

**République du Sénégal**

Un Peuple-Un But-Une Foi



**Un Peuple-Un But-Une foi**



**Agence nationale de la statistique et de la démographie**

\*\*\*\*\*



**Ecole nationale de la statistique et de l'analyse économique**

**Ensae-Pierre-NDIAYE**

**Projet final - Logiciel statistique R**

Rédigé par :

**NGATCHA NOUTCHA Stephy Nadia**

Elève ingénieur statisticienne économiste

Sous la supervision de :

**M. HEMA Aboubacar**

Research-Analyst

**Juillet 2023**

# Sommaire

<b>PARTIE I</b>	<b>4</b>
1.Préparation des données . . . . .	4
1.1 Description . . . . .	4
1.2 importation et mise en forme . . . . .	4
1.2.1 importation de la base de donné . . . . .	4
1.2.2 faisons un tableau résumant les valeurs manquantes par valeur . . . . .	4
1.2.3 Vérifions s'il y a des valeurs manquantes pour la variable key dans la base projet. Si oui, identifier la (ou les) PME concernée(s). . . . .	4
1.3 Création de variables . . . . .	5
1.3.1 Renomons les variables q1,q2 et q23 respectivement en région,département et sexe	5
1.3.2 Créons la variable sexe_2 qui vaut 1 si sexe égale à femme et 0 sinon . . . . .	5
1.3.3 Créons un data frame nommer langue qui prend la variable key et les variables correspondantes aux langues parlées. . . . .	5
ANALYSES DESCRIPTIVES . . . . .	5
1.3 UN PEU DE CARTOGRAPHIE . . . . .	6
1.3.1 Transformons le data.frame en données géographique dont l'objet sera nommé projet_map . . . . .	6
1.3.2 représentation spatial des PME suivant le sexe . . . . .	6
1.3.3 représentation spatial des PME suivant le niveau d'instruction . . . . .	7
1.3.4 faisons une analyse spatiale de notre choix . . . . .	8
<b>PARTIE 2: NETTOYAGE ET GESTION DES DONNEES importation de la base de</b>	<b>10</b>
Importation de Base_partie 2 . . . . .	10
2.1 Nettoyage et gestion des données . . . . .	10
2.1.1 Renommons la variable "country_destination" en "destination" et définissons les valeurs négatives comme manquantes. . . . .	10
2.1.2 Créons une nouvelle variable contenant des tranches d'âge de 5 ans en utilisant la variable "age". . . . .	11
2.1.3 Créer une nouvelle variable contenant le nombre d'entretiens réalisés par chaque agent recenseur. . . . .	11
2.1.4 Créer une nouvelle variable qui affecte aléatoirement chaque répondant à un groupe de traitement (1) ou de controle (0). . . . .	11
2.1.5 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. . . . .	12
2.1.6 Calculer la durée de l'entretien et indiquer la durée moyenne de l'entretien par enquêteur. . . . .	12
2.1.7 Renommez toutes les variables de l'ensemble de données en ajoutant le préfixe "endline_" . . . . .	12
2.2 ANALYSE ET VISUALISATION DES DONNEES . . . . .	13
2.2.1 Créez un tableau récapitulatif contenant l'âge moyen et le nombre moyen d'enfants par district. . . . .	13
2.2.2 Testons si la différence d'âge entre les sexes est statistiquement significative au niveau de 5 %. . . . .	13
2.2.3 Créer un nuage de points de l'âge en fonction du nombre d'enfants . . . . .	14
2.2.4 La variable "intention" indiquons si les migrants potentiels ont l'intention de migrer sur une échelle de 1 à 7. Estimons l'effet de l'appartenance au groupe de traitement sur l'intention de migrer. . . . .	15

2.2.5 Créez un tableau de régression avec 3 modèles. Les résultats des trois modèles doivent être affichés dans un seul tableau. . . . .	15
<b>FIN PARTIE I et II</b>	<b>16</b>
<b>PARTIE III</b>	<b>16</b>

# PARTIE I

## 1.Préparation des données

### 1.1 Description

Dans cette sous partie aucune n'instruction n'a été demandée.Elle a été faite afin de permettre à l'élève de mieux se familiariser avec la base Base\_Partie 1 qui: regroupe les variables relatives à certains caractéristiques socio-professionnels des dirigeants des différents PME, les coordonnées géographique des PME et la date de soumission des informations de la PME à l'enqueteur. La résolution des questions suivantes se fera essentiellement à l'aide de ces données.

### 1.2 importation et mise en forme

dans section il s'agira de faire quelques opérations sur la base de donnée Base\_partie 1 à savoir l'importation des données et la détection des valeurs manquantes par variable.

```
projet<- read_excel("Base_Partie 1.xlsx",
                    range=NULL,
                    col_names = TRUE,
                    col_types = NULL,
                    ) #la base de donnée étant sous l'extension xlsx,
#la fonction read_excel du package readxl est la mieux adapter pour son importation.
```

#### 1.2.1 importation de la base de donné

**1.2.2 faisons un tableau résumant les valeurs manquantes par valeur** pour le faire nous ferons usage de la fonction `miss_var_summary` du package `nanianr` qui à pour role principal de ressortir un résumé des valeurs manquantes pour chaque variables de la base de donnée. En effet, elle permet de calculer pour chaque variable le nombre total des valeurs manquantes ainsi que la proportion de valeur manquante pour chaque variable.

```
data_miss=miss_var_summary(projet)
kable(head(data_miss))
```

variable	n_miss	pct_miss
q17	131	52.4
q19	120	48.0
q14b	1	0.4
q16	1	0.4
key	0	0.0
q1	0	0.0

