Représentation et traitement d
l'information géographique

Timothée Giraud (CNRS - UMS RIATE)

Quatrièmes Rencontres R - Grenoble - 26/06/2015

Représentation et traitement de l'information géographique

1. Les API de cartographie
2. Représentations cartographiques
3. Opérations de géotraitements
4. Lissage spatial
5. Ressources

Les API des cartographie

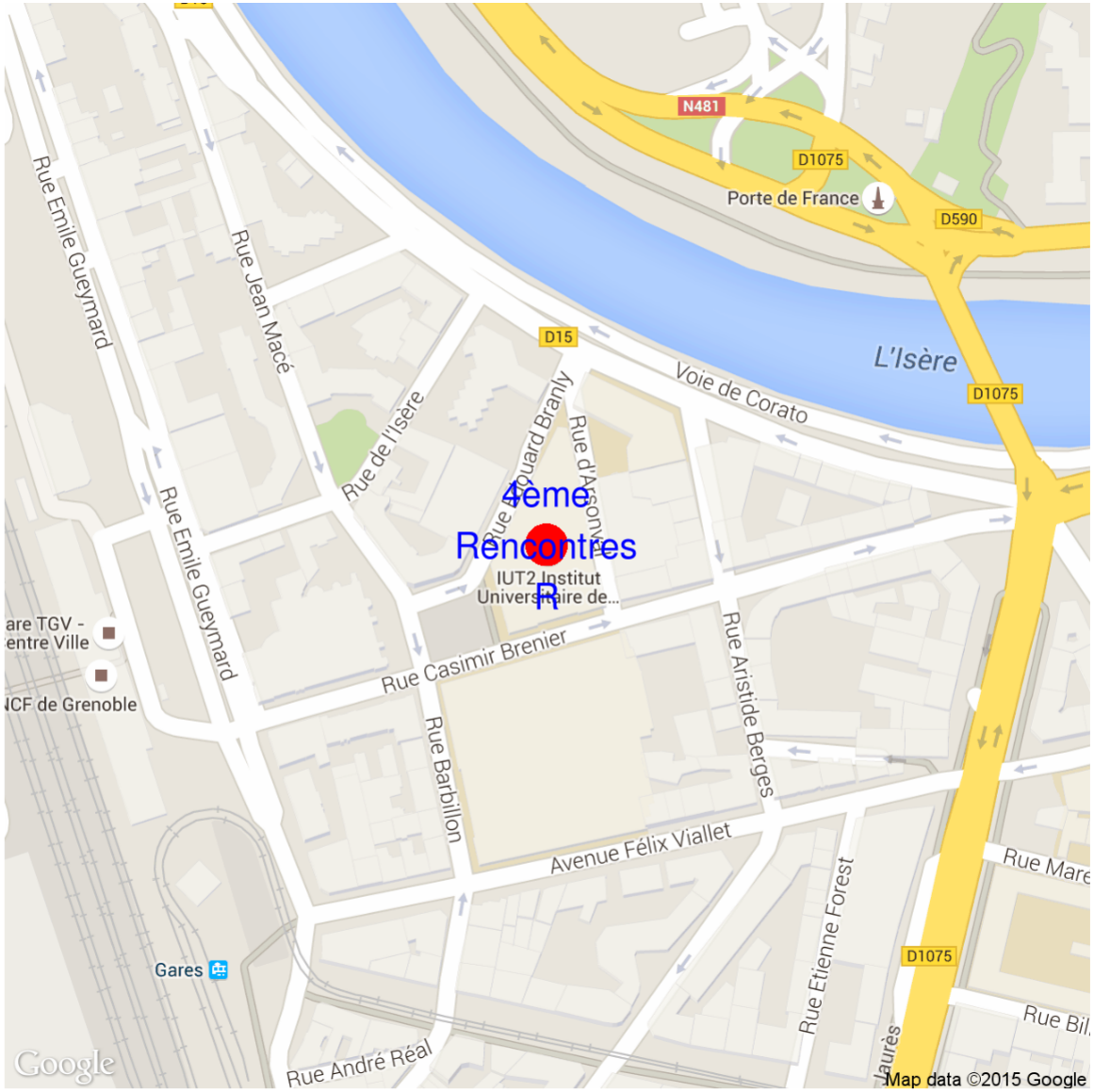
Acquisition de fonds de carte statiques

Le package ggmap

```
library(ggmap)
# téléchargement de la carte
mapIUT<- get_map(location = c(lon = 5.717324, lat = 45.192043),
                 color = "color",
                 source = "google",
                 maptype = "satellite",
                 zoom = 17)
# affichage de la carte
ggmap(mapIUT, extent = "device")
```

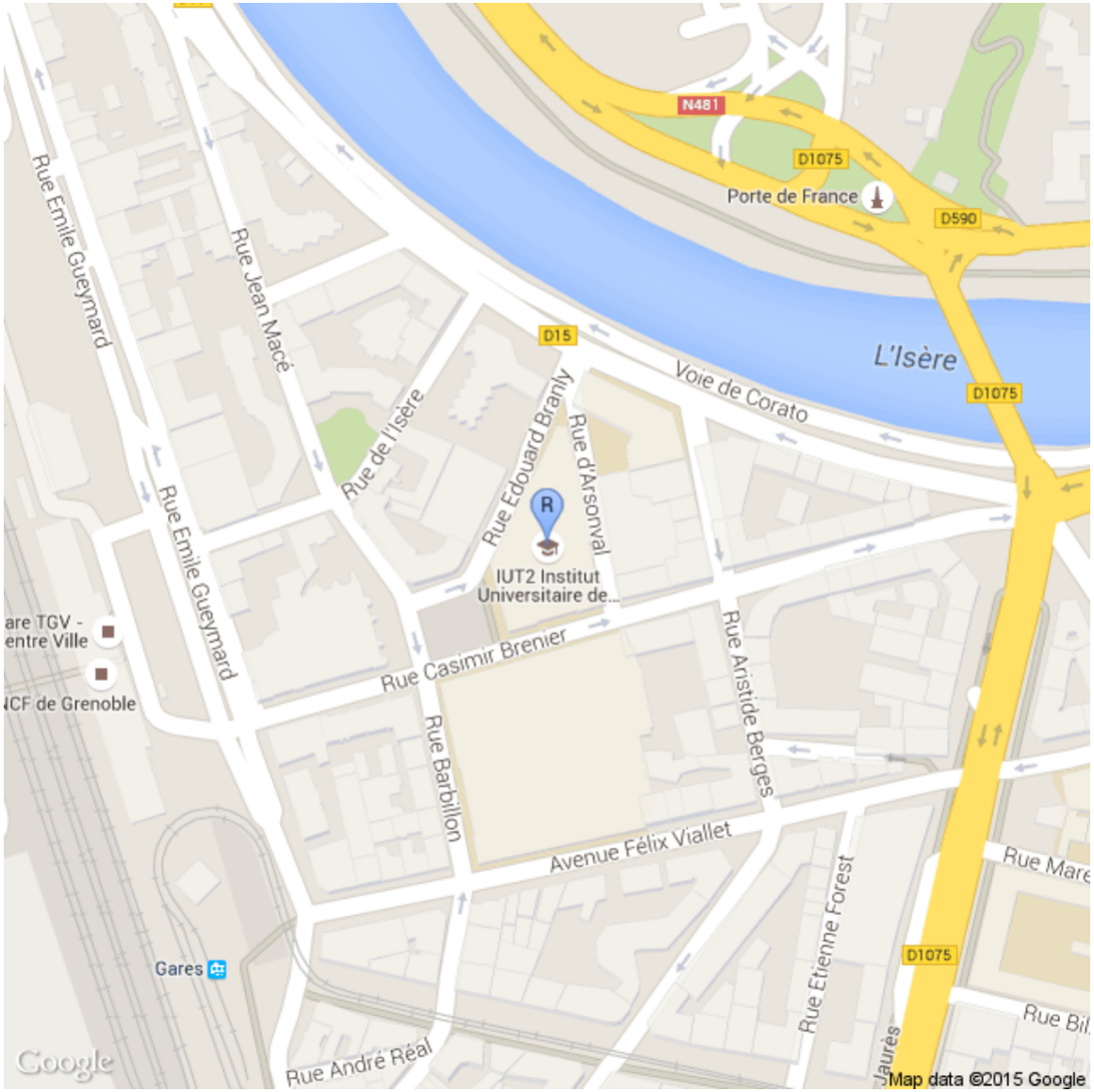

Google
Image © 2015 DigitalGlobe

```r
library(ggmap)
# téléchargement de la carte
mapIUT<- get_map(location = c(lon = 5.717324, lat = 45.192043),
                 color = "color",
                 source = "google",
                 maptype = "roadmap",
                 zoom = 17)
# affichage de la carte et d'un marqueur sur le lieu de la conférence
library(ggplot2)
d <- data.frame(lat = 45.192043, lon = 5.717324, text ="4ème\nRencontres\nR")
p <- ggmap(mapIUT, extent = "device")
p <- p + geom_point(data=d, aes(x=lon, y=lat), color="red", size=8, alpha=1)
p + annotate("text", x = d$lon, y = d$lat, label = d$text, col = "blue")
```


4ème
Rencontres
R
N481
D1075
Porte de France
D590
L'Isère
D15
Voie de Corato
Rue Jean Macé
Rue Emile Gueymard
Rue de l'Isère
Rue Casimir Brenier
Rue Barbillon
Rue Aristide Berges
Avenue Félix Viallet
Rue Etienne Forest
Gares
Rue André Réal
IUT2 Institut Universitaire de...
Google
Map data ©2015 Google

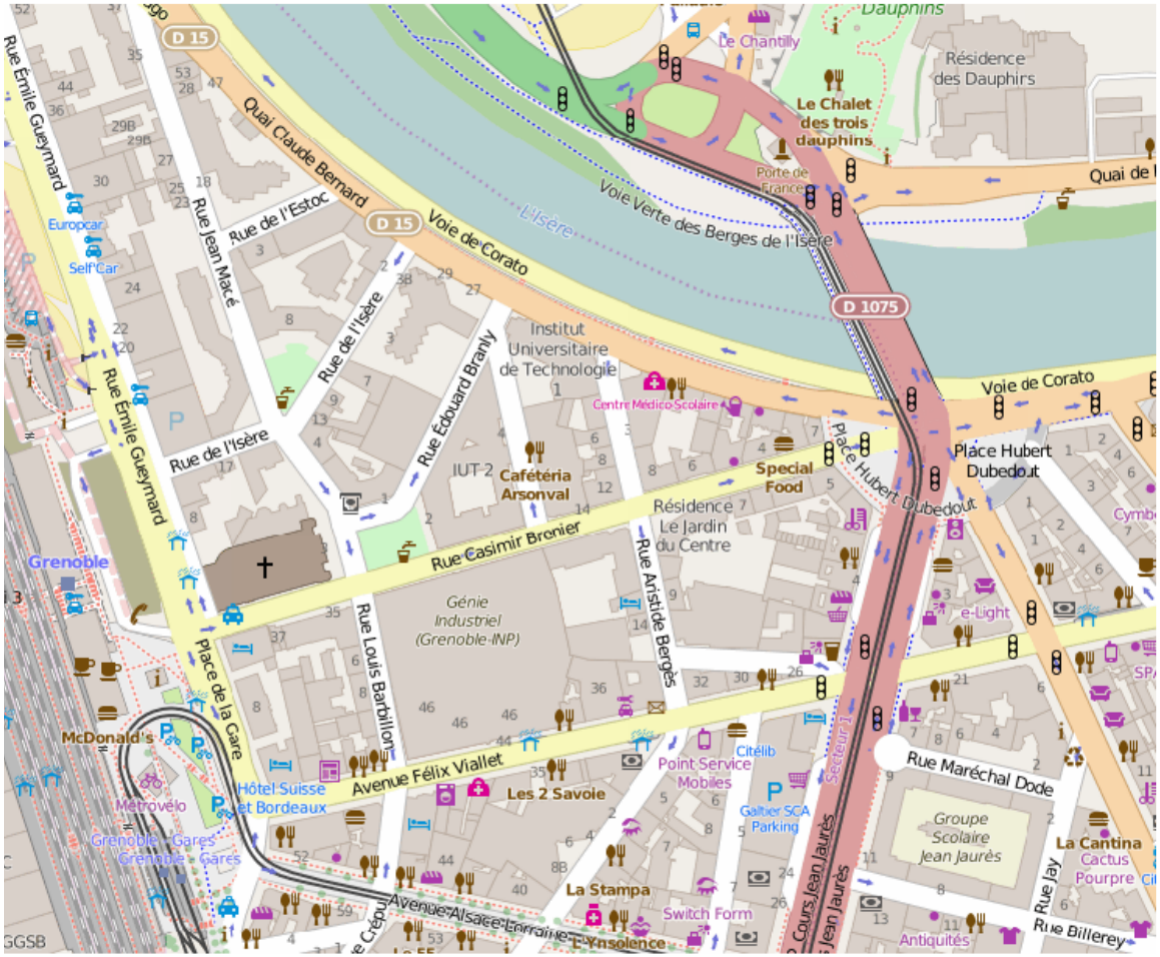
Le package RgoogleMaps

```
library(RgoogleMaps)
# création des marqueurs
markers = paste0("&markers=color:blue|label:R|48.836016,2.372199")
# téléchargement de la carte
mapIUT <- GetMap(center = c( 45.192043,5.717324), zoom = 17, markers = markers,
                 destfile = "img/mapIUT.png")
# affichage de la carte
PlotOnStaticMap(mapIUT)
```

