PARTIE 1 ET 2 DU PROJET R

Mouhamadou Moustapha WADE

Table of contents

#Partie I

1.1.2 Importation et mise en forme

```
# Installer et charger la bibliothèque readxl pour importer des fichiers Excel
library(readxl)
library(dplyr)
library(flextable)
library(gt)
```

1.1.2.1: Importer la base de données dans un objet de type data.frame nommé “projet”

```
projet <- read_excel("Base_Partie_1.xlsx")
```

1.1.2.2: Sélectionner les variables mentionnées dans la section de description

```
variables_selectionnees <-projet %>% select(-c("key"))
```

1.1.2.3: Faire un tableau résumant les valeurs manquantes par variable

```
tableau_valeurs_manquantes <- data.frame(Variables = names(variables_selectionnees),
                                         Valeurs_Manquantes = colSums(is.na(variables_sele
# Display the gt table
tab_valeurs_manquantes <- gt(tableau_valeurs_manquantes)
tableau_valeurs_manquantes
```

	Variables	Valeurs_Manquantes
q1	q1	0
q2	q2	0
q23	q23	0
q24	q24	0
q24a_1	q24a_1	0
q24a_2	q24a_2	0
q24a_3	q24a_3	0
q24a_4	q24a_4	0
q24a_5	q24a_5	0

```
q24a_6                     q24a_6                 0
q24a_7                     q24a_7                 0
q24a_9                     q24a_9                 0
q24a_10                   q24a_10                 0
q25                           q25                 0
q26                           q26                 0
q12                           q12                 0
q14b                         q14b                 1
q16                           q16                 1
q17                           q17               131
q19                           q19               120
q20                           q20                 0
filiere_1                 filiere_1               0
filiere_2                 filiere_2               0
filiere_3                 filiere_3               0
filiere_4                 filiere_4               0
q8                               q8               0
q81                             q81               0
gps_menlatitude     gps_menlatitude               0
gps_menlongitude   gps_menlongitude               0
submissiondate       submissiondate               0
start                         start               0
today                         today               0
```

1.1.2.4: Vérifier s'il y a des valeurs manquantes pour la variable "key" dans la base de données "projet"

```
valeurs_manquantes_key <-projet[is.na(projet$key),"key"]
#  Faire un tableau
tableau_valeurs_manquantes_key <- data.frame(Variables = names(valeurs_manquantes_key),
                                             Valeurs_Manquantes = colSums(is.na(valeurs_ma
# Convert the data frame to a flex table
tableau_valeurs_manquantes_key <- gt(tableau_valeurs_manquantes_key)
tableau_valeurs_manquantes_key
```

Variables	Valeurs_Manquantes
key	0

1.1.3 Création de variables

1.1.3.1: Renommer les variables spécifiées

```
#1.3) Creation de variables

library(dplyr)
library(flextable)

#  1: Renommer les variables spécifiées
projet <- projet %>%
  rename(region = q1,
         departement = q2,
         sexe = q23)
```

1.1.3.2: Créer la variable sexe_2 qui vaut 1 si sexe est égal à “Femme”et 0 sinon

```
projet <- projet %>%
  mutate(sexe_2 = ifelse(sexe == "Femme", 1, 0))
```

1.1.3.3: Créer le data.frame langues en extrayant les variables correspondantes

```
variables_langues <- grep("^q24a_", names(projet), value = TRUE)
langues <- projet %>%
  select(key, all_of(variables_langues))
```

1.1.3.4: Créer la variable “parle” qui représente le nombre de langues parlées par le dirigeant de la PME

```
langues <- langues %>%
  mutate(parle = rowSums(.[variables_langues]))
```

1.1.3.5: Sélectionner uniquement les variables “key” et “parle” pour obtenir l’objet “langues”

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

1.1.3.6 : Fusionner les data.frames “projet_final” et “langues” en utilisant la variable “key”

```
projet <- projet %>%
  left_join(langues, by = "key")
```

```
#  Faire un tableau
tableau_projet <- data.frame(Variables = names(projet),
                                          Valeurs_Manquantes = colSums(is.na(projet)))
tab_Mean <- flextable::as_flextable(tableau_projet)
tab_Mean
```

Variables	Valeurs_Manquantes
character	numeric
key	0
region	0
departement	0
sexe	0
q24	0
q24a_1	0
q24a_2	0
q24a_3	0
q24a_4	0
q24a_5	0
n: 35	

1.2 Analyses descriptives

```
# Charger les packages nécessaires
library(dplyr)
library(gtsummary)
library(lubridate)
library(ggplot2)
```

###(1.2.1 Répartition suivant les variables sexe,niveau d'instruction, ###proprietaire ou locataire et statut juridique)

```
#Quelle est la répartion des PME suivant:
projet <- projet %>%
  rename(niveau_instruction= q25,
         proprietaire_locataire = q81,
         statut_juridique = q12)
# Répartition suivant les variables denandées

Statisitiques_des_variables<-projet %>% tbl_summary(include =c(sexe,niveau_instruction,pro
missing_text=("valeurs manquantes")
    )
Statisitiques_des_variables
```

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

Characteristic	N = 250
SUARL	7 (2.8%)

