

# Package ‘concaveman’

July 26, 2017

**Type** Package

**Title** A Very Fast 2D Concave Hull Algorithm

**Version** 1.0.0

**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

**Depends** R (>= 2.10)

**Imports** V8, sf, magrittr, jsonlite, methods, dplyr

**RoxygenNote** 6.0.1

**Suggests** testthat, sp

**URL** <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)

**NeedsCompilation** no

**Author** Joël Gombin [cre, aut],  
Ramnath Vaidyanathan [aut],  
Vladimir Agafonkin [aut],  
Mapbox [cph]

**Maintainer** Joël Gombin <[joel.gombin@gmail.com](mailto:joel.gombin@gmail.com)>

**Repository** CRAN

**Date/Publication** 2017-07-25 22:37:09 UTC

## R topics documented:

concaveman . . . . .	2
points . . . . .	3

<b>Index</b>	<b>4</b>
--------------	----------

---

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 'SpatialPoints'
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, an <code>sf</code> object or a <code>SpatialPoints*</code> 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, an `sf` object or a `SpatialPoints*` 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

## \*Topic **datasets**

points, [3](#)

concaveman, [2](#)

concaveman-package (concaveman), [2](#)

concaveman.matrix (concaveman), [2](#)

concaveman.sf (concaveman), [2](#)

concaveman.SpatialPoints (concaveman), [2](#)

points, [3](#)