
Recueil de données

Devoir Maison

Maxime ABITBOL

Table des matières

1	Exercice 1 -	3
1.1	Charger les contours des départements français contenus dans le répertoire <code>exo6_dep</code> .	3
1.2	Calculer à partir des données communales contenues dans le fichier <code>exo6_data.csv</code> des taux de naissances par départements.	3
1.3	Joindre les deux tables et vérifier que des données sont associées à chaque départements.	4
1.4	Exporter les données en geojson.	5
1.5	Réaliser une carte coroplèthe avec ces données.	5
2	Exercice 2 -	9
2.1	Calculer et afficher les voronois associés aux stations velib de new-york.	9
2.2	Utiliser pour cela la fonction <code>st_as_sf</code> avec l'option <code>coords</code> .	9
3	Exercice 3 -	11
3.1	Faire une carte interactive des monuments historiques à paris	11
3.2	Faire une carte en symbol proportionel avec le nombre de monuments par iris	11
4	Exercice 4 -	12
4.1	Faire une carte en symboles proportionels du nombre de vélos dans les stations vélib à NewYork.	12
4.2	Utiliser un fond de carte open street map avec la library cartography	12

1 Exercice 1 -

```
# Imports
library(sf)
library(leaflet)
library(cartography)
library(tidyverse)
library(tidyr)
library(cartography)
library(rjson)
```

1.1 Charger les contours des départements français contenus dans le répertoire `exo6_dep`.

```
dep <- st_read("data/exo6_dep/dep.shp")
```



Figure 1: Contour de la France et de ses DOM-TOM

1.2 Calculer à partir des données communales contenues dans le fichier `exo6_data.csv` des taux de naissances par départements.

```
data_com <- read.csv("data/exo6_data.csv", encoding="UTF-8")
data_com <- subset(data_com, data_com$code_reg != 'NA')
```

```
naissance_france_2011 <- sum(data_com$nombre_naissances_2011)

#aggregate(data_com$nombre_naissances_2011,
#          by=list(Category=data_com$code_dept), FUN=sum)

taux_naissances <- data_com %>%
  group_by(code_dept) %>%
  summarise(nombre_naissance = sum(nombre_naissances_2011),
            population_2011=sum(Population_2011)) %>%
  mutate(taux_naissance = nombre_naissance/population_2011*100)

head(taux_naissances)
```

```
## # A tibble: 6 x 4
##   code_dept nombre_naissance population_2011 taux_naissance
##   <fct>          <int>          <int>          <dbl>
## 1 1              7265          603827          1.20
## 2 10             3635          303997          1.20
## 3 11             3789          359967          1.05
## 4 12             2682          275813          0.972
## 5 13            25889          1975896          1.31
## 6 14             8041          685262          1.17
```

1.3 Joindre les deux tables et vérifier que des données sont associées à chaque départements.

```
dep2 <- data.frame(dep$code_insee, dep$nom)

data_com2 <- data_com %>%
  left_join(taux_naissances)
```

```
## Joining, by = "code_dept"
```

```
# data_com2 <- merge(data_com,taux_naissances)
head(taux_naissances)
```

```
## # A tibble: 6 x 4
##   code_dept nombre_naissance population_2011 taux_naissance
##   <fct>          <int>          <int>          <dbl>
## 1 1              7265          603827          1.20
## 2 10             3635          303997          1.20
## 3 11             3789          359967          1.05
## 4 12             2682          275813          0.972
## 5 13            25889          1975896          1.31
## 6 14             8041          685262          1.17
```

```
head(subset(data_com2, data_com2$code_dept==1))[,1:4]
```

```
##   Catégorie code_reg code_dept code_zoneempl
## 41 Métropole      82         1         8202
## 208 Métropole     82         1         8201
```

```
## 294 Métropole      82      1      8216
## 663 Métropole      82      1      8201
## 779 Métropole      82      1      8219
## 780 Métropole      82      1      8201
```

```
head(subset(data_com2, data_com2$code_dept==13))[,1:4]
```

```
##      Catégorie code_reg code_dept code_zoneempl
## 1912 Métropole      93      13      9308
## 5930 Métropole      93      13      9311
## 6292 Métropole      93      13      9308
## 7232 Métropole      93      13      9312
## 7320 Métropole      93      13      9312
## 7482 Métropole      93      13      9312
```

```
data_taux_naissance = taux_naissances %>%
  mutate(code_dept=as.character(code_dept)) %>%
  mutate(code_dept = ifelse(str_length(code_dept)==1,
                             paste0("0",code_dept),code_dept))
```

