

PROJET DE R

Abdoulaye NDIAYE ISE1-ECO

La date actuelle est 2023-07-24.

Contents

1	PARTIE1	4
1.1	Préparation des données	4
1.1.1	1.1 Description	4
1.1.2	1.2 Importation et mise en forme	4
1.1.3	Sélection les variables mentionnées dans la section description.	5
1.1.4	Faites un tableau qui resume les valeurs manquantes par variable	5
1.1.5	1.3 Création de variables	8
1.1.6	• Créer la variable sexe_2 qui vaut 1 si sexe égale à Femme et 0 sinon.	8
1.1.7	• Créer un data.frame nommé langues qui prend les variables key et les variables correspondantes décrites plus haut.	8
1.1.8	Sélectionnez uniquement les variables key et parle, l'objet de retour sera langues . . .	8
1.2	2 Analyses descriptives	9
1.2.1	Répartition des PME selon le sexe	9
1.2.2	Répartition des PME selon le niveau d'instruction	10
1.2.3	Répartition des PME selon le statut juridique	10
1.2.4	Répartition des PME selon le propriétaire/locataire	11
1.2.5	Représentation graphique en diagramme circulaire	12
1.2.6	Répartition des PME selon le statut juridique et le sexe	12
1.2.7	Répartition des PME selon le niveau d'instruction et le sexe	13
1.2.8	Répartition du propriétaire/locataire selon le sexe	14
1.2.9	TABLEAUX STATISTIQUES	15
2	UN PEU DE CARTOGRAPHIE	19
2.0.1	Représentation spatiale des PME suivant le niveau d'instruction	21
2.0.2	Chargement des données	22

3	chargement des données du senegal gadm	23
3.0.1	Charger les données sur le Sénégal	23
3.0.2	Transformer le data.frame en données géographiques	23
4	PARTIE 2	25
4.1	Nettoyage et gestion de données	25
5	PARTIE3	33

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



Utilisation de l'application R

Rédigé par :

Abdoulaye NDIAYE

Élève ingénieur statisticien économiste

Sous la supervision de :

HEMA Aboubacar

Data Analyst

© **Juillet-2023**

Dans le cadre du cours du Cours de Projet Statistique sous R, il nous a été demandé de réalié un travaillé pour appliqué sur le logiciel R les différentes notions vues en classe Pour cela, une base de données a été mis à notre disposition.

#L'objectif de l'enquête: Cette enquête vise à identifier et à caractériser des bioénergies durables pour les petites et moyennes entreprises (PME) agroalimentaires d'Afrique de l'Ouest.

1 PARTIE1

1.1 Préparation des données

1.1.1 1.1 Description

1.1.2 1.2 Importation et mise en forme

```
library(readxl)
```

```
## Warning: le package 'readxl' a été compilé avec la version R 4.1.3
```

```
library(dplyr)
```

```
## Warning: le package 'dplyr' a été compilé avec la version R 4.1.3
```

```
##
```

```
## Attachement du package : 'dplyr'
```

```
## Les objets suivants sont masqués depuis 'package:stats':
```

```
##
```

```
##      filter, lag
```

```
## Les objets suivants sont masqués depuis 'package:base':
```

```
##
```

```
##      intersect, setdiff, setequal, union
```

```
library(kableExtra)
```

```
## Warning: le package 'kableExtra' a été compilé avec la version R 4.1.3
```

```
##
```

```
## Attachement du package : 'kableExtra'
```

```
## L'objet suivant est masqué depuis 'package:dplyr':
```

```
##
```

```
##      group_rows
```

```
projet<- read_excel("Base_Partie 1.xlsx") ##importation de la base de données  
print(projet)##Afficher la base de données
```

```
## # A tibble: 250 x 33
##   key          q1    q2    q23    q24 q24a_1 q24a_2 q24a_3 q24a_4 q24a_5 q24a_6
##   <chr>      <chr> <chr> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 uuid:68bff~ Diou~ Bamb~ Femme  65     0     1     0     1     0     0
## 2 uuid:d70b3~ Thiès Mbour Femme  52     1     1     0     0     1     0
## 3 uuid:0ac18~ Thiès Mbour Femme  65     1     1     0     0     0     0
## 4 uuid:c52cf~ Thiès Mbour Femme  38     1     1     0     0     1     0
## 5 uuid:ac177~ Zigu~ Bign~ Homme  40     1     1     1     0     0     1
## 6 uuid:57809~ Zigu~ Ouss~ Femme  43     1     1     1     0     0     0
## 7 uuid:c3065~ Thiès Thiès Femme  53     0     1     0     1     0     0
## 8 uuid:74e60~ Zigu~ Zigu~ Homme  33     1     0     0     0     0     0
## 9 uuid:2ee01~ Diou~ Bamb~ Femme  67     0     1     0     1     0     0
## 10 uuid:5c801~ Sain~ Daga~ Homme  35     1     1     0     0     0     0
## # i 240 more rows
## # i 22 more variables: q24a_7 <dbl>, q24a_9 <dbl>, q24a_10 <dbl>, q25 <chr>,
## #   q26 <dbl>, q12 <chr>, q14b <chr>, q16 <chr>, q17 <chr>, q19 <chr>,
## #   q20 <chr>, filiere_1 <dbl>, filiere_2 <dbl>, filiere_3 <dbl>,
## #   filiere_4 <dbl>, q8 <chr>, q81 <chr>, gps_menlatitude <dbl>,
## #   gps_menlongitude <dbl>, submissiondate <dtm>, start <dtm>, today <dtm>
```

1.1.3 Selection les variables mentionnees dans la section description.

```
library(dplyr)
variables_manquantes <- projet %>%
  select_if(function(row) any(is.na(row))) %>%
  colnames()
print(variables_manquantes)
```

```
## [1] "q14b" "q16" "q17" "q19"
```

1.1.4 Faites un tableau qui resume les valeurs manquantes par variable

```
library(knitr)
library(dplyr)
library(kableExtra)
table_valeurs_manquantes <- colSums(is.na(projet))
table_valeurs_manquantes %>% kable(format = "latex",caption = " Tableau des valeurs manquantes")
```

```
print(table_valeurs_manquantes)
```

```
##           key          q1          q2          q23
##           0           0           0           0
##          q24          q24a_1          q24a_2          q24a_3
##           0           0           0           0
##          q24a_4          q24a_5          q24a_6          q24a_7
##           0           0           0           0
##          q24a_9          q24a_10          q25          q26
##           0           0           0           0
```

Table 1: Tableau des valeurs manquantes

	x
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
q25	0
q26	0
q12	0
q14b	1
q16	1
q17	131
q19	120
q20	0
filiere_1	0
filiere_2	0
filiere_3	0
filiere_4	0
q8	0
q81	0
gps_menlatitude	0
gps_menlongitude	0
submissiondate	0
start	0
today	0

```
##          q12          q14b          q16          q17
##          0          1          1          131
##          q19          q20          filiere_1          filiere_2
##          120          0          0          0
##          filiere_3          filiere_4          q8          q81
##          0          0          0          0
##  gps_menlatitude gps_menlongitude  submissiondate          start
##          0          0          0          0
##          today
##          0
```

```
kable(table_valeurs_manquantes)
```

	x
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
q25	0
q26	0
q12	0
q14b	1
q16	1
q17	131
q19	120
q20	0
filiere_1	0
filiere_2	0
filiere_3	0
filiere_4	0
q8	0
q81	0
gps_menlatitude	0
gps_menlongitude	0
submissiondate	0
start	0
today	0

• 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).

