

# Analyse des Heures de Pointe et des Quartiers Principaux

## Identification des Moments Forts de Trafic et des Zones Clés

Hermann Agossou; Salma Khmassi; Mohamed Lamine Bamba

2023-02-24

## Contents

<b>1</b>	<b>Data Loading</b>	<b>2</b>
1.1	Load packages . . . . .	2
1.2	R . . . . .	2
1.3	Including Plots . . . . .	2
1.4	Intersecter les points x et les quartiers y . . . . .	2
<b>2</b>	<b>OD Journalière : Analyse des Déplacements par Jour</b>	<b>3</b>
2.1	Load packages . . . . .	3
2.2	Load data and project spatial features . . . . .	3
2.3	Créer la section du 1er Mars . . . . .	4
2.4	Grouper par heure/ par identifiant chauffeur et par quartier . . . . .	4
2.5	Selectionner le quartier principal pour chaque chauffeur à chaque heure . . . . .	5
2.6	Pour chaque heure, calculer le nombre de chauffeurs . . . . .	5
2.7	Pour chaque heure, pour chaque quartier, calculer le nombre de chauffeurs . . . . .	5
2.8	Trouver les heures de pointes . . . . .	5
2.9	Résultats . . . . .	6
2.10	grouper par jour vs nuit . . . . .	6
2.11	Résultats . . . . .	6
2.12	Trouver les heures de pointes: jour vs nuit . . . . .	6
2.13	get neib centroids . . . . .	7
2.14	filter on the peak hour . . . . .	7
2.15	Identifier les zones de concentration à l'heure de pointe . . . . .	8
2.16	Résultats . . . . .	12

# 1 Data Loading

## 1.1 Load packages

```
library(dplyr)
library(sf)
library(leaflet)
library(lubridate)
library(mapsf)
library(tidyr)
```

## 1.2 R

```
heetchPoints <-readRDS("../data/heetchmarchcrop.Rds")
casaNeib <-st_read("../data/casaneib.geojson")
casaNeibProj <-st_transform(casaNeib, crs=26191)
heetchPointsProj <-st_transform(heetchPoints, crs=26191)
rm(heetchPoints)
rm(casaNeib)
#heetch_points_proj
head(heetchPointsProj) #(driver_id, location_at_local_time, geometry)
class(heetchPointsProj)
str(heetchPointsProj)

#heetch_points_proj
head(casaNeibProj) #(driver_id, location_at_local_time, geometry)
class(casaNeibProj)
str(casaNeibProj)
```

## 1.3 Including Plots

You can also embed plots, for example:

```
plot(pressure)
```

## 1.4 Intersecter les points x et les quartiers y

```
casaNeibUnion <-st_union(casaNeibProj)
selectPtsInCasa <-st_contains(x=casaNeibUnion, y=heetchPointsProj)%>%
  unlist()

heetchPointsProj <-heetchPointsProj[selectPtsInCasa,]
ptsInNeib <-st_within(x=heetchPointsProj, y=casaNeibProj) %>%
  unlist()
heetchPointsProj$NEIB <-ptsInNeib
rm(casaNeibUnion)
rm(selectPtsInCasa)
rm(ptsInNeib)
```

```
saveRDS(heetchPointsProj, file = "../data/heetchmarchcropwithneib.Rds")
head(heetchPointsProj)  #(driver_id, location_at_local_time, geometry)
str(heetchPointsProj)
class(heetchPointsProj)
```

## 2 OD Journalière : Analyse des Déplacements par Jour

### 2.1 Load packages

```
library(dplyr)
library(sf)
library(leaflet)
library(lubridate)
library(mapsf)
library(tidyr)
library(ggplot2)
```

### 2.2 Load data and project spatial features

- Load data and project spatial features
  - heetchmarchcropwithneib.Rds: les données de heetch avec les quartiers associés
  - casaneib.geojson: les quartiers de casa
  - casabound.geojson: les bords de casa