1.2.2 Créer le tableau récapitulatif global avec toutes les analyses demandées

```
Differentes_croisement<-projet%>%tbl_summary(include=c(sexe,niveau_instruction,
                                                       proprietaire_locataire,statut_jurid
  add_n()%>%
  add_stat_label()%>%
  add_overall()%>%
  as_flex_table()
Differentes_croisement
```

Characteristic	N	Overall, N = 250[1]	Femme, N = 191	Homme, N = 59
niveau_instruction, %	250			
Aucun niveau		32%	37%	15%
Niveau primaire		22%	25%	14%
Niveau secondaire		30%	29%	31%
Niveau Superieur		16%	8.9%	41%
proprietaire_locataire, %	250			
Locataire		9.6	8.4	14
Propriétaire		90	92	86
statut_juridique, %	250			
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

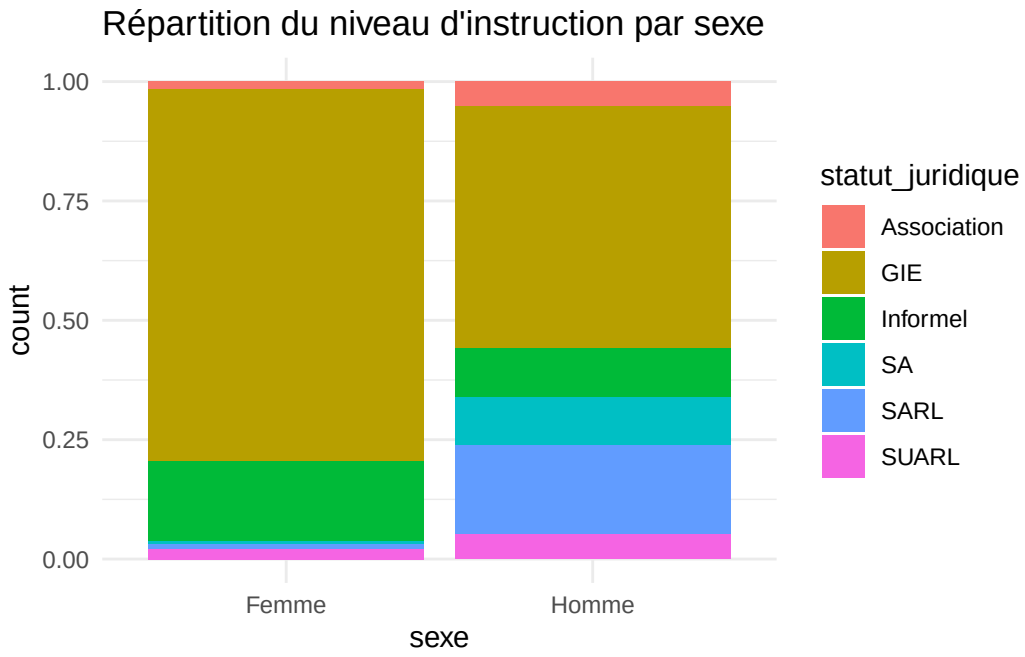
[1]%

1.2.2.1 Répartition par statut juridique , propriétaire locataire et niveau d'instruction par la variable sexe et différentes graphiques

```
# Créer le tableau récapitulatif pour la répartition par statut juridique etsexe
tableau_repartition_statut_sexe <- projet %>%
  tbl_cross(
    row =statut_juridique,
    col = sexe,
    percent = "row"
  )%>%
  add_p(source_note=TRUE)
tableau_repartition_statut_sexe
```

	Femme	Homme	Total
statut_juridique			
Association	3 (50%)	3 (50%)	6 (100%)
GIE	149 (83%)	30 (17%)	179 (100%)
Informel	32 (84%)	6 (16%)	38 (100%)
SA	1 (14%)	6 (86%)	7 (100%)
SARL	2 (15%)	11 (85%)	13 (100%)
SUARL	4 (57%)	3 (43%)	7 (100%)
Total	191 (76%)	59 (24%)	250 (100%)

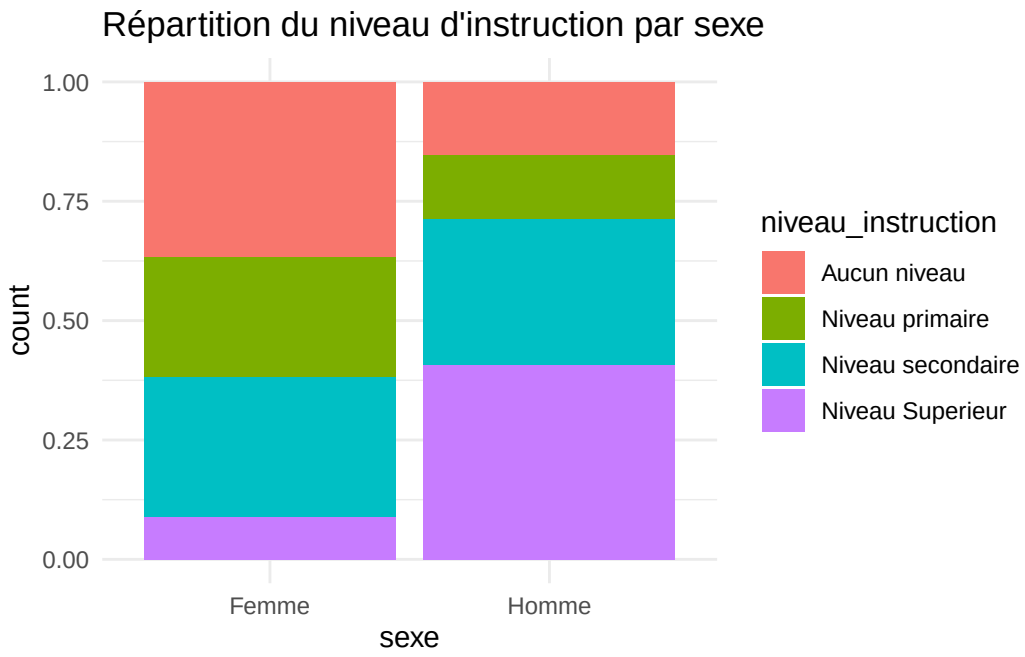
```
# Créer le graphique pour mieux visualiser les résultats
graphique1<- ggplot(projet, aes(x = sexe, fill = statut_juridique)) +
  geom_bar(position = "fill") +
  labs(title = "Répartition du niveau d'instruction par sexe") +
  theme_minimal()
graphique1
```




```
# Créer le tableau récapitulatif pour la répartition par niveau d'instruction
#et sexe
tableau_repartition_niveau_sexe <- projet %>%
  tbl_cross(
    row =niveau_instruction,
    col = sexe,
    percent = "row"
  )%>%
  add_p(source_note=TRUE)
tableau_repartition_niveau_sexe
```