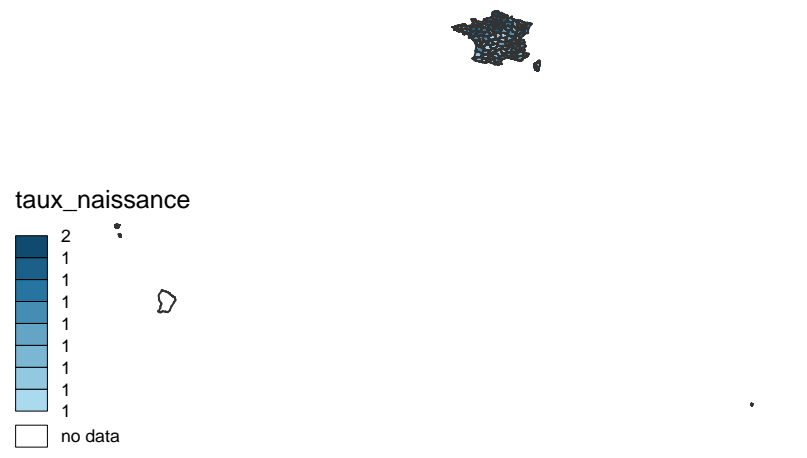
1.4 Exporter les données en geojson.

```
data_json <- toJSON(unname(split(data_com2, 1:nrow(data_com2))))
cat(data_json)
write(data_json, "data/data_exo6.GeoJSON")
```

1.5 Réaliser une carte coroplèthe avec ces données.

```
carte = dep %>%
  left_join(data_taux_naissance,by=c("code_insee"="code_dept"))
```

```
# ?choroLayer
choroLayer(carte,var="taux_naissance", colNA = "white")
```



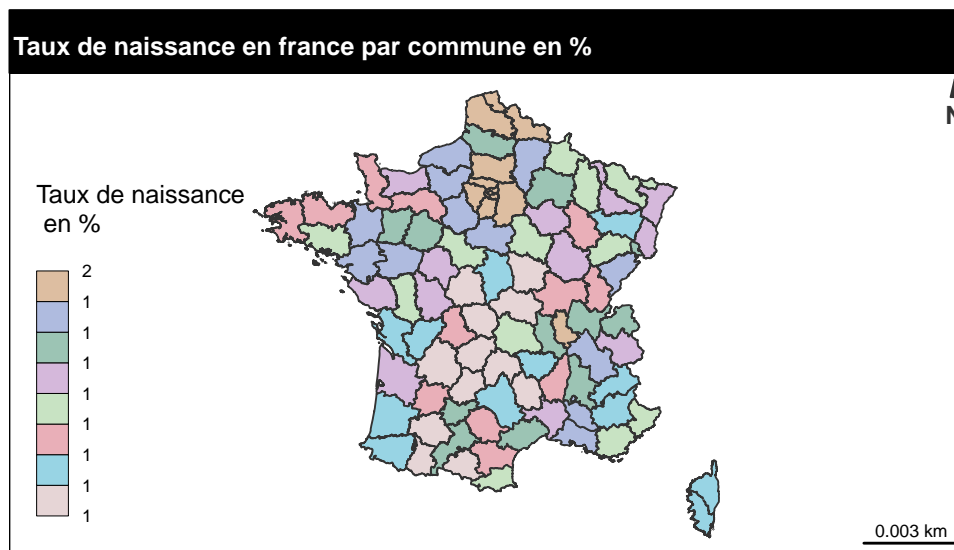
On remarque ici que, plus la couleur est claire, plus le taux de naissance est faible.

On remarque que certaines données sont manquantes, on peut donc refaire la carte en filtrant ces données manquantes.

```
carte2 = dep %>%
  left_join(data_taux_naissance, by=c("code_insee"="code_dept")) %>%
  filter(!is.na(taux_naissance))

choroLayer(carte2, var="taux_naissance", border="grey20",
  lwd=1, legend.pos = "bottomleft",
  legend.title.cex = 0.8,
  legend.title.txt = "Taux de naissance \n en %",
  legend.values.cex = 0.6,
  legend.values.rnd = 0, legend.nodata = "no data",
  legend.frame = FALSE, legend.border = "black",
  legend.horiz = FALSE, add = FALSE,
  col = carto.pal(pal1 = "pastel.pal", n1 = 8))
north(pos = "topright")

layoutLayer(title = "Taux de naissance en france par commune en %")
```



