

# Cartographie avec R

*Timothée Giraud & Hugues Pécout*

*2018-12-12*

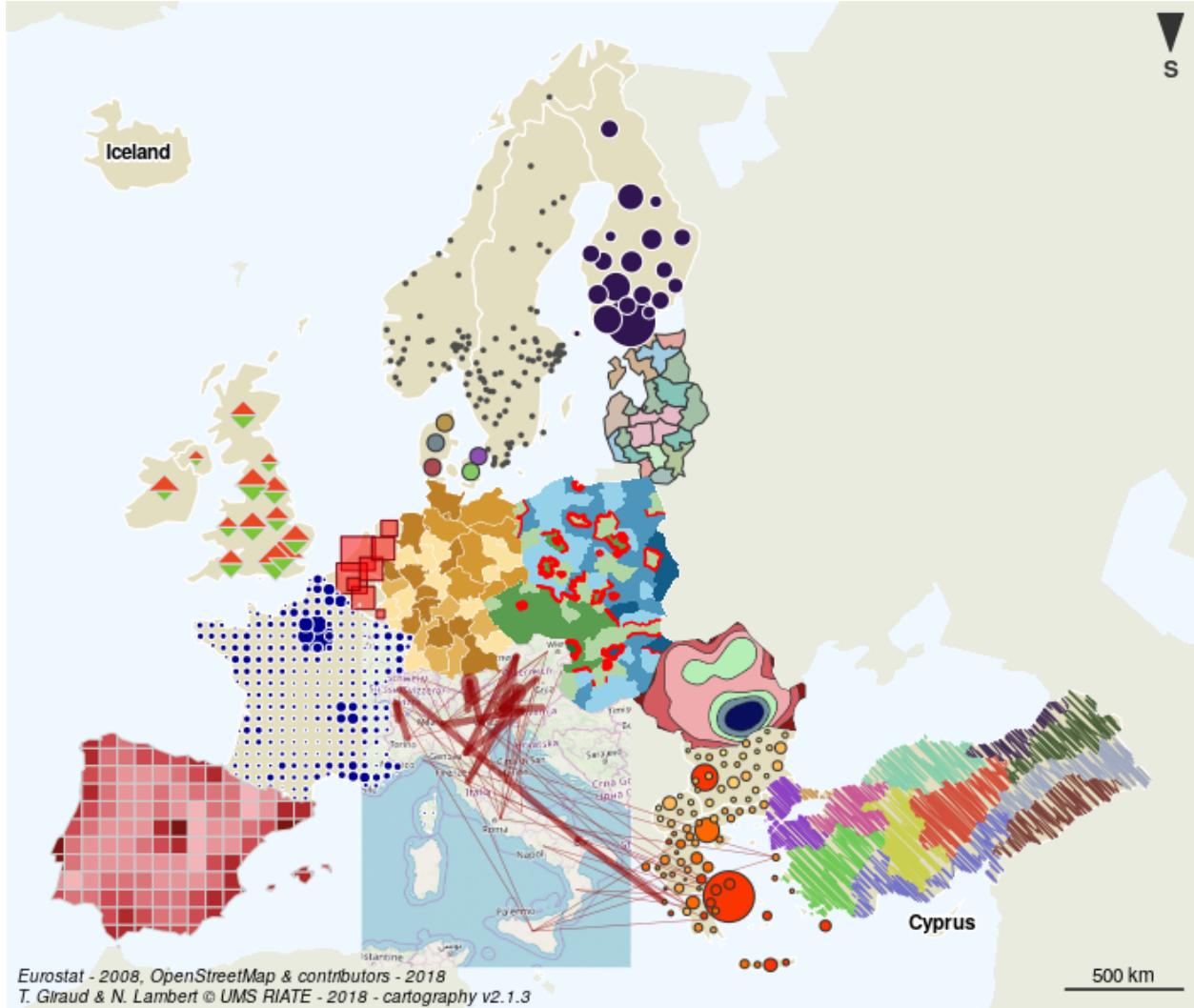


# Contents

<b>Introduction</b>	<b>5</b>
<b>1 Les données spatiales</b>	<b>7</b>
1.1 Le package <code>sf</code> . . . . .	7
1.2 Import / Export . . . . .	8
1.3 Les projections . . . . .	9
1.4 Les opérations de géotraitement . . . . .	9
1.5 Le package <code>raster</code> . . . . .	12
<b>2 Cartographie thématique</b>	<b>13</b>
2.1 Le package <code>cartography</code> . . . . .	13
2.2 Palettes de couleurs . . . . .	15
2.3 Discrétisations . . . . .	17
2.4 Combinaisons . . . . .	18
2.5 Labels . . . . .	19
2.6 Les données OSM . . . . .	20
2.7 Cartographie interactive . . . . .	21
2.8 Géocodage d'adresses . . . . .	21
2.9 Création de cartons . . . . .	21
<b>3 Cartographie thématique avancée</b>	<b>23</b>
3.1 Les anamorphoses . . . . .	23
3.2 Les grilles régulières . . . . .	30
3.3 Les discontinuités . . . . .	32
3.4 Les lissages . . . . .	34
3.5 3D . . . . .	36



# Introduction



Voici le document associé au cours de cartographie avec R.

Pour suivre ce cours vous aurez besoin des dernières versions de R et de RStudio.  
Vous aurez aussi besoin d'un certain nombre de *packages* additionnels :

- sf
- cartography
- osrm
- mapview

- SpatialPosition
- mapinsetr
- raster
- linemap
- spatstats
- ...

Ce cours se déroule sur 3 jours :

- Jour 1 - Les données spatiales
- Jour 2 - Cartographie thématique
- Jour 3 - Cartographie thématique avancée

# Chapter 1

## Les données spatiales

Il est possible d'importer, de manipuler, de traiter, d'afficher et d'exporter des données spatiales avec R. La grande majorité des opérations de géotraitement sont disponible dans R grâce au package **sf**. Il devient alors possible d'utiliser R comme un SIG.

### 1.1 Le package **sf**

Trois packages “historiques” \* **rgdal** : interface entre R et les librairies GDAL (Geospatial Data Abstraction Library) et PROJ4. \* **sp** : classes et méthodes pour les données spatiales dans R. \* **rgeos** : accès à la librairie d’opérations spatiales GEOS (Geometry Engine - Open Source) : area, perimeter, distances, dissolve, buffer, overlap, union, contains...

- Les fonctionnalités de **sp**, **rgeos** et **rgdal** dans un package unique.
- Manipulation plus aisée, objets plus simples
- Auteur principal et *maintainer* : Edzer Pebesma (auteur de **sp**)
- Compatible avec les syntaxes *pipe* et les opérateurs du **tidyverse**.

Format des objets spatiaux **sf**

## Simple feature collection with 100 features and 6 fields						geom
## 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
##	2	487	0	10	542	3
##	3	3188	5	208	3616	6
						19
						MULTIPOLYGON((( -81.47275543...
						12
						MULTIPOLYGON((( -81.23989105...
						260
						MULTIPOLYGON((( -80.45634460...

Simple feature      Simple feature geometry list-column (sf)      Simple feature geometry (sfg)

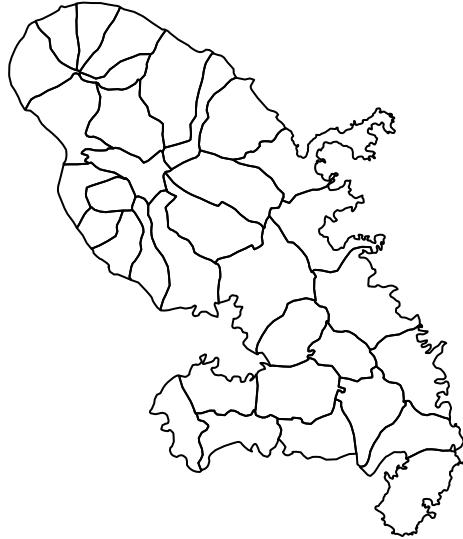
Import de données

```
library(sf)
mtq <- st_read("data/martinique.shp")
```

```
Reading layer `martinique' from data source `/home/tim/Documents/prz/carto_avec_r/data/martinique.shp'
Simple feature collection with 34 features and 23 fields
geometry type:  POLYGON
dimension:      XY
bbox:           xmin: 690574.4 ymin: 1592426 xmax: 736126.5 ymax: 1645660
epsg (SRID):   32620
proj4string:    +proj=utm +zone=20 +datum=WGS84 +units=m +no_defs
```

Affichage de données

```
plot(st_geometry(mtq))
```



## 1.2 Import / Export

Import de données

```
library(sf)
mtq <- st_read("data/martinique.shp")
```

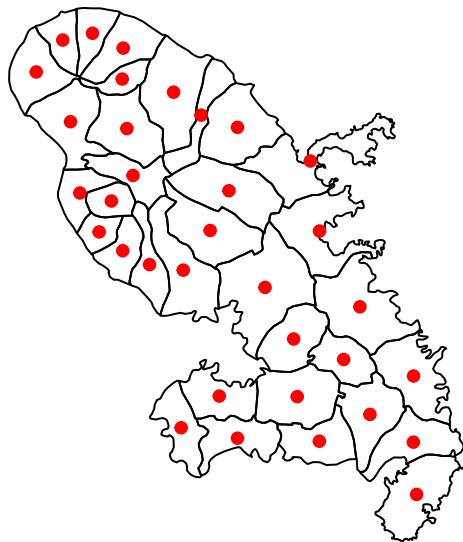
```
Reading layer `martinique' from data source `/home/tim/Documents/prz/carto_avec_r/data/martinique.shp'
Simple feature collection with 34 features and 23 fields
geometry type:  POLYGON
dimension:      XY
bbox:           xmin: 690574.4 ymin: 1592426 xmax: 736126.5 ymax: 1645660
epsg (SRID):   32620
proj4string:    +proj=utm +zone=20 +datum=WGS84 +units=m +no_defs
```

## 1.3 Les projections

### 1.4 Les opérations de géotraitements

Utiliser R comme un SIG

```
mtq_c <- st_centroid(mtq)
plot(st_geometry(mtq))
plot(st_geometry(mtq_c), add=TRUE, cex=1.2, col="red", pch=20)
```

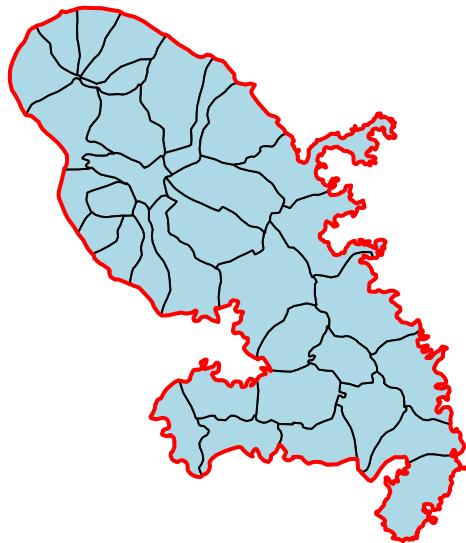


```
mat <- st_distance(x=mtq_c, y=mtq_c)
mat[1:5, 1:5]
```

```
Units: [m]
      [,1]     [,2]     [,3]     [,4]     [,5]
[1,] 0.000 35297.56 3091.501 12131.617 17136.310
[2,] 35297.557 0.000 38332.602 25518.913 18605.249
[3,] 3091.501 38332.600 0.000 15094.702 20226.198
[4,] 12131.617 25518.91 15094.702 0.000 7177.011
[5,] 17136.310 18605.25 20226.198 7177.011 0.000
```

