

Package ‘rollply’

August 29, 2016

Title Moving-Window Add-on for 'plyr'

Version 0.5.0

Author ``Alexandre Genin <alex@lecairn.org> [aut, cre]''

Maintainer Alexandre Genin <alex@lecairn.org>

Description Apply a function in a moving window, then
combine the results in a data frame.

Depends R (>= 3.1.1)

Imports plyr, Rcpp, stringr

License GPL (>= 2)

LazyData true

LinkingTo Rcpp

Suggests testthat, knitr, proj4, ggplot2, rgdal, tidyr, alphahull,
rmarkdown

URL <http://alex.lecairn.org/rollply.html>,
<https://github.com/alexgenin/rollply>

BugReports <https://github.com/alexgenin/rollply>

RoxygenNote 5.0.1

VignetteBuilder knitr

NeedsCompilation yes

Repository CRAN

Date/Publication 2016-03-24 23:27:19

R topics documented:

build_grid_ahull_crop	2
build_grid_ahull_fill	3
build_grid_identical	4
build_grid_squaretile	5
meadow	6
rollply	6

build_grid_ahull_crop *Regular grid in alpha-hull*

Description

Create a grid within the alpha hull of a set of points.

Usage

```
build_grid_ahull_crop(coords, npts, pad = NULL, grid_opts = list(alpha =  
  0.3, scale_alpha = TRUE))
```

Arguments

coords	A matrix or data.frame of coordinates with two columns
npts	The number of points before cropping to the alpha hull shape
pad	Ignored
grid_opts	A list with component alpha that controls the shape of the alpha hull (see details).

Details

This function creates a rectangular grid over a set of points, then computes the alpha hull of the set of points, and discards all the points of the new grid that fall outside of the hull.

This function computes the alpha hull of a set of points which is an estimate of its overall outline. The intricateness of this outline is determined by the alpha parameter: a value close to zero will produce a very intricate outline and a high value will produce a coarse outline (similar to the convex hull). It is highly recommended to try different values of alpha to see what is most appropriate for one's datasets.

Value

The coordinates of a grid of points as a data.frame with `ncol(coords)` columns. Names are transferred from the coords data frame.

See Also

[ahull](#)

Other grid.building.functions: [build_grid_ahull_fill](#), [build_grid_identical](#), [build_grid_squaretile](#)

build_grid_ahull_fill *Regular grid in alpha-hull*

Description

Create a grid within the alpha hull of a set of points.

Usage

```
build_grid_ahull_fill(coords, npts, pad = NULL, grid_opts = list(alpha =  
  0.3, error_tol = 0.05, run_max = 20, verbose = FALSE))
```

Arguments

coords	A matrix or data.frame of coordinates with two columns
npts	The approximate number of points of the requested grid
pad	ignored argument
grid_opts	A list with component alpha, error_tol, run_max and verbose that controls the computation of the grid.

Details

This function computes the alpha hull of a set of points, then iteratively finds the best grid of npts points fitting in the hull.

This function computes the alpha hull of a set of points which is an estimate of its overall outline. The intricateness of this outline is determined by the alpha parameter: a value close to zero will produce a very intricate outline and a high value will produce a coarse outline (similar to the convex hull). It is highly recommended to try different values of alpha to see what is most appropriate for one's datasets.

Parameter error_tol controls the acceptable relative error in the grid number of points (e.g. 0.1 for 10 control the iteration process).

Value

The coordinates of a grid of points as a data.frame with approximately npts rows and ncol(coords) columns. Names are transferred from the coords data frame.

See Also

[ahull](#)

Other grid.building.functions: [build_grid_ahull_crop](#), [build_grid_identical](#), [build_grid_squartile](#)

Examples

```

data(meadow)

# Plot a grid with diagnostics
grid <- build_grid_ahull_fill(meadow[,c('lon','lat')], 10000,
                             grid_opts = list(verbose = TRUE))
plot(grid, pch = 21)

# See the influence of alpha parameter
par(mfrow = c(1, 4))
for ( alpha in c( .3, 1, 4, 10) ) {
  grid <- build_grid_ahull_fill(meadow[,c('lon','lat')], 10000,
                               grid_opts = list(alpha = alpha))
  plot(grid, pch = 21)
  title(paste0("alpha=",alpha))
}

```

build_grid_identical *Build a regular grid*

Description

Create a regular grid with the same number of points on each dimension.