**1.2.3 Vérifions s'il y a des valeurs manquantes pour la variable key dans la base projet. Si oui, identifier la (ou les) PME concernée(s).** pour la résolution de cette question nous avons fait usage à une structure conditionnelle qui retournera si la variable key contient ou non des valeurs manquantes et comme après exécution du code key n'a pas de valeur manquantes, aucune PME ne sera identifiée.

```
has_missing <- any(is.na(projet$key))# vérifie si la variable key contient des valeurs NA
# Affichage du résultat
if (has_missing) {
  print("Il y a des valeurs manquantes pour la variable 'key'.")
} else {
```

```
print("Il n'y a pas de valeur manquante pour la variable 'key'.")
}
```

```
## [1] "Il n'y a pas de valeur manquante pour la variable 'key'."
```

### 1.3 Création de variables

**1.3.1 Renomons les variables q1,q2 et q23 respectivement en région,département et sexe** pour cela nous ferons juste usage de la fonction `rename` de `dplyr`

```
projet<- projet %>% dplyr::rename(region= q1, departement=q2, sexe=q23) #"%>% " permet de chaîner des op
```

**1.3.2 Créons la variable sexe\_2 qui vaut 1 si sexe égale à femme et 0 sinon** Pour ceci nous ferons juste usage des fonctions `mutate` et `if_else` du package `dplyr`.

```
projet <- projet %>%
  # Mutate permet l'ajout de la colonne sexe_2 au data frame projet
  mutate(sexe_2 = if_else(sexe=="Femme",1,0))
```

**1.3.3 Créons un data frame nommer langue qui prend la variable key et les variables correspondantes aux langues parlées.** pour le faire nous ferons usage de la fonction `select` de `dplyr`. Comme les variables correspondantes aux langues parlées commence tous par `q24a__`, l'usage de `starts_with` sera très bénéfique car il permettra de sélectionner aisement toutes les variables de la base qui commencent par `q25a__` dont toutes les langues parlées présentes dans la base.

```
langues<-projet %>% dplyr::select(key,starts_with("q24a_"))
```

## ANALYSES DESCRIPTIVES

Déterminons la répartition des PME suivant le sexe, le niveau d'instruction,le statut juridique,le propriétaire/locataire, le statut juridique et le sexe, le niveau d'inscrution et le sexe, propriétaire ou locataire et le sexe.

pour cela nous créerons deux tableaux distinct que nous rélierons par la suite avec la fonction `** **` nous allons reliés les deux tableaux.

```
projet%>%
  dplyr::select(sexe,q25,q12,q81)%>%
  gtsummary::tbl_summary(
    by=sexe,
    label=list(q25~"Niveau d'instruction",
              q12~"statut Juridique",
              q81~"propriétaire/locataire"
    ),
    statistic = list(all_categorical()~ "{n}%"),
    missing="always",
    missing_text="Missing",
    percent="column"
  )%>%
  add_overall()%>%
  bold_labels()
```

```
## Table printed with `knitr::kable()`, not {gt}. Learn why at
## https://www.danielsjoberg.com/gtsummary/articles/rmarkdown.html
## To suppress this message, include `message = FALSE` in code chunk header.
```

Characteristic	Overall, N = 250	Femme, N = 191	Homme, N = 59
<b>Niveau d'instruction</b>			
Aucun niveau	79%	70%	9%
Niveau primaire	56%	48%	8%
Niveau secondaire	74%	56%	18%
Niveau Supérieur	41%	17%	24%
Missing	0	0	0
<b>statut Juridique</b>			
Association	6%	3%	3%
GIE	179%	149%	30%
Informel	38%	32%	6%
SA	7%	1%	6%
SARL	13%	2%	11%
SUARL	7%	4%	3%
Missing	0	0	0
<b>propriétaire/locataire</b>			
Locataire	24%	16%	8%
Propriétaire	226%	175%	51%
Missing	0	0	0

### 1.3 UN PEU DE CARTOGRAPHIE

Dans cette section nous serons appelés à représenter les données sur les cartes plus principalement sur la carte du Sénégal. Notons que nous avons choisi de faire des cartes dynamiques parceque nous vivons dans un monde en parfait évolution de ce fait les données peuvent changer à tous moment raison pour laquelle il est nécessaire de travailler avec les objet dynamic pour facilement les adapter aux différents changements.

#### 1.3.1 Transformons le data.frame en données géographique dont l'objet sera nommé projet\_map

Pour la transformation d'un data frame en des données géographiques nous aurons besoin de la library `sf`

```
projet_map<- st_as_sf(projet, coords = c("gps_menlongitude", "gps_menlatitude"),crs = 4326)
class(projet_map)# projet_map n'est rien d'autre que le data frame projet au quel on a ajouter les coord
```