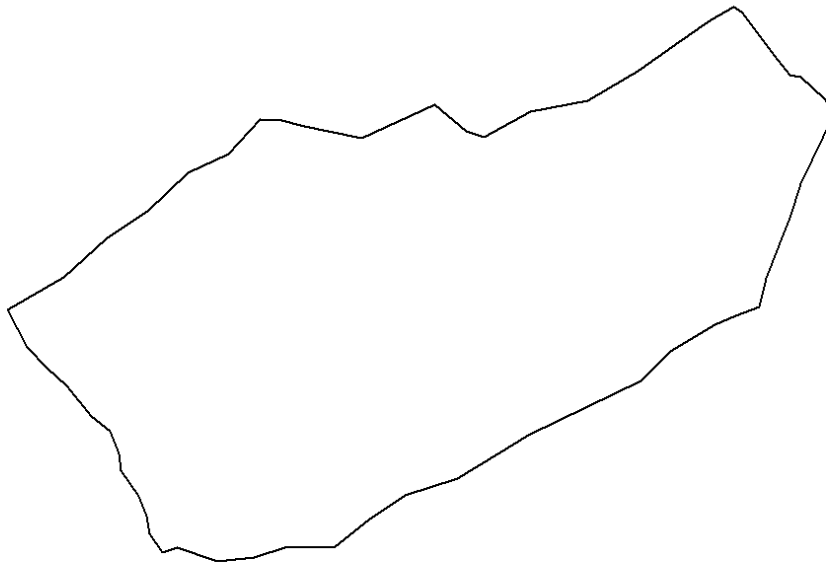
Please run the R file loadHeetch.Rmd to create the file heetchmarchcropwithneib.Rds

```
heetchPoints <-readRDS("../data/heetchmarchcropwithneib.Rds")  #Please run the R file [loadHeetch.Rmd]
casaNeib <-st_read("../data/casaneib.geojson")
casaBound <- st_read("../data/casabound.geojson")  #des points qui forment une
casaNeibProj <-st_transform(casaNeib, crs=26191)
heetchPointsProj <-st_transform(heetchPoints, crs=26191)
casaBoundProj <-st_transform(casaBound, crs=26191)
rm(heetchPoints)
rm(casaNeib)
 #heetch_points_proj
head(heetchPointsProj)  #(driver_id, location_at_local_time, geometry)
class(heetchPointsProj)
str(heetchPointsProj)

 #heetch_points_proj
head(casaNeibProj)  #(driver_id, location_at_local_time, geometry)
class(casaNeibProj)
str(casaNeibProj)

plot(casaBound$geometry)  #un plot de la géométrie
class(casaBound)
```

```
str(casaBound)
```



## 2.3 Créer la section du 1er Mars

Sectionner le fichier `heetchPointsProj` pour n'utiliser que les données de 1er mars

```
heetchPointsProj$Day <-day(heetchPointsProj$location_at_local_time)
heetchPointsProj$Hour <-hour(heetchPointsProj$location_at_local_time)
heetchM1Proj <-heetchPointsProj %>%
  filter(Day==1)
head(heetchM1Proj) #(driver_id, location_at_local_time, geometry, NEIB, Day, Hour)
str(heetchM1Proj)
nrow(heetchM1Proj) #166259
```

## 2.4 Grouper par heure/ par identifiant chauffeur et par quartier

On compte le nombre de tuple (heure, quartier, driver)

```
nestedPts <-heetchM1Proj %>%
  st_drop_geometry() %>%
  group_by(Hour, NEIB, driver_id) %>%
  summarize(NBPTS=n())
nrow(nestedPts) #5032
```

```
head(nestedPts) #Hour, driver_id, NEIB, NBPTS
class(nestedPts)
```

## 2.5 Sélectionner le quartier principal pour chaque chauffeur à chaque heure

Pour chaque tuple (driver, heure), on identifie le quartier où le driver est le plus représenté à cette heure là

```
mainNeib <-nestedPts %>%
  group_by(driver_id, Hour) %>%
  arrange(desc(NBPTS)) %>%
  slice(1)
nrow(mainNeib) #1374
class(mainNeib) #Hour, driver_id, NEIB, NBPTS
head(mainNeib)
```

