RÉPUBLIQUE DU SÉNÉGAL

Un peuple- Un But- Une Foi

****

Agence Nationale de la Statistique et de la Démographie

Ecole Nationale de la Statistique et de l'Analyse Economique Pierre Ndiaye

PROJET R ENSAE 2023

MANIPULATION DES DONNEES - CARTOGRAPHIE - SHINY

Rédigé par :

Moussa AMADOU

Élève ingénieur statisticien économiste

Sous la supervision de :

HEMA Aboubakar

Resaerch-Analyst

Table des matières

1 INTRODUCTION

Après avoir suivi 30 heures de cours intensifs sur le logiciel R, encadrés par notre estimé maître, M. HEMA Aboubakar, analyste en recherche, nous avons eu l'opportunité de nous immerger dans diverses manipulations et fonctionnalités avancées de cet outil puissant. Ces cours ont été enrichissants et nous ont permis de développer un large éventail de compétences essentielles en analyse de données. Pendant cette formation, nous avons abordé de nombreux sujets

passionnants, notamment :

- L'appurement et le traitement des données, où nous avons appris à

nettoyer et préparer des ensembles de données volumineux pour une

analyse précise.

- L'édition de documents sur R, en utilisant des techniques avancées

pour produire des rapports et des présentations de qualité

professionnelle.

- La cartographie, qui nous a permis de représenter graphiquement des

données spatiales et géographiques de manière visuellement attrayante.

- Rshiny, une plateforme qui nous a ouvert les portes à la création

d'applications interactives basées sur des analyses réalisées avec R.

- L'analyse de texte (textmining), une compétence puissante pour

extraire des informations significatives à partir de grandes quantités

de textes non structurés.

- Le calcul parallèle, nous permettant d'accélérer le traitement de

données massives en utilisant plusieurs cœurs de processeurs.

- L'intégration de Python dans R, combinant la puissance des deux

langages pour des analyses plus complexes.

- La résolution de systèmes d'équations linéaires, qui nous a aidés à

résoudre des problèmes mathématiques avancés.

- L'utilisation avancée des tableaux, pour une manipulation plus

efficace et une gestion optimale des données.

Tout au long de ce parcours d'apprentissage enrichissant, nous avons

développé des compétences solides qui nous permettront d'aborder des

projets de données complexes avec confiance et expertise.

Le projet final que nous entreprenons maintenant vise à mettre en

pratique l'ensemble des connaissances acquises lors de ce cours. Nous

sommes impatients de relever ce défi et de démontrer notre

compréhension approfondie du logiciel R. Grâce à cette expérience, nous sommes convaincus que nous serons mieux préparés à relever les défis

analytiques du monde réel et à apporter des solutions innovantes aux

problèmes complexes auxquels nous serons confrontés dans notre domaine

de recherche.et de mettre en pratique les connaissances acquises en classe

2 Chargement des packages importants

```
knitr::opts_chunk$set(warning = FALSE, message = FALSE)
library(readxl) # importer charger des données excel
library(dplyr) # pour la manipulation les bases de données
library(gtsummary)# pour les tableaux statistiques
library(tidyverse)# pour
library(ggplot2) # pour les graphiques
library(ggridges)# pour faire la courbe de densité
library(huxtable) # pour les tableaux
library(fs) # pour la cartographie
library(gtsummary) # tableaux avancés
library(kableExtra) # tableaux
library(gridExtra) # pour mettre deux tableaux cote à cote
library(rgdal) # mettre deux tableaux cote à cote
library(stats) # pour effectuer des tests statistiques
library(devtools)
#devtools:: install_github("mikabr/ggpirate")
library(ggplot2)
```

3 PROJET 1

3.1 Importation et mise en forme

3.2 Importation de la base de données

```
#_____________________on importe avec read_xlsx_______________________________

projet<-read_xlsx("Base_Partie 1.xlsx")
```

Il est necessaire de renommer les variables d'étude

```
#_____________________Renomer les variables_________________________________
projet <- projet %>%
  dplyr::rename(Niveau_instruction = q25,

                Statut_juridique = q12,

                Proprietaire_Locataire=q81,

                Activite_principale=q8)
```

3.3 Tableau qui resume les valeurs manquantes par variable

```
#_________Compter les valeurs manquantes pour chaque variable_________________

missing_counts <- projet %>%

  dplyr:: summarise(across(everything(),
                           ~sum(is.na(.))))

#_______Convertir la table de comptage en format tbl_summary___________________

missing_tbl <- missing_counts %>%
  pivot_longer(everything(),
               names_to = "Variable",
               values_to = "Nbre observations manquantes")

#___________________Afficher le tableau résultant_________________________________

T1<-kableExtra::kable(as.data.frame(missing_tbl),
                      caption = "Nombre d'observations
                      manquantes par variable")
T1
```

Table 1: Nombre d'observations manquantes par variable

Variable	Nbre observations manquantes
key	0
q1	0
q2	0
q23	0
q24	0
q24a_1	0
q24a_2	0
q24a_3	0
q24a_4	0
q24a_5	0
q24a_6	0
q24a_7	0
q24a_9	0
q24a_10	0
Niveau_instruction	0
q26	0
Statut_juridique	0
q14b	1
q16	1
q17	131
q19	120
q20	0
filiere_1	0
filiere_2	0
filiere_3	0
filiere_4	0
Activite_principale	0
Proprietaire_Locataire	0
gps_menlatitude	0
gps_menlongitude	0
submissiondate	0
start	0
today	0

3.4 Vérifions s'il y a des valeurs manquantes pour la variable key

```
valeurs_manquantes_key <- sum(is.na(projet$key))

#_____________Vérification et identification des PME concernées_____________

if (valeurs_manquantes_key > 0) {
  pme_manquantes <- projet$nom_de_lacolonnedesPME[which(is.na(projet$key))]
  cat("Il y a",
      valeurs_manquantes_key,
      "valeurs manquantes pour la variable 'key' dans la base de données du projet.")
  cat("\nLes PME concernées sont :",
      unique(pme_manquantes))
} else {
  cat("pas de valeurs manquantes pour la variable 'key'.")
}
```

```
## pas de valeurs manquantes pour la variable 'key'.
```

3.5 Création de variables

```
projet <- dplyr::rename(projet,
                        region=q1,
                        departement=q2,
                        sexe=q23)
```

3.6 Créer la variable sexe_2

```
#_________________________initialisation_____________________________________

projet$sexe_2 <- 0

#_________________________affection des valeurs selon la condition___________

projet$sexe_2[projet$sexe == "Femme"] <- 1
```

