projet_R_ENSAE_2023

GUEYE_Aissata

2023-07-24

Contents

L'objectif de ce projet est que nous appliquions les outils que nous avons étudiés dans le cours du logiciel statistique R, dans le cas d'une étude de cas réelle. Sur une enquête vise à identifier et à caractériser des bioénergies durables pour les petites et moyennes entreprises (PME) agroalimentaires d'Afrique de l'Ouest.

Installation des packages nécessaire

```
#install.packages("gtsummary")
#install.packages("kableExtra")#rmar

library(readxl)
library(dplyr)
library(tidyverse)
library(rmarkdown)
library(stringr)
library(kableExtra)
library(gtsummary)
library(tinytex)
library("haven")
```

1- importation et mise en forme

1-1-1 Importer la base de données dans un objet de type data.frame nommé projet

```
#Lire le fichier Excel et stocker son contenu dans un data frame nommé "projet"

projet<-data.frame(read_excel("Base_Partie 1.xlsx"))
```

#1-1-2 Selectionner les variales mentionnées dans la section description

```
section_description <- projet %>% dplyr::select(-key)##Dans cette expression, nous utilisons l'opérateu
```

1-1-3 Un tableau qui résume les valeurs manquantes par variable

```
# Calculer le nombre de valeurs manquantes par variable
 valeurs_manquantes<- colSums(is.na(projet))

# Calculer la proportion des valeurs manquantes par variable

somme<-sum(valeurs_manquantes)# la somme des valeurs manquante dans la base
proportions <- ((valeurs_manquantes/somme)*100)

# Créer le tableau résumé des valeurs manquantes et l'afficher avec kable()

tableau_resume <- data.frame(valeurs_manquantes, proportions)

kable(tableau_resume, caption = "Tableau résumé des valeurs manquantes")
```

```
#kable(tableau_resume)
```

1-1-4 Vérifier s'il y a des valeurs manquantes pour la variable key dans la base projet. Si oui, identifier la (ou les) PME concernée(s).

D'après le tableau précédent la variable key n'a aucune valeur manquante. On peut aussi le voir à partir du code suivant:

Table 1: Tableau résumé des valeurs manquantes

	valeurs_manquantes	proportions
key	0	0.0000000
q1	0	0.0000000
q2	0	0.0000000
q23	0	0.0000000
q24	0	0.0000000
q24a_1	0	0.0000000
q24a_2	0	0.0000000
q24a_3	0	0.0000000
q24a_4	0	0.0000000
q24a_5	0	0.0000000
q24a_6	0	0.0000000
q24a_7	0	0.0000000
q24a_9	0	0.0000000
q24a_10	0	0.0000000
q25	0	0.0000000
q26	0	0.0000000
q12	0	0.0000000
q14b	1	0.3952569
q16	1	0.3952569
q17	131	51.7786561
q19	120	47.4308300
q20	0	0.0000000
filiere_1	0	0.0000000
filiere_2	0	0.0000000
filiere_3	0	0.0000000
filiere_4	0	0.0000000
q8	0	0.0000000
q81	0	0.0000000
gps_menlatitude	0	0.0000000
gps_menlongitude	0	0.0000000
submissiondate	0	0.0000000
start	0	0.0000000
today	0	0.0000000

```r
attach(projet)# fixer la base pour l'utilisé avec la fonction suivante

# les indices de ligne où la colonne "key" a des valeurs manquantes (NA)
which(is.na(key),arr.ind=TRUE)
```

```
## integer(0)
```

1-2 Création de variables

#1-2-1 Rénommer les variable q1,q2 et q3 respectivement en region,en departement et en sexe

```r
projet<-rename(projet,region=q1)##la fonction rename est une fonction de base R permet de renommer le n
projet<-rename(projet,departement=q2)
projet<-rename(projet,sexe=q23)
print(head(projet[2:4]))
```

```
##       region departement  sexe
## 1   Diourbel      Bambey Femme
## 2      Thiès       Mbour Femme
## 3      Thiès       Mbour Femme
## 4      Thiès       Mbour Femme
## 5 Ziguinchor     Bignona Homme
## 6 Ziguinchor    Oussouye Femme
```

