Package ‘RcppRoll’

October 12, 2022

Type  Package
Title  Efficient Rolling / Windowed Operations
Version  0.3.0
Date  2018-06-05
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  2018-06-05 18:35:35 UTC

R topics documented:

RcppRoll .......................................................... 2
RcppRoll-exports .................................................. 2

Index  5

1
This package implements a number of 'roll'-ing functions for \( \mathbb{R} \) vectors and matrices.

Currently, the exported functions are:

- `roll_max`
- `roll_mean`
- `roll_median`
- `roll_min`
- `roll_prod`
- `roll_sd`
- `roll_sum`
- `roll_var`

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

**Usage**

```r
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),
RcppRoll-exports

```
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 left / right alignment of the windowed operations.
Index

RcppRoll, 2
RcppRoll-exports, 2
RcppRoll-package (RcppRoll), 2
roll_max, 2
roll_max (RcppRoll-exports), 2
roll_maxl (RcppRoll-exports), 2
roll_maxr (RcppRoll-exports), 2
roll_mean, 2
roll_mean (RcppRoll-exports), 2
roll meanl (RcppRoll-exports), 2
roll meanr (RcppRoll-exports), 2
roll_median, 2
roll_median (RcppRoll-exports), 2
roll medianl (RcppRoll-exports), 2
roll medianr (RcppRoll-exports), 2
roll_min, 2
roll_min (RcppRoll-exports), 2
roll_minl (RcppRoll-exports), 2
roll_minr (RcppRoll-exports), 2
roll_prod, 2
roll_prod (RcppRoll-exports), 2
roll prod1 (RcppRoll-exports), 2
roll prodr (RcppRoll-exports), 2
roll_sd, 2
roll_sd (RcppRoll-exports), 2
roll sd1 (RcppRoll-exports), 2
roll sdr (RcppRoll-exports), 2
roll_sum, 2
roll_sum (RcppRoll-exports), 2
roll suml (RcppRoll-exports), 2
roll sumr (RcppRoll-exports), 2
roll_var, 2
roll_var (RcppRoll-exports), 2
roll varl (RcppRoll-exports), 2
roll varr (RcppRoll-exports), 2