

Données géospatiales et cartographie avec R

Nicolas Roelandt

vendredi 28 juin 2019



Intervenant

- Nicolas Roelandt
- Géomaticien
- membre de l'[OSGeo](#) et de l'[OSGeo-fr](#)



- Manipule de la data géo
 - avec QGIS, Python, PostGreSQL/PostGIS
 - et R ;)
- Cartographie

Contenu

1. Comprendre les données géospatiales
2. les packages R dédiés
3. Code Officiel Géographique avec R

1. Comprendre les données géospatiales

Data géo

Un monde de couches

- superposition de couches
- couches vectorielles
 - bâtiments
 - routes
 - parcellaire
 - etc

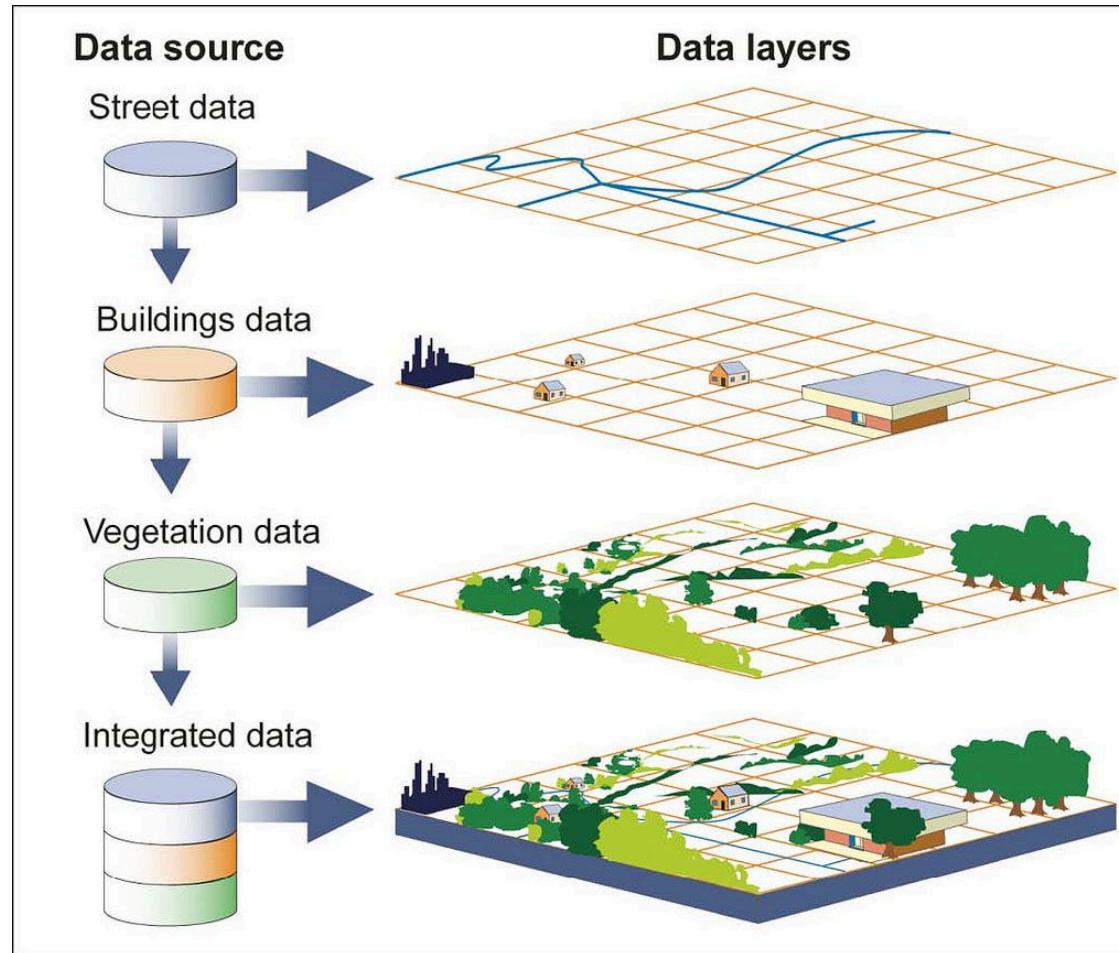
Data géo

Un monde de couches

- couches raster
 - imagerie satellites ou aérienne
 - élévation
 - fond de plan cartographique
 - etc

Data géo

Un monde de couches

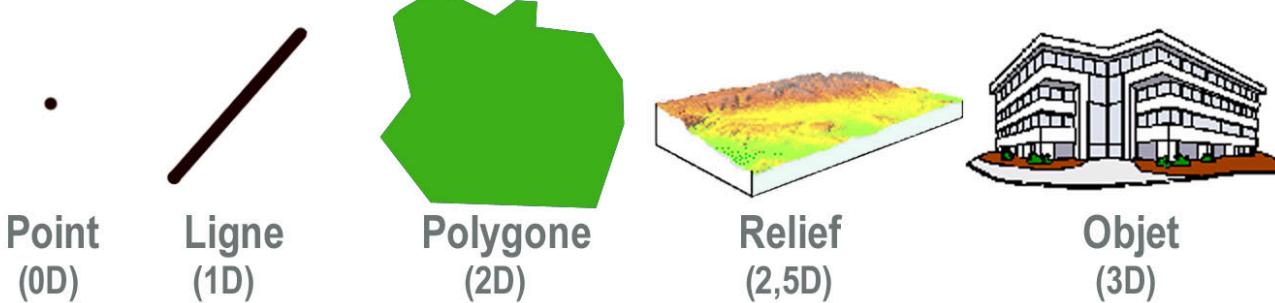


Data géo

Données vectorielles

Types

- points
- lignes
- polygones
- Multi-*
- Collections



Data géo

Données vectorielles

Formats

- shapefile
- (Geo)JSON
- CSV
- **GeoPackage**



Data géo

Données raster

"Données images où l'espace est divisé de manière régulière (en petits rectangles); à chaque petit rectangle (pixel) sont associées une ou plusieurs valeurs décrivant les caractéristiques de l'espace." [Vince - PortailSIG](#)

Data géo

Données raster

- Types

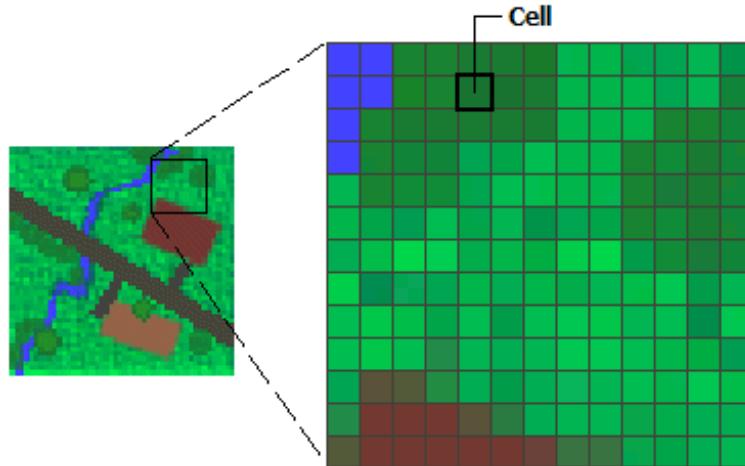
- Modèle Numérique de Terrain / Elevation
- imagerie visible et/ou multi-spectrale
- imagerie radar
- LIDAR
- NetCDF

