circular = TRUE)
out2 <- RunningL2Norm(x, y, circular = FALSE)
plot(out1, type = "l"); points(out2, col = "red")
```

`RunningMean` *Fast Running Mean Computation*

Description

Computes running sample mean of a time-series `x` in a fixed length window.

Usage

```
RunningMean(x, W, circular = FALSE)
```

Arguments

`x`	A numeric vector.
`W`	A numeric scalar; length of `x` window over which sample mean is computed.
`circular`	Logical; whether running sample mean is computed assuming circular nature of `x` time-series (see Details).

Details

The length of output vector equals the length of `x` vector. Parameter `circular` determines whether `x` time-series is assumed to have a circular nature. Assume l_x is the length of time-series `x`, `W` is a fixed length of `x` time-series window.

If `circular` equals `TRUE` then

- first element of the output time-series corresponds to sample mean of `x[1:W]`,
- last element of the output time-series corresponds to sample mean of `c(x[l_x], x[1:(W - 1)])`.

If `circular` equals `FALSE` then

- first element of the output time-series corresponds to sample mean of `x[1:W]`,
- $l_x - W + 1$-th element of the output time-series corresponds to sample mean of `x[(l_x - W + 1):l_x]`,
- last `W-1` elements of the output time-series are filled with `NA`.

See `runstats.demo(func.name = "RunningMean")` for a detailed presentation.

Value

A numeric vector.

Examples

```
x <- rnorm(10)
RunningMean(x, 3, circular = FALSE)
RunningMean(x, 3, circular = TRUE)
```

`RunningSd` *Fast Running Standard Deviation Computation*

Description

Computes running sample standard deviation of a time-series `x` in a fixed length window.

Usage

```
RunningSd(x, W, circular = FALSE)
```

Arguments

`x`	A numeric vector.
`W`	A numeric scalar; length of `x` window over which sample variance is computed.
`circular`	Logical; whether running sample standard deviation is computed assuming circular nature of `x` time-series (see Details).

Details

The length of output vector equals the length of `x` vector. Parameter `circular` determines whether `x` time-series is assumed to have a circular nature. Assume l_x is the length of time-series `x`, `W` is a fixed length of `x` time-series window.

If `circular` equals `TRUE` then

- first element of the output time-series corresponds to sample standard deviation of `x[1:W]`,
- last element of the output time-series corresponds to sample standard deviation of `c(x[l_x], x[1:(W - 1)])`.

If `circular` equals `FALSE` then

- first element of the output time-series corresponds to sample standard deviation of `x[1:W]`,
- the $l_x - W + 1$-th element of the output time-series corresponds to sample standard deviation of `x[(l_x - W + 1):l_x]`,
- last `W-1` elements of the output time-series are filled with `NA`.

See `runstats.demo(func.name = "RunningSd")` for a detailed presentation.

Value

A numeric vector.

Examples

```
x <- rnorm(10)
RunningSd(x, 3, circular = FALSE)
RunningSd(x, 3, circular = FALSE)
```

`RunningVar` *Fast Running Variance Computation*

Description

Computes running sample variance of a time-series `x` in a fixed length window.

Usage

```
RunningVar(x, W, circular = FALSE)
```

Arguments

`x`	A numeric vector.
`W`	A numeric scalar; length of `x` window over which sample variance is computed.
`circular`	Logical; whether running sample variance is computed assuming circular nature of `x` time-series (see Details).

Details

The length of output vector equals the length of `x` vector. Parameter `circular` determines whether `x` time-series is assumed to have a circular nature. Assume l_x is the length of time-series `x`, `W` is a fixed length of `x` time-series window.

If `circular` equals `TRUE` then

- first element of the output time-series corresponds to sample variance of `x[1:W]`,
- last element of the output time-series corresponds to sample variance of `c(x[l_x], x[1:(W - 1)])`.

If `circular` equals `FALSE` then

- first element of the output time-series corresponds to sample variance of `x[1:W]`,
- the $l_x - W + 1$-th element of the output time-series corresponds to sample variance of `x[(l_x - W + 1):l_x]`,
- last `W-1` elements of the output time-series are filled with `NA`.

See `runstats.demo(func.name = "RunningVar")` for a detailed presentation.

Value

A numeric vector.

Examples

```
x <- rnorm(10)
RunningVar(x, W = 3, circular = FALSE)
RunningVar(x, W = 3, circular = TRUE)
```

`runstats.demo` *Demo visualization of package functions*

Description

Generates demo visualization of output of methods for computing running statistics.

Usage

```
runstats.demo(func.name = "RunningCov")
```

Arguments

`func.name` Character value; one of the following:

- "RunningMean",
- "RunningSd",
- "RunningVar",
- "RunningCov",
- "RunningCor",
- "RunningL2Norm".

Value

`NULL`

Examples

```
## Not run:
runstats.demo(func.name = "RunningMean")
runstats.demo(func.name = "RunningSd")
runstats.demo(func.name = "RunningVar")
runstats.demo(func.name = "RunningCov")
runstats.demo(func.name = "RunningCor")
runstats.demo(func.name = "RunningL2Norm")

## End(Not run)
```

Index