1.1.5 1.3 Création de variables

```
names(projet)[names(projet) == "q1"] <- "region"
names(projet)[names(projet) == "q2"] <- "departement"
names(projet)[names(projet) == "q23"] <- "sexe"
```

1.1.6 • Créer la variable `sexe_2` qui vaut 1 si `sexe` égale à `Femme` et 0 sinon.

```
projet$sexe_2 <- ifelse(projet$sexe == "Femme", 1, 0)## La fonction ifelse vérifie si sexe est égale à .
```

1.1.7 • Créer un `data.frame` nommé `langues` qui prend les variables `key` et les variables correspondantes décrites plus haut.

```
variables_langues <- grep("^q24a_", names(projet), value = TRUE)
langues <- projet[c("key", variables_langues)]
```

• Créer une variable `parle` qui est égale au nombre de langue parlée par le dirigeant de la PME.

```
library(dplyr)
langues$parle <- rowSums(!is.na(langues[, variables_langues]))
```

1.1.8 Sélectionnez uniquement les variables `key` et `parle`, l'objet de retour sera `langues`

```
library(dplyr)
langues <- langues[, c("key", "parle")]
print(langues)
```

```
## # A tibble: 250 x 2
##   key                               parle
##   <chr>                           <dbl>
## 1 uuid:68bff42b-1228-4c66-9bcc-e6d312d9fea6      9
## 2 uuid:d70b3c7e-3ca0-4358-bc59-3f7f6baf55e9      9
## 3 uuid:0ac18b64-7d85-4bb9-a842-698ac79909af      9
## 4 uuid:c52cf5e4-8c28-4e65-998b-3fe2a971a1a3      9
## 5 uuid:ac177870-001c-4ada-8747-c22ffe4e4596      9
## 6 uuid:578097cf-9af7-46e6-8992-d9079b14c342      9
## 7 uuid:c3065ed7-c16a-4578-b8fa-e09ffc7bced7      9
## 8 uuid:74e608ca-a37e-47e3-8b22-e1e4bd51fa44      9
## 9 uuid:2ee01315-87e2-4b4a-a19b-107f77bdfc78      9
## 10 uuid:5c801b15-e8fd-4029-af6d-a79efa016520      9
## # i 240 more rows
```

• Merger les `data.frame` `projet` et `langues`:


```
tbl_merge <- merge(projet, langues, by = "key")
dim(tbl_merge )
```

```
## [1] 250 35
```

1.2 2 Analyses descriptives

1.2.1 Répartition des PME selon le sexe

```
library(kableExtra)
repartition_sexe <- table(projet$sexe)
print("Répartition des PME selon le sexe :")
```

```
## [1] "Répartition des PME selon le sexe :"
```

```
print(repartition_sexe)
```

```
##
## Femme Homme
## 191 59
```

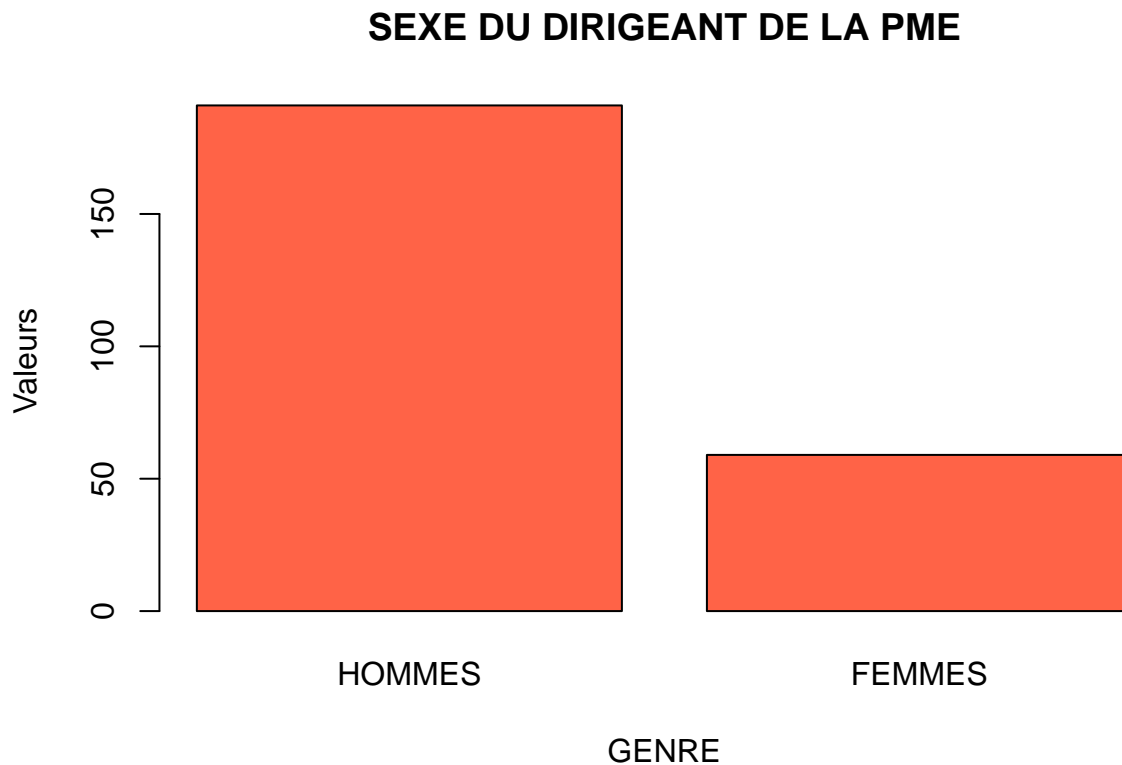
```
kable(repartition_sexe)
```

Var1	Freq
Femme	191
Homme	59

```
# Créer un vecteur de données
repartition_sexe<- c(191, 59)
```

```
# Créer un vecteur d'étiquettes pour les barres
labels <- c("HOMMES", "FEMMES")
```

```
barplot(repartition_sexe, names.arg = labels, col = "tomato", main = "SEXE DU DIRIGEANT DE LA PME", xlab = "SEXE")
```



1.2.2 Répartition des PME selon le niveau d'instruction

```
table_niveau_instruction <- table(projet$q25)
print("Répartition des PME selon le niveau d'instruction:")
```

```
## [1] "Répartition des PME selon le niveau d'instruction:"
```

```
print(table_niveau_instruction)
```

```
##
##      Aucun niveau  Niveau primaire Niveau secondaire  Niveau Superieur
##              79             56              74             41
```

1.2.3 Répartition des PME selon le statut juridique

```
table_statut_juridique <- table(projet$q12)
print("Répartition des PME selon le statut juridique:")
```

```
## [1] "Répartition des PME selon le statut juridique:"
```

```
print(table_statut_juridique)
```

```
##
## Association      GIE      Informel      SA      SARL      SUARL
##           6      179      38      7      13      7
```