#1-2-2 Créer la variable sexe_2 qui vaut 1 si sexe égale à Femme et 0 sinon.

```r
projet<-projet %>% dplyr::mutate(sexe_2=ifelse(sexe=="Femme","1","0"))#mutate()de dplyr pour créer une
```

#1-2-3 Créer un data.frame nommé langues qui prend les variables key et les variables correspondantes décrites plus haut.

```r
# création d'un data.frame avec comme variable "key" et les variables "q24a" par la fonction starts_wit

langues<-data.frame(projet%>%dplyr::select("key",starts_with("q24a")))
```

#1-2-3 Créer la variable parle qui est égale au nombre de langue parlée par le dirigeant de la PME.

```r
#Ajouter une nouvelle variable "parle" à "langues" avec la somme des colonnes par ligne commençant par

langues<-langues %>% dplyr::mutate(parle=rowSums(select(.,starts_with("q24a"))))
```

#1-2-4 Sélection uniquement des variables key et parle, l'objet de retour sera langues.

```r
langues<-langues%>%dplyr::select(key,parle)
```

- Merger les data.frame projet et langues:

```r
# nous utilisons la fonction merge pour associer les deux data frame avec clé la variable "key"
New_projet<-merge(projet,langues,by="key")
```

2 analyse descriptive

Nous allons nous proposer de faire une analyse descriptive univariée et bivariée

Création du tableau pour les variables univariées

```r
#Répartition des PME suivant le sexe, le niveau d'instruction, le statut juridique et le propriétaire l
tbl1 <- projet %>% tbl_summary(include = c("sexe", "q25", "q12", "q81"))
tbl1
```

Characteristic	**N = 250**
sexe	
Femme	191 (76%)
Homme	59 (24%)
q25	
Aucun niveau	79 (32%)
Niveau primaire	56 (22%)
Niveau secondaire	74 (30%)
Niveau Superieur	41 (16%)
q12	
Association	6 (2.4%)
GIE	179 (72%)
Informel	38 (15%)
SA	7 (2.8%)
SARL	13 (5.2%)
SUARL	7 (2.8%)
q81	
Locataire	24 (9.6%)
Propriétaire	226 (90%)

```
# Répartition des PME suivant le statut juridique et le sexe,   le niveau d'instruction et le sexe,
tbl2 <- projet %>% tbl_summary(
  include = c("q25", "q12", "q81"),
  by ="sexe", label=list(q12~ "Statut juridique",
                         q25~ "Niveau d'instruction",
                         q81~ "Propriétaire/locataire")) %>%
add_overall()
```