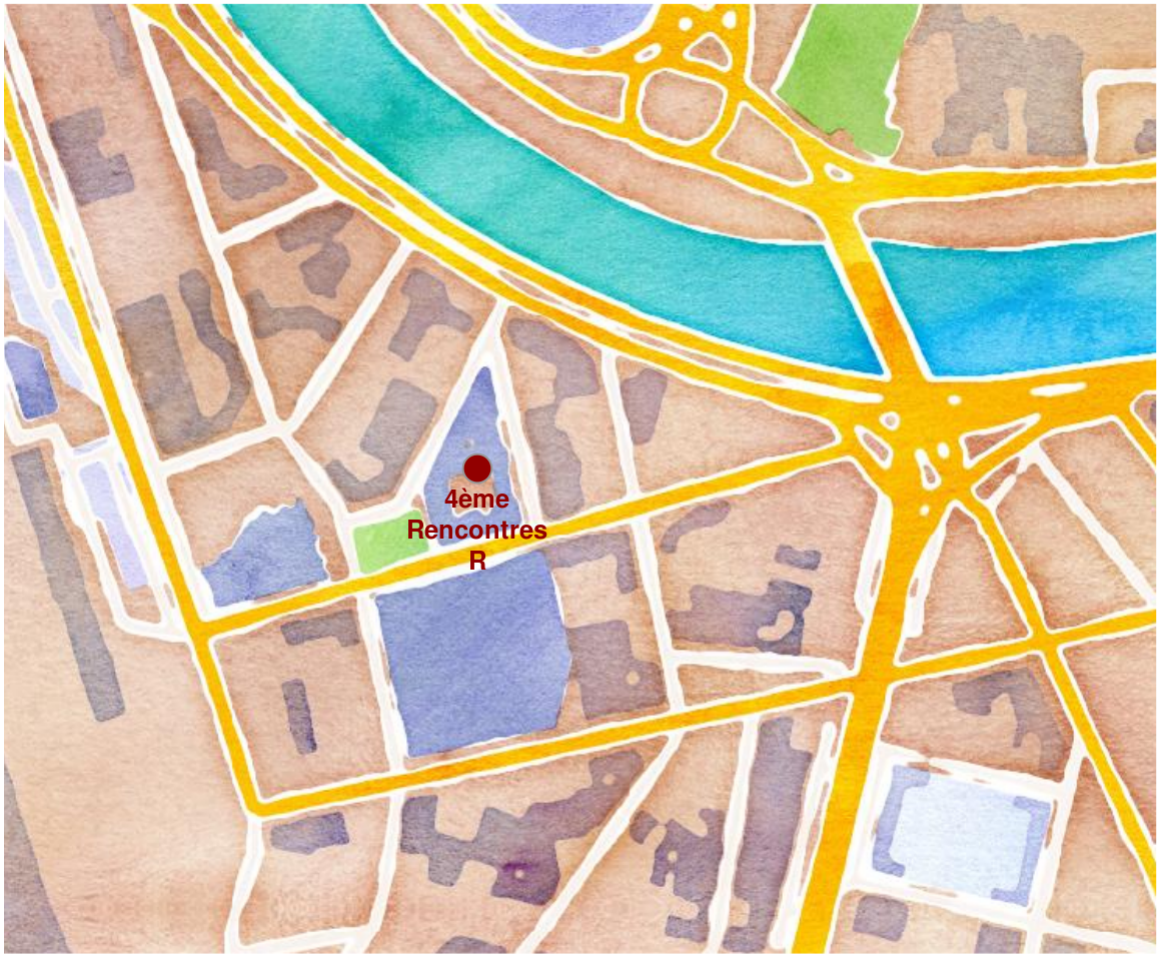

N481
D1075
Porte de France
D590
D15
L'Isère
D1075
Voie de Corato
Rue Emile Gueymard
Rue Jean Macé
Rue de l'Isère
Rue Edouard Branly
Rue d'Arsonval
R
IUT2 Institut Universitaire de...
Rue Casimir Brenier
Rue Aristide Berges
Rue Barbillon
Avenue Félix Viallet
Rue Etienne Forest
Rue Mare
D1075
Rue Bil
Gares
Rue André Réal
Google
Map data ©2015 Google

Le package `OpenStreetMap`

```
library(OpenStreetMap)
# téléchargement de la carte
mapIUT <- openmap(upperLeft = c(45.194303, 5.714051),
                  lowerRight = c(45.189676, 5.722023),
                  type = "osm")
# affichage de la carte
plot(mapIUT)
```


Dauphins
Le Chantilly
Résidence des Dauphirs
Le Chalet des trois dauphins
Porte de France
Quai de l
D 15
Quai Claude Bernard
Voie de Corato
L'Isère
Voie Verte des Berges de l'Isère
D 1075
Rue Émile Gueymard
Rue Jean Macé
Rue de l'Estoc
Rue de l'Isère
Rue Édouard Branly
Institut Universitaire de Technologie
Centre Médico Scolaire
Voie de Corato
Place Hubert Dubedout
Europcar
Self'Car
IUT-2
Cafétéria Arsonval
Special Food
Résidence Le Jardin du Centre
Rue Casimir Brenier
Grenoble
Génie Industriel (Grenoble-INP)
Rue Louis Barbillon
Rue Aristide Bergès
e-Light
Place de la Gare
McDonald's
Métrovélo
Hôtel Suisse et Bordeaux
Avenue Félix Viallet
Les 2 Savoie
Citélib
Point Service Mobiles
Galtier SCA Parking
Secteur 1
Rue Maréchal Dode
Groupe Scolaire Jean Jaurès
La Cantina
Cactus Pourpre
Grenoble Gares
La Stampa
Switch Form
Avenue Alsace-Lorraine
Cours Jean Jaurès
Rue Jay
Rue Billerey
Antiquités
GGSB

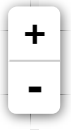
```r
library(OpenStreetMap)
# téléchargement de la carte
mapIUT <- openmap(upperLeft = c(45.194303, 5.714051),
                  lowerRight = c(45.189676, 5.722023),
                  type = "stamen-watercolor")
# affichage de la carte
plot(mapIUT)
# les cartes utilisent la projection mercator, il faut donc reprojeter
# les points à afficher dans cette projection
rconf <- projectMercator(lat = 45.192043, long = 5.717324)
points(x = rconf[1], y = rconf[2], col = 'grey60', bg = "#920000",
       pch = 21, cex = 2)
text(x = rconf[1], y = rconf[2], col = "#920000", font = 2,cex = 0.8,
     labels = "4ème\nRencontres\nR",  adj = c(0.5,1.25))
```


4ème
Rencontres
R

Affichage de cartes interactives

Le package `leaflet`

```
library(leaflet)
# initialiser une carte
m <- leaflet()
# carte avec emprise mondiale
m <- addTiles(map = m)
# centrer sur un point
m <- setView(map = m, lng = 5.717324, lat = 45.192043, zoom = 15)
# ajout d'un pop-up
m <- addPopups(map = m, lng = 5.717324, lat = 45.192043,
               popup = '<a href="http://r2015-grenoble.sciencesconf.org/">
               4ème rencontres R</a>')
# affichage de la carte
m
```



Géocodage

Des points à géocoder

```
# table des points d'intérêt à trouver
poi <- read.csv("data/poi.csv", colClasses = c("character", "character"))
poi
```

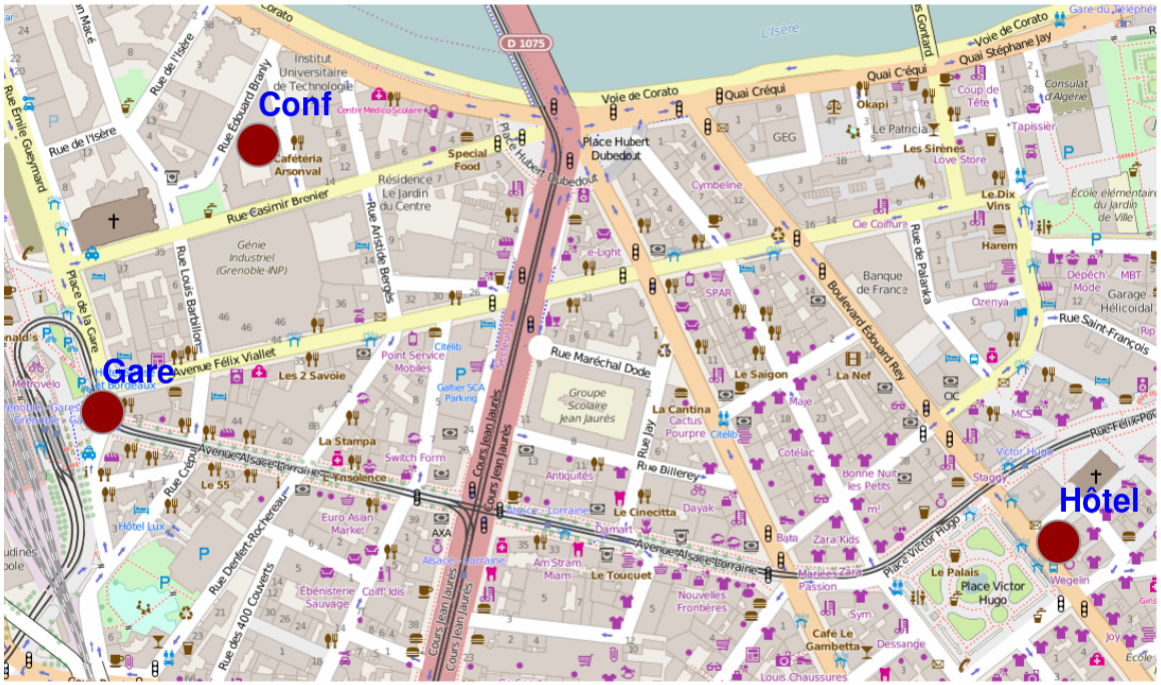
```
##     nom                                adresse
## 1  Gare   1, Place de la gare 38000 Grenoble
## 2 Hôtel 5, place Victor Hugo, 38000 Grenoble
## 3  Conf 2, Place Doyen Gosse, 38000 Grenoble
```

Le package ggmap

```
# recherche des points d'intérêt
library(ggmap)
poigeo <- geocode(poi$adresse, output = "latlona")
poigeo <- cbind(poi, poigeo)
# affichage de la table avec les coordonnées
poigeo
```

```
##     nom                                adresse      lon      lat
## 1  Gare   1, Place de la gare 38000 Grenoble 5.715747 45.19020
## 2 Hôtel 5, place Victor Hugo, 38000 Grenoble 5.725448 45.18927
## 3  Conf 2, Place Doyen Gosse, 38000 Grenoble 5.717328 45.19211
##                                                 address
## 1         1 place de la gare, 38000 grenoble, france
## 2        5 place victor hugo, 38000 grenoble, france
## 3 iut 2, 2 place doyen gosse, 38000 grenoble, france
```

```r
# affichage des points d'intérêt sur la carte
mapGre <- openmap(upperLeft = c(max(poigeo$lat)+0.001,min(poigeo$lon)-0.001),
                  lowerRight = c(min(poigeo$lat)-0.001, max(poigeo$lon)+0.001),
                  type = "osm")
# affichage de la carte
plot(mapGre)
poiproj <- projectMercator(lat = poigeo[,"lat"], long = poigeo[,"lon"])
points(x = poiproj[,1], y = poiproj[,2], col = 'grey60', bg = "#920000",
       pch = 21, cex = 3)
text(x = poiproj[,1], y = poiproj[,2], col = "blue",
     font = 2, labels = poigeo[,1],
     adj = c(0,-1.25))
```


Conf
Gare
Hôtel
D 1075
Voie de Corato
Quai Créqui
Quai Stéphane Jay
Rue de l'Hôtel
Rue Edouard Branly
Rue Casimir Brenier
Rue Aristide Berglès
Rue Louis Barthélemy
Place de la Gare
Avenue Félix Viallet
Rue Maréchal Dode
Boulevard Edouard Rey
Rue de Palanka
Rue Saint-François
Rue Jay
Rue Billerey
Cours Jean Jaurès
Avenue Alsace-Lorraine
Place Victor Hugo
Rue Denfert-Rochereau
Rue des 400 Couverts
Institut Universitaire de Technologie
Cafétéria Arsonval
Résidence Le Jardin du Centre
Génie Industriel (Grenoble-INP)
Place Hubert Dubedout
Groupe Scolaire Jean Jaurès
Les 2 Savoie
Le 55
La Stampa
Le Saigon
La Nef
La Cantina
Le Cinecitta
Le Touquet
Le Palais
Banque de France
Okapi
Le Patricia
Les Sirènes
Le Dix Vins
Harem
Ozenya
Café Le Gambetta
Euro Asian Market
Hôtel Lux
Special Food
Point Service Mobiles
Galtier SCA Parking
Switch Form
Ebénisterie Sauvage
Coiff' Idis
Nouvelles Frontières
Zara Kids
Bata
Wegelin
Staggy
Bonne Nuit les Petits
Cie Coiffure
Coup de Tête
Consulat d'Algérie
École élémentaire du Jardin de Ville
Garage Hélicoïdal
Dépêch Mode
MBT
MCS
Cotélac
Cymbeline
e-Light
SPAR
GEG
Tapissier
Gare du Téléphérique
L'Isère

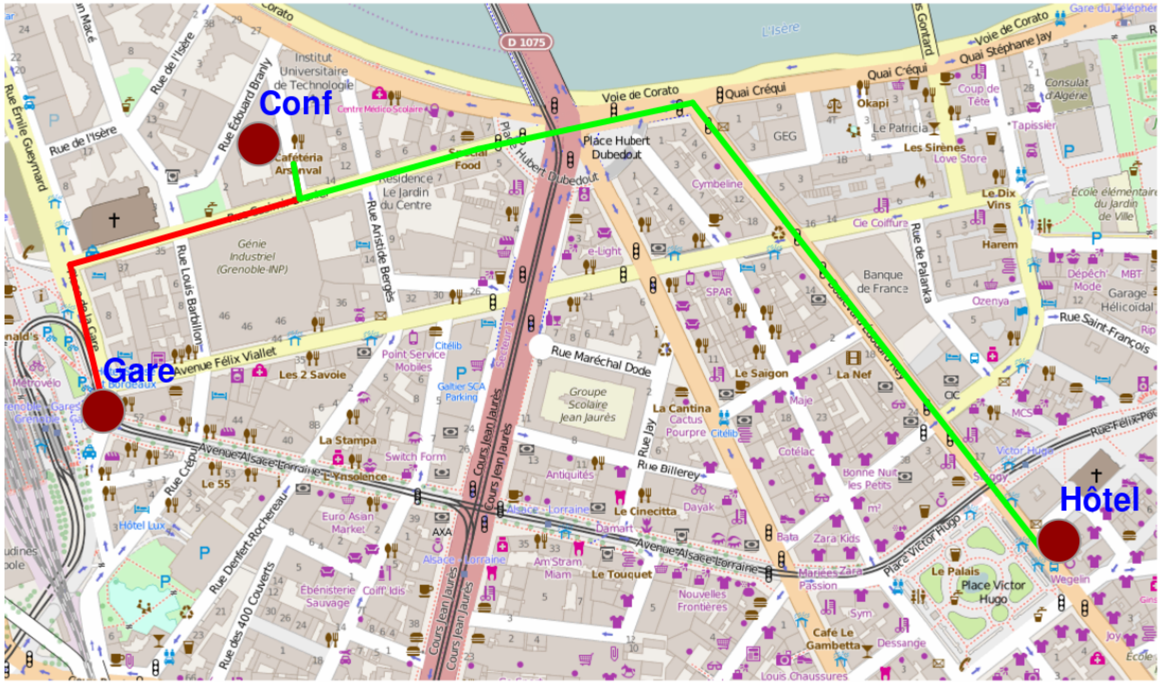
Routage

Le package ggmap

```
# Trajet entre la gare et la conf
gare_conf <- route(from = as.numeric(poigeo[1,3:4]),
                        to =  as.numeric(poigeo[3,3:4]),
                        structure = 'leg',
                        mode = "walking")
# affichage de la table du trajet
gare_conf
```

```
##     m    km     miles seconds   minutes       hours startLon startLat
## 1 123 0.123 0.0764322      92 1.5333333 0.025555556 5.715766 45.19020
## 2 190 0.190 0.1180660     143 2.3833333 0.039722222 5.715413 45.19124
## 3  28 0.028 0.0173992      23 0.3833333 0.006388889 5.717745 45.19171
##     endLon   endLat leg
## 1 5.715413 45.19124   1
## 2 5.717745 45.19171   2
## 3 5.717681 45.19197   3
```

```r
# conversion des coordonnées des points du trajet en mercator
gare_conf[, 7:8] <- projectMercator(lat = gare_conf[,8], long = gare_conf[, 7])
gare_conf[, 9:10] <- projectMercator(lat = gare_conf[,10], long = gare_conf[, 9])
# affichage de la carte
plot(mapGre)
# ajout du trajets
segments(x0 = gare_conf$startLon, y0 = gare_conf$startLat,
         x1 = gare_conf$endLon, y1 = gare_conf$endLat,
         col= 'red', lwd = 3)
# affichage des points
text(x = poiproj[,1], y = poiproj[,2], col = "blue",
     font = 2, labels = poigeo[,1],
     adj = c(0,-1.25))
points(x = poiproj[,1], y = poiproj[,2], col = 'grey60', bg = "#920000",
       pch = 21, cex = 3)
```



