Projet R

Fatou DIOP

Table of Contents

Partie 1

1. Préparation des données

2. Importation et mise en forme

Nous allons importer la base de données dans un objet de type data.frame nommé projet

```
library(readxl) #importation du package  readxl pour lire les fichers excel
library(dplyr) ## il va nous permettre de manipuler les données

##
## 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(gtsummary) ## permet de resumer les données dans un tableau
projet <- readxl::read_excel("Base_Partie 1.xlsx") # importation de la base
head(projet,n=15) # d'avoir un aperçu  de la base
```

```
## # A tibble: 15 × 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
## 11 uuid:a7513… Diou… Diou… Femme    35      0      1      0      1      0
0
## 12 uuid:d1d36… Thiès Thiès Femme    61      0      1      0      1      0
0
## 13 uuid:a583c… Thiès Thiès Femme    54      0      1      0      0      0
0
## 14 uuid:4eead… Diou… Bamb… Femme    40      0      1      0      1      0
0
## 15 uuid:232e9… Diou… Bamb… Femme    54      0      1      0      1      0
0
## # 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 <dttm>, start <dttm>, today
<dttm>
```

Faisons un tableau qui permet de résumer les valeurs manquantes par variable

```
#projet %>%  tbl_summary(statistic = ~"{p_miss}",missing_text = "
**valeurs_manquantes** ",missing="always" )
# ce code permet d'afficher pour chaque variable le nombre de valeurs
manquantes
```

```
val_manquante = colSums(is.na(projet))
val_manquante
```

```
##               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
##               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
```

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).

```
manquant <- is.na(projet$key) # la fonction is.na  renvoie TRUE si la valeur vaut NA et FALSE sinon.
which(manquant) # renvoie les indices des variables ayant des valeurs manquantes
```

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

1.3 creation de nouvelles variables

Renommons ces variables q1,q2 et q23 avec la fonction rename du package dplyr

```
## la fonction rename du package dplyr permet de renommer les variables
projet = dplyr::rename(projet,region=q1,
                       departement=q2,
                       sexe=q23)
head(projet,n=15)
```

```
## # A tibble: 15 × 33
##    key         region departement sexe    q24 q24a_1 q24a_2 q24a_3 q24a_4
q24a_5
##    <chr>       <chr>  <chr>       <chr> <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
<dbl>
##  1 uuid:68bff… Diour… Bambey      Femme    65      0      1      0      1
0
##  2 uuid:d70b3… Thiès  Mbour       Femme    52      1      1      0      0
1
##  3 uuid:0ac18… Thiès  Mbour       Femme    65      1      1      0      0
```

```
0
##  4 uuid:c52cf… Thiès  Mbour       Femme    38      1      1      0      0
1
##  5 uuid:ac177… Zigui… Bignona     Homme    40      1      1      1      0
0
##  6 uuid:57809… Zigui… Oussouye    Femme    43      1      1      1      0
0
##  7 uuid:c3065… Thiès  Thiès       Femme    53      0      1      0      1
0
##  8 uuid:74e60… Zigui… Ziguinchor  Homme    33      1      0      0      0
0
##  9 uuid:2ee01… Diour… Bambey      Femme    67      0      1      0      1
0
## 10 uuid:5c801… Saint… Dagana      Homme    35      1      1      0      0
0
## 11 uuid:a7513… Diour… Diourbel    Femme    35      0      1      0      1
0
## 12 uuid:d1d36… Thiès  Thiès       Femme    61      0      1      0      1
0
## 13 uuid:a583c… Thiès  Thiès       Femme    54      0      1      0      0
0
## 14 uuid:4eead… Diour… Bambey      Femme    40      0      1      0      1
0
## 15 uuid:232e9… Diour… Bambey      Femme    54      0      1      0      1
0
## # ℹ 23 more variables: q24a_6 <dbl>, 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 <dttm>, start <dttm>, today
<dttm>
```