	Femme	Homme	Total
niveau_instruction			
Aucun niveau	70 (89%)	9 (11%)	79 (100%)
Niveau primaire	48 (86%)	8 (14%)	56 (100%)
Niveau secondaire	56 (76%)	18 (24%)	74 (100%)
Niveau Superieur	17 (41%)	24 (59%)	41 (100%)
Total	191 (76%)	59 (24%)	250 (100%)

```r
# Créer le graphique pour mieux visualiser les résultats
graphique2 <- ggplot(projet, aes(x = sexe, fill = niveau_instruction)) +
  geom_bar(position = "fill") +
  labs(title = "Répartition du niveau d'instruction par sexe") +
  theme_minimal()
graphique2
```



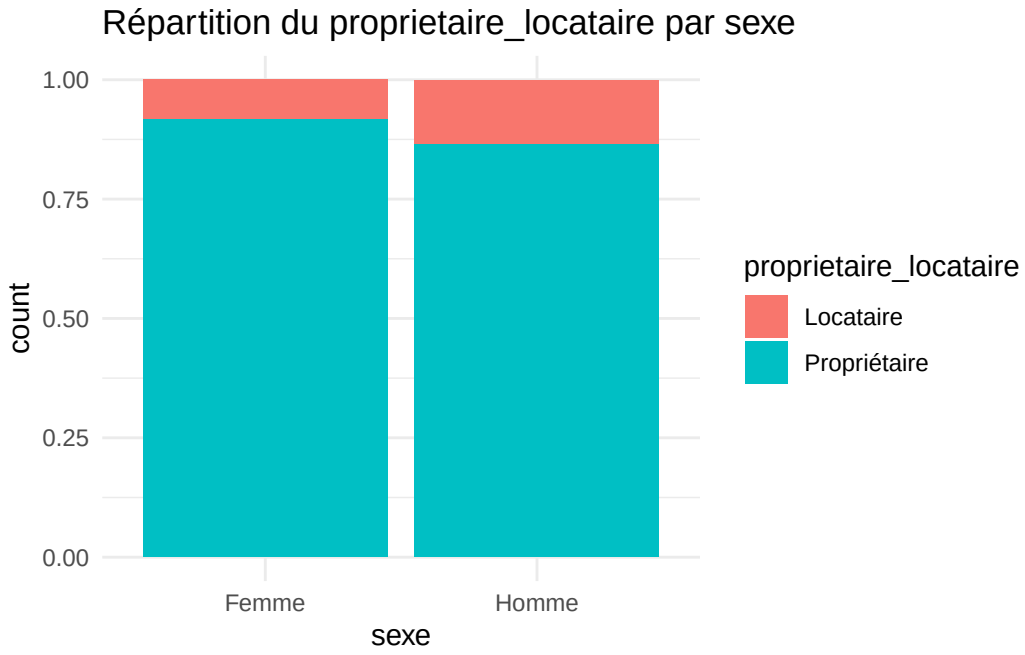

```r
# Créer le tableau récapitulatif pour la répartition de propriétaire/locataire suivant le
tableau_repartition_proprietaire_sexe <- projet %>%
  tbl_cross(
    row =proprietaire_locataire,
    col = sexe,
    percent = "row"
  )%>%
  add_p(source_note=TRUE)

tableau_repartition_proprietaire_sexe
```

	Femme	Homme	Total
proprietaire_locataire			

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

```
# Créer le graphique pour mieux visualiser les résultats
graphique3 <- ggplot(projet, aes(x = sexe, fill = proprietaire_locataire)) +
  geom_bar(position = "fill") +
  labs(title = "Répartition du proprietaire_locataire par sexe") +
  theme_minimal()
graphique3
```



