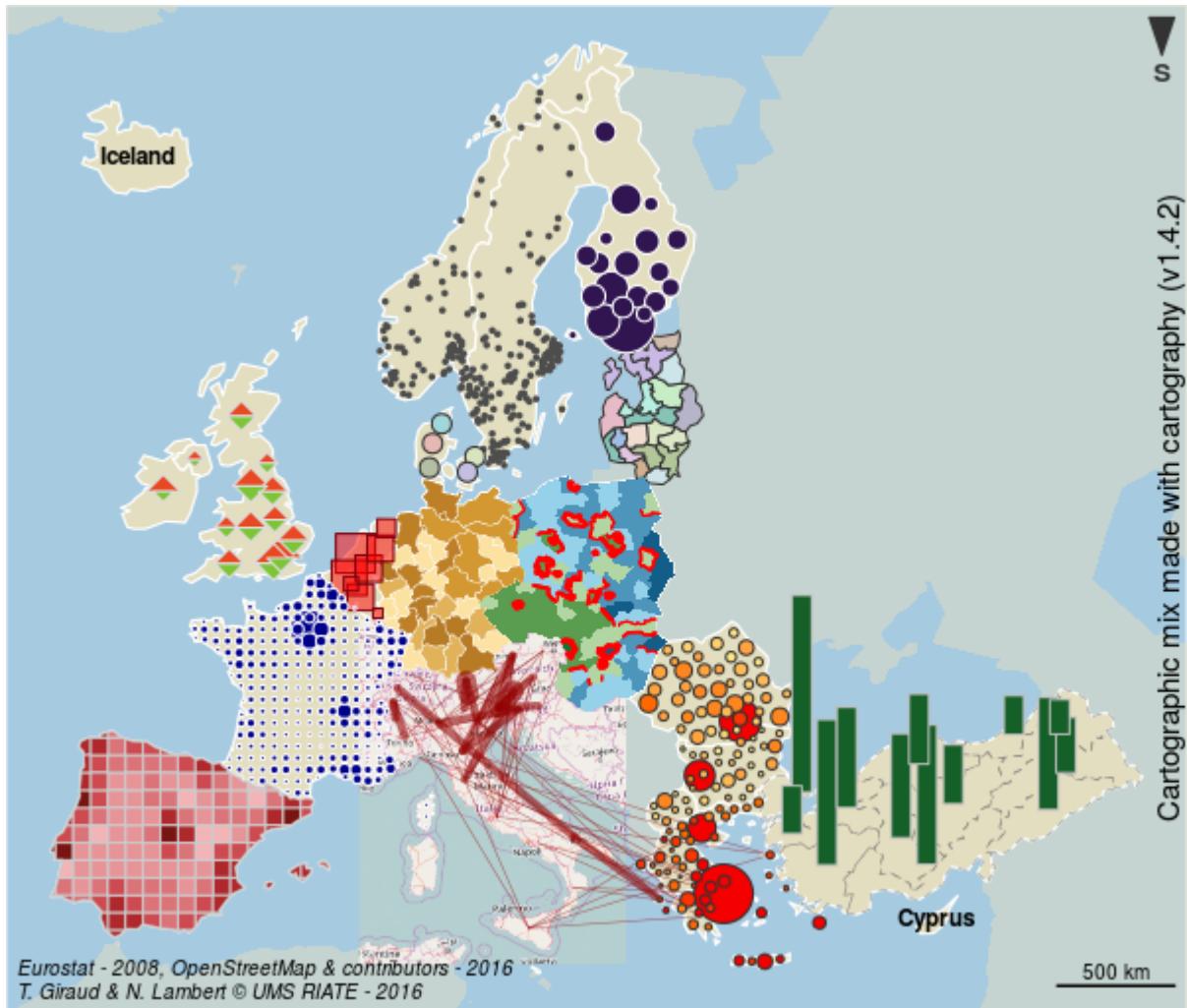


# Cartographie avec R : le package cartography

T. Giraud

16 juin 2017

- 1 Les fonctionnalités spatiales de R
  - 1.1 Les indispensables
    - 1.1.1 Import / Export
    - 1.1.2 Manipulation et affichage
    - 1.1.3 Géotraitements
  - 1.2 Le futur : le package sf
    - 1.2.1 Import / Export
    - 1.2.2 Manipulation et affichage
    - 1.2.3 Géotraitements
- 2 Le package cartography
  - 2.1 Installation
  - 2.2 Utilisation
    - 2.2.1 Cartes choroplèthes
    - 2.2.2 Cartes en symboles proportionnels
    - 2.2.3 Cartes en symboles proportionnels colorés
    - 2.2.4 Cartes de flux
    - 2.2.5 Discontinuités
    - 2.2.6 Carroyages
    - 2.2.7 Cartes lissées
- 3 Ressources
  - 3.1 *Package cartography*
  - 3.2 *Package sf*
  - 3.3 La cartographie



SEMIN-R (<http://rug.mnhn.fr/semin-r/>)

Museum National d'Histoire Naturelle | Vendredi 16 juin 2017

Timothée Giraud

UMS RIATE (<http://riate.cnrs.fr>)

<http://rgeomatic.hypotheses.org/> (<http://rgeomatic.hypotheses.org/>)

# 1 Les fonctionnalités spatiales de R

## 1.1 Les indispensables

### 1.1.1 Import / Export

`rgdal` est une interface entre R et les librairies GDAL (Geospatial Data Abstraction Library (<http://www.gdal.org/>)) et PROJ4 (<https://github.com/OSGeo/proj.4>).

```
library("rgdal")
nuts3 <- readOGR(dsn = "data", layer = "nuts3")
```

```
OGR data source with driver: ESRI Shapefile
Source: "data", layer: "nuts3"
with 1448 features
It has 7 fields
```

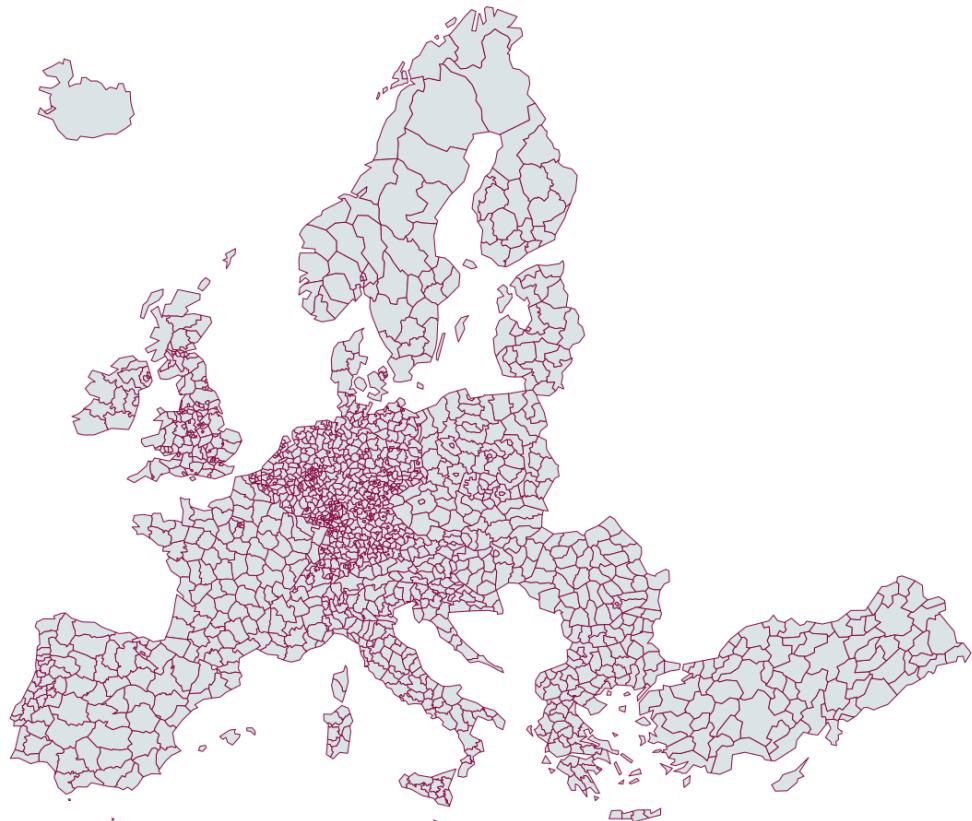
## 1.1.2 Manipulation et affichage

sp fournit des classes et des méthodes pour les données spatiales dans R.

```
library("sp")
plot(nuts3)
```



```
plot(nuts3, col = "#DAE3E6", border = "#8A0641", lwd = 0.5)
```



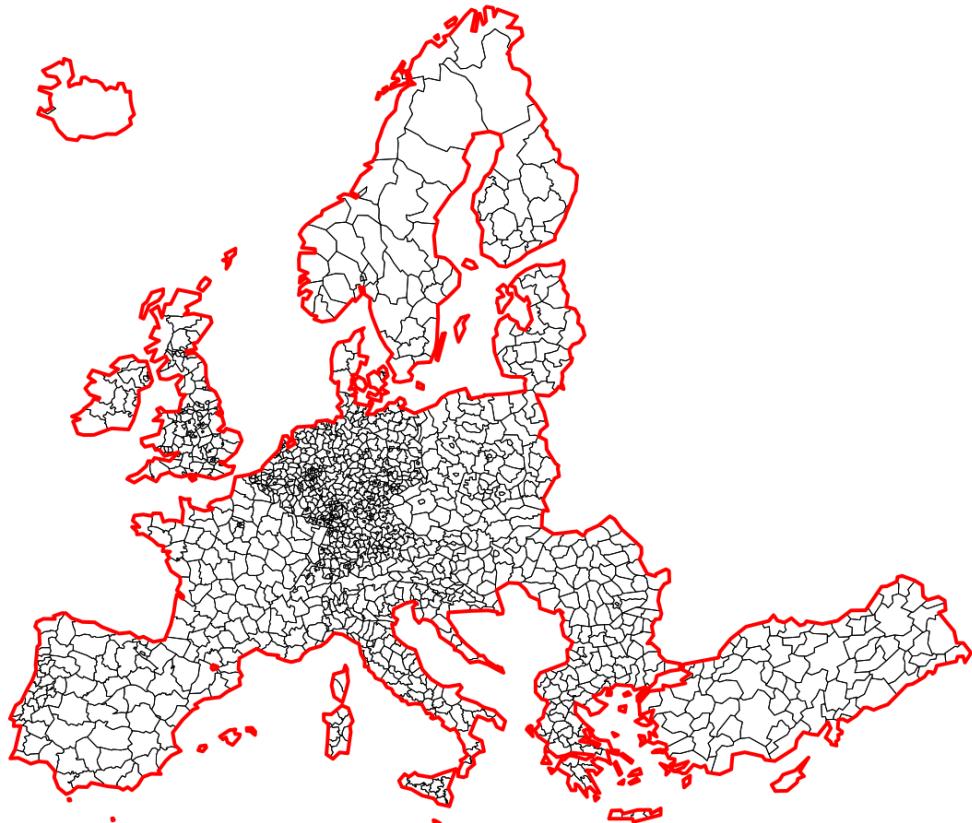
### 1.1.3 Géotraitements

`rgeos` donne accès à la librairie d'opérations spatiales GEOS (Geometry Engine - Open Source (<http://trac.osgeo.org/geos/>)) qui permet notamment d'effectuer les géotraitements suivants :

