

Package ‘GADMTools’

June 26, 2019

Type Package

Title Easy Use of 'GADM' Maps

Version 3.7-1

Date 2019-06-25

Description Manipulate, assemble, export <<http://www.gadm.org>> maps. Create 'choropleth', 'isopleth', dots plot, proportional dots, dot-density and more.

Depends R (>= 3.5.0), sp, tidyverse, classInt, sf, rgdal

Imports methods, RColorBrewer, maptools, stringr, raster, rosm, lattice, jsonlite, gridExtra, rgeos, ggmap, ggspatial, ggplot2, dplyr, prettymapr

Suggests knitr, rmarkdown, kableExtra, mapproj, tidyr, testthat

License GPL-3

URL <https://github.com/Epiconcept-Paris/GADMTools>

Encoding UTF-8

VignetteBuilder knitr

NeedsCompilation no

Author Jean Pierre Decorps [aut, cre],
Morgane Vallee [ctb]

Maintainer Jean Pierre Decorps <jp.decorps@epiconcept.fr>

Repository CRAN

Date/Publication 2019-06-26 08:10:10 UTC

R topics documented:

GADMTools-package	2
choropleth	4
classDots	6
Corsica	7
dotDensity	8

dots	9
fast.choropleth	11
GADM36SF	12
gadm_crop	13
gadm_getBackground	14
gadm_getBbox	15
gadm_loadStripped	16
gadm_longTo360	17
gadm_plot	18
gadm_remove	19
gadm_removeBackground	20
gadm_saveStripped	21
gadm_sf_loadCountries	22
gadm_showNorth	23
gadm_showScale	24
gadm_sp_loadCountries	24
gadm_subset	26
gadm_union	27
grid.map	28
isopleth	29
json.choropleth	30
listNames	31
propDots	32
saveAs	33
saveAsStripped	34
strippedExists	35
stripSP	36
vignette	37
Index	39

GADMTools-package *Easy use of GADM shapefiles*

Description

See; <https://gadm.org/>

GADM is a spatial database of the world's administrative boundaries for use in **GIS** and similar software. Administrative areas in this database are countries and lower level subdivisions such as provinces, departments, cantons, etc.

With **GADMTools**, a wrapper for **GADM** shapefiles, you can easily manipulate, assemble, and create subsets of these objects.

GADMTools can use 2 shapefile formats, **SpatialPolygonsDataFrame** and **Simple Features (SF)**, both provided by GADM as .rds files.

NB: the SF format is supported only from version 3.5 of GADMTools.

Details

Package: GADMTools
 Type: Package
 Version: 3.7-1
 Date: 2019-06-25
 License: GPL-3

Author(s)

Jean Pierre Decorps <jp.decorps@epiconcept.fr>
 Maintainer: Jean Pierre Decorps <jp.decorps@epiconcept.fr>

choropleth

Draw a choropleth on selected regions

Description

Drawing a choropleth (colored regions based on data values) with GADMTools is straightforward. You just have to select your shape(s) file(s) with `gadm_loadcountries`, load your data from a csv file for example, and call the `choropleth` function with the right arguments.

Usage

```
choropleth (x, data, value=NULL, breaks = NULL, steps = 5, adm.join=NULL,
           legend = NULL, labels = NULL, palette=NULL,
           title="", subtitle = NULL, caption = NULL)
```

Arguments

x	Object <code>gadm_sf</code> or <code>gadm_sp</code>
data	data.frame - data to plot
value	String - the name of the column in the <code>data.frame</code> we want to plot (eg: an incidence in epidemiology studies)
breaks	Vector of breaks values or a String name of a function from <code>classIntervals</code> (one of "sd", "equal", "pretty", "quantile", "kmeans", "hclust", "bclust", "fisher", or "jenks")
steps	Integer - number of breaks. Default = 5. If <code>breaks</code> is NOT NULL this value is used internally with <code>cut()</code> .
adm.join	String - the name in your dataset joined with the field <code>NAME_X</code> of the map, where X is the level of the administrative boundaries. For instance if the level is about 'Districts' of a country, and your dataset has a field named "Study_Location" containing a list of districts, just do <code>adm.join = "Study_Location"</code> .

legend	String - legend title. Default NULL .
labels	String vector labels for the legend. Default NULL
palette	String - An RColorBrewer palette name or a String vector vector of colors. Default NULL .
title	String - Title of the plot. Default is an empty string.
subtitle	String - subtitle of the plot. Default is NULL .
caption	String - caaption of the plot. Default is NULL .

Details

Since this relase, it's no longer necessary to rename the field of your dataset that is joined with the right field of the map. Just write **adm.join="data_field_to_link"**.

Value

Object ggplot2

Note

Author(s)

Jean Pierre Decorps <jp.decorps@epiconcept.fr>

References

See Also

[classIntervals](#)

Examples

```
library(GADMTools)
data("Corsica")
Cantons <- listNames(Corsica, 4)
pop <- floor(runif(length(Cantons), min=15200, max=23500))
DAT <- data.frame(Cantons, pop)

choropleth(Corsica, DAT,
           adm.join = "Cantons",
           value = "pop",
           breaks = "sd",
           palette="Oranges",
           legend = "Population",
           title="Population Cantons de Corse")
```

`classDots`*Plot dots on a map with values between different fixed classes.*

Description

Plot values as discretized scale circles on a map.