On peut constater que l'affichage en pourcentage n'est pas optimal, on peut pour cela modifier et mettre en pourmille.

```

taux_naissances_1000 <- data_com %>%
  group_by(code_dept) %>%
  summarise(nombre_naissance = sum(nombre_naissances_2011),
            population_2011=sum(Population_2011)) %>%
  mutate(taux_naissance = nombre_naissance/population_2011*1000)

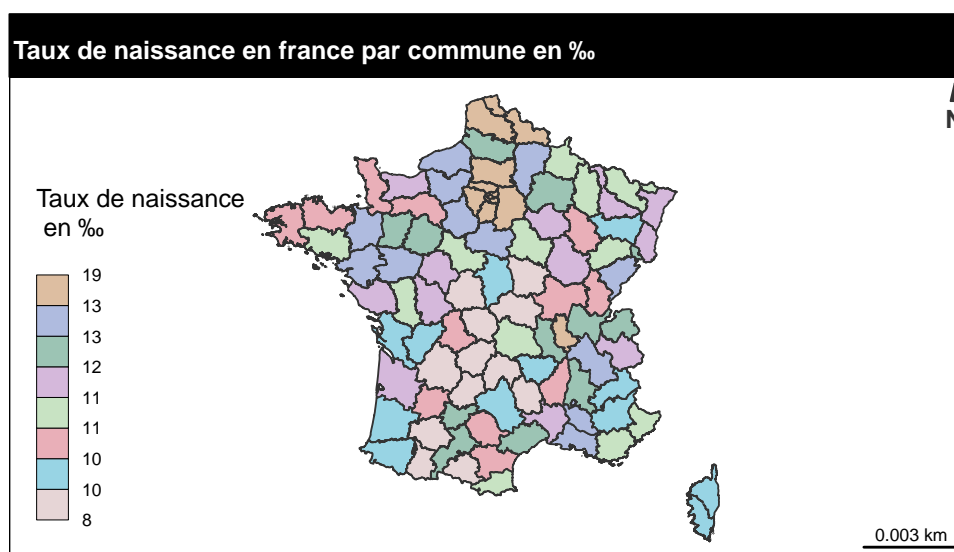
data_taux_naissance_1000 = taux_naissances_1000 %>%
  mutate(code_dept=as.character(code_dept)) %>%
  mutate(code_dept = ifelse(str_length(code_dept)==1,
                            paste0("0",code_dept),code_dept))

carte3 = dep %>%
  left_join(data_taux_naissance_1000,by=c("code_insee"="code_dept")) %>%
  filter(!is.na(taux_naissance))

choroLayer(carte3, var="taux_naissance", border="grey20",
           lwd=1, legend.pos = "bottomleft",
           legend.title.cex = 0.8,
           legend.title.txt = "Taux de naissance \n en %",
           legend.values.cex = 0.6,
           legend.values.rnd = 0, legend.nodata = "no data",
           legend.frame = FALSE, legend.border = "black",
           legend.horiz = FALSE, add = FALSE,
           col = carto.pal(pal1 = "pastel.pal", n1 = 8))
north(pos = "topright")

layoutLayer(title = "Taux de naissance en france par commune en %")

```



2 Exercice 2 -

2.1 Calculer et afficher les voronois associés aux stations velib de new-york.

```
datajson <- fromJSON(file = "data/input_NewYork.json")
id <- lapply(datajson$stationBeanList, function(st){as.data.frame(st)})
voronois = do.call(rbind,id)
head(voronois)[1:length(voronois)]
```