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

**1.3.2 représentation spatial des PME suivant le sexe** Pour se faire, nous ferons usage de GADM qui est une base de donnée mondiale qui fournit les informations géospaciales sur les subdivisions des différents pays par régions,départements, communes etc...

```
sen_polyg = getData("GADM", country= "senegal", level = 1)
```

```
## Warning in getData("GADM", country = "senegal", level = 1): getData will be removed in a future vers
## . Please use the geodata package instead
```

```
# couleur représentative du genre de chaque chef des PME sur la carte
color_sexe<-colorFactor(c("red","black"),domain=projet_map$sexe)
#création d'une carte interactive
m<-leaflet(sen_polyg) %>% addPolygons(data = sen_polyg,
                                       color = "black",weight = 2, #ajouter un polygone sur la carte

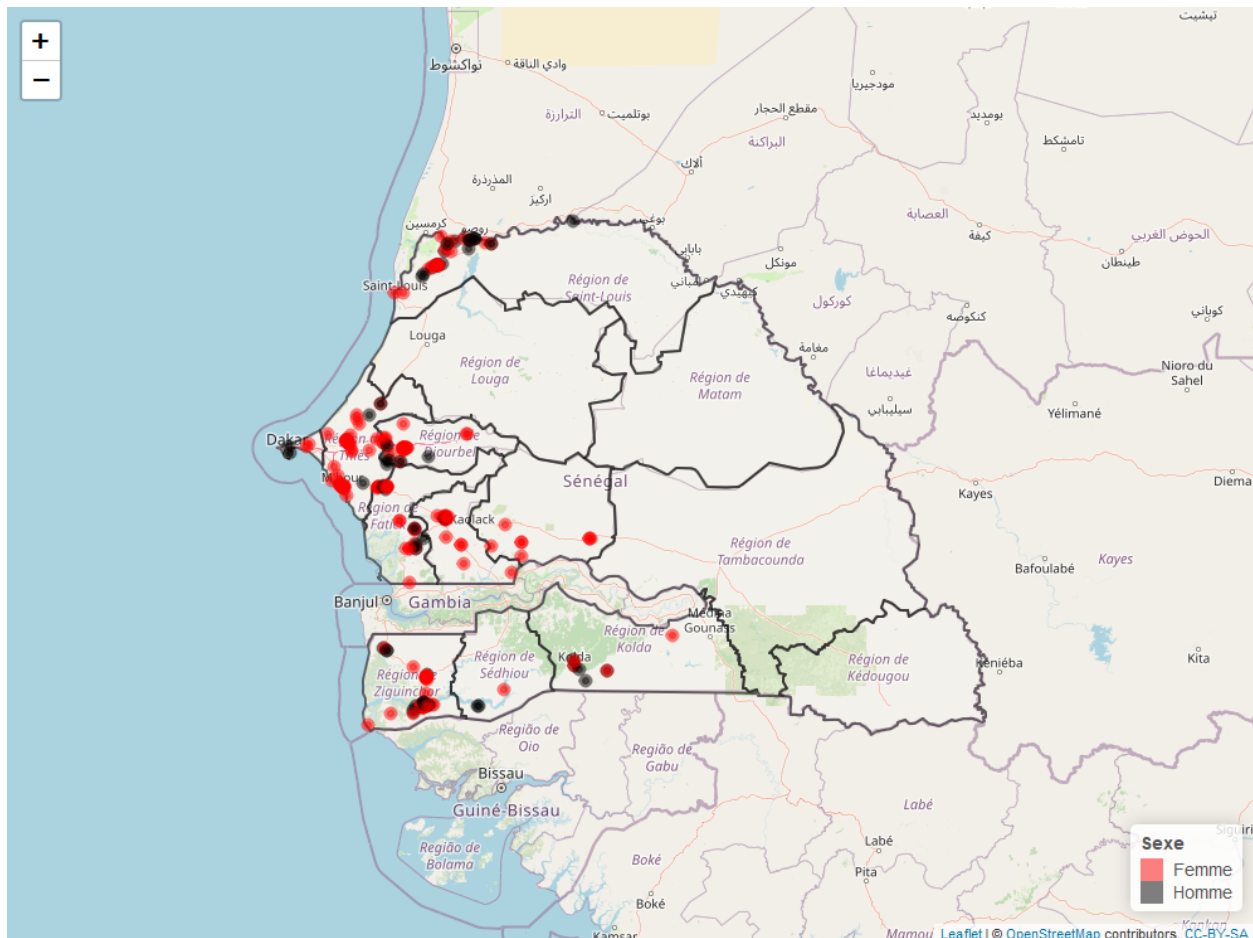
  opacity = 0.5,
  fillColor = "white",
  fillOpacity = 0.1) %>%
  addCircleMarkers(
    data=projet_map,
    radius = 3,
```

```

color = ifelse(projet_map$sexe == "Homme", "black", "red"), # Utilisation de couleurs différentes s
fillOpacity = 0.5,
popup = ifelse(projet_map$sexe == "Homme", "Homme", "Femme")
) %>%
addTiles() %>%
addLegend(position="bottomright",
           pal=color_sexe,
           title="Sexe", values=projet_map$sexe) # ajouter une légende sur la carte

# Etant donné que le graphes est interactif il faut faire la capture d'écran
saveWidget(m, file="Sexe.html")
webshot("Sexe.html", "Sexe.png")

