

# Présentation outils de dessins pour R

Jerome Guitton

23/02/2021

# Présentation arts plastiques et spatialisation (mais surtout arts plastiques)

Téléchargement des données : [donnees.zip](#)

A décompresser dans votre répertoire de travail

# Boite à outils

- Les librairies
  - dplyr, tidyr, ggplot2 (Well known one)
  - sf (Simple Features for R)
  - leaflet (pour se la pêter - importance de la forme sur le fond)
- Des données
  - un fichier csv avec des positions GPS de pêcheurs artisans en guinée
  - un shapefile des pays
  - un shapefile d'une grille régulière (rectangles statistiques)

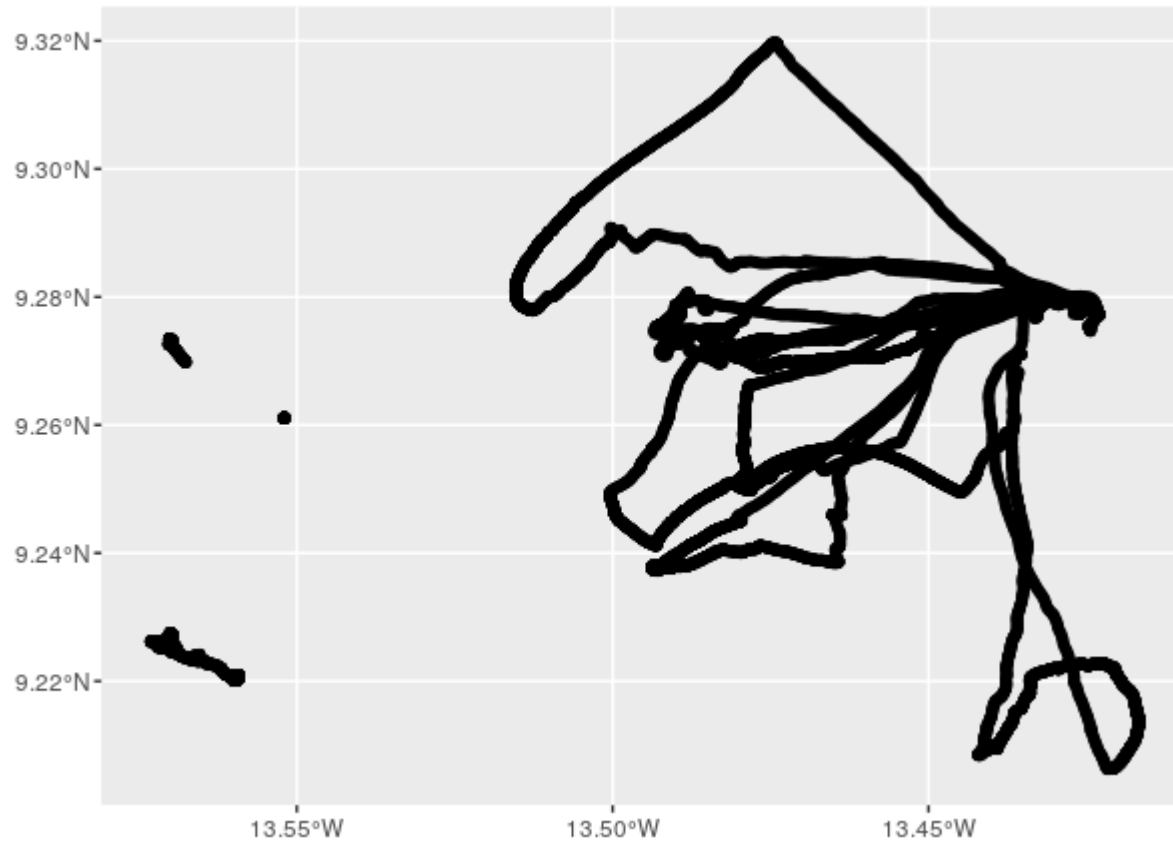
# Les données

Des points, le fichier points.csv

X.1	X	code_village	code_engin	code_pecheur	date_heure	longitude	latitude	coup_peche	site_peche	etat_detaille	etat_agrege	idpos	n
1	1	MTK	FMCy	4	2019-06-10 06:27:13	-13.42464	9.275370	NA	NA	NA	NA	2820034	
2	2	MTK	FMCy	4	2019-06-24 10:04:20	-13.42368	9.277128	NA	NA	NA	NA	2823301	
3	3	MTK	FMD	2	2019-07-20 09:06:13	-13.42375	9.277345	NA	NA	NA	NA	2831056	
4	4	MTK	FMCy	4	2019-06-10 06:41:18	-13.42761	9.279283	NA	NA	NA	NA	2820069	
5	5	MTK	FMCy	4	2019-06-10 06:41:34	-13.42784	9.279324	NA	NA	NA	NA	2820070	
6	6	MTK	FMCy	4	2019-06-10 06:41:48	-13.42812	9.279317	NA	NA	NA	NA	2820071	