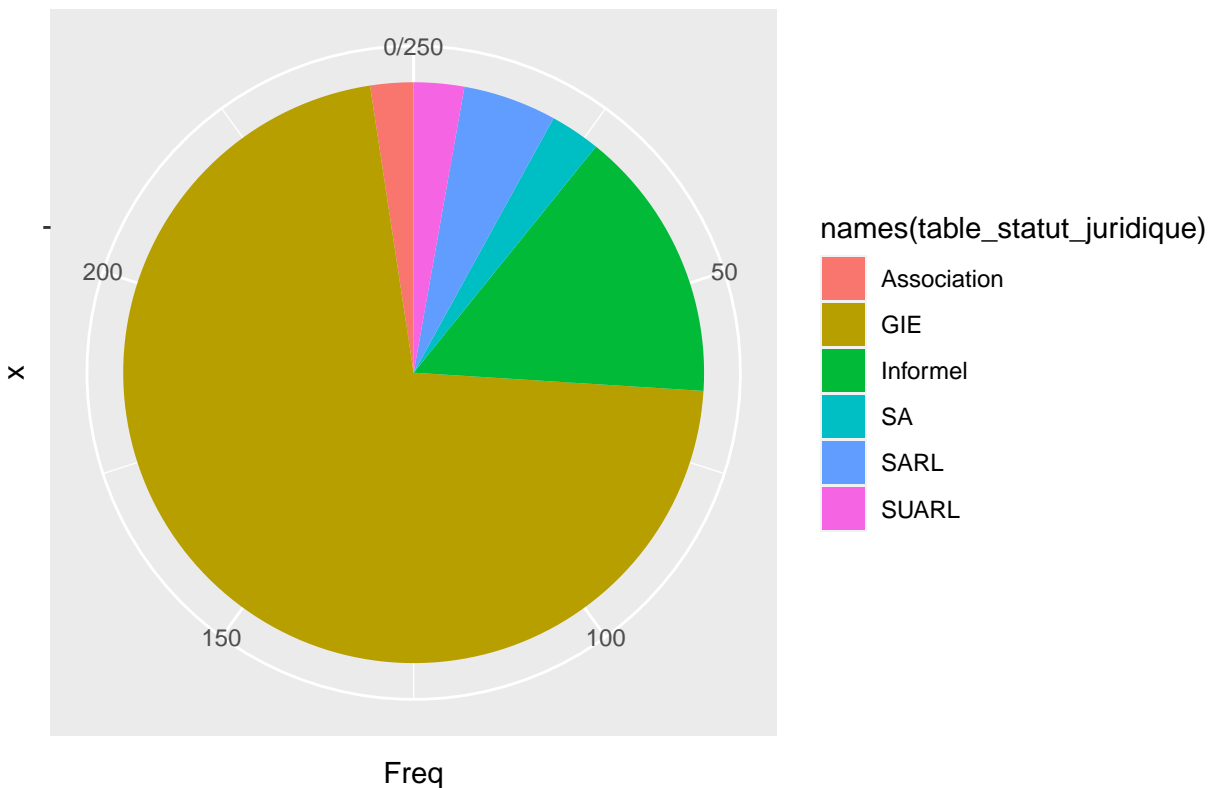
```
library(ggplot2)
```

```
## Warning: le package 'ggplot2' a été compilé avec la version R 4.1.3
```

```
p <- ggplot(data = as.data.frame(table_statut_juridique), aes(x = "", y = Freq, fill = names(table_statut_juridique))) +
  geom_bar(stat = "identity") +
  coord_polar("y", start = 0) +
  labs(title = "Répartition des PME selon le propriétaire/locataire")

# Affichez le diagramme
print(p)
```

Répartition des PME selon le propriétaire/locataire



1.2.4 Répartition des PME selon le propriétaire/locataire

```
table_proprietaire_locataire <- table(projet$q81)
print("Répartition des PME selon le propriétaire/locataire:")
```

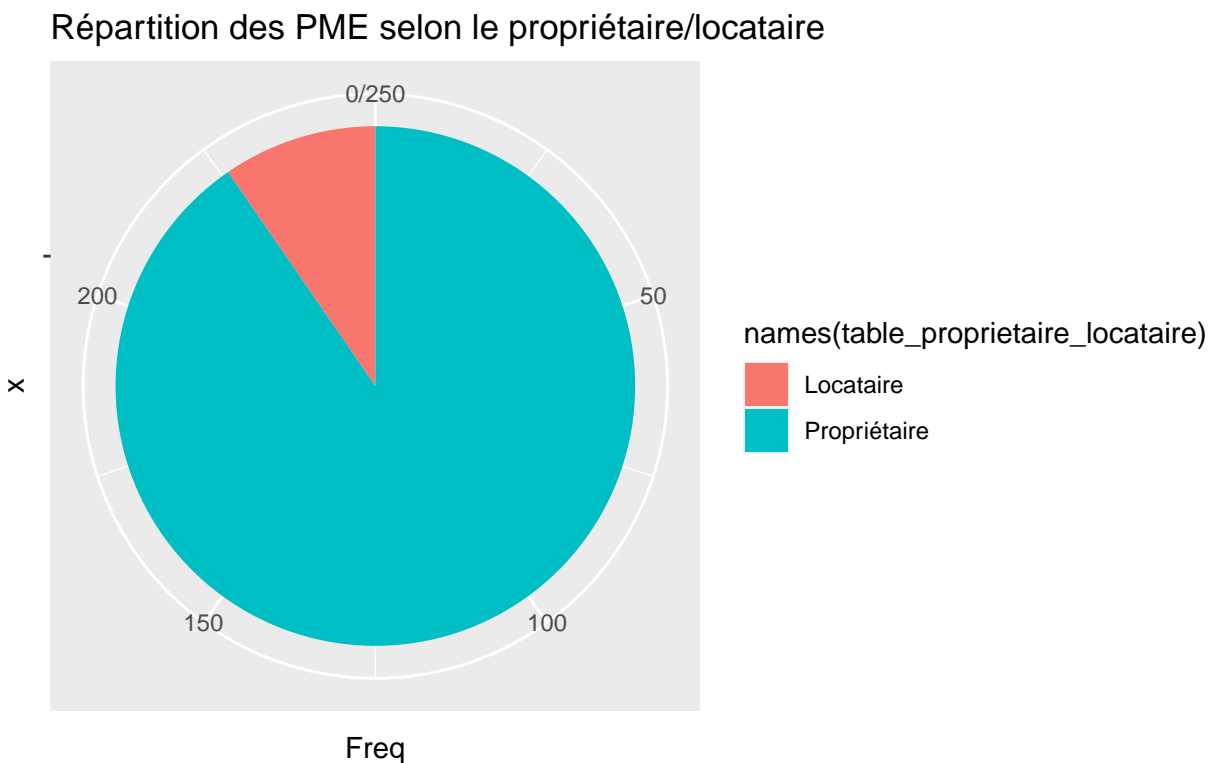
```
## [1] "Répartition des PME selon le propriétaire/locataire:"
```

```
print(table_proprietaire_locataire)
```

```
##  
##   Locataire Propriétaire  
##         24         226
```

1.2.5 Réprésentation graphique en diagramme circulaire

```
library(ggplot2)  
p <- ggplot(data = as.data.frame(table_proprietaire_locataire), aes(x = "", y = Freq, fill = names(table_proprietaire_locataire))) +  
  geom_bar(stat = "identity") +  
  coord_polar("y", start = 0) +  
  labs(title = "Répartition des PME selon le propriétaire/locataire")  
  
# Affichez le diagramme  
print(p)
```



1.2.6 Répartition des PME selon le statut juridique et le sexe

#Nous allons d'abord labéliser les variables q12 et q23

```
names(projet)[names(projet) == "q23"] <- "sexe"
names(projet)[names(projet) == "q12"] <- "Statut juridique"
```

```
library(dplyr)
library(tidyr)
```

```
## Warning: le package 'tidyr' a été compilé avec la version R 4.1.3
```

```
library(gtsummary)
table1<-projet%>%
  tbl_cross(
    row = `Statut juridique`,
    col = sexe,
    percent = "row"
  ) %>%
  add_p(source_note = TRUE)
```