creation d'une nouvelle variable sexe_2 avec la fonction mutate de dplyr

on crée une nouvelle variable à partir de la variable sexe qui exister déjà dans la base avec la fonction mutate

```
projet=dplyr:: mutate(projet,sexe_2=if_else(sexe=="Femme","1","0"))
head(projet,n=15)
```

```
## # A tibble: 15 × 34
##    key         region departement sexe    q24 q24a_1 q24a_2 q24a_3 q24a_4
q24a_5
##    <chr>       <chr>  <chr>       <chr> <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
<dbl>
##  1 uuid:68bff… Diour… Bambey      Femme    65      0      1      0      1
0
##  2 uuid:d70b3… Thiès  Mbour       Femme    52      1      1      0      0
1
```

```
##  3 uuid:0ac18… Thiès  Mbour       Femme    65      1      1      0      0
0
##  4 uuid:c52cf… Thiès  Mbour       Femme    38      1      1      0      0
1
##  5 uuid:ac177… Zigui… Bignona     Homme    40      1      1      1      0
0
##  6 uuid:57809… Zigui… Oussouye    Femme    43      1      1      1      0
0
##  7 uuid:c3065… Thiès  Thiès       Femme    53      0      1      0      1
0
##  8 uuid:74e60… Zigui… Ziguinchor  Homme    33      1      0      0      0
0
##  9 uuid:2ee01… Diour… Bambey      Femme    67      0      1      0      1
0
## 10 uuid:5c801… Saint… Dagana      Homme    35      1      1      0      0
0
## 11 uuid:a7513… Diour… Diourbel    Femme    35      0      1      0      1
0
## 12 uuid:d1d36… Thiès  Thiès       Femme    61      0      1      0      1
0
## 13 uuid:a583c… Thiès  Thiès       Femme    54      0      1      0      0
0
## 14 uuid:4eead… Diour… Bambey      Femme    40      0      1      0      1
0
## 15 uuid:232e9… Diour… Bambey      Femme    54      0      1      0      1
0
## # ℹ 24 more variables: q24a_6 <dbl>, 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 <dttm>, start <dttm>, today
<dttm>,
##   sexe_2 <chr>
```

Création d'un dataframe nomme langues

on cree une nouvelle base appelée qui contient l'ensemble des langues présents dans la base et les identifiants des PME avec la fonction select du package dplyr

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

head(langues,n=15)
```

```
## # A tibble: 15 × 10
##    key           q24a_1 q24a_2 q24a_3 q24a_4 q24a_5 q24a_6 q24a_7 q24a_9
q24a_10
##    <chr>          <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
<dbl>
##  1 uuid:68bff42…      0      1      0      1      0      0      0      0
```

```
0
##  2 uuid:d70b3c7…      1      1      0      0      1      0      0      0
0
##  3 uuid:0ac18b6…      1      1      0      0      0      0      0      0
0
##  4 uuid:c52cf5e…      1      1      0      0      1      0      0      0
0
##  5 uuid:ac17787…      1      1      1      0      0      1      0      0
0
##  6 uuid:578097c…      1      1      1      0      0      0      0      0
0
##  7 uuid:c3065ed…      0      1      0      1      0      0      0      0
0
##  8 uuid:74e608c…      1      0      0      0      0      0      0      0
1
##  9 uuid:2ee0131…      0      1      0      1      0      0      0      0
0
## 10 uuid:5c801b1…      1      1      0      0      0      0      0      0
1
## 11 uuid:a75139c…      0      1      0      1      0      0      0      0
0
## 12 uuid:d1d36e0…      0      1      0      1      0      0      0      0
0
## 13 uuid:a583cca…      0      1      0      0      0      0      0      0
0
## 14 uuid:4eeadb3…      0      1      0      1      0      0      0      0
0
## 15 uuid:232e9cf…      0      1      0      1      0      0      0      0
0
```

La creation d'une variable parle qui est égale au nombre de langue parlée par le dirigeant de la PME.

```
langues=langues %>% dplyr::mutate(parle = rowSums(langues%>%
dplyr::select(contains('q24a_'))))
head(langues,n=6)
```

```
## # A tibble: 6 × 11
##   key      q24a_1 q24a_2 q24a_3 q24a_4 q24a_5 q24a_6 q24a_7 q24a_9 q24a_10
parle
##   <chr>     <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>   <dbl>
<dbl>
## 1 uuid:68…      0      1      0      1      0      0      0      0       0
2
## 2 uuid:d7…      1      1      0      0      1      0      0      0       0
3
## 3 uuid:0a…      1      1      0      0      0      0      0      0       0
2
## 4 uuid:c5…      1      1      0      0      1      0      0      0       0
3
## 5 uuid:ac…      1      1      1      0      0      1      0      0       0
```

```
4
## 6 uuid:57…      1      1      1      0      0      0      0      0       0
3
```

Merger les data.frame projet et langues:

Nous allons utiliser la fonction merge qui va nous permettre de fusionner les deux bases en fonction de de la variable qu'elles ont en commun 'key'