Usage

```
build_grid_identical(coords, npts, pad = 0, ...)
```

Arguments

coords	A matrix or data frame of coordinates as columns
npts	The approximate total number of points of the output grid
pad	Padding on each dimension (a positive number makes a grid that is larger than the ranges of the coordinates).
...	other arguments are silently ignored

Details

This function creates a grid that covers a set of points. The number of points of the output grid is the same on all dimensions. This is probably the only useful option for 1D moving-window computations.

Value

The coordinates of a grid of points as a data frame with approximately npts rows and ncol(coords) columns. Names are transferred from the coords data frame.

See Also

Other grid.building.functions: [build_grid_ahull_crop](#), [build_grid_ahull_fill](#), [build_grid_squaretile](#)

build_grid_squaretile *Create a grid with square tiles*

Description

Create a rectangular grid of regularly-spaced points (square tiles).

Usage

```
build_grid_squaretile(coords, npts, pad = 0, ...)
```

Arguments

coords	A matrix or data.frame of coordinates with two columns
npts	The approximate total number of points of the output grid
pad	Padding on each dimension (a positive number makes a grid that is larger than the ranges of the coordinates).
...	other arguments are silently ignored

Details

This function creates a grid that covers a set of points. The distance between points is the same on all dimensions (tiles are squared. It is only implemented for 2D grid so coords must have at most two columns.

Value

The coordinates of a grid of points as a data.frame with approximately npts rows and ncol(coords) columns. Names are transferred from the coords data frame.

See Also

Other grid.building.functions: [build_grid_ahull_crop](#), [build_grid_ahull_fill](#), [build_grid_identical](#)

meadow	<i>Partial Data from a vegetation survey in a Californian subalpine meadow.</i>
--------	---

Description

Part of the results of a vegetation survey carried out in Yosemite, CA.

Usage

meadow

Format

A data frame with 217 rows and 4 variables:

lon approximate longitude of the plot

lat approximate latitude of the plot

meanvwc the mean volumetric water content of the superficial soil

tcover the total cover of vegetation in the plot

Source

<http://www.diamondse.info/>

References

A Multi-scale Approach to Searching for Disturbance Effects: Pack Stock and Sierra Nevada Meadows. Steven R. Lee, Eric L. Berlow, Steven M. Ostoja, John R. Matchett, Alexandre Genin, Stephen C. Hart, Matthew L. Brooks (in prep)

rollply	<i>Rollply</i>
---------	----------------

Description

Applies a function in a moving window then combines the results in a data.frame.

Usage

```
rollply(.data, .rollvars, wdw.size, fun, grid = NULL, grid_npts = NULL,
  grid_type = "identical", grid_opts = NULL, padding = "none",
  .parallel = FALSE, ...)
```

Arguments

<code>.data</code>	data.frame to be processed
<code>.rollvars</code>	a formula of the form $\sim a + b + \dots c$ describing over which variables the window should move. <code>a</code> and <code>b</code> denote the variables used for the rolling window and <code>c</code> an optional grouping variable.
<code>wdw.size</code>	window size
<code>fun</code>	function to apply
<code>grid</code>	data.frame of points at which the computation is done. If NULL then a grid is generated using <code>grid_npts</code> and <code>grid_type</code> .
<code>grid_npts</code>	if <code>grid</code> is unspecified, the number of points of the grid to build.
<code>grid_type</code>	The type of grid to generate (one of "squaretile", "identical", "ahull_crop", "ahull_fill").
<code>grid_opts</code>	A list of named options to be passed to the grid-generating function.
<code>padding</code>	padding policy at the edges of the dataset, one of 'none', 'outside', 'inside', or a numeric value
<code>.parallel</code>	whether to use parallel processing (see ddply for more information on parallelism).
<code>...</code>	other arguments passed to ddply and <code>fun</code>

Details

rollply applies a function in a window moving over one or more variables. It is built upon [ddply](#) so it inherits many of its useful options such as `.parallel` or `.progress`.

