

Package ‘mapiso’

September 8, 2022

Type Package

Title Create Contour Polygons from Regular Grids

Version 0.2.0

Description Regularly spaced grids containing continuous data are transformed to contour polygons. A grid can be defined by a data.frame (x, y, value), an 'sf' object or a raster from 'terra'.

URL <https://github.com/riatelab/mapiso>

BugReports <https://github.com/riatelab/mapiso/issues/>

Depends R (>= 3.6.0)

Imports sf, isoband

Suggests covr, mapsf, terra, tinytest

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.2.1

NeedsCompilation no

Author Timothée Giraud [cre, aut] (<<https://orcid.org/0000-0002-1932-3323>>)

Maintainer Timothée Giraud <timothee.giraud@cnrs.fr>

Repository CRAN

Date/Publication 2022-09-08 10:02:58 UTC

R topics documented:

mapiso-package	2
mapiso	2

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

mapiso-package	<i>Package description</i>
----------------	----------------------------

Description

Regularly spaced grids containing continuous data are transformed into contour polygons. A grid can be defined by a `data.frame` (`x`, `y`, `value`), an `sf` object or a `terra SpatRaster`.

mapiso	<i>Create Contour Polygons from Regular Grids</i>
--------	---

Description

Regularly spaced grids containing continuous data are transformed into contour polygons. A grid can be defined by a `data.frame` (`x`, `y`, `value`), an `sf` object, a `terra SpatRaster` or `SpatVector`.

Usage

```
mapiso(x, var, breaks, nbreaks = 8, mask, coords, crs)
```

Arguments

<code>x</code>	a <code>data.frame</code> , an <code>sf</code> object or a <code>SpatRaster</code>
<code>var</code>	name of the variable, for <code>data.frames</code> and <code>sf</code> objects only
<code>breaks</code>	list of break values (default to equal interval)
<code>nbreaks</code>	number of classes
<code>mask</code>	an <code>sf</code> object or <code>SpatVector</code> of polygons or multipolygons. <code>mask</code> is used to clip contour polygons
<code>coords</code>	names of the coordinates variables (e.g. <code>c("lon", "lat")</code>), for <code>data.frames</code> only
<code>crs</code>	CRS code (e.g. "epsg:2154"), for <code>data.frames</code> only.

Value

The output is an `sf` object of polygons (or a `SpatVector` if `x` is a `SpatVector`). The `data.frame` contains three fields: `id` (id of each polygon), `isomin` and `isomax` (minimum and maximum breaks of the polygon).

Examples

```
# sf, using a mask
library(sf)
s <- st_read(system.file("gpkg/elevation.gpkg", package = "mapiso"),
  layer = "elevation", quiet = TRUE
)
m <- st_read(system.file("gpkg/elevation.gpkg", package = "mapiso"),
  layer = "com", quiet = TRUE
)
isos <- mapiso(
  x = s, var = "elevation",
  mask = m
)
plot(isos)

# data.frame, using user breaks values
d <- read.csv(system.file("csv/elevation.csv", package = "mapiso"))
bks <- c(98, 100, 150, 200, 250, 300, 350, 400, 412.6)
isod <- mapiso(
  x = d, var = "elevation",
  breaks = bks, coords = c("x", "y"), crs = "epsg:2154"
)
plot(isod)
if (require(mapsf, quietly = TRUE)) {
  mf_map(isod, "isomin", "choro", breaks = bks, leg_title = "Elevation")
}
## Not run:
if (require(terra, quietly = TRUE)) {
  # terra SpatRaster
  r <- rast(system.file("tif/elevation.tif", package = "mapiso"))
  isor <- mapiso(x = r)
  plot(r)
  plot(st_geometry(isor), add = TRUE, col = NA)
  # terra SpatVector
  s_terra <- vect(s)
  m_terra <- vect(m)
  isost <- mapiso(
    x = s_terra, var = "elevation", mask = m_terra
  )
  plot(isost)
}

## End(Not run)
```

Index

mapiso, [2](#)

mapiso-package, [2](#)