## 2.6 Pour chaque heure, calculer le nombre de chauffeurs

```
NbDriverPerHour <-nestedPts %>%
  group_by(Hour) %>%
  summarize(NBDRiver=n()) %>%
  arrange(desc(NBDRiver))
nrow(NbDriverPerHour) #8
class(NbDriverPerHour) #Hour, NBDRiver
head(NbDriverPerHour)
```

## 2.7 Pour chaque heure, pour chaque quartier, calculer le nombre de chauffeurs

```
mainNeibN <-nestedPts %>%
  group_by(Hour, NEIB) %>%
  summarize(NBDRiver=n()) %>%
  arrange(desc(NBDRiver))
nrow(mainNeibN) #198
class(mainNeibN) #Hour, NBDRiver, NEIB
head(mainNeibN)
```

## 2.8 Trouver les heures de pointes

- On filtre les drivers les tuple (heure, quartier) pour lequel on a plus de 30 drivers.
- Puis, on compte à chaque heure, le nombre de quartiers qu'on trouve

```
PeakHours <-mainNeibN %>%
  filter(NBDRiver>30) %>%
  select(Hour, NEIB) %>%
  group_by(Hour) %>%
```

```

  summarize(NBNEIB=n()) %>%
  arrange(desc(NBNEIB))
nrow(PeakHours) #198
class(PeakHours) #Hour, NBNEIB
head(PeakHours)

```

## 2.9 Résultats

On remarque 22h est la principale heure de pointe devant (21h, 20h, 23h), ce qui montre que les flux sont assez concentrées en soirées. Ce qui suggère, vue la classe sociale des utilisateurs de Heetch, que les chauffeurs sont appelés en soirées. On identifiera les flux dans la partie 2

## 2.10 grouper par jour vs nuit

Ajouter à mainNeibNGrouped, une colonne GRP = fct(Hour)

- 4 pour l'heure entre 04h et 12h,
- 12 pour l'heure entre 12h et 20h,
- 20 pour l'heure entre 20h et 04h

```

mainNeibNGrouped <-data.frame(mainNeibN)
mainNeibNGrouped['GRP'] <-100
mainNeibNGrouped['GRP'][mainNeibNGrouped['Hour'] >=20 | mainNeibNGrouped['Hour'] < 4] <- 20
mainNeibNGrouped['GRP'][mainNeibNGrouped['Hour'] >=4 & mainNeibNGrouped['Hour'] < 12] <- 12
mainNeibNGrouped['GRP'][mainNeibNGrouped['Hour'] >=12 & mainNeibNGrouped['Hour'] < 20] <- 20
nrow(mainNeibNGrouped) #606
class(mainNeibNGrouped) #Hour, NBDRiver, NEIB, GRP
head(mainNeibNGrouped)

```

```

mainNeibNGrouped <-mainNeibNGrouped %>%
  group_by(GRP, NEIB) %>%
  summarize(NBDRiver=sum(NBDRiver)) %>%
  arrange(desc(NBDRiver))
nrow(mainNeibNGrouped) #82
class(mainNeibNGrouped) #NBDRiver, NEIB, GRP
head(mainNeibNGrouped)

```

## 2.11 Résultats

On remarque également qu'il y a beaucoup plus de concentration en soirée qu'en matinée

## 2.12 Trouver les heures de pointes: jour vs nuit

```

PeakHoursGrouped <-mainNeibNGrouped %>%
  filter(NBDRiver>240) %>%
  select(GRP, NEIB) %>%

```

```

group_by(GRP) %>%
  summarize(NBNEIB=n()) %>%
  arrange(desc(NBNEIB))
nrow(PeakHoursGrouped) #198
class(PeakHoursGrouped) #GRP,NBNEIB
head(PeakHoursGrouped)

