Package 'rLFT'

June 19, 2020

Type Package

Title Processing Linear Features

Version 1.0.0

Date 2020-06-11

Maintainer Shannon E Albeke <salbeke@uwyo.edu>

Description

Assists in the manipulation and processing of linear features with the help of the 'sf' package. Makes use of linear referencing to extract data from most shape files.
Reference for this packages methods: Albeke, S.E. et al. (2010) <doi:10.1007/s10980-010-9528-4>.

BugReports https://gitlab.com/datacorral/rlft/issues

Depends R (>= 3.3.0), Rcpp (>= 1.0.2), sf (>= 0.9-1)

Encoding UTF-8

License GPL (>= 3)

LazyData true

LinkingTo Rcpp

RoxygenNote 7.1.0

Suggests testthat, knitr, rmarkdown, ggplot2

VignetteBuilder knitr

NeedsCompilation yes

Author Samuel Fay [aut],
William Kirkpatrick [aut],
Shannon E Albeke [aut, cre]

Repository CRAN

Date/Publication 2020-06-19 09:50:03 UTC

R topics documented:

`addMValues` *Add M values to given feature*

Description

Add M values to a given linear feature and store them in the m-coordinate of the sf object. Returns the new sf object with added m-values. For more information on m-values and linear referencing see: http://desktop.arcgis.com/en/arcmap/10.3/guide-books/linear-referencing/what-is-linear-referencing.htm

Usage

```
addMValues(sfDataObject)
```

Arguments

`sfDataObject`	An sf object. Must be a LINESTRING, POLYGON, MULTIPOLYGON, or MULTILINESTRING

Value

Returns the new `sfDataObject` with added m-values. The class of the output is `sf`.

Examples

```
library(rLFT)
data("shpObject")
# Assign M Values to each vertex
mValues <- addMValues(shpObject)
print("M Values Added")
head(st_coordinates(mValues))
```

bct *Boundary Convexity Tool*

Description

Calculates raw convexity, convexity index, and sinuosity of a given sf object and returns a data frame with all measurements for each step and feature. If provided, the data will also be output to a tab delimited file.

Usage

```
bct(sfDataObject, step, window, ridName = NULL, filename = "")
```

Arguments

`sfDataObject`	An sf Object containing shape file data.
`step`	A numeric describing the distance between measurements along an arc.
`window`	A numeric describing the diameter of the window used to measure convexity.
`ridName`	A character denoting the column name where the unique ID for each feature is stored in given sf object.
`filename`	A character denoting the name of the file you wish to output convexity data to in tab delimited format. Must have the .txt extension.

Details

This function will reject any sf object with a geographic coordinate system, so consider projecting your features. Your sf object must be of either type LINESTRING, MULTILINESTRING, POLYGON, or MULTIPOLYGON. If a given POLYGON or MUTLIPOLYGON contains inner rings, they will be ignored. If a unique ID Column name is not provided, the function will generate a unique ID for each feature. The arguments `step` and `window` can be any non-negative numeric. The argument `ridName` MUST be a character indicating the name of the column in your sf object where the route id is stored.

Value

The output of this function is a `data.frame` that contains all measurements for each step and feature.

Reference

Albeke, S.E. et al. *"Measuring boundary convexity at multiple spatial scales using a linear "moving window" analysis: an application to coastal river otter habitat selection."* Landscape Ecology 25 (2010): 1575-1587. [linked phrase](https://link.springer.com/article/10.1007/s10980-010-9528-4)

Examples

```
library(rLFT)
data("shpObject")
#store convexity output data in a variable 'outputTable'
outputTable <- bct(shpObject, step = 50, window = 100, ridName = "RID")
```

`latlongShpObject` *An example sf object outlining boundaries of a group of islands in Alaska in the lat/lon CRS.*

Description

A dataset with coordinates (x, y, z) of 3 linear features with a geometry type of MULTIPOLYGON and a geographic coordinate system (longlat)

Usage

```
latlongShpObject
```

Format

A data frame with 1 row and 10 variables

`pointObject` *An example sf object with geometry type of POINT, used for testing.*

Description

An example sf object with geometry type of POINT, used for testing.

Usage

```
pointObject
```

Format

An object of class `sf` (inherits from `data.frame`) with 1 rows and 2 columns.

`polygonShpObject` *An example sf object outlining boundaries of a group of islands in Alaska cast as POLYGON.*

Description

A dataset with coordinates (x, y) of 37 features with a geometry type of POLYGON and a projected coordinate system (aea). These features represent the same data as 'shpObject' but this sf object has been cast to be of geometry type POLYGON. There are also inner polygons on some features.

Usage

```
polygonShpObject
```

Format

A data frame with 37 rows and 2 variables

Id Id's of all features

geometry coordinates of feature (x,y)

`shpObject` *An example sf object outlining boundaries of a group of islands in Alaska cast as LINESTRING.*

Description

A dataset with coordinates (x, y) of 37 linear features with a geometry type of LINESTRING and a projected coordinate system (aea). These particular features represent several small islands in Alaska.

Usage

```
shpObject
```

Format

A data frame with 37 rows and 2 variables

RID Route ID of the features

geometry coordinates of feature (x, y)

Index