```
projet_parle = merge(langues,projet,by="key")
head(projet_parle)
```

```
##                                       key q24a_1.x q24a_2.x q24a_3.x
q24a_4.x
## 1 uuid:004b9117-d180-4031-a6af-6b4efabb5f53        0        1        0
0
## 2 uuid:007d8eb4-45eb-44f4-aeac-722adc60aec8        0        1        0
1
## 3 uuid:030ada55-8dd2-4f57-b1b7-aaccd707c118        1        1        0
0
## 4 uuid:04b4cd8d-0297-4dc0-9715-9c07120bab23        1        1        0
0
## 5 uuid:080548bf-e68a-49b8-9f04-f920b44244aa        0        1        0
0
## 6 uuid:087f6e40-9b2a-4252-a1b3-b4b8ec7b3dff        1        1        0
0
##   q24a_5.x q24a_6.x q24a_7.x q24a_9.x q24a_10.x parle      region
departement
## 1        0        0        0        0         0     1    Diourbel
Bambey
## 2        0        0        0        0         0     2       Thiès
Tivaouane
## 3        0        0        0        0         0     2 Saint-Louis
Dagana
## 4        0        0        0        0         0     2    Diourbel
Mbacké
## 5        1        0        0        0         0     2 Saint-Louis
Dagana
## 6        0        0        0        0         0     2 Saint-Louis
Dagana
##    sexe q24 q24a_1.y q24a_2.y q24a_3.y q24a_4.y q24a_5.y q24a_6.y q24a_7.y
## 1 Femme  62        0        1        0        0        0        0        0
## 2 Femme  60        0        1        0        1        0        0        0
## 3 Homme  58        1        1        0        0        0        0        0
## 4 Femme  60        1        1        0        0        0        0        0
## 5 Homme  63        0        1        0        0        1        0        0
## 6 Femme  61        1        1        0        0        0        0        0
##   q24a_9.y q24a_10.y               q25 q26  q12 q14b q16      q17
q19
## 1        0         0      Aucun niveau  20  GIE  Non Non     <NA> Mauvais
état
```

```
## 2        0         0 Niveau secondaire  10  GIE  Non Oui Bon état
<NA>
## 3        0         0 Niveau secondaire  30  GIE  Non Non     <NA> Mauvais
état
## 4        0         0   Niveau primaire  25  GIE  Oui Oui Bon état
<NA>
## 5        0         0      Aucun niveau  21 SARL  Non Oui Bon état
<NA>
## 6        0         0   Niveau primaire  25  GIE  Non Non     <NA> Mauvais
état
##   q20 filiere_1 filiere_2 filiere_3 filiere_4
q8
## 1 Oui         1         0         0         0
Aucun
## 2 Non         1         0         0         1 Tansformation d'autres
céréales
## 3 Oui         0         0         1         0           Transformation du
riz
## 4 Oui         1         0         0         0 Tansformation d'autres
céréales
## 5 Non         0         0         1         0           Transformation du
riz
## 6 Oui         0         0         1         0           Transformation du
riz
##            q81 gps_menlatitude gps_menlongitude      submissiondate
## 1 Propriétaire        14.82743        -16.60590 2021-06-05 15:33:51
## 2    Locataire        15.10929        -16.62974 2021-06-15 01:10:46
## 3 Propriétaire        16.45945        -16.04850 2021-06-21 01:28:51
## 4    Locataire        14.85961        -15.88164 2021-06-07 13:51:55
## 5 Propriétaire        16.27839        -16.14392 2021-06-18 10:20:16
## 6 Propriétaire        16.45927        -16.04855 2021-06-21 01:31:17
##                  start      today sexe_2
## 1 2021-06-04 15:14:14 2021-06-04      1
## 2 2021-06-08 14:40:28 2021-06-08      1
## 3 2021-06-07 18:24:19 2021-06-07      0
## 4 2021-06-07 09:50:58 2021-06-07      1
## 5 2021-05-24 15:33:59 2021-05-24      0
## 6 2021-06-07 18:50:42 2021-06-07      1
```

Analyse descriptive

Dans cette partie, on va créer des tableaux qui vont nous permettre de résumer les variables pour en tirer des informations importantes de nos enquetés

On va faire une analyse univarié pour faire resortir les caractéristiques socio-économique et demongraphiques A travers ,le graphique ci-dessous on peut voir notre base contient 76M de femmes comme dirigeant/responsable de la PMEet et 32% des enquêtés n'ont aucun niveau et 30% ont un niveau secondaire du côté de statut juridique,76% sont des GIE et 15% sont informel et 9 sur 10 dirigeant/responsable de la PME sont les propriétaires