Usage

```
classDots(x, data, color="red", value = NULL, breaks = NULL,  
          steps = 5, labels = NULL, opacity = 0.5, title="",  
          note=NULL, legend = NULL)
```

Arguments

<code>x</code>	Object <code>gadm_sp</code>
<code>data</code>	Object <code>data.frame</code> with columns 'latitude' and 'longitude'
<code>color</code>	a valid color
<code>value</code>	Character Name of a column of the <code>data.frame</code> .
<code>breaks</code>	vector of breaks
<code>steps</code>	unused
<code>labels</code>	vector of labels
<code>opacity</code>	float Background opacity of the filled circles
<code>title</code>	Character The title of the plot
<code>note</code>	Character Add an annotation
<code>legend</code>	Character The title of the legend

Details
---**Value**

Object `ggplot2`

Note
---**Author(s)**

Jean Pierre Decorps <jp.decorps@epiconcept.fr>

References

See Also

Examples

```
library(GADMTools)
data("Corsica")

Corse <- gadm_union(Corsica)
longitude <- runif(6, min=8.74, max = 9.25)
latitude <- runif(6, min=41.7, max = 42.6)
Cases <- runif(6, 25, 112)
DAT <- data.frame(longitude, latitude, Cases)

classDots(Corse, DAT, color="blue", value = "Cases", breaks = NULL,
          steps = 4, labels = NULL, opacity = 0.5, title="",
          note=NULL, legend = NULL)
```

Corsica

Map of Corse (FRA) @ level 4 (Cantons)

Description

This map has been subsetted from the FRA map @ level 4.

Usage

```
data(Corsica)
```

Format

A `gadm_sf` object.

Examples

```
data("Corsica")
listNames(Corsica, 3)
```

dotDensity

*Multivariate Dot-Density maps***Description**

A dot-density map is one way to map aggregated spatial data without some of the distortions inherent in choropleths.

Usage

```
dotDensity(map, data, adm.join = NULL, values = NULL,
           cases.by.dots = 100, dot.size = .25, labels = NULL,
           palette = NULL, title = NULL, subtitle = NULL,
           caption = NULL)
```

Arguments

map	Object gadm_sf
data	data.frame - data to plot
values	String - the names of the columns in the data.frame we want to plot. (eg: number of cases)
cases.by.dots	integer of breaks values
dot.size	numeric - size of dots. Default = 0.25.
adm.join	String - the name in your dataset joined with the field NAME_X of the map, where X is the level of the administrative boundaries. For instance if the level is about 'Districts' of a country, and your dataset has a field named "Study_Location" containing a list of districts, just do adm.join = "Study_Location".
labels	String vector labels for the legend. Default NULL. If NULL values are used as labels
palette	String - An RColorBrewer palette name or a String vector vector of colors. Default NULL .
title	String - title of the plot. Default is NULL
subtitle	String - subtitle of the plot. Default is NULL.
caption	String - caption of the plot. Default is NULL.

Details

Value

Object ggplot2

Note

dotDensity only works with maps loaded with `gadm_sf_loadCountries`

Author(s)

Jean Pierre Decorps <jp.decorps@epiconcept.fr>

References

https://en.wikipedia.org/wiki/Dot_distribution_map

See Also

[classIntervals](#)

Examples

```
library(GADMTools)
data("Corsica")

# Creates test data.frame (fake data) -----
# -----
VAR_1 <- as.integer(runif(n = 43, min = 800, max = 15800))
VAR_2 <- as.integer(runif(n = 43, min = 1000, max = 15800))
VAR_3 <- as.integer(runif(n = 43, min = 1500, max = 15800))
Cantons <- listNames(Corsica, 4)
DF <- data.frame(Cantons, VAR_1, VAR_2, VAR_3, stringsAsFactors = FALSE)

dotDensity(Corsica,
           DF,
           adm.join="Cantons",
           values = c("VAR_1", "VAR_2", "VAR_3"),
           labels = c("H1N1", "H1N2", "H2N2"),
           palette = c("#ffff00", "#ffaa00", "#FF3200"))
```

dots

Plot dots on a map

Description

Plot points on a map with different colors and shapes.

Usage

```
dots(x, points, color="red", size = 8, value = NULL,
     breaks = NULL, steps = 5, palette = NULL, labels = NULL, strate = NULL,
     title="", subtitle = "", caption = "", legend = NULL, note=NULL)
```

Arguments

x	Object gadm_sp or gadm_sf
points	Object data.frame with columns 'latitude' and 'longitude'
color	a valid color
size	integer size of point
value	Character Name of a column in the data.frame. If is not null, colored dots are displayed according to the value.
breaks	vector of breaks
steps	Integer Number of breaks for the value field.
palette	a valid palette
labels	vector of labels
strate	Character name of a column in the data.frame. If is not null, display dots with different shapes according to the value.
title	Character title of the plot
subtitle	Character subtitle of the plot
caption	Character caption of the plot
legend	Character The title of the legend
note	Character Add an annotation

Details

Value**Object** ggplot2**Note**

Author(s)