Conf
Gare
Hôtel
L'Isère
D 1075
Quai Créqui
Voie de Corato
Quai Stéphane Jay
Place Hubert Dubedout
Avenue Félix Viallet
Avenue Alsace-Lorraine
Cours Jean Jaurès
Rue Maréchal Dode
Rue Billerey
Place Victor Hugo
Rue de l'Isère
Rue Emile Gueymard
Rue Louis Barbillon
Rue Aristide Bergès
Rue Edouard Branly
Rue Denfert-Rochereau
Rue des 400 Couverts
Rue Crépu
Rue Jay
Rue de Palanka
Rue Saint-François
Génie Industriel (Grenoble-INP)
Institut Universitaire de Technologie
Groupe Scolaire Jean Jaurès
Banque de France

Le package ggmap

```
# calcul de distances et temps entre la conf et la gare et l'hotel
x <- mapdist(from = poigeo$adress[3], poigeo$adress[c(1,2)], mode = 'walking',
             output = "simple")
# affichage de la table des distances
x[,c(1,2,4,7)]
```

```
##                                      from
## 1 2, Place Doyen Gosse, 38000 Grenoble
## 2 2, Place Doyen Gosse, 38000 Grenoble
##                                        to    km   minutes
## 1   1, Place de la gare 38000 Grenoble 0.291  3.283333
## 2 5, place Victor Hugo, 38000 Grenoble 0.879 11.216667
```

Attention

The Geocoding API may only be used in conjunction with a Google map; geocoding results without displayi them on a map is prohibited.
https://developers.google.com/maps/documentation/geocoding/

The Directions API may only be used in conjunction with displaying results on a Google map; using Directio data without displaying a map for which directions data was requested is prohibited.
https://developers.google.com/maps/documentation/directions/

Use of the Distance Matrix API must relate to the display of information on a Google Map; for example, to determine origin-destination pairs that fall within a specific driving time from one another, before requesti and displaying those destinations on a map. Use of the service in an application that doesn't display a Goo map is prohibited.
https://developers.google.com/maps/documentation/distancematrix/

Alternatives libres

Pour le géocodage :

- Nominatim
- photon
- MapQuest Open Geocoding API, basé sur Nominatim
- GeoNames, et package `geonames`

Pour le routage :

- GraphHopper
- Open Source Routing Machine (OSRM)

Possibilité de traitements massifs, sous réserve d'installation locale.

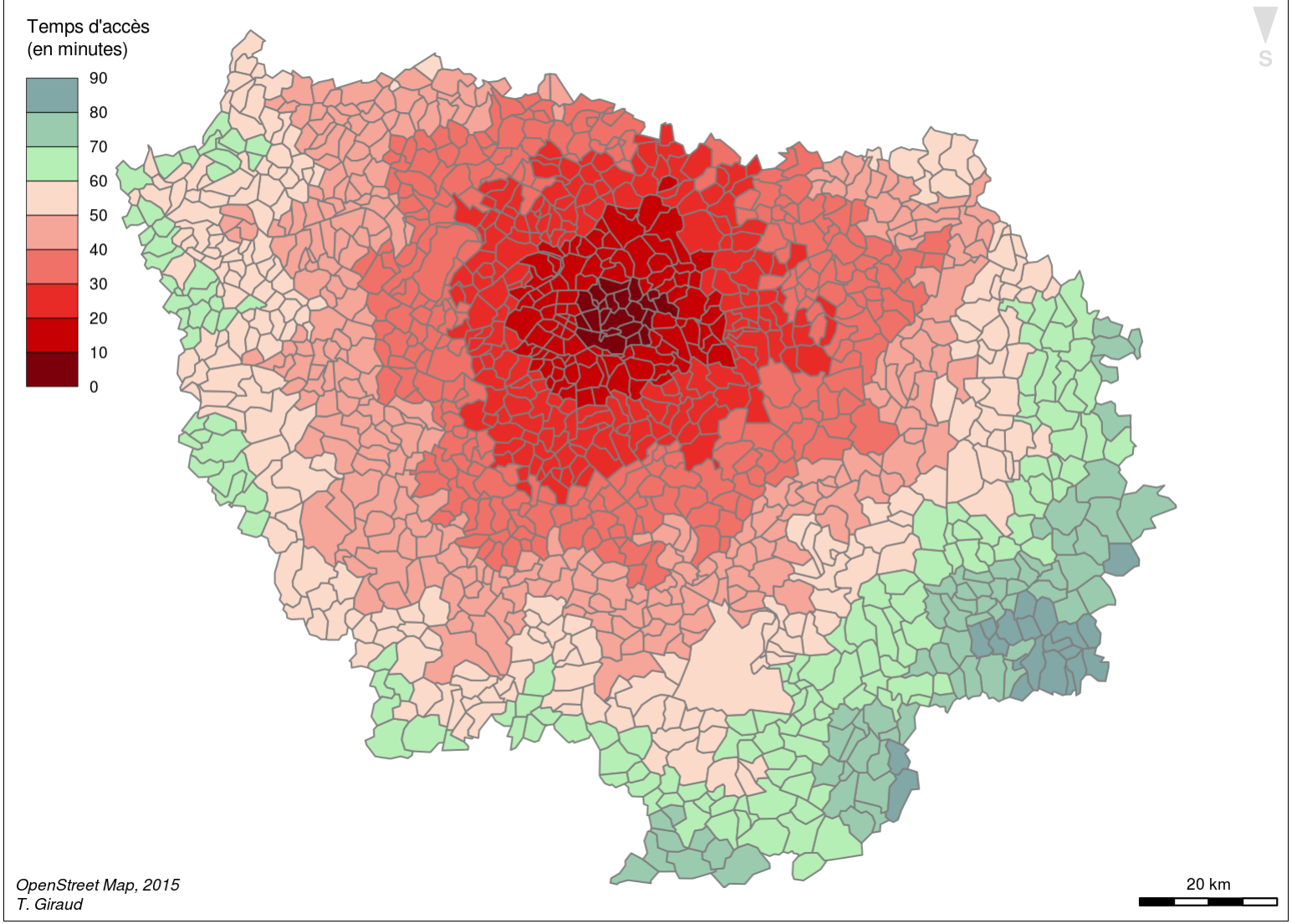
Avec OSRM

```r
# Import de la matrice de distances créée via un appel à l'API d'OSRM
roadDist <- read.csv("data/matrix/matmin.csv",
                         check.names = FALSE,
                         row.names = 1)
# Matrice de distances en minutes
roadDist[1:10, 1:10]
```

```
##       75101 75102 75103 75104 75105 75106 75107 75108 75109 75110
## 75101   0.0   2.0   4.2   4.3   4.7   4.9   4.1   3.3   2.4   4.2
## 75102   1.8   0.0   2.2   4.1   4.7   4.9   4.8   3.4   1.7   2.8
## 75103   4.5   4.2   0.0   2.5   4.5   5.5   6.9   5.6   4.0   2.5
## 75104   2.7   4.0   2.1   0.0   2.6   3.5   4.9   5.8   4.4   3.5
## 75105   4.5   5.8   4.7   3.9   0.0   2.4   4.7   7.1   6.2   5.3
## 75106   4.1   5.4   4.7   4.0   2.0   0.0   2.6   6.0   5.8   5.3
## 75107   4.5   5.0   6.6   5.8   4.6   3.3   0.0   4.8   5.0   6.9
## 75108   3.6   3.4   5.6   6.0   6.3   5.1   4.2   0.0   2.5   4.5
## 75109   2.7   1.7   3.9   5.8   6.4   6.0   5.1   2.7   0.0   2.1
## 75110   4.8   3.8   2.7   4.5   5.4   6.3   7.5   5.1   2.4   0.0
```

```r
# Selection des distances à partir de montreuil
roadDistMontreuil <-  data.frame(id = colnames(roadDist),
                                 t(roadDist[row.names(roadDist) == "93048", ]))
names(roadDistMontreuil)[2] <- "dist"
# Import du fond de carte communal d'IDF
library(rgdal)
comidf.spdf <- readOGR(dsn = "data/map/", layer = "map",verbose = F)
# Cartographie
library(cartography)
cols <- carto.pal(pal1 = "red.pal", n1 = 6, pal2 = "turquoise.pal", n2 = 3)
par(mar=c(0,0,1.2,0))
choroLayer(spdf = comidf.spdf, df = roadDistMontreuil, col = cols,
           legend.title.txt = "Temps d'accès\n(en minutes)", legend.pos = "topleft",
           legend.title.cex = 0.7, border = "grey50",
           var = "dist",
           distr = seq(0,90,10), add=F)
layoutLayer(title = "Accessibilité aux communes d'IDF depuis Montreuil en voiture",
            sources = "OpenStreet Map, 2015", author = "T. Giraud",
            frame = T, south = T)
```

Accessibilité aux communes d'IDF depuis Montreuil en voiture



OpenStreet Map, 2015
T. Giraud

Cartographie

Les indispensables

Le package `sp` : manipulation et affichage des objets spatiaux.

Le package `rgdal` : import/export des objets spatiaux et gestion des projectio cartographiques.

Jeu de données : répartition des groupes sociaux en Île-de-France

```
comidf <- read.csv("data/dataset/data.csv")
head(comidf)
```

```
##      ID                    NAMES population agriculteurs artisans cadres
## 1 75101 Paris 1er Arrondissement      17443           19      763   5230
## 2 75102 Paris 2e  Arrondissement      22927            7      832   7138
## 3 75103 Paris 3e  Arrondissement      36120           24     1326  11154
## 4 75104 Paris 4e  Arrondissement      27887           21     1036   7770
## 5 75105 Paris 5e  Arrondissement      60800           28     1388  16527
## 6 75106 Paris 6e  Arrondissement      43880           61     1551  10687
##   intermediaires employe ouvrier total
## 1           2134    1778     438 10362
## 2           3052    2248    1169 14446
## 3           5003    3418    1160 22085
## 4           3578    2696     638 15739
## 5           6857    4671    1282 30753
## 6           4065    3414     897 20675
```

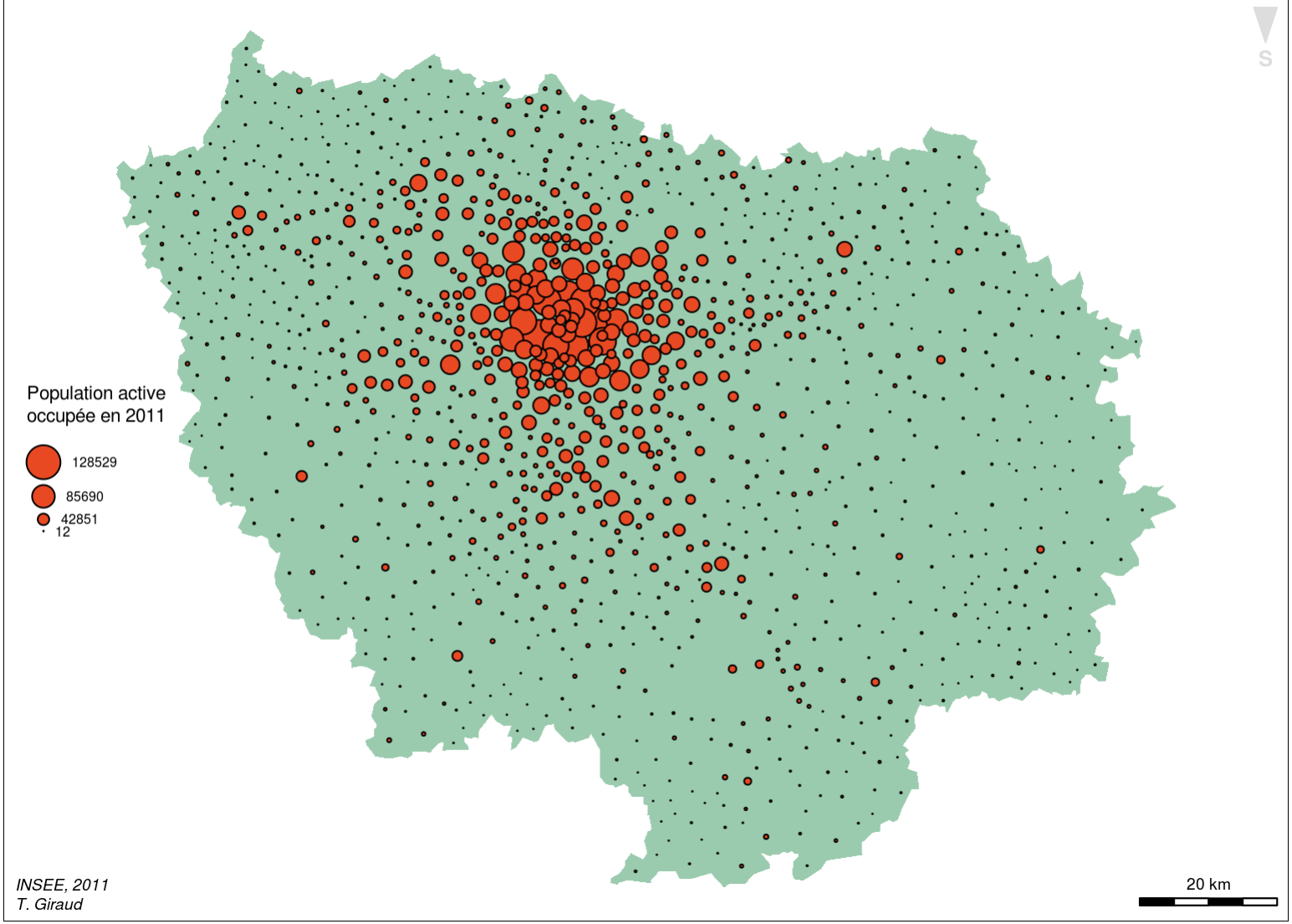
Cartographie en symboles proportionnels

Le package `cartography`

```
# Affichage du fond de carte
par(mar=c(0,0,1.2,0))
plot(comidf.spdf, border = NA, col = "#9BCBAE")

library(cartography)
# Affichage des effectifs de la population active occupée
propSymbolsLayer(spdf = comidf.spdf, df = comidf, var = "total",
                 k = 0.001 ,legend.style = "b",
                 legend.pos = "left", legend.values.cex = 0.5,
                 legend.title.cex = 0.7,
                 legend.title.txt = "Population active\noccupée en 2011")
layoutLayer(title = "Répartition de la population active occupée en IDF",
            sources = "INSEE, 2011", author = "T. Giraud", south = T)
```