- Area / Perimeter
- Distances
- **Dissolve**
- **Buffer**
- Overlap / intersect / difference
- Contains / within
- Union
- ...

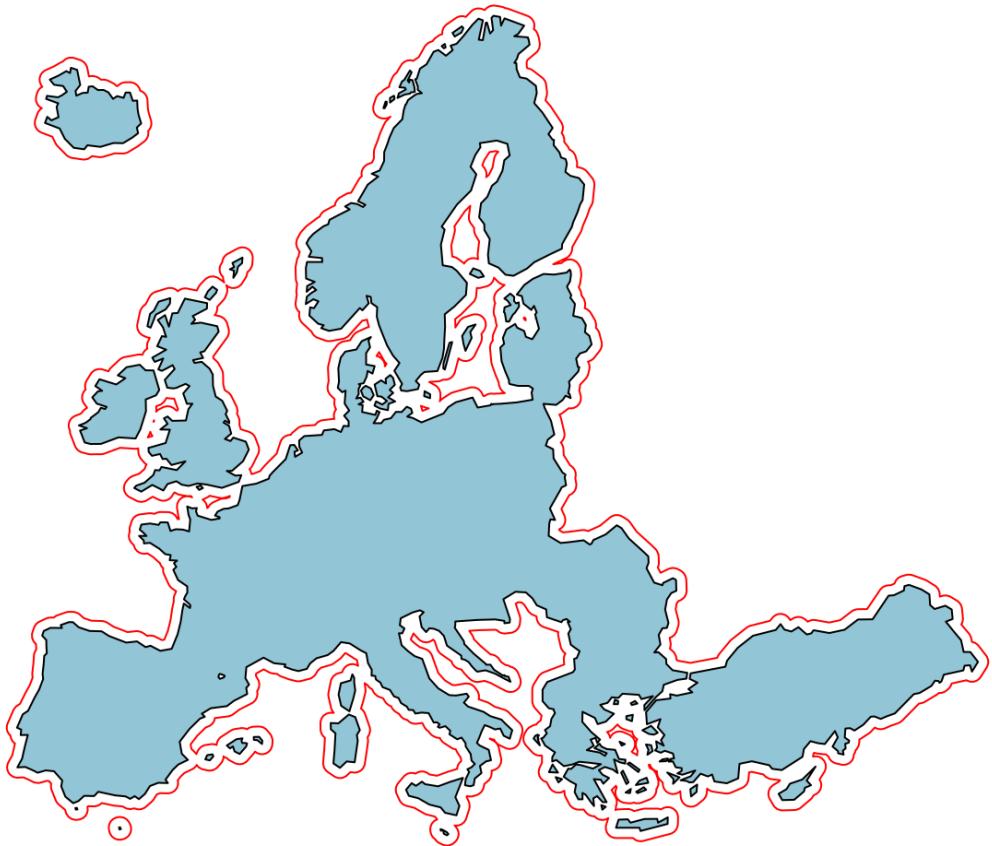
#### 1.1.3.1 Agrégation des polygones / dissolve

```
library("rgeos")
europe <- gUnaryUnion(sppgeom = nuts3)
plot(nuts3, lwd = 0.5)
plot(europe, lwd = 2, border = "red", add=T)
```



#### 1.1.3.2 Crédation de zones tampons / buffer

```
library("rgeos")
europeBuffer <- gBuffer(spgeom = europe, width = 50000)
plot(europe, col = "#92C5D6")
plot(europeBuffer, add = T, border = "red")
```



## 1.2 Le futur : le package sf

- Première *release* : 31 octobre 2016
- Auteur principal et *maintainer* : Edzer Pebesma (<https://github.com/edzer>)
- Financement :  **consortium** (<https://www.r-consortium.org/projects/awarded-projects>)

Avec **sp** et **rgdal** :

```
library('sp')
library('rgdal')
nuts3 <- readOGR(dsn = "data", layer = "nuts3", verbose = FALSE)
str(nuts3[1:3,])
```

```

Formal class 'SpatialPolygonsDataFrame' [package "sp"] with 5 slots
..@ data      :'data.frame': 3 obs. of 7 variables:
.. ..$ id       : Factor w/ 1448 levels "AT111","AT112",...: 1 2 3
.. ..$ birth_2008: num [1:3] 272 1195 748
.. ..$ death_2008: num [1:3] 445 1480 1142
.. ..$ gdppps1999: num [1:3] 509 2262 1368
.. ..$ gdppps2008: num [1:3] 641 3272 1783
.. ..$ pop1999   : num [1:3] 39149 137469 100114
.. ..$ pop2008   : num [1:3] 37452 146383 97350
..@ polygons  :List of 3
.. ..$ :Formal class 'Polygons' [package "sp"] with 5 slots
.. .. . . . @ Polygons :List of 1
.. .. . . . . $ :Formal class 'Polygon' [package "sp"] with 5 slots
.. .. . . . . . @ labpt   : num [1:2] 4806396 2728404
.. .. . . . . . @ area    : num 8.58e+08
.. .. . . . . . @ hole    : logi FALSE
.. .. . . . . . @ ringDir: int 1
.. .. . . . . . @ coords  : num [1:10, 1:2] 4819333 4824015 4808360 4
802891 4797862 ...
.. .. . . . . . @ plotOrder: int 1
.. .. . . . . . @ labpt   : num [1:2] 4806396 2728404
.. .. . . . . . @ ID      : chr "0"
.. .. . . . . . @ area    : num 8.58e+08
.. ..$ :Formal class 'Polygons' [package "sp"] with 5 slots
.. .. . . . @ Polygons :List of 1
.. .. . . . . $ :Formal class 'Polygon' [package "sp"] with 5 slots
.. .. . . . . . @ labpt   : num [1:2] 4827772 2765414
.. .. . . . . . @ area    : num 1.81e+09
.. .. . . . . . @ hole    : logi FALSE
.. .. . . . . . @ ringDir: int 1
.. .. . . . . . @ coords  : num [1:9, 1:2] 4819333 4806425 4800667 47
93126 4799329 ...
.. .. . . . . . @ plotOrder: int 1
.. .. . . . . . @ labpt   : num [1:2] 4827772 2765414
.. .. . . . . . @ ID      : chr "1"
.. .. . . . . . @ area    : num 1.81e+09
.. ..$ :Formal class 'Polygons' [package "sp"] with 5 slots
.. .. . . . @ Polygons :List of 1
.. .. . . . . $ :Formal class 'Polygon' [package "sp"] with 5 slots
.. .. . . . . . @ labpt   : num [1:2] 4795824 2692300
.. .. . . . . . @ area    : num 1.34e+09
.. .. . . . . . @ hole    : logi FALSE
.. .. . . . . . @ ringDir: int 1
.. .. . . . . . @ coords  : num [1:9, 1:2] 4808360 4813350 4781472 47
90339 4776667 ...
.. .. . . . . . @ plotOrder: int 1
.. .. . . . . . @ labpt   : num [1:2] 4795824 2692300
.. .. . . . . . @ ID      : chr "2"
.. .. . . . . . @ area    : num 1.34e+09

```

```

..@ plotOrder : int [1:3] 2 3 1
..@ bbox       : num [1:2, 1:2] 4776667 2655076 4855527 2793852
.. .- attr(*, "dimnames")=List of 2
.. . . $ : chr [1:2] "x" "y"
.. . . $ : chr [1:2] "min" "max"
..@ proj4string:Formal class 'CRS' [package "sp"] with 1 slot
.. . . @ projargs: chr "+proj=laea +lat_0=52 +lon_0=10 +x_0=4321000 +y_0=3210000 +ellps=GRS80 +units=m +no_defs"

```

Avec sf :

```

library(sf)
nuts3 <- st_read(dsn = "data", layer = "nuts3", quiet = TRUE)
str(nuts3[1:3,])

```

```

Classes 'sf' and 'data.frame': 3 obs. of 8 variables:
$ id      : Factor w/ 1448 levels "AT111","AT112",...: 1 2 3
$ birth_2008: num 272 1195 748
$ death_2008: num 445 1480 1142
$ gdppps1999: num 509 2262 1368
$ gdppps2008: num 641 3272 1783
$ pop1999   : num 39149 137469 100114
$ pop2008   : num 37452 146383 97350
$ geometry  :sfc_MULTIPOLYGON of length 3; first list element: List of 1
..$ :List of 1
...$ : num [1:10, 1:2] 4819333 4824015 4808360 4802891 4797862 ...
..- attr(*, "class")= chr "XY" "MULTIPOLYGON" "sfg"
- attr(*, "sf_column")= chr "geometry"
- attr(*, "agr")= Factor w/ 3 levels "constant","aggregate",...: NA NA NA
NA NA NA NA
...- attr(*, "names")= chr "id" "birth_2008" "death_2008" "gdppps1999" .
..