Jean Pierre Decorps <jp.decorps@epiconcept.fr>

References

See Also[RColorBrewer](#)

Examples

```
library(GADMTools)
data("Corsica")

longitude <- runif(6, min=8.74, max = 9.25)
latitude <- runif(6, min=41.7, max = 42.6)
Cases <- runif(6, 25, 112)
DAT <- data.frame(longitude, latitude, Cases)

dots(Corsica, DAT, color="red", size = 8, value = "Cases")
```

fast.choropleth *Draw a choropleth on selected regions with lattice.*

Description

Drawing a choropleth (colored regions based on data values) with GADMTools is straightforward. You just have to select your shape(s) file(s) with *gadm_loadcountries*, load your data from a csv file for example, and call the *fast.choropleth* function with the right arguments. *fast.choropleth* does not use *ggplot2* but *lattice*, so it is very fast.

Usage

```
fast.choropleth (x, data, value=NULL, breaks = NULL, steps = 5,
  adm.join=NULL, legend = NULL, labels = NULL, palette=NULL,
  title="")
```

Arguments

x	Object <i>gadm_sp</i>
data	data.frame - data to plot
value	String - the name of the column in the data.frame we want to plot (eg: an incidence in epidemiology studies)
breaks	
steps	Integer - number of breaks. Default = 5. If <i>breaks</i> is NOT NULL this value is used internally with <i>cut()</i> .
adm.join	String - the name in GADM spdf dataset which will be joined with a column of the data.
legend	String - legend title. Default NULL .
labels	String vector labels for the legend. Default NULL
palette	String - An RColorBrewer palette name or a String vector vector of colors. Default NULL .
title	String - Title of the plot. Default is an empty string.

Details

—

Value**Object** a lattice plot of class "trellis"**Note**

—

Author(s)

Jean Pierre Decorps <jp.decorps@epiconcept.fr>

References

—

See Also[classIntervals](#)**Examples**

```
# MAP <- gadm_loadCountries("BEL", level = 3, simplify=0.01)
# DAT = read.csv2("BE_chlamydia_incidence.csv")

# DAT <- rename(DAT, NAME_3 = district)

# fast.choropleth(MAP, DAT,
#                 adm.join = "NAME_3",
#                 value = "rate03",
#                 steps = 4,
#                 breaks = "jenks",
#                 palette="Greens",
#                 legend = "Incidence",
#                 title="Chlamydia incidence by Belgian district (2003)")
```

GADM36SF

data.frame of maps provided by gadm_org

Description

Dataset of description of all maps provided by gadm_org. This has been used to generate the vignette GADMTools_ISO_3166-1_alpha-3

Usage

```
data(GADM36SF)
```

Format

A data.frame.

gadm_crop	<i>crop a region to a specific rectangle</i>
-----------	--

Description

crop a region to a specific rectangle

Usage

```
gadm_crop(x, xmin, ymin, xmax, ymax)
```

Arguments

x	gadm_sp or gadm_sf Object containing regions.
xmin	numeric Longitude min
ymin	numeric Latitude min
xmax	numeric Longitude max
ymax	numeric Latitude max

Value

Object gadm_sf or gadm_sp

Author(s)

Jean Pierre Decorps <jp.decorps@epiconcept.fr>

Examples

```
library(GADMTools)
data("Corsica")

area <- gadm_crop(Corsica, xmin=9.3, ymin=42.96, xmax=9.566, ymax=43.02819)
plotmap(area)
```

gadm_getBackground *Gets tiles with 'rosm' from OpenStreetMap*

Description

Load tiles from OpenStreetMap create and save a .tif file with assembled tiles. The bounding box is automatically retrieved from the GADM shapefile passed as argument. The .tif file is stored in the working directory.

Usage

```
gadm_getBackground(x, name, type="osm", clip=TRUE)
gadm.getBackground(x, name, type="osm", clip=TRUE) # deprecated
```

Arguments

x	Object gadm_sf or gadm_sp (region that you want to add a background).
name	character the name of the TIFF file generated by this function. The .tif extension is automatically added.
type	Character type (default "osm") of the map provided by osm.types().
clip	boolean if TRUE (the default), background is clipped by the the external border of the spatial object. If FALSE, spatial object is drawn upper the background using the full bounding box.

Value

Object As input, gadm_sf or gadm_sp

Note

gadm.getBackground() is deprecated, it will be removed in the next release. Please use gadm_getBackground()

Author(s)

Jean Pierre Decorps <jp.decorps@epiconcept.fr>

See Also

[osm.types](#)

Examples

```
# library(GADMTools)
# library(rosm)
# FRA = gadm_loadCountries("FRA", 2, basefile = "./")
# BRE = GADMTools::subset(FRA, level=1, regions=c("Bretagne"))
# BRE2 <- gadm_getBackground(BRE, "BRE", "osm")
# plotmap(BRE2, title = "Map of Bretagne (FRANCE)")
```

gadm_getBbox	<i>get the bounding box of the map</i>
--------------	--

Description

get the bounding box of the map

Usage

```
gadm_getBbox(x)
```

Arguments

x **Object** of class `gadm_sf` or `gadm_sp`

Value

vector of numeric values of:

- **xmin** minimum longitude
- **ymin** minimum latitude
- **xmax** maximum longitude
- **ymax** maximum latitude

Author(s)

Jean Pierre Decorps <jp.decorps@epiconcept.fr>

See Also

[gadm_crop](#)

Examples

```
library(GADMTools)
data("Corsica")

gadm_getBbox(Corsica)
```

gadm_loadStripped *Load one GADM stripped shapefile*

Description

Load one GADM stripped shapefiles from a local path for use with ggplot2.