- Formats

- (geo)TIFF
- JPEG
- ERDAS
- CSV

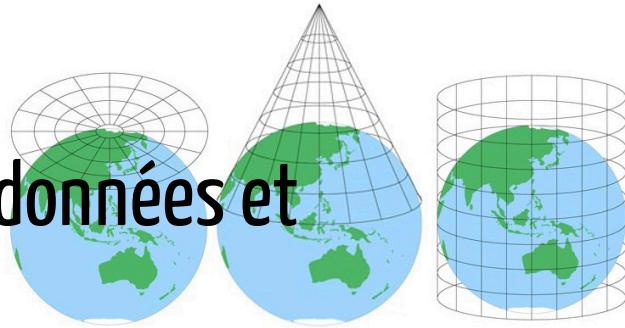
Data géo

Données raster



Data géo

Systèmes de référence de coordonnées et projections



- Formules mathématiques permettant de passer des coordonnées d'une sphère à un plan 2D (et inversement)

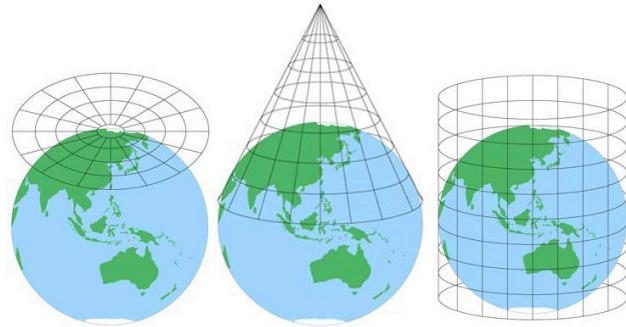
2 cas :

- Systèmes de coordonnées **géographiques**
 - Latitude : ϕ ,
 - Longitude : λ ,
 - (Hauteur ellipsoïdale : h)
- Systèmes de coordonnées **plans** (ou projetés)
 - Coordonnées X/Y ou E/N

Data géo

Types de projections

- 2 cas généraux :
 - *conforme*: conserve les angles
 - *équivalente*: conserve les surfaces
- Types courants :
 - projection cylindrique
 - projection conique
 - projection azimutale

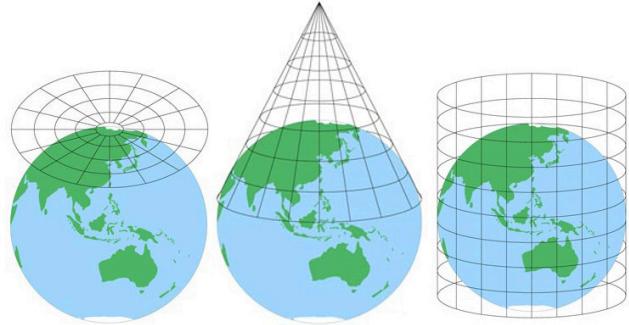


Pour aller plus loin : [Video "Why every world map is wrong" by Jay Foreman \(YT\)](#)

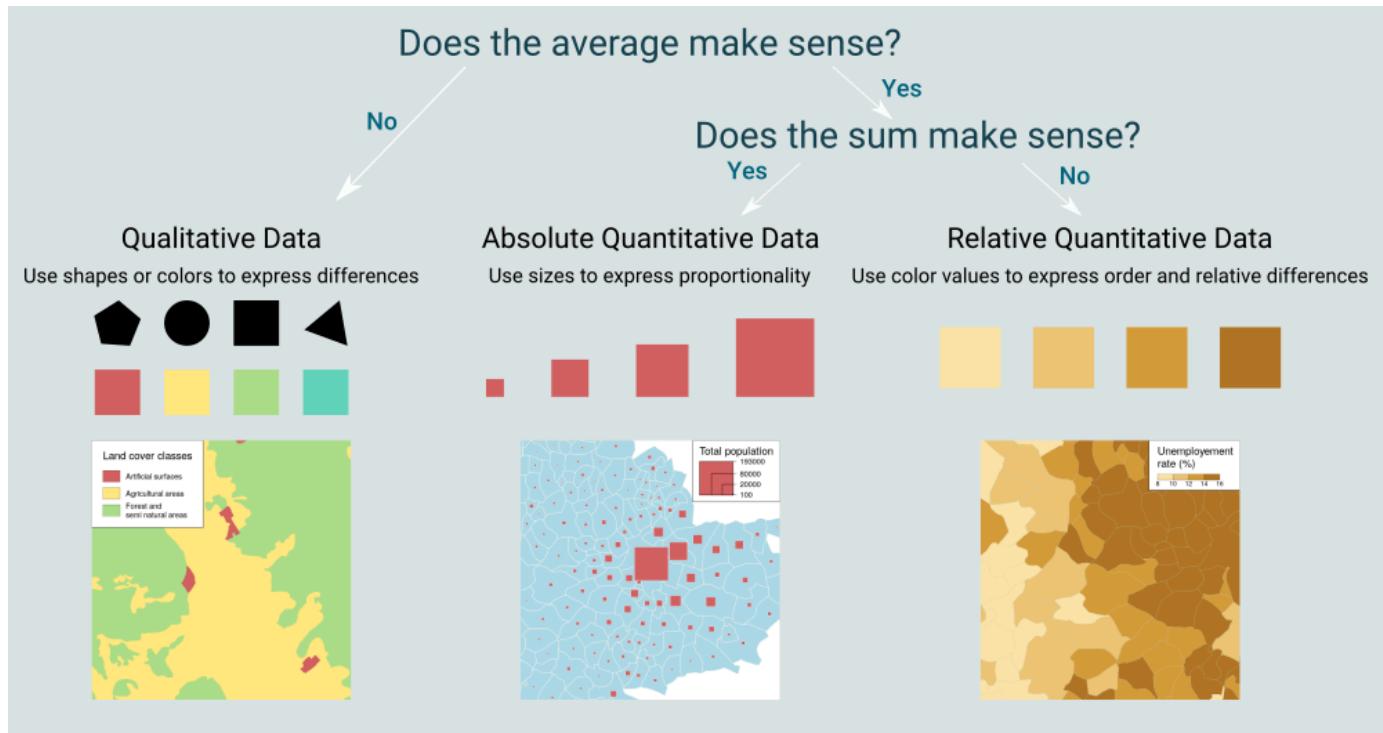
Data géo

SRC et projections

- En France métropolitaine :
 - RGF93/Lambert93, système légal 2D projeté : **EPSG:2154**
 - WGS84 (système GPS) : **EPSG:4326**
- Pour les DOM:
 - Systèmes UTM locaux
 - WGS84



Sémiologie graphique



(Illustration (c) by Thimothée Giraud - UMR RIATE/CNRS)

Où trouver de la data géo ?

- données IGN :
 - Open data :
 - <http://www.professionnels.ign.fr/donnees>
 - [adminexpress](#)
- données open data
 - data.gouv.fr
 - Portails open data locaux : [Paris](#), [Nantes Métropole](#), [Grand Lyon](#), etc.
- données enquêtes
 - géolocalisation GPS
 - Adresses => [géocodage](#)

Et R dans tout ça ?

De nombreux packages peuvent manipuler de la data géo

{sf}

{ggplot}

{cartography}

...