####1.2.2.2Priorisez une analyse par filière

```
#Nommer les filières d'abord
projet <- projet %>%
  rename("Arachide" = filiere_1,
         "Anacarde" = filiere_2,
         "Mangue" = filiere_3,
         "Riz" = filiere_4)
```

```r
library(gtsummary)
library(dplyr)
t1<-subset(projet,Arachide==1)%>%
dplyr:: select(sexe,niveau_instruction,statut_juridique,proprietaire_locataire,Arachide) %
  gtsummary::tbl_summary(
    by=Arachide,
    statistic = list(
      all_categorical()~ "{n}/{N} ({p}%)"
    ),
    missing = "no",
    percent = "column"
    ) %>%
  modify_header(label ~ "**variable**") %>%
  bold_labels()
t1
```

variable	**1**, N = 108
sexe	
Femme	93/108 (86%)
Homme	15/108 (14%)
niveau_instruction	
Aucun niveau	43/108 (40%)
Niveau primaire	23/108 (21%)
Niveau secondaire	34/108 (31%)
Niveau Superieur	8/108 (7.4%)
statut_juridique	
Association	2/108 (1.9%)
GIE	79/108 (73%)
Informel	23/108 (21%)
SA	2/108 (1.9%)
SARL	1/108 (0.9%)
SUARL	1/108 (0.9%)
proprietaire_locataire	
Locataire	12/108 (11%)
Propriétaire	96/108 (89%)

```r
t2<-subset(projet,Anacarde==1)%>%
  dplyr:: select(sexe,niveau_instruction,statut_juridique,proprietaire_locataire,Anacarde)
  gtsummary::tbl_summary(
    by=Anacarde,
```

```
    statistic = list(
      all_categorical()~ "{n}/{N} ({p}%)"
    ),
    missing = "no",
    percent = "column"
  ) %>%
  modify_header(label ~ "**variable**") %>%
  bold_labels()
t2
```

variable	**1**, N = 61
sexe	
Femme	40/61 (66%)
Homme	21/61 (34%)
niveau_instruction	
Aucun niveau	13/61 (21%)
Niveau primaire	17/61 (28%)
Niveau secondaire	15/61 (25%)
Niveau Superieur	16/61 (26%)
statut_juridique	
Association	3/61 (4.9%)
GIE	35/61 (57%)
Informel	12/61 (20%)
SA	2/61 (3.3%)
SARL	6/61 (9.8%)
SUARL	3/61 (4.9%)
proprietaire_locataire	
Locataire	7/61 (11%)
Propriétaire	54/61 (89%)

```
t3<-subset(projet,Mangue==1)%>%
  dplyr:: select(sexe,niveau_instruction,statut_juridique,proprietaire_locataire
                 ,Mangue) %>%
  gtsummary::tbl_summary(
    by=Mangue,
    statistic = list(
      all_categorical()~ "{n}/{N} ({p}%)"
    ),
    missing = "no",
    percent = "column"
```

```
  ) %>%
  modify_header(label ~ "**variable**") %>%
  bold_labels()
t3
```

variable	**1**, N = 89
sexe	
Femme	68/89 (76%)
Homme	21/89 (24%)
niveau_instruction	
Aucun niveau	26/89 (29%)
Niveau primaire	24/89 (27%)
Niveau secondaire	25/89 (28%)
Niveau Superieur	14/89 (16%)
statut_juridique	
GIE	73/89 (82%)
Informel	5/89 (5.6%)
SA	3/89 (3.4%)
SARL	6/89 (6.7%)
SUARL	2/89 (2.2%)
proprietaire_locataire	
Locataire	11/89 (12%)
Propriétaire	78/89 (88%)

```
t4<-subset(projet,Riz==1)%>%
  dplyr:: select(sexe,niveau_instruction,statut_juridique,proprietaire_locataire
                 ,Riz) %>%
  gtsummary::tbl_summary(
    by=Riz,
    statistic = list(
      all_categorical()~ "{n}/{N} ({p}%)"
    ),
    missing = "no",
    percent = "column"
  ) %>%
  modify_header(label ~ "**variable**") %>%
  bold_labels()
t4
```

variable	1, N = 92
sexe	
Femme	77/92 (84%)
Homme	15/92 (16%)
niveau_instruction	
Aucun niveau	11/92 (12%)
Niveau primaire	26/92 (28%)
Niveau secondaire	32/92 (35%)
Niveau Superieur	23/92 (25%)
statut_juridique	
Association	2/92 (2.2%)
GIE	77/92 (84%)
Informel	3/92 (3.3%)
SA	3/92 (3.3%)
SARL	5/92 (5.4%)
SUARL	2/92 (2.2%)
proprietaire_locataire	
Locataire	9/92 (9.8%)
Propriétaire	83/92 (90%)

```
gtsummary:: tbl_merge(list(t1,t2,t3,t4),
                      tab_spanner = c("Arachide","Anacarde","Mangue","Riz"))
```