Usage

```
gadm_loadStripped(name, level, basefile='./')
```

Arguments

name	Character vector of a named region. An ISO-3166-1 code or a custom name. You don't have to specify the suffix (admX) nor the file extension (.rds).
level	Integer the level of the administrative boundaries (0 is the country, higher values equal finer divisions)
basefile	Character vector the path of the directory where shapefiles are stored. Default is "./"

Value

Object gadm_sp with stripped properties == TRUE

ISO-3166-1

See : https://en.wikipedia.org/wiki/ISO_3166-1_alpha-3

"ABW", "AFG", "AGO", "AIA", "ALA", "ALB", "AND", "ANT", "ARE", "ARG", "ARM", "ASM", "ATA", "ATF", "ATG", "AUS", "

Note

Author(s)

Jean Pierre Decorps <jp.decorps@epiconcept.fr>

References

See Also

Examples

```
# library(GADMTools)
# library(sp)
# BE <- gadm_loadStripped('BEL', level=2)
# plotmap(BE)
```

gadm_longTo360	<i>Converts longitudes from -180° - 0° - 180° to 0° - 360°</i>
----------------	--

Description

Converts longitudes of a GADM shapefile to a range of 0° - 360° using the modulo R function.

Usage

```
gadm_longTo360(x)
```

Arguments

x **Object** gadm_sf or gadm_sp.

Value

Object gadm_sp

Note

For gadm_sp maps, the transformation is done only when rendering a graph. The original data are not modified. For gadm_sf maps, the internal geometry is modified.

Author(s)

Jean Pierre Decorps <jp.decorps@epiconcept.fr>

Examples

```
# library(GADMTools)
# MAP <- gadm_sf.loadCountries("FJI", level = 0)
# plotmap(MAP)
# MAP <- gadm_longTo360(MAP)
# plotmap(MAP)
```

gadm_plot	<i>Draw a gadm_sf or gadm_sp object</i>
-----------	---

Description

Draw a gadm_sf or gadm_sp object with ggplot2

Usage

```
gadm_plot(x, title="")  
plotmap(x, title="") # deprecated
```

Arguments

x	Object gadm_sf or gadm_sp
title	String - Title of the plot. Default is an empty string

Value

Object ggplot2

Note

plotmap() is deprecated, it will be removed in the next release

Author(s)

Jean Pierre Decorps <jp.decorps@epiconcept.fr>

Examples

```
library(GADMTools)  
data("Corsica")  
  
gadm_plot(Corsica)
```

gadm_remove	<i>Remove one or more regions from a map</i>
-------------	--

Description

Remove the polygons of one or more regions from a map.

Usage

```
gadm_remove(x, level=NULL, regions=NULL)
gadm.remove(x, level=NULL, regions=NULL) # deprecated
```

Arguments

x	Object gadm_sf or gadm_sp
level	Integer - level from which shapes are removed. If NULL, current level is used.
regions	String - vector of regions to be removed

Value

Object - As input object, gadm_sf or gadm_sp.

Author(s)

Jean Pierre Decorps <jp.decorps@epiconcept.fr>

See Also

[listNames](#)

Examples

```
library(GADMTools)
data("Corsica")

HCorse <- gadm_remove(Corsica, level=2, "Corse-du-Sud")
plotmap(HCorse)
```

`gadm_removeBackground` *Removes the background of a map*

Description

Removes the background previously loaded with `gadm_getBackground`. Original .tif file is not deleted.

Usage

```
gadm_removeBackground(x)
gadm.removeBackground(x) # deprecated
```

Arguments

`x` **Object** `gadm_sp` or `gadm_sf` of the region that you want to remove the background.

Value

Object `gadm_sp` or `gadm_sf`

Author(s)

Jean Pierre Decorps <jp.decorps@epiconcept.fr>

See Also

[gadm_getBackground](#)

Examples

```
# library(GADMTools)
# Loads France @ level 2 (departements)
# FRA <- gadm_sf.loadCountries("FRA", level = 2, basefile = "DATA/")
# FRA <- gadm_getBackground(FRA, name = "FRABGND", clip = FALSE)
# plotmap(FRA)
# FRA <- gadm_removeBackground(FRA)
# plotmap(FRA)
```

gadm_saveStripped *Save a stripped GADM object*

Description

Save a stripped (with stripSP()) GADM object for later use it with ggplot2.

Usage

```
gadm_saveStripped(x, fname, basefile = './')
```

Arguments

x	Object gadm_sp with stripped property == TRUE
fname	String file name of a region. You don't have to specify the suffix (admX) nor the file extension (.rds).
basefile	Character vector the path of the directory where shapefiles are stored. Default is "./"

Value

Boolean TRUE

Note

Author(s)

Jean Pierre Decorps <jp.decorps@epiconcept.fr>

References

See Also

Examples

```
# library(GADMTools)
# library(sp)
# BE <- gadm_loadCountries('BEL', level=2)
# S_BE <- stripSP(BE)
# gadm_saveStripped(S_BE, "BEL")
```

gadm_sf_loadCountries *Load one or more GADM shapefiles*

Description

Load one or more GADM shapefiles as Simple Features (SF) format from a local path or from a remote repository.