```

## 2.13 get neib centroids

Identifier le centroïde de chaque quartier

```

polCentroids <- st_centroid(casaNeibProj) # pour avoir les centroides
coordCentroids <-st_coordinates((polCentroids)) %>%as_data_frame()
# pour avoir les coordonnées des centroides
nrow(coordCentroids) #29
class(coordCentroids) #X,Y
head(coordCentroids)

```

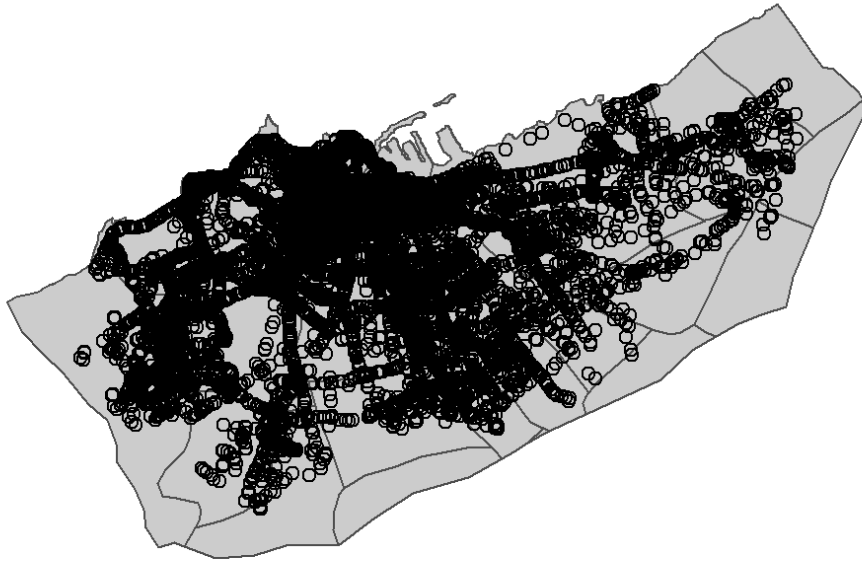
## 2.14 filter on the peak hour

Faire une section des données de heetch pour filter uniquement sur l'heure de pointe (22h)

```

heetchM1ProjAtPeakHour<-heetchM1Proj %>%
  filter(Hour==22)
head(heetchM1ProjAtPeakHour) #(driver_id, location_at_local_time, geometry, NEIB, Day, Ho
str(heetchM1ProjAtPeakHour)
nrow(heetchM1ProjAtPeakHour) #166259
plot(casaNeibProj$geometry, col="grey80", border ="grey30")
plot(heetchM1ProjAtPeakHour$geometry, add=TRUE)