```

```

## Simple feature collection with 100 features and 6 fields
## geometry type: MULTIPOLYGON
## dimension: XY
## bbox: xmin: -84.32385 ymin: 33.88199 xmax: -75.45698 ymax: 36.58965
## epsg (SRID): 4267
## proj4string: +proj=longlat +datum=NAD27 +no_defs
## precision: double (default; no precision model)
## First 3 features:
##   BIR74 SID74 NWBIR74 BIR79 SID79 NWBIR79
## 1 1091    1     10  1364     0    19 MULTIPOLYGON((( -81.47275543...
## 2 487     0     10   542     3    12 MULTIPOLYGON((( -81.23989105...
## 3 3188    5     208  3616     6   260 MULTIPOLYGON((( -80.45634460...

```

## 1.2.1 Import / Export

```
library(sf)
nuts3 <- st_read(dsn = "data", layer = "nuts3")
```

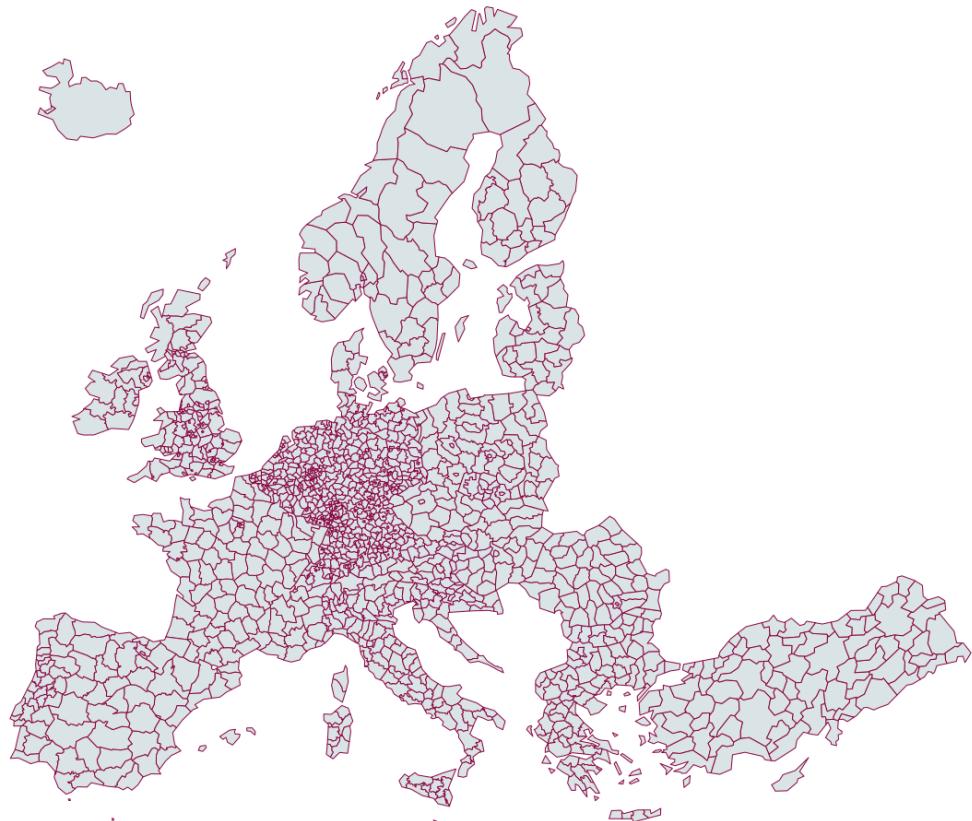
```
Reading layer `nuts3' from data source `/home/tg/Documents/prz/semin-r_201
7/data' using driver `ESRI Shapefile'
converted into: POLYGON
Simple feature collection with 1448 features and 7 fields
geometry type:  MULTIPOLYGON
dimension:      XY
bbox:           xmin: 2641758 ymin: 1427614 xmax: 7313157 ymax: 5411284
epsg (SRID):   NA
proj4string:    +proj=laea +lat_0=52 +lon_0=10 +x_0=4321000 +y_0=3210000 +
ellps=GRS80 +units=m +no_defs
```

## 1.2.2 Manipulation et affichage

```
plot(st_geometry(nuts3))
```



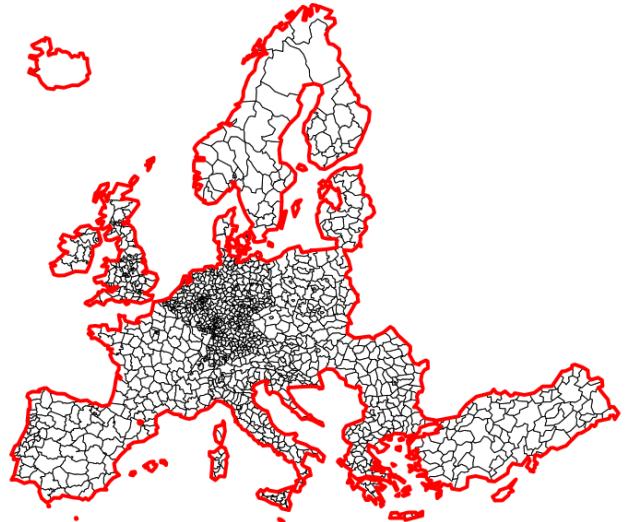
```
plot(st_geometry(nuts3), col = "#DAE3E6", border = "#8A0641", lwd = 0.5)
```



## 1.2.3 Géotraitements

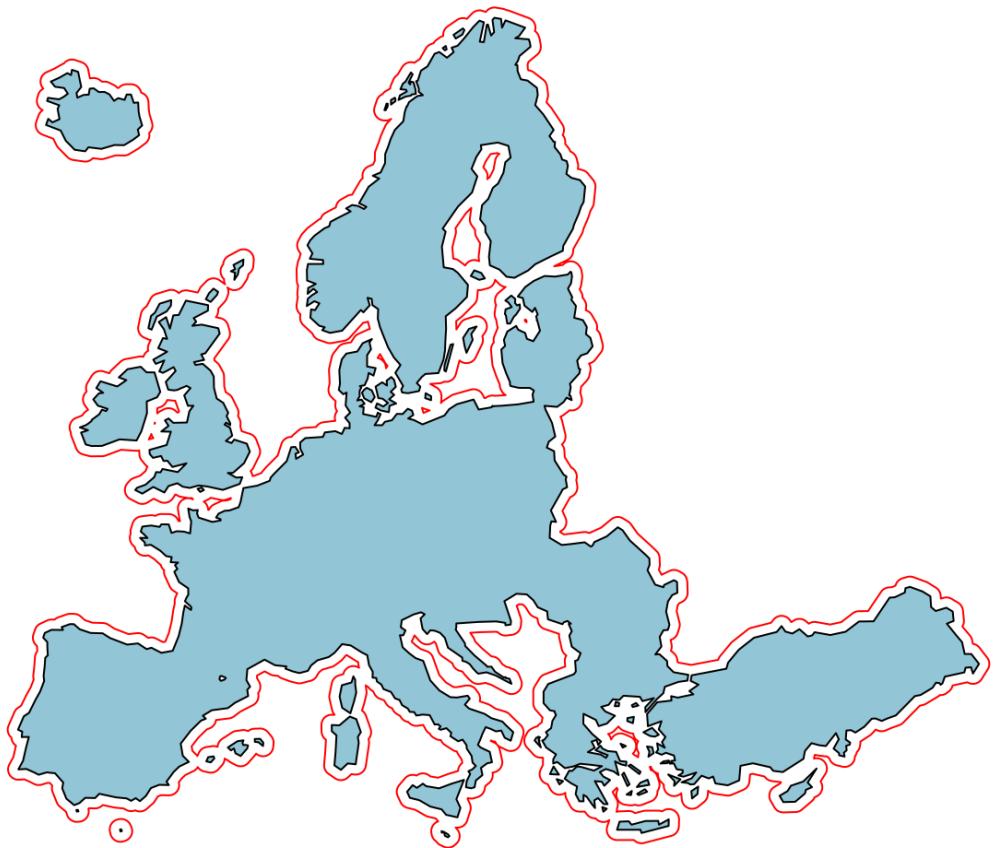
### 1.2.3.1 Agrégation des polygones / dissolve

```
europe <- st_union(x = nuts3)
plot(st_geometry(nuts3), lwd = 0.5)
plot(europe, lwd = 2, border = "red", add=T)
```



#### 1.2.3.2 Crédation de zones tampons / buffer

```
europeBuffer <- st_buffer(x = europe, dist = 50000)
plot(st_geometry(europe), col = "#92C5D6")
plot(europeBuffer, add = T, border = "red")
```



## 2 Le package cartography

### 2.1 Installation

- Version stable (CRAN)

```
install.packages("cartography")
```

- Version de développement (Github)

Cette version permet d'utiliser les objets `sf`.

```
devtools::install_github(repo = "Groupe-ElementR/cartography", ref = "devsf")
```