Usage

```
gadm_sf_loadCountries(fileNames, level = 0, basefile=".",
                      baseurl=GADM_SF_URL, simplify=NULL)
```

deprecated :

```
gadm_sf.loadCountries(fileNames, level = 0, basefile=".",
                      baseurl=GADM_SF_URL, simplify=NULL)
```

Arguments

fileNames	Character vector of named regions. An ISO-3166-1 code or a custom name. You don't have to specify the suffix (admX) nor the file extension (.rds).
level	Integer the level of the administrative boundaries (0 is the country, higher values equal finer divisions)
basefile	Character vector the path of the directory where shapefiles are stored. Default is "."
baseurl	Character vector The url of GADM files. Default is "https://biogeo.ucdavis.edu/data/gadm3.6/Rsf/"
simplify	Numeric Numerical tolerance value to be used by the Douglas-Peucker algorithm. Higher values use less polygon points (and less memory) and lower values use more polygon points (and more memory). We suggest not going higher than 0.025 in order for intra-country boundaries to align.

Value

Object of class `gadm_sf` (Simple Features wrapper)

Note

See : https://en.wikipedia.org/wiki/ISO_3166-1_alpha-3 for a list of ISO3 codes or take a look on the vignette "GADMTools - ISO 3166-1 alpha-3".

`gadm_sf.loadCountries()` is deprecated, it will be removed in the next release. Please use `gadm_sf_loadCountries()`

Author(s)

Jean Pierre Decorps <jp.decorps@epiconcept.fr>

References

—

See Also

[gadm_sp_loadCountries](#)

Examples

```
# library(GADMTools)
# library(sp)
# Belgium = gadm_sf_loadCountries("BEL", level=2, basefile="./")
# plotmap(Belgium)
```

gadm_showNorth	<i>display a north arrow on a plot</i>
----------------	--

Description

display a north arrow on a plot (ggplot2)

Usage

```
gadm_showNorth(plot, where="br")
```

Arguments

plot	ggplot2
where	character location of the arrow. Can be: <ul style="list-style-type: none">• "tl" - top left• "tr" - top right• "bl" - bottom left• "br" - bottom right (default)

Value

Object ggplot2

Author(s)

Jean Pierre Decorps <jp.decorps@epiconcept.fr>

Examples

```
library(GADMTools)
data("Corsica")

plotmap(Corsica) %>% gadm_showNorth()
```

gadm_showScale *display a scale on a plot*

Description

display a scale for measuring distances on a plot (ggplot2)

Usage

```
gadm_showScale(plot, where="bl")
```

Arguments

plot	ggplot2
where	character location of the scale. Can be: <ul style="list-style-type: none"> • "tl" - top left • "tr" - top right • "bl" - bottom left (default) • "br" - bottom right

Value

Object ggplot2

Author(s)

Jean Pierre Decorps <jp.decorps@epiconcept.fr>

Examples

```
library(GADMTools)
data("Corsica")

plotmap(Corsica) %>% gadm_showScale()
```

gadm_sp_loadCountries *Load one or more GADM shapefiles (SpatialPolygonsDataFrame)*

Description

Load one or more GADM shapefiles as SpatialPolygonsDataFrame from a local path or from a remote repository.

Usage

```
gadm_sp.loadCountries(fileNames, level = 0, basefile=GADM_BASE,
                      baseurl=GADM_URL, simplify=NULL)

# deprecated
gadm_sp.loadCountries(fileNames, level = 0, basefile=GADM_BASE,
                      baseurl=GADM_URL, simplify=NULL)
```

Arguments

fileNames	Character vector of named regions. An ISO-3166-1 code or a custom name. You don't have to specify the suffix (admX) nor the file extension (.rds).
level	Integer the level of the administrative boundaries (0 is the country, higher values equal finer divisions)
basefile	Character vector the path of the directory where shapefiles are stored. Default is <code>"/GADM"</code>
baseurl	Character vector The url of GADM files. Default is <code>"https://biogeo.ucdavis.edu/data/gadm3.6/Rsp/"</code>
simplify	Numeric Numerical tolerance value to be used by the Douglas-Peucker algorithm. Higher values use less polygon points (and less memory) and lower values use more polygon points (and more memory). We suggest not going higher than 0.025 in order for intra-country boundaries to align.

Value

Object `gadm_sp`

ISO-3166-1

See : https://en.wikipedia.org/wiki/ISO_3166-1_alpha-3 or take a look on the vignette "GADMTools - ISO 3166-1 alpha-3"

Note

`gadm_sp.loadCountries()` and `gadm.loadCountries` are deprecated, they will be removed in the next release. Please use `gadm_sp_loadCountries()`

Author(s)

Jean Pierre Decorps <jp.decorps@epiconcept.fr>

References

See Also

[gadm_sf.loadCountries](#)

Examples

```
# library(GADMTools)
#
# Belgium = gadm_sp_loadCountries("BEL", level=2, basefile=".")
# plotmap(Belgium)
```

gadm_subset

Extract regions

Description

With subset you can extract one or more regions from a country at the current level.