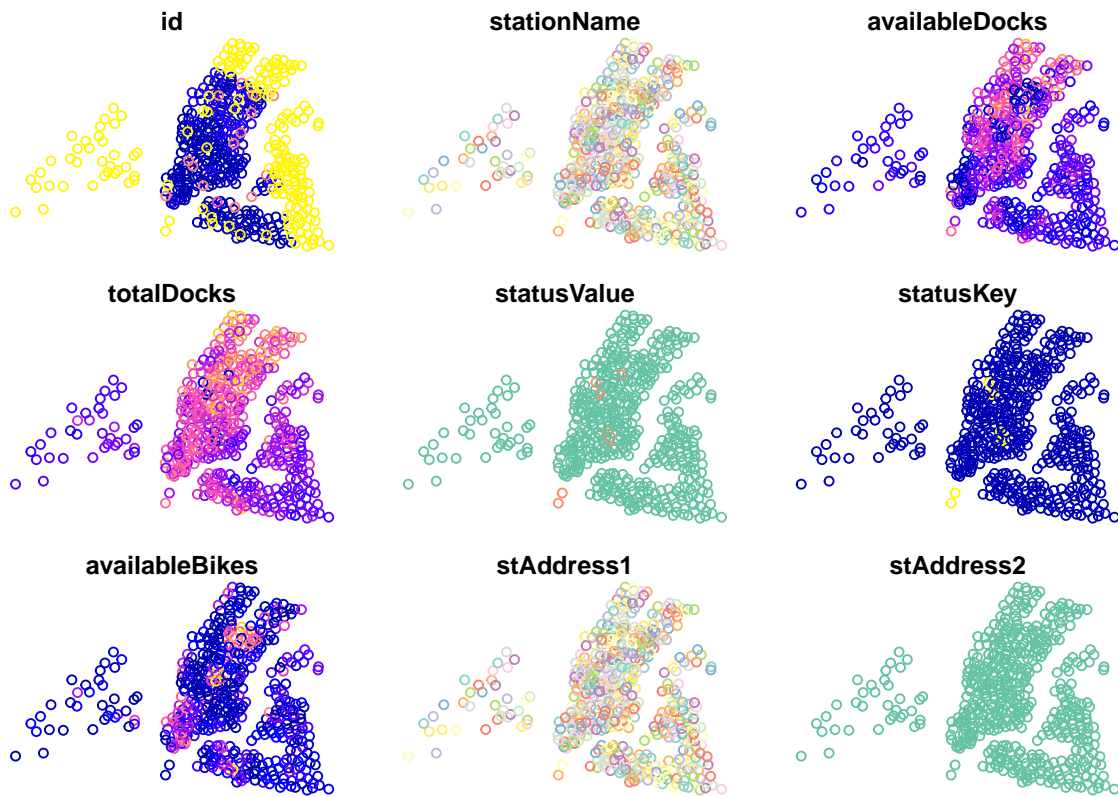
```
##      id                stationName availableDocks totalDocks latitude
## 1  72                W 52 St & 11 Ave             32         39 40.76727
## 2  79            Franklin St & W Broadway          26         33 40.71912
## 3  82              St James Pl & Pearl St          26         27 40.71117
## 4  83 Atlantic Ave & Fort Greene Pl              36         62 40.68383
## 5 116                W 17 St & 8 Ave             32         39 40.74178
## 6 119            Park Ave & St Edwards St          10         19 40.69609
##      longitude statusValue statusKey availableBikes          stAddress1
## 1 -73.99393 In Service      1             6            W 52 St & 11 Ave
## 2 -74.00667 In Service      1             7            Franklin St & W Broadway
## 3 -74.00017 In Service      1             1              St James Pl & Pearl St
## 4 -73.97632 In Service      1            25 Atlantic Ave & Fort Greene Pl
## 5 -74.00150 In Service      1             4                W 17 St & 8 Ave
## 6 -73.97803 In Service      1             8            Park Ave & St Edwards St
##      stAddress2 city postalCode location altitude testStation
## 1
## 2
## 3
## 4
## 5
## 6
##      lastCommunicationTime landMark
## 1 2016-05-25 09:24:59 AM
## 2 2016-05-25 09:26:47 AM
## 3 2016-05-25 09:25:50 AM
## 4 2016-05-25 09:24:56 AM
## 5 2016-05-25 09:25:29 AM
## 6 2016-05-25 09:26:23 AM
```

2.2 Utiliser pour cela la fonction st_as_sf avec l'option coords.

```
# ?st_as_sf
voronois = st_as_sf(voronois %>%
                    filter(longitude!=0,latitude!=0),coords = c("longitude","latitude"))

plot(voronois)
```

```
## Warning: plotting the first 9 out of 16 attributes; use max.plot = 16 to plot
## all
```



3 Exercice 3 -

3.1 Faire une carte interactive des monuments historiques à paris

```
monument = read_sf("data/monuments_paris.geojson")
contours = read_sf("data/iris75.geojson")
monuments_historique = read.csv("data/monuments-historiques.csv", sep=";", encoding = "UTF-8")

monuments = monument %>%
  left_join(monuments_historique, by=c("id"="Réf  rence"))

leaflet(monuments) %>%
  addTiles() %>%
  addCircleMarkers(popup = ~ Appellation.courante,
                    stroke=FALSE,
                    fillOpacity = 0.35,
                    radius = 3,
                    color = "darkred")
```

3.2 Faire une carte en symbol proportionel avec le nombre de monuments par iris

```
plot(st_geometry(contours), col = "ivory", border="ivory3", bg="azure")
propSymbolsLayer(monuments, var= "Appellation.courante")
```

4 Exercice 4 -

4.1 Faire une carte en symboles proportionnels du nombre de vélos dans les stations vélib à NewYork.

```
leaflet(data=voronois) %>%
  addTiles() %>%
  addCircleMarkers(radius = ~ sqrt(availableBikes),
    fillColor = "#449944", stroke=FALSE,
    fillOpacity = 1)
```

4.2 Utiliser un fond de carte open street map avec la library cartography

```
plot(st_geometry(voronois))
propSymbolsLayer(voronois, var = "availableBikes", legend.pos = "topleft")
north(pos = "topright", col = "darkblue")
layoutLayer(title = "Répartition des Velibs à New-York",
  col = "darkblue", coltitle = "white", tabtitle = TRUE,
  frame = TRUE, scale = NULL, north = FALSE)
```