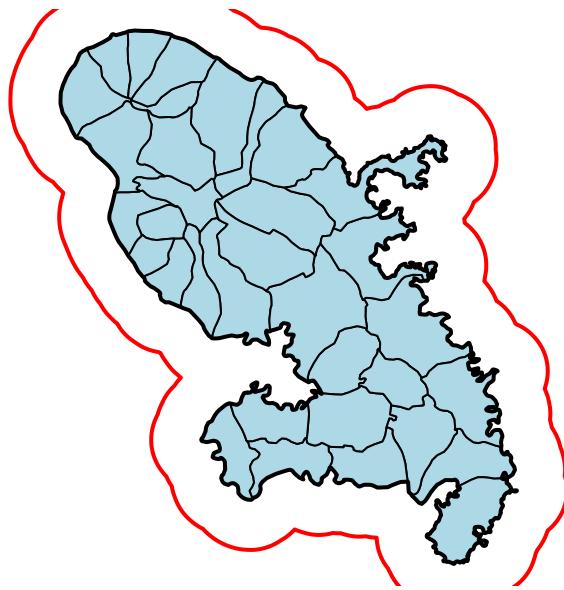
Agréger des polygones

```
mtq_u <- st_union(mtq)
plot(st_geometry(mtq), col="lightblue")
plot(st_geometry(mtq_u), add=T, lwd=2, border = "red")
```



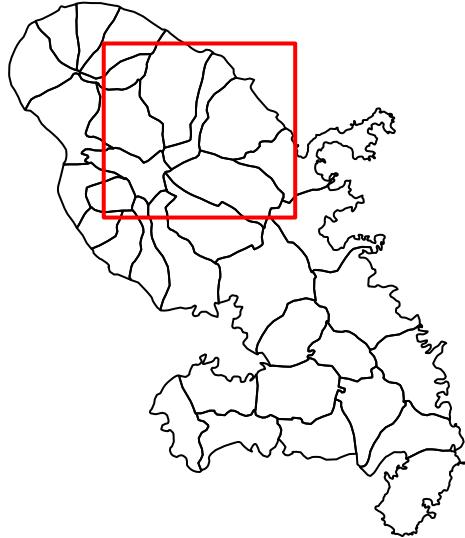
Construire une zone tampon

```
mtq_b <- st_buffer(x = mtq_u, dist = 5000)
plot(st_geometry(mtq), col="lightblue")
plot(st_geometry(mtq_u), add=T, lwd=2)
plot(st_geometry(mtq_b), add=T, lwd=2, border = "red")
```



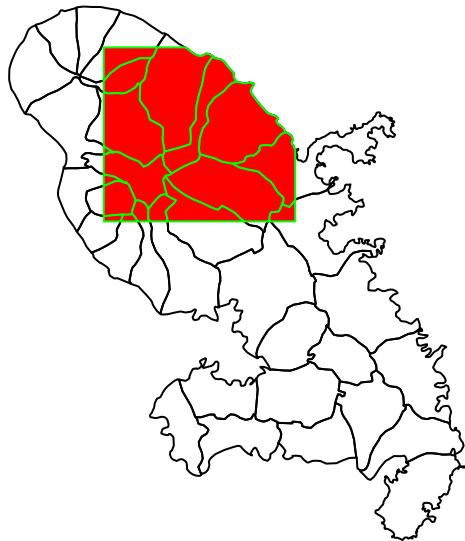
Réaliser une intersection

```
m <- rbind(c(700015,1624212), c(700015,1641586), c(719127,1641586), c(719127,1624212), c(700015,1624212))
p <- st_sf(st_sfc(st_polygon(list(m))), crs = st_crs(mtq))
plot(st_geometry(mtq))
plot(p, border="red", lwd=2, add=T)
```



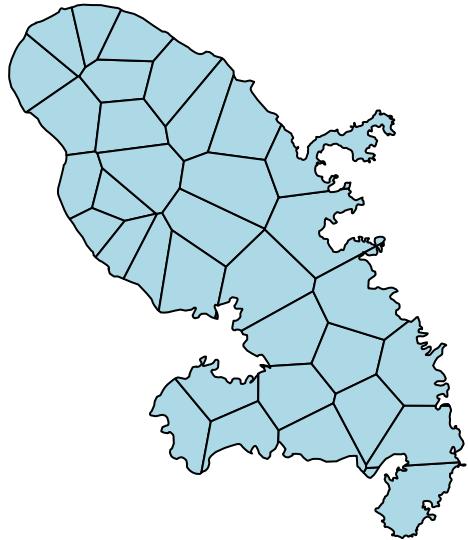
Réaliser une intersection

```
mtq_z <- st_intersection(x = mtq, y = p)
plot(st_geometry(mtz))
plot(st_geometry(mtz_z), col="red", border="green", add=T)
```



Construire des polygones de Voronoi google: “st\_voronoi R sf” (<https://github.com/r-spatial/sf/issues/474> & <https://stackoverflow.com/questions/45719790/create-voronoi-polygon-with-simple-feature-in-r>)

```
mtq_v <- st_voronoi(x = st_union(mtz_c))
mtq_v <- st_intersection(st_cast(mtz_v), st_union(mtz))
mtq_v <- st_join(x = st_sf(mtz_v), y = mtq_c, join=st_intersects)
mtq_v <- st_cast(mtz_v, "MULTIPOLYGON")
plot(st_geometry(mtz_v), col='lightblue')
```



## 1.5 Le package `raster`

# Chapter 2

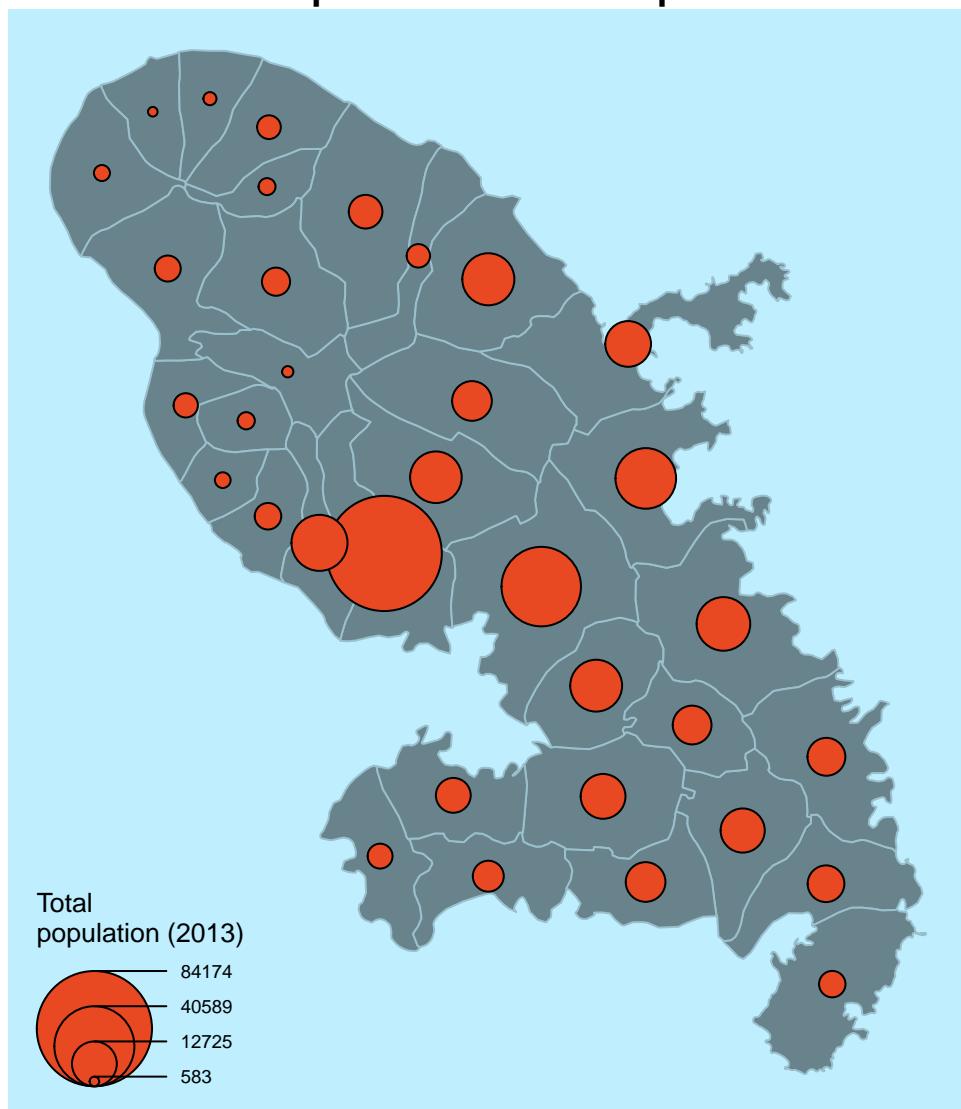
## Cartographie thématique

### 2.1 Le package `cartography`

#### 2.1.1 Symboles proportionnels

```
library(cartography)
library(sf)
# Import des données
mtq <- st_read(system.file("shape/martinique.shp", package="cartography"))
# Communes
plot(st_geometry(mtq), col = "lightblue4", border = "lightblue3",
     bg = "lightblue1")
# Symboles proportionnels
propSymbolsLayer(x = mtq, var = "P13_POP",
                  legend.title.txt = "Total\\npopulation (2013)")
# Titre
title(main = "Population en Martinique")
```

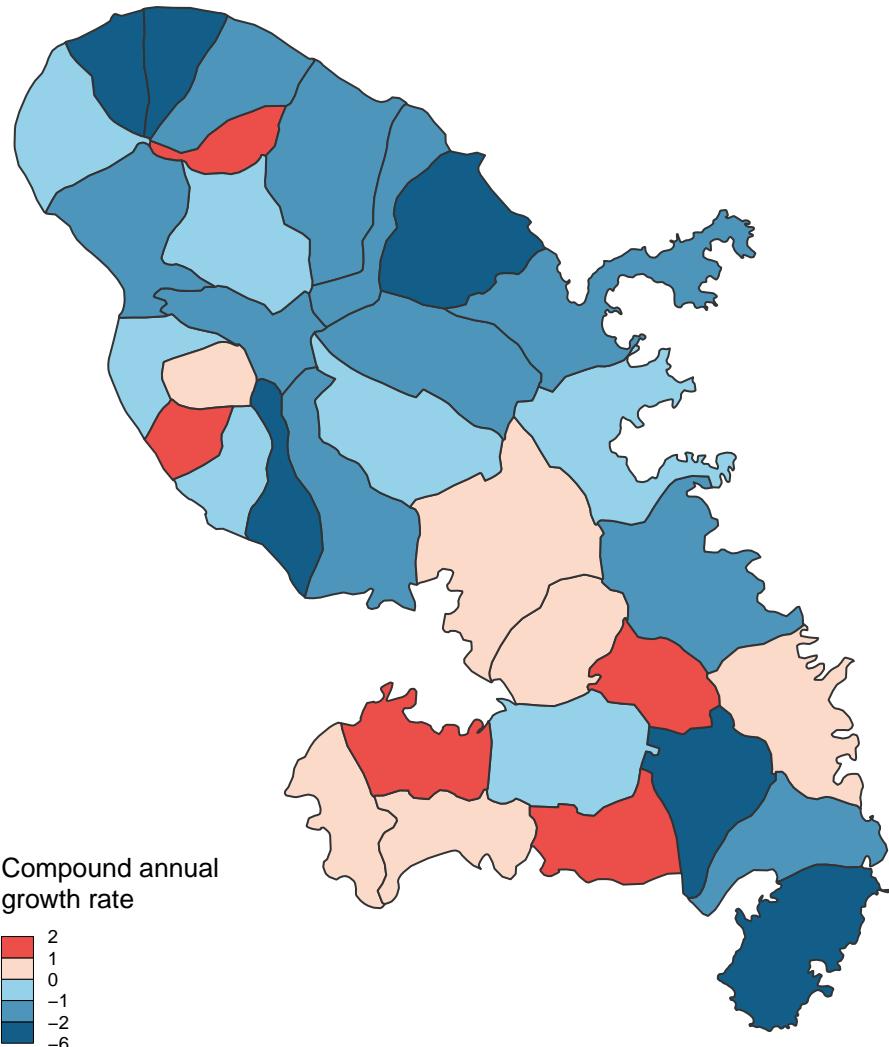
## Population en Martinique



### 2.1.2 Carte choroplèthe

```
mtq$cagr <- (((mtq$P13_POP / mtq$P08_POP)^(1/4)) - 1) * 100
choroLayer(x = mtq, var = "cagr", breaks = c(-6.14,-2,-1,0,1,2),
            col = c("#135D89", "#4D95BA", "#96D1EA", "#FCDACA", "#EC4E49"),
            legend.title.txt = "Compound annual\ngrowth rate")
title(main = "Evolution de la population")
```