Répartition de la population active occupée en IDF

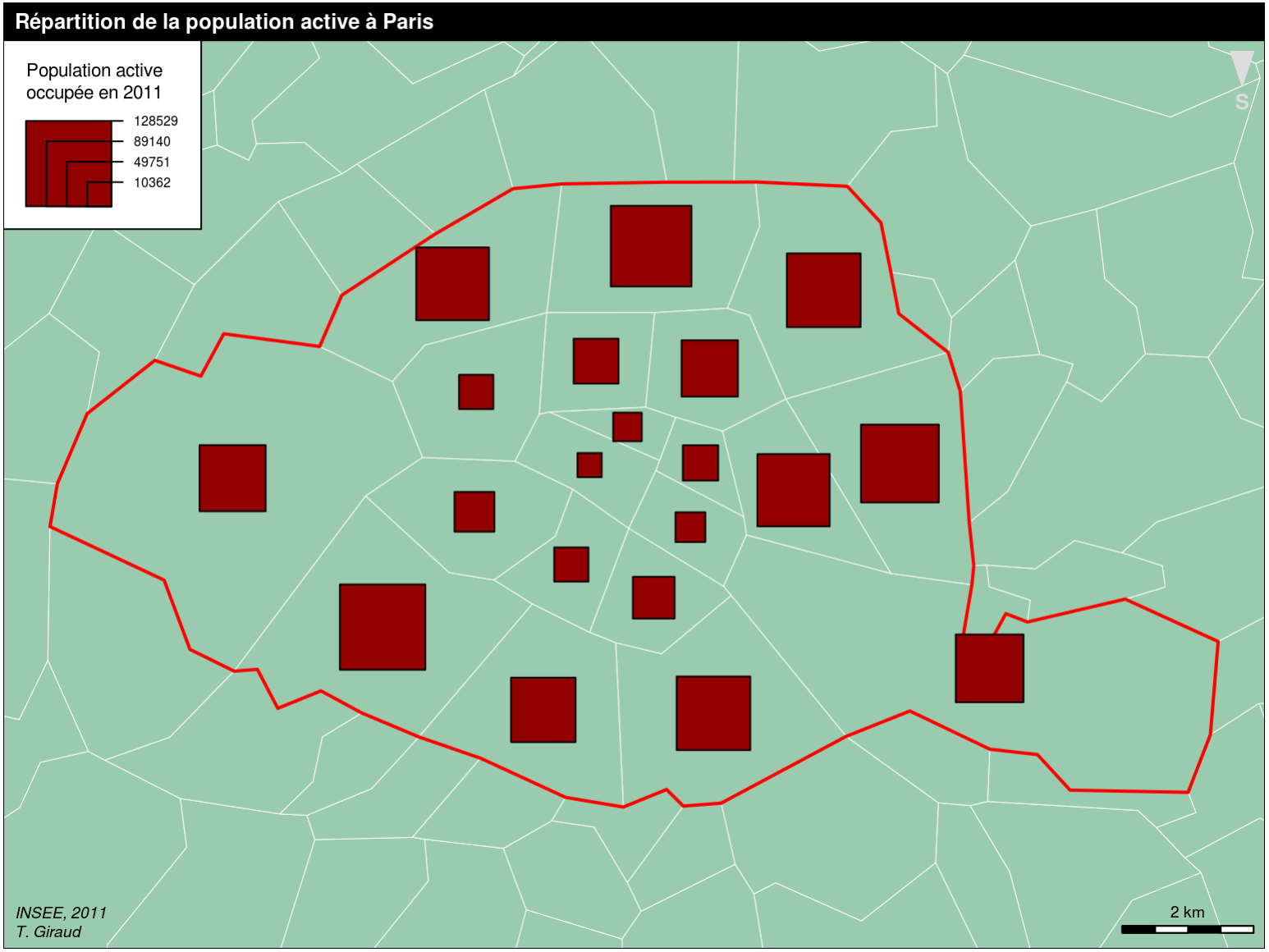


INSEE, 2011
T. Giraud

Autre symbologie

```
# Affichage du fond de carte
par(mar=c(0,0,1.2,0))
# limites de paris
paris <- comidf.spdf[substr(comidf.spdf$ID, 1,2)==75,]
parisC <- rgeos::gUnaryUnion(paris)
# cartographie
plot(parisC, border = "red", lwd = 2, add = F)
plot(comidf.spdf, border = "white", col = "#9BCBAE", lwd = 0.5, add=T)
plot(parisC, border = "red", lwd = 2, add = T)
layoutLayer(title = "Répartition de la population active à Paris",
            sources = "INSEE, 2011", author = "T. Giraud", south = T)
# Affichage des effectifs de la population active occupée
propSymbolsLayer(spdf = paris, df = comidf, var = "total",
                 k = 0.01 ,legend.style = "a", symbols = "squares",
                 legend.pos = "topleft", legend.values.cex = 0.5,
                 legend.title.cex = 0.7, col = "#920000", legend.frame = T,
                 legend.title.txt = "Population active\noccupée en 2011")
```

Répartition de la population active à Paris



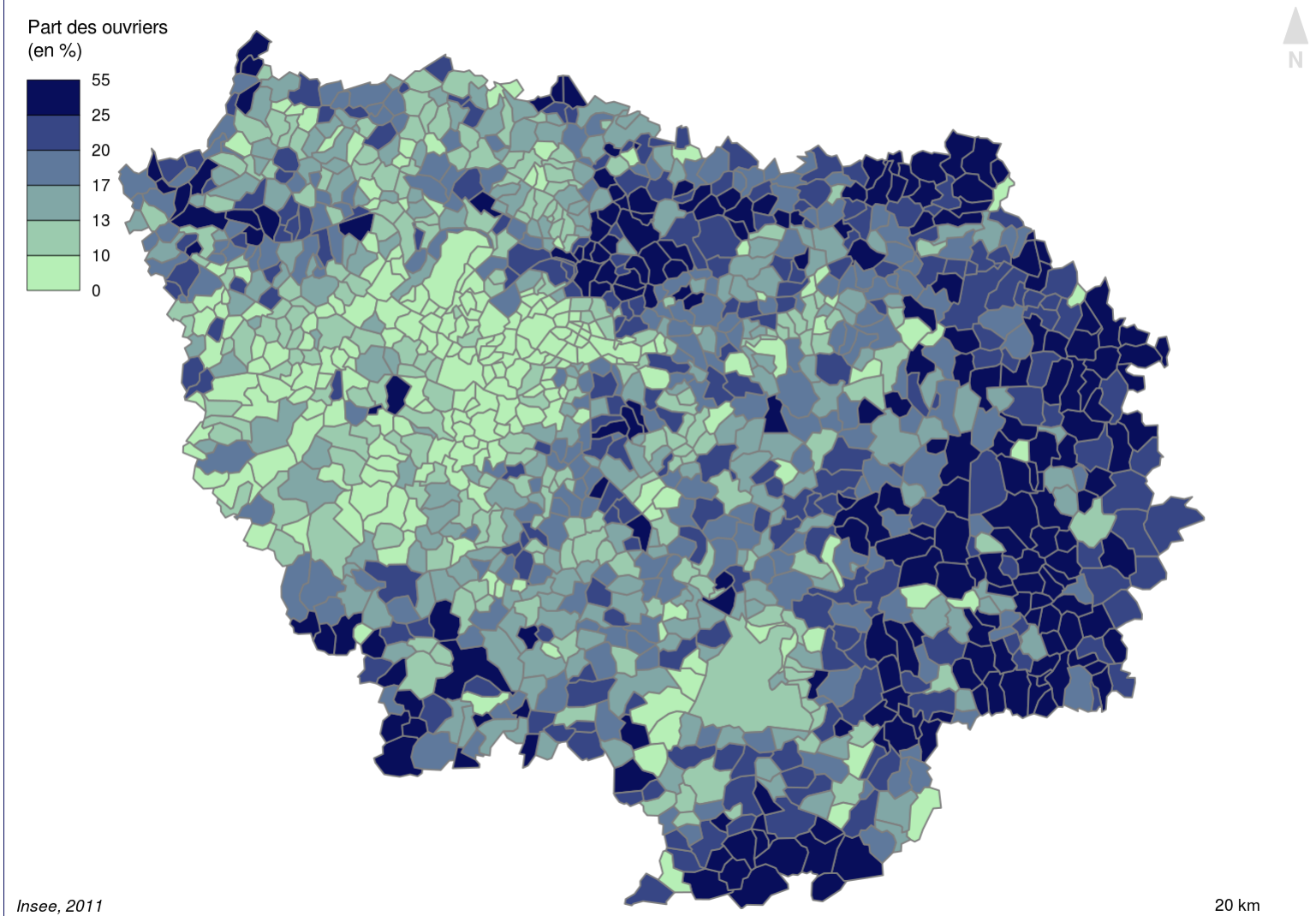
Cartographie choroplèthe

Le package `cartography`

```
# Calcul de la part des ouvriers dans la population active occupée
comidf$partouvrier <- comidf$ouvrier/comidf$total * 100

# Affichage du fond de carte
cols <- carto.pal(pal1 = "turquoise.pal", n1 = 6)
par(mar=c(0,0,1.2,0))
layoutLayer(title = "Répartition des ouvriers en IDF",col = "#080E5B",
            sources = "Insee, 2011", author = "T. Giraud",
            frame = T, north = T, extent = comidf.spdf)
choroLayer(spdf = comidf.spdf, df = comidf, col = cols, border = "grey50",
           legend.title.txt = "Part des ouvriers\n(en %)", legend.pos = "topleft",
           legend.title.cex = 0.7,
           var = "partouvrier", nbclass = 6,
           method = "quantile", add=T)
```

Répartition des ouvriers en IDF



Insee, 2011
T. Giraud

Géotraitements

L'indispensable

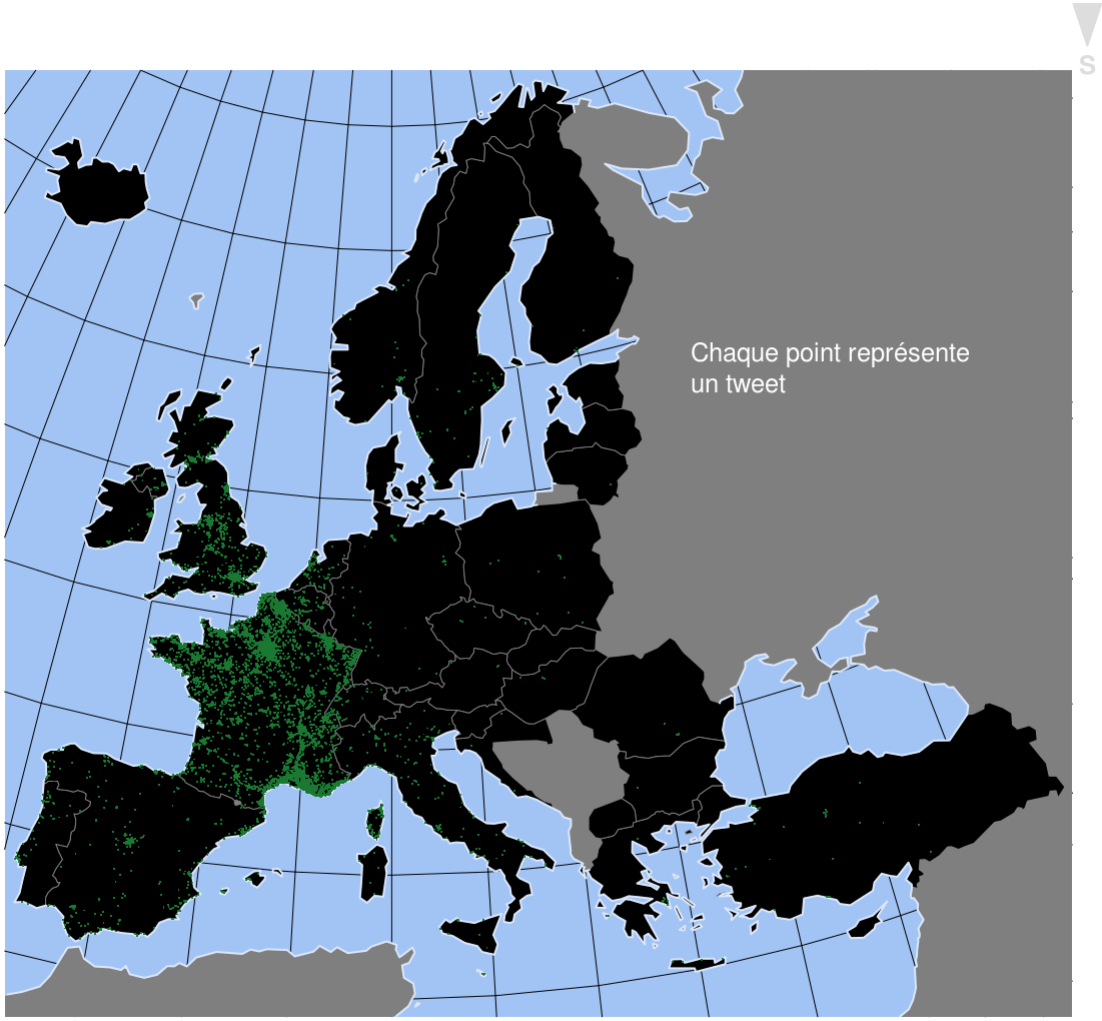
Le package `rgeos` donne accès à la librairie d'opérations spatiales GEOS (Geometry Engine - Open Source).