3.7 Créer un data.frame nommé langues

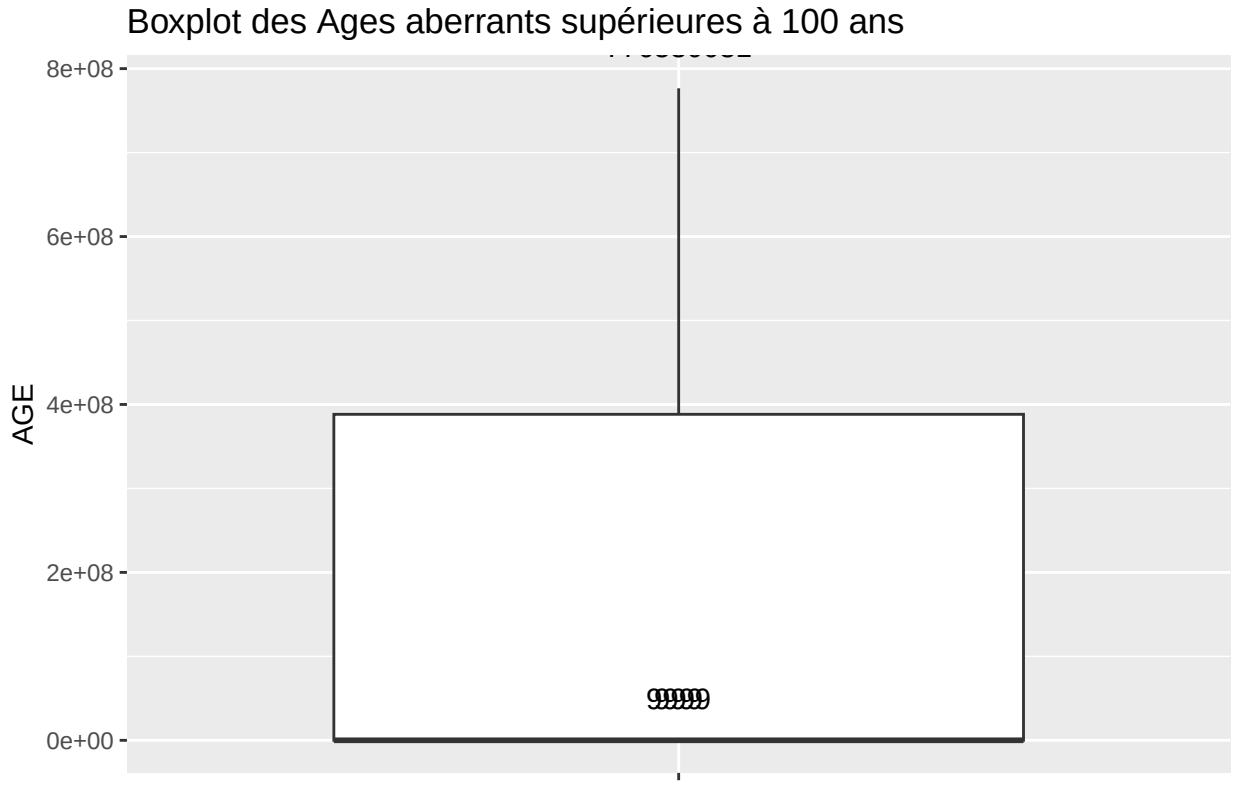
```
#_________________ Variables communes_____________________________

cle_variable <- "key"
#_____________________________________________________________________

language_variables <- grep("^q24a_",
                           names(projet),
                           value = TRUE)

langues <- projet[,
```

```
                  c(cle_variable,
                    language_variables)]

# Créer la variable "parle" égale au nombre de langues parlées

langues$parle <- rowSums(langues[,
                                 language_variables])

# Sélectionner uniquement les variables "key" et "parle"

langues <- langues[,
                   c(cle_variable,
                     "parle")]

# Fusionner les data.frames "projet" et "langues" en utilisant la variable "key"

projet <- merge(projet,
                langues,
                by = cle_variable)
```

4 Analyses descriptives

4.1 Detection des valeures manquantes

```
# Filtrer les observations avec des valeurs aberrantes

projet_aberrantes <- subset(projet,
                            q24 >99)

# Créer un boxplot avec les valeurs aberrantes

ggplot(projet_aberrantes,
       aes(x = "",
           y = q24)) +
  geom_boxplot(outlier.shape =NA) +

#Supprime les marqueurs par défaut des valeurs aberrantes

  geom_text(aes(label = q24),
            vjust = -1.5) +

#Affiche les valeurs numériques des valeurs aberrantes

  labs(title = "Boxplot des Ages aberrants supérieures à 100 ans",
       x = "",
       y = "AGE")
```



4.2 Analyse univariée Bivariée

```
tableau1 <- projet %>%
  gtsummary::tbl_summary(
    include = c("sexe",
                "Niveau_instruction",
                "Statut_juridique",
                "Proprietaire_Locataire"),
    statistic = all_categorical() ~ "{p} % "
  )

# Analyse bivariée

tableau2 <- projet %>%
  gtsummary::tbl_summary(
  include = c("sexe",
              "Niveau_instruction",
              "Statut_juridique",
              "Proprietaire_Locataire"),
  by="sexe",
  statistic = all_categorical() ~ "{p} % "
)

gtsummary:: tbl_merge(
```

```
 list(tableau1,
      tableau2),
 tab_spanner = c("**Analyse univariée**",
                 "**Analyse bivariée**"))%>%
 bold_labels() %>%
 italicize_levels() %>%
 modify_header(
   list(
     label ~ "**Variables**")
 )%>%
 add_header_above()%>%
as_gt()
```

Variables	**Analyse univariée** **N = 250**[1]	**Analyse bivariée** **Femme**, N = 191[1]	**Homme**, N = 59[1]
sexe			
Femme	76 %		
Homme	24 %		
Niveau_instruction			
Aucun niveau	32 %	37 %	15 %
Niveau primaire	22 %	25 %	14 %
Niveau secondaire	30 %	29 %	31 %
Niveau Superieur	16 %	8.9 %	41 %
Statut_juridique			
Association	2.4 %	1.6 %	5.1 %
GIE	72 %	78 %	51 %
Informel	15 %	17 %	10 %
SA	2.8 %	0.5 %	10 %
SARL	5.2 %	1.0 %	19 %
SUARL	2.8 %	2.1 %	5.1 %
Proprietaire_Locataire			
Locataire	9.6 %	8.4 %	14 %
Propriétaire	90 %	92 %	86 %