1.2.7 Répartition des PME selon le niveau d'instruction et le sexe

```
names(projet)[names(projet) == "q23"] <- "sexe"
names(projet)[names(projet) == "q25"] <- "niveau instruction"
```

```
library(dplyr)
library(tidyr)
library(gtsummary)
table2<- projet%>%
  tbl_cross(
    col = `niveau instruction`,
    row = sexe,
    percent = "row"
  ) %>%
  add_p(source_note = TRUE)
```

```
library(flextable)
```

```
##
## Attachement du package : 'flextable'

## Les objets suivants sont masqués depuis 'package:gtsummary':
##
##   as_flextable, continuous_summary

## Les objets suivants sont masqués depuis 'package:kableExtra':
##
##   as_image, footnote
```

```
tbl_merge(list(table2,table1))
```

```
## 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.
```

	Aucun niveau	Niveau primaire	Niveau secondaire	Niveau Supérieur	Total	Femme
sexe						
Femme	70 (37%)	48 (25%)	56 (29%)	17 (8.9%)	191 (100%)	
Homme	9 (15%)	8 (14%)	18 (31%)	24 (41%)	59 (100%)	
Total	79 (32%)	56 (22%)	74 (30%)	41 (16%)	250 (100%)	191 (76%)
Statut juridique						
Association						3 (50%)
GIE						149 (83%)
Informel						32 (84%)
SA						1 (14%)
SARL						2 (15%)
SUARL						4 (57%)

1.2.8 Répartition du propriétaire/locataire selon le sexe

```
names(projet)[names(projet) == "q23"] <- "sexe"
names(projet)[names(projet) == "q81"] <- "proprietaire ou locataire"
```

```
library(dplyr)
library(tidyr)
library(gtsummary)
projet%>%
  tbl_cross(
    col = 'proprietaire ou locataire',
    row = `sexe`,
    percent = "col"
  ) %>%
  add_p(source_note = TRUE)
```

```
## 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.
```

	Locataire	Propriétaire	Total
sexe			
Femme	16 (67%)	175 (77%)	191 (76%)
Homme	8 (33%)	51 (23%)	59 (24%)
Total	24 (100%)	226 (100%)	250 (100%)

```
my_theme <-
  list(
    # round large p-values to two places
    "pkgwide-fn:pvalue_fun" = function(x) style_pvalue(x, digits = 2),
```

```

"pkgwide-fn:prependpvalue_fun" = function(x) style_pvalue(x, digits = 2, prepend_p = TRUE),
# report median (IQR) and n (percent) as default stats in `tbl_summary()`
"tbl_summary-str:continuous_stat" = "{median} ({p25} - {p75})",
"tbl_summary-str:categorical_stat" = "{n} ({p})"
)

set_gtsummary_theme(my_theme) #créer et appliquer le thème

```

1.2.9 TABLEAUX STATISTIQUES

#TABLE ARACHIDE

```

projet<- read_excel("Base_Partie 1.xlsx") ##importation de la base de données
names(projet)[names(projet) == "q23"] <- "sexe"
names(projet)[names(projet) == "q81"] <- "proprietaire_locataire"
names(projet)[names(projet) == "q25"] <- "niveau_instruction"
names(projet)[names(projet) == "q12"] <- "statut_juridique"

projet = dplyr::rename(projet, arachide= filiere_1,
                        anacarde= filiere_2,

                        mangue= filiere_3, riz= filiere_4)

#View(projet)
#(projet)
## filière arachide
B_arachide=projet[projet$arachide==1,]
#(B_arachide)
tbl_arachide <- B_arachide %>%
  tbl_summary(include = c(sexe,
                          proprietaire_locataire,
                          niveau_instruction,
                          statut_juridique ),
              #label=list(Stat_juridique ~ "q12",
                          # prop_loca~ "q81",
                          # niv_instruction~"q25"),
              by=sexe,
              # statistic = list(all_continuous()~ "{mean}", all_categorical() ~ "{p} %"),
              #type=list(sexe="categorical", prop_loca="categorical"),
              digits = list(all_continuous() ~ 1,
                            all_categorical() ~ c(0, 1)))%>%
  bold_labels() %>%
  italicize_levels()
# %>%
# modify_header(list(label ~ "**Les Variables**"))
tbl_arachide

```

```

## 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	**Femme**, N = 93	**Homme**, N = 15
__propietaire_locataire__		
__Locataire__	9 (9.7)	3 (20.0)
__Propriétaire__	84 (90.3)	12 (80.0)
__niveau_instruction__		
__Aucun niveau__	38 (40.9)	5 (33.3)
__Niveau primaire__	20 (21.5)	3 (20.0)
__Niveau secondaire__	30 (32.3)	4 (26.7)
__Niveau Supérieur__	5 (5.4)	3 (20.0)
__statut_juridique__		
__Association__	2 (2.2)	0 (0.0)
__GIE__	70 (75.3)	9 (60.0)
__Informel__	20 (21.5)	3 (20.0)
__SA__	0 (0.0)	2 (13.3)
__SARL__	0 (0.0)	1 (6.7)
__SUARL__	1 (1.1)	0 (0.0)

#TABLE FILIERE ANACARDE

```

projet<- read_excel("Base_Partie 1.xlsx") ##importation de la base de données
names(projet)[names(projet) == "q23"] <- "sexe"
names(projet)[names(projet) == "q81"] <- "propietaire_locataire"
names(projet)[names(projet) == "q25"] <- "niveau_instruction"
names(projet)[names(projet) == "q12"] <- "statut_juridique"

projet = dplyr::rename(projet,arachide= filiere_1,
                        anacarde=   filiere_2,

                        mangue= filiere_3,riz=   filiere_4)

#View(projet)
#(projet)
## filière anacarde
B_anacarde=projet[projet$anacarde==1,]
#(B_arachide)
tbl_anacarde <- B_anacarde %>%
  tbl_summary(include = c(sexe,
                          proprietaire_locataire,
                          niveau_instruction,
                          statut_juridique ),
              #label=list(Stat_juridique ~ "q12",
                          # prop_loca~ "q81",
                          # niv_instruction~"q25"),
              by=propietaire_locataire,
              # statistic = list(all_continuous()~ "{mean}",all_categorical() ~ "{p} %"),
              #type=list(sexe="categorical",prop_loca="categorical"),
              digits = list(all_continuous() ~ 1,
                            all_categorical() ~ c(0, 1)))%>%

  bold_labels() %>%
  italicize_levels()
# %>%
# modify_header(list(label ~ "**Les Variables**"))
tbl_anacarde

```

Table printed with 'knitr::kable()', not {gt}. Learn why at

<https://www.danieldsjoberg.com/gtsummary/articles/rmarkdown.html>
 ## To suppress this message, include 'message = FALSE' in code chunk header.

Characteristic	Locataire , N = 7	Propriétaire , N = 54
sexe		
Femme	3 (42.9)	37 (68.5)
Homme	4 (57.1)	17 (31.5)
niveau_instruction		
Aucun niveau	0 (0.0)	13 (24.1)
Niveau primaire	1 (14.3)	16 (29.6)
Niveau secondaire	2 (28.6)	13 (24.1)
Niveau Supérieur	4 (57.1)	12 (22.2)
statut_juridique		
Association	0 (0.0)	3 (5.6)
GIE	2 (28.6)	33 (61.1)
Informel	0 (0.0)	12 (22.2)
SA	1 (14.3)	1 (1.9)
SARL	2 (28.6)	4 (7.4)
SUARL	2 (28.6)	1 (1.9)

#TABLE FILIERE MANGUE

```

projet<- read_excel("Base_Partie 1.xlsx") ##importation de la base de données
names(projet)[names(projet) == "q23"] <- "sexe"
names(projet)[names(projet) == "q81"] <- "proprietaire_locataire"
names(projet)[names(projet) == "q25"] <- "niveau_instruction"
names(projet)[names(projet) == "q12"] <- "statut_juridique"

projet = dplyr::rename(projet,arachide= filiere_1,
                        anacarde=   filiere_2,