3.Cardiographie

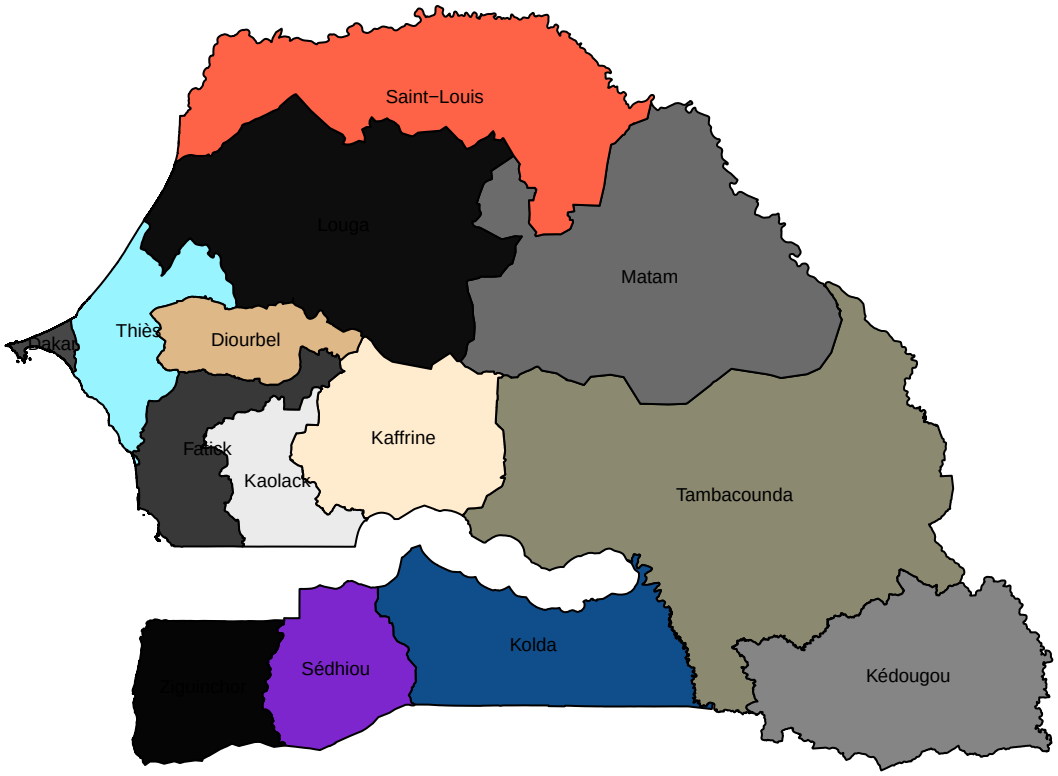
Importation des packages nécessaire afin de réalise notre travail sur la cartographie

```
#install.packages(c("cartography", # réaliser des cartes
#"classInt", # discrétisation de variables quantitatives
#"ggspatial", # syntaxe complémentaires à la ggplot
#"GISTools", # outils pour faire de la carto
#"leaflet", # interractivité avec JavaScript
#"maptools", # manipulation de données "spatial",
#"OpenStreetMap", # OSM
#"osrm", # openstreetmap avec R
#"popcircle", # représentation style bubble plot
#"raster", # manipulation de données raster
#"RColorBrewer", # palette de couleurs pour carto
#"rgdal", # import de données spatiales
#"rgeos", # manipulation de données spatiales
#"sf", # nouvelle classe d'objets spatials
#"sp", # ancienne classe d'objets spatials
#"tidyverse", # ggplot, dplyr, etc
#"tmaptools" # pour la carto

#),

#install.packages("OpenStreetMap")
library(sf)
```

```r
library(sp)
library(cartography)
library(rgdal)
library(maptools)
library(osrm)
library(spdep)
library(raster)
library(rgeos)
library(RColorBrewer)
library(classInt)
#library(popcircle)
library(ggspatial)
#library(GISTools)
library(leaflet)
#library(OpenStreetMap)
library(tidyverse)
library(igraph)
library(tmap)
library(plotly)
```

#3-1 Transformation du data.frame en données géographiques dont l'objet sera nommé projet_map.

```r
# récupère les données administratives de niveau 1 pour le senegal
Senegal <- raster::getData("GADM", country = "Senegal", level = 1)
centre<-coordinates(Senegal)
noms1<-Senegal$NAME_1

# Créer une palette de couleurs aléatoires

couleurs<- sample(colors(), length(noms1))

w.nb1 <- poly2nb(Senegal,row.names = noms1,queen=TRUE)
par(oma = c(0, 0, 0, 0), mar = c(0, 0, 1, 0))

# Afficher la carte du Sénégal avec des couleurs aléatoires pour chaque région

plot(Senegal, col = couleurs)
text(centre[, 1], centre[, 2], noms1, cex = 0.55, col = "black")

title("Carte du Sénégal")
layoutLayer( sources = "Source:Calculs de l'auteur", frame = TRUE,
col = "NA", scale = NULL)
```



Source:Calculs de l'auteur

Pour transformé le data.frame en données géographique on utilise la fonction st_as_sf du package "sf"