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

Exemple avec un semis de points

```
# import des données
load("data/tweets.RData")
data(nuts2006)
# affichage des tweets
par(mar = c(0,0,1.2,0))
plot(frame.spdf, col = "#A2C4F5", add= F, border = NA)
plot(graticule.spdf, add=T, border = "white", lwd = 0.5)
plot(countries.spdf, col = "grey50", border = NA, add=T)
plot(nuts0.spdf, add=T, col = "black", border = "grey50", lwd = 0.5)
plot(coasts.spdf, col = '#DCE8FA', add=T)
plot(marseillegeo, col = '#1A7832', pch = ".", add=T)
layoutLayer(frame = F, title = 'Tweets citant Marseille',
            sources = "DMI - Amsterdam, 2015",
            author = "T. Giraud",
            south = T, scale = NULL)
text(x = 5564810,y = 4098997, labels = "Chaque point représente\nun tweet",
     pos = 4, cex = 0.8, col = "white" )
```

Tweets citant Marseille



DMI - Amsterdam, 2015
T. Giraud

Compter des points dans des polygones

```
# intersection des tweets et des pays
tweetsInNuts0 <- over(x = marseillegeo, y = nuts0.spdf)
tweetsInNuts0$nb <- 1
# comptage des tweets dans les polygones
tweetspercountry <- aggregate(x = tweetsInNuts0$nb,
                              by = list(tweetsInNuts0$name),
                              FUN = sum)
names(tweetspercountry) <- c("name", "nbtweets")
head(tweetspercountry[order(tweetspercountry$nbtweets, decreasing = T),])
```

```
##                name nbtweets
## 8            France    65292
## 27 United Kingdom     1225
## 23           Spain      880
## 2          Belgium      302
## 13           Italy      298
## 25     Switzerland      187
```

Création d'une grille régulière de polygones

```
# création d'une grille de points
hexapoints <- spsample(x = nuts0.spdf, cellsize = 40000, type = 'hexagonal')
# transformation en grille de polygones
hexapoly <- HexPoints2SpatialPolygons(hex = hexapoints)
datahexa <- data.frame(id = row.names(hexapoly), x = 1)
row.names(datahexa) <- datahexa$id
hexapoly <- SpatialPolygonsDataFrame(Sr = hexapoly, data = datahexa)
# affichage des hexagones
par(mar = c(0,0,0,0))
plot(hexapoly)
```

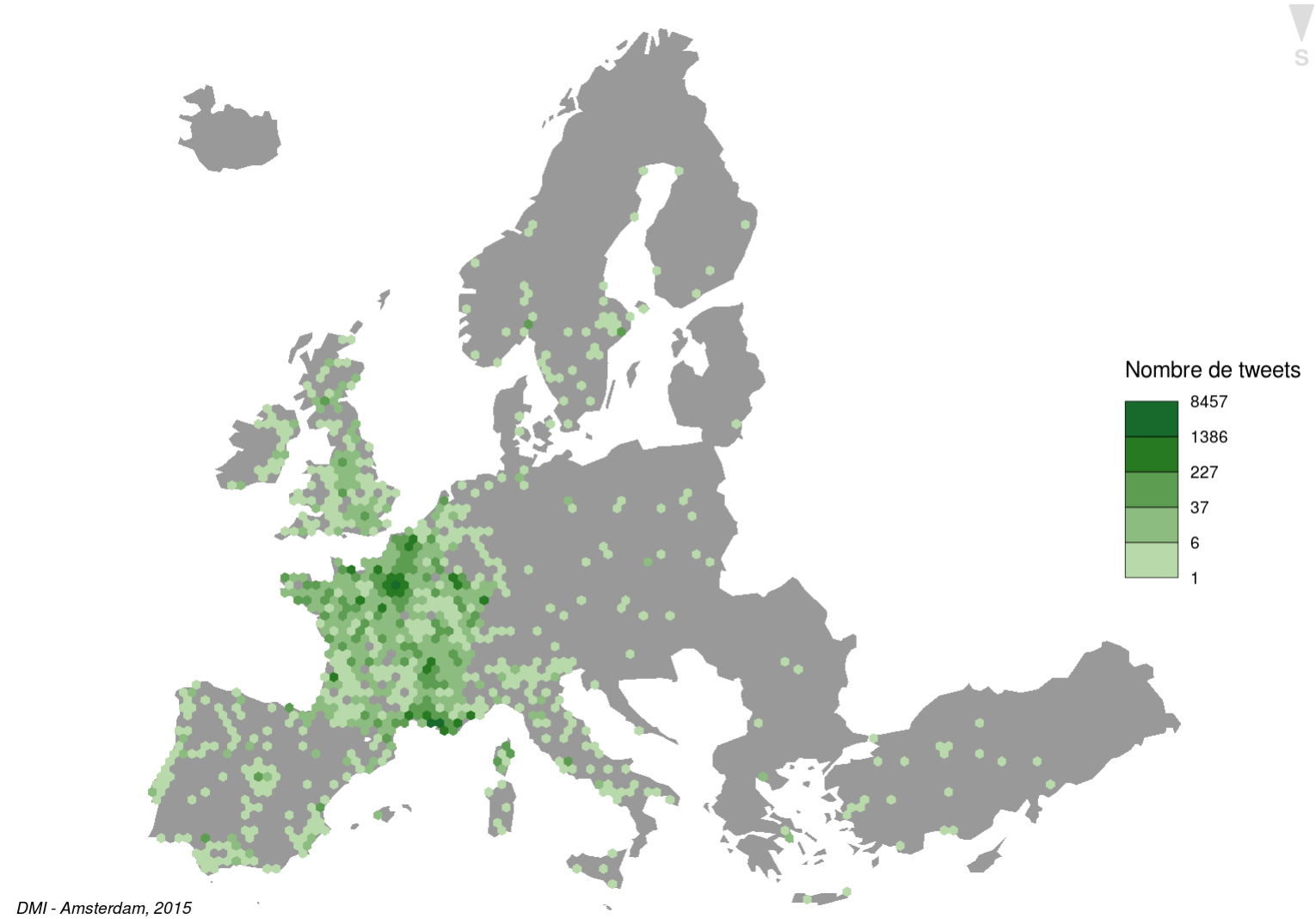
Compter les points dans les polygones

```
# intersection entre les polygones et les tweets
tweetsInHexaPoly <- over(x = marseillegeo, y = hexapoly)
# comptage des tweets dans les polygones
marseilletweethexapoly <- aggregate(x = tweetsInHexaPoly$x,
                                    by = list(tweetsInHexaPoly$id), FUN = sum)
names(marseilletweethexapoly) <- c("id", "n")
```

Cartographie

```
# cartographie des polygones
par(mar = c(0,0,1.2,0))
plot(nuts0.spdf, col = "grey60", border = NA)
choroLayer(spdf = hexapoly, df = marseilletweethexapoly,
           var = "n", border = NA, method = "geom",
           nbclass = 5, legend.pos = "right",
           col = c("#B8D9A9" ,"#8DBC80" ,"#5D9D52" ,
                   "#287A22" ,"#17692C"),
           legend.title.txt = "Nombre de tweets")
layoutLayer(frame = F, title = 'Tweets citant Marseille',
            sources = "DMI - Amsterdam, 2015",
            author = "T. Giraud",
            south = T, scale = NULL)
```

Tweets citant Marseille

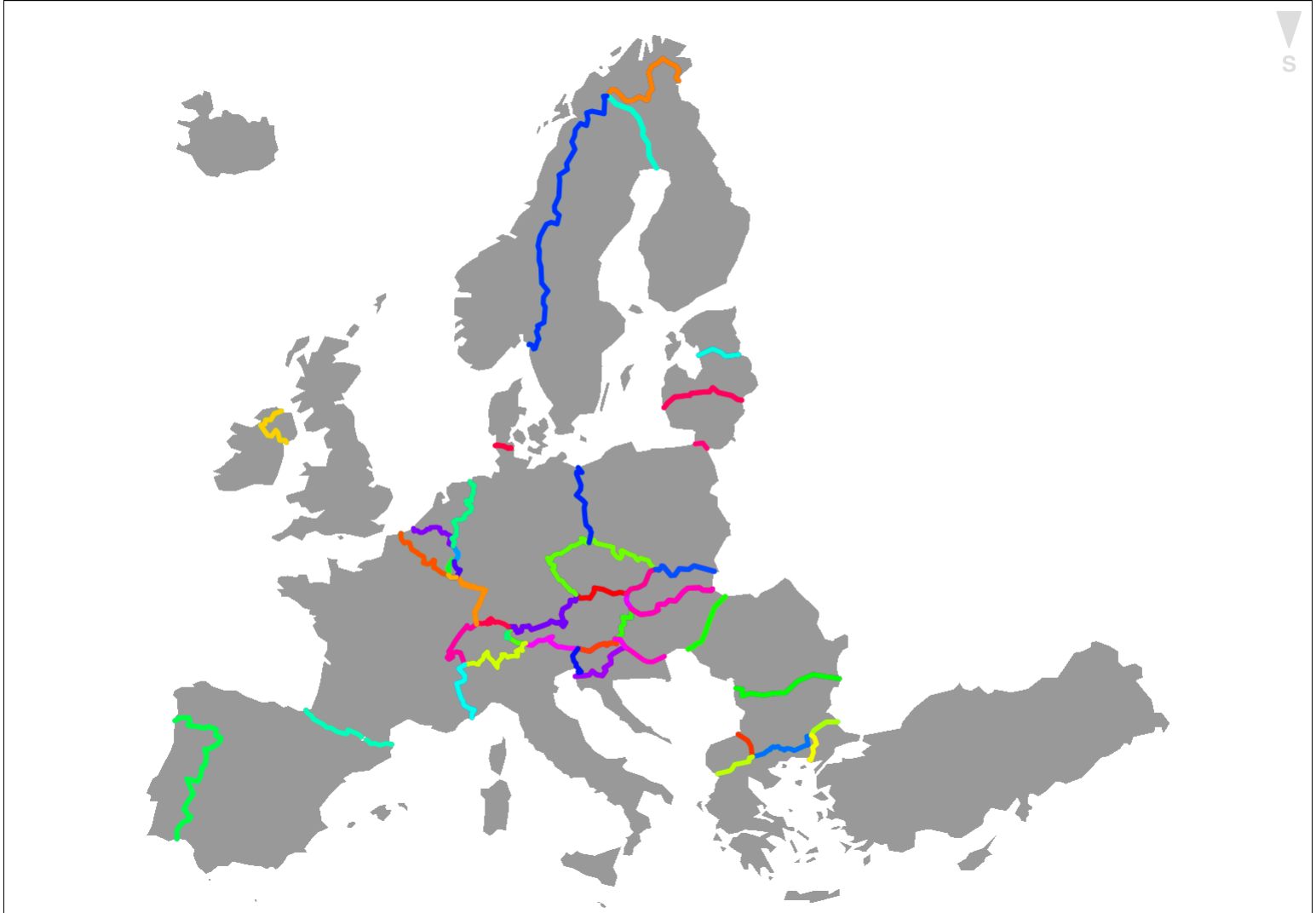


DMI - Amsterdam, 2015
T. Giraud

Extraire des frontières et étudier les discontinuités

```
data(nuts2006)
nuts0.contig.spdf <- getBorders(nuts0.spdf)
nuts0.contig.spdf$col <- sample(x = rainbow(96))
plot(nuts0.spdf, border = NA, col = "grey60")
plot(nuts0.contig.spdf, col = nuts0.contig.spdf$col, lwd = 3, add=T)
layoutLayer(title = 'Frontières inter-étatiques terrestres',
            author = "T. Giraud", sources = "",
            south = T, scale = NULL)
```

Frontières inter-étatiques terrestres



T. Giraud

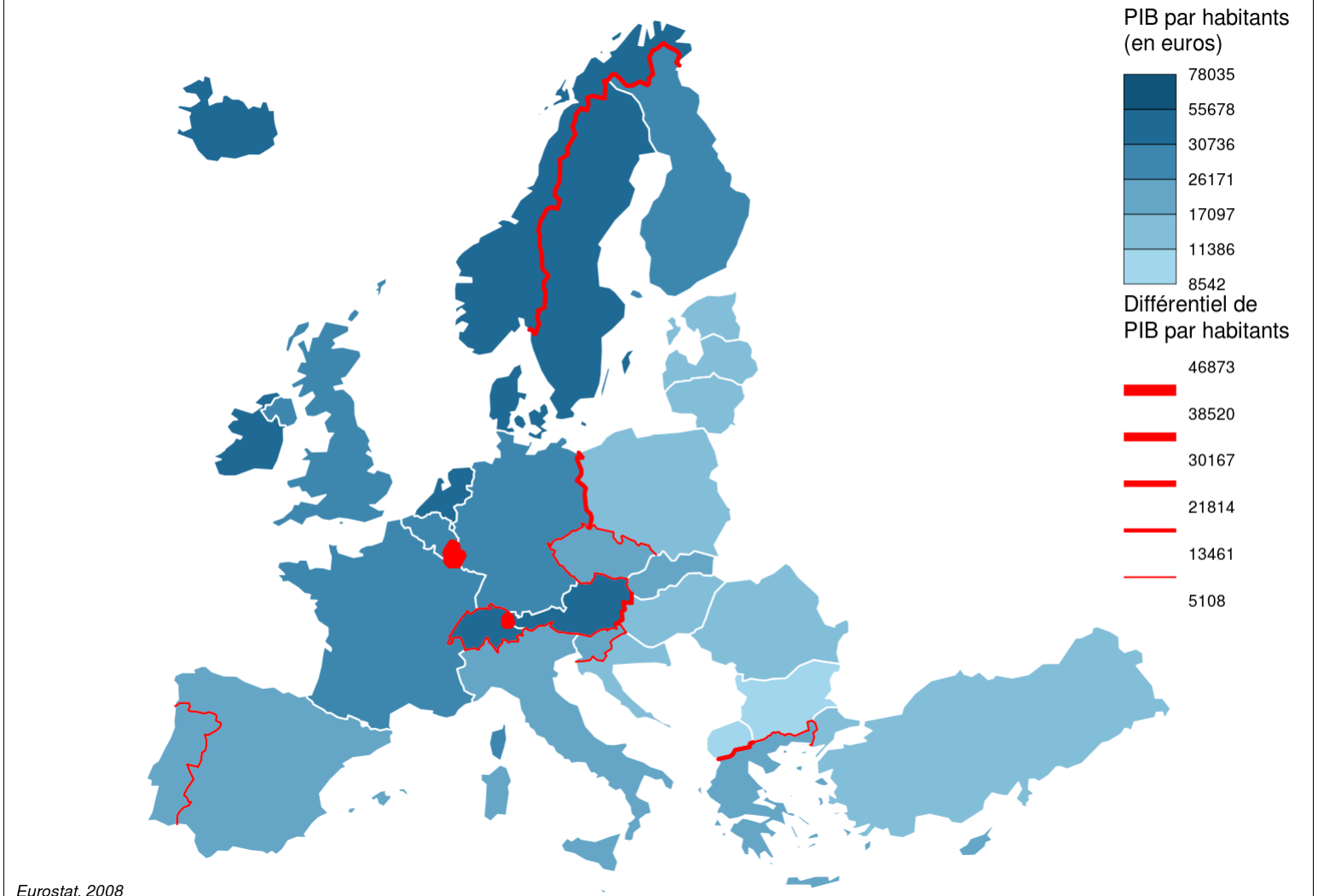
Fonction `getBorders` du package `cartography` :