                        mangue= filiere_3,riz=   filiere_4)

#View(projet)
#(projet)
## filiÃ©re mangue
B_mangue=projet[projet$mangue==1,]
#(B_mangue)
tbl_mangue <- B_mangue %>%
  tbl_summary(include = c(sexe,
                          proprietaire_locataire,
                          niveau_instruction,
                          statut_juridique ),
              #label=list(Stat_juridique ~ "q12",
                          # prop_loca~ "q81",
                          # niv_instruction~"q25"),
              by=niveau_instruction,
              # statistic = list(all_continuous()~ "{mean}",all_categorical() ~ "{p} %"),
              #type=list(sexe="categorical",prop_loca="categorical"),
              digits = list(all_continuous() ~ 1,
                           all_categorical() ~ c(0, 1)))%>%

  bold_labels() %>%
  italicize_levels()
# %>%

```

```
# modify_header(list(label ~ "**Les Variables**"))
tbl_mangue
```

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	**Aucun niveau**, N = 26	**Niveau Supérieur**, N = 14	**Niveau primaire**, N = 2
__sexe__			
__Femme__	22 (84.6)	5 (35.7)	20 (83.3)
__Homme__	4 (15.4)	9 (64.3)	4 (16.7)
__proprietaire_locataire__			
__Locataire__	2 (7.7)	5 (35.7)	1 (4.2)
__Propriétaire__	24 (92.3)	9 (64.3)	23 (95.8)
__statut_juridique__			
__GIE__	21 (80.8)	7 (50.0)	22 (91.7)
__Informel__	2 (7.7)	0 (0.0)	1 (4.2)
__SA__	0 (0.0)	3 (21.4)	0 (0.0)
__SARL__	3 (11.5)	3 (21.4)	0 (0.0)
__SUARL__	0 (0.0)	1 (7.1)	1 (4.2)

#TABLE FILIERE RIZ

```
projet<- read_excel("Base_Partie 1.xlsx") ##importation de la base de données
names(projet)[names(projet) == "q23"] <- "sexe"
names(projet)[names(projet) == "q81"] <- "proprietaire_locataire"
names(projet)[names(projet) == "q25"] <- "niveau_instruction"
names(projet)[names(projet) == "q12"] <- "statut_juridique"

projet = dplyr::rename(projet,arachide= filiere_1,
                      anacarde=   filiere_2,

                      mangue= filiere_3,riz=   filiere_4)

#View(projet)
#(projet)
## filière riz
B_riz=projet[projet$riz==1,]
#(B_riz)
tbl_riz <- B_riz %>%
  tbl_summary(include = c(sexe,
                          proprietaire_locataire,
                          niveau_instruction,
                          statut_juridique ),
              #label=list(Stat_juridique ~ "q12",
                          # prop_loca~ "q81",
                          # niv_instruction~"q25"),
              by=statut_juridique,
              # statistic = list(all_continuous()~ "{mean}",all_categorical() ~ "{p} %"),
              #type=list(sexe="categorical",prop_loca="categorical"),
              digits = list(all_continuous() ~ 1,
                            all_categorical() ~ c(0, 1)))%>%

  bold_labels() %>%
  italicize_levels()
```

```
# %>%
# modify_header(list(label ~ "**Les Variables**"))
tbl_riz
```

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

Characteristic	**Association**, N = 2	**GIE**, N = 77	**Informel**, N = 3	**SA**, N = 3	**S
__sexe__					
__Femme__	0 (0.0)	73 (94.8)	1 (33.3)	0 (0.0)	
__Homme__	2 (100.0)	4 (5.2)	2 (66.7)	3 (100.0)	
__proprietaire_locataire__					
__Locataire__	0 (0.0)	7 (9.1)	0 (0.0)	1 (33.3)	
__Propriétaire__	2 (100.0)	70 (90.9)	3 (100.0)	2 (66.7)	
__niveau_instruction__					
__Aucun niveau__	0 (0.0)	10 (13.0)	1 (33.3)	0 (0.0)	
__Niveau primaire__	0 (0.0)	24 (31.2)	0 (0.0)	0 (0.0)	
__Niveau secondaire__	0 (0.0)	29 (37.7)	2 (66.7)	0 (0.0)	
__Niveau Supérieur__	2 (100.0)	14 (18.2)	0 (0.0)	3 (100.0)	

```
#TABLEAUX MERGER
```

2 UN PEU DE CARTOGRAPHIE

```
#Chargement des packages nécessaires
```

```
library(dplyr)
library(tidyr)
library(gtsummary)
library(lubridate)
```

```
## Warning: le package 'lubridate' a été compilé avec la version R 4.1.3
```

```
##
## Attachement du package : 'lubridate'
```

```
## Les objets suivants sont masqués depuis 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(kableExtra)
library(sf)
```

```
## Warning: le package 'sf' a été compilé avec la version R 4.1.3
```

```
## Linking to GEOS 3.10.2, GDAL 3.4.1, PROJ 7.2.1; sf_use_s2() is TRUE
```

```
library(ggplot2)
library(sf)
```

#Représentation spatiale des PME suivant le sexe

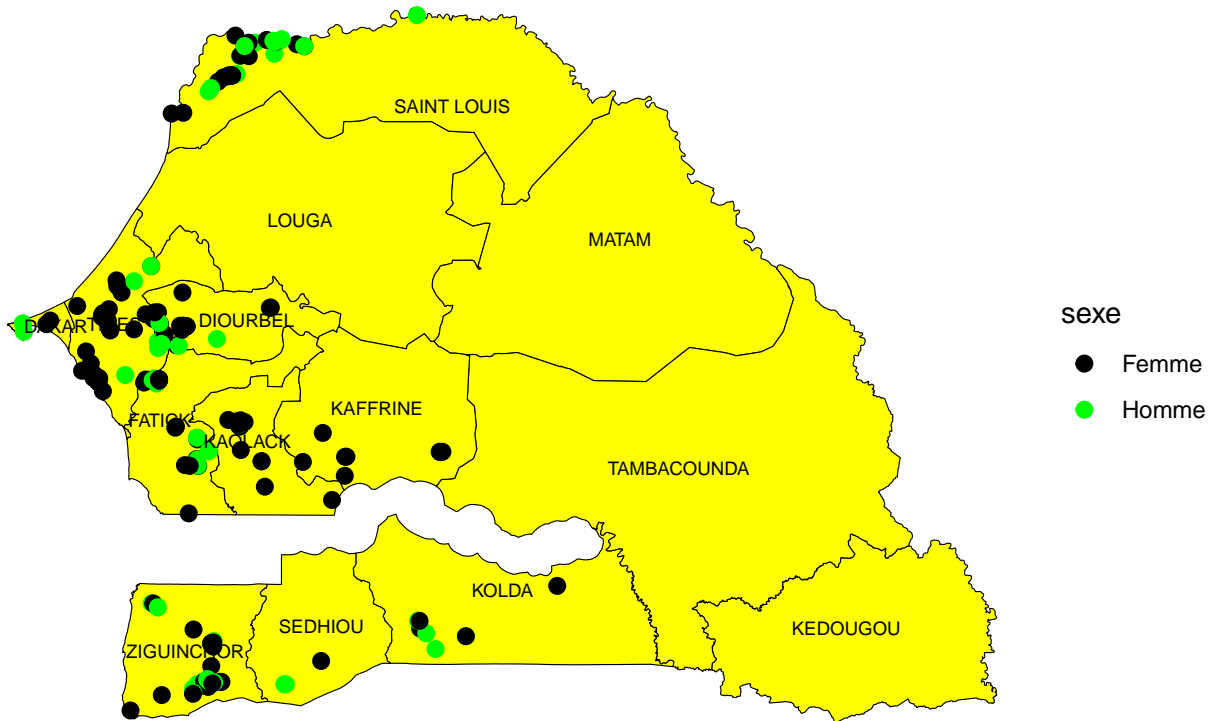
```
projet_map<-st_as_sf(projet,coords = c("gps_menlongitude","gps_menlatitude"),crs=4326)