## Evolution de la population

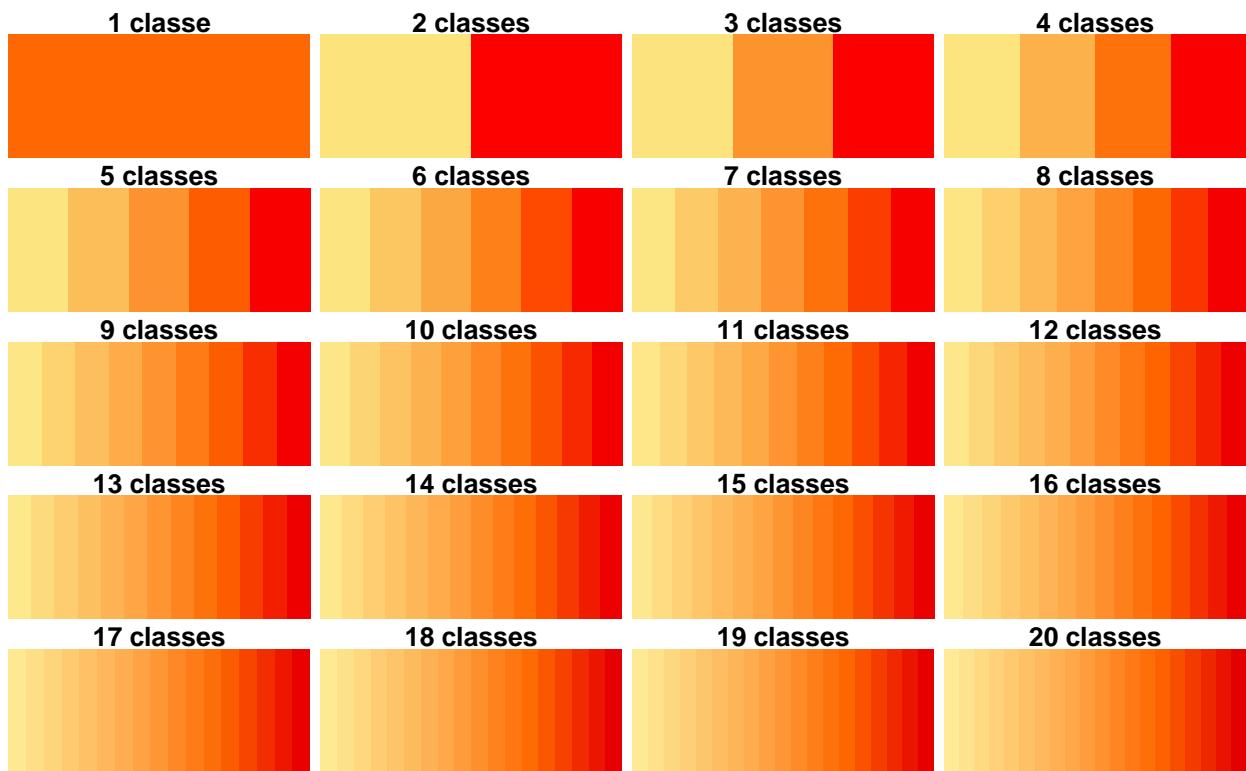


## 2.2 Palettes de couleurs

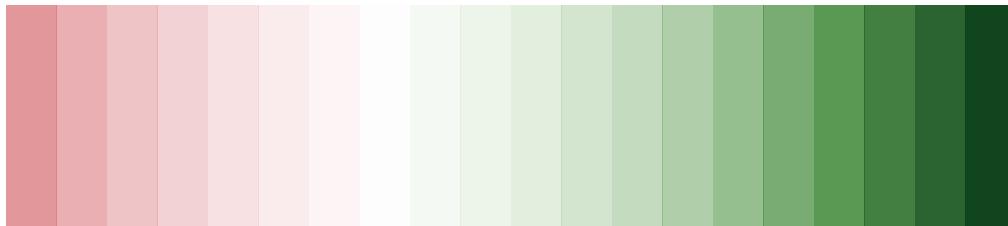
```
display.carto.all(20)
```



```
display.carto.pal("orange.pal")
```



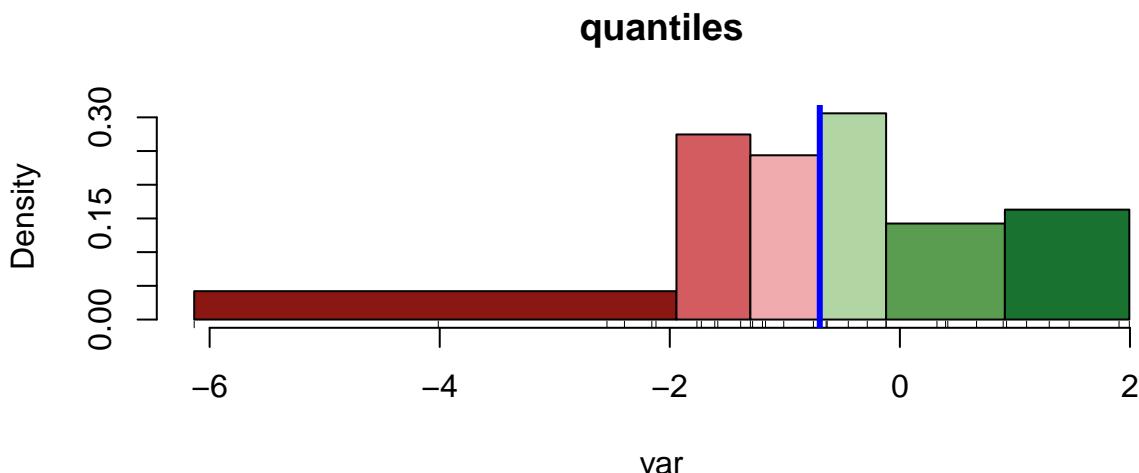
```
mypal <- carto.pal(pal1 = "wine.pal", n1 = 7, pal2 = "green.pal", n2 = 12,
                     middle = TRUE, transparency = TRUE)
k <- length(mypal)
image(1:k, 1, as.matrix(1:k), col=mypal, xlab = paste(k, " classes", sep=""),
      ylab = "", xaxt = "n", yaxt = "n", bty = "n")
```



20 classes

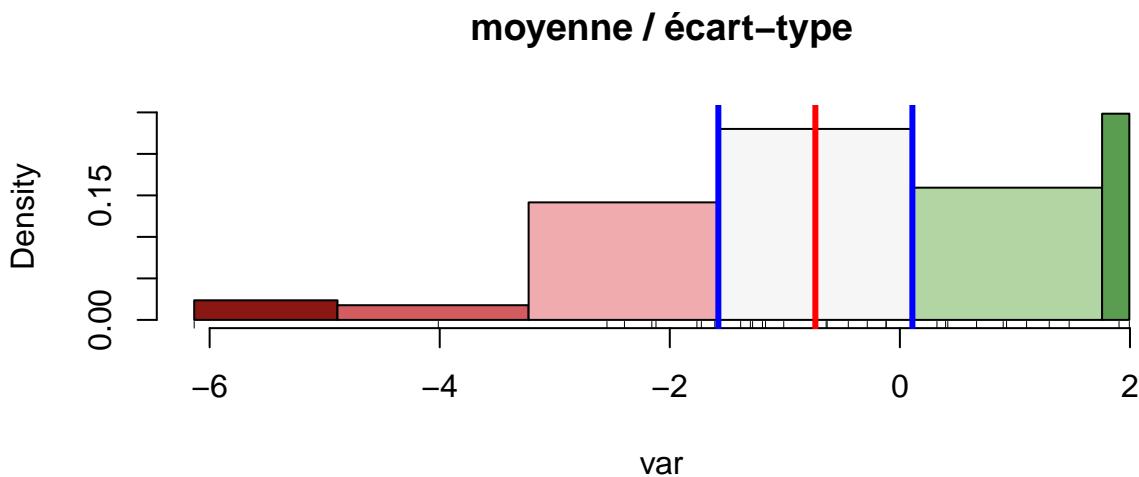
## 2.3 Discrétilisations

```
var <- mtq$cagr
moy <- mean(var)
med <- median(var)
std <- sd(var)
# Quantile intervals
breaks <- getBreaks(v = var, nclass = 6, method = "quantile")
hist(var, probability = TRUE, breaks = breaks, main="quantiles",
     col = carto.pal(pal1 = "wine.pal", 3, "green.pal", 3))
rug(var)
abline(v = med, col = "blue", lwd = 3)
```



```
# Mean and standard deviation (msd)
breaks <- getBreaks(v = var, method = "msd", k = 1, middle = TRUE)
```

