

Package ‘runner’

March 13, 2020

Title Running Operations for Vectors

Type Package

Version 0.3.2

Depends R (>= 3.0)

Language en-US

Encoding UTF-8

Maintainer Dawid Kałedkowski <dawid.kaledkowski@gmail.com>

Description Calculates running functions with varying window size and missing handling options. Package brings also running streak and running which, what extends beyond range of functions already implemented in R packages. Calculating on windows depending on date also available.

License GPL (>= 2)

BugReports <https://github.com/gogonzo/runner/issues>

LinkingTo Rcpp

Imports methods, Rcpp

Suggests dplyr, knitr, magrittr, rmarkdown, spelling, testthat

VignetteBuilder knitr

RoxygenNote 7.0.2

NeedsCompilation yes

Author Dawid Kałedkowski [aut, cre] (<<https://orcid.org/0000-0001-9533-457X>>)

Repository CRAN

Date/Publication 2020-03-13 20:10:02 UTC

R topics documented:

fill_run	2
lag_run	3
length_run	3
max_run	4

mean_run	5
minmax_run	6
min_run	7
runner	8
streak_run	10
sum_run	11
which_run	12
window_run	14

Index	16
--------------	-----------

fill_run	<i>Fill NA with previous non-NA element</i>
----------	---

Description

Fill NA with last non-NA element.

Usage

```
fill_run(x, run_for_first = FALSE, only_within = FALSE)
```

Arguments

x	Vector of any type
run_for_first	If first elements are filled with NA, run_for_first = TRUE allows to fill all initial NA with nearest non-NA value. By default run_for_first = TRUE
only_within	NA are replaced only if previous and next non-NA values are the same. By default only_within = TRUE

Value

vector - x containing all x elements with NA replaced with previous non-NA element.

Examples

```
fill_run(c(NA, NA, 1:10, NA, NA), run_for_first = TRUE)
fill_run(c(NA, NA, 1:10, NA, NA), run_for_first = TRUE)
fill_run(c(NA, NA, 1:10, NA, NA), run_for_first = FALSE)
fill_run(c(NA, NA, 1, 2, NA, NA, 2, 2, NA, NA, 1, NA, NA), run_for_first = TRUE, only_within = TRUE)
```

lag_run	<i>Lag dependent on variable</i>
---------	----------------------------------

Description

Vector of input lagged along integer vector

Usage

```
lag_run(x, lag = 1L, idx = integer(0), nearest = FALSE)
```

Arguments

x	Vector of any type
lag	(integer) vector or single value denoting window lag. If lag is a single value then window lag is constant for all elements, otherwise if <code>length(lag) == length(x)</code> different window size for each element. Negative value shifts window forward.
idx	(date or integer) an optional integer vector containing index of observation. If specified then k and lag are depending on idx. Length of idx should be equal of length x
nearest	logical single value. Applied when idx is used, then nearest = FALSE returns observation lagged exactly by the specified number of "periods". When nearest = TRUE function returns latest observation within lag window.

Examples

```
lag_run(1:10, lag = 3)
lag_run(letters[1:10], lag = -2, idx = c(1, 1, 1, 2, 3, 4, 6, 7, 8, 10))
lag_run(letters[1:10], lag = 2, idx = c(1, 1, 1, 2, 3, 4, 6, 7, 8, 10), nearest = TRUE)
```

length_run	<i>Length of running windows</i>
------------	----------------------------------

Description

Number of elements in k-long window calculated on idx vector. If idx is an 'as.integer(date)' vector, then k=number of days in window - then the result is number of observations within k days window.

Usage

```
length_run(k = integer(1), lag = integer(1), idx = integer(0))
```

Arguments

k	(integer) vector or single value denoting size of the running window. If k is a single value then window size is constant for all elements, otherwise if $\text{length}(k) == \text{length}(x)$ different window size for each element.
lag	(integer) vector or single value denoting window lag. If lag is a single value then window lag is constant for all elements, otherwise if $\text{length}(\text{lag}) == \text{length}(x)$ different window size for each element. Negative value shifts window forward.
idx	(date or integer) an optional integer vector containing index of observation. If specified then k and lag are depending on idx. Length of idx should be equal of length x

Examples

```
length_run(k = 3, idx = c(1, 2, 2, 4, 5, 5, 5, 5, 5, 5))
```

max_run	<i>Running maximum</i>
---------	------------------------

Description

min_run calculates running max on given x numeric vector, specified k window size.