```r
library(flextable) ## il va nous permettre de créer des tableaux

##
## Attachement du package : 'flextable'

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

## En première lieu, on renomme quelques variables
theme_gtsummary_compact(set_theme = TRUE, font_size = NULL)

## Setting theme `Compact`

## Format de la sortie
theme_gtsummary_printer(
  print_engine = "flextable",  set_theme=TRUE)
projet =dplyr::rename(projet,niv_instruction=q25,Stat_juridique=q12,
prop_loca=q81)
print(projet) # juste pour vérifier

## # A tibble: 250 × 34
##    key         region departement sexe    q24 q24a_1 q24a_2 q24a_3 q24a_4
q24a_5
##    <chr>       <chr>  <chr>       <chr> <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
<dbl>
##  1 uuid:68bff… Diour… Bambey      Femme    65      0      1      0      1
0
##  2 uuid:d70b3… Thiès  Mbour       Femme    52      1      1      0      0
1
##  3 uuid:0ac18… Thiès  Mbour       Femme    65      1      1      0      0
0
##  4 uuid:c52cf… Thiès  Mbour       Femme    38      1      1      0      0
1
##  5 uuid:ac177… Zigui… Bignona     Homme    40      1      1      1      0
0
##  6 uuid:57809… Zigui… Oussouye    Femme    43      1      1      1      0
0
##  7 uuid:c3065… Thiès  Thiès       Femme    53      0      1      0      1
0
##  8 uuid:74e60… Zigui… Ziguinchor  Homme    33      1      0      0      0
0
##  9 uuid:2ee01… Diour… Bambey      Femme    67      0      1      0      1
0
## 10 uuid:5c801… Saint… Dagana      Homme    35      1      1      0      0
0
## # ℹ 240 more rows
## # ℹ 24 more variables: q24a_6 <dbl>, q24a_7 <dbl>, q24a_9 <dbl>, q24a_10
<dbl>,
##   niv_instruction <chr>, q26 <dbl>, Stat_juridique <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>,
## #   prop_loca <chr>, gps_menlatitude <dbl>, gps_menlongitude <dbl>,
## #   submissiondate <dttm>, start <dttm>, today <dttm>, sexe_2 <chr>

## on crée un tableau
tab1 <- projet %>%
gtsummary::tbl_summary(include =
c(sexe,niv_instruction,Stat_juridique,prop_loca),## spécifier les variables
que nous voulons résumer
label=list(Stat_juridique ~ "Statut jurudique",prop_loca~
"Propiété/locataire",niv_instruction~"niveau instruction"), ## permet de
renommer ces variables
statistic = list(all_continuous()~ "{mean}",all_categorical() ~ "{p} %
({n}/{N})"),# de choisir les statistiques qu'on veut sortir
type=list(sexe~"categorical"), ## de donner les types
digits = list(all_continuous() ~ 1,
      all_categorical() ~ c(0, 1)))%>%
  bold_labels() %>% # permet de mettre en gras les variables
  italicize_levels() %>% ## permet de mettre en gras les variables
  modify_header(list(label ~ "**Les Variables**")) # de changer l'entête

tab1 ## on affiche le tableau
```

Les Variables	**N = 250**[1]
sexe	
Femme	76 % (191.0/250)
Homme	24 % (59.0/250)
niveau instruction	
Aucun niveau	32 % (79.0/250)
Niveau primaire	22 % (56.0/250)
Niveau secondaire	30 % (74.0/250)
Niveau Superieur	16 % (41.0/250)
Statut jurudique	
Association	2 % (6.0/250)
GIE	72 % (179.0/250)
Informel	15 % (38.0/250)
SA	3 % (7.0/250)
SARL	5 % (13.0/250)
SUARL	3 % (7.0/250)
Propiété/locataire	
Locataire	10 % (24.0/250)
Propriétaire	90 % (226.0/250)

[1]% % (n/N)

on crée un tableau de contingence avec la fonction tbl_cross

Les statistiques nous montre que 60% des GIE sont dirigés pas des femmes et 30% par des femmes

```
tab2 = projet %>% gtsummary:: tbl_cross(
    row = Stat_juridique, #spécifier la variable à lettre en ligne
    col = sexe,#spécifier la variable à lettre en colonne
    percent = "cell" #Indique le type de pourcentage à retourner
  ) %>%
  bold_labels() %>%
```