[1]% %

4.3 ANALYSES SUPPLEMENTAIRES

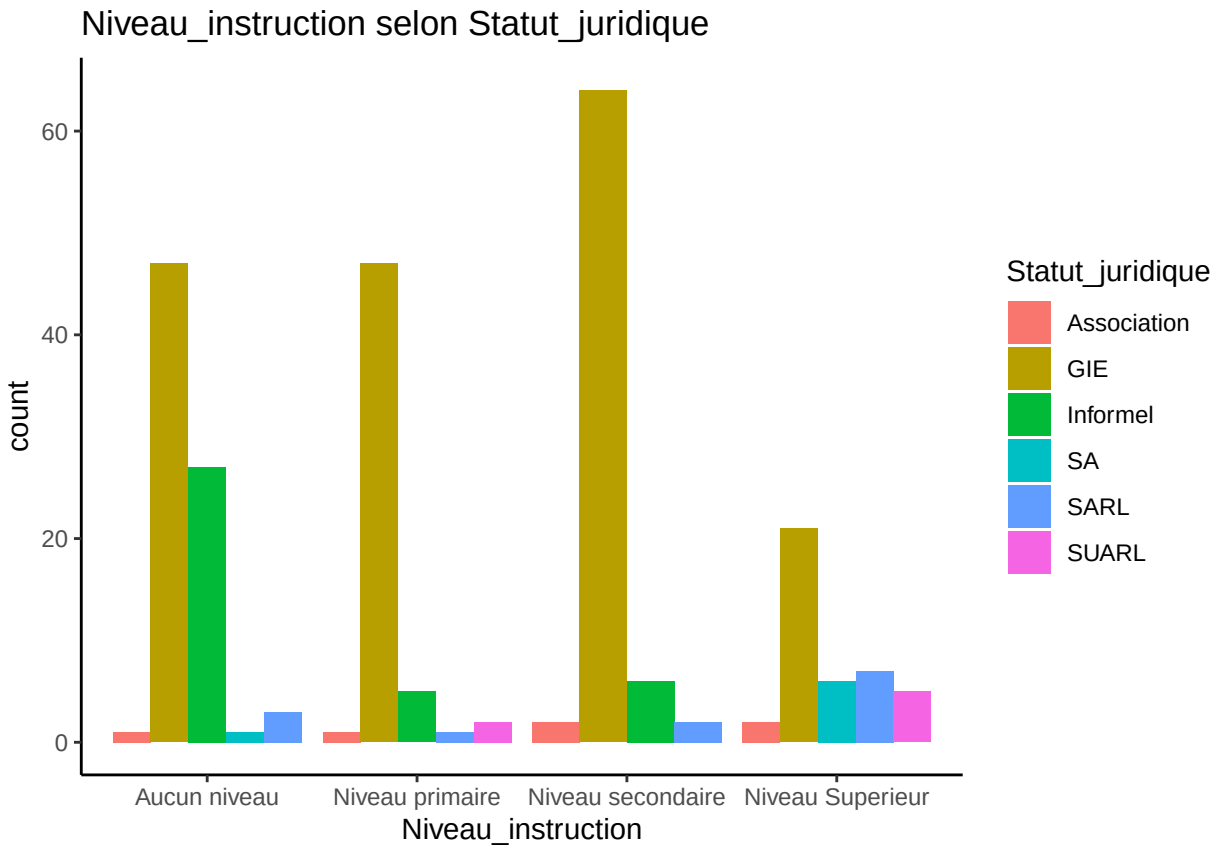
```
# Sélectionner les variables pour l'analyse par filière selon le sexe

tableau3 <- projet %>%
  subset(filiere_1 == 1) %>%
  gtsummary:: tbl_summary(
    include = region ,
    by = filiere_1 ,
    statistic = all_categorical() ~ "{p} % ",
    percent = "col"
  )
tableau4 <- projet %>%
  subset(filiere_2==1) %>%
  gtsummary:: tbl_summary(
    include = region ,
    by = filiere_2 ,
    statistic = all_categorical() ~ "{p} % ",
    percent = "col"
  )
tableau5 <- projet %>%
  subset(filiere_3==1) %>%
  gtsummary:: tbl_summary(
    include = region ,
    by = filiere_3 ,
    statistic = all_categorical() ~ "{p} % ",
    percent = "col"
  )
tableau6 <- projet %>%
  subset(filiere_4==1) %>%
  gtsummary:: tbl_summary(
    include = region ,
    by = filiere_4 ,
    statistic = all_categorical() ~ "{p} % ",
    percent = "col"
  )
tbl_merge(
  list(tableau3, tableau4,tableau5,tableau6),
  tab_spanner = c("**FILIERE ARACHIDE**",
                  "**FILIERE ANACARDE**",
                  "**FILIERE MANGUE**",
                  "**FILIERE RIZ**"))%>%
  bold_labels() %>%
  italicize_levels() %>%
  modify_header(
    list(
      label ~ "**PRODUCTION**")
  )%>%
  add_header_above() %>%
  as_hux_table()# transformer le tableau
```

	FILIERE ARACHIDE	FILIERE ANACARDE	FILIERE MANGUE	FILIERE RIZ
PRODUCTION	**1**, N = 108	**1**, N = 61	**1**, N = 89	**1**, N = 92
region				
Diourbel	31 %		1.1 %	
Fatick	11 %	34 %	3.4 %	4.3 %
Kaffrine	7.4 %		5.6 %	1.1 %
Kaolack	19 %		7.9 %	4.3 %
Kolda	0.9 %	8.2 %		4.3 %
Saint-Louis	0.9 %		47 %	
Thiès	25 %		28 %	35 %
Ziguinchor	5.6 %	51 %	6.7 %	47 %
Dakar		1.6 %		1.1 %
Sédhiou		4.9 %		3.3 %

% %

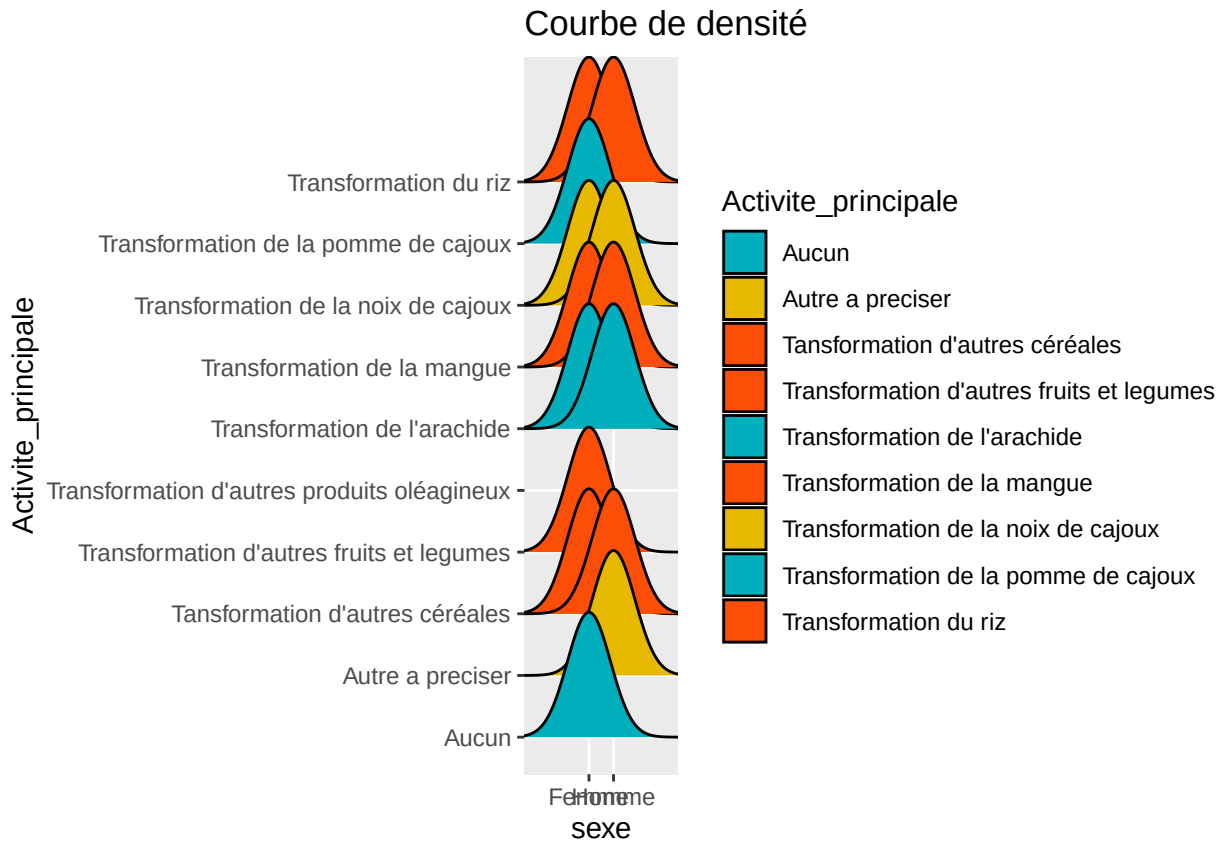
4.3.1 Representation

```
ggplot(projet,
       aes(x = Niveau_instruction,
                   fill = Statut_juridique))+
 geom_bar(position = "dodge",
          stat = "count")+
  labs(title = "Niveau_instruction selon Statut_juridique")+
 theme_classic()
```