# Les données

Des points, le dataframe devient un objet sf —> st\_as\_sf



# Les données

Des shapefiles : des données dans un contexte spatial

1 - world\_borders.shp (et dbf associés)

```
## Reading layer `world_borders' from data source `/home2/jerome/BAS_jerome/39/world_borders.shp'
## Simple feature collection with 3784 features and 5 fields
## geometry type:  POLYGON
## dimension:      XY
## bbox:           xmin: -180 ymin: -90 xmax: 180 ymax: 83.6236
## CRS:            EPSG:4326
```



# Pourquoi du sf

- un objet spatial est plus qu'un data frame
- ce plus, c'est qu'on y associe un système de coordonnées
  - les x y sont ils en mètres, en degrés
  - le x0,y0 est ce le 12 rue de la perrière, La chappelle-Launay 44260 ou le 0,0 de Null Island (cf wikipedia)

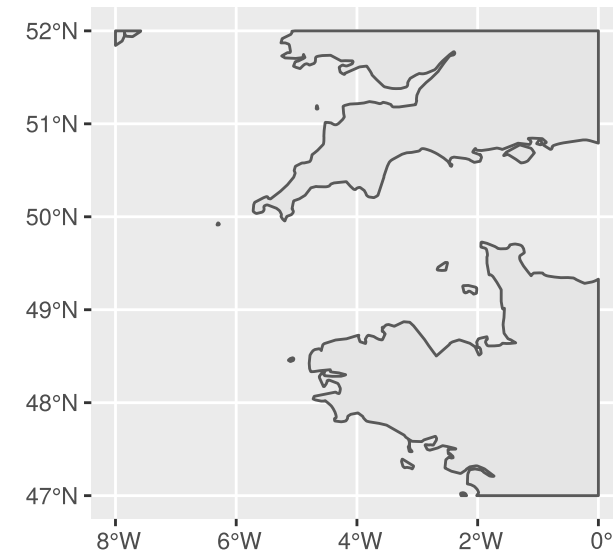
Cela permet de mettre ensemble des données spatiales décrites en degré depuis Bastia ou en mètre depuis Ajaccio

Le code CRS (n° de la projection) : <https://epsg.org/search/by-name>

# En fait un objet spatial qu'est ce que c'est

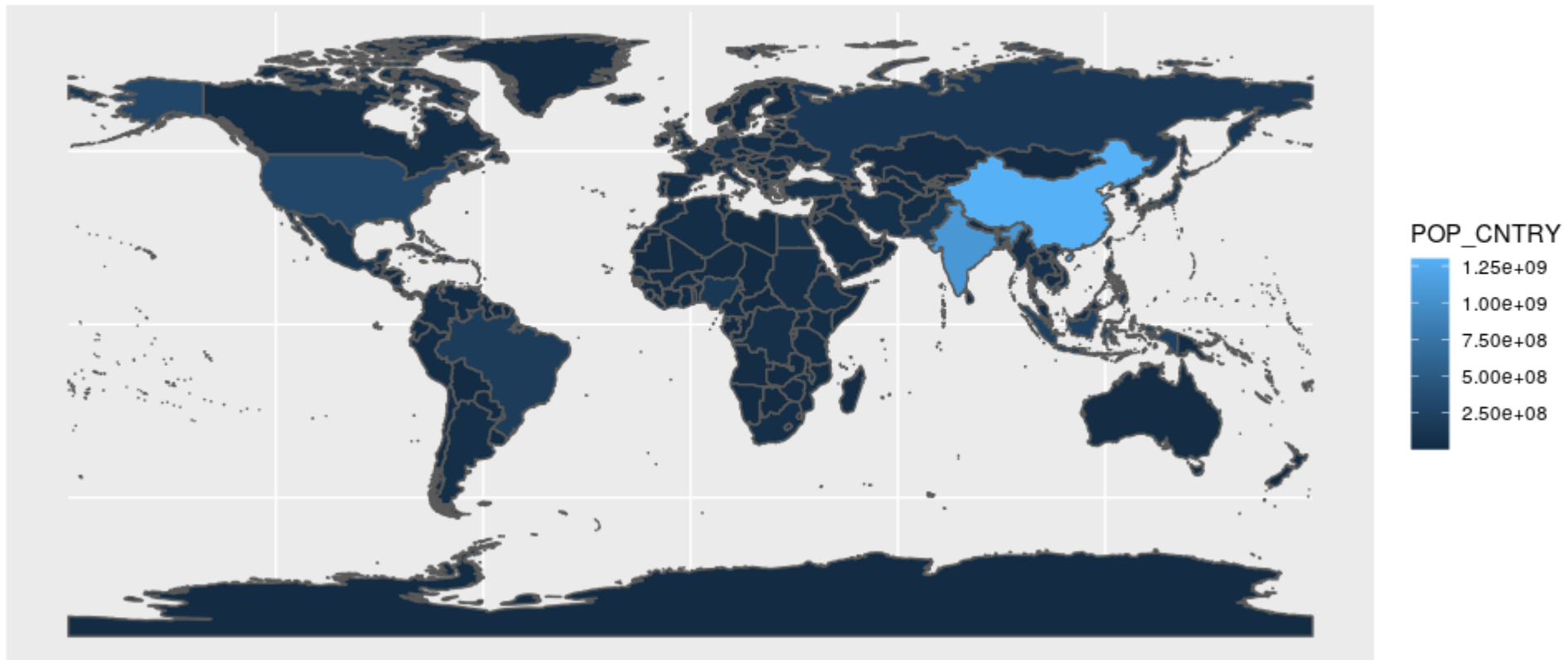
- Des données attributaires qui vont caractériser chaque éléments de mon dessin (contenues dans le .dbf associé)
- Un fichier de dessin (shapefile) dans un système de coordonnées