### 2.2 Utilisation

#### 2.2.1 Cartes choroplèthes

```

library(cartography)

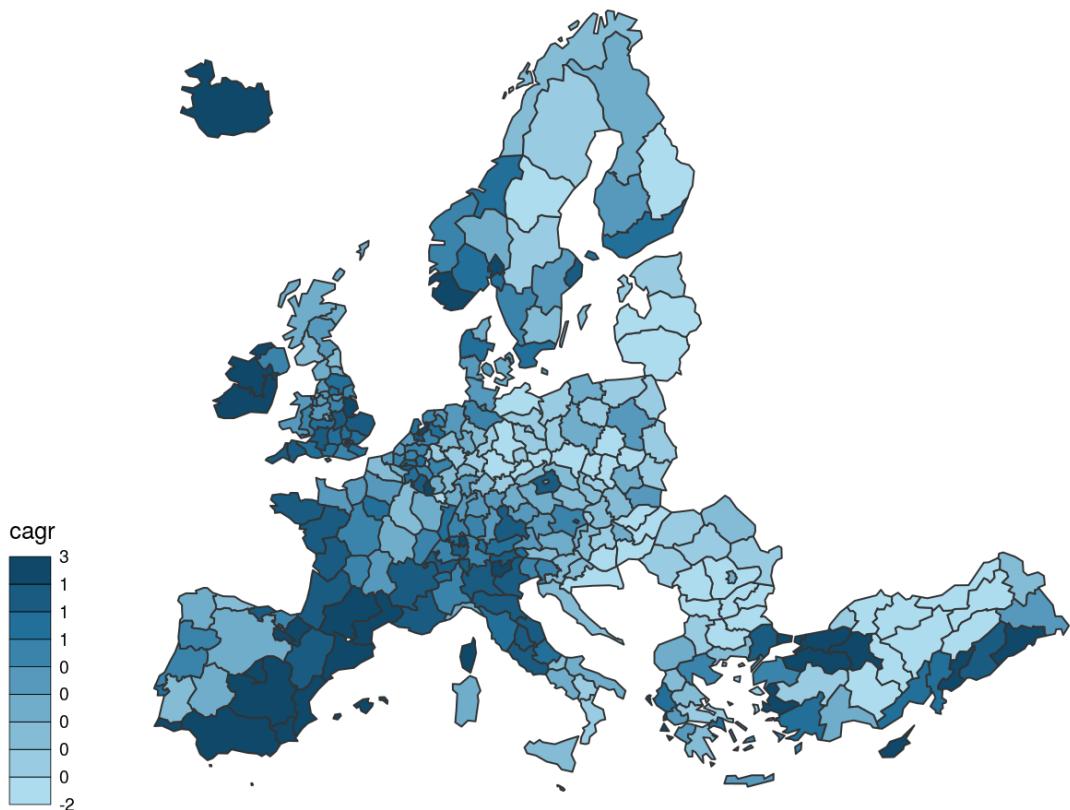
# chargement de données
data(nuts2006)

# Calcul du taux de croissance annuel moyen
nuts2.df$cagr <- 100 * (((nuts2.df$pop2008 / nuts2.df$pop1999)^(1/9)) - 1)

# Cartographie
choroLayer(spdf = nuts2.spdf, df = nuts2.df, var = "cagr")
title("Taux de croissance en Europe")

```

## Taux de croissance en Europe



Après ce premier jet, il est ensuite possible de paramétrer très finement la carte : palette de couleurs, discréétisation, légende, couches d'habillage...

```

# Construire une palette de couleurs
cols <- carto.pal(pal1 = "green.pal", n1 = 2,
                  pal2 = "red.pal", n2 = 4)

# Affichage de couches d'habillage
plot(nuts0.spdf, border = NA, col = NA, bg = "#A6CAE0")
plot(world.spdf, col = "#E3DEBF", border=NA, add=TRUE)

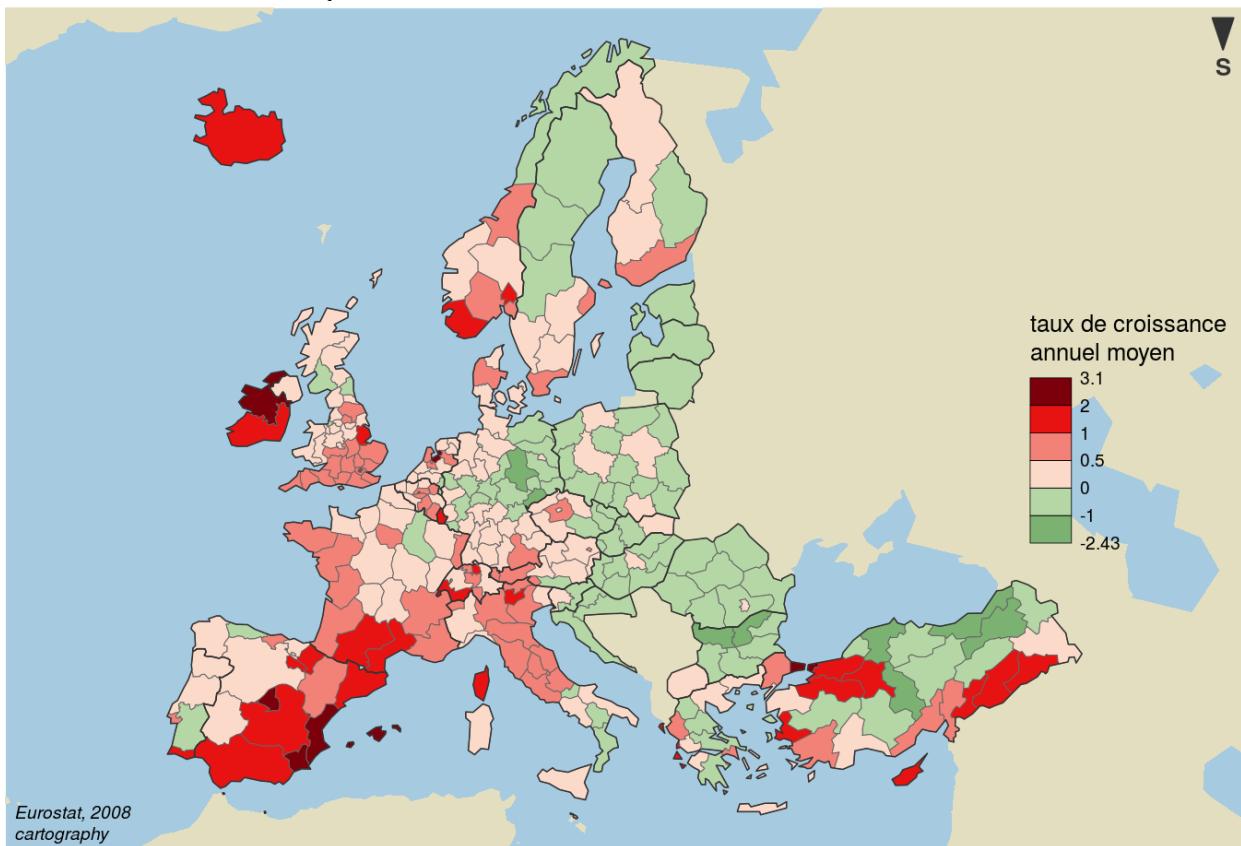
# Cartographie du taux de croissance annuel moyen
choroLayer(spdf = nuts2.spdf, df = nuts2.df, var = "cagr",
           breaks = c(-2.43, -1.0, 0.0, 0.5, 1.0, 2.0, 3.1),
           col = cols,
           border = "grey40",
           lwd = 0.5,
           legend.pos = "right",
           legend.title.txt = "taux de croissance\annuel moyen",
           legend.values.rnd = 2,
           add = TRUE)

# Affichage de couches d'habillage
plot(nuts0.spdf, border = "grey20", lwd=0.75, add=TRUE)

# Ajout des titres, légende, sources, etc.
layoutLayer(title = "Taux de croissance en Europe",
            author = "cartography",
            sources = "Eurostat, 2008", frame = TRUE, col = NA,
            scale = NULL, coltitle = "black",
            south = TRUE)

```

Taux de croissance en Europe



## 2.2.2 Cartes en symboles proportionnels

Cartographie d'un stock (la population nationale) avec des figurés proportionnels.

```

# Affichage de couches d'habillage
plot(nuts0.spdf, border = NA, col = NA, bg = "#A6CAE0")
plot(world.spdf, col = "#E3DEBF", border=NA, add=TRUE)
plot(nuts0.spdf, col = "#D1914D",border = "grey80", add=TRUE)

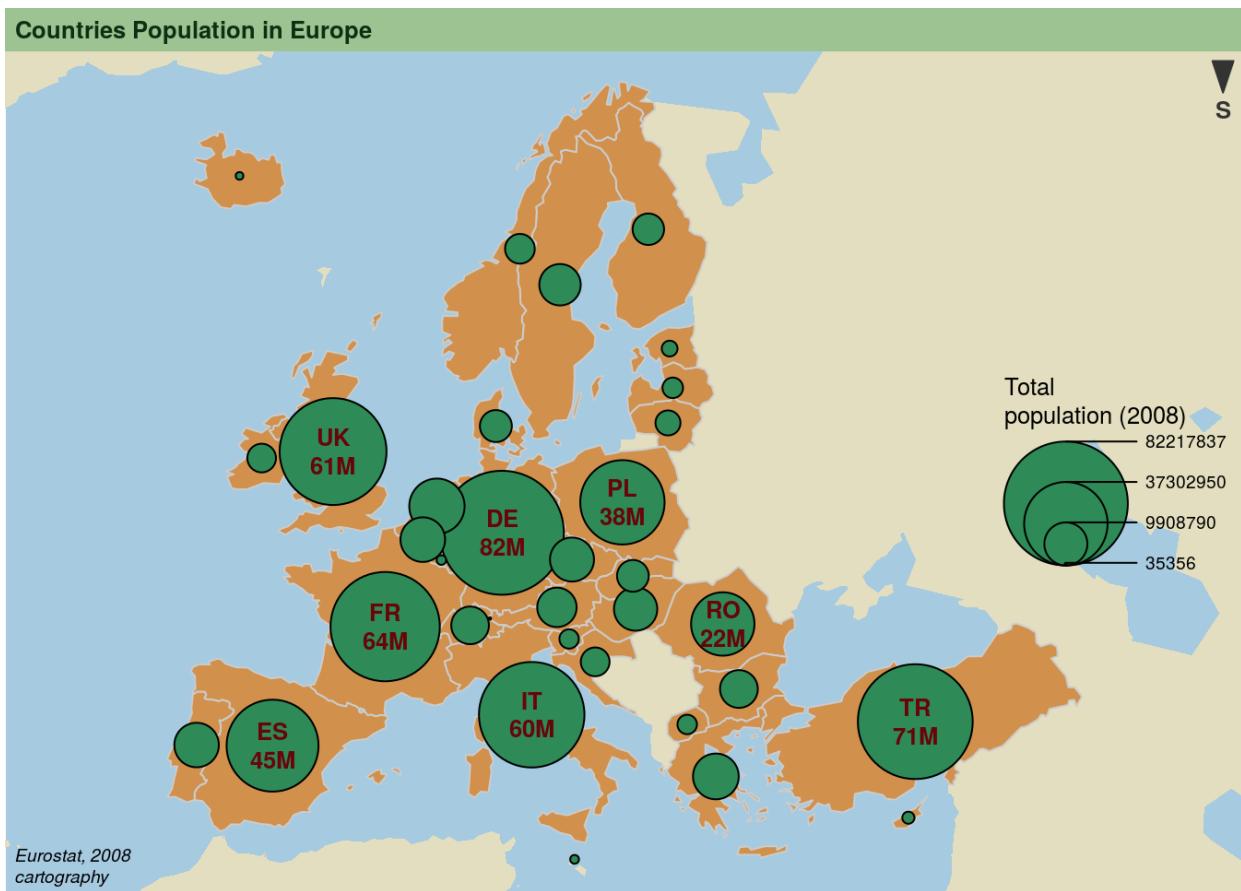
# Cartographie de la population des pays en cercles proportionnels
propSymbolsLayer(spfd = nuts0.spdf, df = nuts0.df,
                 var = "pop2008",
                 symbols = "circle", col = "seagreen4",
                 legend.pos = "right", inches = 0.35,
                 legend.title.txt = "Total\ncpopulation (2008)",
                 legend.style = "c")

# Ajout de labels
dflab <- nuts0.df[order(nuts0.df$pop2008, decreasing = TRUE),][1:8,]
dflab$lab <- paste(dflab$id, "\n", round(dflab$pop2008/1000000,0), "M", se
p = "")

# Label plot of the 8 most populated countries
labelLayer(spfd = nuts0.spdf,
            df = dflab,
            txt = "lab",
            col = "#690409",
            cex = 0.8,
            font = 2)

# Ajout des titres, légende, sources, etc.
layoutLayer(title = "Countries Population in Europe",
            theme = 'green.pal',
            frame = FALSE,
            author = "cartography",
            sources = "Eurostat, 2008",
            scale = NULL,
            south = TRUE)