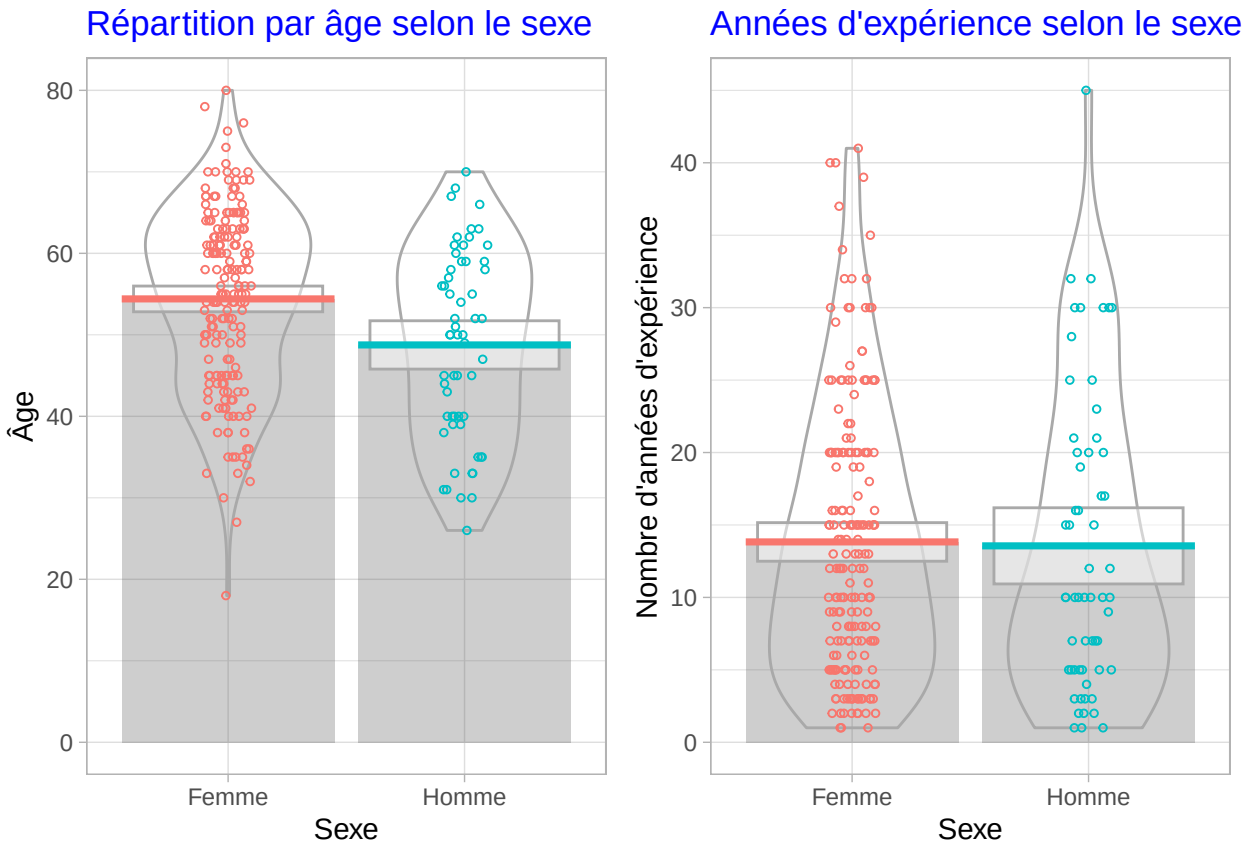

Niveau_instruction selon Statut_juridique
count
60
40
20
0
Aucun niveau
Niveau primaire
Niveau secondaire
Niveau Superieur
Niveau_instruction
Statut_juridique
Association
GIE
Informel
SA
SARL
SUARL

4.4 Courbes de densité

```
ggplot(projet, aes(x = sexe,
                   y = Activite_principale)) +

  geom_density_ridges(aes(fill =  Activite_principale)) +

  scale_fill_manual(values = c("#00AFBB",
                               "#E7B800",
                               "#FC4E07",
                               "#FC4E07",
                               "#00AFBB",
                               "#FC4E07",
                               "#E7B800",
                               "#00AFBB",
                               "#FC4E07"))+

  labs(title = "Courbe de densité")
```



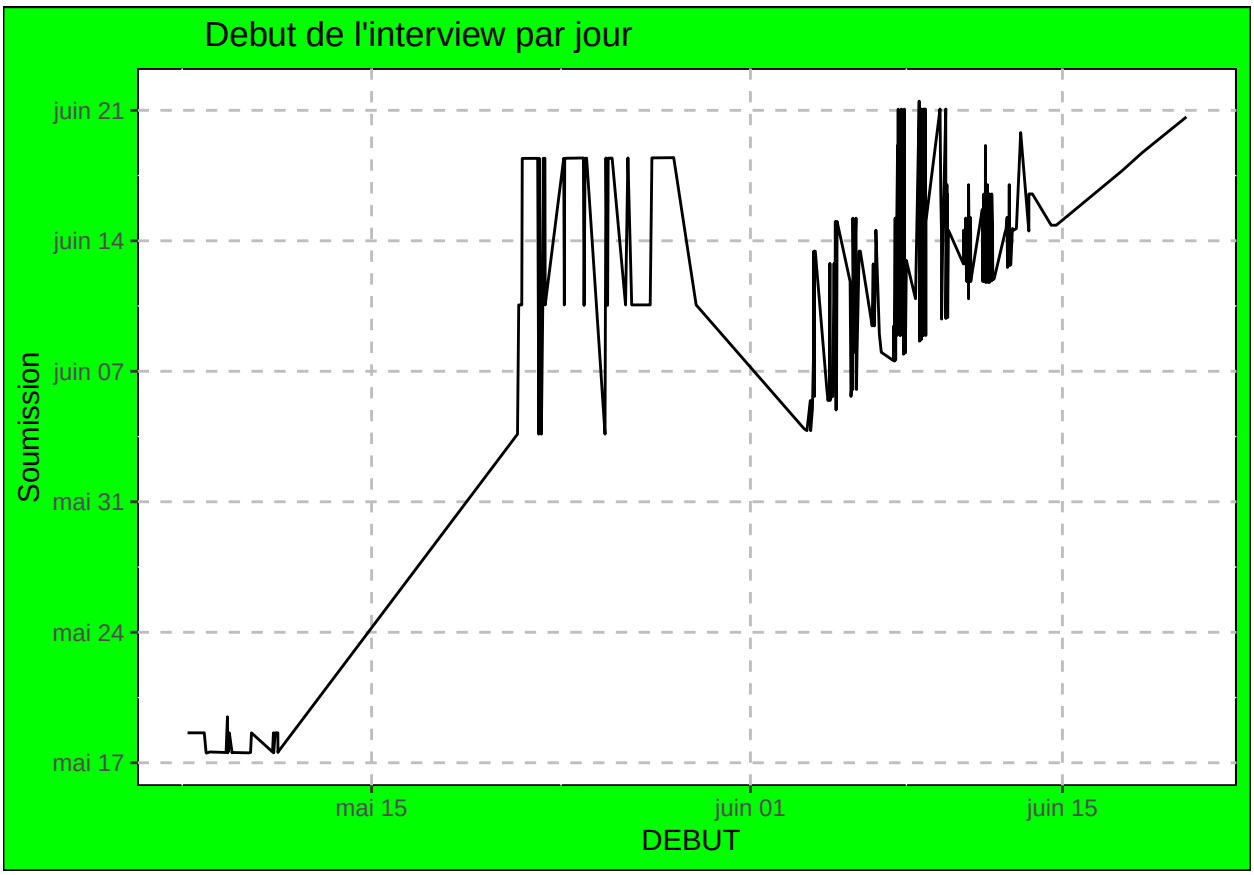
```r
# Chargement des bibliothèques nécessaires
library(ggplot2)
library(ggpirate)
library(gridExtra)

# Répartition 1 : Filtrer les données et renommer la variable "q24" en "age"
Repartition1 <- projet %>%
  rename(age = q24) %>%
  filter(age < 120)

# Répartition 2 : Filtrer les données et renommer la variable "q26" en "Annees_experience"
Repartition2 <- projet %>%
  rename(Annees_experience = q26) %>%
  filter(Annees_experience < 50)

# Création du premier graphique (Répartition par âge selon le sexe)
plot1 <- ggplot(Repartition1, aes(x = sexe, y = age)) +
  geom_pirate(aes(colour = sexe)) +
  xlab("Sexe") +
  ylab("Âge") +
  ggtitle("Répartition par âge selon le sexe") +
  theme_light() +
  theme(plot.title = element_text(color = "blue"))

# Création du deuxième graphique (Années d'expérience selon le sexe)
plot2 <- ggplot(Repartition2, aes(x = sexe, y = Annees_experience)) +
  geom_pirate(aes(colour = sexe)) +
  xlab("Sexe") +
  ylab("Nombre d'années d'expérience") +
  ggtitle("Années d'expérience selon le sexe") +
  theme_light() +
  theme(plot.title = element_text(color = "blue"))

# Afficher les graphiques côte à côte sur une même ligne
grid.arrange(plot1, plot2, ncol = 2)
```



