

Package ‘geometries’

June 22, 2023

Type Package

Title Convert Between R Objects and Geometric Structures

Date 2023-06-22

Version 0.2.3

Description

Geometry shapes in 'R' are typically represented by matrices (points, lines), with more complex shapes being lists of matrices (polygons). 'Geometries' will convert various 'R' objects into these shapes.

Conversion functions are available at both the 'R' level, and through 'Rcpp'.

License MIT + file LICENSE

URL <https://dcooley.github.io/geometries/>

BugReports <https://github.com/dcooley/geometries/issues>

Encoding UTF-8

RoxygenNote 7.2.3

LinkingTo Rcpp

Imports Rcpp (>= 1.0.10)

Suggests covr, knitr, rmarkdown, tinytest

VignetteBuilder knitr

NeedsCompilation yes

Author David Cooley [aut, cre]

Maintainer David Cooley <david.cooley.au@gmail.com>

Repository CRAN

Date/Publication 2023-06-22 05:40:02 UTC

R topics documented:

gm_coordinates	2
gm_geometries	4

Index	7
--------------	----------

gm_coordinates	<i>gm_coordinates</i>
----------------	-----------------------

Description

Converts all coordinates from various geometric shapes into a single data.frame.

Usage

```
gm_coordinates(x)
```

Arguments

x object representing geometry shapes (e.g., list of matrices)

Details

The data.frame returned will always have an 'id' column. Then will follow an 'id+counter' column for every level of nesting the geometry is within.

The coordinates always start in column 'c1', the first column after all the id columns. Then there is a column 'c+counter' for every coordinate in the geometry.

This function is designed to handle multiple and different nested of geometry structures.

Value

a single data.frame representing all the values in the input lists and matrices.

Examples

```
x <- 1:3
gm_coordinates( x )

m <- matrix(1:12, ncol = 3)
gm_coordinates( m )

l <- list(
  matrix(1:12, ncol = 2 )
)
gm_coordinates( l )

l <- list(
  matrix(1:12, ncol = 4 )
)
gm_coordinates( l )

l <- list(
  list(
    matrix(1:12, ncol = 2)
  )
)
```

```
)
)
gm_coordinates( l )

l <- list(
  list(
    matrix(1:12, ncol = 2)
    , matrix(1:4, ncol = 2)
  )
)
gm_coordinates( l )

l <- list(
  list(
    matrix(1:12, ncol = 2)
    , matrix(1:4, ncol = 2)
  )
  , 1:5
  , 1:2
  , matrix(1:9, ncol = 3)
)
gm_coordinates( l )

l <- list(
  matrix(1:4, ncol = 2)
  , list(
    matrix(1:9, ncol = 3)
  )
)
gm_coordinates( l )

l <- list(
  list(
    list(
      matrix(1:12, ncol = 2)
    )
  )
  , list(
    list(
      matrix(1:24, ncol = 2)
    )
  )
)
gm_coordinates( l )

l <- list(
  list(
    list(
      matrix(1:12, ncol = 2)
    )
  )
  , list(
    list(
```

```

        matrix(1:3, ncol = 3)
      , matrix(1:24, ncol = 2)
    )
  )
)
gm_coordinates( l )

```

gm_geometries

geometries

Description

Converts a data.frame into a collection of geometries.

Usage

```

gm_geometries(
  obj,
  id_cols,
  geometry_cols,
  class_attributes = list(),
  close = FALSE,
  closed_attribute = FALSE
)

```

Arguments

obj	data.frame
id_cols	vector of id columns (either integer or string)
geometry_cols	vector of geometry columns (either integer or string)
class_attributes	class attributes to assign to each geometry
close	logical stating if the last row must equal the first row of each geometry
closed_attribute	logical, if true a 'has_been_closed' attribute is added to each matrix that has been closed.

Value

A list of matrices representing the input object, split by the id column(s).

Examples

```
df <- data.frame(
  id = c(1,1,1,1,1,2,2,2,2,2)
  , x = 1:10
  , y = 10:1
)

gm_geometries(
  df
  , id_cols = c(1L)
  , geometry_cols = c(2L,3L)
)

## Adding a class attribute
gm_geometries(
  df
  , id_cols = c(1)
  , geometry_cols = c(2,3)
  , list( class = "my_line_object" )
)

## Adding a second ID column
df$id1 <- c(1,1,1,2,2,1,1,2,2,3)

gm_geometries(
  df
  , id_cols = c(1,4)
  , geometry_cols = c(2,3)
  , list( class = "my_multiline_object" )
)

## Using character column names
gm_geometries(
  df
  , id_cols = c("id","id1")
  , geometry_cols = c("x","y")
)

## matrix input
m <- as.matrix( df )
gm_geometries(
  m
  , id_cols = c("id","id1")
  , geometry_cols = c("x","y")
)

gm_geometries(
  m
  , id_cols = c(1,4)
  , geometry_cols = c(2,3)
)
```

```
## use close to make the last row the same as the first row
df <- data.frame(
  id = c(1,1,1,1)
  , x = c(1,1,2,2)
  , y = c(1,2,2,1)
)

gm_geometries(
  df
  , id_cols = "id"
  , geometry_cols = c("x","y")
  , close = TRUE
)
```

Index

gm_coordinates, [2](#)
gm_geometries, [4](#)