{sf}



(Illustration (c) by Allison Horst)

{sf}

Intérêts



- hérite de {sp}, {rgdal}, {rgeos}
- se connecte à GDAL, GEOS, PROJ
- étend le `dataframe` en ajoutant une colonne `GEOMETRY`
- compatible Tidyverse
- fonctions spatiales de base
 - `st_read()` / `st_write()`
 - `st_intersection()` / `st_intersects()`
 - `st_buffer()` / `st_union()`
 - etc

{sf}

Structure de données

```
## 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	geom
## 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...

Simple feature

Simple feature geometry list-column (sfc)

Simple feature geometry (sfg)

{sf}

Astuces

Manipulation de la data géo facile mais :



- `st_set_CRS()` ou `st_transform()` au chargement
- `st_drop_geometry()`
- `nested-join` à préférer à `left-join` (`dplyr >= 0.8.0`)

Données raster

- {raster}
- {stars}
- {rayshader}

Que puis-je faire de mes data géo ?

- data classique : filtrage, sélection, mutation, statistiques
- data géo: statistiques spatiales (régression spatiale, Moran, Geary, Kriegage)
 - `{spdep}`
 - `{ade4}`

Que puis-je faire de mes data géo ?

- Cartographie
 - {ggplot2}
 - {cartography}
 - {tmap}
 - {leaflet}
 - {mapview}

{ggplot2} + {ggspatial}

- Part of the {tidyverse}
- Hadley Wickham, Thomas Lin Pedersen, Kara Woo
- Basé sur une accumulation de couche
- compatible avec {sf} et les outils du {tidyverse}
- Orienté data visualisation
- {ggspatial} fournit les décorations (flèche nord, échelle, etc)

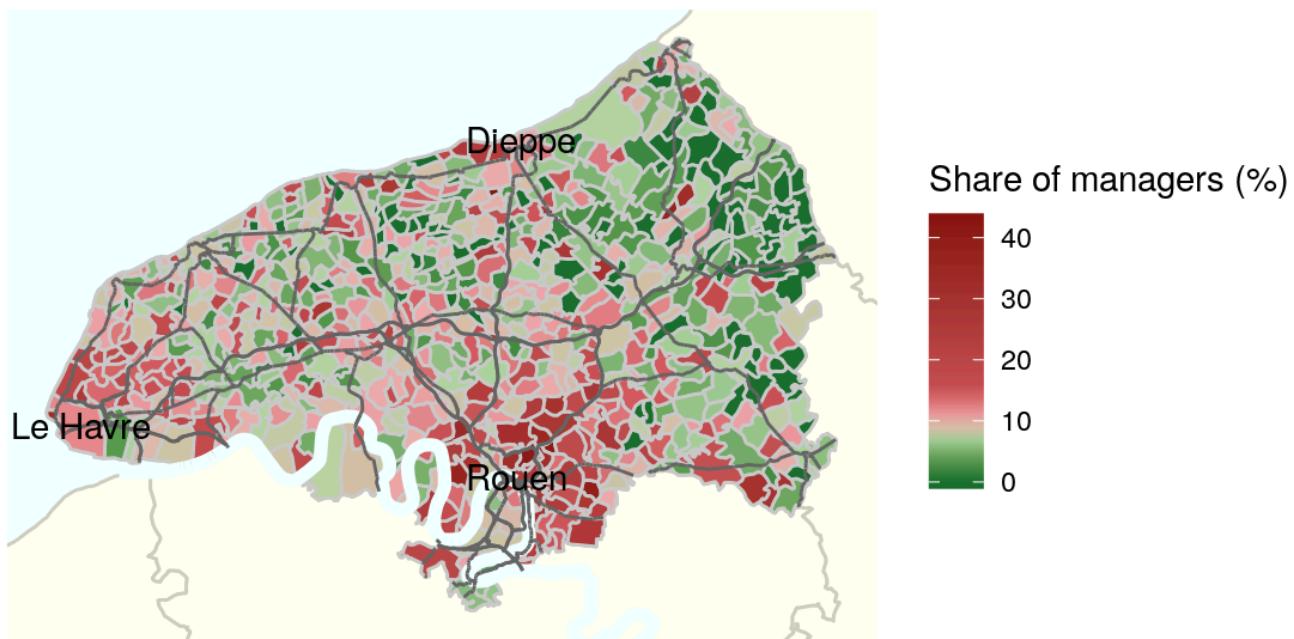
ggplot2

```
library(sf)
library(ggplot2)
ggplot() +
  geom_sf(data = dep, colour = "ivory3", fill = "ivory3")
  geom_sf(data = sm, aes(fill = pcad), colour = "black")
  geom_sf(data = river.geom, colour = "azure", size = 1)
  geom_sf(data = roads.geom, colour = "#666666", size = 0.5)
  geom_text(data=labgg,aes(x=x,y=y,label=LIBELLE))
  scale_fill_gradientn(name = "Share of managers"
                        values=bks/max(bks),
                        colours = carto.pal("green"))
  coord_sf(crs = 2154, datum = NA,
            xlim = st_bbox(sm)[c(1,3)],
            ylim = st_bbox(sm)[c(2,4)]) +
  theme_minimal() +
  theme(panel.background = element_rect(fill = "white"))
  labs(title = "Managers",
       caption = "Insee, 2018\nKim, Tim & Comeet")
```



{ggplot2}

Managers



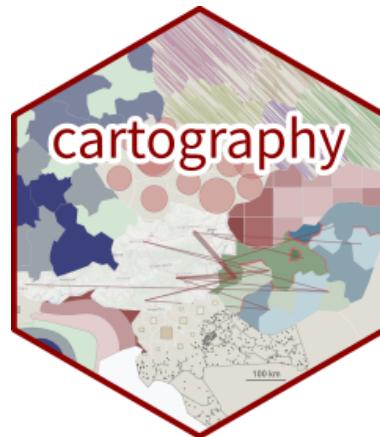
Insee, 2018
Kim, Tim & Comeetie, 2019

{Cartography}

- RIATE (laboratoire du CNRS)
 - Timothée Giraud
 - Nicolas Lambert
- dédié à la cartographie

Documentation :

riatelab.github.io/cartography/



{Cartography}

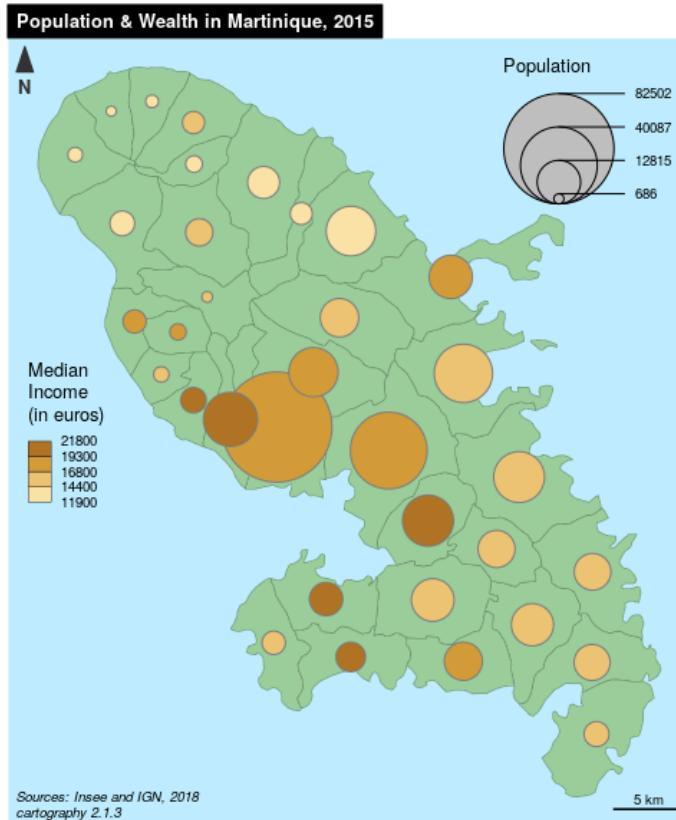
```
library(sf, cartography)
# path to the geopackage file embedded in cartography
path_to_gpkg <- system.file("gpkg/mtq.gpkg", package="cartography")
# import to an sf object
mtq <- st_read(dsn = path_to_gpkg, quiet = TRUE)
# Plot the municipalities
plot(st_geometry(mtq), col="darkseagreen3", border="darkseagreen4",
      bg = "lightblue1", lwd = 0.5)
# Plot symbols with choropleth coloration
propSymbolsChoroLayer(
  x = mtq,
  var = "POP",
  inches = 0.4,
  border = "grey50",
  lwd = 1,
  legend.var.pos = "topright",
  legend.var.title.txt = "Population",
  var2 = "MED",
  method = "equal",
  nclass = 4,
  col = carto.pal(pal1 = "sand.pal", n1 = 4),
  legend.var2.values.rnd = -2,
  legend.var2.pos = "left",
  legend.var2.title.txt = "Median\nIncome\n(in euros)")
)
```



{Cartography}