CAT	FIPS_CNTRY	CNTRY_NAME	AREA	POP_CNTRY	geometry
1	AA	Aruba	193	71218	POLYGON ((-69.88223 12.4111...
2	AC	Antigua and Barbuda	443	68320	POLYGON ((-61.73889 17.5405...
2	AC	Antigua and Barbuda	443	68320	POLYGON ((-61.73806 16.9897...
4	AG	Algeria	2381740	32129324	POLYGON ((2.96361 36.80222,...
5	AJ	Azerbaijan	86600	7868385	POLYGON ((48.58395 41.83577...
6	AL	Albania	28748	3544808	POLYGON ((19.43621 41.02107...





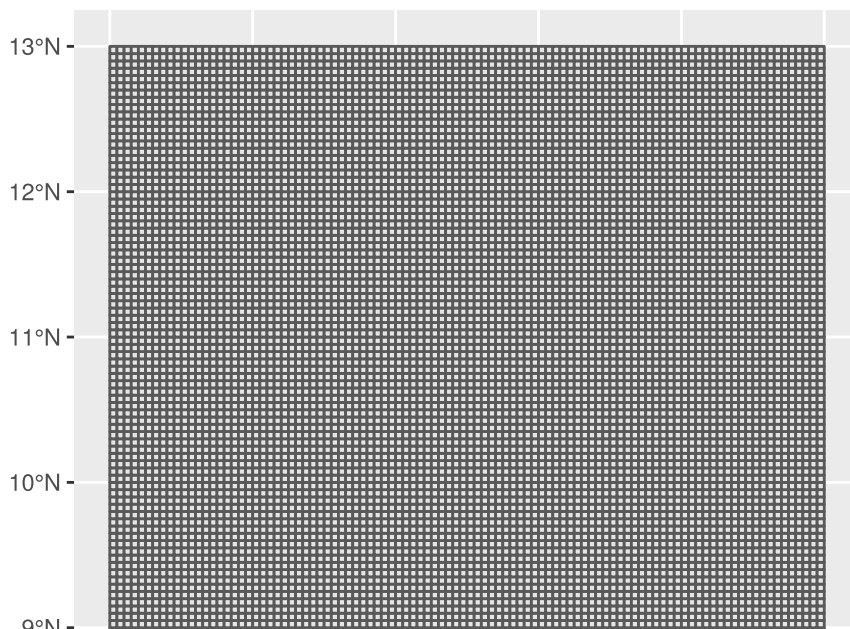
Des attributs une valeur associée à chaque  
portion de mon dessin → contenu dans le fichier  
.dbf