Usage

```
max_run(
  x,
  k = integer(0),
  lag = integer(1),
  idx = integer(0),
  at = integer(0),
  na_rm = TRUE,
  na_pad = FALSE
)
```

Arguments

x	Vector of any type
k	(integer) vector or single value denoting size of the running window. If k is a single value then window size is constant for all elements, otherwise if $\text{length}(k) == \text{length}(x)$ different window size for each element.
lag	(integer) vector or single value denoting window lag. If lag is a single value then window lag is constant for all elements, otherwise if $\text{length}(\text{lag}) == \text{length}(x)$ different window size for each element. Negative value shifts window forward.

idx	(date or integer) an optional integer vector containing index of observation. If specified then k and lag are depending on idx. Length of idx should be equal of length x
at	(date or integer) vector of any size and any value defining output data points. Values of the vector defines the indexes which data is computed at. If idx is missing then uses indices from 1 to length(x), otherwise depends on indexes passed with idx. If at is defined then k and lag should be of length equal one or length of the at.
na_rm	logical single value (default na_rm = TRUE) - if TRUE sum is calculating excluding NA.
na_pad	logical single value (default na_pad=FALSE) - if TRUE calculation on incomplete window will return NA. Incomplete window is when some parts of the window are out of range

Value

max numeric vector of length equals length of x.

Examples

```
set.seed(11)
x1 <- sample( c(1,2,3), 15, replace=TRUE)
x2 <- sample( c(NA,1,2,3), 15, replace=TRUE)
k <- sample( 1:4, 15, replace=TRUE)
max_run(x1) # simple cumulative maximum
max_run(x2, na_rm = TRUE) # cumulative maximum with removing NA.
max_run(x2, na_rm = TRUE, k=4) # maximum in 4-element window
max_run(x2, na_rm = FALSE, k=k) # maximum in varying k window size
```

mean_run

Running mean

Description

Running mean in specified window of numeric vector.

Usage

```
mean_run(
  x,
  k = integer(0),
  lag = integer(1),
  idx = integer(0),
  at = integer(0),
  na_rm = TRUE,
  na_pad = FALSE
)
```

Arguments

x	numeric vector which running function is calculated on
k	(integer) vector or single value denoting size of the running window. If k is a single value then window size is constant for all elements, otherwise if $\text{length}(k) == \text{length}(x)$ different window size for each element.
lag	(integer) vector or single value denoting window lag. If lag is a single value then window lag is constant for all elements, otherwise if $\text{length}(\text{lag}) == \text{length}(x)$ different window size for each element. Negative value shifts window forward.
idx	(date or integer) an optional integer vector containing index of observation. If specified then k and lag are depending on idx. Length of idx should be equal of length x
at	(date or integer) vector of any size and any value defining output data points. Values of the vector defines the indexes which data is computed at. If idx is missing then uses indices from 1 to $\text{length}(x)$, otherwise depends on indexes passed with idx. If at is defined then k and lag should be of length equal one or length of the at.
na_rm	logical single value (default na_rm = TRUE) - if TRUE sum is calculating excluding NA.
na_pad	logical single value (default na_pad=FALSE) - if TRUE calculation on incomplete window will return NA. Incomplete window is when some parts of the window are out of range

Value

mean numeric vector of length equals length of x.

Examples

```
set.seed(11)
x1 <- rnorm(15)
x2 <- sample(c(rep(NA,5), rnorm(15)), 15, replace = TRUE)
k <- sample(1:15, 15, replace = TRUE)
mean_run(x1)
mean_run(x2, na_rm = TRUE)
mean_run(x2, na_rm = FALSE )
mean_run(x2, na_rm = TRUE, k=4)
```

minmax_run

Running min/max

Description

min_run calculates running minimum-maximum on given x numeric vector, specified k window size.