variable	1, N = 108	1, N = 61	1, N = 89	1, N = 92
sexe				
Femme	93/108 (86%)	40/61 (66%)	68/89 (76%)	77/92 (84%)
Homme	15/108 (14%)	21/61 (34%)	21/89 (24%)	15/92 (16%)
niveau_instruction				
Aucun niveau	43/108 (40%)	13/61 (21%)	26/89 (29%)	11/92 (12%)
Niveau primaire	23/108 (21%)	17/61 (28%)	24/89 (27%)	26/92 (28%)
Niveau secondaire	34/108 (31%)	15/61 (25%)	25/89 (28%)	32/92 (35%)
Niveau Superieur	8/108 (7.4%)	16/61 (26%)	14/89 (16%)	23/92 (25%)
statut_juridique				
Association	2/108 (1.9%)	3/61 (4.9%)		2/92 (2.2%)
GIE	79/108 (73%)	35/61 (57%)	73/89 (82%)	77/92 (84%)
Informel	23/108 (21%)	12/61 (20%)	5/89 (5.6%)	3/92 (3.3%)
SA	2/108 (1.9%)	2/61 (3.3%)	3/89 (3.4%)	3/92 (3.3%)
SARL	1/108 (0.9%)	6/61 (9.8%)	6/89 (6.7%)	5/92 (5.4%)
SUARL	1/108 (0.9%)	3/61 (4.9%)	2/89 (2.2%)	2/92 (2.2%)
proprietaire_locataire				

variable	1, N = 108	1, N = 61	1, N = 89	1, N = 92
Locataire	12/108 (11%)	7/61 (11%)	11/89 (12%)	9/92 (9.8%)
Propriétaire	96/108 (89%)	54/61 (89%)	78/89 (88%)	83/92 (90%)

1.3 Un peu de cartographie

```
library(sf)
library(ggplot2)
library(rnaturalearth)
library(RColorBrewer)
library(leaflet)
library(htmlwidgets)
library(dplyr)
## Obtenir les limites géographiques du Sénégal à partir de rnaturalearth
senegal <- ne_countries(country = "Senegal", returnclass = "sf")
```

###1.3.1 Charger les données depuis le fichier Excel et créer un objet sf

```
data <- readxl::read_excel("Base_Partie_1.xlsx")
projet_map <- st_as_sf(data, coords = c("gps_menlongitude", "gps_menlatitude"), crs = 4326

# Jointure spatiale entre les données de projet_map et les limites géographiques du Sénéga
projet_map <- st_join(projet_map, senegal)
```

1.3.2 Représentation spatiale des PME suivant le sexe

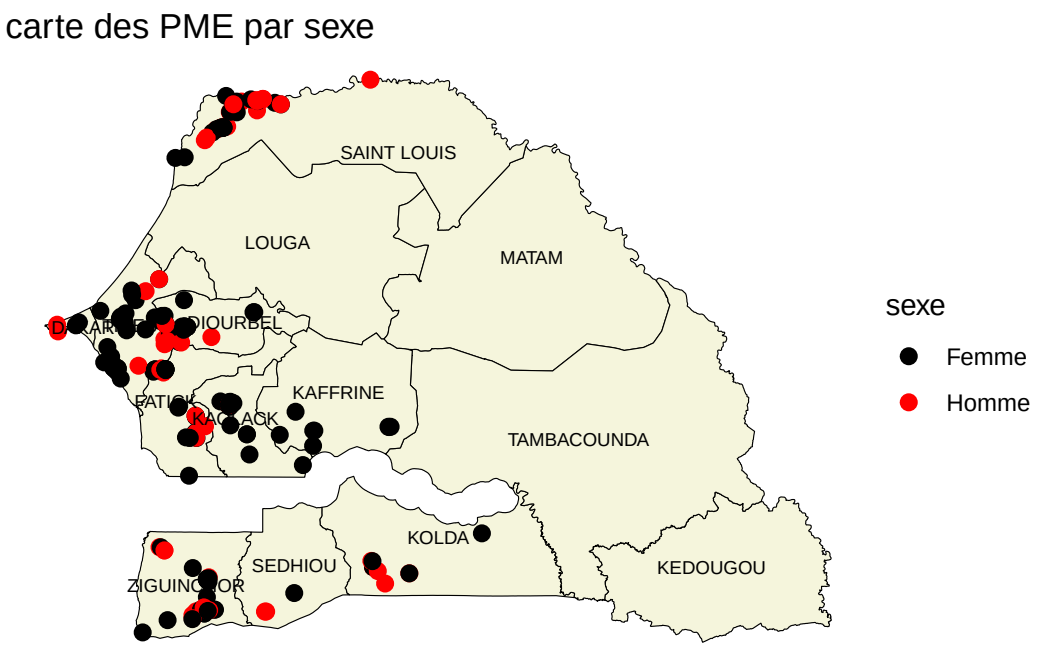
```
library(sf)
projet_map <- st_as_sf(projet, coords = c("gps_menlongitude", "gps_menlatitude")
                       , crs = 4326)

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

```
Reading layer `Limite_Région' from data source
  `C:\Users\utilisateur\Documents\Shiny2\Limite_Région.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
```