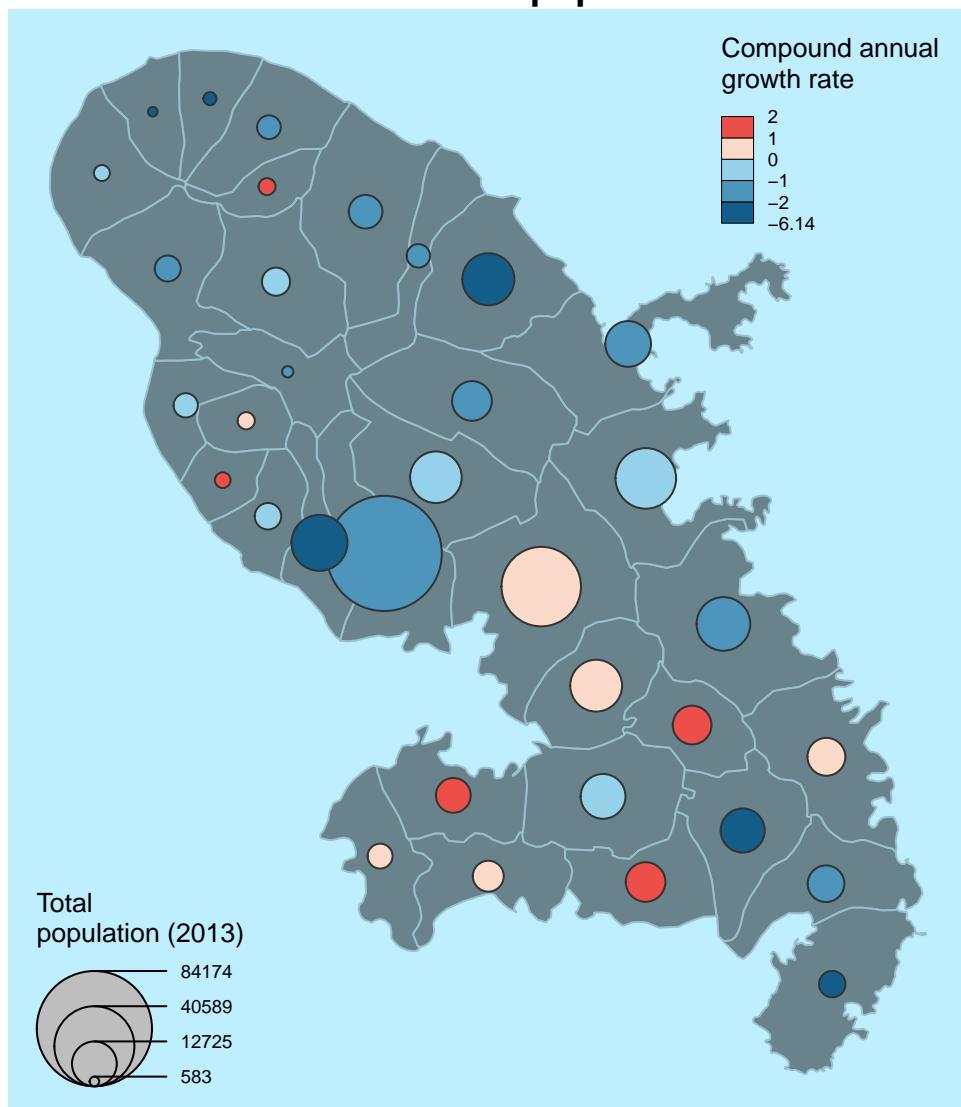
```
hist(var, probability = TRUE, breaks = breaks, main="moyenne / écart-type",
  col = carto.pal(pal1 = "wine.pal", 3, "green.pal", 2, middle = TRUE))
rug(var)
abline(v = moy, col = "red", lwd = 3)
abline(v = moy + 0.5 * std, col = "blue", lwd = 3)
abline(v = moy - 0.5 * std, col = "blue", lwd = 3)
```



## 2.4 Combinaisons

```
plot(st_geometry(mtq), col = "lightblue4",
  border = "lightblue3", bg = "lightblue1")
propSymbolsChoroLayer(x = mtq, var= "P13_POP",
  legend.var.title.txt = "Total\npopulation (2013)",
  var2 = "cagr", legend.var.pos = "bottomleft",
  breaks = c(-6.14, -2, -1, 0, 1, 2),
  col = c("#135D89", "#4D95BA", "#96D1EA", "#FCDACA", "#EC4E49"),
  legend.var2.title.txt = "Compound annual\ngrowth rate")
# Title
title(main = "Evolution de la population")
```

## Evolution de la population



## 2.5 Labels

```

plot(st_geometry(mtq), col = "darkseagreen3", border = "darkseagreen4",
      bg = "#A6CAE0")
labelLayer(x = mtq, txt = "LIBGEO", col= "black", cex = 0.7, font = 4,
            halo = TRUE, bg = "white", r = 0.1, overlap = FALSE,
            show.lines = FALSE)

```

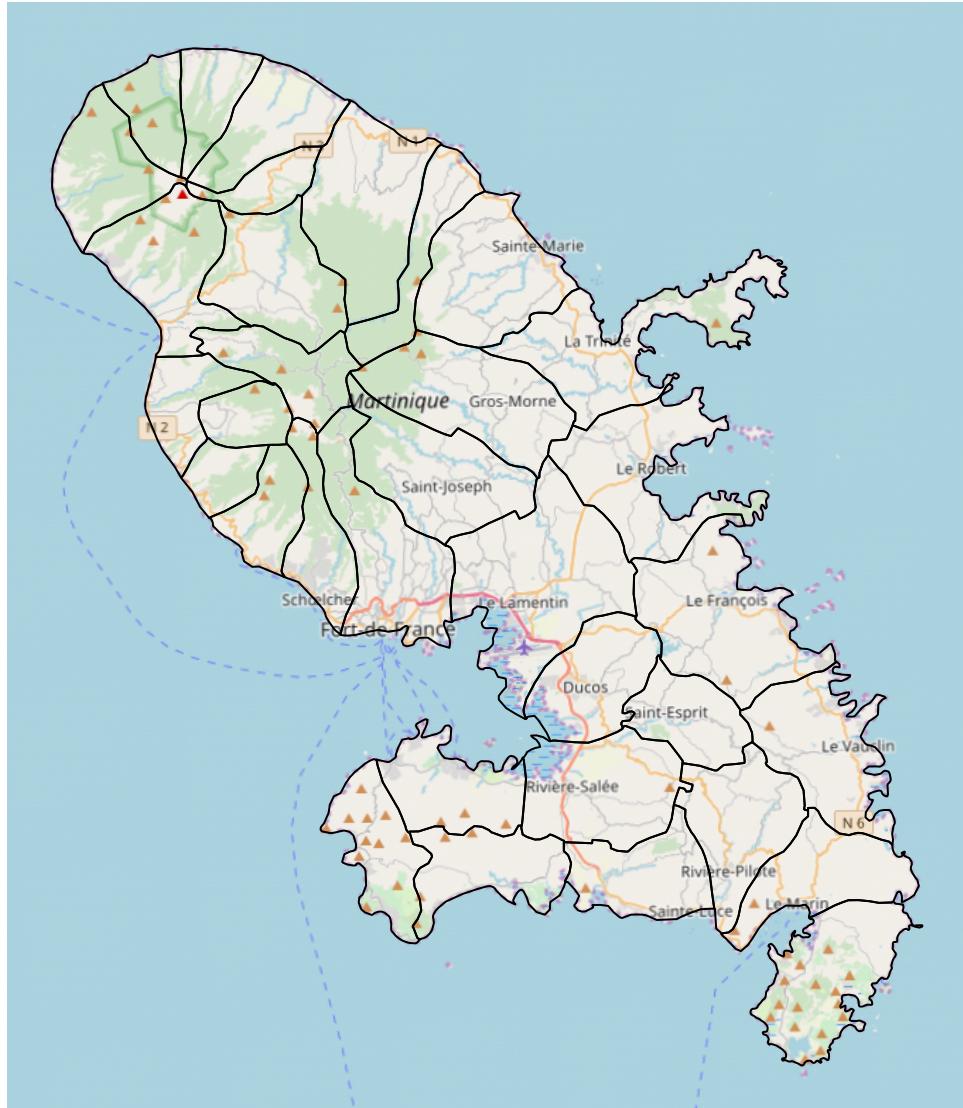


## 2.6 Les données OSM

### 2.6.1 Données vectorielles

### 2.6.2 Données raster

```
tiles <- getTiles(x = mtq, type = "osm", crop=T, zoom = 11)
tilesLayer(tiles)
plot(st_geometry(mtq), add=T)
```



## 2.7 Cartographie interactive

leaflet / mapview

## 2.8 Géocodage d'adresses

## 2.9 Créditation de cartons



# Chapter 3

## Cartographie thématique avancée

### 3.1 Les anamorphoses

Voir : Les anamorphoses cartographiques  
*Nicolas Lambert, 2015*

“L’anamorphose classique est une représentation des États (ou de mailles quelconques) par **des rectangles ou des polygones quelconques** en fonction d’une **quantité** qui leur est rattaché.”

“On s’efforce de **garder l’arrangement général** des mailles ou la silhouette du continent.”

*Brunet, R., Ferras, R., & Théry, H. (1993). Les mots de la géographie: dictionnaire critique (No. 03) 911 BRU).*

#### 3.1.1 Les cartogrammes de Dorling

La taille des cercles est proportionnelle à une variable.

La position des cercles est définie selon les positions de départ.

*Dorling, Daniel (1996): Area Cartograms: Their Use and Creation, Concepts and Techniques in Modern Geography (CATMOG), 59*

### 3.1.2 Le principe

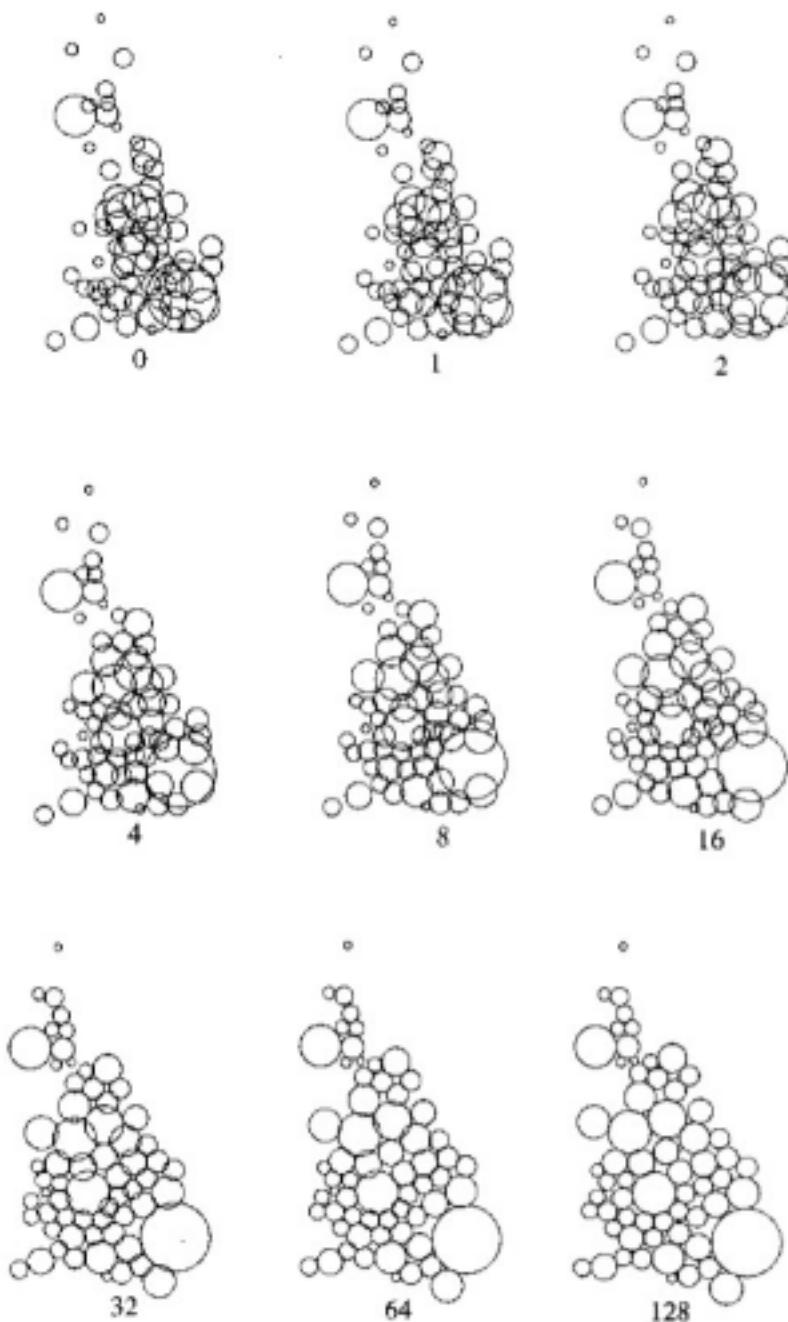
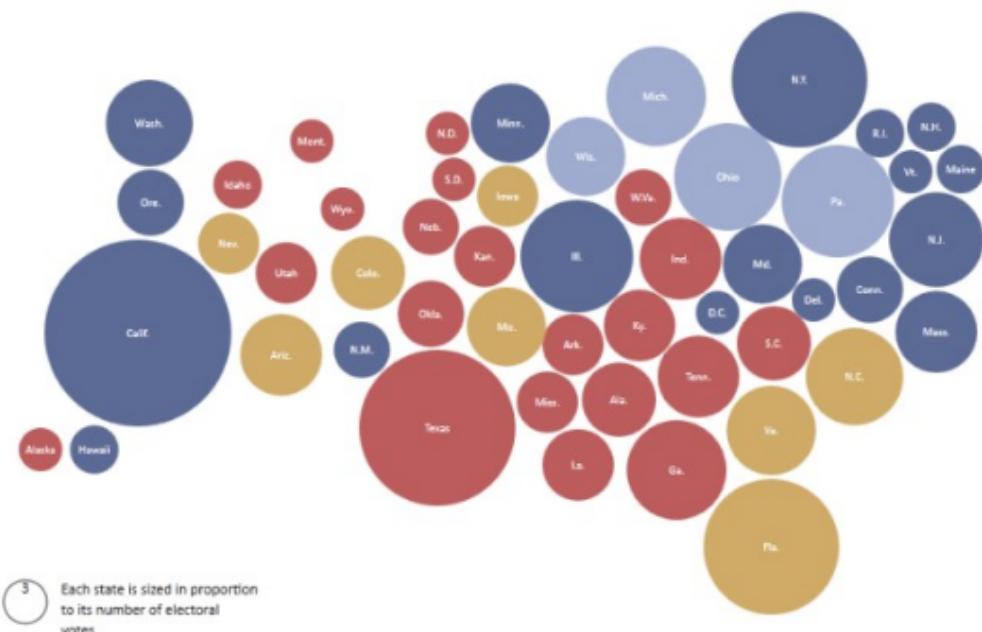
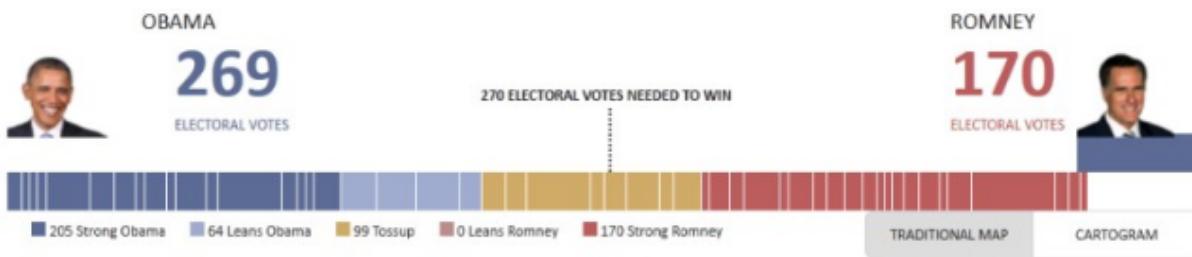