# contours
sen_contours <- st_read("Sen/Limite_R gion.shp")
```

```
## Reading layer 'Limite_R gion' from data source
##   'C:\Users\DELL\OneDrive\Bureau\PROJET 24 JUILLET 2023\Sen\Limite_R gion.shp'
##   using driver 'ESRI Shapefile'
## Simple feature collection with 14 features and 4 fields
## Geometry type: POLYGON
## Dimension:      XY
## Bounding box:   xmin: 227586.3 ymin: 1362012 xmax: 897104.7 ymax: 1845672
## Projected CRS: WGS 84 / UTM zone 28N
```

```
names(sen_contours)[1] <-"region"
ggplot()+
  geom_sf(data=sen_contours,fill="yellow",color="black")+
  geom_sf(data=projet_map,aes(color=sexe),size=2.5)+
  geom_sf_text(data=sen_contours,aes(label=region),size=2.5)+
  scale_color_manual(values = c("black", "green")) +
  theme_void()+
  theme(legend.position = "right")+
  labs(title="carte des PME par sexe",color="sexe")
```

carte des PME par sexe



```
#title("Carte du Sénégal avec les régions")
```

```
names(projet)[names(projet) == "q25"] <- "niveau d'instruction"
```

2.0.1 Représentation spatiale des PME suivant le niveau d'instruction

```
projet_map<-st_as_sf(projet,coords = c("gps_menlongitude","gps_menlatitude"),crs=4326)
```

```
# contours
sen_contours <- st_read("Sen/Limite_Région.shp")
```

```
## Reading layer 'Limite_RÃ©gion' from data source
## 'C:\Users\DELL\OneDrive\Bureau\PROJET 24 JUILLET 2023\Sen\Limite_RÃ©gion.shp'
## using driver 'ESRI Shapefile'
## Simple feature collection with 14 features and 4 fields
## Geometry type: POLYGON
## Dimension: XY
## Bounding box: xmin: 227586.3 ymin: 1362012 xmax: 897104.7 ymax: 1845672
## Projected CRS: WGS 84 / UTM zone 28N
```

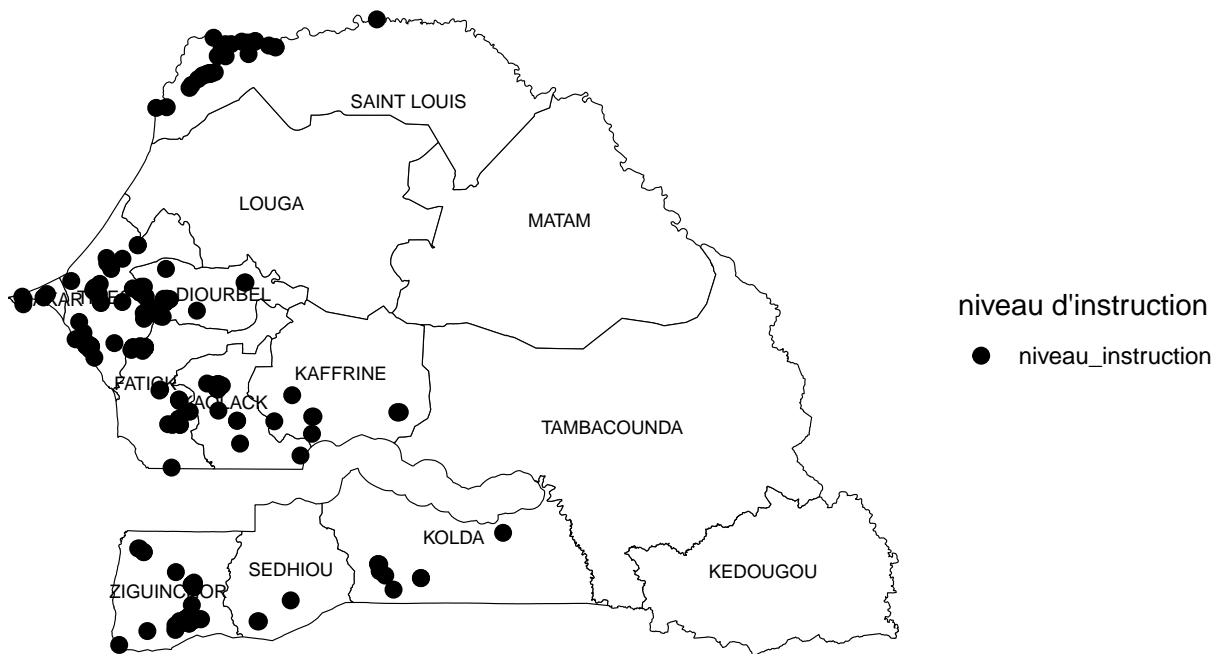
```
names(sen_contours)[1] <- "region"
ggplot()+
```

```

geom_sf(data=sen_contours,fill="white",color="black")+
geom_sf(data=projet_map,aes(color= "niveau_instruction"),size=2.5)+
geom_sf_text(data=sen_contours,aes(label=region),size=2.5)+
scale_color_manual(values = c("black", "green")) +
theme_void()+
theme(legend.position = "right")+
labs(title="carte des PME par sexe",color="niveau d'instruction")

```

carte des PME par sexe



```

#title("Carte du Sénégal avec les régions")

```

2.0.2 Chargement des données

```

library(readxl)
library(dplyr)
library(sp)

```

```

## Warning: le package 'sp' a été compilé avec la version R 4.1.3

```

```

library(sf)
base<- read_excel("Base_Partie 1.xlsx")

```

3 chargement des données du senegal gadm

```
projet_map <- st_as_sf(projet, coords = c("gps_menlongitude", "gps_menlatitude"), crs = 4326)
class(projet_map)
```

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

3.0.1 Charger les données sur le Sénégal

```
Senegal <- sf::st_read("gadm41_SEN.gpkg")
```

```
## Multiple layers are present in data source C:\Users\DELL\OneDrive\Bureau\PROJET 24 JUILLET 2023\gadm
## Use 'st_layers' to list all layer names and their type in a data source.
## Set the 'layer' argument in 'st_read' to read a particular layer.
```

```
## Warning in CPL_read_ogr(dsn, layer, query, as.character(options), quiet, :
## automatically selected the first layer in a data source containing more than
## one.
```

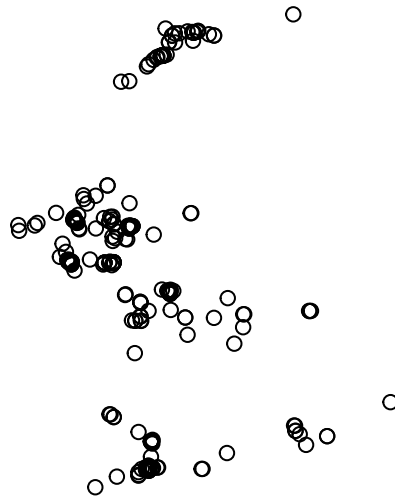
```
## Reading layer 'ADM_ADM_0' from data source
## 'C:\Users\DELL\OneDrive\Bureau\PROJET 24 JUILLET 2023\gadm41_SEN.gpkg'
## using driver 'GPKG'
## Simple feature collection with 1 feature and 2 fields
## Geometry type: MULTIPOLYGON
## Dimension: XY
## Bounding box: xmin: -17.54319 ymin: 12.30786 xmax: -11.34247 ymax: 16.69207
## Geodetic CRS: WGS 84
```

3.0.2 Transformer le data.frame en données géographiques

