Package 'geogrid'

August 19, 2023

Title Turn Geospatial Polygons into Regular or Hexagonal Grids

Version 0.1.2

Description Turn irregular polygons (such as geographical regions) into regular or hexagonal grids. This package enables the generation of regular (square) and hexagonal grids through the package 'sp' and then assigns the content of the existing polygons to the new grid using the Hungarian algorithm, Kuhn (1955) (<doi:10.1007/978-3-540-68279-0_2>). This prevents the need for manual generation of hexagonal grids or regular grids that are supposed to reflect existing geography.

Imports methods, sp, sf, Rcpp

License MIT + file LICENSE

Encoding UTF-8

LinkingTo Rcpp, RcppArmadillo

Suggests testthat, lintr, covr

RoxygenNote 7.2.3

URL https://github.com/jbaileyh/geogrid

BugReports https://github.com/jbaileyh/geogrid/issues

NeedsCompilation yes

Author Joseph Bailey [aut, cph],
Ryan Hafen [ctb, cre] (<https://orcid.org/0000-0002-5516-8367>),
Jakub Nowosad [ctb] (<https://orcid.org/0000-0002-1057-3721>),
Lars Simon Zehnder [ctb] (RcppArmadillo implmentation of Munkres' Assignment Algorithm)

Maintainer Ryan Hafen <rhafen@gmail.com>

Repository CRAN

Date/Publication 2023-08-19 04:52:36 UTC

R topics documented:

`assign_polygons` *Assign the polygons in the original spatial data to their new location.*

Description

Assigns each polygon in the original file to a new location in the gridded geometry using the Hungarian algorithm.

Usage

```
assign_polygons(shape, new_polygons)

## S3 method for class 'SpatialPolygonsDataFrame'
assign_polygons(shape, new_polygons)

## S3 method for class 'sf'
assign_polygons(shape, new_polygons)
```

Arguments

`shape`	A "SpatialPolygonsDataFrame" or an sf object representing the original spatial polygons.
`new_polygons`	A "geogrid" object returned from `calculate_grid`.

Value

An object of the same class as shape

Examples

```
library(sf)
input_file <- system.file("extdata", "london_LA.json", package = "geogrid")
original_shapes <- st_read(input_file) %>% st_set_crs(27700)

# calculate grid
new_cells <- calculate_grid(shape = original_shapes,
  grid_type = "hexagonal", seed = 1)
grid_shapes <- assign_polygons(original_shapes, new_cells)
plot(grid_shapes)
```

```
par(mfrow = c(1, 2))
plot(st_geometry(original_shapes))
plot(st_geometry(grid_shapes))

## Not run:
# look at different grids using different seeds
par(mfrow=c(2, 3), mar = c(0, 0, 2, 0))
for (i in 1:6) {
  new_cells <- calculate_grid(shape = original_shapes,
    grid_type = "hexagonal", seed = i)
  plot(new_cells, main = paste("Seed", i, sep=" "))
}

## End(Not run)
```

`calculate_cell_size` *Calculate size of grid items (deprecated).*

Description

Given an input multipolgyon spatial data frame this function calculates the required cell size of a regular or hexagonal grid.

Usage

```
calculate_cell_size(
  shape,
  shape_details = NULL,
  learning_rate = 0.03,
  grid_type = c("hexagonal", "regular"),
  seed = NULL,
  verbose = FALSE
)
```

Arguments

shape	A 'SpatialPolygonsDataFrame' object representing the original spatial polygons.
shape_details	deprecated.
learning_rate	The rate at which the gradient descent finds the optimum cellsize to ensure that your gridded points fit within the outer boundary of the input polygons.
grid_type	Either 'hexagonal' for a hexagonal grid (default) or 'regular' for a regular grid.
seed	An optional random seed integer to be used for the grid calculation algorithm.
verbose	A logical indicating whether messages should be printed as the algorithm iterates.

calculate_grid *Calculate grid from spatial polygons.*

Description

Given an input multipolgyon spatial data frame this function calculates a hexagonal or regular grid that strives to preserve the original geography.

Usage

```
calculate_grid(
  shape,
  learning_rate = 0.03,
  grid_type = c("hexagonal", "regular"),
  seed = NULL,
  verbose = FALSE
)

## S3 method for class 'SpatialPolygonsDataFrame'
calculate_grid(
  shape,
  learning_rate = 0.03,
  grid_type = c("hexagonal", "regular"),
  seed = NULL,
  verbose = FALSE
)

## S3 method for class 'sf'
calculate_grid(
  shape,
  learning_rate = 0.03,
  grid_type = c("hexagonal", "regular"),
  seed = NULL,
  verbose = FALSE
)
```

Arguments

shape	A 'SpatialPolygonsDataFrame' or an sf object representing the original spatial polygons.
learning_rate	The rate at which the gradient descent finds the optimum cellsize to ensure that your gridded points fit within the outer boundary of the input polygons.
grid_type	Either 'hexagonal' for a hexagonal grid (default) or 'regular' for a regular grid.
seed	An optional random seed integer to be used for the grid calculation algorithm.
verbose	A logical indicating whether messages should be printed as the algorithm iterates.

Examples

```
library(sf)
input_file <- system.file('extdata', 'london_LA.json', package = 'geogrid')
original_shapes <- st_read(input_file) %>% st_set_crs(27700)

# calculate grid
new_cells <- calculate_grid(shape = original_shapes,
  grid_type = 'hexagonal', seed = 1)
grid_shapes <- assign_polygons(original_shapes, new_cells)
plot(grid_shapes)

par(mfrow = c(1, 2))
plot(st_geometry(original_shapes))
plot(st_geometry(grid_shapes))

## Not run:
# look at different grids using different seeds
par(mfrow=c(2, 3), mar = c(0, 0, 2, 0))
for (i in 1:6) {
  new_cells <- calculate_grid(shape = original_shapes,
    grid_type = 'hexagonal', seed = i)
  plot(new_cells, main = paste('Seed', i, sep=' '))
}

## End(Not run)
```

---

get_shape_details *Extract details from provided polygons (deprecated).*

Description

Extract spatial extent, range and other geospatial features from the output of read_polygons. Items are returned as a list for use in calculate_grid.

Usage

```
get_shape_details(input_shape)
```

Arguments

input_shape	A "SpatialPolygonsDataFrame" object representing the original spatial polygons.

get_shape_details_internal *Extract details from provided polygons.*

Description

Extract spatial extent, range and other geospatial features from the output of read_polygons. Items are returned as a list for use in calculate_grid.

Usage

```
get_shape_details_internal(input_shape)
```

Arguments

input_shape	A "SpatialPolygonsDataFrame" object representing the original spatial polygons.

hungariansafe_cc *hungariansafe_cc*

Description

hungariansafe_cc

Usage

```
hungariansafe_cc(cost)
```

Arguments

cost	cost matrix

hungarian_cc *hungarian_cc*

Description

hungarian_cc

Usage

```
hungarian_cc(cost)
```

Arguments

cost	cost matrix

plot.geogrid *Plot a 'geogrid' object*

Description

Plot a 'geogrid' object

Usage

```
## S3 method for class 'geogrid'
plot(x, y, ...)
```

Arguments

x	An object of class 'geogrid' to plot.
y	ignored
...	Additional parameters passed to the 'sp' package's plot method.

read_polygons *Import spatial data.*

Description

Simple function to read spatial data into a SpatialPolygonsDataFrame. Based on st_read from package sf.

Usage

```
read_polygons(file)
```

Arguments

file	A file path pointing to a shapefile or GeoJSON file, or a character string holding GeoJSON data. See the dsn argument of st_read for more details.

Index