```r
names(sen_contours)[1] <-"region"
ggplot()+
  geom_sf(data=sen_contours,fill="beige",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", "red")) +
theme_void()+
theme(legend.position = "right")+
labs(title="carte des PME par sexe",color="sexe")
```



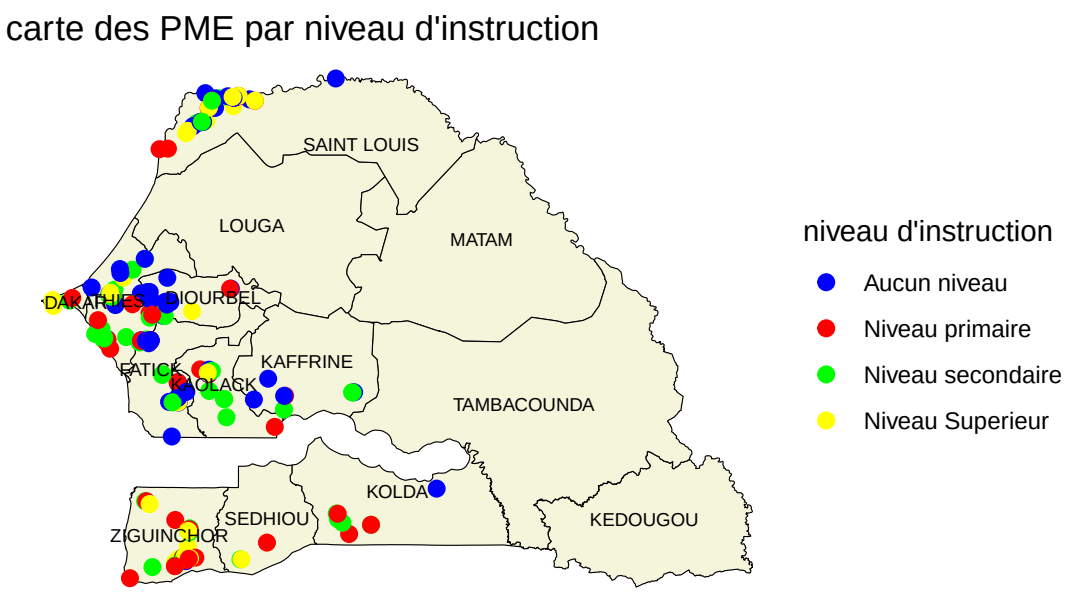
1.3.3 Représentation spatiale des PME suivant le niveau d'instruction

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

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

```
Reading layer `Limite_Région' from data source
  `C:\Users\utilisateur\Documents\Shiny2\Limite_Région.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
```

```r
names(sen_contours)[1] <-"region"
ggplot()+
  geom_sf(data=sen_contours,fill="beige",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("blue", "red","green","yellow")) +
theme_void()+
theme(legend.position = "right")+
labels(title="carte des PME par niveau d'instruction",color="niveau d'instruction")
```



1.3.3Analyse spatiale de votre choix (projet cartographie)

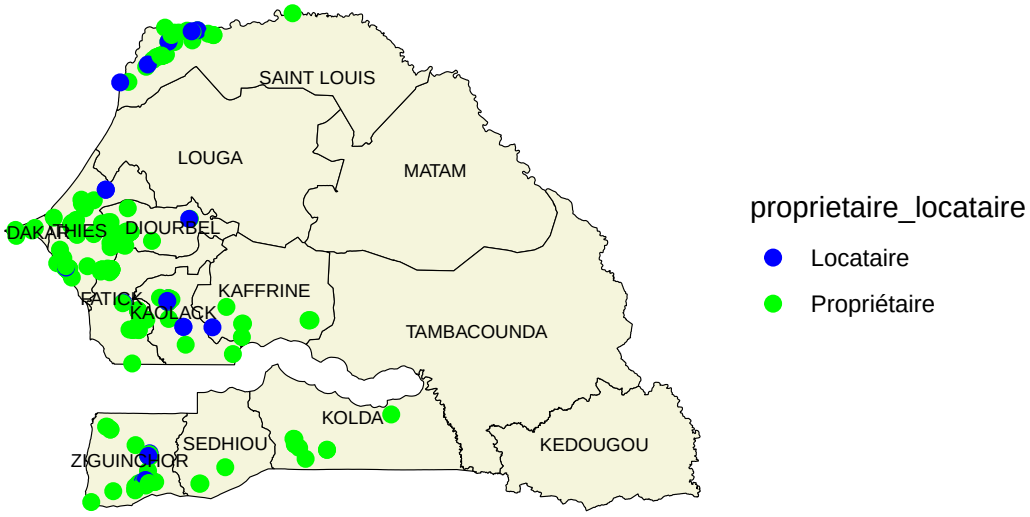
```
library(sf)
projet_map <- st_as_sf(projet, coords = c("gps_menlongitude", "gps_menlatitude")
                       , crs = 4326)
# Par exemple, représenter le nombre de PME par situation proprietaire ou locataire

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

```
Reading layer `Limite_Région' from data source
  `C:\Users\utilisateur\Documents\Shiny2\Limite_Région.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="beige",color="black")+
   geom_sf(data=projet_map,aes(color=proprietaire_locataire),size=2.5)+
   geom_sf_text(data=sen_contours,aes(label=region),size=2.5)+
   scale_color_manual(values = c("blue","green")) +
 theme_void()+
 theme(legend.position = "right")+
 labs(title="carte des PME par situation proprietaire oulocataire",color="proprietaire_loc
```

carte des PME par situation proprietaire oulocataire



#Partie II

##2.1 Nettoyage et gestion des données

```
library(dplyr)
library(readxl)

# Importer les données de la feuille 1 du fichier Excel
data_feuille1 <- read_excel("Base_Partie 2.xlsx", sheet = 1)
```

###2.1.1 Renommer la variable “country_destination” en “destination” et remplacer les valeurs négatives par NA

```
data_feuille1 <- data_feuille1 %>%
  rename(destination = country_destination) %>%
  mutate(destination = ifelse(destination < 0, NA, destination))
```

###2.1.2 Créer une nouvelle variable avec des tranches d’âge de 5 ans en utilisant la variable “age”

```
Tranche_age<- data_feuille1 %>%
  mutate(age_group = cut(age, breaks = seq(0, max(age), by = 5)))
```