Rollply uses internally a grid spanning the coordinates specified in the formula. For each point of this grid, it then selects the corresponding subset of the data frame within `wdw.size`, and pass it to the function `fun`. The return values of `fun` are then combined into a data frame.

If specified, the grid should have column names matching the ones provided in the left-hand side of the formula. If the grid is unspecified, then rollply automatically computes an appropriate grid depending on the option `grid_type`:

- "identical" Creates a grid with an identical number of points on each dimension.
- "squaretile" Creates a grid with square tiles (points are equidistant on all dimensions)
- "ahull_crop" Creates a grid with square tiles, then crop the result to the alpha hull of the set of points
- "ahull_fill" Same as above, but iteratively tries to return a grid with a number of points similar to what was requested using `grid_npts`

Each of these grid types correspond to a function with prefix `build_grid`. Some of them have options that can be passed by providing a named list as argument `grid_opts`.

The padding policy indicates what to do at the edges of the dataset. A value of "inside" indicates that no value will be computed when the window is not entirely contained within the range of the dataset. This parameter does not apply for ahull-based grids.

Value

A data.frame with the function results at each grid point. Make sure the function results are fit for merging into a data.frame (i.e. that they can be merged using `rbind.fill`).

See Also

`build_grid_identical`, `build_grid_squaretile`, `build_grid_ahull_crop`, `build_grid_ahull_fill`

Examples

```
# see also vignette("rollply") for a visual introduction
library(plyr)

# 1D example: make a trendline for a time-series
dat <- data.frame(time = seq.int(1000),
                  position = cumsum(rnorm(1000,0,10)))

rollav <- rollply(dat, ~ time, wdw.size=10,
                 summarise, position=mean(position))

plot(position ~ time, data = dat, pch = 20)
lines(rollav, col = 'red', lwd = 3)

# 2D example

# Generate three 2D random walks
dat <- ddply(data.frame(person = c('francois', 'nicolas', 'jacques')),
            ~ person,
            summarise, time = seq.int(1000),
                    x = cumsum(rnorm(1000,0,1)),
                    y = cumsum(rnorm(1000,0,1)))

# Smoothed trajectory over ten time-steps
rollav <- rollply(dat, ~ time | person, wdw.size = 10, grid_res = 1000,
                 summarise, x = mean(x), y = mean(y))

if ( require(ggplot2) ) {
  ggplot(dat, aes(x, y, color = person)) +
    geom_point(alpha = .5, shape = '+') +
    geom_path(data = rollav)
}

# Where did people spend their time ?
# we pregenerate the grid to fix it across groups
fixed_grid <- build_grid_squaretile(dat[,c('x','y')], 2000)
rollav <- rollply(dat, ~ x + y | person, wdw.size = 2, grid = fixed_grid,
                 summarise, time.spent = length(time))

if ( require(ggplot2) ) {
  ggplot(subset(rollav, time.spent > 0)) +
    geom_point(aes(x, y, color = person, size = time.spent), alpha = .5) +
```


rollply

9

```
    facet_grid(~person)  
  }
```

Index

*Topic **datasets**

meadow, [6](#)

ahull, [2](#), [3](#)

build_grid_ahull_crop, [2](#), [3](#), [5](#)

build_grid_ahull_fill, [2](#), [3](#), [5](#)

build_grid_identical, [2](#), [3](#), [4](#), [5](#)

build_grid_squartile, [2](#), [3](#), [5](#), [5](#)

ddply, [7](#)

meadow, [6](#)

rbind.fill, [8](#)

rollply, [6](#)