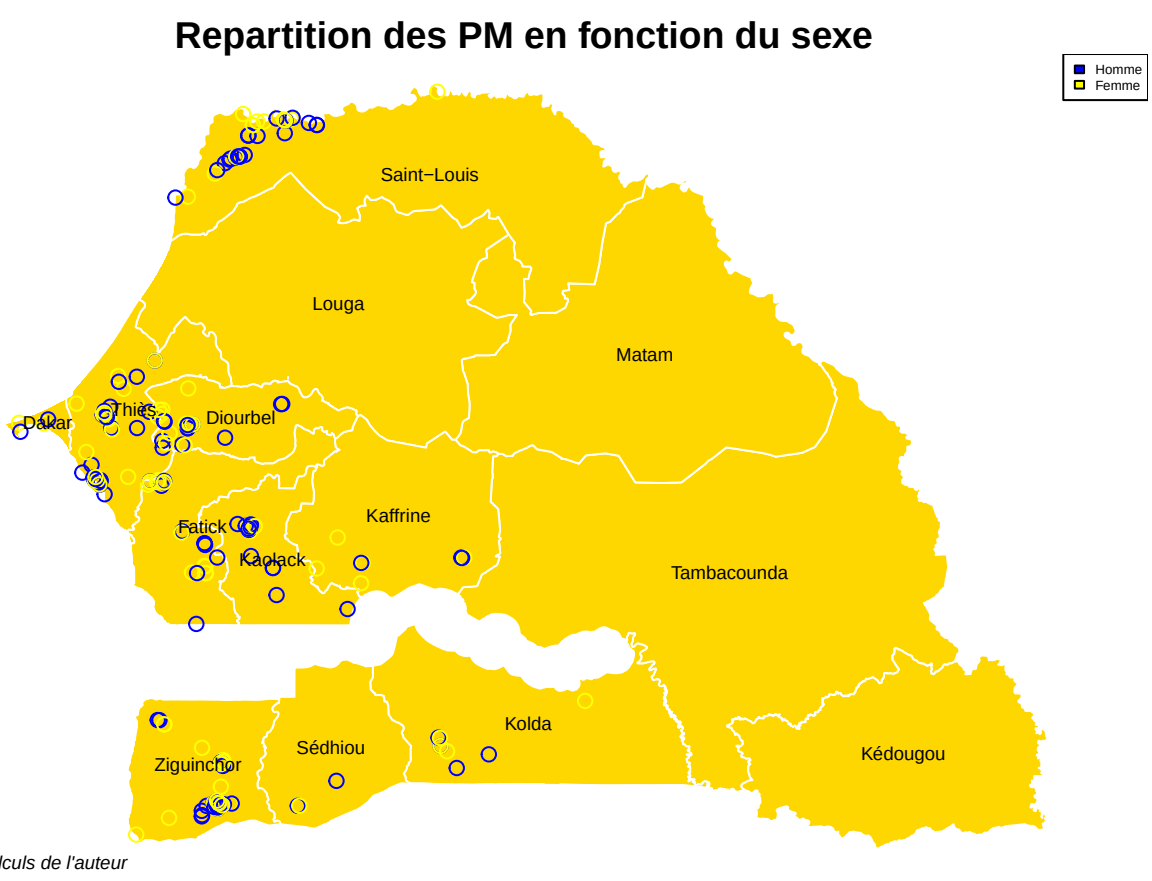
```
#Convertir l'objet "projet" en un objet spatial "sf" avec les coordonnées GPS spécifiées

projet_map<- st_as_sf(projet, coords = c("gps_menlongitude", "gps_menlatitude"))
class(projet_map)
```

```
## [1] "sf"         "data.frame"
```

#3-2 Faites une réprésentation spatiale des PME suivant le sexe

```
# Obtenir les données de la carte du Sénégal

Senegal <- raster::getData("GADM", country = "Senegal", level = 1)

# Obtenir les coordonnées des centres des régions

centre<-coordinates(Senegal)

# Obtenir les noms des régions

noms1<-Senegal$NAME_1
w.nb1 <- poly2nb(Senegal,row.names = noms1,queen=TRUE)
par(oma = c(0, 0, 0, 0), mar = c(0, 0, 1, 0))
# Créer une carte de base du Sénégal avec une seule couleur de base
carte<-plot(Senegal,col = "gold", border = "white")

# Ajouter les points pour indiquer la répartition des PME en fonction du sexe
```

```
points(projet_map$Longitude, projet_map$Latitude, col = c("blue", "yellow")[projet_map$sexe], pch = 19)

plot(projet_map["sexe"],col=c("blue","yellow"),add=TRUE)

ma_palette <- terrain.colors(length(unique(projet_map$variable_couleur)))

attach(projet_map)
legend("topright",
legend = c("Homme","Femme"),
cex = 0.4,
fill = c("blue","yellow"))
text(centre[,1],centre[,2],noms1,cex=.55)
title("Repartition des PM en fonction du sexe")
layoutLayer( sources = "Source:Calculs de l'auteur", frame = TRUE,
col = "NA", scale = NULL)
```