# Les données

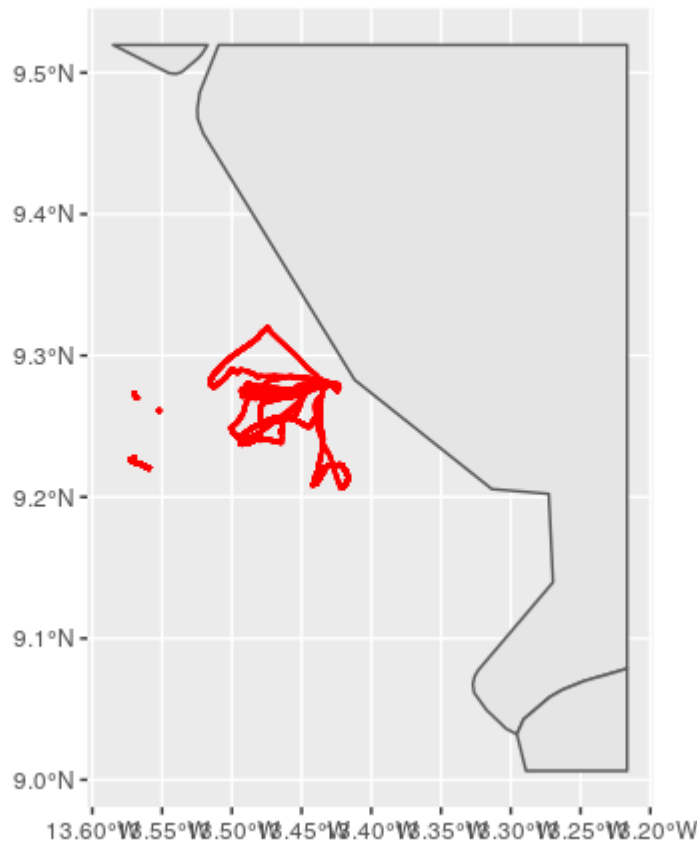
Des shapefile : des données dans un contexte spatial 2 - grille.shp (et dbf associés)

```
## Reading layer `grille' from data source `/home2/jerome/BAS_jerome/39/grille.shp' using driver  
## Simple feature collection with 10000 features and 1 field  
## geometry type:  POLYGON  
## dimension:      XY  
## bbox:           xmin: -18 ymin: 8 xmax: -13 ymax: 13  
## CRS:            EPSG:4326
```



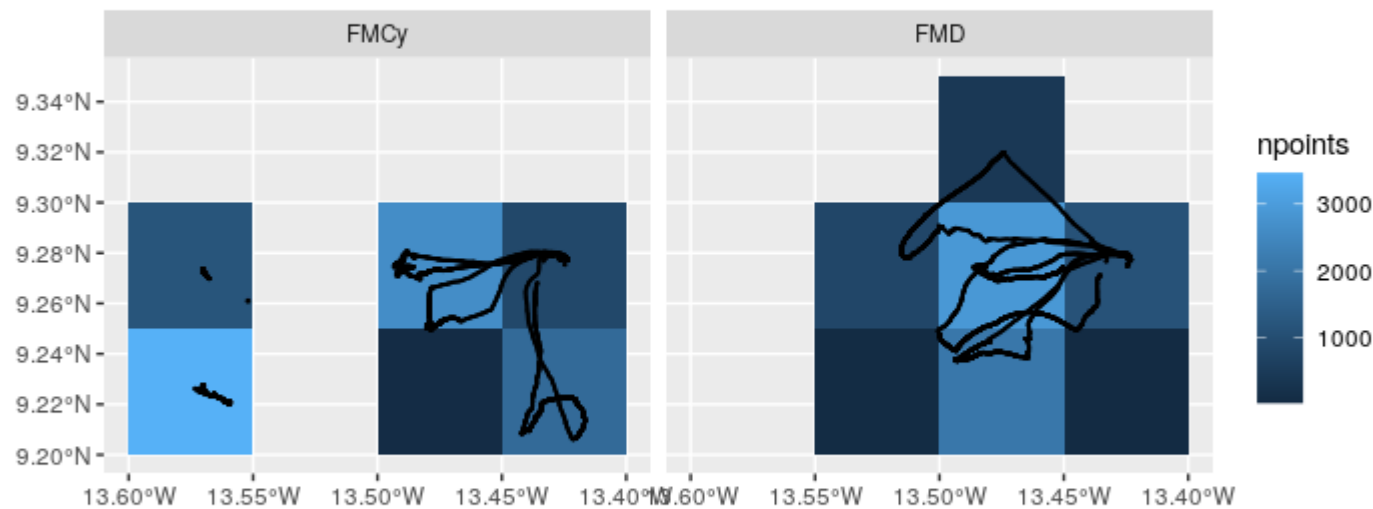
# Ce que l'on va faire 1

- Lire le fichier CSV (read.csv2)
- Transformer les points en objet sf (st\_as\_sf - système de coordonnées 4326)
- Lire le fichier world\_borders.shp (st\_read - système de coordonnées 4326)
- Faire une carte avec les points et un fond de carte - world\_borders (st\_crop + st\_buffer + geom\_sf)



