GADMTools - Manipulating Shapefiles

Jean Pierre Decorps - Epiconcept

2019-02-03

Epiconcept is made up of a team of doctors, epidemiologists, data scientists and digital specialists. For more than 20 years, Epiconcept has been contributing to the improvement of public health programs by providing software, epidemiological studies, counseling, evaluation and training to better prevent, detect and treat people.

Epiconcept delivers software and services in the following areas :

- Software for managing public health programs,
- Secure cloud solutions for health data collection, reporting and processing,
- The implementation of research projects on measuring the effectiveness and impact of vaccines,
- Services in the field of epidemiology (protocols, analyzes, training, etc.),
- Expertise in data analysis,
- Counseling, coaching and assistance to project owners for public health programs,
- Training (short introductory modules, training through long-term practice).

To achieve such goals Epiconcept :

- Recognized research organization,
- Certified datacenter for hosting personal health data,
- Training organisation.

Epiconcept relies on :

- Its expertise in epidemiology
- Its IT expertise,
- Ethical values rooted in practice (responsibility and quality of services, data security and confidentiality, scientific independence, etc.),
- Capabilities to answer and anticipate tomorrow's challenges (Research - evaluation, e-health, Big Data, IoT, etc.),
- A desire to build long-term relationships with its clients and partners.

Its current customers and partners include some of the greatest names in the world such as : Santé Publique France (and many public health organizations around the world), WHO, eCDC, AFD, MSF, World Bank, etc.

What is GADM?

GADM, the Database of Global Administrative Areas, is a high-resolution database of country administrative areas, with a goal of “all countries, at all levels, at any time period. The database has a few export formats, including shapefiles that are used in most common GIS applications.[2] Files formatted for the programming language R are also available, allowing the easy creation of descriptive data plots that include geographical maps. Although it is a public database, GADM has a higher spatial resolution than other free databases and also higher than commercial software such as ArcGIS. GADM is not freely available for commercial use. The GADM project created the spatial data for many countries from spatial databases provided by national governments, NGO, and/or from maps and lists of names available on the Internet (e.g. from Wikipedia).

The GADM website and data repository is hosted at UC Davis in the Hijmans Lab. The Hijman lab is run by Robert Hijmans an Environmental Science and Policy faculty member in the Geography Graduate Group. [source Wikipedia - https://en.wikipedia.org/wiki/GADM]

What is GADMTools?

GADMTools is an R package to manipulate shapefiles from GADM and to make geo-statistical representations easily.

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

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

Manipulating shapefiles

functions

SpatialPolygons	Simple Features	Description
gadm.loadCountries		[deprecated] use *gadm.sp.loadCountries* function instead
gadm.sp.loadCountries	gadm.sf.loadCountries	downloads or loads one or more shapefiles
gadm.getBackground	gadm.getBackground	Gets tiles with ‘rosm’ from OpenStreetMap
gadm.loadStripped		Load a GADM stripped shapefile
gadm.remove	gadm.remove	Removes one or more regions from a map in a GADMWrapper/GT2 object
gadm.removeBackground	gadm.removeBackground	Removes the background of a map
gadm.saveStripped		Save a stripped GADM object
gadm.subset	gadm.subset	Extract regions. “subset” does not work since release 3.5-1
gadm.union	gadm.union	Merges regions
listNames	listNames	List the region names for an administrative level
saveAs	saveAs	Save your own GADM shapefile as a .rds file
stripSP		Strip a GADMWrapper object

Format SP : gadm.sp.loadCountries() / gadm.load.countries

This is the main function of GADMTools, with it, you can load or download one or more shapefiles. If you load many shapefiles, the function assembles the shapefiles into one.

The old function *gadm.loadCountries* is deprecated.

gadm.sp.loadCountries(

```
  fileNames,

  level = 0,

  basefile=GADM_BASE,

  baseurl=GADM_URL,

  simplify=NULL

)
```

Parameter	Description
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** - the path of the directory where shapefiles are stored. Default is "./GADM"
baseurl	**Character** - the url of GADM files. Default is http://biogeo.ucdavis.edu/data/gadm2.8/rds/
simplify	**Numeric** numerical tolerance value to be used by the Douglas-Peuker 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 0.01 in order for intra-country boundaries to align.