```



**1.3.3 représentation spatial des PME suivant le niveau d'instruction** Ici, nous procédons de la même façon qu'à la question 1.3.2

```

sen_polyg = getData("GADM", country= "senegal", level = 1)

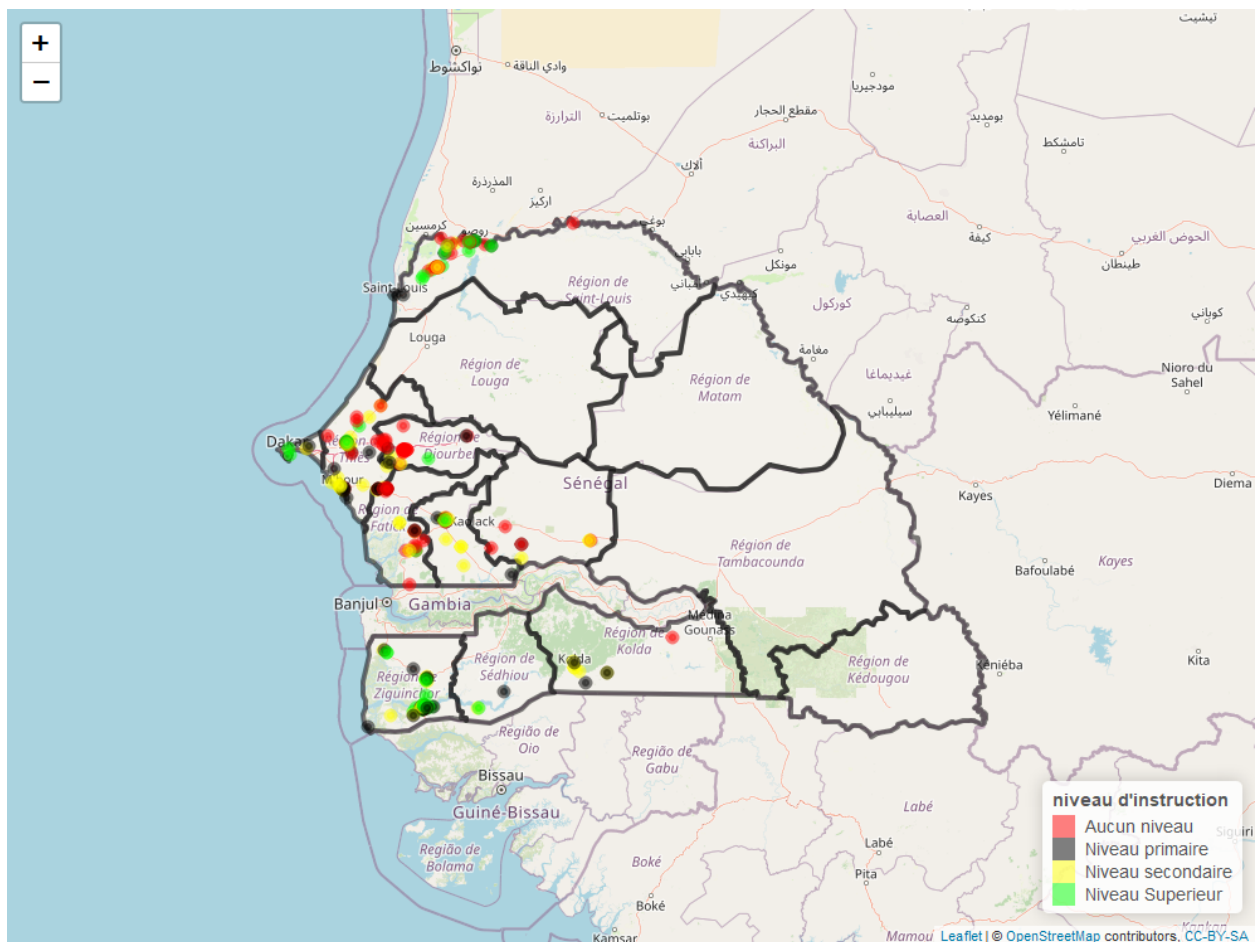
## Warning in getData("GADM", country = "senegal", level = 1): getData will be removed in a future vers
## . Please use the geodata package instead

#Definition d'un jeu de 4 couleurs qui représentera chaque niveau scolaire.
color<-colorFactor(c("red","black","yellow","green"),domain=projet_map$q25)

```

```
#Création d'une carte interactive
m1<-leaflet(sen_polyg) %>% addPolygons(data = sen_polyg,
                                       color = "black",weight = 4,
                                       opacity = 0.5,
                                       fillColor = "white",
                                       fillOpacity = 0.1) %>%
addCircleMarkers(data=projet_map,color=~color(q25),
                 radius = 3,
                 # Utilisation de couleurs différentes selon le sexe
                 fillOpacity = 0.5,
                 ) %>%
addTiles() %>%
addLegend(position="bottomright",
          pal=color,
          title="niveau d'instruction",values=projet_map$q25)# positionnement de la légende à droite

# Etant donné que le graphes est interactif il faut faire la capture d'écran
saveWidget(m1,file="q25.html")
webshot("q25.html", "q25.png")
```