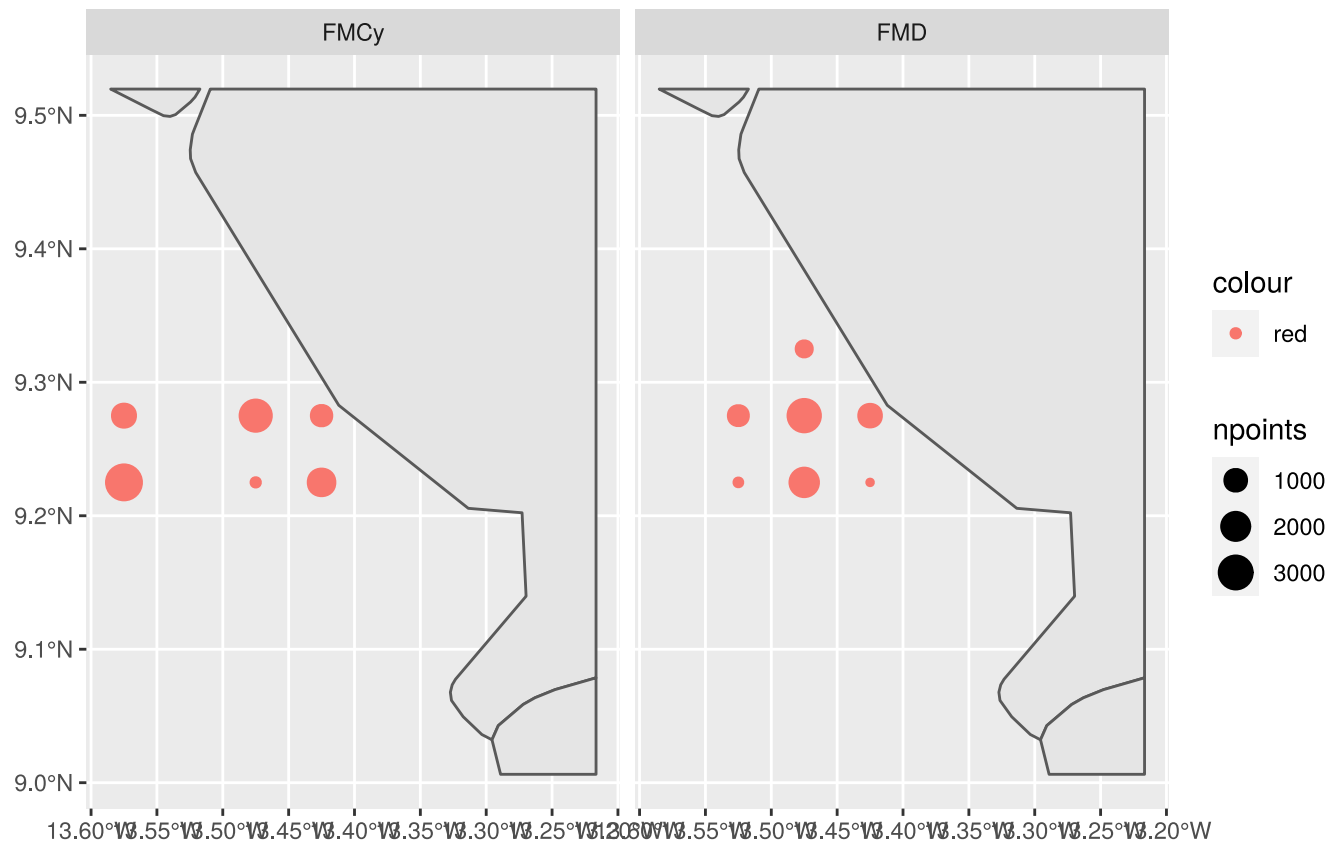
# Ce que l'on va faire 2

- Lire le fichier grille (shapefile grille.shp et dbf associé - st\_read)
- Compter le nombre de points dans chaque éléments de ma grille (st\_join)
- Faire une carte avec les nombre de points par carrés statistiques (geom\_sf)



# Ce que l'on va faire 3

Je veux des ronds proportionnels Mdame !! (st\_centroid + parametre size)