Source:Calculs de l'auteur

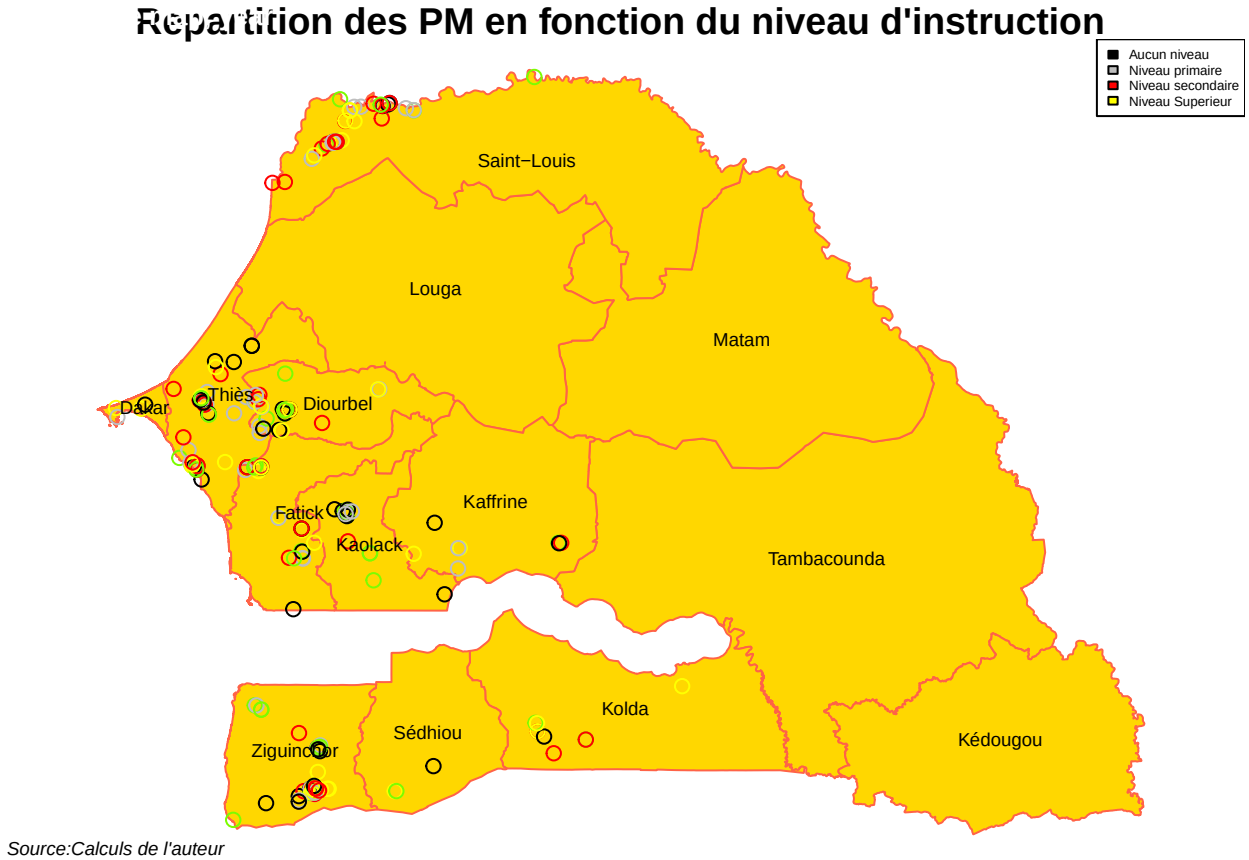
#3-3 Faites une réprésentation spatiale des PME suivant le niveau d'instruction

```
unique(q12)
```

```
## [1] "GIE"         "Informel"    "SUARL"       "SARL"        "Association"
## [6] "SA"
```

```
# Obtenir les données de la carte du Sénégal

Senegal <- raster::getData("GADM", country = "Senegal", level = 1)
centre<-coordinates(Senegal)
```

```r
noms1<-Senegal$NAME_1
w.nb1 <- poly2nb(Senegal,row.names = noms1,queen=TRUE)
par(oma = c(0, 0, 0, 0), mar = c(0, 0, 1, 0))

# Créer une carte de base du Sénégal avec une seule couleur de base

carte<-plot(Senegal,col = "gold", border = "tomato")

plot(projet_map["q12"],col=c("black","gray","red","yellow", "lawngreen"),add=TRUE)
attach(projet_map)
legend("topright",
legend = c("Aucun niveau","Niveau primaire","Niveau secondaire","Niveau Superieur"),
cex = 0.4,
fill = c("black","gray","red","yellow",
"lawngreen"))
text(centre[,1],centre[,2],noms1,cex=.55)
title("Repartition des PM en fonction du niveau d'instruction")
layoutLayer( sources = "Source:Calculs de l'auteur", frame = TRUE,
col = "NA", scale = NULL)
```