4.5 Visualisation des données temporelles

```
ggplot(projet,
       aes(x=start,
           y=submissiondate))+

   geom_line()+

  theme(plot.title = element_text(hjust = 0.1)) +

  theme(plot.background = element_rect(fill = "green",

                                       colour = "black")) +

  theme(panel.background = element_rect(fill = "white",

                                        colour = "black")) +

  theme(panel.grid.major = element_line(colour = "gray",

                                        linetype = "dashed"))+

  labs(title = "Debut de l'interview par jour",
```

```
        x = " DEBUT",

        y = "Soumission")
```



5 Cartographie

5.1 Transformer le data.frame en données géographiques dont l'objet sera nommé projet_map.

```
library(sf)

# chargement des données du senegal gadm

sen_sf <- st_read("données_SEN/gadm36_SEN_1.shp")
```

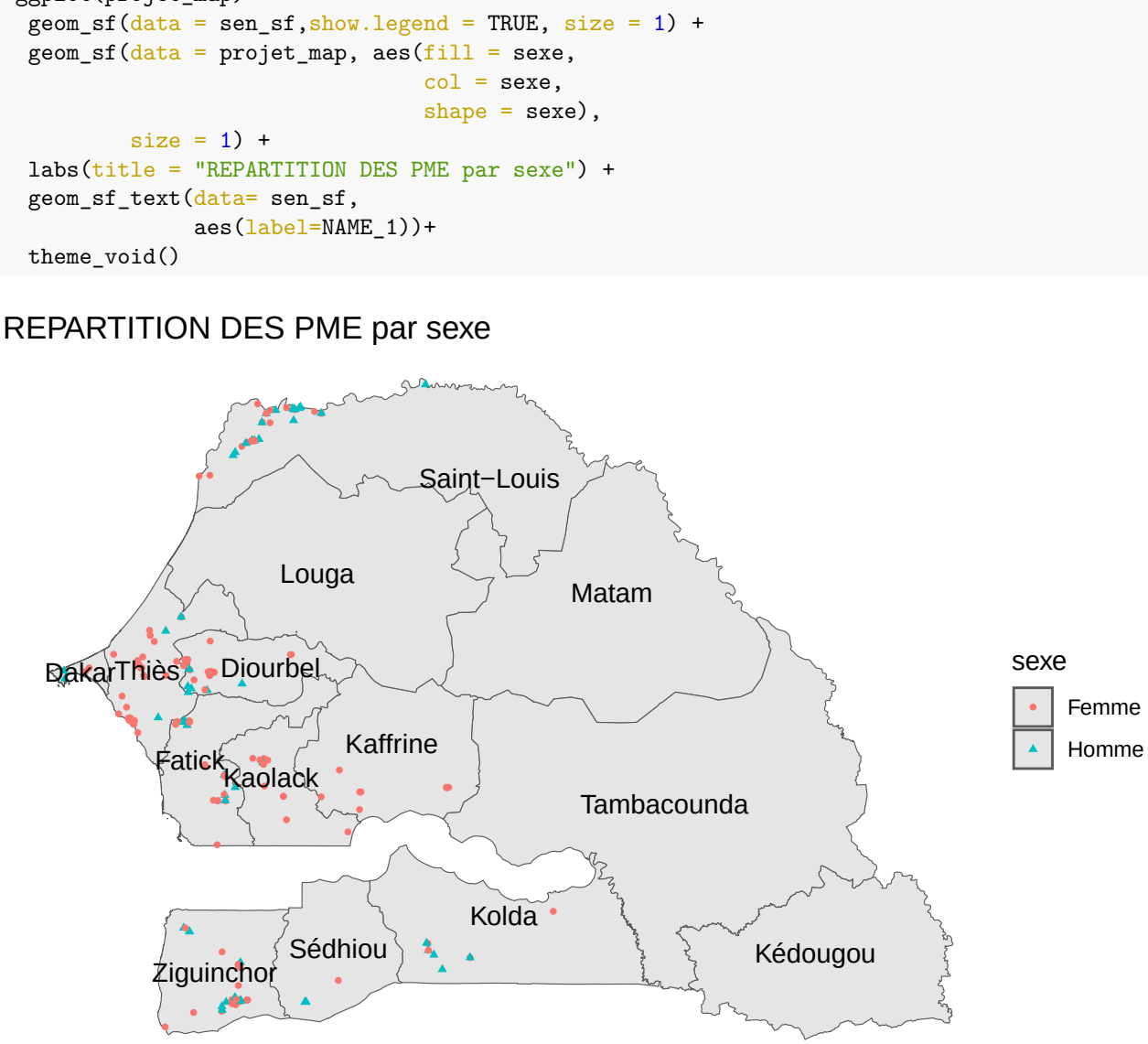
```
## Reading layer `gadm36_SEN_1' from data source
##   `C:\Users\USER\Documents\Projet_R_Ensae_2023\Projet_R_Ensae\données_SEN\gadm36_SEN_1.shp'
##   using driver `ESRI Shapefile'
## Simple feature collection with 14 features and 10 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: -17.54319 ymin: 12.30786 xmax: -11.34247 ymax: 16.69207
## Geodetic CRS:  WGS 84
```

```
# Extraire le CRS
crs_sen_sf <- st_crs(sen_sf)

# Créer un objet sf à partir des coordonnées géographiques

 projet_map <- st_as_sf(projet,
                        coords=c("gps_menlongitude",

                                 "gps_menlatitude"),

                        crs=crs_sen_sf)

# mergeons les deux bases

projet_map<-st_join(projet_map,

                    sen_sf)

# afficher la classe

class(projet_map)
```

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

5.2 Réprésentation spatiale des PME suivant le sexe

```
ggplot(projet_map) +
 geom_sf(data = sen_sf,show.legend = TRUE, size = 1) +
 geom_sf(data = projet_map, aes(fill = sexe,
                                col = sexe,
                                shape = sexe),
         size = 1) +
 labs(title = "REPARTITION DES PME par sexe") +
 geom_sf_text(data= sen_sf,
              aes(label=NAME_1))+
 theme_void()
```