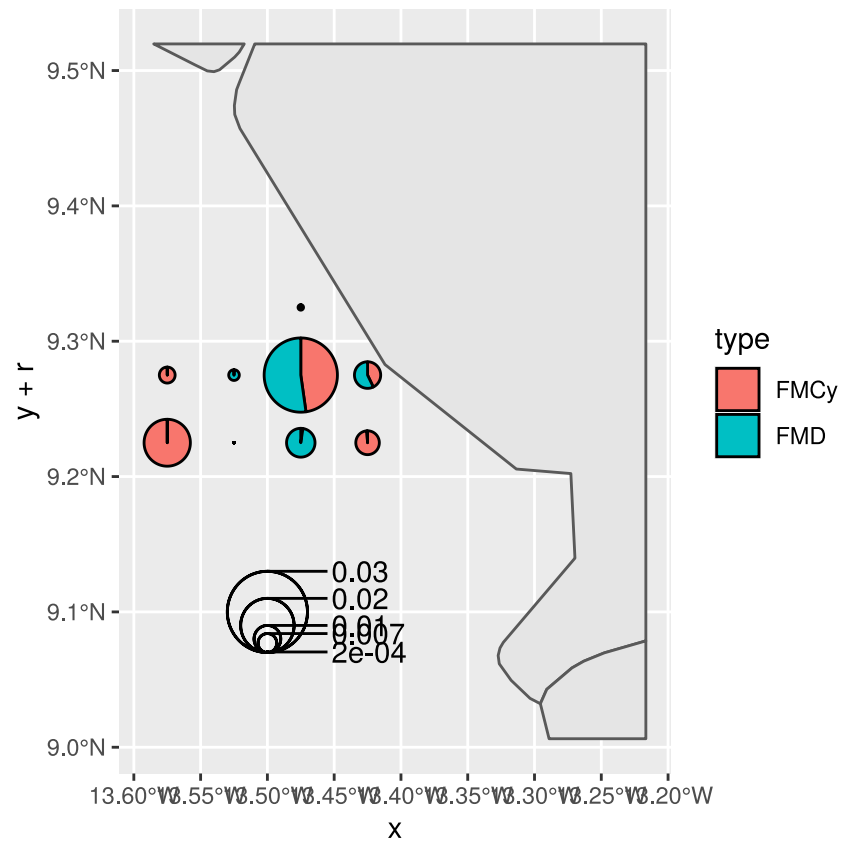
# Ce que l'on va faire 4

- Je veux des du St Nectaire maintenant !!
- pour un camembert, il faut un dataframe avec les latitude et longitude du centre de mon camembert (=au centre de ma grille -> st\_centroid) et pour obtenir st\_x
- il faut mettre les éléments de mon camembert en colonnes puisque je veux mettre les parts de chaque engins dans mon camembert (pivot\_wider)
- en mettant les engins en colonne, je me retrouve avec des NA pour certains engins qui ne sont pas présents dans les rectangles statistiques, je vais remplacer ces na par des 0 (replace\_na)
- Je vais créer une colonne avec le total des présences des 2 engins

id	longi	lati	FMCy	FMD	total
2489	-13.575	9.225	3465	0	3465
2491	-13.475	9.225	52	2078	2130
2492	-13.425	9.225	1742	19	1761
2589	-13.575	9.275	1181	0	1181
2591	-13.475	9.275	2618	2867	5485
2592	-13.425	9.275	841	1123	1964
2490	-13.525	9.225	0	40	40
2590	-13.525	9.275	0	793	793
2691	-13.475	9.325	0	409	409

# Ce que l'on va faire 4

- enfin je fais la carte (geom\_scatterpie)



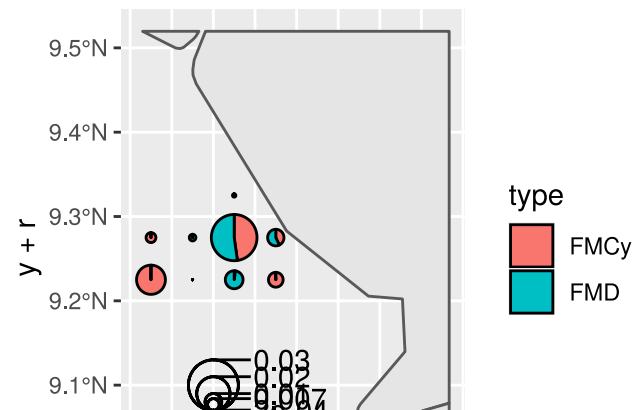
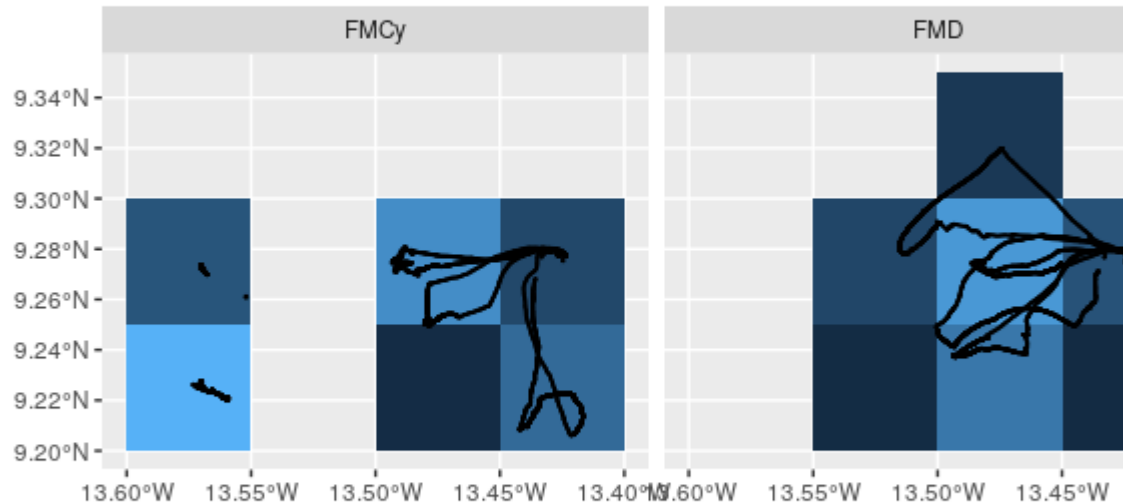
# Rappels