Return: Object *GADMWrapper*

ISO3 CODES

ABW	AFG	AGO	AIA	ALA	ALB	AND	ANT	ARE	ARG
ARM	ASM	ATA	ATF	ATG	AUS	AUT	AZE	BDI	BEL
BEN	BFA	BGD	BGR	BHR	BHS	BIH	BLM	BLR	BLZ
BMU	BOL	BRA	BRB	BRN	BTN	BVT	BWA	CAF	CAN
CCK	CHE	CHL	CHN	CIV	CMR	COD	COG	COK	COL
COM	CPV	CRI	CUB	CXR	CYM	CYP	CZE	DEU	DJI
DMA	DNK	DOM	DZA	ECU	EGY	ERI	ESH	ESP	EST
ETH	FIN	FJI	FLK	FRA	FRO	FSM	GAB	GBR	GEO
GGY	GHA	GIB	GIN	GLP	GMB	GNB	GNQ	GRC	GRD
GRL	GTM	GUF	GUM	GUY	HKG	HMD	HND	HRV	HTI
HUN	IDN	IMN	IND	IOT	IRL	IRN	IRQ	ISL	ISR
ITA	JAM	JEY	JOR	JPN	KAZ	KEN	KGZ	KHM	KIR
KNA	KOR	KWT	LAO	LBN	LBR	LBY	LCA	LIE	LKA
LSO	LTU	LUX	LVA	MAC	MAF	MAR	MCO	MDA	MDG
MDV	MEX	MHL	MKD	MLI	MLT	MMR	MNE	MNG	MNP
MOZ	MRT	MSR	MTQ	MUS	MWI	MYS	MYT	NAM	NCL
NER	NFK	NGA	NIC	NIU	NLD	NOR	NPL	NRU	NZL
OMN	PAK	PAN	PCN	PER	PHL	PLW	PNG	POL	PRI
PRK	PRT	PRY	PSE	PYF	QAT	REU	ROU	RUS	RWA
SAU	SDN	SEN	SGP	SGS	SHN	SJM	SLB	SLE	SLV
SMR	SOM	SPM	SRB	STP	SUR	SVK	SVN	SWE	SWZ
SYC	SYR	TCA	TCD	TGO	THA	TJK	TKL	TKM	TLS
TON	TTO	TUN	TUR	TUV	TWN	TZA	UGA	UKR	UMI
URY	USA	UZB	VAT	VCT	VEN	VGB	VIR	VNM	VUT
WLF	WSM	YEM	ZAF	ZMB	ZWE				

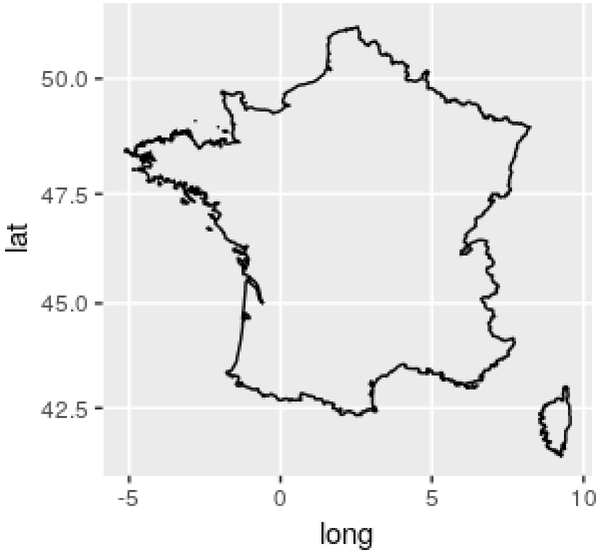

Figure 1: loading a single country (France) @ level = 0

Loading a shapefile

```
library(GADMTools)
library(sp)

# Loading country border (level=0 [default])
# -----------------------------------------------------------------
map <- gadm.sp.loadCountries("FRA", basefile = "./")
plotmap(map)
```

Format SF : gadm.sf.loadCountries()

