Package ‘sugarbag’

November 9, 2022

Title Create Tessellated Hexagon Maps

Version 0.1.6

Description Create a hexagon tile map display from spatial polygons. Each polygon is represented by a hexagon tile, placed as close to its original centroid as possible, with a focus on maintaining spatial relationship to a focal point. Developed to aid visualisation and analysis of spatial distributions across Australia, which can be challenging due to the concentration of the population on the coast and wide open interior.

       https://github.com/srkobakian/sugarbag

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Depends R (>= 3.5.0), dplyr (>= 1.0.0)

Imports geosphere (>= 1.5), progress (>= 1.2.2), purrr (>= 0.3.4),
        rlang (>= 1.0.4), rmapshaper (>= 0.4.6), sf (>= 1.0-8), tibble
        (>= 3.1.7), tidyR (>= 1.2.0)

RoxygenNote 7.2.1

Suggests ggplot2 (>= 3.3.6), knitr, pkgdown, rmarkdown, testthat (>=
        2.1.0)

VignetteBuilder knitr

NeedsCompilation no

Author Dianne Cook [aut, cre, ths],
        Stephanie Kobakian [aut]

Maintainer Dianne Cook <dicook@monash.edu>

Repository CRAN

Date/Publication 2022-11-09 10:50:02 UTC
allocate

Allocate polygon centroids to hexagons in a grid

Description

Chooses a hexagon centroid for each polygon in the shape file, from a grid spanning the longitudes and latitudes in the expanded bounding box.

Usage

allocate(
  centroids,
  hex_grid,
  sf_id = names(centroids)[1],
  hex_size,
  hex_filter,
  focal_points = NULL,
  order_sf_id = NULL,
  width = 30,
  verbose
)

Arguments

centroids a data frame with centroids of non empty polygons
hex_grid a data frame containing all possible hexagon points
sf_id a string to indicate the column to identify individual polygons
hex_size       a float value in degrees for the diameter of the hexagons
hex_filter     amount of hexagons around centroid to consider
focal_points   a data frame of reference locations when allocating hexagons, capital cities of
               Australia are used in the example
order_sf_id    a string to indicate the column used to order polygons
width          a numeric indicating the angle used to filter the hexagon grid
verbose        a boolean to indicate whether to show polygon id

Value

a data frame of all allocated hexagon points

Examples

# Create centroids set
centroids <- create_centroids(tas_lga, sf_id = "lga_code_2016")
# Smaller set for faster example
centroids <- centroids[1:10,]
# Create hexagon location grid
data(capital_cities)
grid <- create_grid(centroids = centroids, hex_size = 0.2, buffer_dist = 1.2)
# Allocate polygon centroids to hexagon grid points
hex_allocated <- allocate(
  centroids = centroids,
  hex_grid = grid,
  hex_size = 0.2, # same size used in create_grid
  hex_filter = 3,
  focal_points = capital_cities,
  width = 30,
  verbose = TRUE
)
# NEXT:
# create a set of hexagon points for plotting
# using fortify_hexagon, and
# plot the hexagons with geom_polygon, see vignette

capital_cities               The point locations of Australian capital cities.

Description

A dataset containing the longitude and latitude values of Australian capital cities.

Usage

capital_cities
Format

A data frame with 8 rows and 3 variables:

- **points** name of cities
- **longitude** location of point in longitude degrees
- **latitude** location of point in latitude degrees

---

closest_focal_point  For the polygon provided, find the closest focal point in the set provided

---

Description

For one row of an sf data frame, calculate the distance to the closest focal point. Return the name of the focal point, and the angle between focal point and centroid.

Usage

```r
closest_focal_point(centroid, focal_points)
```

Arguments

- **centroid** a data frame describing one centroid
- **focal_points** a data frame of the longitude and latitude values

Value

data frame containing the name and location of the closest focal

Examples

```r
# Create a set of polygon centroids
centroids <- create_centroids(tas_sa2, "sa2_5dig_2016")

# Find the closest capital city for the first centroid
closest_focal_point(centroids[1, ], capital_cities)
```
**create_buffer**

*Expand points to extend beyond the outermost centroids*

**Description**

Called from within `create_grid` function, this function takes the bounding box of a group of polygons, or a specific table of minimum and maximum longitudes and latitudes to create points for each polygon to be allocated to that will tessellate into hexagons.

**Usage**

```r
create_buffer(centroids, grid, hex_size, buffer_dist, verbose = FALSE)
```

**Arguments**

- `centroids`: data frame of centroids to be allocated
- `grid`: data frame of hexagon centroids
- `hex_size`: a float value in degrees for the diameter of the hexagons
- `buffer_dist`: distance to extend beyond the geometry provided
- `verbose`: a boolean to indicate whether to show function progress

**Value**

data frame of hexagon centroids

**Examples**

```r
lga_centroids <- create_centroids(sugarbag::tas_lga, "lga_code_2016")
lga_grid <- create_grid(lga_centroids, hex_size = 0.2, buffer_dist = 1.2)
```

**create_centroids**

*Create a data frame of longitude and latitude centroids of each polygon.*

**Description**

Create a data frame of longitude and latitude centroids of each polygon.