Usage

```
minmax_run(x, metric = "min", na_rm = TRUE)
```

Arguments

x	Vector of any type
metric	character what to return, minimum or maximum
na_rm	logical single value (default na_rm = TRUE) - if TRUE sum is calculating excluding NA.

Value

list.

min_run	<i>Running minimum</i>
---------	------------------------

Description

min_run calculates running min on given x numeric vector, specified k window size.

Usage

```
min_run(
  x,
  k = integer(0),
  lag = integer(1),
  idx = integer(0),
  at = integer(0),
  na_rm = TRUE,
  na_pad = FALSE
)
```

Arguments

x	Vector of any type
k	(integer) vector or single value denoting size of the running window. If k is a single value then window size is constant for all elements, otherwise if length(k) == length(x) different window size for each element.
lag	(integer) vector or single value denoting window lag. If lag is a single value then window lag is constant for all elements, otherwise if length(lag) == length(x) different window size for each element. Negative value shifts window forward.
idx	(date or integer) an optional integer vector containing index of observation. If specified then k and lag are depending on idx. Length of idx should be equal of length x

at	(date or integer) vector of any size and any value defining output data points. Values of the vector defines the indexes which data is computed at. If idx is missing then uses indices from 1 to length(x), otherwise depends on indexes passed with idx. If at is defined then k and lag should be of length equal one or length of the at.
na_rm	logical single value (default na_rm = TRUE) - if TRUE sum is calculating excluding NA.
na_pad	logical single value (default na_pad=FALSE) - if TRUE calculation on incomplete window will return NA. Incomplete window is when some parts of the window are out of range

Value

min numeric vector of length equals length of x.

Examples

```
set.seed(11)
x1 <- sample(c(1, 2, 3), 15, replace = TRUE)
x2 <- sample(c(NA, 1, 2, 3), 15, replace = TRUE)
k <- sample(1:4, 15, replace = TRUE)
min_run(x1)
min_run(x2, na_rm = TRUE)
min_run(x2, na_rm = TRUE, k = 4)
min_run(x2, na_rm = FALSE, k = k)
```

runner	<i>Apply running function</i>
--------	-------------------------------

Description

Applies custom function on running windows.

Usage

```
runner(
  x,
  f,
  k = integer(0),
  lag = integer(1),
  idx = integer(0),
  at = integer(0),
  na_pad = FALSE,
  ...,
  type = "auto"
)
```


Arguments

x	Vector of any type
f	function to be applied on windows created from x
k	(integer) vector or single value denoting size of the running window. If k is a single value then window size is constant for all elements, otherwise if $\text{length}(k) == \text{length}(x)$ different window size for each element.
lag	(integer) vector or single value denoting window lag. If lag is a single value then window lag is constant for all elements, otherwise if $\text{length}(\text{lag}) == \text{length}(x)$ different window size for each element. Negative value shifts window forward.
idx	(date or integer) an optional integer vector containing index of observation. If specified then k and lag are depending on idx. Length of idx should be equal of length x
at	(date or integer) vector of any size and any value defining output data points. Values of the vector defines the indexes which data is computed at. If idx is missing then uses indices from 1 to $\text{length}(x)$, otherwise depends on indexes passed with idx. If at is defined then k and lag should be of length equal one or length of the at.
na_pad	logical single value (default na_pad=FALSE) - if TRUE calculation on incomplete window will return NA. Incomplete window is when some parts of the window are out of range
...	other arguments passed to the function f.
type	output type ("logical", "numeric", "integer", "character"). runner by default returns numeric values, but if function is expected to return other type, user should specify this in type argument.

Value

vector with aggregated values for each window. Length of output is the same as $\text{length}(x)$ or $\text{length}(\text{at})$ if specified. Type of the output is taken from type argument.