```
library(ggplot2)
library(sf)
projet<- st_as_sf(projet, coords = c("gps_menlongitude", "gps_menlatitude"), crs = "+proj=gps_menlongi
```

```
## Warning in CPL_crs_from_input(x): GDAL Error 1: PROJ: proj_create: Error -5
## (unknown projection id)
```

```
plot(projet["geometry"])
```



#Données spaciales du Sénégal

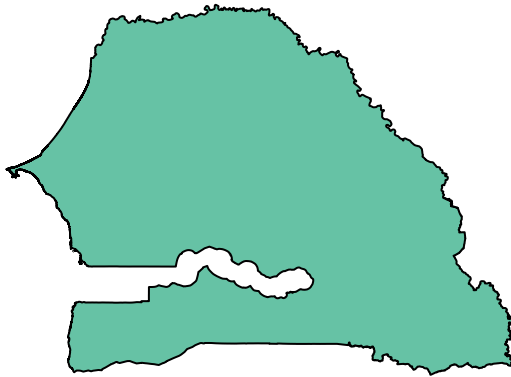
```
head(Senegal, 4)
```

```
## Simple feature collection with 1 feature and 2 fields
## Geometry type: MULTIPOLYGON
## Dimension:      XY
## Bounding box:   xmin: -17.54319 ymin: 12.30786 xmax: -11.34247 ymax: 16.69207
## Geodetic CRS:   WGS 84
##   GID_0 COUNTRY          geom
## 1  SEN Senegal MULTIPOLYGON (((-15.95338 1...
```

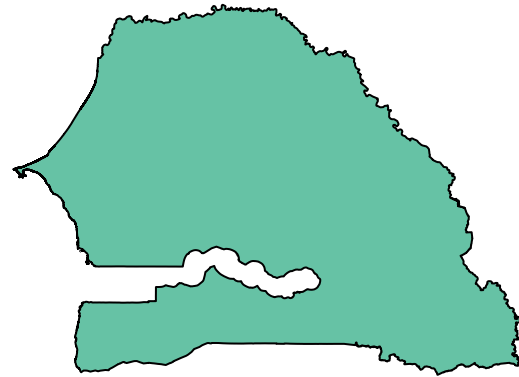
#Carte du Sénégal

```
plot(Senegal[])
```


GID_0



COUNTRY



4 PARTIE 2

4.1 Nettoyage et gestion de données

#Importation de la base

```
#Importation de la fichier Base_Partie 2.xlsx qui est de type excel
#Data se trouve à la feuille 1 du fichier
Base2 <- read_excel("Base_Partie 2.xlsx", sheet = 1)
#District se trouve à la feuille 2 du fichier
district <- read_excel("Base_Partie 2.xlsx", sheet = 2)
#Codebook se trouve à la feuille 3 du fichier
codebook <- read_excel("Base_Partie 2.xlsx", sheet = 3)
#Renommons les variables comme on avait fait avant
Base2 <- Base2 %>%dplyr::rename(destination = country_destination)
```

```
library(readxl)
library(dplyr)
Base2<- read_excel("Base_Partie 2.xlsx",1)
```

• Renommer la variable “country_destination” en “destination”

```
Base2<-Base2 %>%  
rename(destination=country_destination)
```

```
Base2$destination<-ifelse(Base2$destination<0,NA,Base2$destination)  
print(Base2$destination)
```

```
## [1] 8 6 NA 9 9 8 3 13 10 3 NA 8 NA NA 3 8 5 10 10 4 NA 9 5 10 13  
## [26] 10 NA 9 NA NA 9 4 9 9 8 5 10 8 NA 5 5 10 9 9 9 11 8 9 9 10  
## [51] 10 9 NA 10 3 NA 6 NA 3 9 9 9 10 10 9 10 NA 8 8 10 NA 9 9 NA 9  
## [76] 10 5 NA 8 13 13 13 NA 10 NA 5 NA 3 NA 9 10 3 13 9 6 10 5
```

```
kable(Base2$destination)
```

x
8
6
NA
9
9
8
3
13
10
3
NA
8
NA
NA
3
8
5
10
10
4
NA
9
5
10
13
10
NA
9
NA
NA
9
4
9
9
8
5
10
8
NA
5
5
10
9
9
9
11
8
9
9
10
10
9
NA
10
3
NA
6
NA

```
which(is.na(Base2$destination))
```

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

#définition les valeurs negatives comme manquantes.

```
Base2$destination<-ifelse(Base2$destination<0,NA,Base2$destination)
print(Base2$destination)
```

```
## [1] 8 6 NA 9 9 8 3 13 10 3 NA 8 NA NA 3 8 5 10 10 4 NA 9 5 10 13
## [26] 10 NA 9 NA NA 9 4 9 9 8 5 10 8 NA 5 5 10 9 9 9 11 8 9 9 10
## [51] 10 9 NA 10 3 NA 6 NA 3 9 9 9 10 10 9 10 NA 8 8 10 NA 9 9 NA 9
## [76] 10 5 NA 8 13 13 13 NA 10 NA 5 NA 3 NA 9 10 3 13 9 6 10 5
```

• Créer une nouvelle variable contenant des tranches d'âge de 5 ans en utilisant la variable "age".

```
library(kableExtra)
attach(Base2)
```

```
## L'objet suivant est masqué _par_ .GlobalEnv:
##
## district
```

```
##Recuperation du premier quartile
a<-quantile(Base2$age)[2]
##Recuperation du premier quartile
b<-quantile(Base2$age)[4]
##calcul de la borne inferieur
born_inf=a-1.5*(b-a)
##calcul de la borne superieur
born_sup=b+1.5*(b-a)
## Detection et imputation des valeurs aberrantes par la moyenne des ages
Base2$age_aberr<-ifelse((Base2$age<born_inf)|(Base2$age>born_sup),mean(Base2$age),Base2$age)
#Nous allons à present recoder la variable age en une variable categorielle.
attach(Base2)
```

```
## L'objet suivant est masqué _par_ .GlobalEnv:
##
## district
```

```
## Les objets suivants sont masqués depuis Base2 (pos = 3):
##
## age, children_num, destination, district, endtime, enumerator, id,
## intention, sex, starttime
```

```
##creation des bornes de la tranche d'age
ecart<-5
bornes<-seq(min(Base2$age_aberr),max(Base2$age_aberr),by=ecart)
##decoupage de la variable age des tranches d'age
Base2$age_categ<-cut(Base2$age_aberr,breaks = bornes)
kable(Base2$age_categ)
```

x
(30,35]
(35,40]
(25,30]
(20,25]
(25,30]
(20,25]
(20,25]
(15,20]
(20,25]
(15,20]
(20,25]
(15,20]
(20,25]
(15,20]
(15,20]
(15,20]
(35,40]
(15,20]
(20,25]
(15,20]
(20,25]
(25,30]
(35,40]
(15,20]
(20,25]
(35,40]
(20,25]
(35,40]
(30,35]
(20,25]
(15,20]
(25,30]
(25,30]
(20,25]
(25,30]
(35,40]
(20,25]
(30,35]
(30,35]
(20,25]
(15,20]
(20,25]
(25,30]
(35,40]
NA
(25,30]
(20,25]
(35,40]
(15,20]
(15,20]
(30,35]
(15,20]
(20,25]
(20,25]
(20,25]

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

```
library(dplyr)