```
# layout
layoutLayer(title="Population & Wealth in Martinique, 2015",
            author = "cartography 2.1.3",
            sources = "Sources: Insee and IGN, 2018",
            scale = 5, tabtitle = TRUE, frame = FALSE)
# north arrow
north(pos = "topleft")
```

{Cartography}



Source: [cran.r-](#)

project.org/web/packages/cartography/vignettes

{Tmap}

- conçu pour les cartes thématiques
- écrit par Martijn Tennekes
- philosophie de *A Layered Grammar of Graphics* by Hadley Wickham
- sortie plot + JS

<https://cran.r-project.org/web/packages/tmap/vignettes/tmap-getstarted.html>

{Tmap}

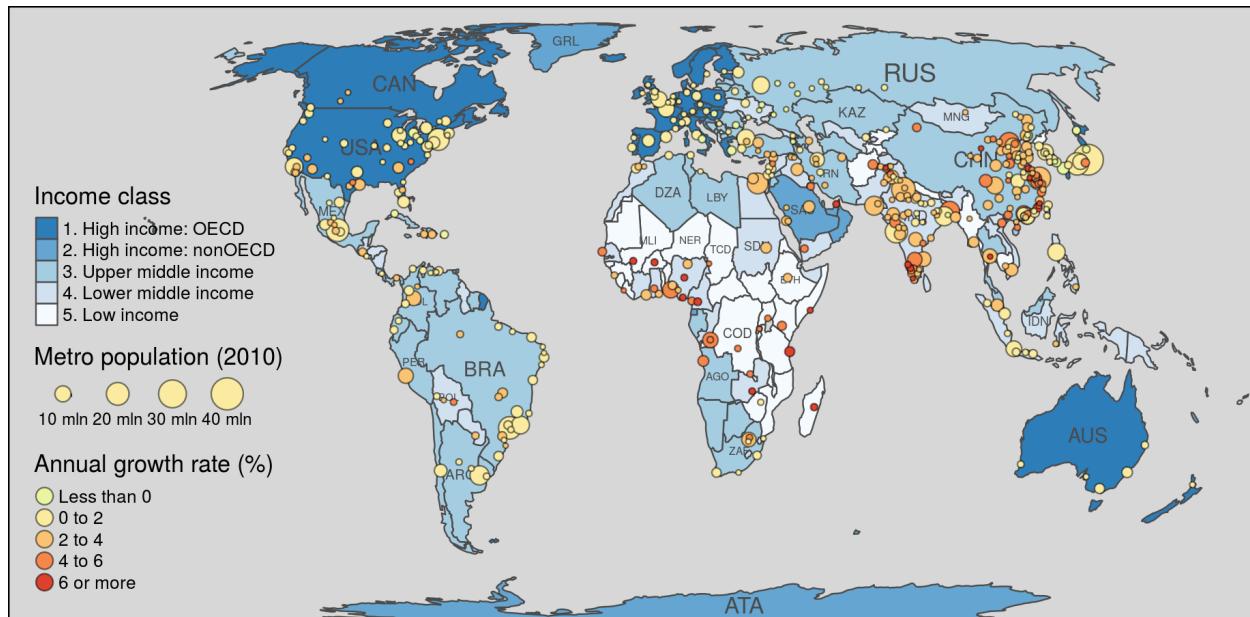
```
library(tmap)
# load spatial data included in the tmap package
data("World", "metro")

# calculate annual growth rate
metro$growth <- (metro$pop2020 - metro$pop2010) / (metro$pop2010 *

# plot
tm_shape(World) +
  tm_polygons("income_grp", palette = "-Blues",
              title = "Income class", contrast = 0.7, border.col = "gray30",
              tm_text("iso_a3", size = "AREA", col = "gray30", root=3) +
  tm_shape(metro) +
    tm_bubbles("pop2010", col = "growth", border.col = "black",
               border.alpha = 0.5,
               breaks = c(-Inf, 0, 2, 4, 6, Inf) ,
               palette = "-RdYlGn",
               title.size = "Metro population (2010)",
               title.col = "Annual growth rate (%)",
               id = "name",
               popup.vars=c("pop2010", "pop2020", "growth")) +
  tm_format_World() +
  tm_style_gray()
```

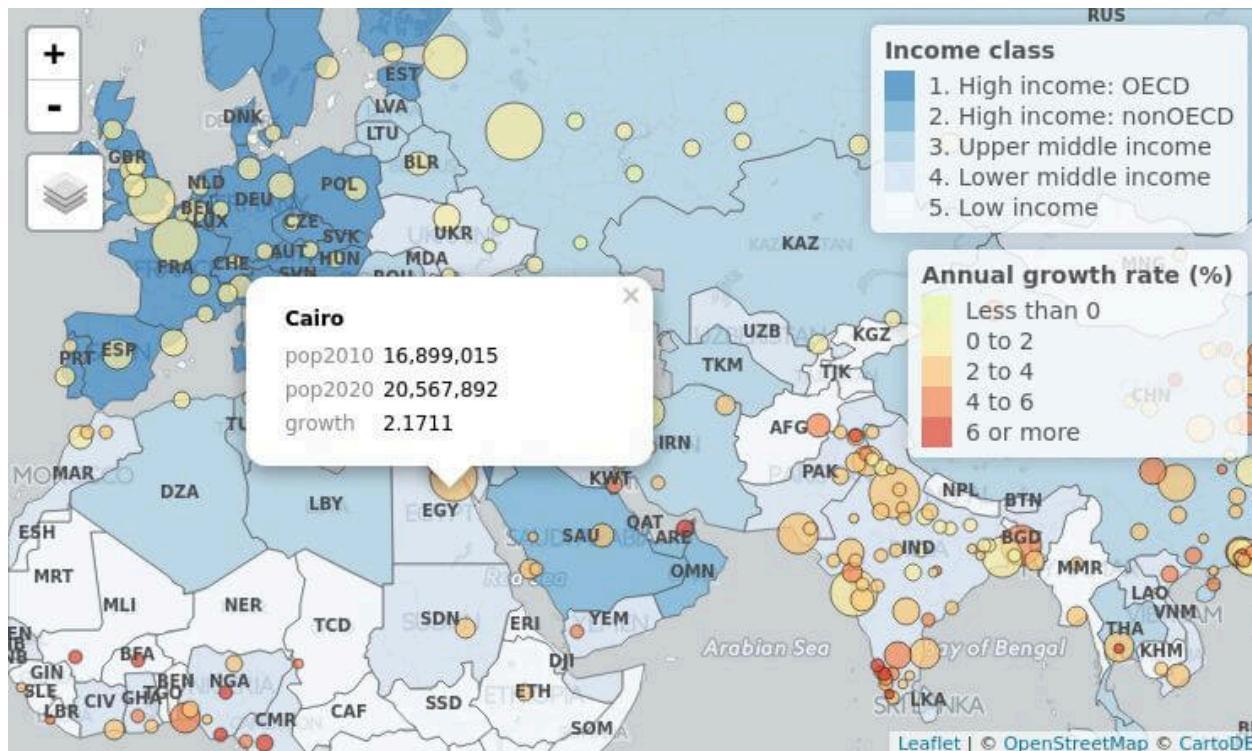


{Tmap}



Tmap