La liste des fonctions et packages

- **sf**

- st\_read : Lecture d'un fichier spatial (shapefile, postgres, mapinfo ....)
- st\_as\_sf : Convertir un objet (dataframe par exemple) en objet sf, objet spatial.
- st\_crop : Découper un objet spatial selon une emprise
- st\_centroid : Renvoyer le centre d'un polygone
- st\_join : Faire une jointure spatiale, sur la base d'une interaction de 2 objets dans l'espace

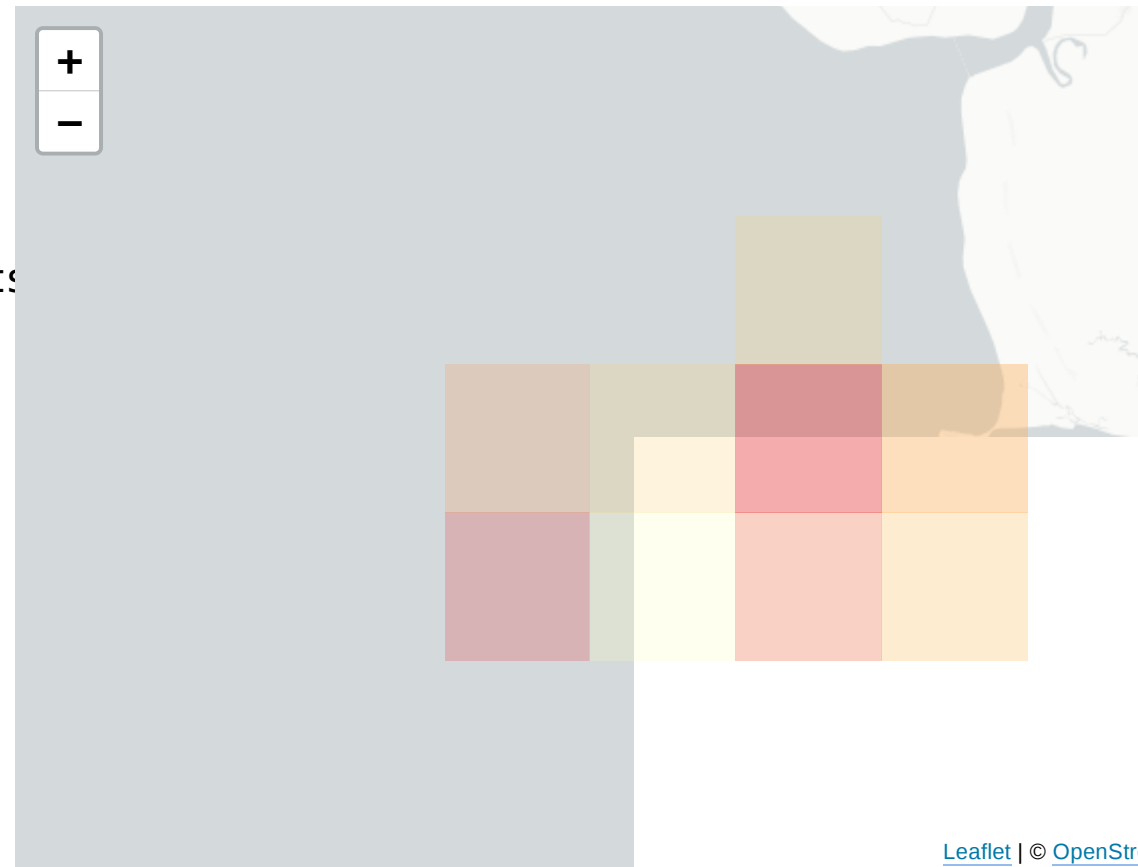
- **ggplot2**





# Extensions leaflet

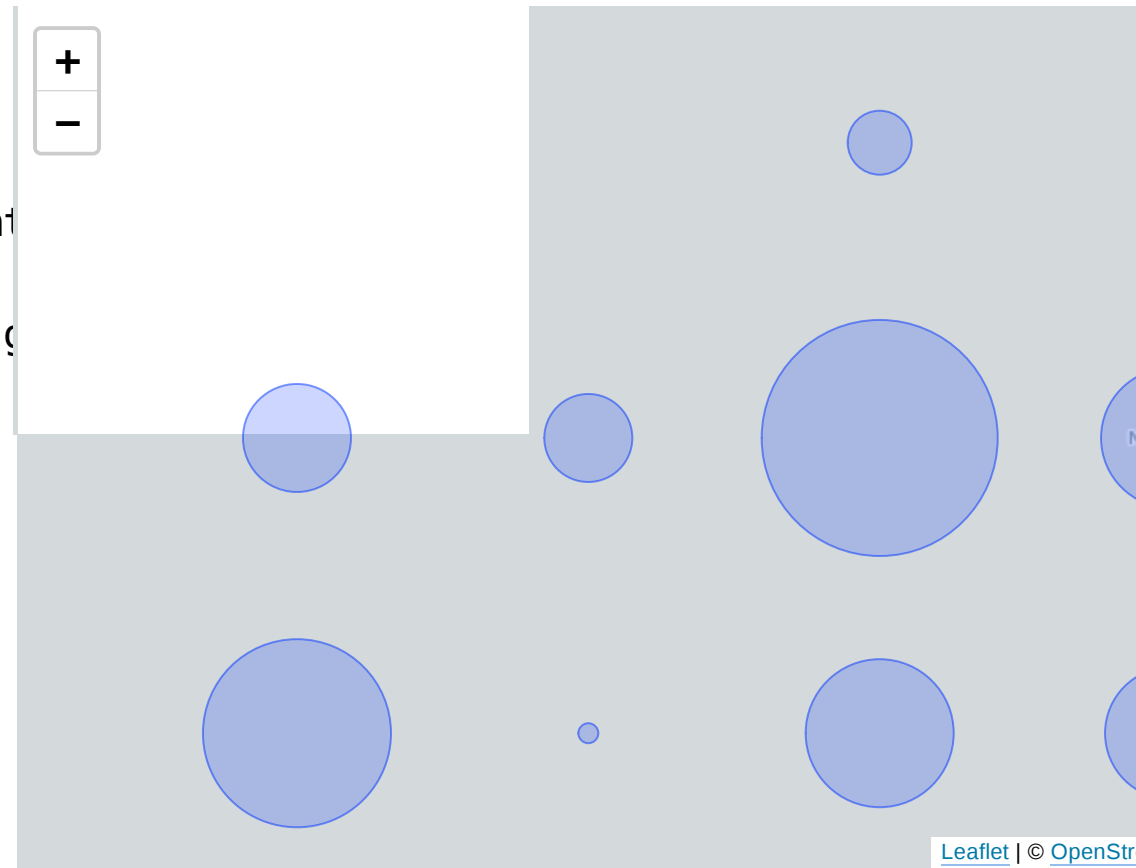
```
library(leaflet)
leaflet(grille_points) %>%
  addProviderTiles("CartoDB.Positron") %>%
  addPolygons(color = "black",
    fillColor = ~colorQuantile("YlOrRd", npoints
```



[Leaflet](#) | © [OpenStreetMap](#)

# Extensions leaflet

```
library(leaflet)
leaflet(grille_cam) %>%
  addProviderTiles("CartoDB.Positron") %>%
  addCircles(lng = ~longi, lat = ~lati, weight = 1,
    radius = ~sqrt(total) * 30,
    popup = paste('Le nombre de position est ', total))
```



# Extensions leaflet.extra2

```
library(leaflet.extra2)
library(geojsonsf)

gps$date_heure2<-as.POSIXct(gps$date_heure)

gps %>% filter(no_trajet==1175) -> gps.1175

leaflet(gps.1175) %>%
  addProviderTiles("CartoDB.Positron") %>%
  addCircleMarkers(radius = 0.1) %>%
  addTiles() %>%addTimeslider(data = gps.1175,
  options = timesliderOptions(position = "top",
  timeAttribute = "date_heure2",range = TRUE,
  setView(-13.5, 9.3, 11)
```