This is the main function of GADMTools, with it, you can load or download one or more shapefiles. If you load many shapefiles, the function assembles the shapefiles into one.

gadm.sf.loadCountries(

```
  fileNames,

  level = 0,

  basefile=GADM_BASE,

  baseurl=GADM_URL,

)
```

Parameter	Description
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** - the path of the directory where shapefiles are stored. Default is "./GADM"
baseurl	**Character** - the url of GADM files. Default is http://biogeo.ucdavis.edu/data/gadm2.8/rds/

NB: parameter simplify is not yet implemented for SF format

Return: Object *GT2*

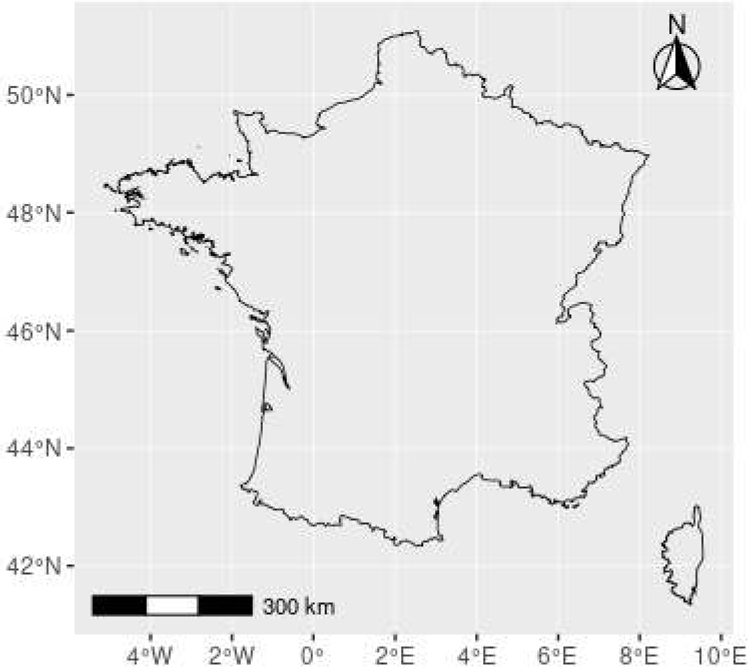

Figure 2: loading a single country (France) @ level = 0

Loading a shapefile

```r
library(GADMTools)

# Loading country border (level=0 [default])
# ---------------------------------------------------------------
map <- gadm.sf.loadCountries("FRA", basefile = "./")
plotmap(map)
```

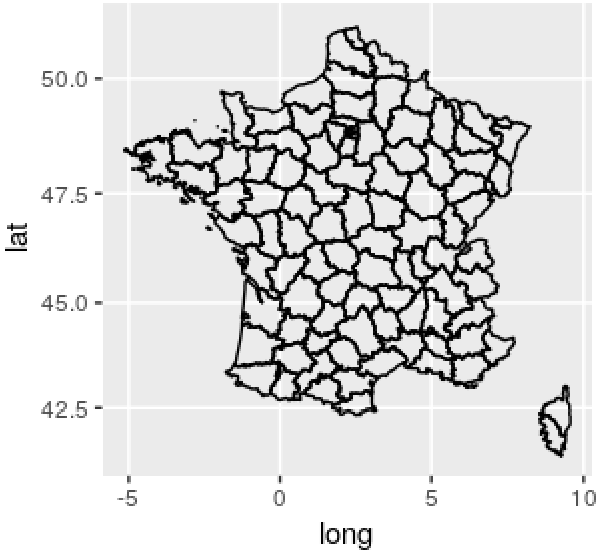

Figure 3: loading regions of a country (France) @ level = 2

Loading an administrative level

```
library(GADMTools)
library(sp)

# Loading regions @ level = 2])
# -----------------------------------------------------------------
map <- gadm.sp.loadCountries(c("FRA"), level=2, basefile = "./")
plotmap(map)
```