**Usage**

```r
create_centroids(shp_sf, sf_id, largest = TRUE, verbose = FALSE)
```
create_grid

Arguments

shp_sf
sf_id
largest
verbose

an sf object, a data set with a simple feature list column
a string to indicate the column to identify individual polygons
logical; for st_centroid: if TRUE, return centroid of the largest subpolygon of a MULTIPOLYGON rather than the whole MULTIPOLYGON
a boolean to indicate whether to show function progress

Value

tibble containing longitude and latitude

Examples

centroids <- create_centroids(tas_lga, "lga_code_2016")

create_grid create a grid of evenly spaced points to allow hexagons to tessellate

Description

This function takes the output from the create_centroids function, or a set of centroids in a table with the columns latitude and longitude

Usage

create_grid(
  centroids,
  hex_size,
  buffer_dist,
  latitude = "latitude",
  longitude = "longitude",
  verbose = FALSE
)

Arguments

centroids
hex_size
buffer_dist
latitude
longitude
verbose

data frame of centroids to be allocated, this should have columns for longitude and latitude value of centroids, as
a float value in degrees for the diameter of the hexagons
distance to extend beyond the geometry provided
the column name for the latitude values of the centroids
the column name for the longitude values of the centroids
a boolean to indicate whether to show function progress
create_hexmap

Value

grid

Examples

# Create a set of centroids for grid to overlay
centroids <- create_centroids(tas_lga, "lga_code_2016")
# Create the grid
grid <- create_grid(centroids = centroids, hex_size = 0.2, buffer_dist = 1.2, verbose = FALSE)

create_hexmap

Create a tessellated hexagon map from a set of polygons

Description

Allocates each polygon in a shape file to a grid point to create a map of tessellated hexagons. The spatial relationships of areas are preserved while the geographic shape of each area is lost.

Usage

create_hexmap(
  shp,  # a shape file, if class is SPDF, will be converted to sf
  sf_id,  # name of a unique column that distinguishes areas
  hex_size = NULL,  # a float value in degrees for the diameter of the hexagons
  buffer_dist = NULL,  # distance in degrees to extend beyond the geometry provided
  hex_filter = 10,  # amount of hexagons around centroid to consider
  f_width = 30,  # the angle used to filter the grid points around a centroid
  focal_points = NULL,  # a data frame of reference locations when allocating hexagons, capital cities of Australia are used in the example
  order_sf_id = NULL,  # a string name of a column to order by for allocating
  export_shp = FALSE,  # export the simple features set
  verbose = FALSE  # a boolean to indicate whether to show function progress
)

Arguments

shp  
sf_id 
hex_size  
buffer_dist  
hex_filter  
f_width  
focal_points  
order_sf_id  
export_shp  
verbose
filter_grid_points

Value

a data set containing longitude and latitude of allocated hexagon points for each non null geometry passed in the shape file

Examples

data(tas_lga)
# Smaller set for faster example
tas_lga_sub <- tas_lga[1:10,]
data(capital_cities)
hexmap <- create_hexmap(
  shp = tas_lga_sub,
  sf_id = "lga_code_2016",
  hex_filter = 3,
  focal_points = capital_cities,
  verbose = TRUE)

filter_grid_points

Filter full set of grid points for those within range of original point

Description

Takes only the closest available gridpoints as possible hexagon centroids to allocate polygons.

Usage

filter_grid_points(
  f_grid,
  f_centroid,
  focal_points = NULL,
  f_dist = filter_dist,
  angle_width = width,
  h_size = hex_size
)

Arguments

f_grid complete grid of hexagon centroids
f_centroid the longitude and latitude values for the current centroid
focal_points a tibble of focal locations, an optional argument that allows allocation of polygons to hexagon centroids in ascending order of the distance to the closest focal point. It also filters the grid points to those within a 30 degree range of the angle from focal point to centroid. The default "capitals" uses the locations of the Australian capital cities as focal points.
**fortify_hexagon**

- **f_dist**: A distance in degrees, used as a boundary to filter the hexagon centroids considered for each polygon centroid to be allocated.
- **angle_width**: A numeric used to filter the hexagon grid.
- **h_size**: A float value in degrees for the diameter of the hexagons.

**Value**

A tibble of filtered grid points.

---

| fortify_hexagon | Creates the points that define a hexagon polygon for plotting |

**Description**

Creates the points that define a hexagon polygon for plotting.

**Usage**

```r
fortify_hexagon(data, sf_id, hex_size)
```

**Arguments**

- **data**: A data frame created by the allocate function.
- **sf_id**: A string to indicate the column to identify individual polygons.
- **hex_size**: A float value in degrees for the diameter of the hexagons.