```



## 2.15 Identifier les zones de concentration à l'heure de pointe

Faire une affichage qui montre le nombre de points par quartiers à l'heure de pointe

```
my_fct_ggplot <- function(grid_geometry ){
  # create sample data
  data <- data.frame(
    Name = polCentroids$NAME_4,
    #Value = c(10, 20, 30),
    geometry = grid_geometry
  )

  # convert to sf object
  sf_data <- st_as_sf(data)

  # create the map
  ggplot(sf_data) + geom_sf_text(aes(label = Name), size = 3, color = "red", fontface = "bold")
}

my_fct2 <- function(my_points_to_plot, grid_geometry ){
  # count the nb the point per cell
```



```

# method 1: est-ce que les cellules(carreaux) contiennent les points
#my_points_to_plot <- heetchM1ProjAtPeakHour#polCentroids
inter_grid = st_contains(x = grid_geometry, y=my_points_to_plot) #la grille et les points
inter_grid[72] #le carreau 72 est vide
inter_grid[73] #il va afficher les indexations des points: 882

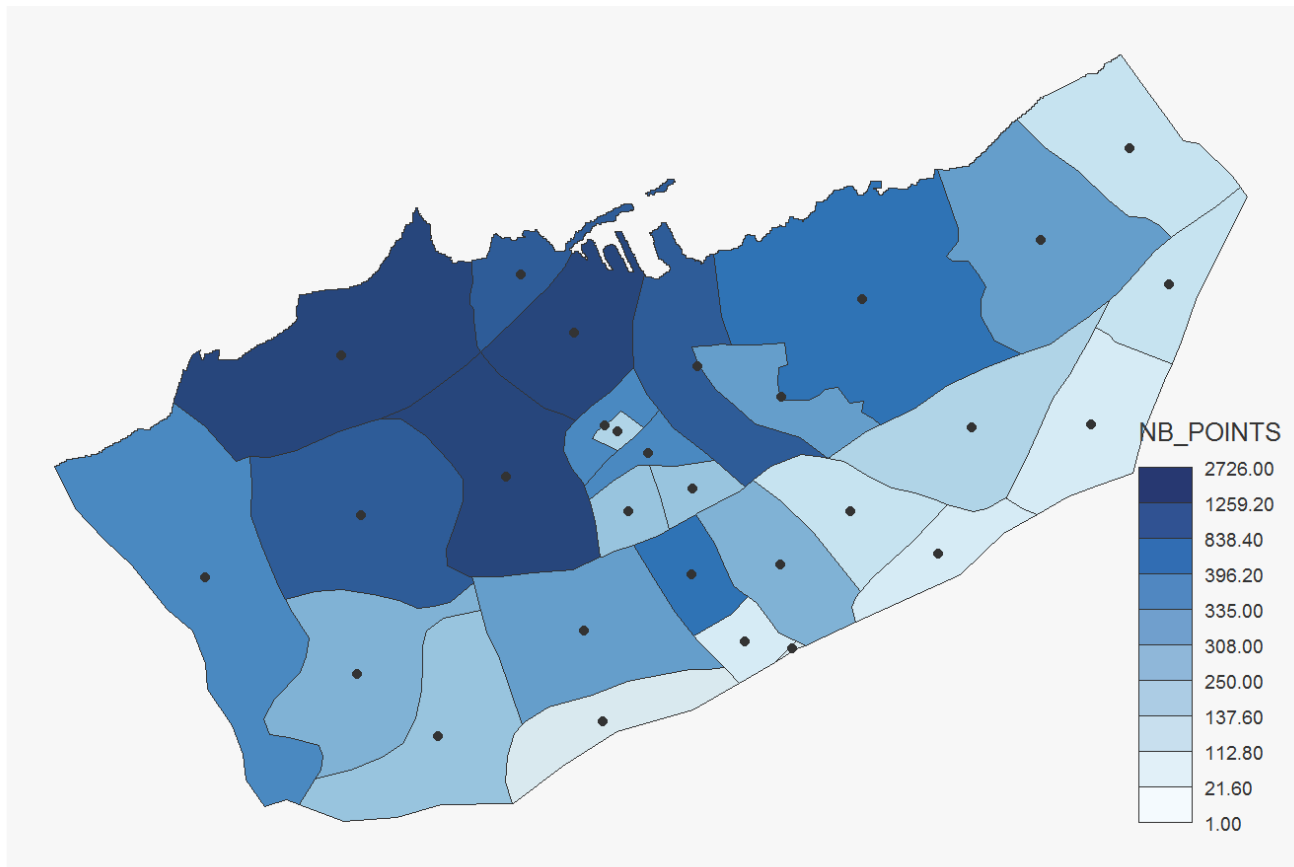
# create dataframe (cell_id, nb_of_points, geometry_of_the_cell)
list_lengths_per_cell = sapply(X = inter_grid, FUN = length)
casagrid_points = st_sf(ID_GRID = 1:length(list_lengths_per_cell),
                        NB_POINTS = list_lengths_per_cell,
                        geometry = grid_geometry)
class(casagrid_points) #un obj sf et dataf --> cartographiable

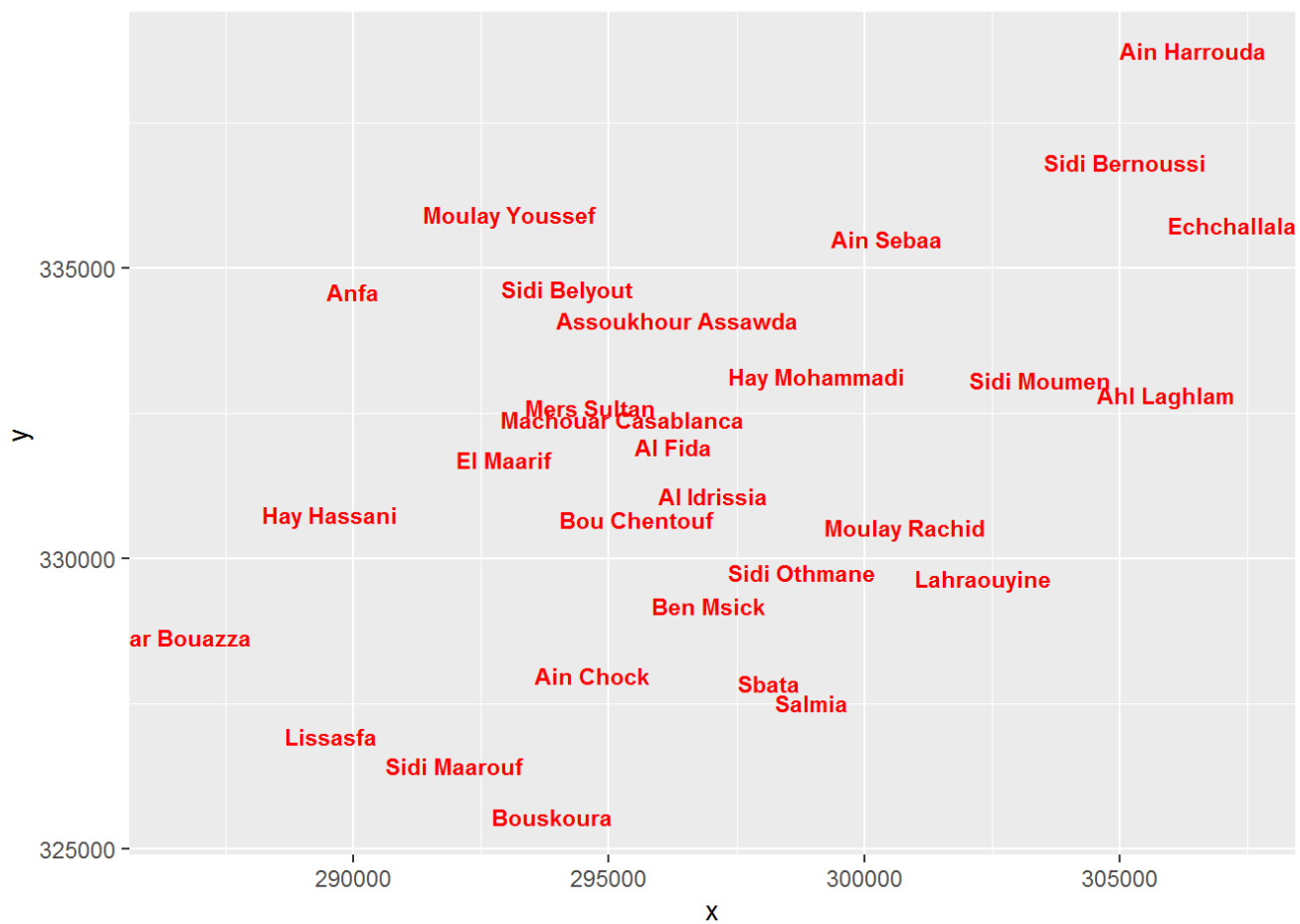
# create a carto ---
# beaucoup de carreaux en mers (avec nb_points=0) et ça étire la distribution
casagrid_points_crop = casagrid_points %>% filter(NB_POINTS > 0)

