Package: mapinsetr (via r-universe)

March 20, 2025

Type Package

Title Create Map Insets

Description Map insets are small zoom-in or zoom-out maps that focus on particular territories. mapinsetr provides a set of functions that helps to create such insets.

Version 0.3.0

License GPL-3

Imports sf, graphics

Suggests mapsf

URL https://github.com/riatelab/mapinsetr/

BugReports https://github.com/riatelab/mapinsetr/issues/

Encoding UTF-8

RoxygenNote 7.3.2

Config/pak/sysreqs libgdal-dev gdal-bin libgeos-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev

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

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

RemoteRef HEAD

RemoteSha 3b90f464894d3c559d9ddb843641239746abbd5b

Contents

`create_mask` *Create a Mask*

Description

Create a mask based on a bounding box (`bbox`), a simple feature collection (`sf`) extent, a simple feature geometry list column (`sfc`) extent or an interactively defined rectancle.

Usage

```
create_mask(bb, prj, interactive = FALSE, add = FALSE)
```

Arguments

`bb`	either a bounding box, a numeric vector (xmin, ymin, xmax, ymax), an sf or sfc object.
`prj`	a CRS string.
`interactive`	define the mask interactively.
`add`	add the mask to the current plot.

Value

An sf object is returned.

Examples

```
library(sf)
nc <- st_read(system.file("shape/nc.shp", package="sf"))
nc <- st_transform(nc, 32119)

plot(st_geometry(nc))
bb <- st_bbox(nc[nc$CNTY_ID %in% c('2030', '1989', '1938'),])
mask <- create_mask(bb = bb, add = TRUE)

plot(st_geometry(nc))
bb <- nc[nc$CNTY_ID %in% c('2030', '1989', '1938'),]
mask <- create_mask(bb = bb, add = TRUE)

plot(st_geometry(nc))
bb <- c(589912, 159757, 694332, 257053)
mask <- create_mask(bb = bb, prj=32119, add = TRUE)
```

inset_rbinder *Rbind sf Objects With Rows and Cols Handling*

Description

Takes a list of sf polygons or multipolygons objects and output a single sf polygon or multipolygon object.

Usage

```
inset_rbinder(l = list())
```

Arguments

l	a list of sf POLYGON or MULTIPOLYGON objects.

Value

An sf polygon or multipolygon object is returned.

Examples

```
library(sf)
nc <- st_read(system.file("shape/nc.shp", package="sf"))
nc <- st_transform(nc, 32119)

plot(st_geometry(nc))
mask1 <- st_buffer(st_centroid(st_geometry(nc[nc$CNTY_ID == 2026,])),dist = 30000)
mask2 <- st_buffer(st_centroid(st_geometry(nc[nc$CNTY_ID == 2016,])),dist = 30000)
plot(st_geometry(mask1), border = "red", lwd = 2, add = TRUE)
plot(st_geometry(mask2), border = "red", lwd = 2, add = TRUE)
inset1 <- move_and_resize(nc, mask1, xy = c(200000, 5000), k = 2)
inset2 <- move_and_resize(nc, mask2, xy = c(200000 + 130000, 5000), k = 2)
plot(st_geometry(inset1), add = TRUE)
plot(st_geometry(inset2), add = TRUE)

nc_insets <- inset_rbinder(l = list(nc, inset1, inset2))
plot(st_geometry(nc_insets))
plot(nc_insets[,9])
```

mapinsetr *mapinsetr Package*

Description

Map insets are small zoom-in or zoom-out maps that focus on particular territories. `mapinsetr` provides a set of functions that helps to create such insets.

Author(s)

Maintainer: Timothée Giraud <timothee.giraud@cnrs.fr> (ORCID)

Authors:

- Nicolas Lambert <nicolas.lambert@cnrs.fr>

See Also

Useful links:

- https://github.com/riatelab/mapinsetr/
- Report bugs at https://github.com/riatelab/mapinsetr/issues/

move_and_resize *Move and Resize an sf Object*

Description

Move and resize a simple feature collection of polygons, multipolygons or points.

