Package 'SpatialKDE'

October 26, 2022

Type Package

Title Kernel Density Estimation for Spatial Data

Version 0.8.0

URL https://jancaha.github.io/SpatialKDE/index.html,
https://github.com/JanCaha/SpatialKDE

Description Calculate Kernel Density Estimation (KDE) for spatial data.
The algorithm is inspired by the tool 'Heatmap' from 'QGIS'. The method is described by:
Hart, T., Zandbergen, P. (2014) <doi:10.1108/PIJPSM-04-2013-0039>,
Nelson, T. A., Boots, B. (2008) <doi:10.1111/j.0906-7590.2008.05548.x>,
Chainey, S., Tompson, L., Uhlig, S.(2008) <doi:10.1057/palgrave.sj.8350066>.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.2.1

VignetteBuilder knitr

LinkingTo cpp11, progress

SystemRequirements C++11

Imports sf, dplyr, glue, magrittr, rlang, vctrs, methods, raster

Suggests tmap, sp, knitr, rgdal, rmarkdown, testthat (>= 2.99.0), xml2

Config/testthat/edition 3

NeedsCompilation yes

Author Jan Caha [aut, cre] (<https://orcid.org/0000-0003-0165-0606>)

Maintainer Jan Caha <jan.caha@outlook.com>

Repository CRAN

Date/Publication 2022-10-26 16:45:12 UTC

R topics documented:

`create_grid_rectangular` *Create grid*

Description

Create grid of equally spaced rectangles or hexagons. The distance between centre points in both x and y dimension is equal to `cell_size`. The function is effectively a wrapper around `st_make_grid` with a little bit of preprocessing including generation of grid only inside `st_convex_hull`.

Usage

```
create_grid_rectangular(
  geometry,
  cell_size,
  side_offset = 0,
  only_inside = FALSE
)

create_grid_hexagonal(
  geometry,
  cell_size,
  side_offset = 0,
  only_inside = FALSE
)
```

Arguments

`geometry`	`sf` `data.frame` containing geometry which should be cover by the grid.
`cell_size`	`numeric` specifying the distance for equally spaced centers of polygons (rectangular or hexagonal).
`side_offset`	`numeric` specifying the side offset, distance added to the convex hull of input geometry to generate grid for KDE. Good estimate is usually the same value as band width of KDE.
`only_inside`	`logical` specifying if the grid cells should be generated only inside of the geometry. Default value is `FALSE`.

Value

`sf` `data.frame`.

Functions

- `create_grid_rectangular()`: Create rectangular grid
- `create_grid_hexagonal()`: Create hexagonal grid

Examples

```
library(sf)
nc <- st_read(system.file("shape/nc.shp", package = "sf")) %>% st_transform(32031)
grid <- create_grid_hexagonal(nc, cell_size = 100000)
grid <- create_grid_rectangular(nc, cell_size = 100000, only_inside = TRUE)
```

---

create_raster *Create raster*

Description

Create raster of equally spaced cells. The distance between centre of cells in both x and y dimension is equal to `cell_size`.

Usage

```
create_raster(geometry, cell_size, side_offset = 0)
```

Arguments

`geometry`	`sf` `data.frame` containing geometry which should be cover by the raster.
`cell_size`	`numeric` specifying the distance for equally spaced cells.
`side_offset`	`numeric` specifying the side offset, distance added to the convex hull of input geometry to generate raster for KDE. Good estimate is usually the same value as band width of KDE.

Value

`Raster-class`

Examples

```
library(sf)
nc <- st_read(system.file("shape/nc.shp", package = "sf")) %>% st_transform(32031)
raster <- create_raster(nc, cell_size = 100000)
```

kde *Kernel Density Estimation*

Description

KDE for spatial data. The algorithm is heavily inspired by Heatmap tool in QGIS. The help for QGIS tools is provided at the QGIS website. The a tutorial is provided here.

Usage

```
kde(
  points,
  band_width,
  decay = 1,
  kernel = c("quartic", "uniform", "triweight", "epanechnikov", "triangular"),
  scaled = FALSE,
  weights = c(),
  grid,
  cell_size
)
```

Arguments

points	sf data.frame containing only POINTS.
band_width	numeric specifying the band width for KDE.
decay	numeric specifying the decay parameter for "triangular" kernel. For other kernels besides "triangular" the parameter is not used.
kernel	character specifying type of kernel to use. Available implemented kernels are "uniform", "quartic", "triweight", "epanechnikov", "triangular". Default is "quartic" and if unknown kernel name is used it falls back to the default value.
scaled	logical specifying if the output values should be scaled. Default value is FALSE.
weights	numeric vector of weights for individual points.
grid	either sf data.frame (outcome of function create_grid_rectangular or create_grid_hexagonal) or Raster-class (outcome of function create_raster). Does not have to be specified if cell_size is set.
cell_size	numeric specifying the distance for equal spaced points. Must be higher than 0. Can be left out if grid is provided as grid is used instead. The code used to generate grid is create_grid_rectangular(points, cell_size, band_width).

Details

grid parameter specifies output of the function. KDE is calculated on the specified grid. If grid is Raster-class then outcome is also Raster-class. If grid is sf data.frame then outcome is also sf data.frame.

Value

either sf data.frame or Raster-class depending on class of grid parameter.

Examples

```
library(sf)
nc <- st_read(system.file("shape/nc.shp", package = "sf")) %>% st_transform(32031)
grid <- create_grid_hexagonal(nc, cell_size = 100000)
points <- st_sample(nc, 500) %>% st_as_sf()
kde_estimate_grid <- kde(points, band_width = 150000, grid = grid)
raster <- create_raster(nc, cell_size = 100000)
kde_estimate_raster <- kde(points, band_width = 150000, grid = raster)
```

Index