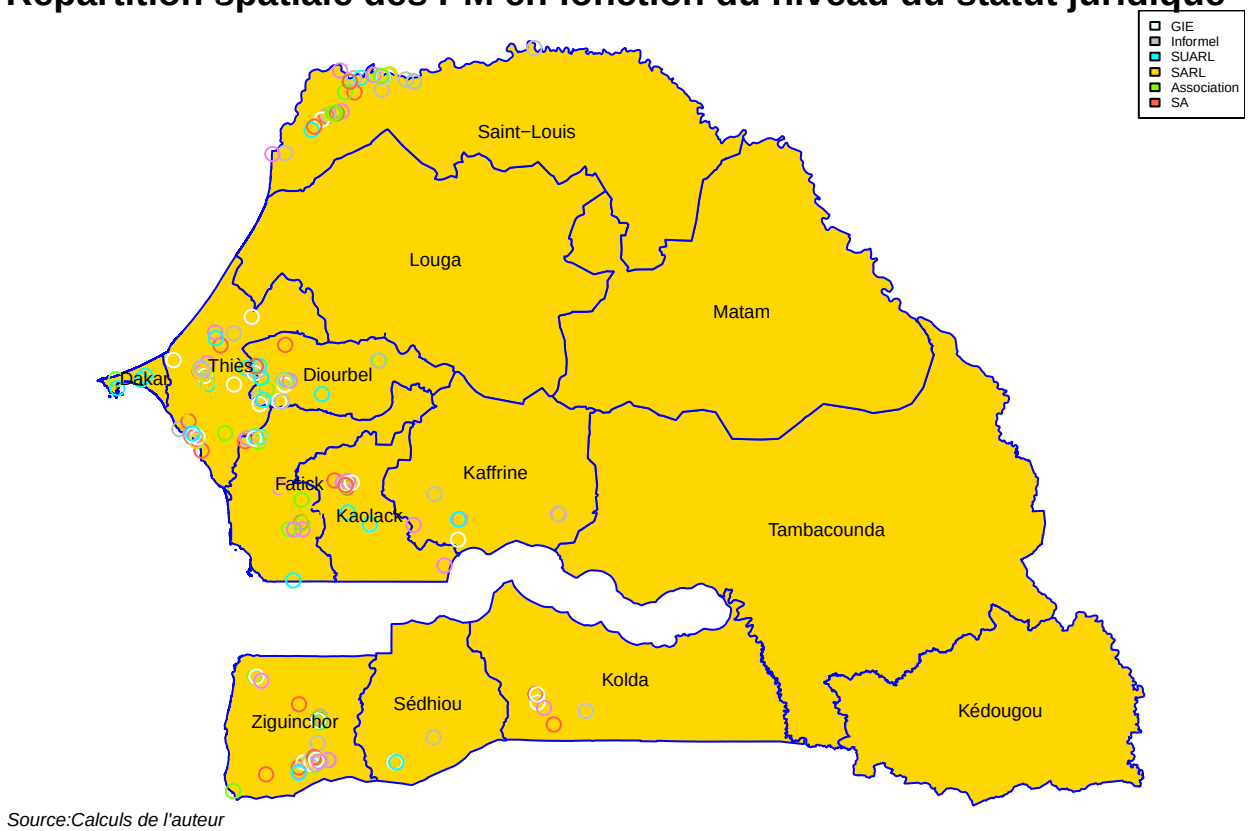
Source:Calculs de l'auteur

#3-4 Faites une analyse spatiale de votre choix

Mon choix se fait sur la variable *statut juridique* .En répartissant les PME agroalimentaires par les statuts juridiques sur une carte, nous pouvons obtenir plusieurs informations intéressantes qui pourraient nous aider à mieux comprendre le paysage économique des PME dans la région, **Distribution géographique des statuts juridiques** : nous pourrez visualiser comment les différents statuts juridiques sont répartis

géographiquement dans la région et cela nous permettra de voir s'il y a des concentrations particulières de certains types de statuts juridiques dans des régions du Sénégal.

```
#Obtenir les données de la carte du Sénégal

Senegal <- raster::getData("GADM", country = "Senegal", level = 1)

#Obtenir les coordonnées des centres des régions

centre<-coordinates(Senegal)

#Obtenir les coordonnées des centres
noms1<-Senegal$NAME_1
w.nb1 <- poly2nb(Senegal,row.names = noms1,queen=TRUE)
par(oma = c(0, 0, 0, 0), mar = c(0, 0, 1, 0))
carte<-plot(Senegal,col = "gold", border = "blue")
plot(projet_map["q12"],col=c("azure","gray","cyan","gold", "lawngreen","tomato","violet"),add=TRUE)
attach(projet_map)
legend("topright",
legend = c("GIE","Informel","SUARL","SARL","Association","SA"),
cex = 0.4,
fill = c("azure","gray","cyan","gold","lawngreen","tomato","violet"))
text(centre[,1],centre[,2],noms1,cex=.55)
title("Repartition spatiale des PM en fonction du niveau du statut juridique ")
layoutLayer(sources = "Source:Calculs de l'auteur", frame = TRUE,col = "NA", scale = NULL)
```



Source:Calculs de l'auteur

PARTIE 2

Nettoyage et gestion des données

- Renommer la variable “country_destination” en “destination” et définir les valeurs négatives comme manquantes.

```