```



### 2.2.3 Cartes en symboles proportionnels colorés

```

# Load data
data(nuts2006)

# Compute the compound annual growth rate
nuts2.df$cagr <- (((nuts2.df$pop2008 / nuts2.df$pop1999)^(1/9)) - 1) * 100

# Plot a layer with the extent of the EU28 countries with only a background color
plot(nuts0.spdf, border = NA, col = NA, bg = "#A6CAE0")
# Plot non european space
plot(world.spdf, col = "#E3DEBF", border = NA, add = TRUE)
# Plot Nuts2 regions
plot(nuts2.spdf, col = "grey60", border = "white", lwd = 0.4, add = TRUE)

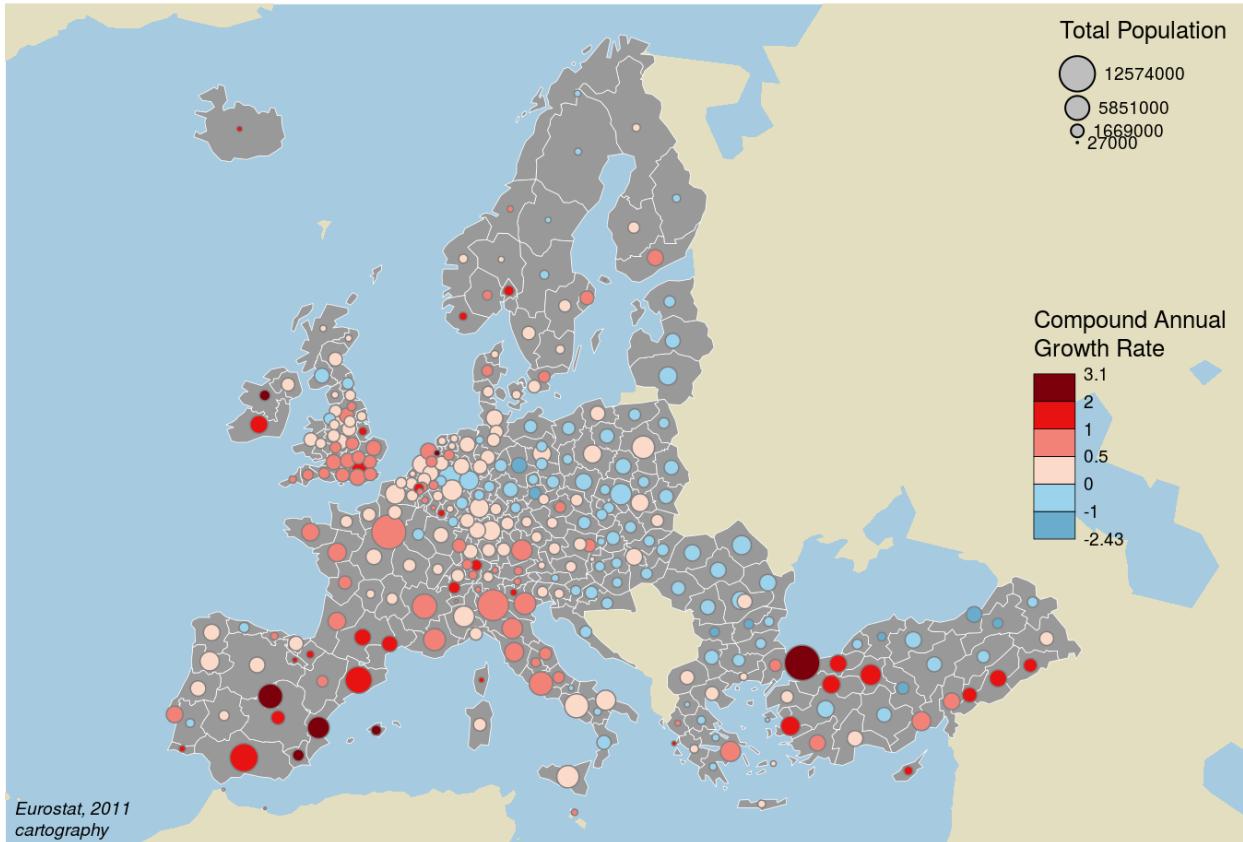
# Set a custom color palette
cols <- carto.pal(pal1 = "blue.pal", n1 = 2, pal2 = "red.pal", n2 = 4)

# Plot symbols with choropleth coloration
propSymbolsChoroLayer(spdf = nuts2.spdf,
                      df = nuts2.df,
                      var = "pop2008",
                      inches = 0.1,
                      var2 = "cagr",
                      col = cols,
                      breaks = c(-2.43,-1,0,0.5,1,2,3.1),
                      border = "grey50",
                      lwd = 0.75,
                      legend.var.pos = "topright",
                      legend.var.values.rnd = -3,
                      legend.var.title.txt = "Total Population",
                      legend.var.style = "e",
                      legend.var2.pos = "right",
                      legend.var2.title.txt = "Compound Annual\nGrowth Rate")

# layout
layoutLayer(title = "Demographic trends, 1999-2008", coltitle = "black",
            sources = "Eurostat, 2011", scale = NULL,
            author = "cartography", frame = "", col = NA)

```

### Demographic trends, 1999-2008



### 2.2.4 Cartes de flux

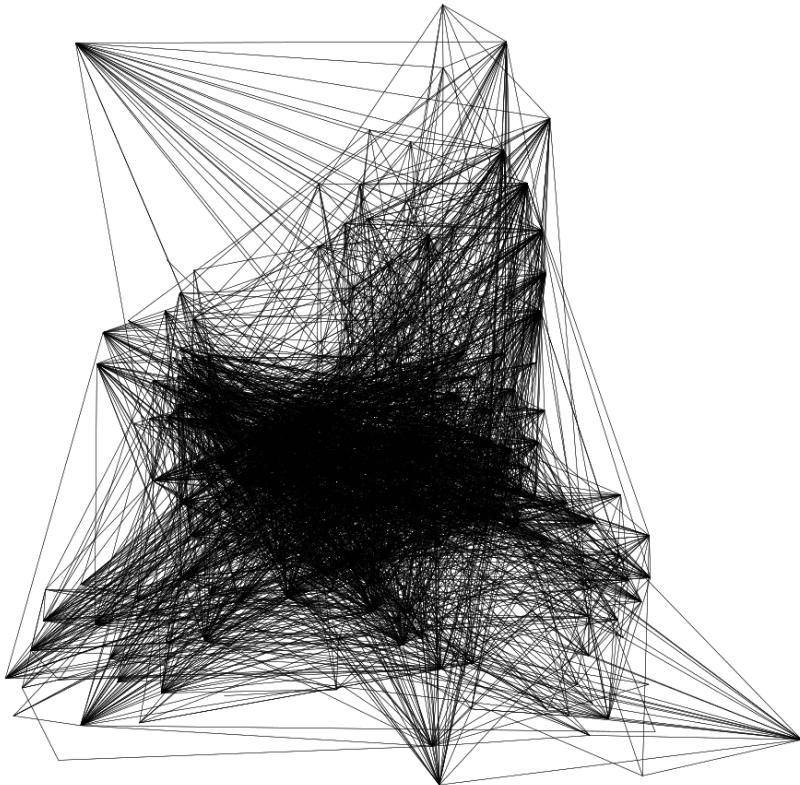
Il s'agit de représenter des données, agrégées à un niveau régional, sur les jumelages entre villes.

```
# Données sur les jumelages
head(twincities.df)
```

