

# Package: RcppUTS (via r-universe)

July 7, 2024

**Type** Package

**Title** Rcpp Bindings for Algorithms for Unevenly Spaced Time Series

**Version** 0.0.1

**Date** 2018-06-04

**Author** Dirk Eddelbuettel

**Maintainer** Dirk Eddelbuettel <edd@debian.org>

**Description** Algorithms and operators for unevenly-spaced time series  
are provided based on the 'UTS' library by Andreas Eckner.

**License** GPL (>= 2)

**Imports** Rcpp (>= 0.12.17)

**LinkingTo** Rcpp

**Suggests** xts

**RoxygenNote** 6.0.1

**Repository** <https://eddelbuettel.r-universe.dev>

**RemoteUrl** <https://github.com/eddelbuettel/rcpputs>

**RemoteRef** HEAD

**RemoteSha** 82485ad74f00b635ac10c6486d7d0e574b0d393a

## Contents

RcppUTS-package	2
EMAnext	2
rollingCentralMoment	3
SMAnext	4
utsExample	6

<b>Index</b>	<b>7</b>
--------------	----------

---

RcppUTS-package

*Rcpp Bindings for Algorithms for Unevenly Spaced Time Series*

---

### **Description**

Algorithms and operators for unevenly-spaced time series are provided based on the 'UTS' library by Andreas Eckner.

### **Details**

The DESCRIPTION file: This package was not yet installed at build time.

Index: This package was not yet installed at build time.

This section should provide a more detailed overview of how to use the package, including the most important functions.

### **Author(s)**

Dirk Eddelbuettel

Maintainer: Dirk Eddelbuettel <edd@debian.org>

### **References**

This optional section can contain literature or other references for background information.

### **See Also**

Optional links to other man pages

### **Examples**

```
## Optional simple examples of the most important functions
## Use \dontrun{} around code to be shown but not executed
```

---

EMAnext

*EMA functions for unevenly spaced time series*

---

### **Description**

The UTS library by Andreas Eckner provides algorithms for unevenly spaced time-series data. This package brings a few of them to R. The functions describe here offer exponentially-decaying weighted moving average, or EMA, for short. Three variants are provided considering the last or next observation relative to time 't', as well as linear interpolation between them.

**Usage**

```
EMANext(times, values, tau)
```

```
EMALast(times, values, tau)
```

```
EMALinear(times, values, tau)
```

**Arguments**

times	A Datetime vector
values	A numeric vector
tau	A double with the decay factor

**Value**

A numeric vector with EMA-weighted values. package at the given position is available.

**Author(s)**

Dirk Eddebuettel for the package, Andreas Eckner for the underlying code.

**Examples**

```
if (requireNamespace("xts", quietly=TRUE)) {
  suppressMessages(library(xts))
  times <- ISOdatetime(2010, 1, 2, 8, 30, 0) + c(0, 1.0, 1.2, 2.3, 2.9, 5.0)
  values <- seq(0, 10, by=2)
  plot(xts(values, order.by=times), type="b",
       main="Series and last/next/linear EMAs",
       major.ticks="auto", grid.ticks.on="auto")
  lines(xts(EMALast(times,values, 1), values, order.by=times),
        type="b", col="lightblue")
  lines(xts(EMANext(times,values, 1), values, order.by=times),
        type="b", col="darkblue")
  lines(xts(EMALinear(times,values, 1), values, order.by=times),
        type="b", col="mediumblue")
  addLegend("topleft", legend.names=c("series", "EMALast", "EMANext", "EMALinear"),
           lty=rep(1,4), lwd=rep(1,4),
           col=c("black", "lightblue", "darkblue", "mediumblue"))
}
```

---

rollingCentralMoment *Rolling operations functions for irregularly spaced time series*

---

**Description**

The UTS library by Andreas Eckner provides algorithms for unevenly spaced time-series data. This package brings a few of them to R. The functions describe here offer various rolling operators.