- gBuffer (créer des zones tampons)
- gInterserction (créer une couche d'intersections)
- gDisjoint (tester l'existence d'intersections)
- gBoundary (extraire des contours)

Cartographie

```
nuts0.df$gdphab <- nuts0.df$gdppps2008 * 1000000 / nuts0.df$pop2008
par(mar = c(0,0,1.2,0))
choroLayer(spdf = nuts0.spdf, df = nuts0.df,
           var = "gdphab", method = "q6",
           legend.title.txt = "PIB par habitants\n(en euros)",
           legend.values.rnd = 0, legend.pos = "topright", add=F)
plot(nuts0.spdf, col=NA, lwd=1, border="white", add=T)
discLayer(spdf = nuts0.contig.spdf, df = nuts0.df,
          dfid = "id", spdfid1 = "id1", spdfid2 = "id2",
          var = "gdphab", col="red", nbclass=5,
          legend.pos = "right", legend.values.rnd = 0,
          legend.title.txt = "Différentiel de \nPIB par habitants",
          method="equal", threshold = 0.5, sizemin = 1,
          sizemax = 8, type = "abs", add=TRUE )
layoutLayer(title = 'Différentiels de richesse en Europe',
            author = "T. Giraud", sources = "Eurostat, 2008",
            scale = NULL)
```

Différentiels de richesse en Europe



Eurostat, 2008
T. Giraud

Mais aussi…

Des liens sont possibles avec des **SIG** ou des **bases de données spatiales** :

- GRASS (`spgrass6`, `rgrass7`)
- QGIS (via Processing, anciennement sextante)
- ArcGis (**RPyGeo** via Python)
- PostGis / PostGreSQl (`RPostgreSQL`)

Lissage spatial

Le package `SpatialPosition`

Des modèles d'interaction spatiale permettant de saisir l'influence exercée pa un lieu sur tous les autres.

Le modèle de Stewart (accessibilité potentielle, potentiel gravitationnel ou accessibilité gravitationnelle) :

$$A_i = \sum_{j=1}^{n} O_j f(d_{ij})$$

- A_i potentiel en i
- O_j stock de population sur j
- $f(d_{ij})$ fonction négative de la distance entre i et j

Exemple sur la capacité des hopitaux parisiens

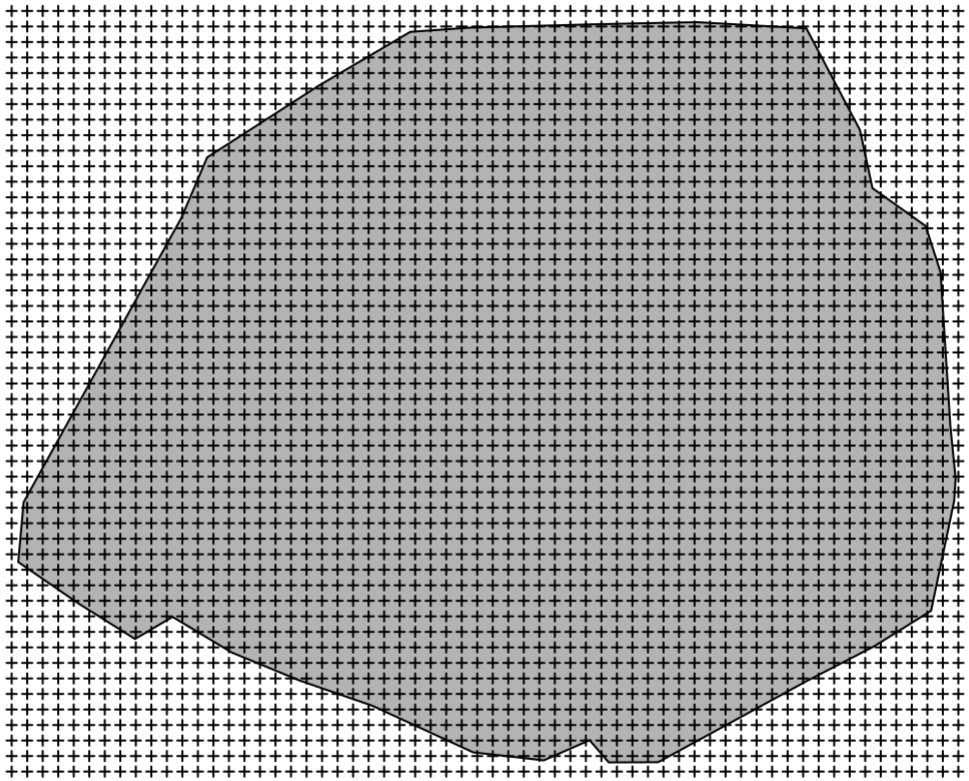
```
library(cartography)
library(SpatialPosition)
data(spatData)
par(mar = c(0,0,1.2,0))
plot(spatMask, col = "grey70")
propSymbolsLayer(spdf = spatPts, df = spatPts@data, var = "Capacite",
                 legend.title.txt = "Nb. de lits", col = "#920000",
                 legend.pos = "topleft")
layoutLayer(title = "Répartitions des hopitaux parisiens", south = T,
            scale = 2)
```

Répartitions des hopitaux parisiens

Source(s)
Author(s)

Calcul des potentiels

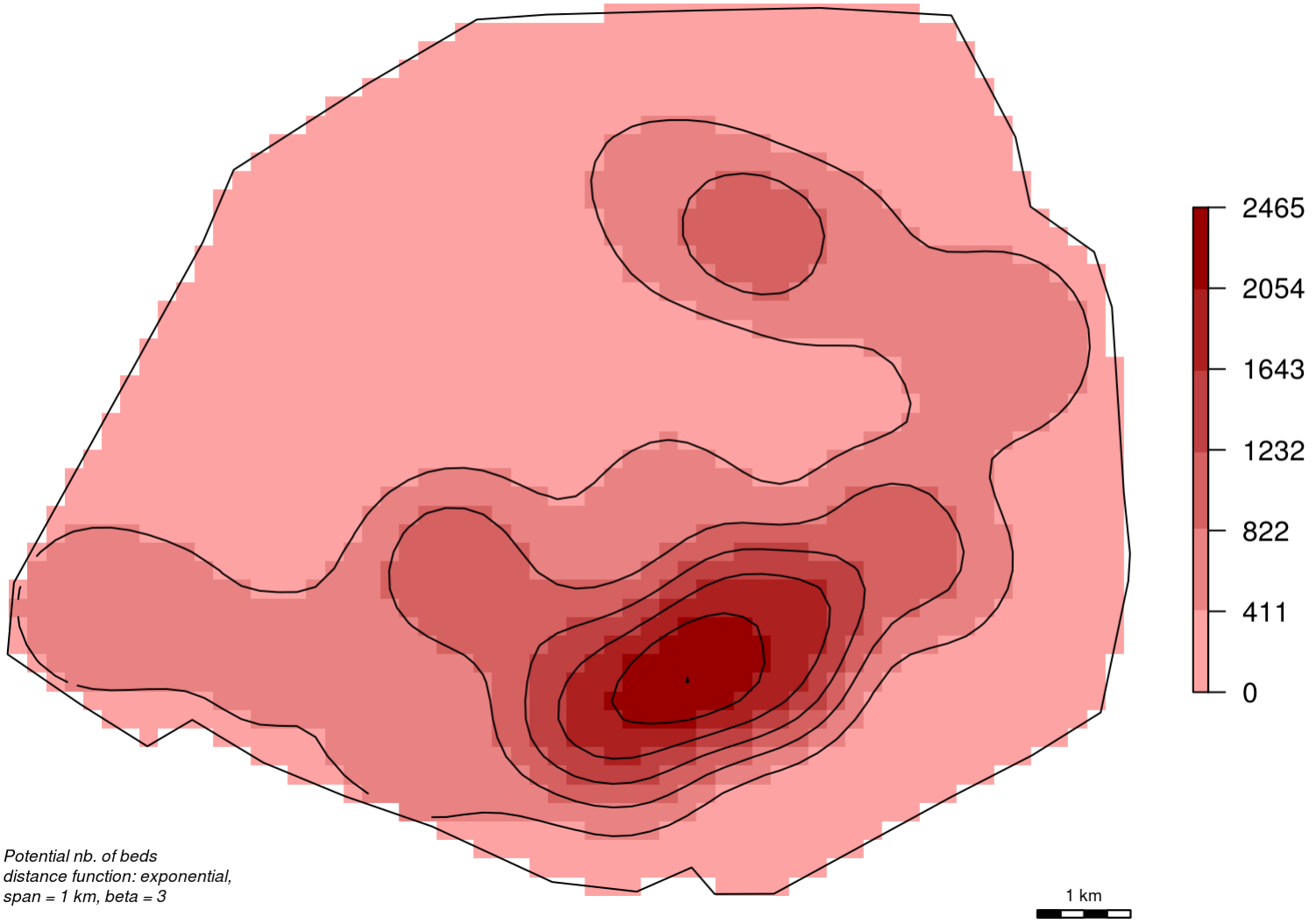
```
hopPot <- stewart(knownpts = spatPts, varname = "Capacite",
                  typefct = "exponential", span = 1000, beta = 3,
                  resolution = 200,
                  mask = spatMask)
plot(spatMask, col = "grey70")
plot(hopPot, cex = 0.5, add=T)
```



Affichage de la carte

```
# création d'un raster
rasterAccessibility <- rasterStewart(x = hopPot, mask = spatMask)
# Affichage du raster
par(mar=c(0,0,1.2,2))
breakValues <- plotStewart(x = rasterAccessibility, add = F, nclass = 6)
# Create contour lines and add them to the plot
contLines <- contourStewart(x = rasterAccessibility, breaks = breakValues)
plot(contLines, add = TRUE)
plot(spatMask, add = TRUE)
sources <-  "Potential nb. of beds\ndistance function: exponential,\nspan = 1 km, bet
layoutLayer(title = "Global Accessibility to Public Hospitals",
            frame = FALSE, sources = sources,
            author = "", coltitle = "black", col = NA)
```

Global Accessibility to Public Hospitals



Potential nb. of beds
distance function: exponential,
span = 1 km, beta = 3

Exemple dans un maillage et avec une matrice d distances

Préparation des données

```
library(SpatialPosition)
# Jointure fond de carte / données
comidf.spdf@data <- data.frame(comidf.spdf@data,
                               comidf[match(comidf.spdf@data$ID,comidf$ID), ])
row.names(comidf.spdf) <- as.character(comidf.spdf@data$ID)
# Ordonner les lignes et les colonnes de la matrice
roadDist <- roadDist[comidf.spdf$ID, comidf.spdf$ID]
```

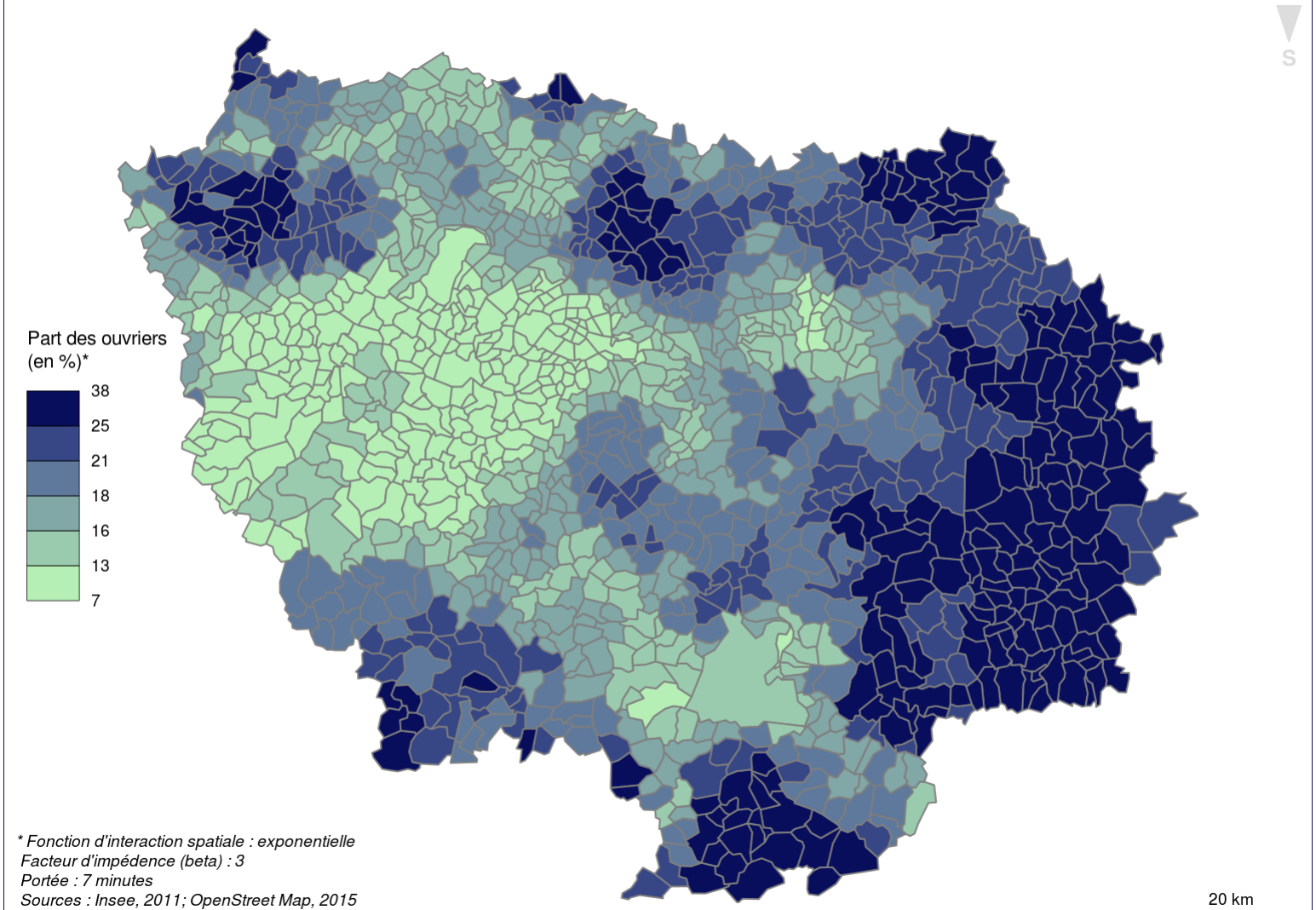
Calcul des potentiels

```
# Potentiel d'ouvriers dans un voisinnage de 7 minutes
ouv <- stewart(knownpts = comidf.spdf, unknownpts = comidf.spdf,
               matdist = roadDist, typefct = "exponential",
               varname = "ouvrier", span = 7, beta = 3)
# Potentiel de population active dans un voisinnage de 7 minutes
tot <- stewart(knownpts = comidf.spdf, unknownpts = comidf.spdf,
               matdist = roadDist, typefct = "exponential",
               varname = "total", span = 7, beta = 3)
pot <- data.frame(id = tot$ID, partouvrierpot = ouv$OUTPUT/tot$OUTPUT * 100)
```