Usage

```
move_and_resize(x, mask = NULL, xy, prj, k = 1)
```

Arguments

`x`	an sf POINT, POLYON or MULTIPOLYGON object to resize and move.
`mask`	an sf or sfc POLYGON or MULTIPOLYGON object used to select the area to move an resize.
`xy`	coordinates used to move the inset, bottomleft corner of the inset.
`prj`	CRS string of the output projection of the inset.
`k`	factor used to resize.

Value

An sf object is returned.

Examples

```
library(sf)
nc <- st_read(system.file("shape/nc.shp", package="sf"))
nc <- st_transform(nc, 32119)

plot(st_geometry(nc))
bb <- st_bbox(nc[nc$CNTY_ID == '2030',])
mask <- create_mask(bb = bb, add = TRUE)
inset <- move_and_resize(nc, mask, xy = c(190000, 1000), k = 3)
plot(st_geometry(inset), add = TRUE)

plot(st_geometry(nc))
mask <- st_buffer(st_centroid(st_geometry(nc[nc$CNTY_ID == 2026,])),
                  dist = 30000)
plot(st_geometry(mask), border = "red", lwd = 2, add = TRUE)
inset <- move_and_resize(nc, mask, xy = c(270000, 5000), k = 2.5)
plot(st_geometry(inset), add = TRUE)
```

`m_r` *Move and Resize in Box*

Description

do stuff, all layers must use a cartographic projection, no lon/lat.

Usage

```
m_r(x, mask, y, return_k = FALSE)
```

Arguments

`x`	the layer to cut, resize and move, sf
`mask`	the targeted area in x, sf or sfc
`y`	destination, sf or sfc
`return_k`	return the k factor

Value

a layer

Examples

```
library(sf)
library(mapsf)
nc <- st_read(system.file("shape/nc.shp", package="sf"))
# Ne fonctionne qu'avec 2 fonds projetés
nc <- st_transform(nc, 32119)
# Créer des boites
```

```
y <- st_as_sfc(st_bbox(
  c(xmin = 270000, ymin = 00000,
    xmax = 550000, ymax = 100000),
  crs = 32119))
y <- st_make_grid(y, n = c(3,2))
mf_map(nc)
mf_map(y, add = TRUE)
# plusieurs objets agrandis
inset1 <- m_r(x = nc, mask = nc[1,], y = y[1])
mf_map(nc[1, ], col = 2, add = TRUE)
mf_map(inset1, col = 2, add = TRUE)
# plusieurs objets agrandis
inset2 <- m_r(x = nc, mask = nc[2,], y = y[2])
mf_map(nc[2, ], col = 3, add = TRUE)
mf_map(inset2, col = 3, add = TRUE)
# un seul objet, réduit
inset3 <- m_r(x = nc[nc$CNTY_ID == 2000, ],
              mask = nc[nc$CNTY_ID == 2000, ],
              y = y[3])
mf_map(nc[nc$CNTY_ID == 2000, ], col = 4, add = TRUE)
mf_map(inset3, col = 4, add = TRUE)
# plusieurs objets dans une autre proj
mtq <- mf_get_mtq()
inset4 <- m_r(x = mtq, mask = mtq, y = y[4])
mf_map(inset4, col = 5, add = TRUE)
# bouger des points
pts <- st_as_sf(st_sample(x = nc[1, ], 10))
inset5 <- m_r(x = pts, mask = nc[1,], y = y[1])
mf_map(pts, cex = .2, add=TRUE)
mf_map(inset5, cex = .2, add = TRUE)
# MULTIPOINT
pts <- (st_sample(x = nc[3, ], 30))
pts <- st_as_sf(st_combine(x = pts))
inset6 <- m_r(x = pts, mask = nc[3,], y = y[5])
mf_map(pts, cex = .5, add = TRUE)
mf_map(inset6, cex = .5, add = TRUE)
# MULTILINESTRING
line <- st_cast(nc, 'MULTILINESTRING')
inset7 <- m_r(x = line, mask = line[4,], y = y[6])
mf_map(line[4, ], col = 3, add = TRUE, lwd = 3)
mf_map(inset7, add = TRUE, col = 3, lwd = 3)
```