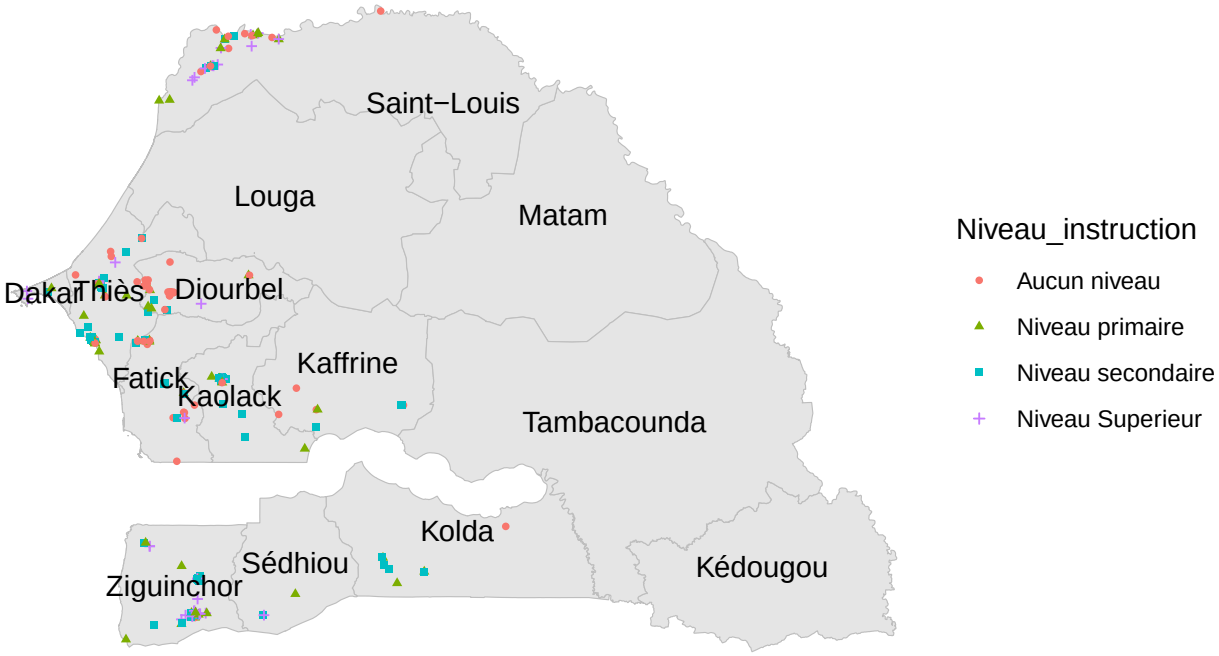


5.3 Réprésentation spatiale des PME suivant le niveau d'instruction

```
ggplot(projet_map) +
  geom_sf(data = sen_sf,color = "grey",
          size = 1) +
  geom_sf(aes(fill = Niveau_instruction,
              col = Niveau_instruction,
              shape=Niveau_instruction),
          size = 1) +
   labs(title = "REPARTITION DES PME selon le Niveau d'instruction")+
```

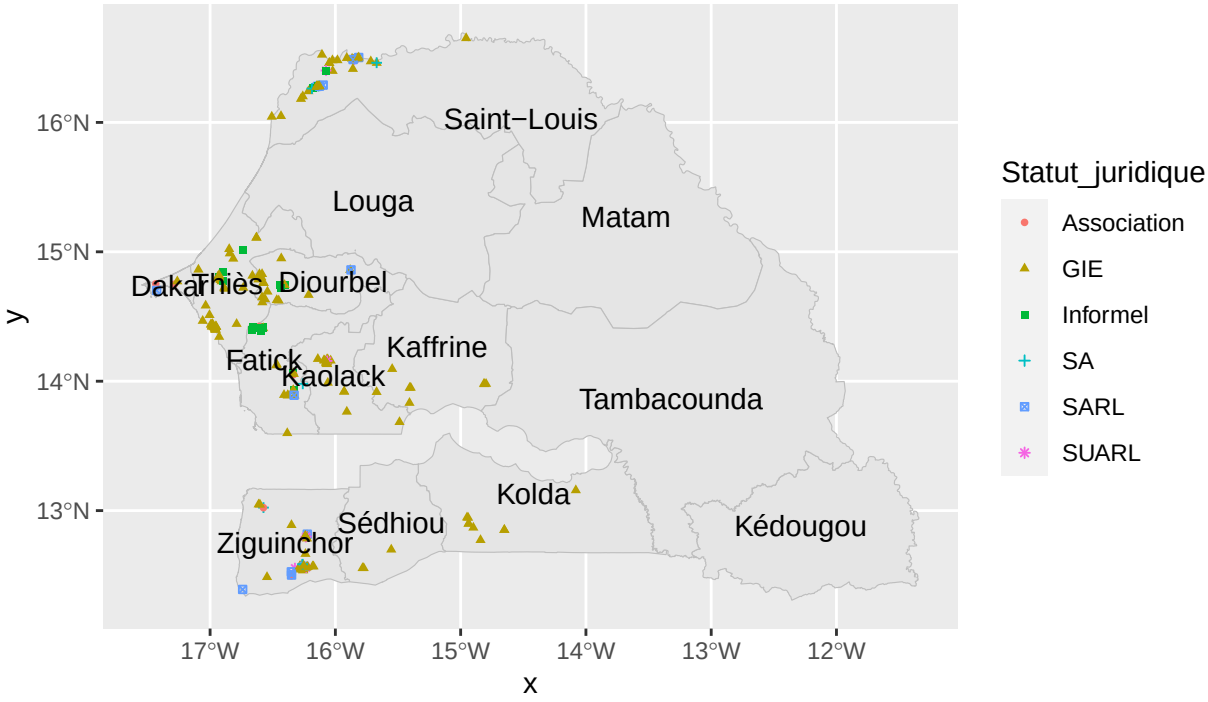
```
geom_sf_text(data= sen_sf,
             aes(label=NAME_1))+
theme_void()
```

REPARTITION DES PME selon le Niveau d'instruction



5.4 Une analyse spatiale de votre choix

```
ggplot(projet_map) +
geom_sf(data = sen_sf, color = "grey",
        size = 1) +
geom_sf(aes(fill = Statut_juridique,
            col = Statut_juridique,
            shape = Statut_juridique),
        size = 1) +
geom_sf_text(data= sen_sf,
             aes(label=NAME_1))
```




```
  labs(title = "PME selon le statut juridique") +
  theme_void()
```

```
## NULL
```

6 Partie 2

6.1 Nettoyage et gestion des données

Importation de la base

```
data <- read_xlsx("Base_Partie 2.xlsx",
                      sheet = "data",
                      col_names = TRUE)
```

6.2 Renommer la variable “country_destination” en “destination”

```
data<-data %>%
  dplyr::rename(destination=country_destination)
```

6.3 Remplacer les valeurs négatives par des valeurs manquantes

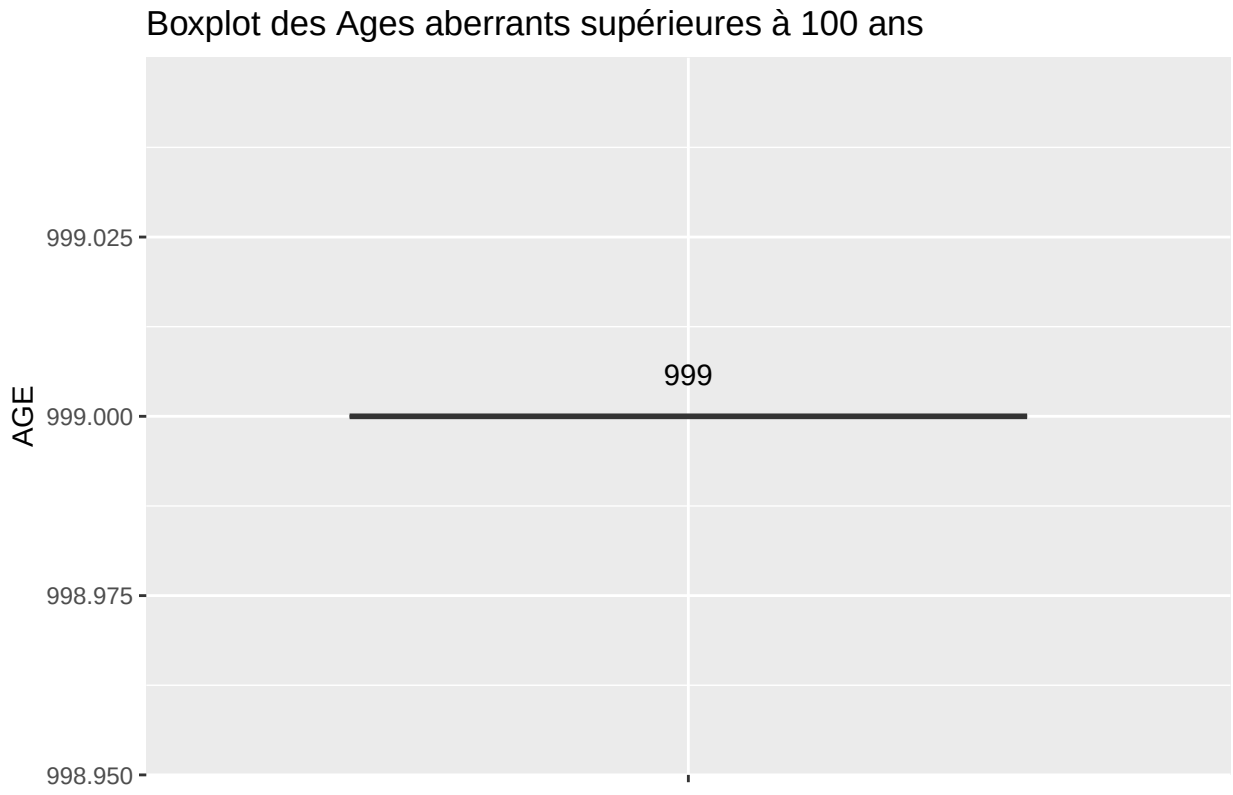
```
data$destination <- ifelse(data$destination >= 0,
                           data$destination,
                           NA)
```