1.3.4 faisons une analyse spatiale de notre choix de répartition des PME suivant qu'ils soient propriétaires ou locaux



```

sen_polyg = getData("GADM", country= "senegal", level = 1)

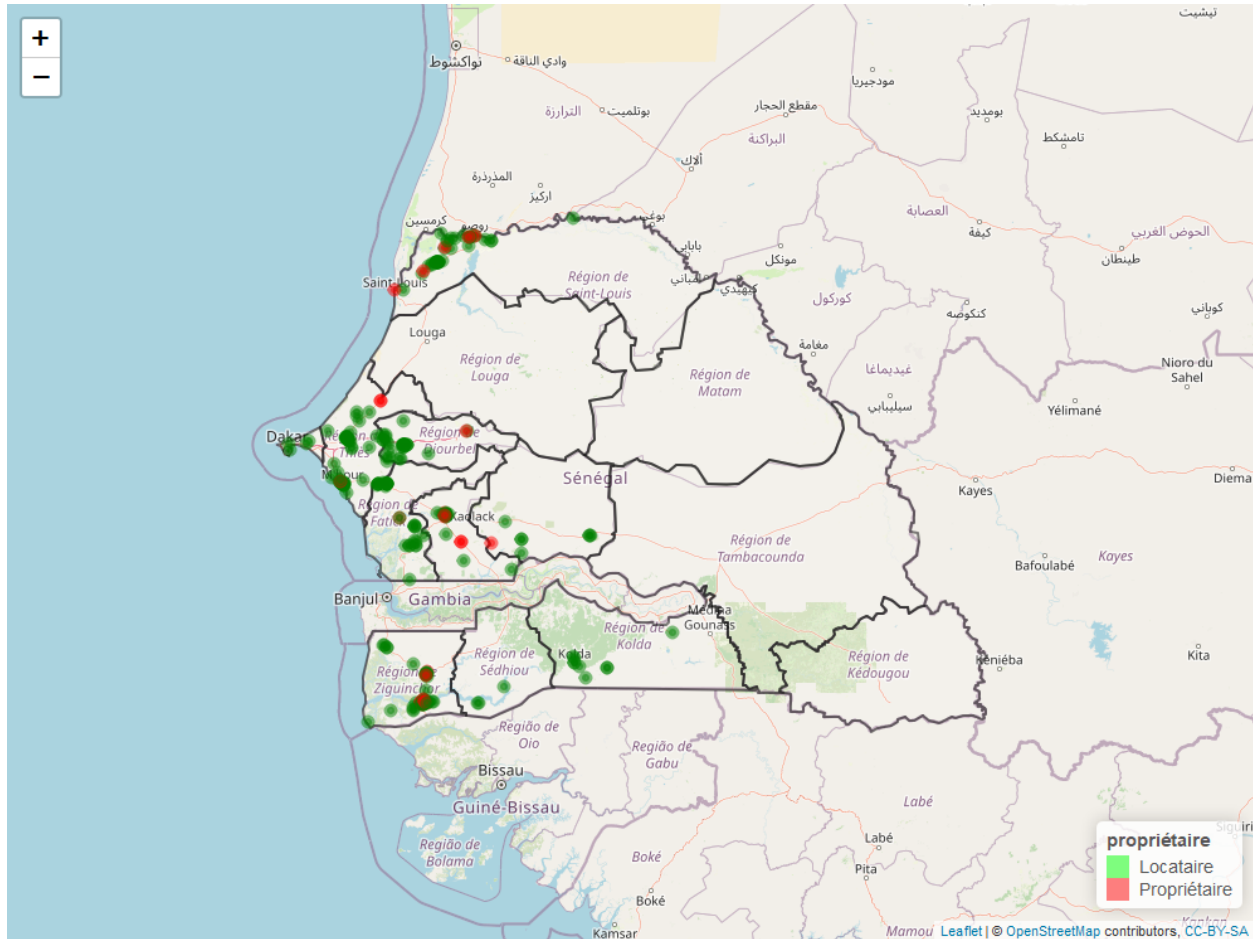
## Warning in getData("GADM", country = "senegal", level = 1): getData will be removed in a future vers
## . Please use the geodata package instead

# couleur représentative du genre de chaque chef des PME sur la carte
color_Pro<-colorFactor(c("green","red"),domain=projet_map$q81)
#création d'une carte interactive
m2<-leaflet(sen_polyg) %>% addPolygons(data = sen_polyg,
                                         color = "black",weight = 2, #ajouter un polygone sur la carte

  opacity = 0.5,
  fillColor = "white",
  fillOpacity = 0.1) %>%
  addCircleMarkers(
    data=projet_map,
    radius = 3,
    color = ifelse(projet_map$q81 == "Propriétaire", "green", "red"), # Utilisation de couleurs différe
    fillOpacity = 0.5,
    popup = ifelse(projet_map$q81 == "Propriétaire", "Propriétaire", "Locataire")
  ) %>%
  addTiles() %>%
  addLegend(position="bottomright",
            pal=color_Pro,
            title="propriétaire",values=projet_map$q81) # ajouter une légende sur la carte

# Etant donné que le graphes est interactif il faut faire la capture d'écran
saveWidget(m2,file="q81.html")
webshot("q81.html","q81.png")

```



## PARTIE 2: NETTOYAGE ET GESTION DES DONNEES importation de la base de

Ici nous ferons usage de la Base\_Partie 2 sur laquelle nous renommerons ou créerons des variables, créerons un nuage de points et effectuerons un ensemble de testes.

### Importation de Base\_partie 2

pour effectuer n'importe quelle opération sur la base il est nécessaire de l'importée.

```
projetN<- read_excel("Base_Partie 2.xlsx",
  range=NULL,
  col_names = TRUE,
  col_types = NULL,
)
```

#### 2.1 Nettoyage et gestion des données

Ici, nous effectuerons certaines tache sur les variables et manipulerons les données manquantes

**2.1.1 Renommons la variable “country\_destination” en “destination” et définissons les valeurs négatives comme manquantes.** Pour exécuter cette tache, nous fairerons usage de la fonction **rename** de **dplyr**.

```

projetN<- projetN %>% dplyr::rename(destination= country_destination)
#filtrage de toutes les valeurs négatives du data frame projet et leur remplacement par les NA supprimées
projetN[projetN < 0] <- NA