```
# view  
tmap_mode("view")  
last_map()  
tmap_mode("plot")
```



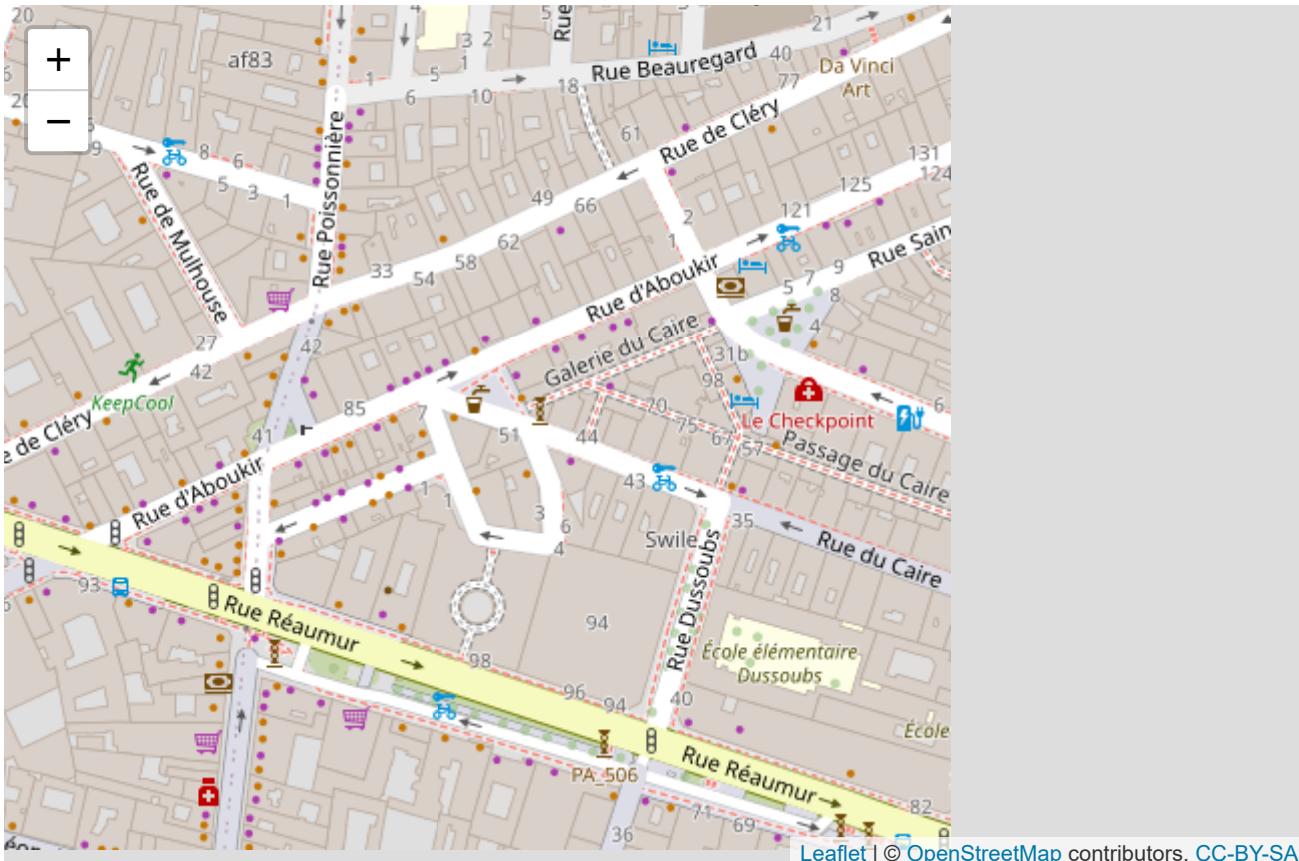
{Leaflet}

- Bibliothèque légère de cartographie
- JS

```
library(leaflet)
long <- 2.346
lat  <- 48.8695
```

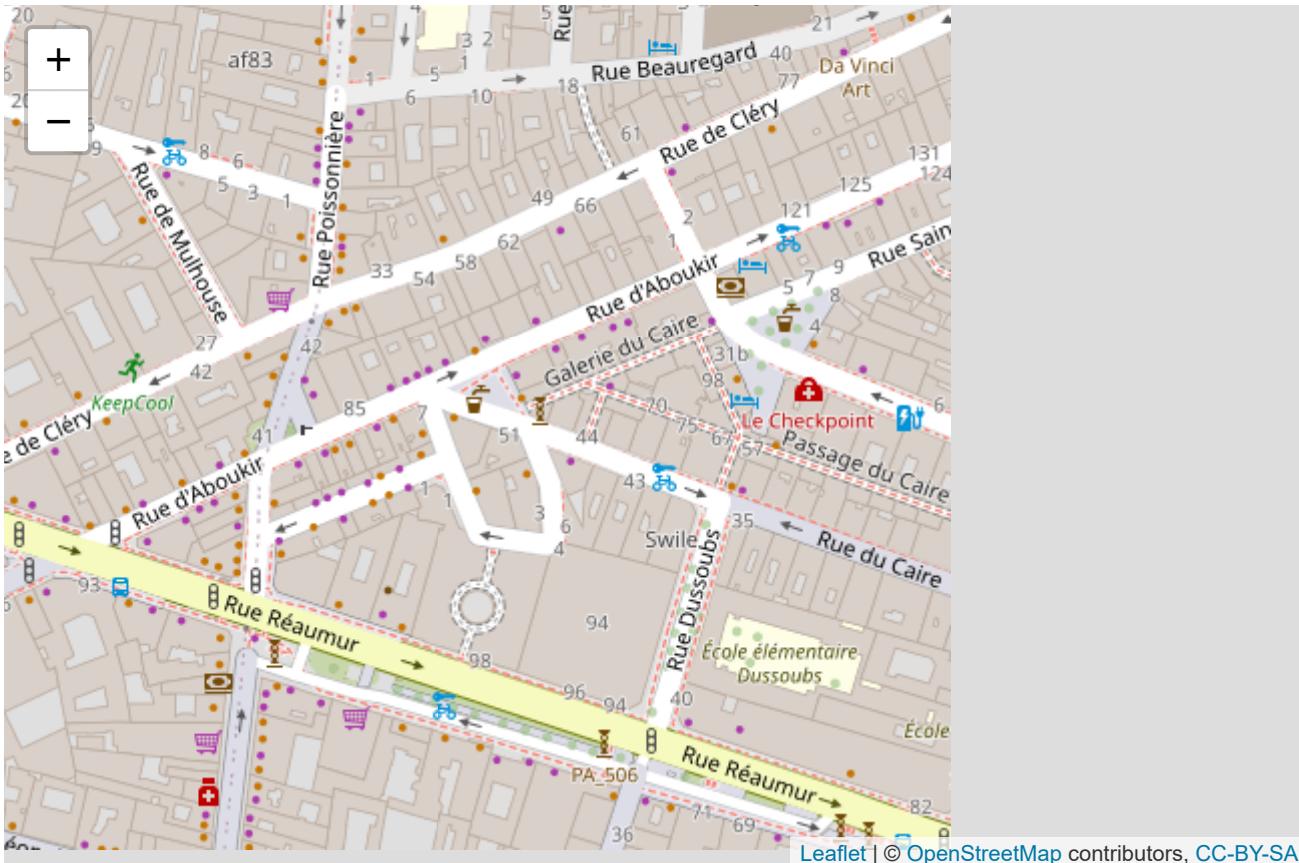
{leaflet}