**Usage**

```
rollingCentralMoment(times, values, widthbefore, widthafter, moment)
```

```
rollingMax(times, values, widthbefore, widthafter)
```

```
rollingMean(times, values, widthbefore, widthafter)
```

```
rollingMedian(times, values, widthbefore, widthafter)
```

```
rollingMin(times, values, widthbefore, widthafter)
```

```
rollingNobs(times, values, widthbefore, widthafter)
```

```
rollingProduct(times, values, widthbefore, widthafter)
```

```
rollingSD(times, values, widthbefore, widthafter)
```

```
rollingSum(times, values, widthbefore, widthafter)
```

```
rollingSumStable(times, values, widthbefore, widthafter)
```

```
rollingVar(times, values, widthbefore, widthafter)
```

**Arguments**

times	A Datetime vector
values	A numeric vector
widthbefore	A double with the preceding observation width
widthafter	A double with the subsequent observation width
moment	A double with the requested moment.

**Value**

A numeric vector with the corresponding result.

**Author(s)**

Dirk Eddelbuettel for the package, Andreas Eckner for the underlying code.

**Description**

The UTS library by Andreas Eckner provides algorithms for unevenly spaced time-series data. This package brings a few of them to R. The functions describe here offer simple moving average, or SMA, for short. Three variants are provided considering the last or next observation relative to time 't', as well as linear interpolation between them.

**Usage**

```
SMAnext(times, values, widthbefore, widthafter)
```

```
SMAlast(times, values, widthbefore, widthafter)
```

```
SMAlinear(times, values, widthbefore, widthafter)
```

**Arguments**

times	A Datetime vector
values	A numeric vector
widthbefore	A double with the preceding observation width
widthafter	gvA double with the subsequent observation width

**Value**

A numeric vector with SMA-weighted values. package at the given position is available.

**Author(s)**

Dirk Eddelbuettel for the package, Andreas Eckner for the underlying code.

**Examples**

```
if (requireNamespace("xts", quietly=TRUE)) {
  suppressMessages(library(xts))
  times <- ISOdatetime(2018, 6, 7, 8, 30, 0) + c(0, 1.0, 1.2, 2.3, 2.9, 5.0)
  values <- seq(0, 10, by=2)
  plot(xts(values, order.by=times), type="b",
        main="Series and last/next/linear SMAs",
        major.ticks="auto", grid.ticks.on="auto")
  lines(xts(SMAlast(times,values, 2.5, 1), values, order.by=times),
        type="b", col="lightblue")
  lines(xts(SMAnext(times,values, 2.5, 1), values, order.by=times),
        type="b", col="darkblue")
  lines(xts(SMAlinear(times,values, 2.5, 1), values, order.by=times),
        type="b", col="mediumblue")
  addLegend("topleft", legend.names=c("series", "SMAlast", "SMAnext", "SMAlinear"),
           lty=rep(1,4), lwd=rep(1,4),
           col=c("black", "lightblue", "darkblue", "mediumblue"))
}
```

---

`utsExample`*Irregularly spaced time series example*

---

**Description**

The UTS library by Andreas Eckner provides algorithms for unevenly spaced time-series data. This package brings a few of them to R. This function shows the original example.

**Usage**`utsExample()`**Value**

Nothing

**Author(s)**

Dirk Eddelbuettel for the package, Andreas Eckner for the underlying code.

**Examples**`utsExample()`

# Index

## \* package

RcppUTS-package, 2

EMAlast (EMAnext), 2

EMAlinear (EMAnext), 2

EMAnext, 2

RcppUTS (RcppUTS-package), 2

RcppUTS-package, 2

rollingCentralMoment, 3

rollingMax (rollingCentralMoment), 3

rollingMean (rollingCentralMoment), 3

rollingMedian (rollingCentralMoment), 3

rollingMin (rollingCentralMoment), 3

rollingNobs (rollingCentralMoment), 3

rollingProduct (rollingCentralMoment), 3

rollingSD (rollingCentralMoment), 3

rollingSum (rollingCentralMoment), 3

rollingSumStable

(rollingCentralMoment), 3

rollingVar (rollingCentralMoment), 3

SMAlast (SMAnext), 4

SMAlinear (SMAnext), 4

SMAnext, 4

utsExample, 6