```

**2.1.2 Créons une nouvelle variable contenant des tranches d'âge de 5 ans en utilisant la variable "age".** la variable age possédant des données aberrantes, nous allons d'abord chercher à éliminer l'effet de ces valeurs. pour cela, nous procéderons à une imputation par la médiane. Le choix de cette méthode se justifie par le fait que la médiane est moins sensible aux valeurs aberrantes.

```

# calcul du premier et du troisième quartile qui serviront à déterminer l'écart interquartile
q1 <- quantile(projetN$age, 0.25)
q3 <- quantile(projetN$age, 0.75)
#calcul de l'écart interquartile iqr.
iqr <- q3 - q1
# calcul des bornes inférieures et supérieures qui nous serviront à détecter les valeurs aberrantes
min1 <- q1 - 1.5 * iqr
max1 <- q3 + 1.5 * iqr

# créons une nouvelle colonne val_aber pour stocker les valeurs aberrantes remplacées par la médiane
projetN$val_aber <- ifelse((projetN$age < min1) | (projetN$age > max1),
                          median(projetN$age,na.rm =TRUE ),
                          projetN$age)

#créons les bornes pour les tranches d'age
bornes<-seq( min(projetN$val_aber),max(projetN$val_aber),by=5)
# création de la variable "tranche_age" contenant les tranches d'age
projetN <- within(projetN, tranche_age <- cut(val_aber, breaks = bornes))
#afficher la table des tranches d'ages
table(projetN$tranche_age)

##
## (15,20] (20,25] (25,30] (30,35] (35,40]
##      20      37      22      10       7

```

**2.1.3 Créer une nouvelle variable contenant le nombre d'entretiens réalisés par chaque agent recenseur.** Pour cela, nous créerons la table nbre\_entre\_ag contiendra le nombre d'occurrences de chaque valeur unique dans la colonne enumerator. la fonction match ici, recherche les positions des valeurs de projetN\$enumerator dans names(nbre\_entre\_ag) (les noms des valeurs uniques dans la table nbre\_entre\_ag). Elle retourne un vecteur avec les positions correspondantes. pour chaque valeur de enumerator, nbre\_entre\_ag[match(...)] renvoie le nombre d'entretiens correspondant à cette valeur.

```

nbre_entre_ag <- table(projetN$enumerator)
#création d'une nouvelle colonne dans le projetN.
projetN$nbre_entretien<-nbre_entre_ag[match(projetN$enumerator,

```

**2.1.4 Créer une nouvelle variable qui affecte aléatoirement chaque répondant à un groupe de traitement (1) ou de contrôle (0).** Pour cela nous allons d'abord déterminer le nombre de lignes de projetN avec la fonction nrow. ensuite, nous ferons usage de la fonction sample qui générera aléatoirement soit 0, soit 1 à chaque différentes lignes.

```

#Détermination du nombre de ligne du data frame projetN
nombre_total_respondants <- nrow(projetN)
#Génération aléatoire des valeurs 0 et 1 aux différentes lignes.
projetN$alea_affect <- sample(c(1, 0), size = nombre_total_respondants, replace = TRUE,prob=NULL)

```

**2.1.5 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.** La feuille une de Base\_partie 2 étant déjà importée, nous importerons maintenant uniquement la feuille deux de ce fichier. Puis à l'aide de la fonction merge nous aurons le résultat souhaiter.

```
#importation de la feuille 2 de Base_partie 2
projetN2<- read_excel("Base_Partie 2.xlsx",sheet="district")
#Création du data frame fusion qui contient la fusion des deux feuilles.
fusion <- merge(projetN, projetN2, by = "district", all.x = TRUE)
```

**2.1.6 Calculer la durée de l'entretien et indiquer la durée moyenne de l'entretien par enquêteur.** Pour cela nous créerons d'abord une variable qui contient la durée de chaque entretien. Ensuite à l'aide ces durées, nous déterminerons la durée moyenne par enquêteur.

```
projetN$starttime <- as.POSIXct(projetN$starttime , format = "%Y-%m-%d %H:%M")
projetN$endtime <- as.POSIXct(projetN$endtime, format = "%Y-%m-%d %H:%M")

# Calculer la durée de l'entretien (en minutes)
projetN$duree_entretien <- difftime(projetN$endtime, projetN$starttime, units = "mins")

# Calculer la durée moyenne de l'entretien par enquêteur
dur_moy_enq <- aggregate(projetN$duree_entretien,
                          by = list(enqueteur = projetN$enumerator),
                          FUN = mean)

# Renommer les colonnes du résultat
colnames(dur_moy_enq) <- c("enqueteur", "duree_moyenne_entretien")

# Afficher le résultat
print(dur_moy_enq)
```

```
##      enqueteur duree_moyenne_entretien
## 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
```

**2.1.7 Renommez toutes les variables de l'ensemble de données en ajoutant le préfixe “endline\_”**

Ici, nous ferons usage de la fonction paste() qui est utilisée pour créer un nouveau vecteur de noms de colonnes en ajoutant le préfixe “endline\_” à chaque nom de colonne existant. Le paramètre sep = “ ” spécifie que les éléments sont concaténés sans espace entre le préfixe et les noms de colonnes existants.

```
# Renommer les variables en ajoutant le préfixe "endline_"
new_names <- paste("endline_", names(projetN), sep = "")
#attribution au nouvel objet de noms de colonnes, new_names, à l'objet de données projetN.
names(projetN) <- new_names
```

## 2.2 ANALYSE ET VISUALISATION DES DONNEES

Dans cette section nous ferons ressortir quelques tableaux,et nous effectuerons également quelques tests.