Examples

```
# mean on k = 3 elements windows
runner(1:10, f = mean, k = 3)

# mean on k = 3 elements windows with different specification
runner(1:10, k = 3, f = function(x) mean(x, na.rm = TRUE))

# number of unique values in each window (varying window size)
runner(letters[1:10],
      k = c(1, 2, 2, 4, 5, 5, 5, 5, 5, 5),
      f = function(x) length(unique(x)))

# concatenate window values
runner(letters[1:10],
```

```

k = c(1, 2, 2, 4, 5, 5, 5, 5, 5, 5),
f = function(x) paste(x, collapse = "-"),
type = "character")

# concatenate only on selected windows index
runner(letters[1:10],
      f = function(x) paste(x, collapse = "-"),
      at = c(1, 5, 8),
      type = "character")

```

streak_run

Running streak length

Description

Calculates running series of consecutive elements

Usage

```

streak_run(
  x,
  k = integer(0),
  lag = integer(1),
  idx = integer(0),
  at = integer(0),
  na_rm = TRUE,
  na_pad = FALSE
)

```

Arguments

x	any type vector which running function is calculated on
k	(integer) vector or single value denoting size of the running window. If k is a single value then window size is constant for all elements, otherwise if $\text{length}(k) == \text{length}(x)$ different window size for each element.
lag	(integer) vector or single value denoting window lag. If lag is a single value then window lag is constant for all elements, otherwise if $\text{length}(\text{lag}) == \text{length}(x)$ different window size for each element. Negative value shifts window forward.
idx	(date or integer) an optional integer vector containing index of observation. If specified then k and lag are depending on idx. Length of idx should be equal of length x
at	(date or integer) vector of any size and any value defining output data points. Values of the vector defines the indexes which data is computed at. If idx is missing then uses indices from 1 to $\text{length}(x)$, otherwise depends on indexes passed with idx. If at is defined then k and lag should be of length equal one or length of the at.

na_rm	logical single value (default na_rm = TRUE) - if TRUE sum is calculating excluding NA.
na_pad	logical single value (default na_pad=FALSE) - if TRUE calculation on incomplete window will return NA. Incomplete window is when some parts of the window are out of range

Value

streak [numeric] vector of length equals length of x containing number of consecutive occurrences.

Examples

```
set.seed(11)
x1 <- sample(c("a","b"), 15, replace = TRUE)
x2 <- sample(c(NA_character_, "a", "b"), 15, replace = TRUE)
k <- sample(1:4, 15, replace = TRUE)
streak_run(x1) # simple streak run
streak_run(x1, k = 2) # streak run within 2-element window
streak_run(x2, na_pad = TRUE, k = 3) # streak run within k=3 with padding NA
streak_run(x1, k = k) # streak run within varying window size specified by vector k
```

sum_run

Running sum

Description

Running sum in specified window of numeric vector.

Usage

```
sum_run(
  x,
  k = integer(0),
  lag = integer(1),
  idx = integer(0),
  at = integer(0),
  na_rm = TRUE,
  na_pad = FALSE
)
```

Arguments

x numeric vector which running function is calculated on

k (integer) vector or single value denoting size of the running window. If k is a single value then window size is constant for all elements, otherwise if $\text{length}(k) == \text{length}(x)$ different window size for each element.

lag	(integer) vector or single value denoting window lag. If lag is a single value then window lag is constant for all elements, otherwise if <code>length(lag) == length(x)</code> different window size for each element. Negative value shifts window forward.
idx	(date or integer) an optional integer vector containing index of observation. If specified then k and lag are depending on idx. Length of idx should be equal of length x
at	(date or integer) vector of any size and any value defining output data points. Values of the vector defines the indexes which data is computed at. If idx is missing then uses indices from 1 to <code>length(x)</code> , otherwise depends on indexes passed with idx. If at is defined then k and lag should be of length equal one or length of the at.
na_rm	logical single value (default <code>na_rm = TRUE</code>) - if TRUE sum is calculating excluding NA.
na_pad	logical single value (default <code>na_pad=FALSE</code>) - if TRUE calculation on incomplete window will return NA. Incomplete window is when some parts of the window are out of range

Value

sum code vector of length equals length of x.