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

```
data_feuille1 <- data_feuille1 %>%
  group_by(enumerator) %>%
  mutate(num_entretiens = n()) %>%
  ungroup()
```

###2.1.4 Créer une nouvelle variable qui affecte aléatoirement chaque répondant à un groupe de traitement (1) ou de contrôle (0)

```
set.seed(123)  # Pour reproduire les mêmes résultats aléatoires
data_feuille1 <- data_feuille1 %>%
  mutate(groupe_traitement = sample(c(0, 1), size = n(), replace = TRUE))
```

###2.1.5 Fusionner la taille de la population de chaque district avec l’ensemble de données

```
# Importer les données de la feuille 2 du fichier Excel
data_feuille2 <- read_excel("Base_Partie 2.xlsx", sheet = 2)
```

```r
data_feuille1 <- data_feuille1 %>%
  left_join(data_feuille2, by = "district")
```

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

```r
a <- data_feuille1 %>%
  mutate(duree_entretien = endtime - starttime) %>%
  group_by(enumerator) %>%
  mutate(duree_moyenne_entretien = mean(duree_entretien)) %>%
  ungroup()
a
```

```
# A tibble: 97 x 15
      id starttime           endtime             enumerator district   age   sex
   <dbl> <dttm>              <dttm>                   <dbl>    <dbl> <dbl> <dbl>
 1     2 2019-01-14 14:56:37 2019-01-14 15:11:10          6        1    33     1
 2     3 2019-01-14 16:12:22 2019-01-14 16:45:52          6        1    43     0
 3     4 2019-01-14 17:15:47 2019-01-14 17:45:47          6        1    28     0
 4     7 2019-01-14 13:04:51 2019-01-14 13:27:38          8        3    24     0
 5     8 2019-01-14 13:38:00 2019-01-14 14:31:16          8        3    29     0
 6    10 2019-01-14 15:52:17 2019-01-14 16:33:39          8        6    22     1
 7    11 2019-01-14 16:52:55 2019-01-14 17:13:39          8        6    21     0
 8    12 2019-01-14 13:17:56 2019-01-14 19:01:39          9        6    20     0
 9    13 2019-01-14 14:14:10 2019-01-14 18:05:26          9        6    21     1
10    14 2019-01-14 16:17:33 2019-01-14 16:41:51          9        6    20     0
# i 87 more rows
# i 8 more variables: children_num <dbl>, intention <dbl>, destination <dbl>,
#   num_entretiens <int>, groupe_traitement <dbl>, population <dbl>,
#   duree_entretien <drtn>, duree_moyenne_entretien <drtn>
```

###2.1.7 Renommer toutes les variables de l'ensemble de données en ajoutant le préfixe "endline_"

```r
data_feuille1 <- data_feuille1 %>%
  rename_with(~paste0("endline_", .), everything())


#  Faire un tableau
tableau_data_feuille_1 <- data.frame(Variables = names(data_feuille1),
                                     Valeurs_Manquantes = colSums(is.na(data_feuille1)
```

```r
# Convert the data frame to a gt table
tab_data_feuille_1 <- gt(tableau_data_feuille_1)
tab_data_feuille_1
```

Variables	Valeurs_Manquantes
endline_id	0
endline_starttime	0
endline_endtime	0
endline_enumerator	0
endline_district	0
endline_age	0
endline_sex	0
endline_children_num	0
endline_intention	0
endline_destination	20
endline_num_entretiens	0
endline_groupe_traitement	0
endline_population	0

##2.2Analyse et visualisation des données

###2.2.1 Tableau récapitulatif de l'age moyen et d'enfants moyen par district

```r
#Analyse et visualisation des données
library(readxl)
library(ggplot2)
library(dplyr)
# Importer les données de la feuille 2 du fichier Excel
data_feuille2 <- read_excel("Base_Partie 2.xlsx", sheet = 2)

# Tableau récapitulatif de l'age moyen et d'enfants moyen par district

tab_Mean <- flextable::as_flextable(data_feuille1 %>% group_by(endline_district) %>% summa
tab_Mean
```

endline_district	Age_Moyen	Enfant_Moyen
numeric	numeric	numeric
1	29.6	1.5
2	62.6	0.9

endline_district	Age_Moyen	Enfant_Moyen
numeric	numeric	numeric
3	26.1	0.0
4	26.0	0.0
5	24.3	0.5
6	23.2	0.1
7	28.0	0.2
8	24.6	1.3

###2.2.2 Test de différence d'âge entre les sexes

```
# Charger les packages nécessaires
library(dplyr)
library(gtsummary)

# Créer une copie du dataframe data_feuille1 pour éviter de modifier
#les données originales
data_feuille1_copy <- data_feuille1

# Sélectionner les colonnes "endline_sex" et "endline_age"
appli <- data_feuille1_copy %>%
  dplyr::select(endline_sex, endline_age) %>%

  # Créer un résumé de table avec gtsummary
  gtsummary::tbl_summary(by = endline_sex,
                         label = list(endline_age ~ "Tranche d'âge"),
                         statistic = list(endline_age ~ "{mean}"),
                         percent = "column") %>%

  # Ajouter le test de différence de moyennes
  add_difference(test = list(all_continuous() ~ "t.test")) %>%

  # Ajouter la statistique globale pour l'ensemble des données
  add_overall() %>%

  # Convertir en flextable (si vous souhaitez une sortie au format FlexTable)
  as_flex_table()
```

```
# Afficher la table résumée
appli
```

