Package 'RcppRoll'

January 20, 2025

Type Package

Title Efficient Rolling / Windowed Operations

Version 0.3.1

Author Kevin Ushey

Maintainer Kevin Ushey <kevinushey@gmail.com>

Description Provides fast and efficient routines for
common rolling / windowed operations. Routines for the
efficient computation of windowed mean, median,
sum, product, minimum, maximum, standard deviation
and variance are provided.

License GPL (>= 2)

Depends R (>= 2.15.1)

Suggests zoo, testthat

Imports Rcpp

LinkingTo Rcpp

RoxygenNote 6.0.1

NeedsCompilation yes

Repository CRAN

Date/Publication 2024-07-07 11:10:02 UTC

Contents

RcppRoll   *RcppRoll*

Description

This package implements a number of 'roll'-ing functions for R vectors and matrices.

Details

Currently, the exported functions are:

- [roll_max](#)
- [roll_mean](#)
- [roll_median](#)
- [roll_min](#)
- [roll_prod](#)
- [roll_sd](#)
- [roll_sum](#)
- [roll_var](#)

RcppRoll-exports   *RcppRoll*

Description

Efficient windowed / rolling operations. Each function here applies an operation over a moving window of size n, with (customizable) weights specified through weights.

Usage

```
roll_mean(x, n = 1L, weights = NULL, by = 1L, fill = numeric(0),
  partial = FALSE, align = c("center", "left", "right"), normalize = TRUE,
  na.rm = FALSE)

roll_meanr(x, n = 1L, weights = NULL, by = 1L, fill = NA,
  partial = FALSE, align = "right", normalize = TRUE, na.rm = FALSE)

roll_meanl(x, n = 1L, weights = NULL, by = 1L, fill = NA,
  partial = FALSE, align = "left", normalize = TRUE, na.rm = FALSE)

roll_median(x, n = 1L, weights = NULL, by = 1L, fill = numeric(0),
  partial = FALSE, align = c("center", "left", "right"), normalize = TRUE,
  na.rm = FALSE)
```

```
roll_medianr(x, n = 1L, weights = NULL, by = 1L, fill = NA,
  partial = FALSE, align = "right", normalize = TRUE, na.rm = FALSE)

roll_medianl(x, n = 1L, weights = NULL, by = 1L, fill = NA,
  partial = FALSE, align = "left", normalize = TRUE, na.rm = FALSE)

roll_min(x, n = 1L, weights = NULL, by = 1L, fill = numeric(0),
  partial = FALSE, align = c("center", "left", "right"), normalize = TRUE,
  na.rm = FALSE)

roll_minr(x, n = 1L, weights = NULL, by = 1L, fill = NA,
  partial = FALSE, align = "right", normalize = TRUE, na.rm = FALSE)

roll_minl(x, n = 1L, weights = NULL, by = 1L, fill = NA,
  partial = FALSE, align = "left", normalize = TRUE, na.rm = FALSE)

roll_max(x, n = 1L, weights = NULL, by = 1L, fill = numeric(0),
  partial = FALSE, align = c("center", "left", "right"), normalize = TRUE,
  na.rm = FALSE)

roll_maxr(x, n = 1L, weights = NULL, by = 1L, fill = NA,
  partial = FALSE, align = "right", normalize = TRUE, na.rm = FALSE)

roll_maxl(x, n = 1L, weights = NULL, by = 1L, fill = NA,
  partial = FALSE, align = "left", normalize = TRUE, na.rm = FALSE)

roll_prod(x, n = 1L, weights = NULL, by = 1L, fill = numeric(0),
  partial = FALSE, align = c("center", "left", "right"), normalize = TRUE,
  na.rm = FALSE)

roll_prodr(x, n = 1L, weights = NULL, by = 1L, fill = NA,
  partial = FALSE, align = "right", normalize = TRUE, na.rm = FALSE)

roll_prodl(x, n = 1L, weights = NULL, by = 1L, fill = NA,
  partial = FALSE, align = "left", normalize = TRUE, na.rm = FALSE)

roll_sum(x, n = 1L, weights = NULL, by = 1L, fill = numeric(0),
  partial = FALSE, align = c("center", "left", "right"), normalize = TRUE,
  na.rm = FALSE)

roll_sumr(x, n = 1L, weights = NULL, by = 1L, fill = NA,
  partial = FALSE, align = "right", normalize = TRUE, na.rm = FALSE)

roll_suml(x, n = 1L, weights = NULL, by = 1L, fill = NA,
  partial = FALSE, align = "left", normalize = TRUE, na.rm = FALSE)

roll_sd(x, n = 1L, weights = NULL, by = 1L, fill = numeric(0),
```

```
    partial = FALSE, align = c("center", "left", "right"), normalize = TRUE,
    na.rm = FALSE)

  roll_sdr(x, n = 1L, weights = NULL, by = 1L, fill = NA,
    partial = FALSE, align = "right", normalize = TRUE, na.rm = FALSE)

  roll_sdl(x, n = 1L, weights = NULL, by = 1L, fill = NA,
    partial = FALSE, align = "left", normalize = TRUE, na.rm = FALSE)

  roll_var(x, n = 1L, weights = NULL, by = 1L, fill = numeric(0),
    partial = FALSE, align = c("center", "left", "right"), normalize = TRUE,
    na.rm = FALSE)

  roll_varr(x, n = 1L, weights = NULL, by = 1L, fill = NA,
    partial = FALSE, align = "right", normalize = TRUE, na.rm = FALSE)

  roll_varl(x, n = 1L, weights = NULL, by = 1L, fill = NA,
    partial = FALSE, align = "left", normalize = TRUE, na.rm = FALSE)
```

Arguments

x	A numeric vector or a numeric matrix.
n	The window size. Ignored when `weights` is non-NULL.
weights	A vector of length n, giving the weights for each element within a window. If NULL, we take unit weights of width n.
by	Calculate at every by-th point rather than every point.
fill	Either an empty vector (no fill), or a vector (recycled to) length 3 giving left, middle and right fills.
partial	Partial application? Currently unimplemented.
align	Align windows on the "left", "center" or "right".
normalize	Normalize window weights, such that they sum to n.
na.rm	Remove missing values?

Details

The functions postfixed with l and r are convenience wrappers that set **l**eft / **r**ight alignment of the windowed operations.

Index