	i	j	fij
1	DE14	AT11	1
2	DE21	AT11	1
3	DE23	AT11	1
4	DE26	AT11	2
5	DE91	AT11	1
6	DEB3	AT11	1

```
# Creation d'une couche de liens
twincities.sf <- getLinkLayer(x = nuts2.spdf, df = twincities.df[,1:2])

# Affichage des liens créés
plot(st_geometry(twincities.sf), lwd = 0.2)
```



```

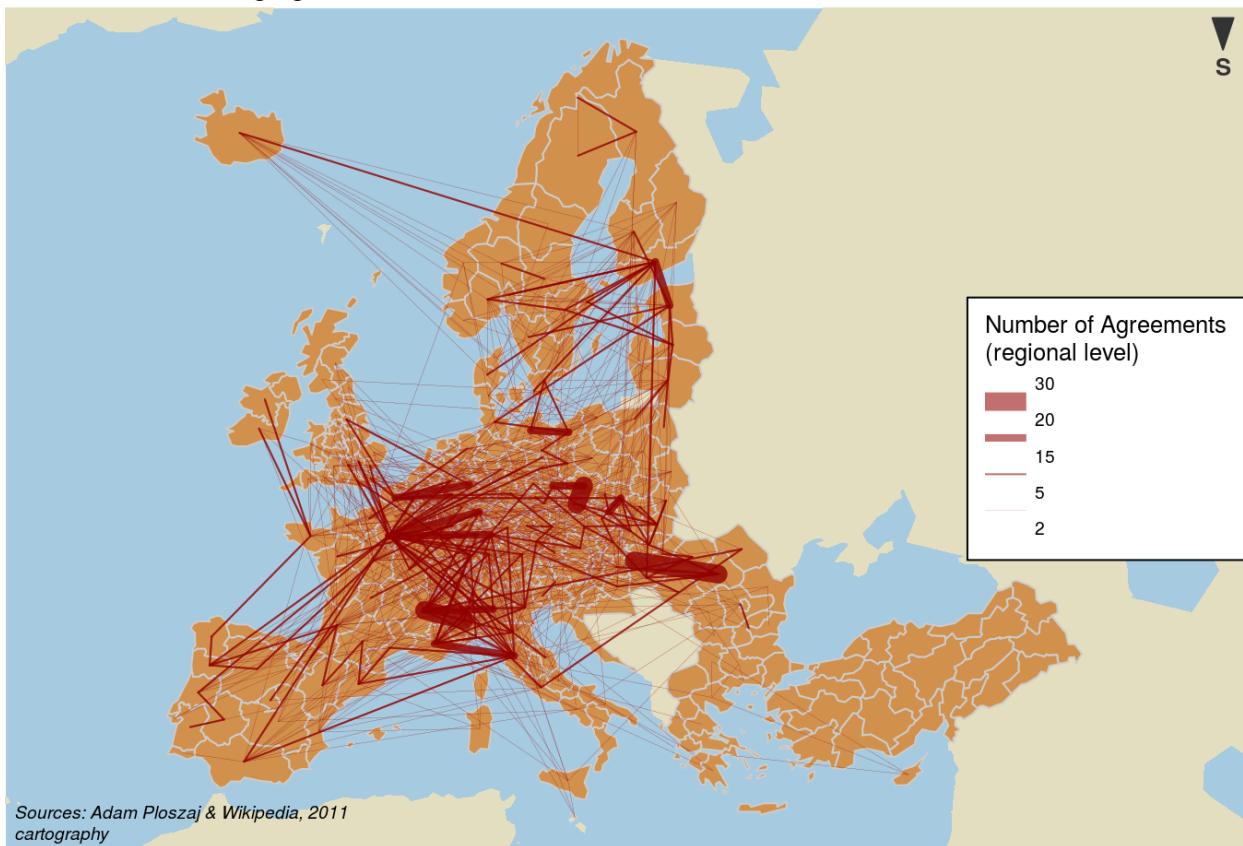
# Affichage de couches d'habillage
plot(nuts0.spdf, border = NA, col = NA, bg = "#A6CAE0")
plot(world.spdf, col = "#E3DEBF", border=NA, add=TRUE)
plot(nuts2.spdf, col = "#D1914D",border = "grey80", add=TRUE)

# Cartographie des liens
gradLinkLayer(x = twincities.sf, df = twincities.df,
               var = "fij",
               breaks = c(2,5,15,20,30),
               lwd = c(0.1,1,4,10),
               col = "#92000090",
               legend.pos = "right", legend.frame = TRUE,
               legend.title.txt = "Number of Agreements\n(region level)",
               add = TRUE)

# Ajout des titres, légende, sources, etc.
layoutLayer(title = "International Twinning Agreements Between Cities",
            author = "cartography",
            sources = "Sources: Adam Ploszaj & Wikipedia, 2011",
            scale = NULL, south = TRUE, frame = TRUE, col = NA,
            coltitle = "black")

```

### International Twinning Agreements Between Cities

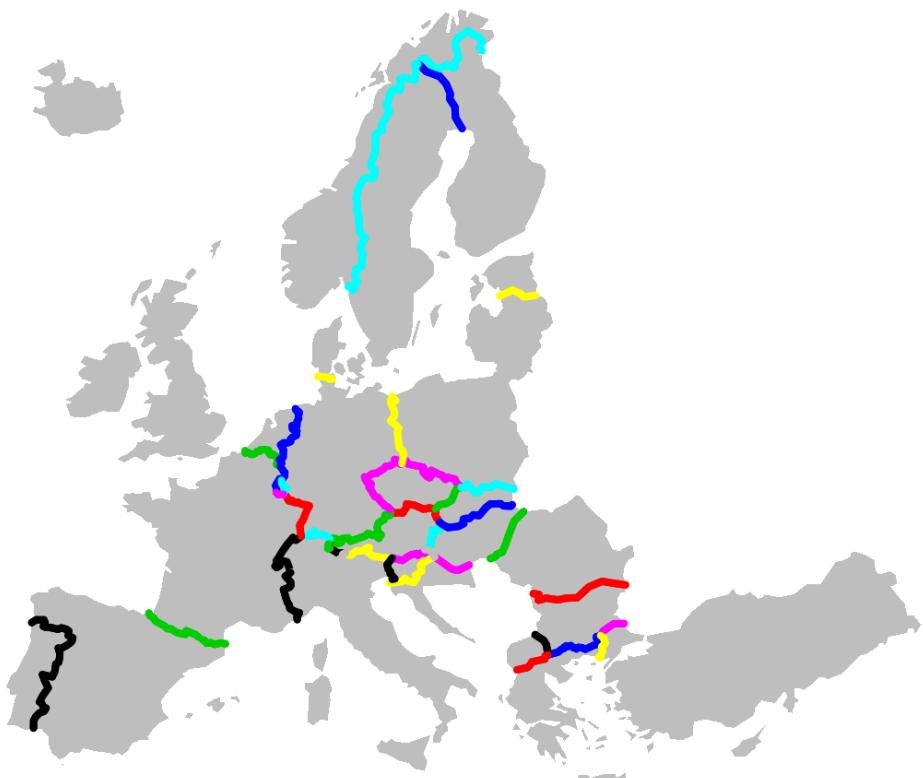


## 2.2.5 Discontinuités

```
# Load data
data(nuts2006)

# Get a SpatialLinesDataFrame of countries borders
nuts0.contig <- getBorders(spdf = nuts0.spdf)

plot(nuts0.spdf, col = "grey", border = NA)
plot(st_geometry(nuts0.contig),
     col = 1:nrow(nuts0.contig),
     add=T, lwd = 4)
```



```

# Get the GDP per capita
nuts0.df$gdpcap <- nuts0.df$gdppps2008/nuts0.df$pop2008*1000000

# Plot a layer with the extent of the EU28 countries with only a background color
plot(nuts0.spdf, border = NA, col = NA, bg = "#A6CAE0")
# Plot non european space
plot(world.spdf, col = "#E3DEBF", border=NA, add=TRUE)

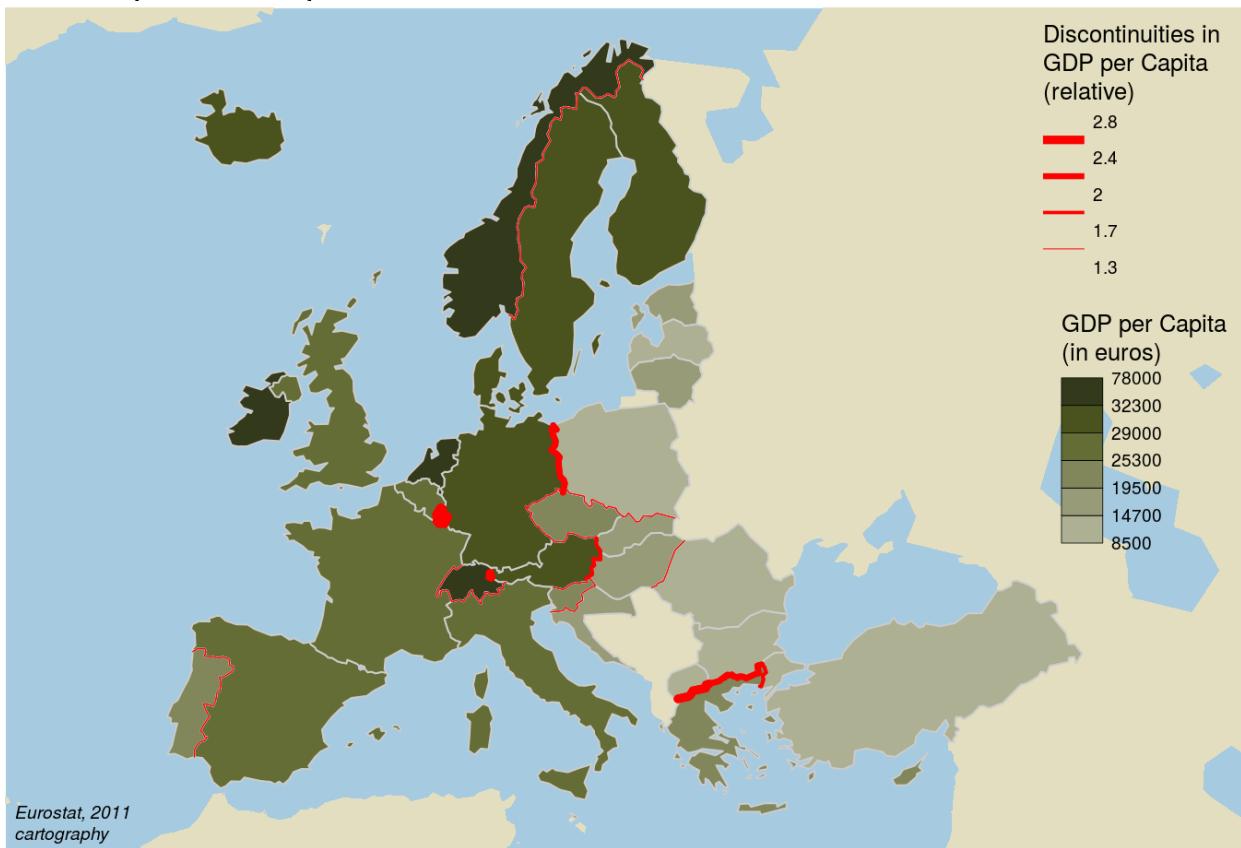
# Plot GDP per capita with a choropleth layer
choroLayer(spdf = nuts0.spdf, df = nuts0.df, var = "gdpcap", border = "grey80",
           col = carto.pal(pal1 = "kaki.pal", nl = 6), method = "quantile",
           nclass = 6, add=TRUE, legend.pos = "right",
           legend.values.rnd = -2,
           legend.title.txt = "GDP per Capita\n(in euros)")

# Plot discontinuities
discLayer(x = nuts0.contig, # sf of borders
           df = nuts0.df, # data frame on countries
           var = "gdpcap", # variable used to compute discontinuities
           type = "rel", # type of discontinuity measure
           method="equal", # discretisation of discontinuities
           nclass=4, # number of discontinuities classes
           threshold = 0.5, # representation threshold of discontinuities
           sizemin = 0.5, # minimum size of discontinuities lines
           sizemax = 6, # maximum size of discontinuities lines
           col="red", # color of the lines
           legend.values.rnd = 1,
           legend.title.txt = "Discontinuities in \nGDP per Capita\n(relative)",
           legend.pos = "topright",
           add=TRUE)

# Layout
layoutLayer(title = "Wealth Disparities in Europe", coltitle = "black",
            sources = "Eurostat, 2011", scale = NULL,
            author = "cartography", frame = "", col = NA)