Usage

```
gadm_subset(x, level = NULL, regions = NULL, usevar = NULL)
```

```
gadm_subset(x, level = NULL, regions = NULL, usevar = NULL) # deprecated
```

Arguments

x	Object gadm_sf or gadm_sp
level	Integer the level at which the regions are extracted from
regions	character vector of named regions
usevar	character name of an other var of the internal dataset of map

Value

Object As input object, gadm_sf or gadm_sp

Note

gadm_subset() is deprecated, it will be removed in the next release. Please use gadm_subset()

Author(s)

Jean Pierre Decorps <jp.decorps@epiconcept.fr>

See Also

[listNames](#)

Examples

```
library(GADMTools)
data("Corsica")

Calvi <- gadm_subset(Corsica, 4, "Calvi")
plotmap(Calvi)
```

gadm_union	<i>Merges regions</i>
------------	-----------------------

Description

This function merges regions by removing common borders.

Usage

```
gadm_union(x, level = 0, type = "?")

gadm.union(x, level = 0, type = "?") # deprecated
```

Arguments

x	Object gadm_sf or gadm_sp containing regions.
level	integer level @ union is processed. For gadm_sf objects only. For gadm_sp objects, union is processed on the whole map.
type	character alternative name.

Value

Object same as input, gadm_sf or gadm_sp

Note

gadm.union() is deprecated, it will be removed in the next release. Please use gadm_union()

Author(s)

Jean Pierre Decorps <jp.decorps@epiconcept.fr>

Examples

```
library(GADMTools)
data("Corsica")

plotmap(Corsica)

Corse <- gadm_union(Corsica, level=2)
plotmap(Corse)
```

`grid.map`*Arrange maps on a grid*

Description

Allows you to arrange multiple maps into one image. This is useful for showing a country together with its territories in other parts of the world (ex: showing France and Reunion island) or placing two or more countries side by side.

Usage

```
grid.map(left, right, center=NULL, title=NULL)
```

Arguments

<code>left</code>	Object <code>gadm_sp</code>
<code>right</code>	data.frame - data to plot
<code>center</code>	String - an RColorBrewer palette name or a String vector vector of colors. Default NULL .
<code>title</code>	String - plot title. Default is an empty string.

Details

Value

Object `ggplot2`

Note

Author(s)

Jean Pierre Decorps <jp.decorps@epiconcept.fr>

References

See Also

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##-- or do help(data=index) for the standard data sets.
```

isopleth

Draw an isopleth on selected regions

Description

Drawing an isopleth (also known as heat maps) with GADMTools is straightforward. You just have to select your shape(s) file(s) with *gadm_loadcountries*, load your data from a csv file for example, and call the *isopleth* function with the right arguments.

Usage

```
isopleth(x, data, palette=NULL, title="", subtitle = "", caption = "")
```

Arguments

x	Object <i>gadm_sp</i>
data	data.frame - data to plot
palette	String - An RColorBrewer palette name or a String vector vector of colors. Default NULL .
title	String - Plot title. Default is an empty string.
subtitle	String - Plot subtitle. Default is an empty string.
caption	String - Plot caption. Default is an empty string.

Value

Object *ggplot2*

Author(s)

Jean Pierre Decorps <jp.decorps@epiconcept.fr>

Examples

```
library(GADMTools)
data(Corsica)

longitude <- runif(6, min=8.74, max = 9.25)
latitude <- runif(6, min=41.7, max = 42.6)
Cases <- runif(6, 25, 112)
DAT <- data.frame(longitude, latitude, Cases)

isopleth(Corsica, data = DAT, palette = "Blues")
```

json.choropleth *Create a geojson choropleth of selected regions.*

Description

Drawing a choropleth (colored regions based on data values) with GADMTools is straightforward. You just have to select your shape(s) file(s) with *gadm_loadcountries*, load your data from a csv file for example, and call the *json.choropleth* function with the right arguments. *json.choropleth* create a GEOJSON file (output.json) that can be used with Leaflet library.

Usage

```
json.choropleth (x, data, value=NULL, breaks = NULL, steps = 5,
  adm.join=NULL, legend = NULL, labels = NULL, palette=NULL,
  title="")
```

Arguments

x	Object <i>gadm_sp</i>
data	data.frame - data to plot
value	String - the name of the column in the data.frame we want to plot (eg: an incidence in epidemiology studies)
breaks	
steps	Integer - number of breaks. Default = 5. If <i>breaks</i> is NOT NULL this value is used internally with <i>cut()</i> .
adm.join	String - the name in GADM spdf dataset which will be joined with a column of the data.
legend	String - legend title. Default NULL .
labels	String vector labels for the legend. Default NULL
palette	String - An RColorBrewer palette name or a String vector vector of colors. Default NULL .
title	String - Title of the plot. Default is an empty string.

Details

Value

Object a lattice plot of class "trellis"

Note

Author(s)

Jean Pierre Decorps <jp.decorps@epiconcept.fr>

References

See Also

[classIntervals](#)

Examples