Examples

```
set.seed(11)
x1 <- rnorm(15)
x2 <- sample(c(rep(NA, 5), rnorm(15)), 15, replace = TRUE)
k <- sample(1:15, 15, replace = TRUE)
sum_run(x1)
sum_run(x2, na_rm = TRUE)
sum_run(x2, na_rm = FALSE)
sum_run(x2, na_rm = TRUE, k = 4)
```

which_run

Running which

Description

`min_run` calculates running which - returns index of element where `x == TRUE`.

Usage

```
which_run(
  x,
  k = integer(0),
  lag = integer(1),
  idx = integer(0),
```

```

    at = integer(0),
    which = "last",
    na_rm = TRUE,
    na_pad = FALSE
  )

```

Arguments

x	Vector of any type
k	(integer) vector or single value denoting size of the running window. If k is a single value then window size is constant for all elements, otherwise if <code>length(k) == length(x)</code> different window size for each element.
lag	(integer) vector or single value denoting window lag. If lag is a single value then window lag is constant for all elements, otherwise if <code>length(lag) == length(x)</code> different window size for each element. Negative value shifts window forward.
idx	(date or integer) an optional integer vector containing index of observation. If specified then k and lag are depending on idx. Length of idx should be equal of length x
at	(date or integer) vector of any size and any value defining output data points. Values of the vector defines the indexes which data is computed at. If idx is missing then uses indices from 1 to <code>length(x)</code> , otherwise depends on indexes passed with idx. If at is defined then k and lag should be of length equal one or length of the at.
which	character value "first" or "last" denoting if the first or last TRUE index is returned from the window.
na_rm	logical single value (default <code>na_rm = TRUE</code>) - if TRUE sum is calculating excluding NA.
na_pad	logical single value (default <code>na_pad=FALSE</code>) - if TRUE calculation on incomplete window will return NA. Incomplete window is when some parts of the window are out of range

Value

integer vector of indexes of the same length as x.

Examples

```

set.seed(11)
x1 <- sample(c(1, 2, 3), 15, replace = TRUE)
x2 <- sample(c(NA, 1, 2, 3), 15, replace = TRUE)
k <- sample(1:4, 15, replace = TRUE)
which_run(x1)
which_run(x2, na_rm = TRUE)
which_run(x2, na_rm = TRUE, k = 4)
which_run(x2, na_rm = FALSE, k = k)

```

 window_run

List of running windows

Description

Creates list of windows with given arguments settings. Length of output list is equal

Usage

```

window_run(
  x,
  k = integer(0),
  lag = integer(1),
  idx = integer(0),
  at = integer(0),
  na_pad = FALSE
)

```

Arguments

x	Vector of any type
k	(integer) vector or single value denoting size of the running window. If k is a single value then window size is constant for all elements, otherwise if $\text{length}(k) == \text{length}(x)$ different window size for each element.
lag	(integer) vector or single value denoting window lag. If lag is a single value then window lag is constant for all elements, otherwise if $\text{length}(\text{lag}) == \text{length}(x)$ different window size for each element. Negative value shifts window forward.
idx	(date or integer) an optional integer vector containing index of observation. If specified then k and lag are depending on idx. Length of idx should be equal of length x
at	(date or integer) vector of any size and any value defining output data points. Values of the vector defines the indexes which data is computed at. If idx is missing then uses indices from 1 to $\text{length}(x)$, otherwise depends on indexes passed with idx. If at is defined then k and lag should be of length equal one or length of the at.
na_pad	logical single value (default na_pad=FALSE) - if TRUE calculation on incomplete window will return NA. Incomplete window is when some parts of the window are out of range

Value

list of vectors (windows). Length of list is the same as $\text{length}(x)$ or $\text{length}(\text{at})$ if specified, and length of each window is defined by k (unless window is out of range).

Examples

```
window_run(1:10, k = 3, lag = -1)
window_run(letters[1:10], k = c(1, 2, 2, 4, 5, 5, 5, 5, 5, 5))
```

Index

`fill_run`, 2

`lag_run`, 3

`length_run`, 3

`max_run`, 4

`mean_run`, 5

`min_run`, 7

`minmax_run`, 6

`runner`, 8

`streak_run`, 10

`sum_run`, 11

`which_run`, 12

`window_run`, 14