```
  italicize_levels()%>% modify_footnote(everything() ~ NA)%>%
  modify_header(list(label ~ "**Les Variables**"))
 tab2
```

	sexe		
Les Variables	*Femme*	*Homme*	**Total**
Stat_juridique			
Association	3 (1.2%)	3 (1.2%)	6 (2.4%)
GIE	149 (60%)	30 (12%)	179 (72%)
Informel	32 (13%)	6 (2.4%)	38 (15%)
SA	1 (0.4%)	6 (2.4%)	7 (2.8%)
SARL	2 (0.8%)	11 (4.4%)	13 (5.2%)
SUARL	4 (1.6%)	3 (1.2%)	7 (2.8%)
Total	191 (76%)	59 (24%)	250 (100%)

nous créons aussi un autre tableau de contingence entre la varaible nivequ d'instruction et sexe le tableau suivant nous montre que 22% des femmes ont un niveau secondaire et 28% n'ont aucun niveau

```
tab3 = projet %>% gtsummary:: tbl_cross(
    row = niv_instruction,
    col = sexe,
    percent = "cell"
  ) %>%
  bold_labels() %>%
  italicize_levels()%>% modify_footnote(everything() ~ NA)%>%
  modify_header(list(label ~ "**Les Variables**"))
tab3
```

	sexe		
Les Variables	*Femme*	*Homme*	**Total**
niv_instruction			
Aucun niveau	70 (28%)	9 (3.6%)	79 (32%)
Niveau primaire	48 (19%)	8 (3.2%)	56 (22%)
Niveau secondaire	56 (22%)	18 (7.2%)	74 (30%)
Niveau Superieur	17 (6.8%)	24 (9.6%)	41 (16%)
Total	191 (76%)	59 (24%)	250 (100%)

un autre tablequ mais cette fois ,nous allons utiliser le paramétre by de la fonction tbl_summary pour regrouper

Les statistiques suivant nous montre que 92% des femmes sont propietaires propriétaires et 14% des homme sont des locataires

```
tab4 = projet %>% gtsummary::tbl_summary(include
=c(prop_loca,sexe),by=sexe)%>%
  bold_labels() %>%
  italicize_levels() %>%
  modify_header(list(label ~ "**Les Variables**"))
print(tab4)

## a flextable object.
## col_keys: `label`, `stat_1`, `stat_2`
## header has 1 row(s)
## body has 3 row(s)
```

```
## original dataset sample:
##          label    stat_1   stat_2
## 1    prop_loca      <NA>     <NA>
## 2    Locataire 16 (8.4%)  8 (14%)
## 3 Propriétaire 175 (92%) 51 (86%)
```

A présent ,on utilise la fonction tbl_stack du package gtsummary qui nous permettre de coller les tableaux créés ci-dessus l'un au-dessus de l'autre ainsi de suite

```
TABLEAU=tbl_stack(list(tab2,tab3,tab4,tab1),group_header =c("**le tableau de contingence du statut juridique et du sexe**","**le tableau de contingence du niveau d instruction et du sexe**","** classement  des Propriétaire/locataire suivant le sexe**" ,"analyse univarié") ,quiet = TRUE)
TABLEAU
```

		sexe		
Group	**Les Variables**	*Femme*	*Homme*	**Total**
le tableau de contingence du statut juridique et du sexe	**Stat_juridique**			
	Association	3 (1.2%)	3 (1.2%)	6 (2.4%)
	GIE	149 (60%)	30 (12%)	179 (72%)
	Informel	32 (13%)	6 (2.4%)	38 (15%)
	SA	1 (0.4%)	6 (2.4%)	7 (2.8%)
	SARL	2 (0.8%)	11 (4.4%)	13 (5.2%)
	SUARL	4 (1.6%)	3 (1.2%)	7 (2.8%)
	Total	191 (76%)	59 (24%)	250 (100%)
le tableau de contingence du niveau d instruction et du sexe	**niv_instruction**			
	Aucun niveau	70 (28%)	9 (3.6%)	79 (32%)
	Niveau primaire	48 (19%)	8 (3.2%)	56 (22%)
	Niveau secondaire	56 (22%)	18 (7.2%)	74 (30%)
	Niveau Superieur	17 (6.8%)	24 (9.6%)	41 (16%)
	Total	191 (76%)	59 (24%)	250 (100%)
** classement des Propriétaire/locataire suivant le sexe**	**prop_loca**			
	Locataire	16 (8.4%)	8 (14%)	
	Propriétaire	175 (92%)	51 (86%)	
analyse univarié	**sexe**			
	Femme			76 % (191.0/250)
	Homme			24 % (59.0/250)
	niveau instruction			
	Aucun niveau			32 % (79.0/250)
	Niveau primaire			22 % (56.0/250)
	Niveau secondaire			30 % (74.0/250)
	Niveau Superieur			16 % (41.0/250)
	Statut jurudique			
	Association			2 % (6.0/250)
	GIE			72 %

		sexe		
Group	**Les Variables**	*Femme*	*Homme*	**Total**
				(179.0/250)
	Informel			15 % (38.0/250)
	SA			3 % (7.0/250)
	SARL			5 % (13.0/250)
	SUARL			3 % (7.0/250)
	Propiété/locataire			
	Locataire			10 % (24.0/250)
	Propriétaire			90 % (226.0/250)

Faisons des analyses sur les autres variables

nous allons renommer toutes variables filiéres avec la fonction rename du package dplyr