6.4 Creation de nouvelle variable

Pour ce cas ci nous devons nous d’abord detecter les outliers

pour ce faire nous procedons comme suit :

```
# Filtrer les observations avec des valeurs aberrantes

age_aberrant <- subset(data,
                       age >100)

# Créer un boxplot avec les valeurs aberrantes

ggplot(age_aberrant, aes(x = "",
                         y = age)) +
  geom_boxplot(outlier.shape =NA) +
  geom_text(aes(label = age),
            vjust = -1.5) +
  labs(title = "Boxplot des Ages aberrants supérieures à 100 ans",
       x = "",
       y = "AGE")
```



Au vu de ce qui préceède nous procéderons à une méthode d'imputation par la moyenne

```
# Déterminer les bornes pour les âges aberrants

age_min_aberrant <- 0
age_max_aberrant <- 100

# Calculer l'âge moyen pour les âges non-aberrants

age_mean_non_aberrant <- mean(data$age[data$age >
                                         age_min_aberrant &
                                         data$age <
                                         age_max_aberrant],
                              na.rm = TRUE)

# Remplacer les âges aberrants par l'âge moyen non-aberrant

data <- data %>%
  mutate(age = ifelse(age <= age_min_aberrant | age >=


                        age_max_aberrant,

                      age_mean_non_aberrant,

                      age))
```

```r
# Conserver uniquement les chiffres après la virgule des âges

data <- data %>%

  mutate(age = floor(age))

# Déterminer l'âge minimum et l'âge maximum

age_min <- min(data$age,
               na.rm = TRUE)
age_max <- max(data$age,
               na.rm = TRUE)

# Calculer le nombre de classes d'âge avec une amplitude de 5 ans

nb_classes <- ceiling((age_max - age_min) / 5)

# Générer les bornes des classes d'âge

bornes_classes <- seq(age_min,
                      length.out = nb_classes + 1,
                      by = 5)

# Utiliser la fonction cut() pour créer la variable "classe_age"

data$classe_age <- cut(data$age,
                       breaks = bornes_classes,
                       include.lowest = TRUE,
                       right = FALSE)
```

6.5 Créer une nouvelle variable contenant le nombre d'entretiens

```r
data <- data %>%

  group_by(enumerator) %>%

  mutate(nombre_entretiens = n()) %>%

  ungroup()
```

6.6 Créer une nouvelle variable "groupe_traitement" avec des affectations aléatoires

```r
set.seed(123)  # Pour fixer l'alea

data$groupe_traitement <- sample(c(0, 1),

                                 size = nrow(data),

                                 replace = TRUE)
```

6.7 Fusionner la taille de la population de chaque district

```
# Lecture des données de la deuxième feuille (districts)

district <- read_excel("Base_Partie 2.xlsx",

                       sheet = "district")

# Fusion des données en utilisant la variable commune/district

donnees_fusionnees <- merge(data,
                            district,
                            by = "district",
                            all.x = TRUE)
```

6.8 Calculer la durée de l'entretien et indiquer la durée moyenne de l'entretien par enquêteur

```
# Calculons la durée de l'entretien par enquêteur

donnees_fusionnees <- donnees_fusionnees %>%

  dplyr:: mutate(duree_entretien = endtime - starttime)

#Calculons la durée moyenne de l'entretient

duree_moyenne_entretient <- weighted.mean(donnees_fusionnees$duree_entretien,

                                          donnees_fusionnees$nombre_entretiens,

                                          na.rm = TRUE )
```

6.9 Indiquons la durée moyenne de l'entretien par enquêteur

```
duree_moyenne_par_enqueteur <- donnees_fusionnees %>%

  group_by(enumerator) %>%

  summarise(duree_moyenne = weighted.mean(duree_entretien,
                                          nombre_entretiens,
                                          na.rm = TRUE))
print(duree_moyenne_par_enqueteur)
```

```
## # A tibble: 16 x 2
##    enumerator duree_moyenne
##         <dbl> <drtn>
##  1          1  68.14667 mins
##  2          4  36.48333 mins
##  3          5  33.55833 mins
##  4          6  25.84667 mins
```

```
##  5          7  37.16429 mins
##  6          8  40.13056 mins
##  7          9 114.76667 mins
##  8         10  55.27667 mins
##  9         11  33.48333 mins
## 10         12  48.16667 mins
## 11         13  31.59583 mins
## 12         14  25.56111 mins
## 13         15  28.65000 mins
## 14         17  29.28611 mins
## 15         18  36.85833 mins
## 16         20  28.76852 mins
```

6.10 Renommez toutes les variables de l'ensemble de données en ajoutant le préfixe "endline_" à l'aide d'une boucle.

```
nouvelles_colonnes <- paste0("endline_",
                             colnames(donnees_fusionnees))

donnees_fusionnees <- donnees_fusionnees %>%

  rename_with(~ nouvelles_colonnes,

              everything())
```

6.11 Analyse et visualisation des données

Créez un tableau récapitulatif contenant l'âge moyen et le nombre moyen

```
# Calculons les centres des d'ages

tableau_recapitulatif <- donnees_fusionnees %>%

  group_by(endline_district) %>%

  summarise(age_moyen = weighted.mean(endline_age,

                                      endline_population,

                                      na.rm = TRUE),

            enfants_moyen = weighted.mean(endline_children_num,

                                          endline_population,

                                          na.rm = TRUE))
```

6.12 Testez si la différence d'âge entre les sexes au seuil de 5%

```
# Recoder la variable "endline_sex" en 1 pour "femme"
#et 0 pour "homme"

donnees_fusionnees$endline_sex <-

  ifelse(donnees_fusionnees$endline_sex == 1,
         "femme",
         "homme")
```

```
donnees_fusionnees %>%   tbl_summary(

  include = "endline_age",

  by= "endline_sex",

  statistic = all_continuous() ~ "{mean} ± {sd}"

) %>%

  add_difference(

  ) %>%
  as_hux_table()
```

Characteristic	**femme**, N = 11	**homme**, N = 86	**Difference**	**95% CI**	**p-value**
endline_age	22 ± 5	26 ± 6	-3.7	-7.3, -0.11	0.044