Cartographie

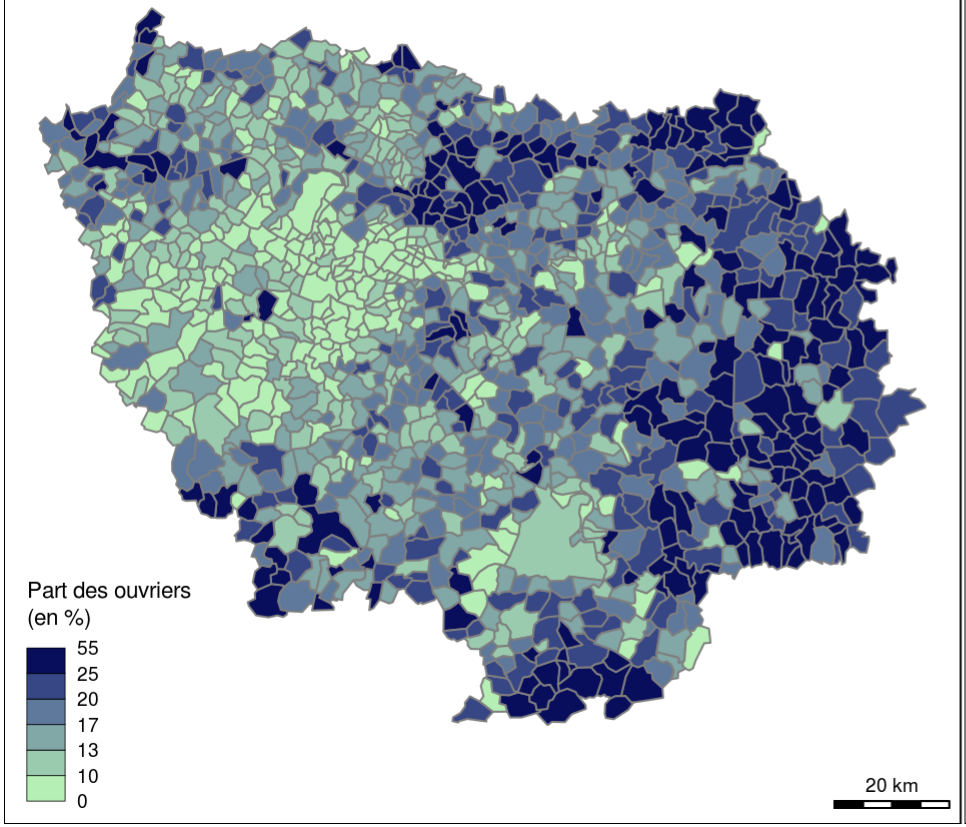
```
par(mar=c(0,0,1.2,0))
cols <- carto.pal(pal1 = "turquoise.pal", n1 = 6)
choroLayer(spdf = comidf.spdf, df = pot, col = cols,border = "grey50",
           legend.title.txt = "Part des ouvriers\n(en %)*",
           legend.pos = "left",
           legend.title.cex = 0.7,
           var = "partouvrierpot", nbclass = 6,
           method = "quantile", add=F)
titre <- "Répartition des ouvriers en IDF - Voisinnage fonctionnel de 7 minutes en vo
layoutLayer(title = titre,
            col = "#080E5B",
            sources = "* Fonction d'interaction spatiale : exponentielle
            Facteur d'impédence (beta) : 3
            Portée : 7 minutes
            Sources : Insee, 2011; OpenStreet Map, 2015 ", author = "T. Giraud",
            frame = T, south = T)
```

Répartition des ouvriers en IDF - Voisinnage fonctionnel de 7 minutes en voiture

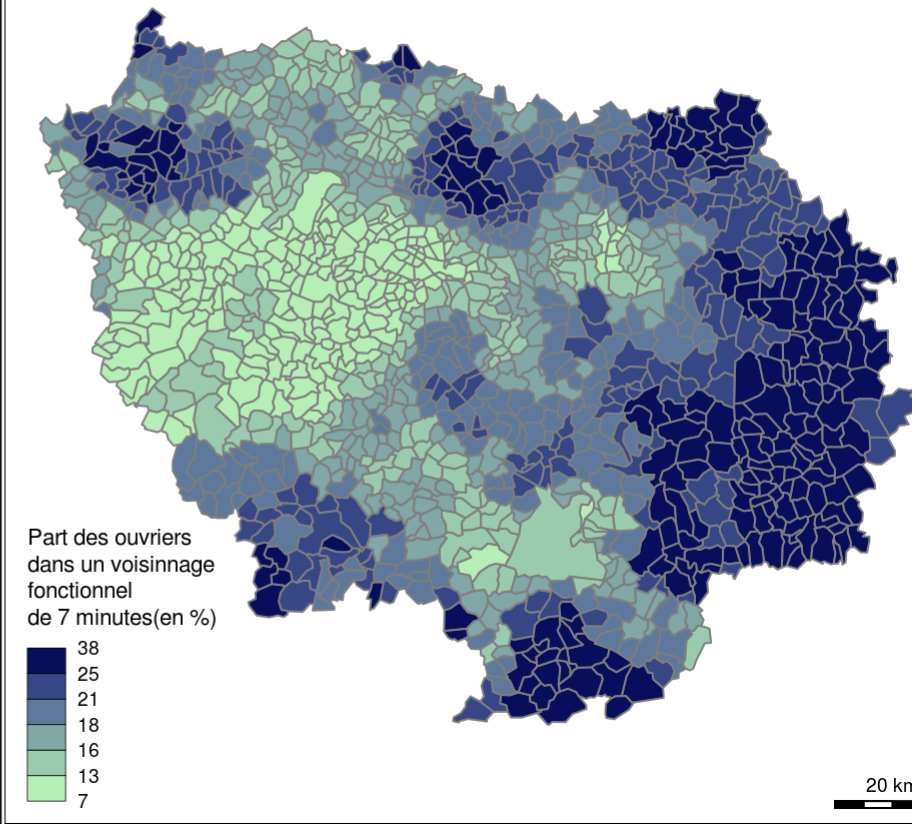


** Fonction d'interaction spatiale : exponentielle*
Facteur d'impédence (beta) : 3
Portée : 7 minutes
Sources : Insee, 2011; OpenStreet Map, 2015
T. Giraud

Répartition des ouvriers en IDF - Sans lissage

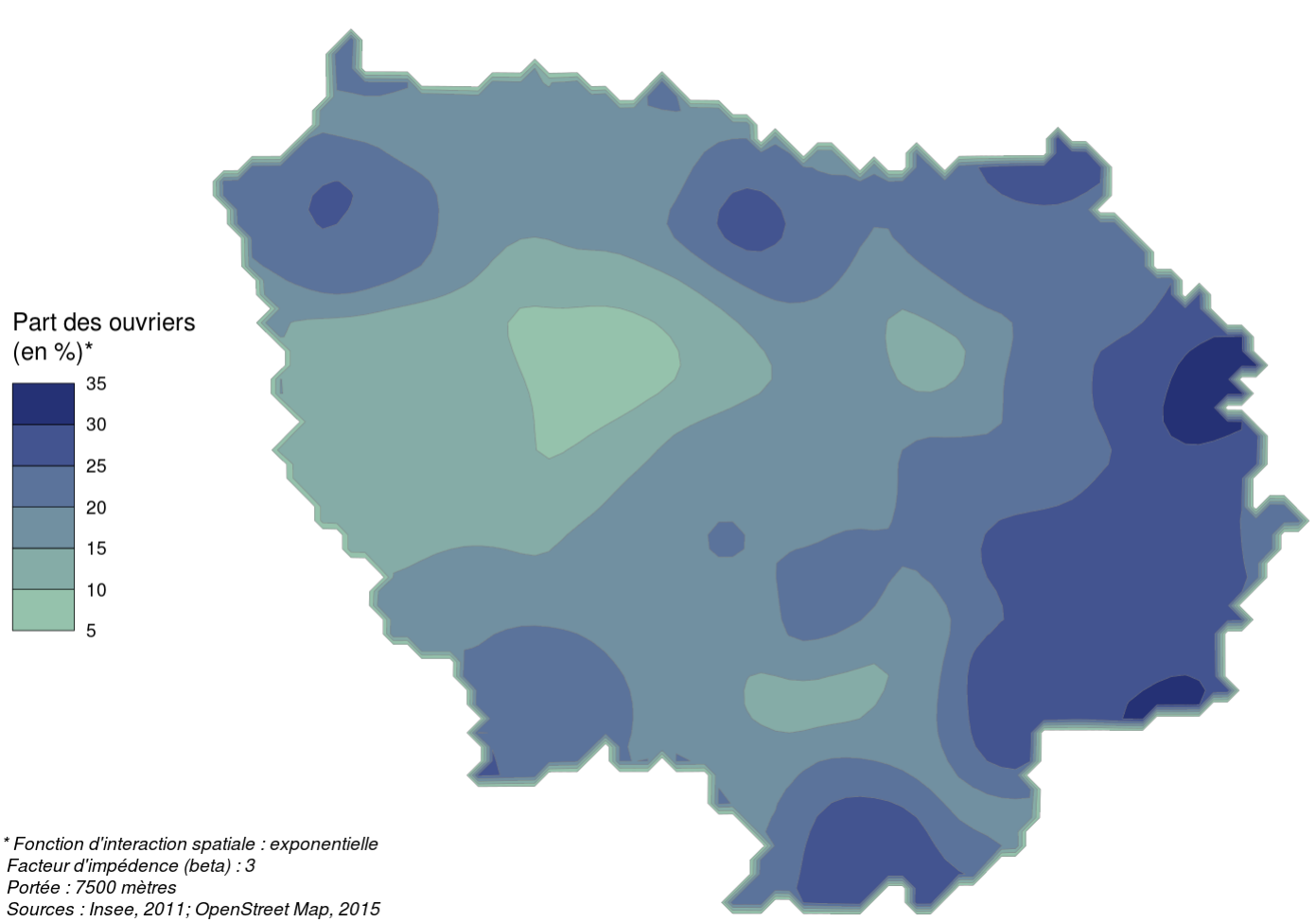


Répartition des ouvriers en IDF - Avec lissage



Représentation continue

Répartition des ouvriers en IDF - Voisinnage fonctionnel de 7,5 km



** Fonction d'interaction spatiale : exponentielle*
Facteur d'impédence (beta) : 3
Portée : 7500 mètres
Sources : Insee, 2011; OpenStreet Map, 2015

Applications web

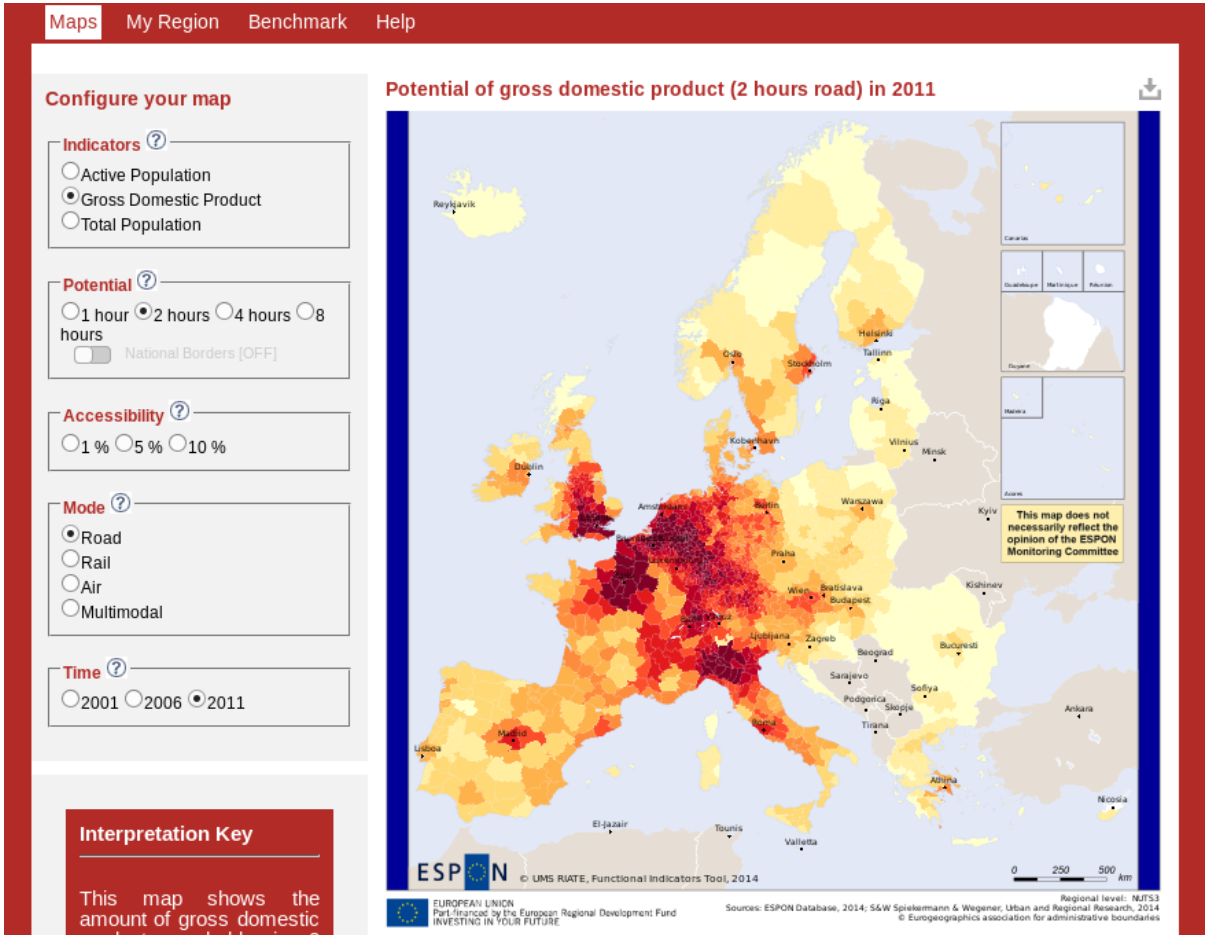
Une application spécialisée : Espon Functional Indicators Tool