# Utilisation de la chaîne de transformation (pipe %>%)
Base2 <- Base2 %>%
  group_by(enumerator) %>%
  mutate(nbre_entretien = n()) %>%
  distinct()

# Créer un tableau récapitulatif du nombre d'entretiens par enquêteur
table_entretiens <- table(Base2$enumerator, Base2$nbre_entretien)

# Afficher le tableau récapitulatif
print(table_entretiens)
```

```
##
##      1 5 6 7 8 9
## 1  0 5 0 0 0 0
## 4  0 0 0 0 0 9
## 5  0 0 6 0 0 0
## 6  0 5 0 0 0 0
## 7  0 0 0 7 0 0
## 8  0 0 6 0 0 0
## 9  0 0 6 0 0 0
## 10 0 5 0 0 0 0
## 11 0 0 0 7 0 0
## 12 0 5 0 0 0 0
## 13 0 0 0 0 8 0
## 14 0 0 6 0 0 0
## 15 1 0 0 0 0 0
## 17 0 0 6 0 0 0
## 18 0 0 6 0 0 0
## 20 0 0 0 0 0 9
```

```
table(table_entretiens)
```

```
## table_entretiens
##  0  1  5  6  7  8  9
## 80  1  4  6  2  1  2
```

• Créer une nouvelle variable qui affecte aléatoirement chaque répondant à un groupe de traitement (1) ou de controle (0).

```
set.seed(42) # Pour reproduire les mêmes résultats aléatoires
projet$groupe_traitement <- sample(0:1, nrow(projet), replace = TRUE)
table(projet$groupe_traitement )
```

```
##
##  0  1
## 113 137
```

• 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.

```
partie2_feuille2<- read_excel("Base_Partie 2.xlsx",sheet="district")
partie2_feuille2<-data.frame(partie2_feuille2)

# Fusionner les données
partie2_taill<-Base2%>%
merge(partie2_feuille2,by="district")## la fonction fusionne les bases de données
print(partie2_feuille2)
```

```
##   district population
## 1         1      10000
## 2         2       5000
## 3         3       3000
## 4         4       2000
## 5         5       1500
## 6         6      15000
## 7         7      50000
## 8         8       1000
```

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

```
Base2<- read_excel("Base_Partie 2.xlsx")
library(lubridate)
partie2_taill<-partie2_taill %>%
mutate(
  dure_entr = time_length(
    interval(
      start = starttime,
      end = endtime
    ),
    unit = "hour"
  )
)
#select(nom, date_naissance, age) %>%
#glimpse()
colnames(partie2_taill)
```

```
## [1] "district"      "id"             "starttime"      "endtime"
## [5] "enumerator"    "age"            "sex"            "children_num"
## [9] "intention"     "destination"    "age_aberr"      "age_categ"
## [13] "nbre_entretien" "population"     "dure_entr"
```

```
partie2_taill%>% group_by(enumerator)%>%
transmute(moyenne=mean(dure_entr))%>% 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

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

```
## recueillir le nombre de colonne de la base
n<-ncol(partie2_taill)
##une boucle qui va parcourir le nombre de ligne pour ajouter les prefixes
for (i in 1:n) {
  colnames(partie2_taill)[i]<-paste("endline_",colnames(partie2_taill)[i],sep ="" )
}
```

#Tableau récapitulatif contenant l'âge moyen et le nombre moyen d'enfants par district

```
Base2<- read_excel("Base_Partie 2.xlsx")
Base2 %>%
  group_by(district)%>%
  dplyr::summarise(mean(age), sum(mean(children_num)))
```

```
## # A tibble: 8 x 3
##   district 'mean(age)' 'sum(mean(children_num))'
##   <dbl>      <dbl>      <dbl>
## 1         1        29.6          1.5
## 2         2        62.6          0.852
## 3         3        26.1           0
## 4         4         26           0
## 5         5        24.3          0.5
## 6         6        23.2          0.115
## 7         7         28          0.167
## 8         8        24.6          1.27
```

```
t.test(Base2$age ~ Base2$sex, Base2 = Base2[-46,])
```

```
##
## Welch Two Sample t-test
##
```



```
## data: Base2$age by Base2$sex
## t = -0.95493, df = 10.001, p-value = 0.3621
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
## -282.7605 113.1009
## sample estimates:
## mean in group 0 mean in group 1
## 25.98837 110.81818
```

#Créer un nuage de points de l'âge en fonction du nombre d'enfants

5 PARTIE3

#SHINY

```
# This is a Shiny web application. You can run the application by clicking
# the 'Run App' button above.
#
# Find out more about building applications with Shiny here:
#
# http://shiny.rstudio.com/
#
library(sp)
library(ggplot2)
library(dplyr)
library(shiny)
```

```
## Warning: le package 'shiny' a été compilé avec la version R 4.1.3
```

```
library(leaflet)
```

```
## Warning: le package 'leaflet' a été compilé avec la version R 4.1.3
```

```
library(rnaturalearth)
```

```
## Support for Spatial objects ('sp') will be deprecated in {rnaturalearth} and will be removed in a future version
```

```
library(rnaturalearthdata)
```

```
## Warning: le package 'rnaturalearthdata' a été compilé avec la version R 4.1.3
```

```
##
## Attachement du package : 'rnaturalearthdata'
```

```
## L'objet suivant est masqué depuis 'package:rnaturalearth':
```

```
##
## countries110
```

```

# Charger les données géographiques de l'Afrique de l'Ouest
ne_countries_data <- ne_countries(scale = "medium", continent = "Africa")

## Warning: The 'returnclass' argument of 'ne_download()' sp as of rnatuarearth 1.0.0.
## i Please use 'sf' objects with {rnatuarearth}, support for Spatial objects
## (sp) will be removed in a future release of the package.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.

west_africa <- subset(ne_countries_data, subregion == "Western Africa")

# Charger les données de base en dehors de la fonction server
base <- read.csv("ACLED-Western_Africa.csv")

ui <- fluidPage(
  # titre de l'application
  titlePanel("shiny map"),

  # Sidebar with a slider input for number of bins
  sidebarLayout(
    sidebarPanel(
      selectInput(
        inputId = "evenement",
        label = "Sélectionnez un evenement",
        choices = c(unique(base$type)),
        selected = "Protests",
        multiple = TRUE
      ),
      selectInput(
        inputId = "pays",
        label = "Sélectionnez un pays",
        choices = c(unique(base$pays)),
        selected = c(unique(base$pays))[sample(1:length(unique(base$pays)), 1)],
        multiple = TRUE
      ),
      selectInput(
        inputId = "annee",
        label = "Sélectionnez une annee",
        choices = c(unique(base$annee)),
        selected = "2023",
        multiple = TRUE
      ),
    ),

    # Show a plot of the generated distribution
    mainPanel(
      leafletOutput(outputId = "map", width = "100%", height = "720px")
    )
  )

```

```

)
)

server <- function(input, output, session) {
  filtered_data <- reactive({
    subset(base, pays %in% input$pays & type %in% input$evenement & annee %in% input$annee)
  })

  output$map <- renderLeaflet({
    filtered_west_africa <- west_africa[west_africa$name %in% input$pays]

    leaflet() %>%
      addProviderTiles(providers$Stamen.Toner) %>%
      addPolygons(data = ne_countries(type = "countries", country = input$pays), fillColor = "green", c

      addCircleMarkers(data = filtered_data(),
                        lat = ~latitude,
                        lng = ~longitude,
                        radius = 3,
                        opacity = 0.7)
  })
}

shinyApp(ui = ui, server = server)

##
## Listening on http://127.0.0.1:7166

```

shiny map

Sélectionnez un événement

Protests

Sélectionnez un pays

Burkina Faso

Sélectionnez une année

2023