**2.2.1 Créez un tableau récapitulatif contenant l'âge moyen et le nombre moyen d'enfants par district.** Comme resultat nous un tableau ayant en colonne les différents discricts et en ligne les variables `endline_val_aber`, `endline_children_num` . les valeurs du tableau contiendront donc l'âge moyen et le nombre moyen d'enfants par discrict. NB: Nous avons travailler avec la variable `endline_val_aber` au lieu de `endline_age` parceque sur `endline_val_aber` nous avons déjà gérer le problème des valeurs abérantes.

```
projetN %>%
  #création du tableau à l'aide de la fonction tbl_summary de gtsummary
  gtsummary::tbl_summary(
    include = c(endline_val_aber,endline_children_num ),
    by =endline_district,
    statistic = all_continuous() ~ " {mean} "
  )%>%
  modify_header(label ~ "**Variables**") %>%
  modify_spanning_header(c("stat_1", "stat_2") ~ "**district**") %>%
  #afficher le titre du tableau
  modify_caption("**Tableau recap age et nombre moyen d'enfant par discrict**") %>%
  # Mettre les variables en gras
  bold_labels()
```

```
## Table printed with `knitr::kable()`, not {gt}. Learn why at
## https://www.danielsjoberg.com/gtsummary/articles/rmarkdown.html
## To suppress this message, include `message = FALSE` in code chunk header.
```

Table 3: Tableau recap age et nombre moyen d'enfant par discrict

Variables	1, N = 8	2, N = 27	3, N = 8	4, N = 5	5, N = 6	6, N = 26	7, N = 6	8, N = 11
<b>endline_val_aber</b>	27.3	26.5	26.1	26.0	24.3	23.2	25.0	24.6
<b>endline_children_num</b>								
0	4 (50%)	17 (63%)	8 (100%)	5 (100%)	4 (67%)	23 (88%)	5 (83%)	7 (64%)
1	1 (13%)	3 (11%)	0 (0%)	0 (0%)	1 (17%)	3 (12%)	1 (17%)	0 (0%)
2	1 (13%)	3 (11%)	0 (0%)	0 (0%)	1 (17%)	0 (0%)	0 (0%)	1 (9.1%)
3	0 (0%)	3 (11%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	2 (18%)
4	1 (13%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)
5	1 (13%)	1 (3.7%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)
6	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	1 (9.1%)

**2.2.2 Testons si la différence d'âge entre les sexes est statistiquement significative au niveau de 5 %.** D'après le test, on obtient une pie-valeur de **0.061** qui est supérieur à 5% donc la différence d'âge entre les sexes n'est pas statistiquement significatif.

```

projetN %>%
  gtsummary::tbl_summary(
    include = endline_val_aber,
    by = endline_sex
  ) %>%
  #indiquons la différence entre les différentes catégories à l'aide de la fonction add_difference()
  add_difference()

```

## Table printed with `knitr::kable()`, not {gt}. Learn why at  
 ## <https://www.danielsjoberg.com/gtsummary/articles/rmarkdown.html>  
 ## To suppress this message, include `message = FALSE` in code chunk header.

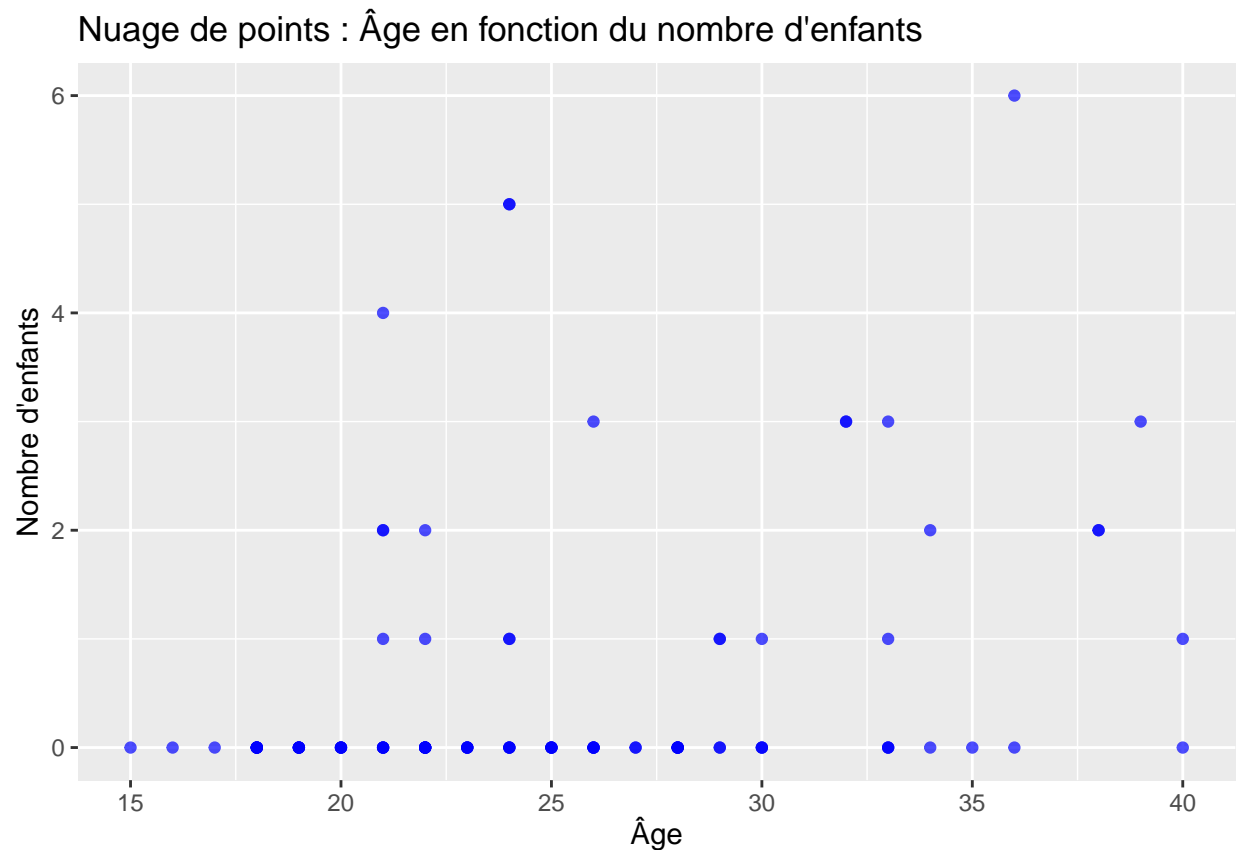
Characteristic	0, N = 86	1, N = 11	Difference	95% CI	p-value
endline_val_aber	24.0 (21.0, 29.0)	21.0 (18.0, 23.0)	3.4	-0.17, 6.9	0.061