```
projet = dplyr::rename(projet,arachide= filiere_1,
                       anacarde=    filiere_2,
                       mangue=  filiere_3,riz=  filiere_4)
print(projet)
```

```
## # A tibble: 250 × 34
##    key         region departement sexe    q24 q24a_1 q24a_2 q24a_3 q24a_4
q24a_5
##    <chr>       <chr>  <chr>       <chr> <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
<dbl>
##  1 uuid:68bff… Diour… Bambey      Femme    65      0      1      0      1
0
##  2 uuid:d70b3… Thiès  Mbour       Femme    52      1      1      0      0
1
##  3 uuid:0ac18… Thiès  Mbour       Femme    65      1      1      0      0
0
##  4 uuid:c52cf… Thiès  Mbour       Femme    38      1      1      0      0
1
##  5 uuid:ac177… Zigui… Bignona     Homme    40      1      1      1      0
0
##  6 uuid:57809… Zigui… Oussouye    Femme    43      1      1      1      0
0
##  7 uuid:c3065… Thiès  Thiès       Femme    53      0      1      0      1
0
##  8 uuid:74e60… Zigui… Ziguinchor  Homme    33      1      0      0      0
0
##  9 uuid:2ee01… Diour… Bambey      Femme    67      0      1      0      1
0
## 10 uuid:5c801… Saint… Dagana      Homme    35      1      1      0      0
0
## # ℹ 240 more rows
## # ℹ 24 more variables: q24a_6 <dbl>, q24a_7 <dbl>, q24a_9 <dbl>, q24a_10
<dbl>,
## #   niv_instruction <chr>, q26 <dbl>, Stat_juridique <chr>, q14b <chr>,
## #   q16 <chr>, q17 <chr>, q19 <chr>, q20 <chr>, arachide <dbl>, anacarde
```

```
<dbl>,
## #   mangue <dbl>, riz <dbl>, q8 <chr>, prop_loca <chr>, gps_menlatitude
<dbl>,
## #   gps_menlongitude <dbl>, submissiondate <dttm>, start <dttm>, today
<dttm>,
## #   sexe_2 <chr>
```

on crée une base nommée B_arachide qui ne va contenir que les PME qui sont dans la filiére Arachide Apres cela,on va créer un tableau avec la fonction tbl_summary pour tirer des informations

```
## filiére arachide
B_arachide=projet[projet$arachide==1,]
View(B_arachide)
tbl_arachide <- B_arachide %>%
  tbl_summary(include = c(sexe,
                          niv_instruction,
                          Stat_juridique,
                          prop_loca),
              label=list(Stat_juridique ~ "Statut juridique",
                         prop_loca~ "Propiété/locataire",
                         niv_instruction~"niveau instruction"),
              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)))%>%
  add_p()%>%
  bold_labels() %>%
  italicize_levels()
# %>%
# modify_header(list(label ~ "**Les Variables**"))
tbl_arachide
```

Characteristic	**Femme**, N = 93[1]	**Homme**, N = 15[1]	**p-value**[2]
niveau instruction			0.3
Aucun niveau	41 %	33 %	
Niveau primaire	22 %	20 %	
Niveau secondaire	32 %	27 %	
Niveau Superieur	5 %	20 %	
Statut juridique			0.016
Association	2 %	0 %	
GIE	75 %	60 %	
Informel	22 %	20 %	
SA	0 %	13 %	
SARL	0 %	7 %	
SUARL	1 %	0 %	
Propiété/locataire			0.4

Characteristic	Femme, N = 93[1]	Homme, N = 15[1]	p-value[2]
Locataire	10 %	20 %	
Propriétaire	90 %	80 %	

[1]% %
[2]Fisher's exact test

on crée une base nommée B_anacarde qui ne va contenir que les PME qui sont dans la filiére anacarde Apres cela,on va créer un tableau avec la fonction tbl_summary pour tirer des informations