donne_art<-read_excel("Base_Partie 2.xlsx")#donne_art,importation de la base contenant des donnés artif
donne_art<- rename(donne_art,destination=country_destination)## avec la fonction rename,je renomme la v

## Definissons les valeurs négative comme valeurs manquantes

donne_art$destination<-ifelse(donne_art$destination<0,NA,donne_art$destination)# avec la fonction ifels
##Verification s'il y'a des valeurs negatives
which(is.na(donne_art$destination))
```

```
##  [1]  3 11 13 14 21 27 29 30 39 53 56 58 67 71 74 78 83 85 87 89
```

```
which(donne_art$destination<0 )
```

```
## integer(0)
```

```
table(donne_art$destination)
```

```
##
##  3  4  5  6  8  9 10 11 13
##  7  2  8  3 10 22 18  1  6
```

Partie 2

2-1 Nettoyage et gestion des données

2-1-1 Créer une nouvelle variable contenant des tranches d’âge de 5 ans en utilisant la variable *“age”*.

```
attach(donne_art)# se fixer la base
#Nous allons déterminer le premier quartile et le troisième quartile de la variable "age" afin de deter
## Premier quartile
Q1<-quantile(donne_art$age)[2]

## Troisième  quartile
Q3<-quantile(donne_art$age)[4]

##calcul de la borne inferieur
born_inf=Q1-1.5*(Q3-Q1)

##calcul de la borne superieur
born_sup=Q3+1.5*(Q3-Q1)

## Detection et imputation des valeurs aberrantes par la moyenne des ages
donne_art$age_aberrante<-ifelse((donne_art$age<born_inf)|(donne_art$age>born_sup),mean(donne_art$age),d

##attach(partie2)
##determination des bornes de la tranche d'age
ecart<-5
bornes<-seq(min(donne_art$age_aberrante),max(donne_art$age_aberrante),by=ecart)
bornes
```

```
## [1] 15 20 25 30 35 40
```

```
##decoupage de la variable age des tranches d'age
donne_art$age<-cut(donne_art$age_aberrante,breaks = bornes)
table(donne_art$age)
```

```
##
## (15,20] (20,25] (25,30] (30,35] (35,40]
##      20      34      22      10      10
```