```

### Wealth Disparities in Europe



## 2.2.6 Carroyages

```

library(cartography)
# Load data
data(nuts2006)

# Create a grid layer
nuts2.spdf@data <- nuts2.df
mygrid <- getGridLayer(x = nuts2.spdf,
                        cellsize = 200000 * 200000,
                        var = "pop2008")

# Plot density of population
## conversion from square meter to square kilometers
mygrid$densitykm <- mygrid$pop2008 * 1000 * 1000 / mygrid$gridarea

# Plot a layer with the extent of the EU28 countries with only a background color
plot(nuts0.spdf, border = NA, col = NA, bg = "#A6CAE0")
# Plot non european space
plot(world.spdf, col = "#E3DEBF", border=NA, add=TRUE)

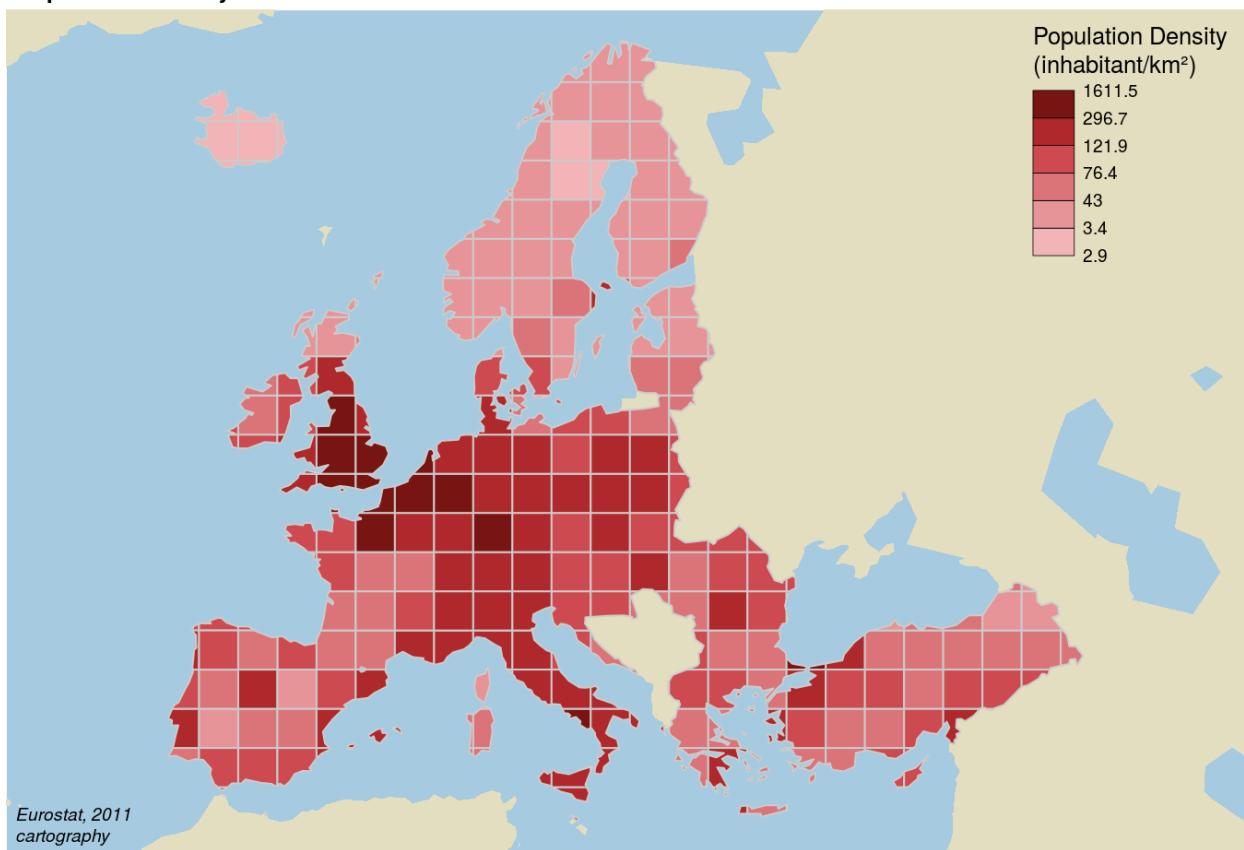
# Set a custom color palette
cols <- carto.pal(pall = "wine.pal", n1 = 6)

# Plot the gridded population density
choroLayer(x = mygrid, var = "densitykm",
            border = "grey80", col = cols, legend.pos = "topright",
            method = "q6", add = TRUE, legend.values.rnd = 1,
            legend.title.txt = "Population Density\n(inhabitant/km2)")

# Layout
layoutLayer(title = "Population Density", coltitle = "black",
            sources = "Eurostat, 2011", scale = NULL,
            author = "cartography", frame = "", col = NA)

```

**Population Density**



## 2.2.7 Cartes lissées

```

# set margins
opar <- par(mar = c(0, 0, 1.2, 0))
# Load data
data(nuts2006)

nuts3.spdf@data = nuts3.df
# Create a grid layer
mygrid <- getGridLayer(x = sf::st_as_sf(nuts3.spdf),
                        cellsize = 50000 * 50000,
                        type = "regular",
                        var = c("pop2008", "gdppps2008"))

# Compute data for the grid layer
mygrid$gdp <- mygrid$gdppps2008*1000000

# list of breaks
v <- c(2920, 5000, 10000, 15000, 20000, 23500, 30000, 35000, 40000, 42720)
# Plot a layer with the extent of the EU28 countries with only a background
# color
plot(nuts0.spdf, border = NA, col = NA, bg = "#A6CAE0")

# Plot non european space
plot(world.spdf, col = "#E3DEBF", border = NA, add = TRUE)

# set a color palette
cols <- c(rev(carto.pal("green.pal", 5)), carto.pal("orange.pal", 4))

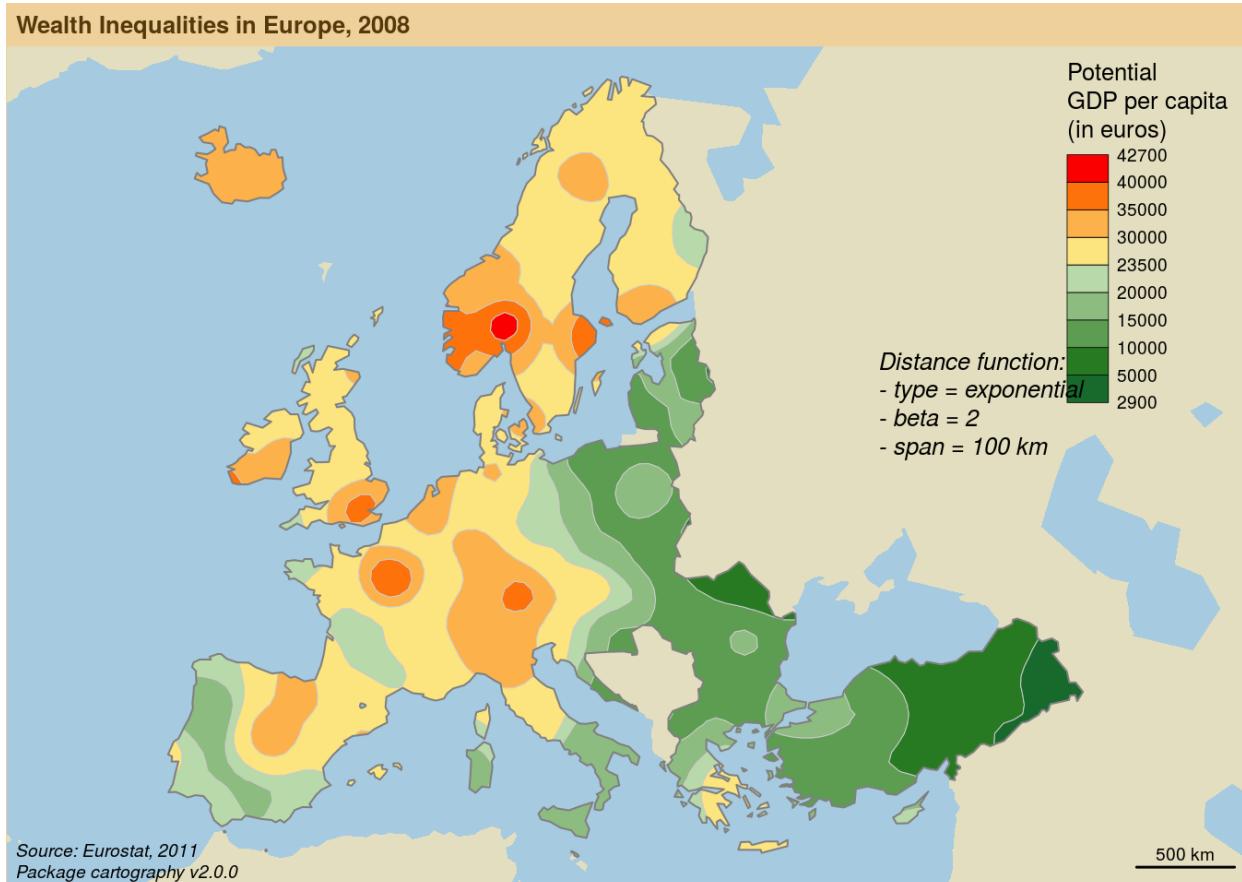
# compute & display the potential map
smoothLayer(x = mygrid, var = "gdp", var2 = "pop2008", breaks = v,
             span = 1e+05, beta = 2, mask = nuts0.spdf, resolution = 49000,
             col = cols,
             legend.title.txt = "Potential\nGDP per capita\n(in euros)", legend.values.rnd = -2,
             border = "grey80", lwd = 0.5, add = T, legend.pos = "topright"
            )

# plot Europe contour
plot(rgeos::gBuffer(nuts0.spdf, FALSE, 1), add = T, col = NA, border = "grey50")

# plot a layout
layoutLayer(title = "Wealth Inequalities in Europe, 2008",
            author = "Package cartography v2.0.0",
            sources = "Source: Eurostat, 2011", frame = TRUE, scale = 500,
            north = FALSE,
            theme = "sand.pal")