```
#filiere anacarde
B_anacarde=projet[projet$anacarde==1,]
View(B_anacarde)

tbl_anacarde<- B_anacarde %>%
  tbl_summary(include = c(sexe,
                          niv_instruction,
                          Stat_juridique,
                          prop_loca),
              label=list(Stat_juridique ~ "Statut juridique",
                         prop_loca~ "Propiété/locataire",
                         niv_instruction~"niveau instruction"),
              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)))%>%
   add_p()%>%
  bold_labels() %>%
  italicize_levels() %>%
  modify_header(list(label ~ "**anacarde**"))
tbl_anacarde
```

anacarde	Femme, N = 40[1]	Homme, N = 21[1]	p-value[2]
niveau instruction			<0.001
Aucun niveau	30 %	5 %	
Niveau primaire	38 %	10 %	
Niveau secondaire	23 %	29 %	
Niveau Superieur	10 %	57 %	
Statut juridique			0.002
Association	3 %	10 %	
GIE	68 %	38 %	
Informel	25 %	10 %	
SA	0 %	10 %	
SARL	3 %	24 %	
SUARL	3 %	10 %	
Propiété/locataire			0.2
Locataire	8 %	19 %	
Propriétaire	93 %	81 %	

anacarde	Femme, N = 40[1]	Homme, N = 21[1]	p-value[2]

[1]% %
[2]Fisher's exact test

on crée une base nommée B_ riz qui ne va contenir que les PME qui sont dans la filiére Riz
Apres cela,on va créer un tableau avec la fonction tbl_summary pour tirer des informations

```
## filiére Riz
B_riz=projet[projet$riz==1,]
View(B_riz)
tbl_riz<- B_riz%>% tbl_summary(include = c(sexe,
                                           niv_instruction,
                                           Stat_juridique,
                                           prop_loca),
                               label=list(Stat_juridique ~ "Statut
juridique",
                                          prop_loca~ "Propiété/locataire",
                                          niv_instruction~"niveau
instruction"),
                               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)))%>%
   add_p()%>%
  bold_labels() %>%
  italicize_levels() %>%
  modify_header(list(label ~ "**riz**"))
tbl_riz
```

riz	Femme, N = 77[1]	Homme, N = 15[1]	p-value[2]
niveau instruction			<0.001
Aucun niveau	13 %	7 %	
Niveau primaire	34 %	0 %	
Niveau secondaire	36 %	27 %	
Niveau Superieur	17 %	67 %	
Statut juridique			<0.001
Association	0 %	13 %	
GIE	95 %	27 %	
Informel	1 %	13 %	
SA	0 %	20 %	
SARL	1 %	27 %	
SUARL	3 %	0 %	
Propiété/locataire			>0.9
Locataire	10 %	7 %	
Propriétaire	90 %	93 %	

[1]% %
[2]Fisher's exact test

on crée une base nommée B_mangue qui ne va contenir que les PME qui sont dans la filiére mangue Apres cela,on va créer un tableau avec la fonction tbl_summary pour tirer des informations

```
##filiere mangue
B_mangue=projet[projet$mangue==1,]
View(B_mangue)
tbl_mangue <- B_mangue %>%
  tbl_summary(include = c(sexe,
                          niv_instruction,
                          Stat_juridique,
                          prop_loca ),
              label=list(Stat_juridique ~ "Statut juridique",
                         prop_loca~ "Propiété/locataire",
                         niv_instruction~"niveau instruction"),
              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)))%>%
   add_p()%>%
  bold_labels() %>%
  italicize_levels() %>%
  modify_header(list(label ~ "**mangues **"))
tbl_mangue
```

mangues	**Femme**, N = 68[1]	**Homme**, N = 21[1]	**p-value**[2]
niveau instruction			0.004
Aucun niveau	32 %	19 %	
Niveau primaire	29 %	19 %	
Niveau secondaire	31 %	19 %	
Niveau Superieur	7 %	43 %	
Statut juridique			<0.001
GIE	91 %	52 %	
Informel	4 %	10 %	
SA	1 %	10 %	
SARL	1 %	24 %	
SUARL	1 %	5 %	
Propiété/locataire			0.7
Locataire	12 %	14 %	
Propriétaire	88 %	86 %	

[1]% %
[2]Fisher's exact test