**2.2.3 Créer un nuage de points de l'âge en fonction du nombre d'enfants** Pour cette question nous ferons usage de ggplot pour avoir un nuage de point un peu jolie.

```

ggplot(projetN) +
  aes(x = endline_val_aber, y = endline_children_num ) +
  geom_point(color = "blue", alpha = 0.7) +
  labs(x = "Âge", y = "Nombre d'enfants", title = "Nuage de points : Âge en fonction du nombre d'enfants")

```



**2.2.4 La variable “intention”** indiquons si les migrants potentiels ont l'intention de migrer sur une échelle de 1 à 7. Estimons l'effet de l'appartenance au groupe de traitement sur l'intention

**de migrer.** Pour l'estimation, nous allons effectuer un test à l'aide d'un model de regression linéaire simple.

**Interprétation du résultat:** le coefficient beta étant positif, une augmentation de l'appartenance au groupe de traitement entraine une augmentation de l'intention de migrer. La p-Value étant supérieur à **0.9**, elle est élevée ce qui fait comprendre que l'effet de l'appartenance au groupe de migrants potentiel sur l'intension de migrer n'est pas statistiquement significatif et que les différences entre les groupes pourraient être attribuées au hasard. L'intervall de confiance étant de **0.7** nous indique que l'estimation est moins précise.

```
#lm nécessaire pour effectuer de la regression
model <- lm(endline_intention ~endline_alea_affect , data = projetN)
tbl_regression(model) #permet d'afficher le tableaude regression

## Table printed with `knitr::kable()`, not {gt}. Learn why at
## https://www.danielsjoberg.com/gtsummary/articles/rmarkdown.html
## To suppress this message, include `message = FALSE` in code chunk header.
```

Characteristic	Beta	95% CI	p-value
endline_alea_affect	-0.11	-0.81, 0.58	0.7

**2.2.5 Créez un tableau de régression avec 3 modèles. Les résultats des trois modèles doivent être affichés dans un seul tableau.** Etant donné que nous n'avons pas encore les baggages nécessaires pour les interprétations des modèles, dans cette partie nous nous limiterons à la présentation du resultat demandé.

```
#MODEL A:Modèle vide - Effet du traitement sur les intentions
modele_A<-lm(endline_intention ~endline_alea_affect , data = projetN)
mod1<-tbl_regression(modele_A)

#Model B: Effet du traitement sur les intentions en tenant compte de l'âge et du sexe
modele_B<-lm(endline_intention ~endline_alea_affect+endline_val_aber+endline_sex , data = projetN)
mod2<-tbl_regression(modele_B)

#Model C: Identique au modèle B mais en contrôlant le district.
modele_C<-lm(endline_intention ~endline_alea_affect+endline_val_aber+endline_sex+endline_district , data = projetN)
mod3<-tbl_regression(modele_C)

#combinaison des modèles dans un seul tableau
tbl_merge(list(mod1, mod2, mod3))%>% modify_caption("**Tableau de regression final**")

## Table printed with `knitr::kable()`, not {gt}. Learn why at
## https://www.danielsjoberg.com/gtsummary/articles/rmarkdown.html
## To suppress this message, include `message = FALSE` in code chunk header.
```

Table 6: Tableau de regression final

Characteristic	Beta	95% CI	p-value	Beta	95% CI	p-value	Beta	95% CI	p-value
endline_alea_affect	-0.11	-0.81, 0.58	0.7	-0.17	-0.87, 0.54	0.6	-0.21	-0.92, 0.49	0.5
endline_val_aber				0.01	-0.05, 0.07	0.7	0.02	-0.04, 0.08	0.5
endline_sex				-0.91	-2.0, 0.21	0.11	-0.80	-1.9, 0.32	0.2
endline_district							0.10	-0.05, 0.26	0.2

## **FIN PARTIE I et II**

## **PARTIE III**

Pour cette partie voici le lien de l'application

<https://ngatchanadia.shinyapps.io/Documents/>

NB: nous tenons à noter que l'application fonctionne toutefois des soucis de réseau peuvent être un frein à son exécution.