```
l <- leaflet() %>% addTiles() %>%
  setView(long, lat, zoom = 17)
l
```

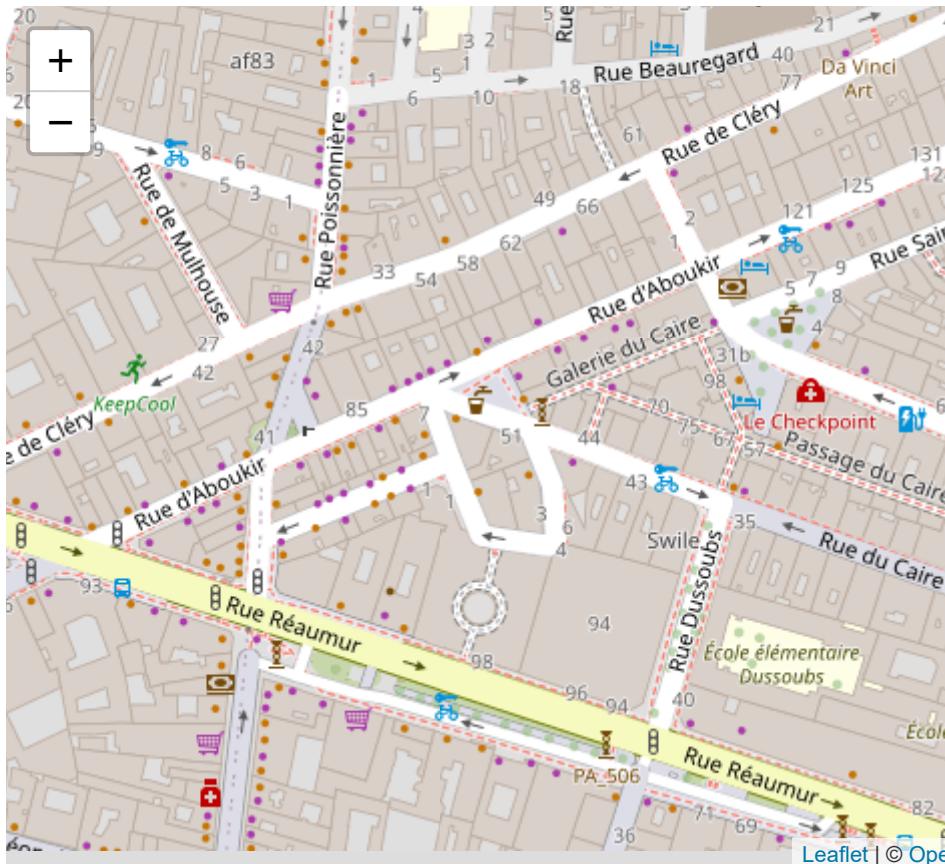


{leaflet}

```
l %>% addMarkers(long, lat,  
                    popup = "AssessFirst",  
                    label = "20 rue du Sentier")
```



```
l %>%
  addRectangles(
    lng1= 2.345, lat1=48.8698,
    lng2= 2.347, lat2=48.8691,
    fillColor = "transparent"
  )
```

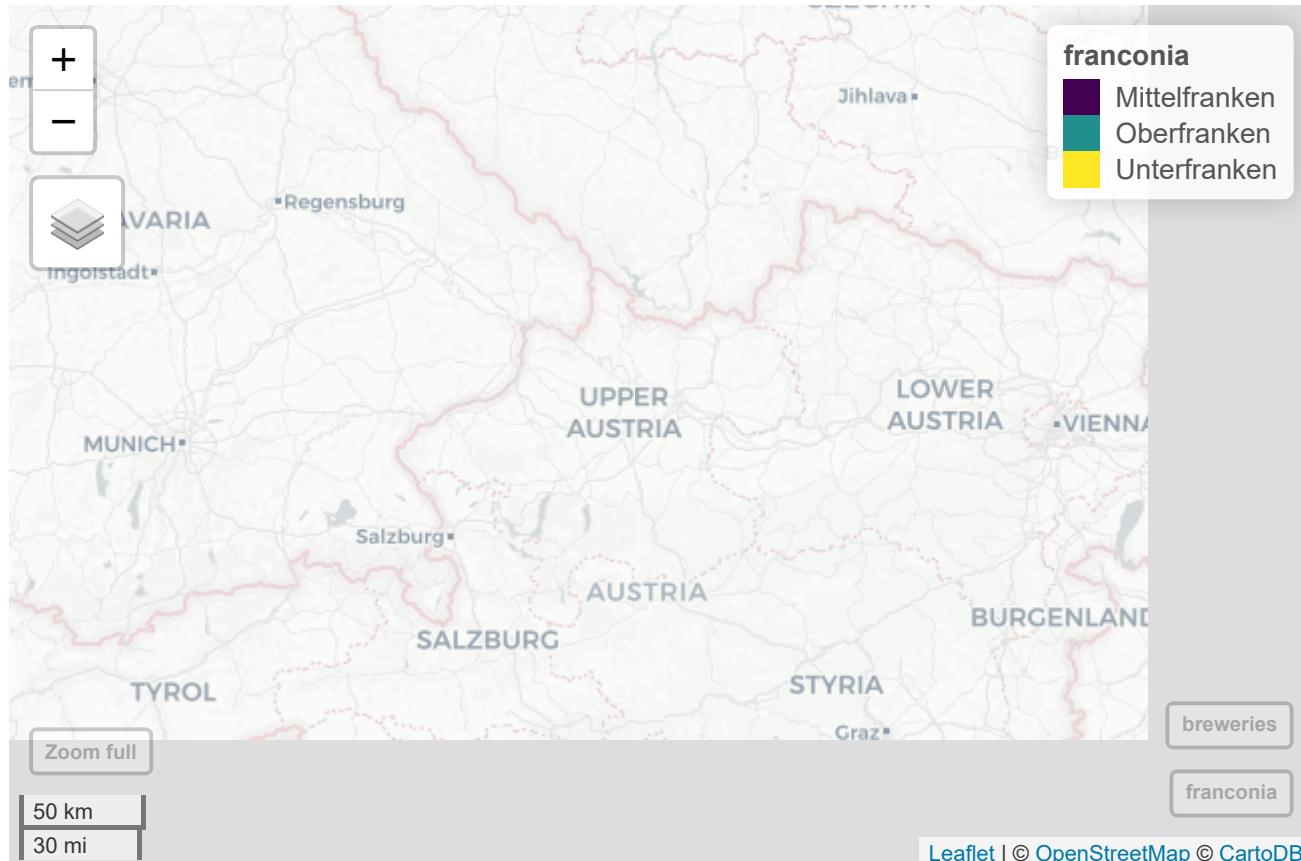


{Mapview}

- créé par Tim Appelhans
- Cartographie dynamique en javascript
- directement intégré à R
- une fonction principale: `mapview()`

{Mapview}