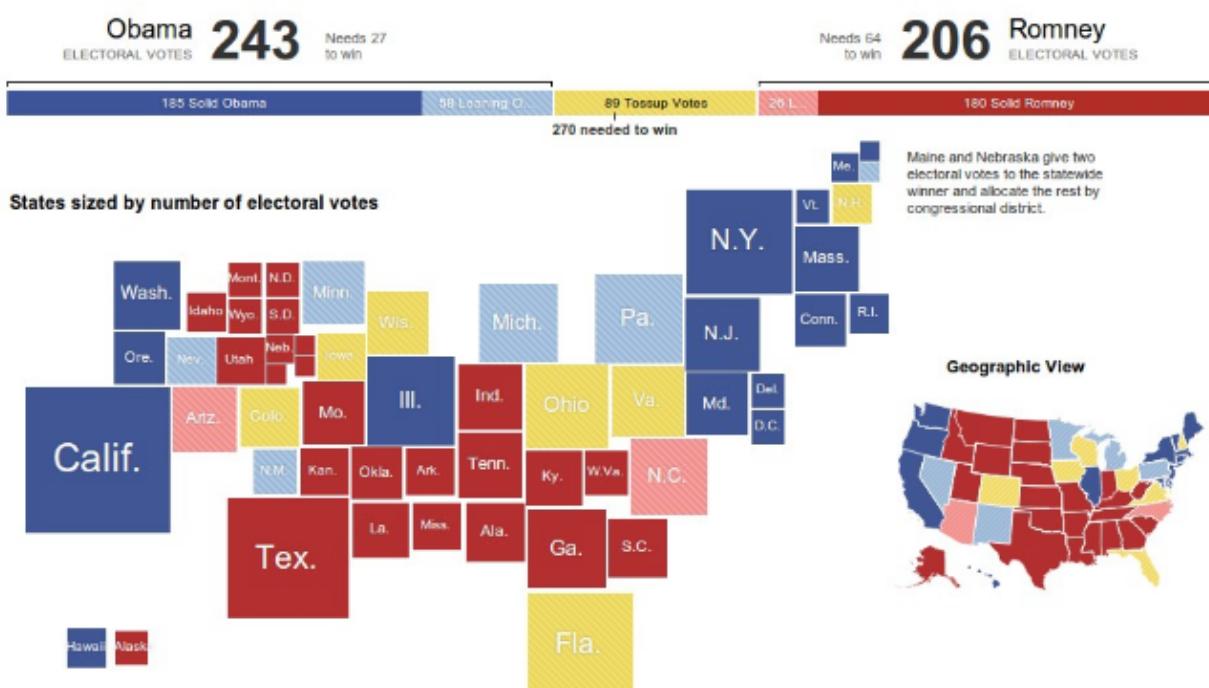
Figure 16: British counties transformed after 128 iterations of the circular algorithm (Author).



### 3.1.3 Exemple



<http://elections.huffingtonpost.com/2012/romney-vs-obama-electoral-map#map>



<http://elections.nytimes.com/2012/ratings/electoral-map>

### 3.1.4 Précautions d'emploi

- On identifie assez mal l'espace

On peut nommer les cercles pour se repérer

On peut s'aider de la couleur pour faire des clusters et mieux identifier les blocks géographiques

+ La perception de la quantité est très bonne.

Les tailles de cercles sont vraiment comparables

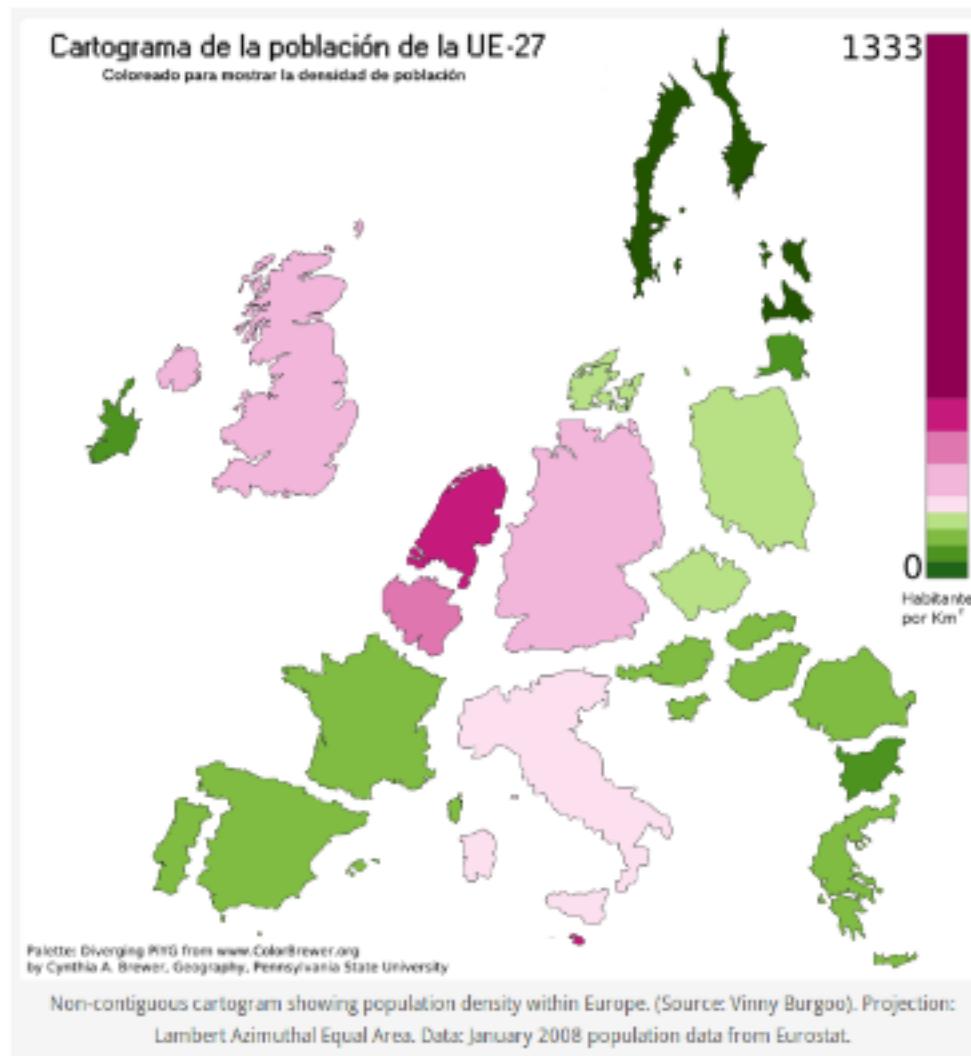
### 3.1.5 Les cartogrammes non continus

La taille des polygones est proportionnelle à une variable.

L'agencement des polygones les uns par rapport aux autres est conservée.

La forme des polygones est ressemblante.

### 3.1.6 Exemple



### 3.1.7 Précautions d'emploi

- Non contigu, la topologie est perdue.
- + La conservation de la forme des polygones est optimisée.

