

Package: concaveman (via r-universe)

September 2, 2024

Type Package

Title A Very Fast 2D Concave Hull Algorithm

Version 1.1.0

Description The concaveman function ports the 'concaveman' (<<https://github.com/mapbox/concaveman>>) library from 'mapbox'. It computes the concave polygon(s) for one or several set of points.

License GPL-3

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

Depends R (>= 2.10)

Imports V8, sf, jsonlite

RoxygenNote 7.1.2

Suggests testthat

URL <https://joelgombin.github.io/concaveman/>,
<http://www.github.com/joelgombin/concaveman/>

BugReports <http://www.github.com/joelgombin/concaveman/issues>

SystemRequirements GDAL (>= 2.0.0), GEOS (>= 3.3.0), PROJ.4 (>= 4.8.0)

Repository <https://joelgombin.r-universe.dev>

RemoteUrl <https://github.com/joelgombin/concaveman>

RemoteRef HEAD

RemoteSha 35f09a51c9da5ce40b07a98d8c72738c3333c989

Contents

concaveman	2
points	3

Index	4
--------------	----------

concaveman

concaveman: A very fast 2D concave hull algorithm.

Description

This package is a simple R port (through [V8](#)) of a [JavaScript library by Vladimir Agafonkin](#).

The `concaveman` function ports the `concaveman` library from mapbox. It computes the concave polygon for one set of points.

Usage

```
concaveman(points, concavity, length_threshold)

## S3 method for class 'matrix'
concaveman(points, concavity = 2, length_threshold = 0)

## S3 method for class 'sf'
concaveman(points, concavity = 2, length_threshold = 0)

## S3 method for class 'sfc'
concaveman(points, concavity = 2, length_threshold = 0)
```

Arguments

<code>points</code>	the points for which the concave hull must be computed. Can be represented as a matrix of coordinates or an sf object.
<code>concavity</code>	a relative measure of concavity. 1 results in a relatively detailed shape, Infinity results in a convex hull. You can use values lower than 1, but they can produce pretty crazy shapes.
<code>length_threshold</code>	when a segment length is under this threshold, it stops being considered for further detalization. Higher values result in simpler shapes.

Details

For details regarding the implementation, please see the original javascript library [github page](#). This is just a thin wrapper, via [V8](#).

Value

an object of the same class as `points`: a matrix of coordinates or an sf object.

Examples

```
data(points)
polygons <- concaveman(points)
plot(points)
plot(polygons, add = TRUE)
```

points

Fixtures data

Description

This is just a test dataset which comes from the original mapbox library.

Usage

```
points
```

Format

an *sf* object with a 1000 points. Each of them is part of a group, indicated by variable *k* (generated by a *k*-means algorithm).

Source

<https://github.com/mapbox/concaveman/blob/master/test/fixtures/points-1k.json>

Index

* **datasets**

points, 3

concaveman, 2

points, 3