Mean ± SD

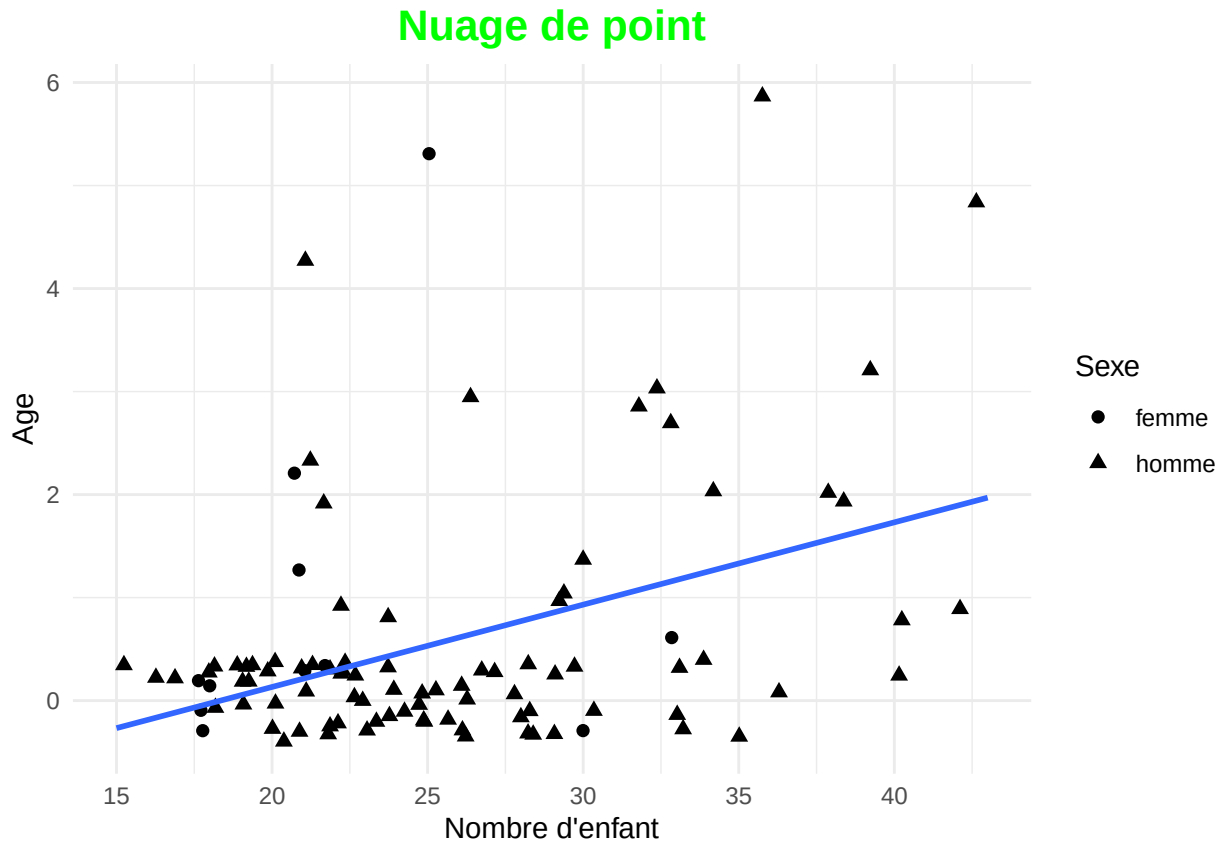
Welch Two Sample t-test

CI = Confidence Interval

6.13 Créer un nuage de points de l'âge

```
library(ggplot2)
nuage_points <- ggplot(data = donnees_fusionnees,
                       aes(x = endline_age,
                           y = endline_children_num)) +
  geom_jitter(aes(shape = endline_sex),
              size = 2)+
  geom_smooth(method = "lm",
              se = FALSE)+
  labs(x="Nombre d'enfant",
       y = "Age",
       title = "Nuage de point")+
  labs( col = "Sexe",
        shape = "Sexe") +
```

```
  theme_minimal()+
  theme(plot.title = element_text(color = "green",
                                  size = 16,
                                  face = "bold",
                                  hjust = 0.5 ))
# Afficher le nuage de points
nuage_points
```



6.14 Estimer l'effet de l'appartenance au groupe de traitement

6.15 Tableau avec trois modeles

```
modele <- stats::lm(endline_intention ~ endline_groupe_traitement,

                    data = donnees_fusionnees)
modele %>%

  gtsummary:: tbl_regression(
    label=list(endline_traitement = "TRAITEMENT")
  ) %>%
  as_flex_table()
```

Characteristic	Beta	95% CI[1]	p-value
endline_groupe_traitement	0.34	-0.36, 1.0	0.3

[1]CI = Confidence Interval

Tableaux des trois modeles

```
model_A <- modele %>%

  gtsummary::tbl_regression()

model_B <- stats::lm(endline_intention~endline_age +endline_sex,
                     data = donnees_fusionnees) %>%

  gtsummary:: tbl_regression()

model_C <- stats::lm(endline_intention~endline_age + endline_sex +
                       endline_district,data = donnees_fusionnees) %>%

  gtsummary::tbl_regression()

gtsummary:: tbl_stack(

  list(model_A ,
       model_B ,
       model_C),
  group_header = c("Modele A",
                   "Modele B",
                   "Model C")
) %>%

  as_flex_table()
```

Group	Characteristic	Beta	95% CI[1]	p-value
Modele A	endline_groupe_traitement	0.34	-0.36, 1.0	0.3
Modele B	endline_age	0.00	-0.05, 0.06	>0.9

[1]CI = Confidence Interval

Group	Characteristic	Beta	95% CI[1]	p-value
	endline_sex			
	femme	—	—	
	homme	0.92	-0.19, 2.0	0.10
Model C	endline_age	0.01	-0.05, 0.07	0.8
	endline_sex			
	femme	—	—	
	homme	0.82	-0.30, 1.9	0.15
	endline_district	0.09	-0.06, 0.25	0.2

[1]CI = Confidence Interval

7 Conclusion

En conclusion, notre parcours de 30 heures de formation intense sur le logiciel R sous la direction bienveillante de M. HEMA Aboubakar, analyste en recherche, a été une expérience extrêmement enrichissante. Nous avons été exposés à une diversité de manipulations avancées et de fonctionnalités puissantes de cet outil , nous permettant de développer un éventail complet de compétences en analyse de données.

Chaque sujet abordé, du nettoyage des données à l'intégration de Python dans R en passant par la résolution de systèmes d'équations linéaires et l'analyse de texte, nous a apporté des connaissances essentielles pour mener à bien des projets de données complexes. La maîtrise de la cartographie et de Rshiny nous a ouvert de nouvelles perspectives dans la création d'applications interactives et la visualisation de données.

Grâce à ce projet final, nous aurons l'occasion de démontrer notre compréhension approfondie du logiciel R et de mettre en pratique l'ensemble de nos apprentissages. Nous sommes convaincus que cette expérience nous a mieux préparés à affronter les défis analytiques du monde réel, et nous sommes impatients d'apporter des solutions innovantes aux problèmes complexes que nous rencontrerons dans notre domaine de recherche.

Nous tenons à exprimer notre gratitude envers notre formateur, M. HEMA Aboubakar, dont l'expertise et le dévouement ont grandement contribué à notre apprentissage. Nous sommes également reconnaissants envers les équipes d'exposés pour cette opportunité d'enrichir nos compétences et de nous épanouir professionnellement.

Enfin nous sommes enthousiastes à l'idée d'appliquer ces nouvelles compétences dans nos projets futurs et de continuer à explorer les possibilités infinies offertes par le logiciel R dans le domaine passionnant de l'analyse de données.