Characteristic	**Overall**, N = 97[1]	**0**, N = 86[1]	**1**, N = 11[1]	**Difference**[2]	**95% CI**[23]	**p-value**[2]
Tranche d'âge	36	26	111	-85	-283, 113	0.4

[1]Mean

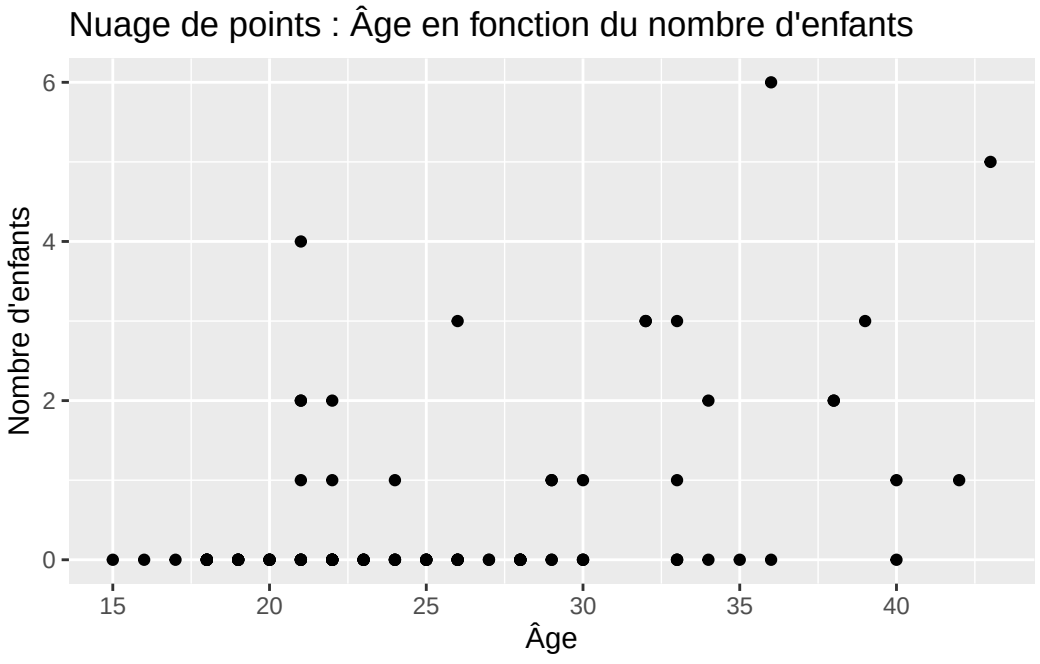
[2]Welch Two Sample t-test

[3]CI = Confidence Interval

###2.2.3 Nuage de points : âge en fonction du nombre d'enfants

```
#Utilisons le package ggplot pour tracer le nuage de points en eliminant la valeur abérant
nuage_points_age_enfants <- ggplot(filter(data_feuille1,!(endline_age==999)), aes(x = endl
y = endline_children_num)) +
  geom_point() +
  labs(x = "Âge", y = "Nombre d'enfants") +
  ggtitle("Nuage de points : Âge en fonction du nombre d'enfants")

nuage_points_age_enfants
```



###2.2.4 Estimation de l'effet de l'appartenance au groupe de traitement sur l'intention de migrer

```
modele_regression <- lm(endline_intention ~endline_groupe_traitement,
                        data = data_feuille1)
```

###2.2.5 Tableau de régression avec 3 modèles

```
# Chargez le package gtsummary s'il n'est pas déjà installé
library(gtsummary)

#installer les  packages nécessaires
library(sjPlot)

# Modèle A : Modèle vide - Effet du traitement sur les intentions
model_A <- lm(endline_intention ~endline_groupe_traitement, data = data_feuille1)

# Modèle B : Effet du traitement sur les intentions en tenant compte de l'âge et du sexe
model_B <- lm(endline_intention ~ endline_groupe_traitement + endline_age + endline_sex, d

# Modèle C : Identique au modèle B mais en contrôlant le district
model_C <- lm(endline_intention ~ endline_groupe_traitement + endline_age + endline_sex +

# Créer un tableau récapitulatif des modèles
tableau_recapitulatif_modele <-tab_model(model_A, model_B, model_C, title = "Tableau de ré
show.ci = TRUE) # Afficher les intervalles de confiance

# Afficher le tableau récapitulatif, le test de différence d'âge, le nuage de points et le
tableau_recapitulatif_modele
```

Table 16: Tableau de régression

	endline_intention				endline_intention		
Predictors	Estimates	std. Error	CI	p	Estimates	std. Error	CI
(Intercept)	1.95	0.23	-Inf – Inf	**<0.001**	2.08	0.25	-Inf – Inf
endline groupe traitement	0.34	0.35	-Inf – Inf	0.337	0.27	0.35	-Inf – Inf
endline age					-0.00	0.00	-Inf – Inf
endline sex					-0.86	0.57	-Inf – Inf
endline district							
Observations	97				97		

	endline_intention	endline_intention
R^2 / R^2 adjusted	0.010 / -0.001	0.036 / 0.005