### 3.1.8 Les cartogrammes continus

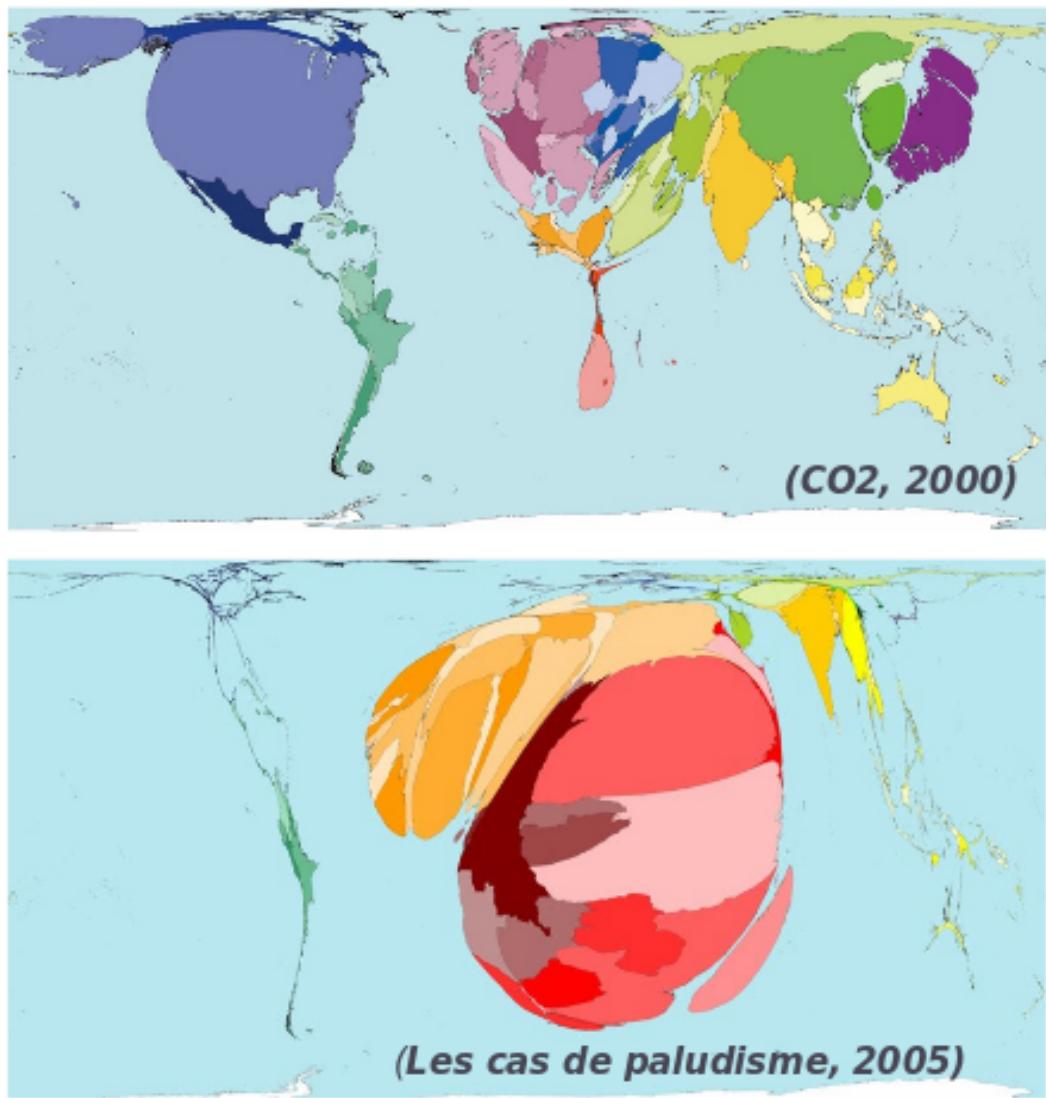
La taille des polygones est proportionnelle à une variable.

L'agencement des polygones les uns par rapport aux autres est conservée.

Pour conserver la contiguïté, la forme des polygones est fortement transformée.

### 3.1.9 Exemple

**WorldMapper** : <http://www.worldmapper.org/>



### 3.1.10 Précautions d'emploi

- Par rapport aux anamorphoses non contigues, la forme des polygones est fortement distordue.
- + C'est une "vraie carte de géographie" : la topologie et la contiguïté sont conservées.

### 3.1.11 Interêts des anamorphoses

Représentation cartographique perçue comme **innovante** (même si la méthode date de 40 ans)  
Image très généralisée qui rend bien compte des **quantités** et des **gradiants**.

Une vraie image de **communication** : provoque, suscite l'intérêt, véhicule un message fort, interpelle.

### 3.1.12 Faiblesses des anamorphoses

Perte des **repères visuels** (difficile de retrouver son pays, ou sa région sur la carte).

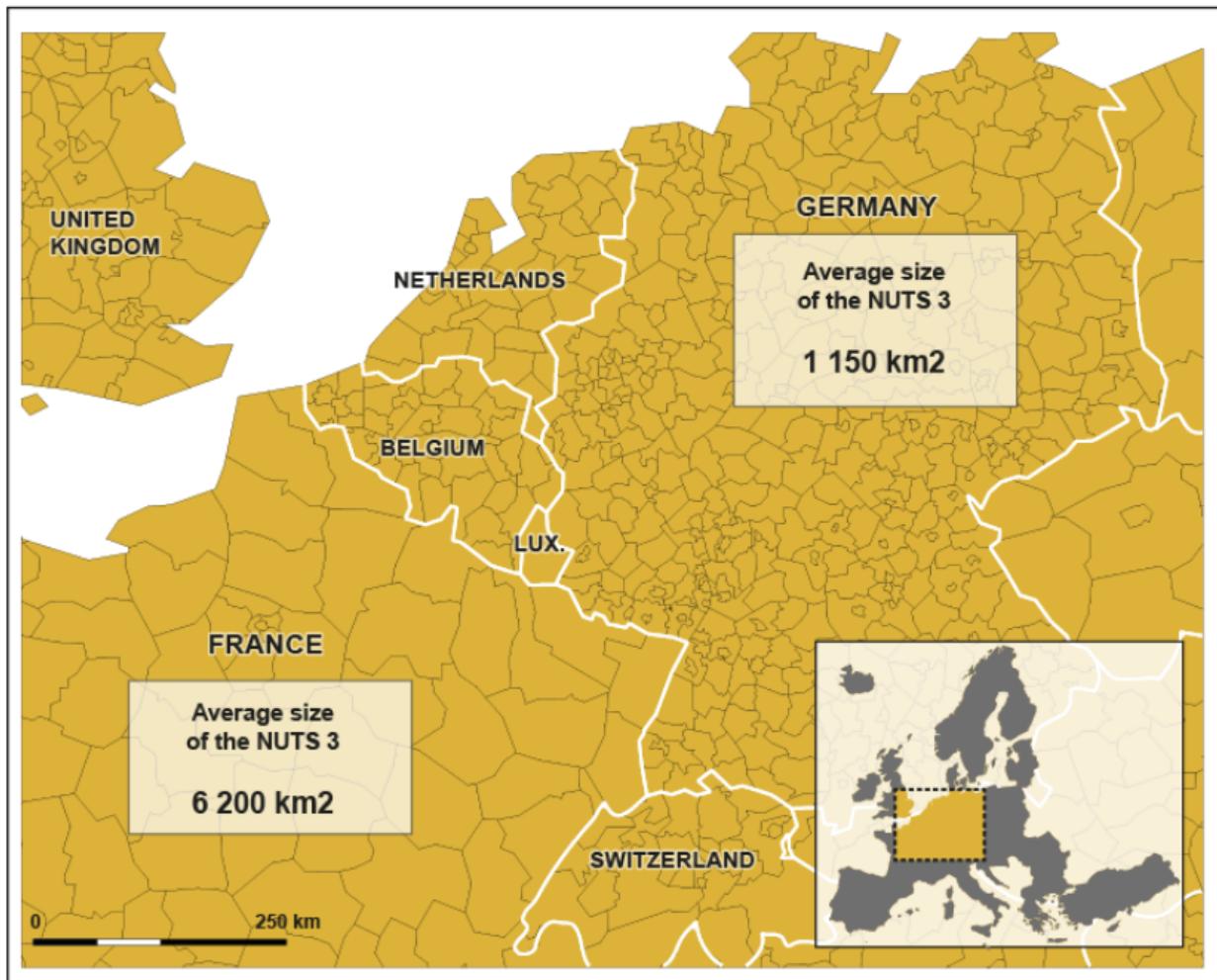
Ne permet pas de connaître les **situations locales**.

Demande un **effort de lecture**.