NB: you can use *gadm.sf.loadCountries* instead of *gadm.sp.loadCountries*

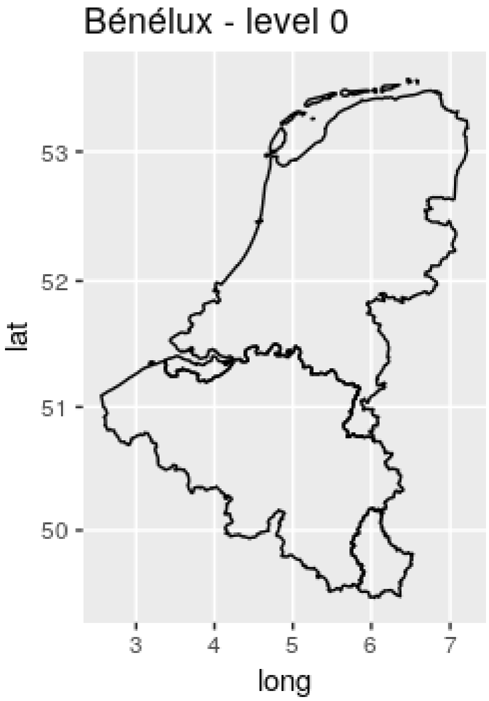

Figure 4: Benelux = Belgium + Luxembourg + Netherlands @ level = 0

Assembling many shapefiles

```
library(GADMTools)
library(sp)

# Assemble administrative boundaries (country level = 0)
# ------------------------------------------------------------------
map <- gadm.sp.loadCountries(c("BEL","LUX","NLD"),  basefile = "./")
plotmap(map, title="Bénélux")
```

NB: you can use *gadm.sf.loadCountries* instead of *gadm.sp.loadCountries*

Extracting regions

In order to extract some regions of a map we need to know them. The listNames function allows this. The subset function is then used to extract the desired regions.

listNames()

listNames(

```
    x,

    level = 0

   )
```

Parameter	Description
x	**Object** - *GADMWrapper* or *GT2*
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.

subset()

gadm.subset(

```
  x,

  level = NULL,

  regions = NULL,

  usevar = NULL

 )
```

Parameter	Description
x	**Object** *GADMWrapper* or *GT2*
level	**Integer** the level at which the regions are extracted from
regions	**String vector** of named regions
usevar	**String vector** name of an other var of spdf@data @ an other level

Example

```
# Extract some regions of a map
# ----------------------------------------------------------------
library(GADMTools)
library(sp)

FR = gadm.sp.loadCountries("FRA", level=2, basefile = "./")
listNames(FR, level=2)
AV = gadm.subset(FR, regions=c("Allier", "Cantal",
            "Haute-Loire","Puy-de-Dôme"))
plotmap(AV)
```

[1] "Bas-Rhin"	"Haut-Rhin"	"Dordogne"	"Gironde"
[5] "Landes"	"Lot-et-Garonne"	"Pyrénées-Atlantiques"	"Allier"
[9] "Cantal"	"Haute-Loire"	"Puy-de-Dôme"	"Essonne"
[13] "Hauts-de-Seine"	"Paris"	"Seine-et-Marne"	"Seine-Saint-Denis"
[17] "Val-d'Oise"	"Val-de-Marne"	"Yvelines"	"Calvados"
[21] "Manche"	"Orne"	"Côte-d'Or"	"Nièvre"
[25] "Saône-et-Loire"	"Yonne"	"Côtes-d'Armor"	"Finistère"
[29] "Ille-et-Vilaine"	"Morbihan"	"Cher"	"Eure-et-Loir"
[33] "Indre-et-Loire"	"Indre"	"Loir-et-Cher"	"Loiret"
[37] "Ardennes"	"Aube"	"Haute-Marne"	"Marne"
[41] "Corse-du-Sud"	"Haute-Corse"	"Doubs"	"Haute-Saône"
[45] "Jura"	"Territoire de Belfort"	"Eure"	"Seine-Maritime"
[49] "Aude"	"Gard"	"Hérault"	"Lozère"
[53] "Pyrénées-Orientales"	"Corrèze"	"Creuse"	"Haute-Vienne"
[57] "Meurthe-et-Moselle"	"Meuse"	"Moselle"	"Vosges"
[61] "Ariège"	"Aveyron"	"Gers"	"Haute-Garonne"
[65] "Hautes-Pyrénées"	"Lot"	"Tarn-et-Garonne"	"Tarn"
[69] "Nord"	"Pas-de-Calais"	"Loire-Atlantique"	"Maine-et-Loire"
[73] "Mayenne"	"Sarthe"	"Vendée"	"Aisne"
[77] "Oise"	"Somme"	"Charente-Maritime"	"Charente"
[81] "Deux-Sèvres"	"Vienne"	"Alpes-de-Haute-Provence"	"Alpes-Maritimes"
[85] "Bouches-du-Rhône"	"Hautes-Alpes"	"Var"	"Vaucluse"
[89] "Ain"	"Ardèche"	"Drôme"	"Haute-Savoie"
[93] "Isère"	"Loire"	"Rhône"	"Savoie"