```
##---- Should be DIRECTLY executable !! ----  
##-- ==> Define data, use random,  
##--or do help(data=index) for the standard data sets.
```

listNames

List the region names for a specific administrative level

Description

Returns a list of the names associated with the particular administration level.

Usage

```
listNames(x, level = 0)
```

Arguments

x	Object - gadm_sf or gadm_sp
level	Integer - the value of the administration level to list. Attention: only the administrative levels that have been loaded in the loadCountries object can be listed. Names are given in the country's language or English.

Details

Some GADM country maps provide five or more administrative levels.

Value

Character vector of names

Author(s)

Jean Pierre Decorps <jp.decorps@epiconcept.fr>

Examples

```
library(GADMTools)
data("Corsica")
listNames(Corsica, level=3)
listNames(Corsica, level=4)
```

propDots

Plot proportionnal circles (dots) on a map

Description

Plot values as proportionnal circles on a map.

Usage

```
propDots(x, data, value, breaks=NULL, range=NULL,
         labels=NULL, color="red", title="",
         subtitle = "", caption = "", note=NULL)
```

Arguments

x	Object gadm_sf or gadm_sp
data	Object data.frame with columns 'latitude' and 'longitude'
value	Character Name of a column of the data.frame.
breaks	a vector of breaks
range	vector min, max
labels	vector of labels
color	a valid color
title	Character title of the plot
subtitle	Character subtitle of the plot
caption	Character caption of the plot
note	Character A note associated with the plot

Value

Object ggplot2

Author(s)

Jean Pierre Decorps <jp.decorps@epiconcept.fr>

See Also

[classDots](#)

Examples

```
library(GADMTools)
data("Corsica")

longitude <- runif(7, min=8.74, max = 9.25)
latitude <- runif(7, min=41.7, max = 42.6)
Cases <- runif(7, 25, 100)
DAT <- data.frame(longitude, latitude, Cases)

propDots(Corsica, data = DAT, value="Cases",
         breaks=c(0, 25, 50, 75, 100), range = c(25, 100))
```

saveAs

Save your own GADM shapefile as an rds file

Description

Save a GADM shapefile (.rds)

Usage

```
saveAs(x, name = NULL, directory = NULL)
```

Arguments

x	Object - GADMWrapper
name	String - filename
directory	String - path to an alternative directory

Details

If directory is NULL (default), the file is stored in the same directory as specified in basefile parameter of `gadm_loadCountries` or `gt2.loadCountries`

Value

Note

Do not specify the rds extension, it is added automatically.

Author(s)

Jean Pierre Decorps <jp.decorps@epiconcept.fr>

References

—

See Also

—

Examples

```
# library(GADMTools)
# library(sp)
# France = gadm_loadCountries("FRA", level=1, basefile=".")
# Auvergne = subset(France,regions = "Auvergne", level=1)
# saveas(Auvergne, "./AUVERGNE")
# AUV <- gadm_loadCountries("AUVERGNE", level=1, basefile=".")
# plotmap(AUV)
```

saveAsStripped	<i>Strip a gadm_sp object</i>
----------------	-------------------------------

Description

Strip a `gadm_sp` object (with property 'stripped' == FALSE) and save it stripped (with property 'stripped' == TRUE).

Usage

```
saveAsStripped(x, fname, name= NULL, basefile = './')
```

Arguments

<code>x</code>	Object <code>gadm_sp</code> with stripped property == FALSE
<code>fname</code>	String file name of the region to save. You don't have to specify the suffix (admX) nor the file extension (.rds).
<code>name</code>	String the name of the field in spdf, like "NAME_1".
<code>basefile</code>	String the path of the directory where shapefiles are stored. Default is "."

Value

Object `gadm_sp` with stripped property == TRUE

Note

—

Author(s)

Jean Pierre Decorps <jp.decorps@epiconcept.fr>

References

See Also

Examples

```
# library(GADMTools)
# library(sp)
# BE <- gadm_loadCountries('BEL', level=2)
# saveAsStripped(BE, "BEL", level=1)
```

strippedExists	<i>Test if a stripped gadm_sp object exists</i>
----------------	---

Description

Test if a stripped gadm_sp object exists on the file system in the directory 'basefile'

Usage

```
strippedExists(name, level, basefile = './')
```

Arguments

name	Character vector of a named region. An ISO-3166-1 code or a custom name. You don't have to specify the suffix (admX) nor the file extension (.rds).
level	Integer the level of the administrative boundaries (0 is the country, higher values equal finer divisions)
basefile	Character vector the path of the directory where shapefiles are stored. Default is "./"

Value

Boolean TRUE if the file exists, FALSE if not

Note

Author(s)

Jean Pierre Decorps <jp.decorps@epiconcept.fr>

References

—

See Also

—

Examples

```
# library(GADMTools)
# library(sp)
# if (strippedExists('BEL', level = 2) {
#   BE <- gadm_loadStripped("BEL", level=2)
# }
```

stripSP

Strip a gadm_sp object

Description

Strip a gadm_sp object (with property 'stripped' == FALSE) and return a stripped gadm_sp object (with property 'stripped' == TRUE)

Usage