#mf_map(x = casaBoundProj,type = "base")
mf_map(x=casagrid_points_crop,
      var = "NB_POINTS",
      type = "choro",
      # 10 classes d'effectifs égaux pour éviter les pb liés aux outiliers
      breaks = "quantile",
      nbreaks = 10,
      pal = rev(hcl.colors(n=10, palette = "Blues")), #un rev pour l'ordre inverse
)
mf_map(x=casaNeibProj, add=TRUE, col="#2596be25", alpha=0.5) #neib
mf_map(x=polCentroids, add = TRUE)
my_fct_ggplot(grid_geometry)
}

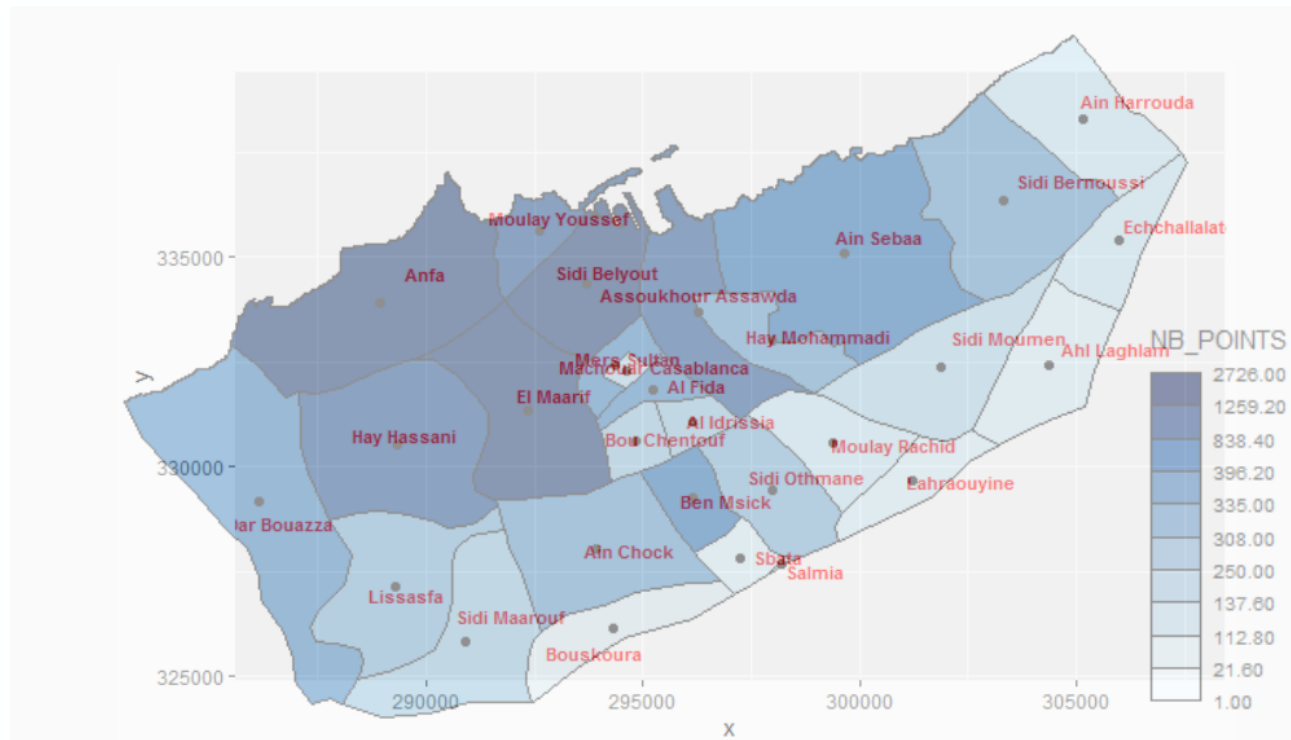
my_fct2(heetchM1ProjAtPeakHour, casaNeibProj$geometry)

```





## 2.16 Résultats



On remarque que les villes Anfa, El Maarif et Sidi Belyout contiennent les plus grandes concentrations à 22h