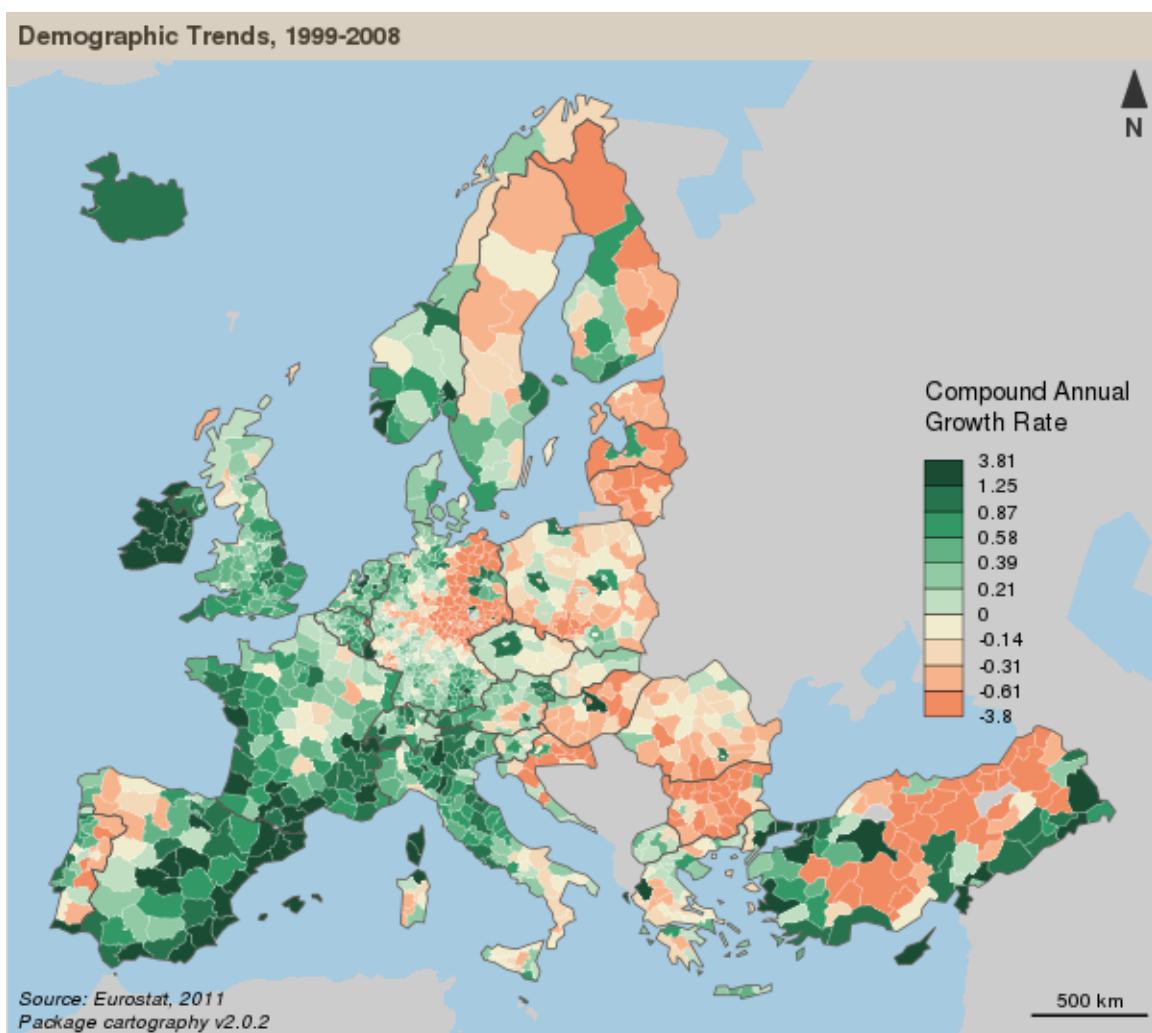
Gestion des données manquantes

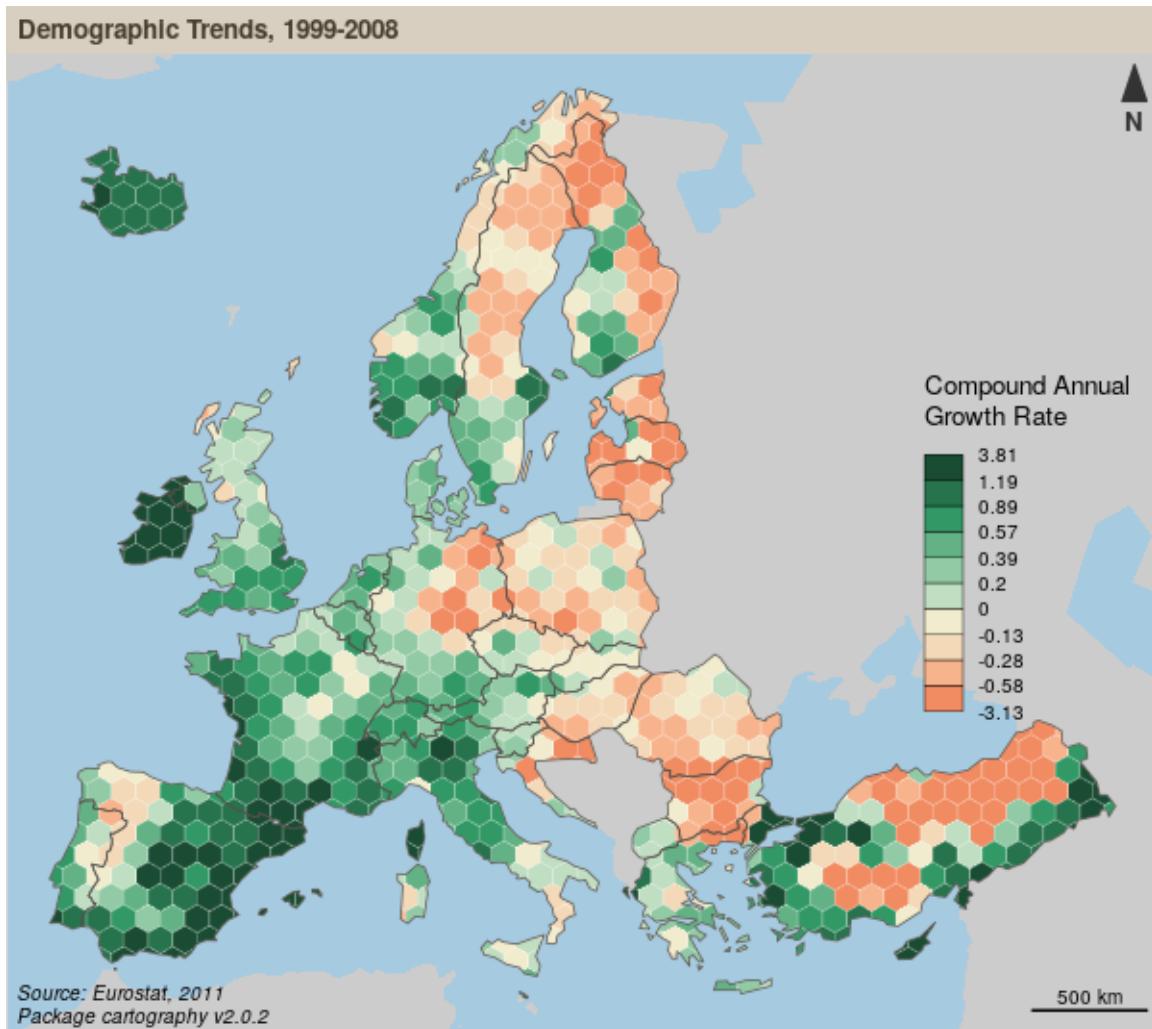
## 3.2 Les grilles régulières

Par une série d'opération SIG assez simple il est possible de transformer des données d'un maillage initial vers un maillage régulier plus neutre et plus simple.



### 3.2.1 Exemples





### 3.2.2 Précautions d'emploi

- Perte de précision, maillage sans signification.

La version simple (au prorata de la surface), implique une équirépartition du phénomène dans chaque unités.

+ Permet la comparaison de maillages différents, à plusieurs dates ou de différentes sources.

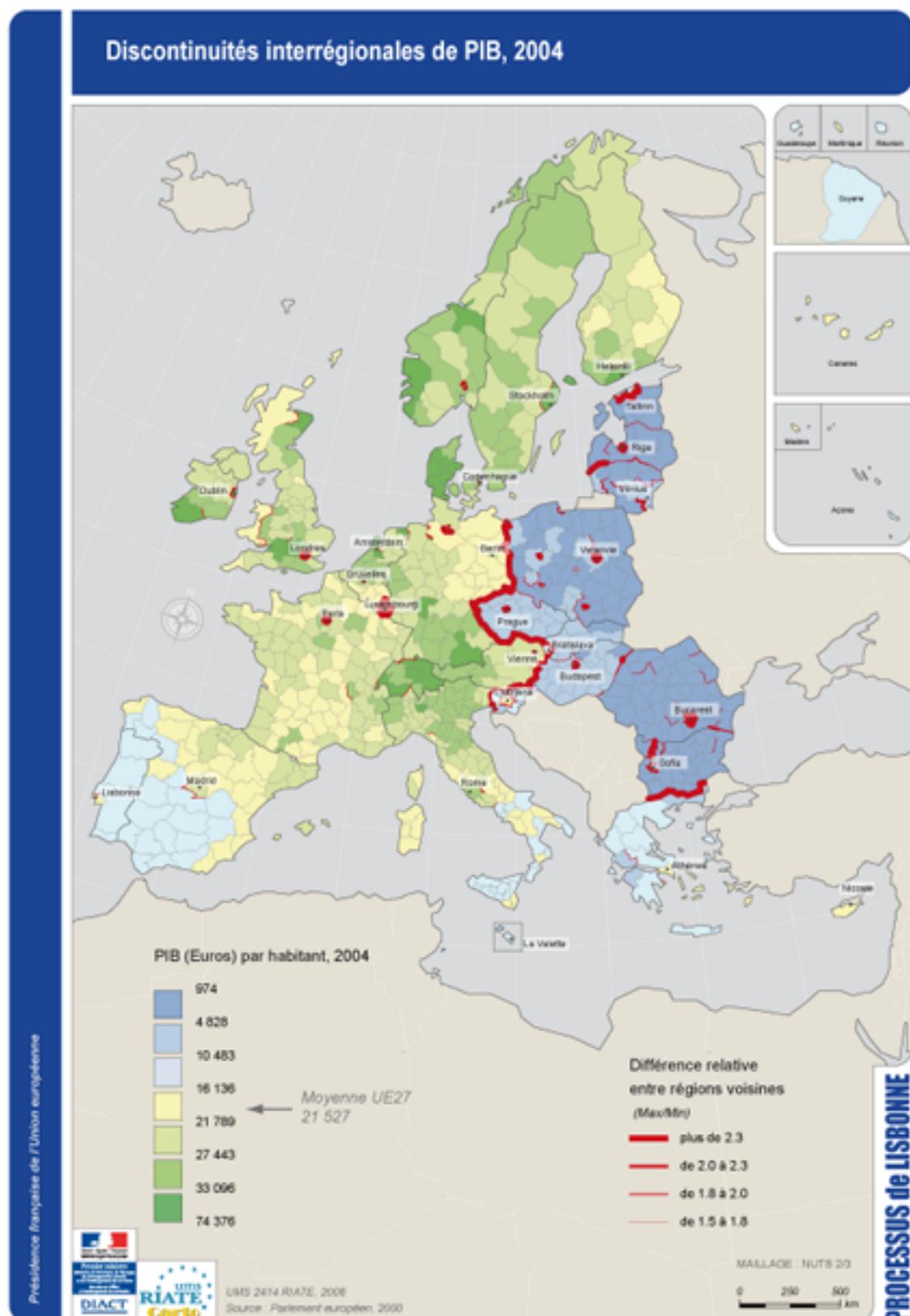
## 3.3 Les discontinuités

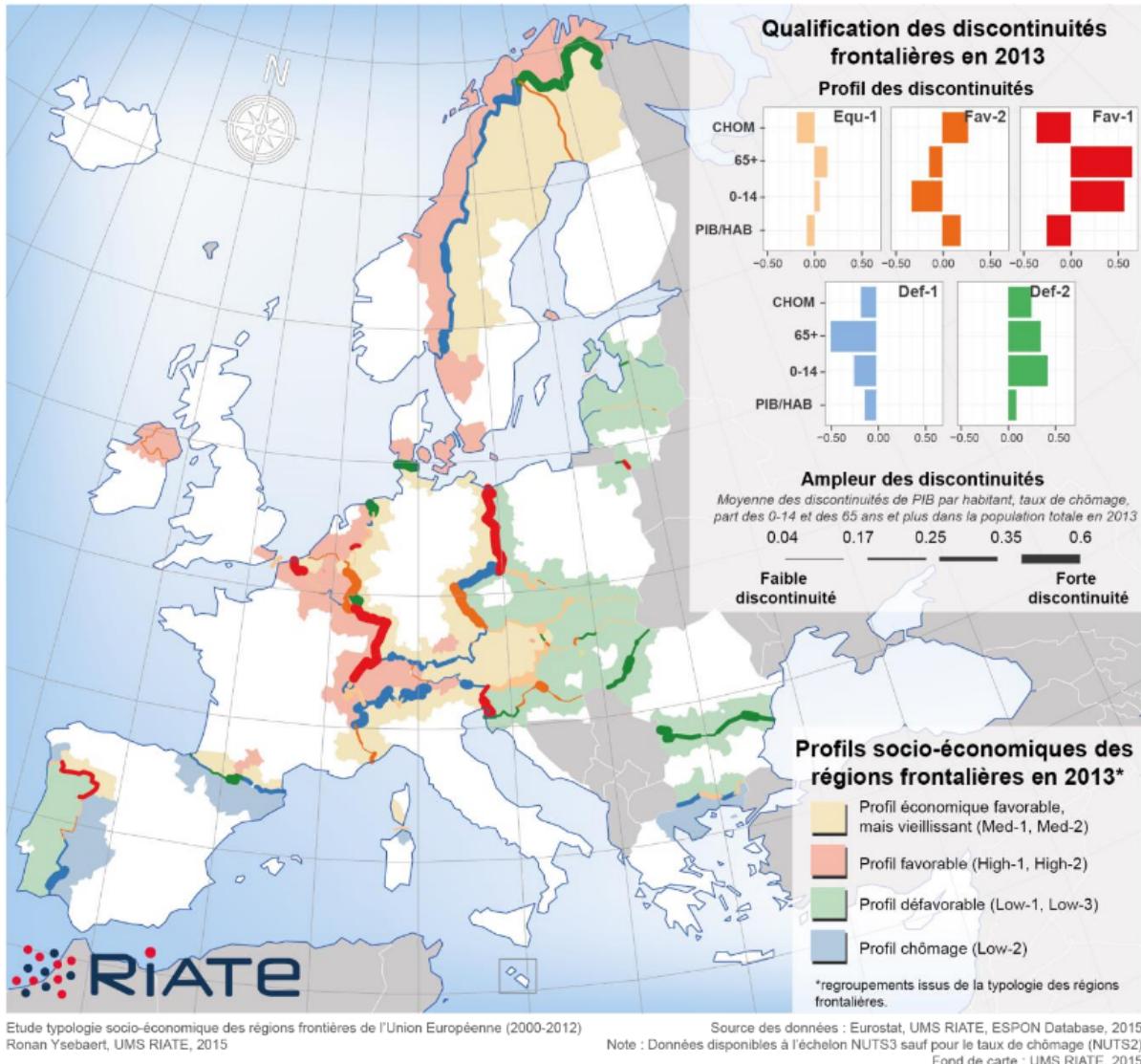
Ce type de représentation permet de souligner cartographiquement les discontinuités territoriales d'un phénomène.

L'accent est porté sur ce qui distingue des territoires.

Pour chaque frontière nous calculons le rapport ou la différence des valeurs des polygones de part et d'autre. Puis nous représentons la frontière par un trait d'autant plus épais que la différence est forte.

Il est souvent bénéfique de coupler ce type de représentation à une représentation choroplète (pour comprendre le sens des discontinuités). #### Exemples





### 3.3.1 Précautions d'emploi

- Ces cartes ne sont pas évidentes à paramétriser.
- Le choix des critères (seuil, type de différences...) va fortement influencer la représentation.
- En fonction du maillage la lisibilité peut être faible.
- + Représentation très puissante pour montrer les inégalités.