```
mapview(list(franconia, breweries),  
        zcol = list("district", NULL),  
        legend = list(TRUE, FALSE))
```



{Rayshader}

- Tyler Morgan-Wall
- Visualisation de données 3D
- Simulation du niveau des eaux
(sécheresse, inondations,)



Documentation :

<https://www.rayshader.com/>

{Rayshader}

```
library(rayshader)

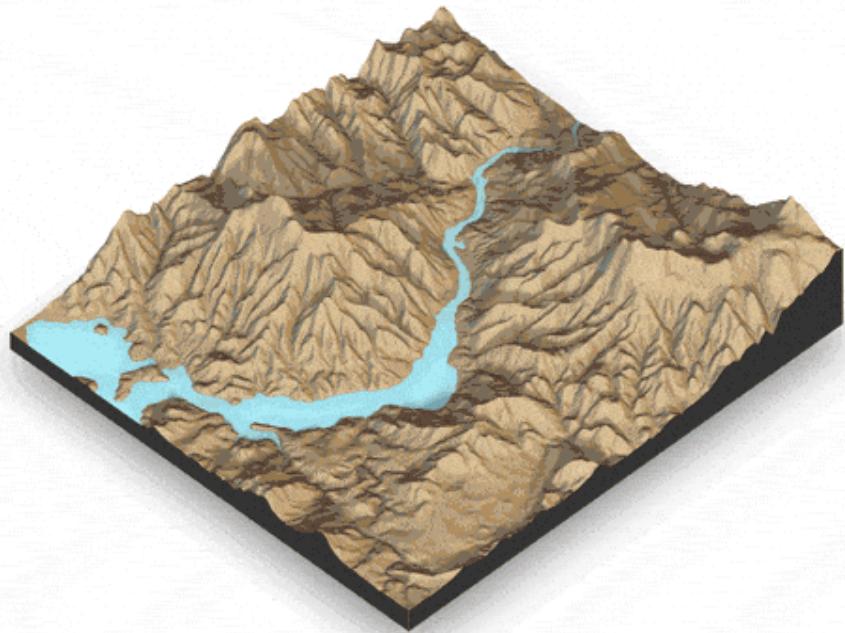
#Here, I load a map with the raster package.
loadzip = tempfile()
download.file("https://tylermw.com/data/dem_01.tif.zip", loadzip)
localtif = raster::raster(unzip(loadzip, "dem_01.tif"))
unlink(loadzip)

#And convert it to a matrix:
elmat = matrix(raster::extract(localtif,raster::extent(localtif),bu
nrow=ncol(localtif),ncol=nrow(localtif))

elmat %>%
  sphere_shade(texture = "desert") %>%
  add_water(detect_water(elmat), color="desert") %>%
  add_shadow(ray_shade(elmat,zscale=3,maxsearch = 300),0.5) %>%
  add_shadow(ambmat,0.5) %>%
  plot_3d(elmat,zscale=10,fov=0,theta=135,zoom=0.75,phi=45, windows
render_snapshot()
```



{Rayshader}



(animation non contractuelle)

Exception culturelle française

Dompter le Code Officiel Géographique

avec **COGugaison** et **CARTElette**

- Crées par Kim AntuneZ (@antuki13)
- **COGugaison**: manipulation du COG pour les données communales et supracommunales
- **CARTElette** : fichiers shapefile correspondants



C'est quoi le COG ?

- Code Officiel Géographique
- édité par l'INSEE
- plusieurs niveaux
 - 13 régions + 5 DROM
 - 100 départements
 - +/- 35000 communes suivant le millésime
 - +/- 50000 IRIS (Îlots regroupés pour l'information statistique)

C'est quoi le COG ?

- permet d'associer une géométrie à des données stats:
 - ADMIN EXPRESS - IGN
 - Régions
 - Départements
 - Arrondissements départementaux
 - EPCI
 - Communes
 - Chef-lieu de commune
 - Contours Iris -IGN

COGugaison et CARTElette dans tout ça ?

- Niveau communal et supra-communal
 - pas les IRIS
 - Communes
 - Arrondissements
 - EPCI
 - Bassin de vie
 - Aires urbaines
 - Unités urbaines
 - Départements
 - Régions
 - Zone d'emploi
 - Cantons

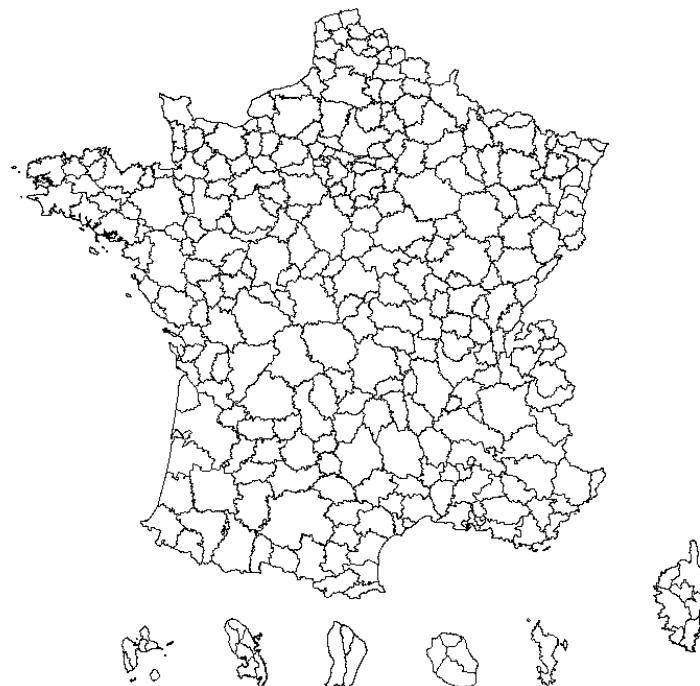
COGugaison et CARTElette dans tout ça ?

- {COGugaison}
 - détecte le millésime d'un jeu de données
 - transformation année X -> année Y (Population, Type, etc)
 - 1968 -> 2019
- {CARTElette}
 - permet de récupérer les géométries
 - 2015 -> 2019
- Présentation RLadies 03/2018:

antuki.github.io/slides/180306_RLadies_COGugaison_carto/180306_RLadies_COGugaison_carto.html

CARTElette

