Package: mapiso (via r-universe)

September 3, 2024

Type Package

Title Create Contour Polygons from Regular Grids

Version 0.3.0

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 7.2.3

Repository https://riatelab.r-universe.dev

RemoteUrl https://github.com/riatelab/mapiso

RemoteRef HEAD

RemoteSha 245adfdf1aa3f9297311d24941ed9b94ba98c34b

Contents

mapiso-package *Package description*

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 *Create Contour Polygons from Regular Grids*

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

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