#2-1-2 Créer une nouvelle variable contenant le nombre d'entretiens réalisés par chaque agent recenseur.

```
attach(donne_art)
donne_art<-donne_art%>%group_by(enumerator)%>%dplyr::mutate(nbre_entretien=n())%>% distinct()
```

#2-1-3 Créer une nouvelle variable qui affecte aléatoirement chaque répondant à un groupe de traitement (1) ou de controle (0)

```
attach(donne_art)
# la taille de l'échantillon avec nrow
size=nrow(donne_art)
# la fonction sample() pour affecter aléatoirement a chaque repondant un groupe de traitement
donne_art$grp_traite<-sample(c(0,1),size,replace=TRUE,prob=NULL)
```

#2-1-4 Fusionner la taille de la population de chaque district (feuille 2) avec l'ensemble de données (feuille 1) afin que toutes les personnes interrogées aient une valeur correspondante représentant la taille de la population du district dans lequel elles vivent.

```
## Importation de la feuille
donne_art_f2<- data.frame(read_excel("Base_Partie 2.xlsx",sheet="district"))
## fusion des deux feuille avec la fonction merge () pour obtenir la base donne_art_taille
donne_art_taille<-donne_art%>%
merge(donne_art_f2,by="district")
```

#2-1-5 Calculer la durée de l'entretien et indiquer la durée moyenne de l'entretien par enquêteur.

```
library(lubridate)# fournit des fonctions pour travailler avec des dates et des heures plus facilement.
donne_art_taille<-donne_art_taille %>%
mutate(
dure_entre = time_length(
interval(
start = starttime,
end = endtime
),
unit = "hour"
)
)# création de la variable durée  de l'entretien
#select(nom, date_naissance, age) %>%
#glimpse()
colnames(donne_art_taille)
```

```
##  [1] "district"       "id"             "starttime"      "endtime"
##  [5] "enumerator"     "age"            "sex"            "children_num"
##  [9] "intention"      "destination"    "age_aberrante"  "nbre_entretien"
## [13] "grp_traite"     "population"     "dure_entre"
```

```
# durée moyen
donne_art_taille%>% group_by(enumerator)%>%
transmute(moyenne=mean(dure_entre))%>% distinct()%>%
```

```
kable
```

enumerator	moyenne
6	0.4307778
14	0.4260185
11	0.5580556
20	0.4794753
18	0.6143056
13	0.5265972
4	0.6080556
1	1.1357778
12	0.8027778
8	0.6688426
15	0.4775000
9	1.9127778
10	0.9212778
5	0.5593056
17	0.4881019
7	0.6194048

#2-1-6 Renommez toutes les variables de l'ensemble de données en ajoutant le préfixe "endline_" à l'aide d'une boucle.

```
## le nombre de colonne de la base
nbre_c<-ncol(donne_art_taille)
##une boucle qui va parcourir le nombre de ligne pour ajouter les prefixes
for (i in 1:nbre_c) {
colnames(donne_art_taille)[i]<-paste("endline_",colnames(donne_art_taille)[i],sep ="" )
}
View(donne_art_taille)
```

2-2 Analyse et visualisation des données

2-2-1 Créez un tableau récapitulatif contenant l'âge moyen et le nombre moyen d'enfants par district.

2-2-2 Testez si la différence d'âge entre les sexes est statistiquement significative au niveau de 5 %.

2-2-3Créer un nuage de points de l'âge en fonction du nombre d'enfants

2-2-4La variable "intention" indique si les migrants potentiels ont l'intention de migrer sur une échelle de 1 à 7. Estimez l'effet de l'appartenance au groupe de traitement sur l'intention de migrer.

2-2-4 Créez un tableau de régression avec 3 modèles. La variable de résultat est toujours "intention". Modèle A : Modèle vide - Effet du traitement sur les intentions. Modèle B : Effet du traitement sur les intentions en tenant compte de l'âge et du sexe. Modèle C : Identique au modèle B mais en contrôlant le district. Les résultats des trois modèles doivent être affichés dans un seul tableau.