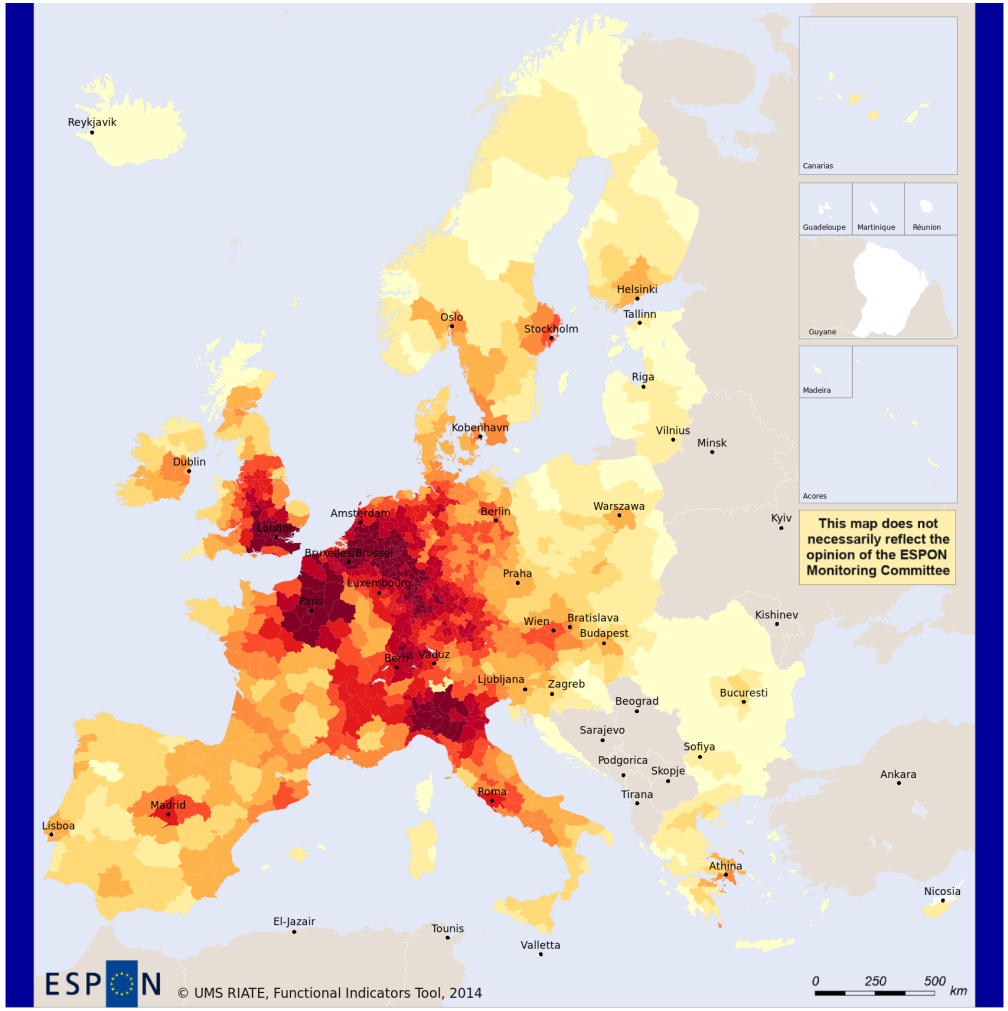
Demande précise et fortes contraintes éditoriales

Une application générique : Func-i

Cartes interactives et déploiement facilité

Espon Functional Indicators Tool





EUROPEAN UNION
Part-financed by the European Regional Development Fund
INVESTING IN YOUR FUTURE

Regional level: NUTS3
Sources: ESPON Database, 2014; S&W Spiekermann & Wegener, Urban and Regional Research, 2014
© Eurogeographics association for administrative boundaries

Amount of Gross Domestic Product reachable in 2h00
(in millions of euros, mode of transp. : Road)

No data

220 9 840 27 300 55 500 91 600 134 000 180 000 237 000 326 000 680 000

Func-i

Maps Graphs Help

Configure your map

Indicators

- Total Population
- Farmers
- Artisans
- Managers
- Intermediate Occupations
- Employees
- Labourers
- Occupied Active Population

Potential

- 0 meters
- 2500 meters
- 5000 meters
- 7500 meters
- 10000 meters

Accessibility

- 0.5 %
- 1 %
- 5 %

Mode

- Euclidean Distance
- Car Distance

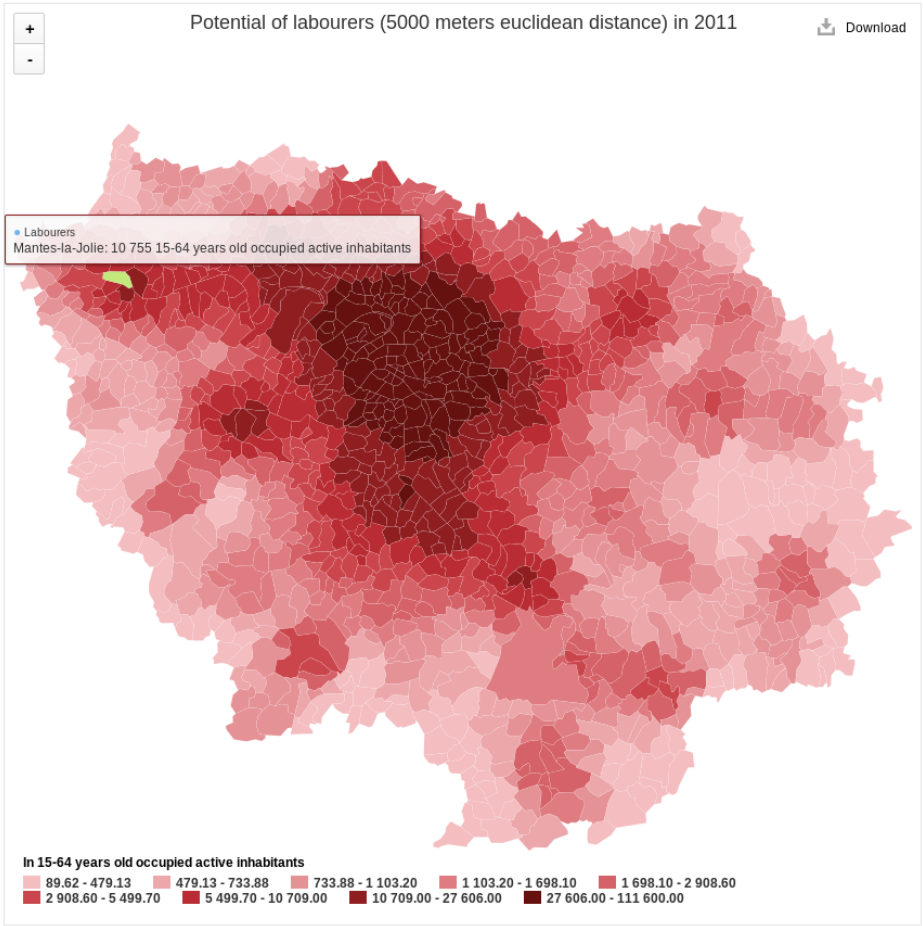
Interpretation Key

This map shows the amount of labourers reachable in 5000 meters by euclidean distance in 2011.

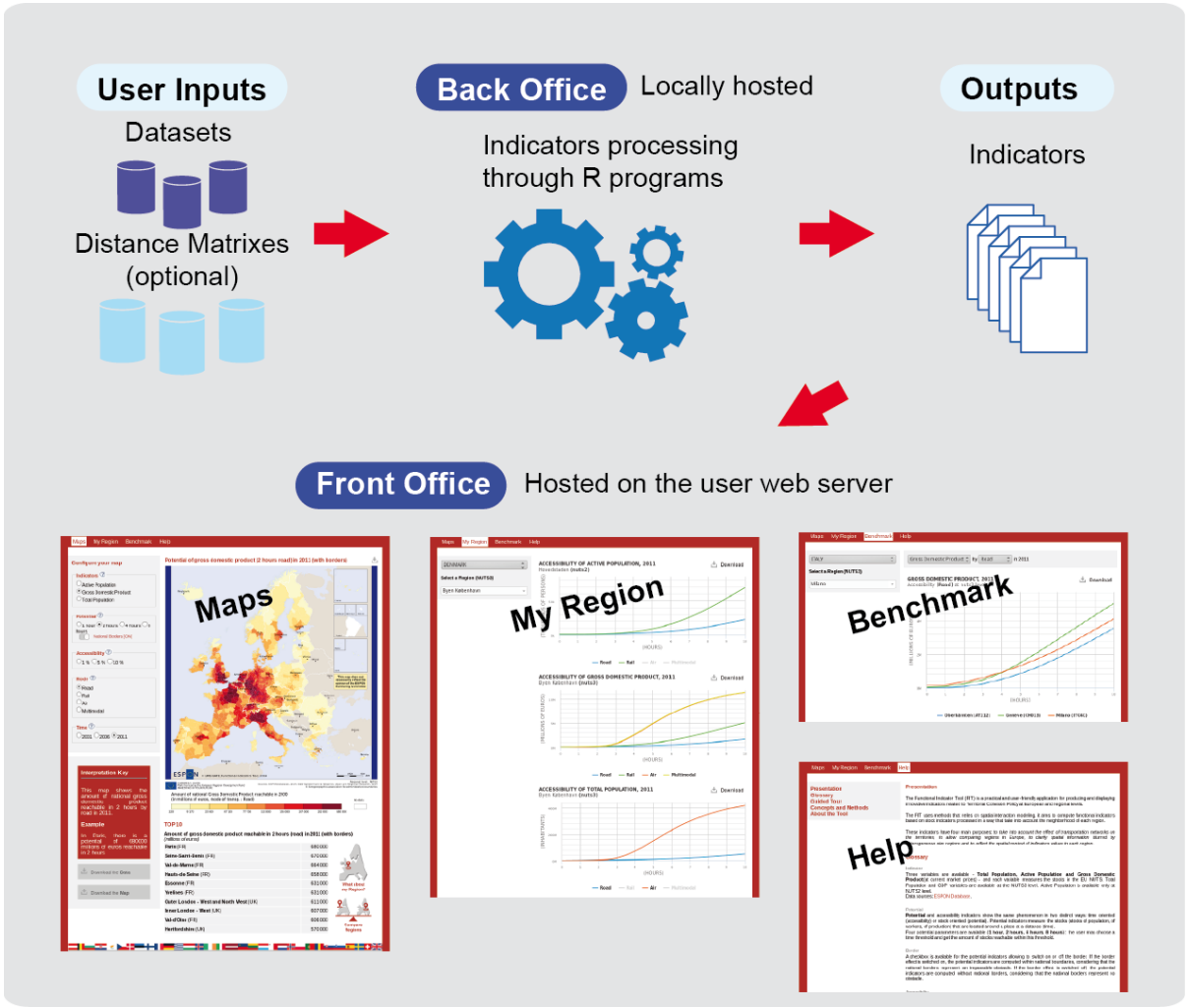
Example

In Paris 19e Arrondissement, there is a potential of 112000 15-64 years old occupied active inhabitants reachable in 5000 meters.

Download the **Data**



Architecture de l'appli



Ressources

Le Groupe ElementR

Les activités de ce groupe consistent à organiser des séances de formation et à produire des matériaux pédagogiques sur l'utilisation du logiciel R en sciences humaines et sociales.

R et espace. Traitement de l'information géographique

Séances 2014 - 2015

- Recherche reproductible et création de packages
- Régression multiniveaux
- Analyse de flux et modèles gravitaires
- Analyse de données spatio-temporelles
- Cartographie de flux
- Créer des applications interactives avec R et shiny
- Analyse de rasters

http://elementr.hypotheses.org/

Quelques ressources en ligne

- *Mailing list* R-sig-Geo
  Mailing list dédiée aux discussions sur l'utilisation des packages et données géographiques dans R
- http://www.r-bloggers.com/
  Agrégateur de blogs sur R, généraliste mais les post géographiques sont assez fréquents.
- http://rgeomatic.hypotheses.org/
  Carnet de recherche de Timothée Giraud destiné au partage et à la diffusion de travaux de géomatique réalisés avec R.
- http://neocarto.hypotheses.org/
  Carnet de recherche de Nicolas Lambert plus généraliste sur la cartographie et les nouvelles technologies liées.
- Cartographie et Analyse Spatiale avec R
  Session du séminaire R à l'Usage des Sciences Sociales organisé par l'INED et l'EHESS en 2014.
- Représentation et traitement de l'information géographique avec R : usage avancé
  Session du séminaire R à l'Usage des Sciences Sociales organisé par l'INED et l'EHESS en 2015.
- Cartographie et Analyse vectorielle avec R
  Présentation à la journée de conférences et ateliers sur les logiciels libres en géomatique (be-OpenGIS-fr).

Merci de votre attention!

package `cartography` :
https://github.com/Groupe-ElementR/cartography/

dépot github : https://github.com/rCarto/

blog : http://rgeomatic.hypotheses.org/

e-mail : timothee.giraud@ums-riate.fr