## 3.4 Les lissages

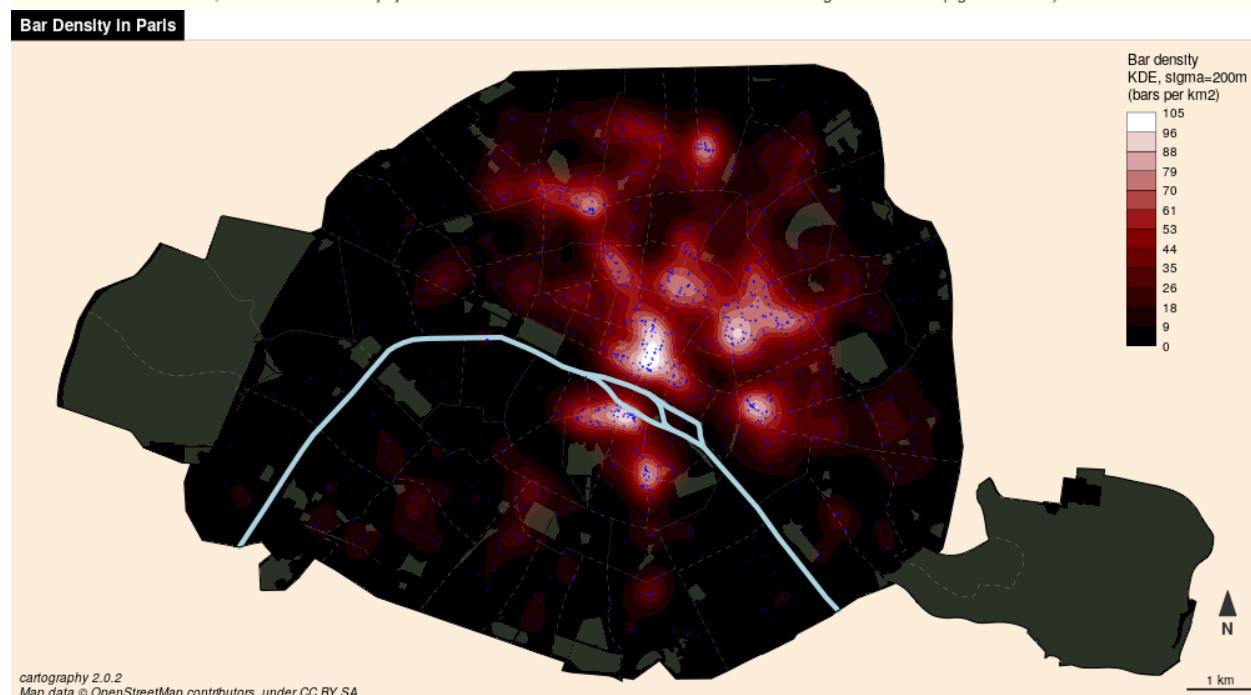
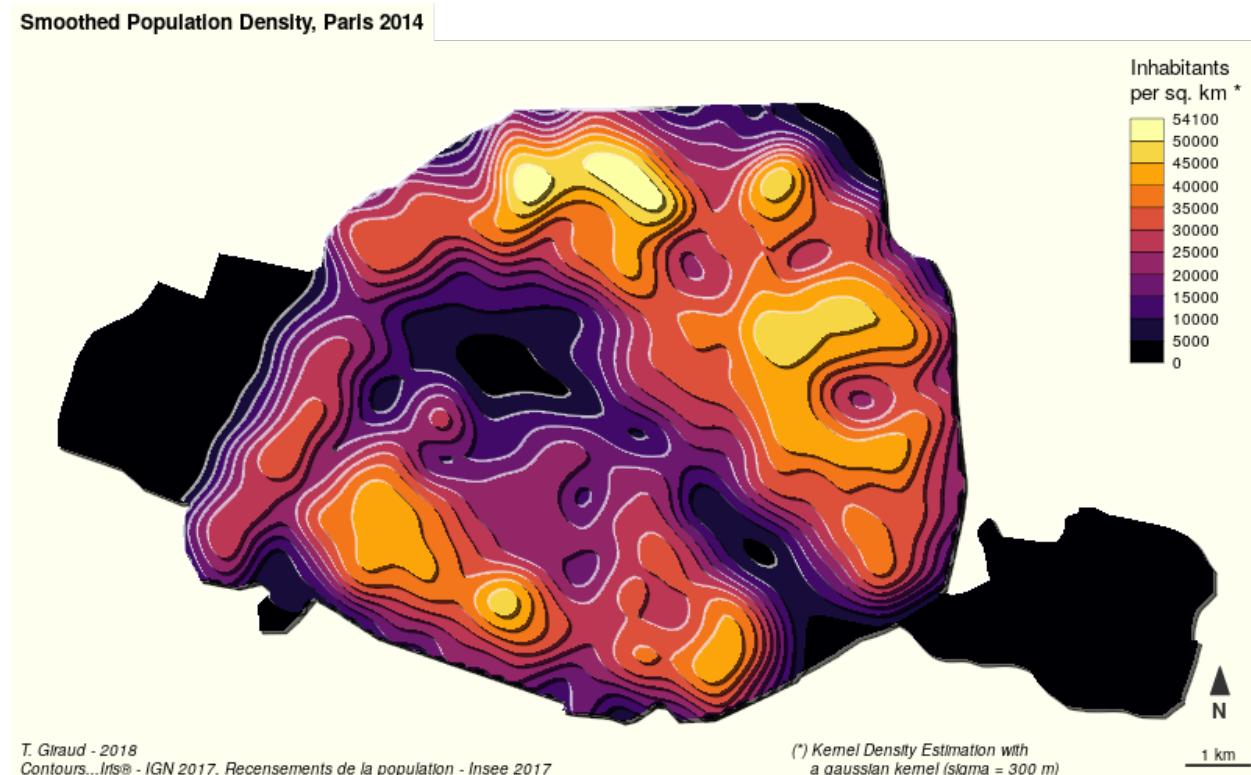
L'idée principale du lissage est de filtrer l'information pour révéler des structures spatiales sous-jacentes.

C'est un ensemble de méthodes qui consistent à affecter aux points que l'on observe une valeur prenant en compte les valeurs de leur voisinage.

Il existe plusieurs méthodes de lissage (kde, potentiels...) plus ou moins paramétrables.

Cette méthode permet de passer représentations ponctuelles à une représentation continu

### 3.4.1 Exemples



### 3.4.2 Précautions d'emploi

- Il est difficile de paramétrier correctement les fonctions de lissages.  
Elles doivent s'appuyer sur des hypothèses de comportement dans l'espace.  
La compréhension par un public large n'est pas évidente, il faut alors simplifier les légendes, la présentation de la méthode.
- + Permet de faire ressortir des phénomènes spatiaux sous-jacents invisibles directement.  
Les cartes produites attirent l'oeil par leur originalité.  
Cette méthode permet de passer d'une représentation ponctuelle ou discontinue (dans un maillage) à une représentation continue s'affranchissant des maillages existants.

### 3.4.3 KDE

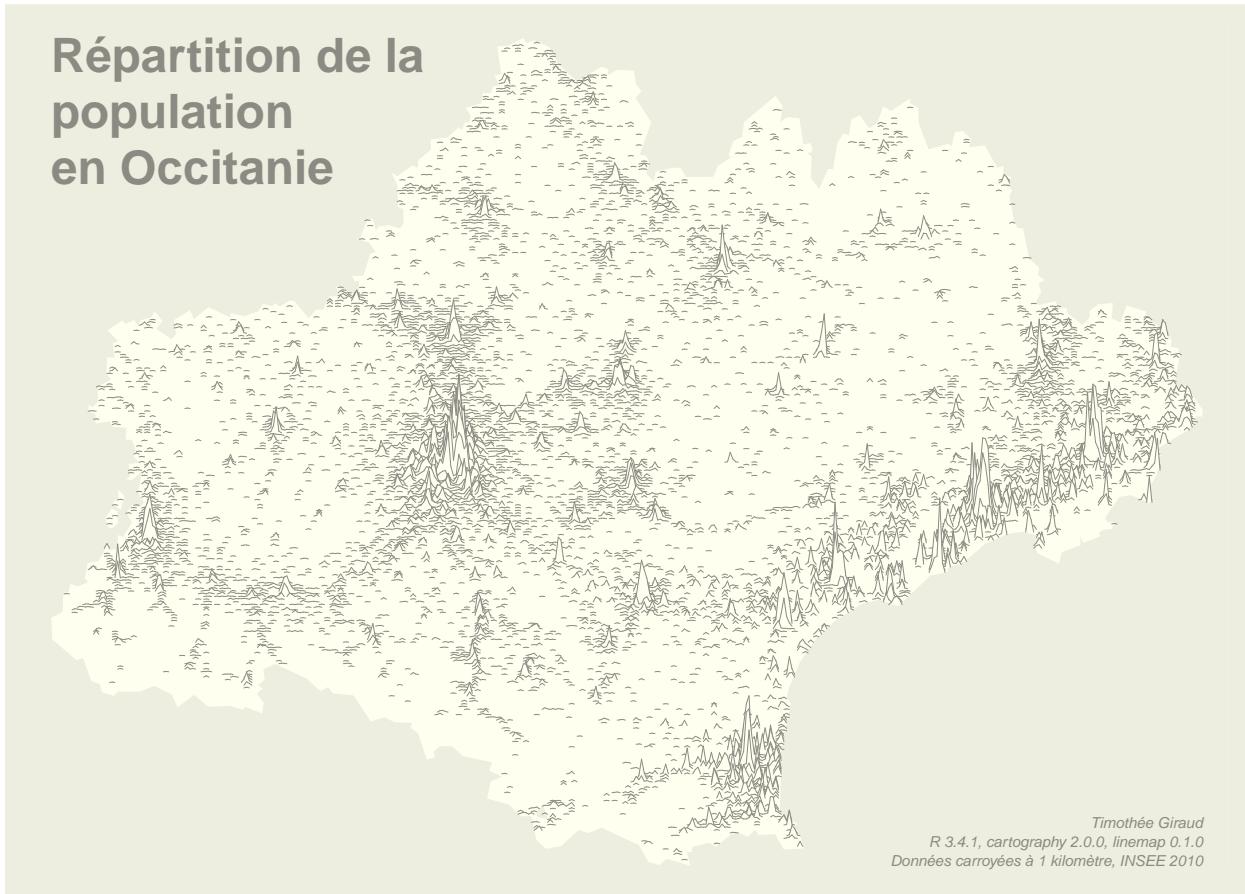
#### 3.4.4 Stewart

Vignette du package SpatialPosition

## 3.5 3D

### 3.5.1 linemap

```
library(linemap)
library(sf)
data("popOcc")
data("occitanie")
opar <- par(mar=c(0,0,0,0), bg = "ivory2")
bb <- st_bbox(occitanie)
plot(st_geometry(occitanie), col="ivory1", border = NA)
linemap(x = popOcc, var = "pop", k = 2.5, threshold = 50,
        col = "ivory1", border = "ivory4", lwd = 0.6, add = TRUE)
text(x = bb[1], y = bb[4], adj = c(0,1),
     labels = "Répartition de la population en Occitanie",
     col = "ivory4", font = 2, cex = 1.8)
# add sources
mapsources <-"Timothée Giraud\nR 3.4.1, cartography 2.0.0, linemap 0.1.0\nDonnées carroyées à 1 kilomètre"
text(x = bb[3], y = bb[2], labels = mapsources,
     col = "ivory4", font = 3, adj = c(1,0), cex = 0.6 )
```



### 3.5.2 rayshader