```

```
# plot a text on the map
text(x = 6271272, y = 3743765, labels = "Distance function:\n- type = expo
nential\n- beta = 2\n- span = 100 km",
cex = 0.8, adj = 0, font = 3)
```



## 3 Ressources

La présentation est accessible à cette adresse :

[https://rcarto.github.io/semin-r\\_2017](https://rcarto.github.io/semin-r_2017) ([https://rcarto.github.io/semin-r\\_2017](https://rcarto.github.io/semin-r_2017))

Son code source est accessible sur GitHub ([https://github.com/rCarto/semin-r\\_2017](https://github.com/rCarto/semin-r_2017)).

### 3.1 Package cartography

**La page GitHub du package cartography**

C'est dans ce dépôt GitHub (<https://github.com/Groupe-ElementR/cartography>) que se déroule le développement du *package* et que se tiennent les discussions (<https://github.com/Groupe-ElementR/cartography/issues>) à son sujet.

**Billets de blogs, tutoriels, présentations**

Demo codes in the R graph Gallery (<http://www.r-graph-gallery.com/portfolio/maps/>) (EN)

Create and integrate maps in your R workflow with the cartography package (<http://rgeomatic.hypotheses.org/842>) (EN)

De superbes cartes thématiques... (<http://rgeomatic.hypotheses.org/1086>)(FR)  
Le package cartography a un an (<http://rgeomatic.hypotheses.org/1016>)(FR)  
Cartographie avec R : le package cartography (<http://rgeomatic.hypotheses.org/659>) (FR)  
R pour les cartographies (<http://neocarto.hypotheses.org/1859>) (FR)  
Comment faire un carton avec R? (<http://rgeomatic.hypotheses.org/category/cartography>) (FR - *How to build inset maps*)  
Tutoriel - Cartographie avec R (<http://wukan.ums-riate.fr/r2016/>) (FR)  
Cartographie et traitement de l'information géographique avec R ([http://wukan.ums-riate.fr/RUSS/RUSS\\_2016/](http://wukan.ums-riate.fr/RUSS/RUSS_2016/)) (FR)  
R pour les cartographies : le package cartography ([https://osgeo-fr.github.io/presentations\\_foss4gfr/2016/J1/R\\_Cartography\\_T\\_Giraud\\_FOSS4G-fr-2016/FOSS4G-fr-2016.html](https://osgeo-fr.github.io/presentations_foss4gfr/2016/J1/R_Cartography_T_Giraud_FOSS4G-fr-2016/FOSS4G-fr-2016.html)) (FR)  
Le blog <http://rgeomatic.hypotheses.org/> (<http://rgeomatic.hypotheses.org/>)

## 3.2 Package sf

### La page GitHub du package sf

C'est dans ce dépôt GitHub (<https://github.com/edzer/sfr>) que se déroule le développement du package et que se tiennent les discussions (<https://github.com/edzer/sfr/issues?utf8=%E2%9C%93&q=is%3Aissue%20>) à son sujet.

### Les ressources publiées par Edzer Pebesma

Le créateur et *maintainer* de sf a publié un certain de nombre de vignettes et de billets autour du package.

- Vignettes
  - Simple Features for R (<https://cran.r-project.org/web/packages/sf/vignettes/sf1.html>)  
Explication très détaillée des principes et du fonctionnement du package.
  - Reading, Writing and Converting Simple Features (<https://cran.r-project.org/web/packages/sf/vignettes/sf2.html>) Comment importer et exporter des objets.
  - Manipulating Simple Feature Geometries (<https://cran.r-project.org/web/packages/sf/vignettes/sf3.html>) Comment manipuler les objets sf , géotraitements, projections...
- Billets de blog
  - Simple features for R (<http://r-spatial.org/r/2016/02/15/simple-features-for-r.html>)
  - Simple features for R, part 2 (<http://r-spatial.org/r/2016/07/18/sf2.html>)
  - Simple features now on CRAN (<http://r-spatial.org/r/2016/11/02/sfcran.html>)
  - sf - plot, graticule, transform, units, cast, is (<http://r-spatial.org/r/2017/01/12/newssf.html>)
- Migration de sp vers sf

Un tableau de migration de sp vers sf (<https://github.com/edzer/sfr/wiki/migrating>) pour passer de l'utilisation des packages sp , rgeos et rgdal à l'utilisation de sf .

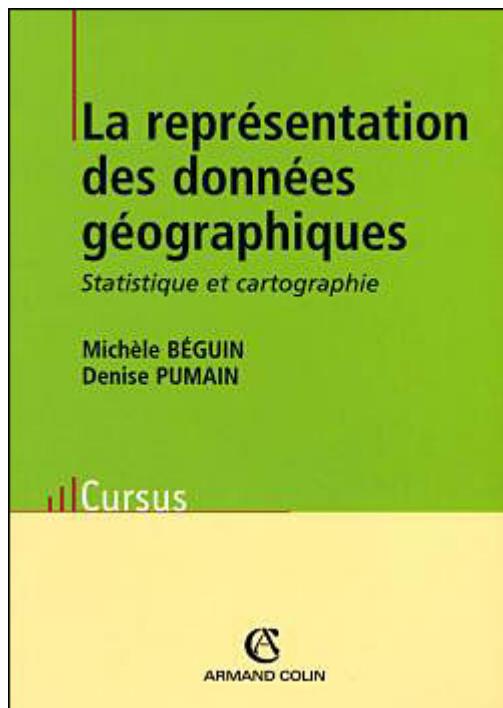
### Billets de blogs / tutoriels

- Faire des cartes avec R - Introduction au package sf ([https://github.com/riatelab/intro\\_sf](https://github.com/riatelab/intro_sf)) -

Santiago Mora

- Tidy spatial data in R: using dplyr, tidyr, and ggplot2 with sf (<http://strimas.com/r/tidy-sf/>) - Matt Strimas-Mackey
- First Impressions From sf – The Simple Features R Package (<https://geographicdatascience.com/2017/01/06/first-impressions-from-sf-the-simple-features-r-package/>) - Geographic Data Science Lab
- Spatial analysis pipelines with simple features in R (<https://walkerke.github.io/2016/12/spatial-pipelines/>) - Kyle Walker
- Spatial analysis in R with the sf package ([http://rhodyrstats.org/geospatial\\_with\\_sf/geospatial\\_with\\_sf.html](http://rhodyrstats.org/geospatial_with_sf/geospatial_with_sf.html)) - rhodyRstats
- Landscape Ecology with R: A tutorial with raster and sf ([http://jwhollister.com/r\\_landscape\\_tutorial/tutorial.html](http://jwhollister.com/r_landscape_tutorial/tutorial.html)) - Jeffrey W. Hollister
- Mapping “France at night” with the new sf package (<http://sharpsightlabs.com/blog/mapping-france-night/>) - Sharp Sight Labs

### 3.3 La cartographie



Béguin & Pumain (2003)

- Michèle Béguin et Denise Pumain. “La représentation des données géographiques, Statistique et cartographie.” (2003). Paris, Armand Colin, Coll. Cursus, 192p.

NICOLAS LAMBERT  
CHRISTINE ZANIN

## Manuel de cartographie



Principes, méthodes,  
applications

ARMAND COLIN

Lambert & Zanin (2016)

- Nicolas Lambert et Christine Zanin. "Manuel de cartographie: principes, méthodes, applications." (2016). Paris, Armand Colin, Coll. Cursus, 224p.