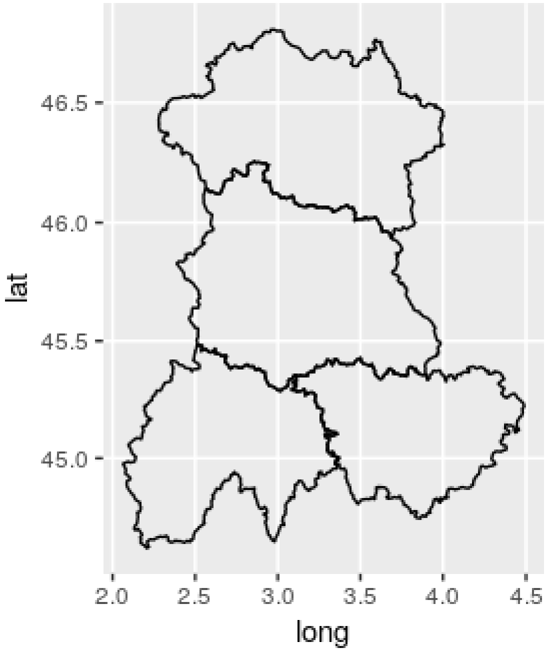

Figure 5: extracting (subset) some departments in France @ level = 2

Merging regions

gadm.union()

gadm.union(x)

Parameter	Description
x	**Object** *GADMWrapper* or *GT2*

Example

```
library(GADMTools)
library(sp)

FR = gadm.loadCountries("FRA", level=2, basefile = "./")
AV = gadm.subset(FR, regions=c("Allier", "Cantal",
             "Haute-Loire","Puy-de-Dôme"))
AV <- gadm.union(AV)
plotmap(AV)
```

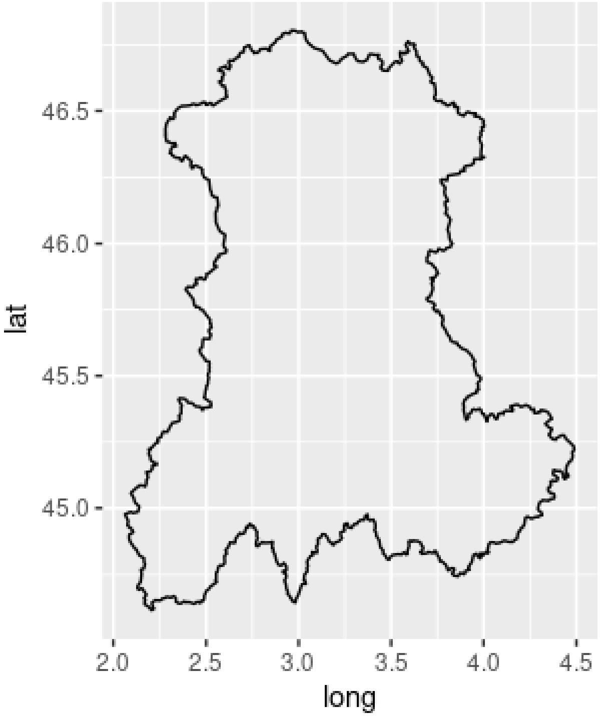

Figure 6: merging 4 departments of Auvergne @ level = 2

Converting longitudes to 0 - 360

gadm.longTo360()

gadm.longTo360(x)