**Value**

A data frame of the seven points used to draw a hexagon.

**Examples**

```r
# same column is used in create_centroids
fortify_hexagon(data = tas_lga_hexctr, sf_id = "lga_code_2016", hex_size = 0.2)
```
fortify_sfc  Convert a simple features tibble to tibble for plotting.

Description
This will contain individual points for plotting the polygon, indicating the longitude and latitude, order of points, if a hole is present, the piece, id and group.

Usage
fortify_sfc(sfc_df, keep = NULL)

Arguments
sfc_df a simples features data set
keep ratio of points to keep

Value
a tibble point of long lat points used to plot polygons

fp19  2019 Australian Federal election data: First preference votes for candidates (House of Representatives) in each electorate.

Description
A dataset containing first preference vote counts, candidate names, and other results for the House of Representatives from the 2016 Australian federal election. The data were obtained from the Australian Electoral Commission, and downloaded from https://results.aec.gov.au/24310/Website/Downloads/HouseFirstPrefsByPartyDownload-24310.csv

Usage
fp19

Format
A data frame with the following variables:
- StateAbAbbreviation for state name
- UniqueIDnumeric identifier that links the electoral division with Census and other election datasets.
- DivisionNmElectoral division name
- BallotPositionCandidate’s position on the ballot
The amount of homeless people in each Statistical Area at Level 2 in 2016.

A data frame of the Statistical Area at Level 2 names and amount of homeless

Usage

homeless

Format

A data frame with 545 rows and 2 variables:

- homeless amount of homeless people
- sa2_name_2016 name of the Statistical Area at Level 2

Description

read_shape

Read in the shape file as sf object

Usage

read_shape(shp_path, simplify = TRUE, keep = 0.1)
Arguments

shp_path character vector location of shape file, extension .shp
simplify a boolean to decide whether to simplify the shape file using rmapshaper, keeping all shapes.
keep ratio of points to keep

Value

an sf data frame, with a column of non null geometries

Examples

# Example of how a shape file is read
shape <- read_shape(shp_path = file.choose())

tas_lga

The polygons of Tasmanian Local Government Areas in 2016.

Description

A simple features dataset containing the polygons for all Australian LGAs in 2016.

Usage

tas_lga

Format

A simple features data frame with 39 rows and 6 variables:

lga_code_2016 code for the Local Government Area
lga_name_2016 name of the Local Government Area
ste_code_2016 code for the state containing the Local Government Area
ste_name_2016 name of the state containing the Local Government Area
areasqkm_2016 area contained in the polygon
geometry describes where on Earth the polygon is located
**tas_lga_hexctr**

The hexagon centres for polygons of Tasmanian Local Government Areas in 2016.

**Description**

A tibble dataset containing the processed data for all Australian LGAs in 2016. Each point corresponds to hexagon centre.

**Usage**

tas_lga_hexctr

**Format**

A simple features data frame with 39 rows and 6 variables:

- **lga_code_2016** code for the Local Government Area
- **longitude, latitude** polygon centroid
- **points, focal_longitude, focal_latitude, focal_dist, focal_angle** Focal point (capital city) information used for each polygon/hexagon
- **rownumber** row number, in case it can be useful
- **hex_long, hex_lat, hex_id** hexagon centre and id

---

**tas_sa2**

The polygons of Tasmanian Statistical Areas in 2016.

**Description**

A simple features dataset containing the polygons for all Tasmanian SA2s in 2016.

**Usage**

tas_sa2

**Format**

A simple features data frame with 99 rows and 15 variables:

- **sa2_main_2016** complete code of the Statistical Area
- **sa2_5dig_2016** simple code for the Statistical Area
- **sa2_name_2016** name for the Statistical Area
- **sa3_code_2016** code for the SA3 containing the Statistical Area
- **sa3_name_2016** name of the SA3 containing the Statistical Area
sa4_code_2016  code for the SA4 containing the Statistical Area
sa4_name_2016  name of the SA4 containing the Statistical Area
gcc_code_2016  code for the Greater Capital City region containing the Statistical Area
gcc_name_2016  name of the Greater Capital City region containing the Statistical Area
ste_code_2016  code for the state containing the Statistical Area
ste_name_2016  name of the state containing the Statistical Area
areasqkm_2016  area contained in the polygon
id   distinguishes SA2 regions
population  amount of people living within the region
sa2_code_2016  code of the Statistical Area
Index

* datasets
  capital_cities, 3
  fp19, 10
  homeless, 11
  tas_lga, 12
  tas_lga_hexctr, 13
  tas_sa2, 13

allocate, 2

capital_cities, 3
closest_focal_point, 4
create_buffer, 5
create_centroids, 5
create_grid, 6
create_hexmap, 7

filter_grid_points, 8
fortify_hexagon, 9
fortify_sfc, 10
fp19, 10

homeless, 11

read_shape, 11

tas_lga, 12

tas_lga_hexctr, 13

tas_sa2, 13