```
stripSP(x, level=NULL)
```

Arguments

x **Object** gadm_sp with property 'stripped' == FALSE
level **Int** admin level to be stripped/extracted. If NULL, the current level is selected

Value

Object gadm_sp with property 'stripped' == TRUE

Note

—

Author(s)

Jean Pierre Decorps <jp.decorps@epiconcept.fr>

References

—

See Also

—

Examples

```
# library(GADMTools)
# library(sp)
# BE <- gadm_loadCountries('BEL', level=2)
# Belgique <- stripSP(BE, level=2)
```

vignette

Create a vignette

Description

Vignette will superimpose a region map over a larger (lower level) map.

Usage

```
vignette(main, region, maincolor = "black",
         regioncolor = "white", mainfill = "grey",
         regionfill = "black",
         mainsize = 1, regionsize = 0.5)
```

Arguments

main	Object gadm_sp
region	Object gadm_sp
maincolor	a valid color
regioncolor	a valid color
mainfill	a valid color
regionfill	a valid color
mainsize	Numeric border size
regionsize	Numeric border size

Details

—

Value

Object ggplot2

Note

—

Author(s)

Jean Pierre Decorps <jp.decorps@epiconcept.fr>

References

See Also

Examples

```
# library(GADMTools)
# library(sp)
# library(ggplot2)
# FR <- gadm_loadCountries("FRA", level=1, basefile=".")
# AU <- subset(FR, regions="Auvergne", level=1)
# vignette(FR, AU)
```

Index

*Topic `\textasciitildedocumentation`

- choropleth, [4](#)
- classDots, [6](#)
- dotDensity, [8](#)
- dots, [9](#)
- fast.choropleth, [11](#)
- gadm_crop, [13](#)
- gadm_getBackground, [14](#)
- gadm_getBbox, [15](#)
- gadm_loadStripped, [16](#)
- gadm_longTo360, [17](#)
- gadm_plot, [18](#)
- gadm_remove, [19](#)
- gadm_removeBackground, [20](#)
- gadm_saveStripped, [21](#)
- gadm_showNorth, [23](#)
- gadm_showScale, [24](#)
- gadm_sp_loadCountries, [24](#)
- gadm_subset, [26](#)
- gadm_union, [27](#)
- grid.map, [28](#)
- isopleth, [29](#)
- json.choropleth, [30](#)
- listNames, [31](#)
- propDots, [32](#)
- saveAs, [33](#)
- saveAsStripped, [34](#)
- strippedExists, [35](#)
- stripSP, [36](#)
- vignette, [37](#)

*Topic `\textasciitildehplot`

- dots, [9](#)
- fast.choropleth, [11](#)
- gadm_plot, [18](#)
- grid.map, [28](#)
- isopleth, [29](#)
- vignette, [37](#)

*Topic `\textasciitildespacial`

- gadm_sf_loadCountries, [22](#)

*Topic `\textasciitildeutilities`

- gadm_getBackground, [14](#)
- gadm_loadStripped, [16](#)
- gadm_longTo360, [17](#)
- gadm_removeBackground, [20](#)
- gadm_saveStripped, [21](#)
- gadm_sp_loadCountries, [24](#)
- gadm_union, [27](#)
- json.choropleth, [30](#)
- listNames, [31](#)
- saveAs, [33](#)
- saveAsStripped, [34](#)
- strippedExists, [35](#)
- stripSP, [36](#)

*Topic `datasets`

- Corsica, [7](#)
- GADM36SF, [12](#)

- choropleth, [4](#)
- classDots, [6](#), [32](#)
- classIntervals, [5](#), [9](#), [12](#), [31](#)
- Corsica, [7](#)

- dotDensity, [8](#)
- dots, [9](#)

- fast.choropleth, [11](#)

- gadm.getBackground
 - (gadm_getBackground), [14](#)
- gadm.remove(gadm_remove), [19](#)
- gadm.removeBackground
 - (gadm_removeBackground), [20](#)
- gadm.subset(gadm_subset), [26](#)
- gadm.union(gadm_union), [27](#)
- GADM36SF, [12](#)
- gadm_crop, [13](#), [15](#)
- gadm_getBackground, [14](#), [20](#)
- gadm_getBbox, [15](#)
- gadm_loadStripped, [16](#)

`gadm_longTo360`, 17
`gadm_plot`, 18
`gadm_remove`, 19
`gadm_removeBackground`, 20
`gadm_saveStripped`, 21
`gadm_sf.loadCountries`, 25
`gadm_sf.loadCountries`
 (`gadm_sf_loadCountries`), 22
`gadm_sf_loadCountries`, 22
`gadm_showNorth`, 23
`gadm_showScale`, 24
`gadm_sp.loadCountries`
 (`gadm_sp_loadCountries`), 24
`gadm_sp_loadCountries`, 23, 24
`gadm_subset`, 26
`gadm_union`, 27
GADMTools (GADMTools-package), 2
GADMTools-package, 2
`grid.map`, 28

`isopleth`, 29

`json.choropleth`, 30

`listNames`, 19, 26, 31

`osm.types`, 14

`plotmap (gadm_plot)`, 18
`propDots`, 32

`RColorBrewer`, 10

`saveAs`, 33
`saveas (saveAs)`, 33
`saveAsStripped`, 34
`strippedExists`, 35
`stripSP`, 36

`vignette`, 37