Parameter	Description
x	**Object** GADMWrapper

Example

```
library(GADMTools)
library(GADMTools)
FJI = gadm.sp.loadCountries("FJI", 1, basefile = "./")
# Fig. 6
plotmap(FJI, title = "Fidji Island with bad coordinates")
```

```
FJI = gadm.longTo360(FJI)
# Fig. 7
plotmap(FJI, title = "Fidji Island with 0 - 360 coordinates")
```

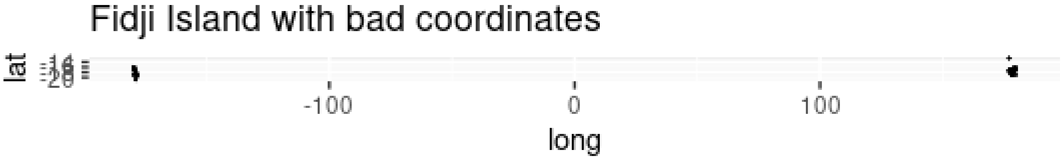

Figure 7: BAD

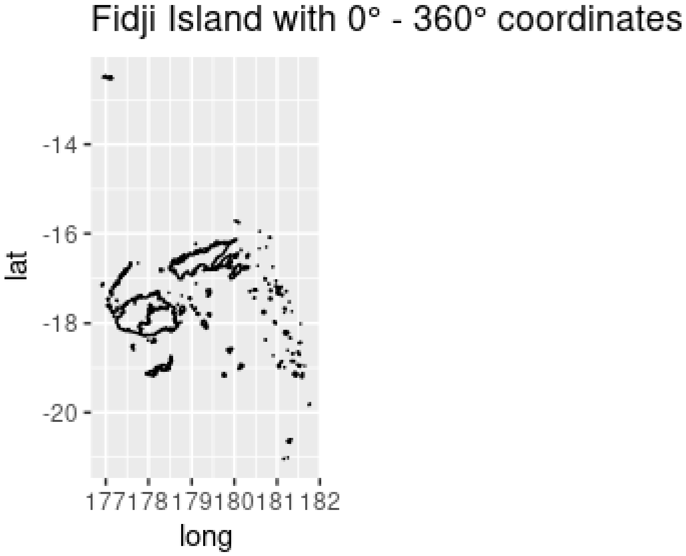

Figure 8: GOOD

Removing regions

remove()

remove(

```
x,

level=NULL,

regions=NULL
```

)

Parameter	Description
x	**Object** *GADMWrapper* or *GT2*
level	**Integer** - level from which shapes are removed. If NULL, curent level is used.
regions	**String vector** of regions to be removed

Example

```r
library(GADMTools)
library(sp)

FR = gadm.loadCountries("FRA", level=1, basefile = "./")
plotmap(FR)
listNames(FR, level=1)
FR2 = remove(FR, level = 1, regions = c("Grand Est"))
plotmap(FR2)
```