```
library(CARTElette)
ZE_sf <- loadMap(COG=2016,nivsupra="ZE2010")
par(mar=c(0,0,0,0))
plot(sf::st_geometry(ZE_sf))
```



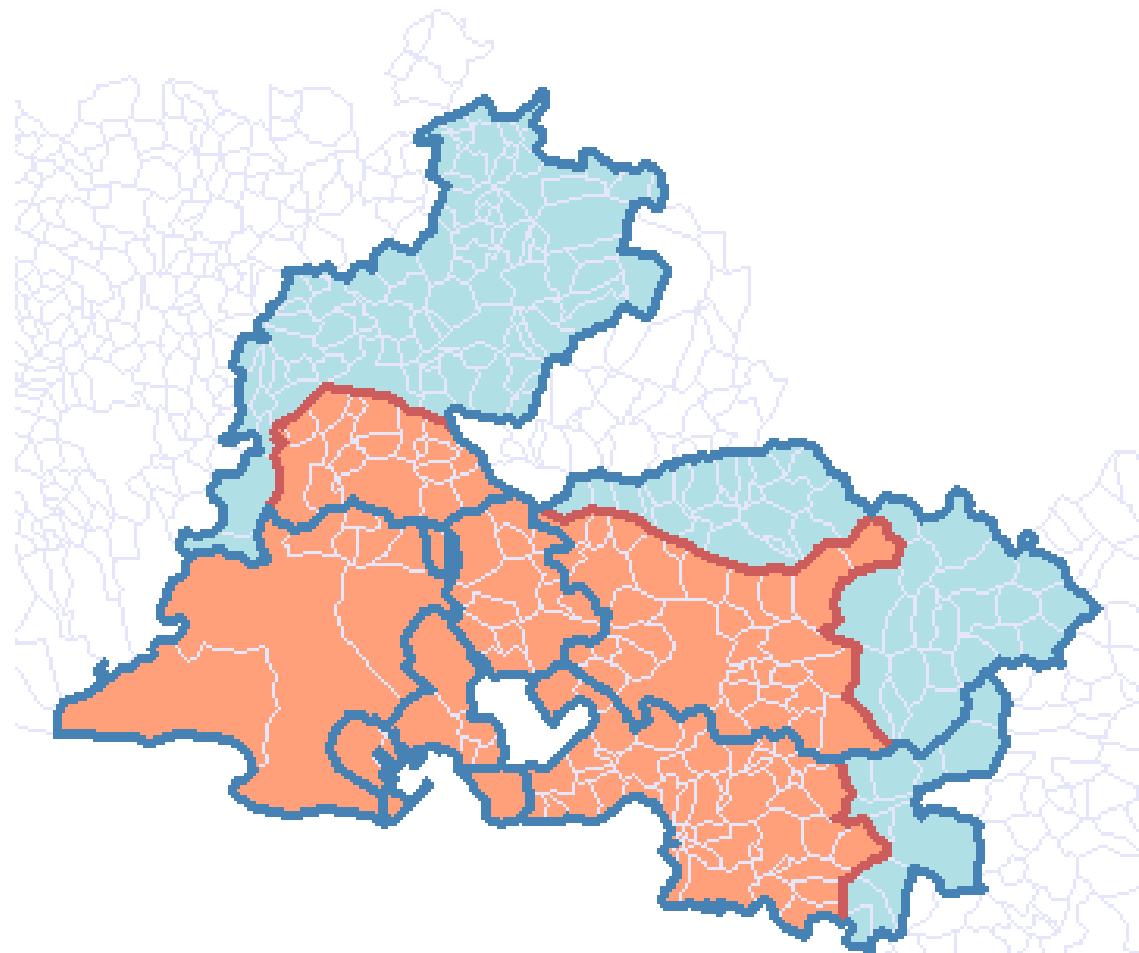
{Oceanis}

Package de cartographie créé par l'INSEE

- Utilise `plot` et `{leaflet}`
- Dispose de fonctions dédiées à `{shiny}`
- CRAN: cran.r-project.org/web/packages/oceanis
- Vignette : <https://cran.r-project.org/web/packages/oceanis/vignettes>

{Oceanis}

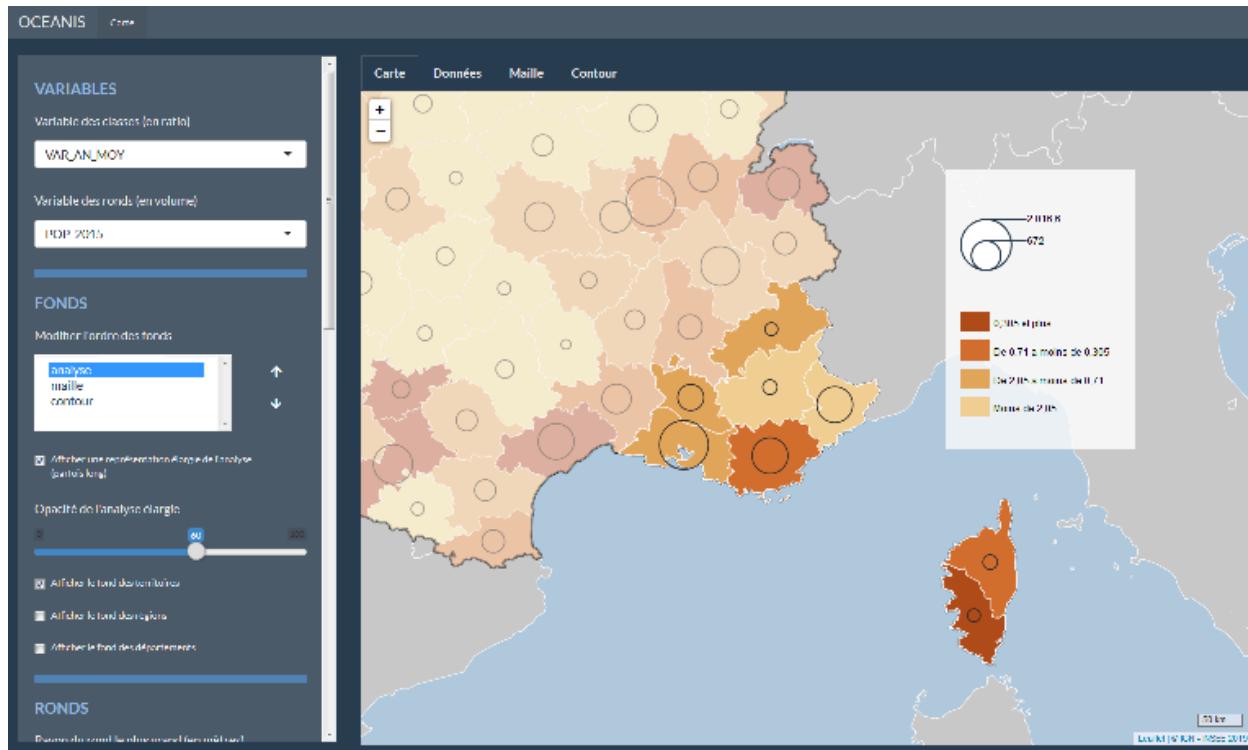
plot()



{Oceanis}

{shiny} + {leaflet}

#



Ressources

- sites et vignettes des paquets
- [Tuto satRday 2019](#)
- [Geocomputation with R - Lovelace, Nowosad, Muenchow](#)
- [Manuel d'analyse spatiale - INSEE - Octobre 2018](#)
- Formation R au MTES/MCTRCT (à venir) : mtes-mct.github.io/parcours-r
- Observatoire des territoires : [Exemple d'utilisation de COGugaison en production](#)

Données géospatiales et cartographie avec R

Nicolas Roelandt

vendredi 28 juin 2019

- Blog : roelandtn.frama.io
- Slides : roelandtn.frama.io/slides/2090628_meetup_Raddict_datageo.html