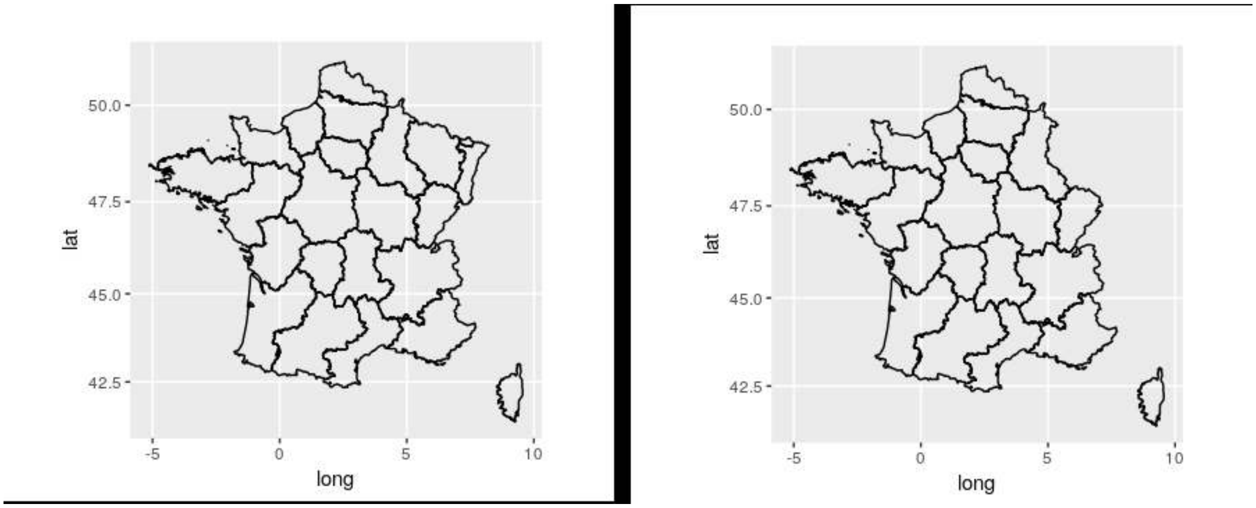

Figure 9: removing 2 regions from France @ level = 1

```
[1] “Alsace”                 “Aquitaine”           “Auvergne”
[4] “Île-de-France”          “Basse-Normandie”     “Bourgogne”
[7] “Bretagne”               “Centre”              “Champagne-Ardenne”
[10] “Corse”                 “Franche-Comté”       “Haute-Normandie”
[13] “Languedoc-Roussillon”  “Limousin”            “Lorraine”
[16] “Midi-Pyrénées”         “Nord-Pas-de-Calais”  “Pays de la Loire”
[19] “Picardie”              “Poitou-Charentes”    “Provence-Alpes-Côte d’Azur”
[22] “Rhône-Alpes”
```

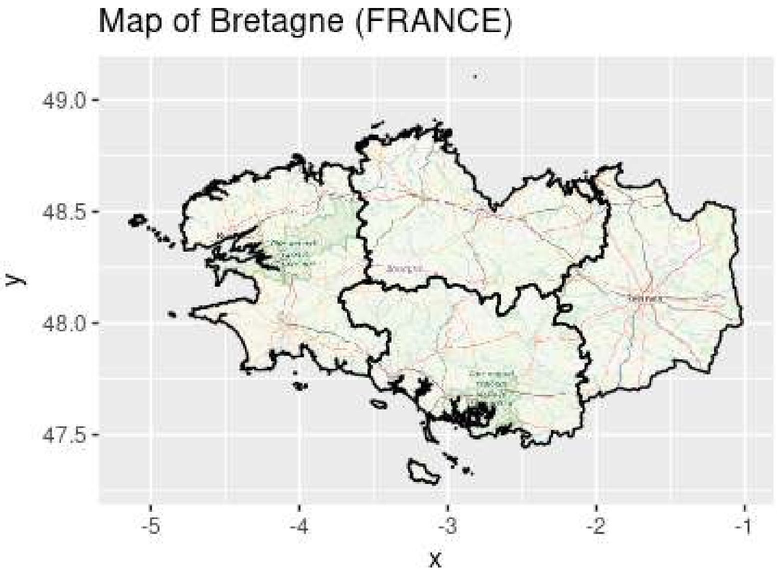

Figure 10: map of Bretagne with background from OSM @ level = 2

Adding a background image from OpenstreetMap

gadm.getBackground()

gadm.getBackground(x, name, type = "osm", clip = TRUE)

Parameter	Description
x	**Object** *GADMWrapper* or *GT2* - the region that you want to add a background.
name	**Object** the name of the TIFF file withhout extension
type	**Character** type (default "osm') of the map provided by *osm.types()*.
clip	**Character** if TRUE (the default), background is clipped by the the external border of the spatial object.

Examples

```
library(GADMTools)
library(rosm)
FRA = gadm.sp.loadCountries("FRA", 2, basefile = "./")
BRE = gadm.subset(FRA, level=1, regions=c("Bretagne"))
BRE2 <- gadm.getBackground(BRE, "BRE", "osm")
plotmap(BRE2, title = "Map of Bretagne (FRANCE)")
```

Remove a background previously loaded with gadm.getBackground

gadm.removeBackground()

gadm.removeBackground(x)

Parameter	Description
x	**Object** *GADMWrapper* or *GT2*