```
tbl_merge(
  list(tbl_mangue,tbl_arachide, tbl_anacarde,tbl_riz),
  tab_spanner = c("mangue","arachide", "anacarde","riz")
)
```

mangue	arachide	anacarde	riz

mangues	Femme, N = 68[1]	Homme, N = 21[1]	p-value[2]	Femme, N = 93[1]	Homme, N = 15[1]	p-value[2]	Femme, N = 40[1]	Homme, N = 21[1]	p-value[2]	Femme, N = 77[1]	Homme, N = 15[1]	p-value[2]
niveau instruction			0.004			0.3			<0.001			<0.001
Aucun niveau	32 %	19 %		41 %	33 %		30 %	5 %		13 %	7 %	
Niveau primaire	29 %	19 %		22 %	20 %		38 %	10 %		34 %	0 %	
Niveau secondaire	31 %	19 %		32 %	27 %		23 %	29 %		36 %	27 %	
Niveau Superieur	7 %	43 %		5 %	20 %		10 %	57 %		17 %	67 %	
Statut juridique			<0.001			0.016			0.002			<0.001
GIE	91 %	52 %		75 %	60 %		68 %	38 %		95 %	27 %	
Informel	4 %	10 %		22 %	20 %		25 %	10 %		1 %	13 %	
SA	1 %	10 %		0 %	13 %		0 %	10 %		0 %	20 %	
SARL	1 %	24 %		0 %	7 %		3 %	24 %		1 %	27 %	
SUARL	1 %	5 %		1 %	0 %		3 %	10 %		3 %	0 %	
Association				2 %	0 %		3 %	10 %		0 %	13 %	
Propiété/locataire			0.7			0.4			0.2			>0.9
Locataire	12 %	14 %		10 %	20 %		8 %	19 %		10 %	7 %	
Propriétaire	88 %	86 %		90 %	80 %		93 %	81 %		90 %	93 %	

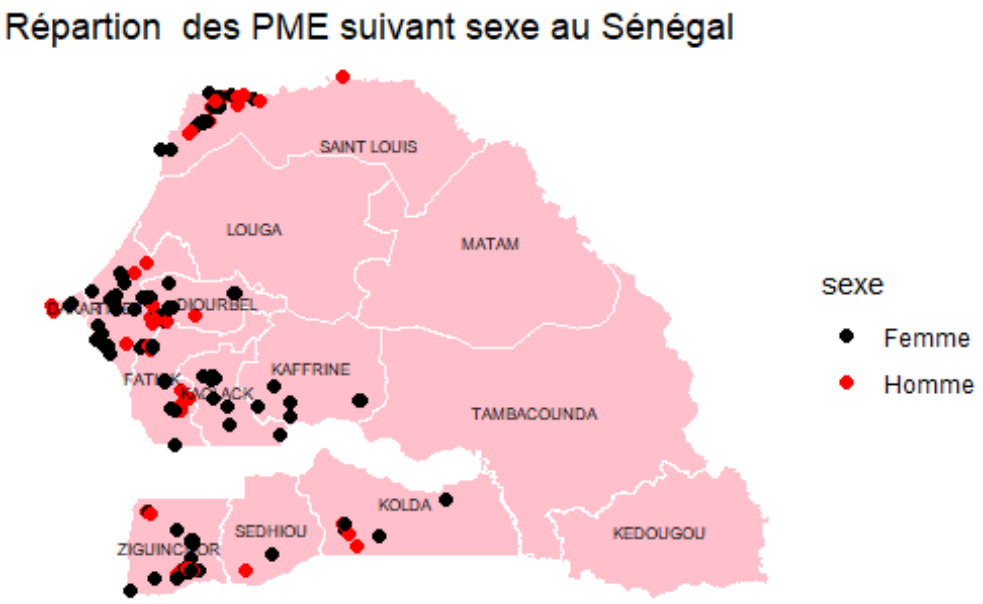
[1]% %
[2]Fisher's exact test

Un peu de cartographie

une représentation spatiale des PME suivant le sexe

```
library(sp) #  permet travailler avec des données géospatiales et des informations spatiales

## The legacy packages maptools, rgdal, and rgeos, underpinning the sp package,
## which was just loaded, will retire in October 2023.
## Please refer to R-spatial evolution reports for details, especially
## https://r-spatial.org/r/2023/05/15/evolution4.html.
## It may be desirable to make the sf package available;
## package maintainers should consider adding sf to Suggests:.
## The sp package is now running under evolution status 2
##      (status 2 uses the sf package in place of rgdal)

library(sf) #permet aussi pour travailler avec des données géospatiales sous forme de classes

## Linking to GEOS 3.9.3, GDAL 3.5.2, PROJ 8.2.1; sf_use_s2() is TRUE

library(dplyr)

library(ggplot2) #permet de créer de graphiques et de visualisations de données

projet_map<- sf::st_as_sf(projet,coords=
```

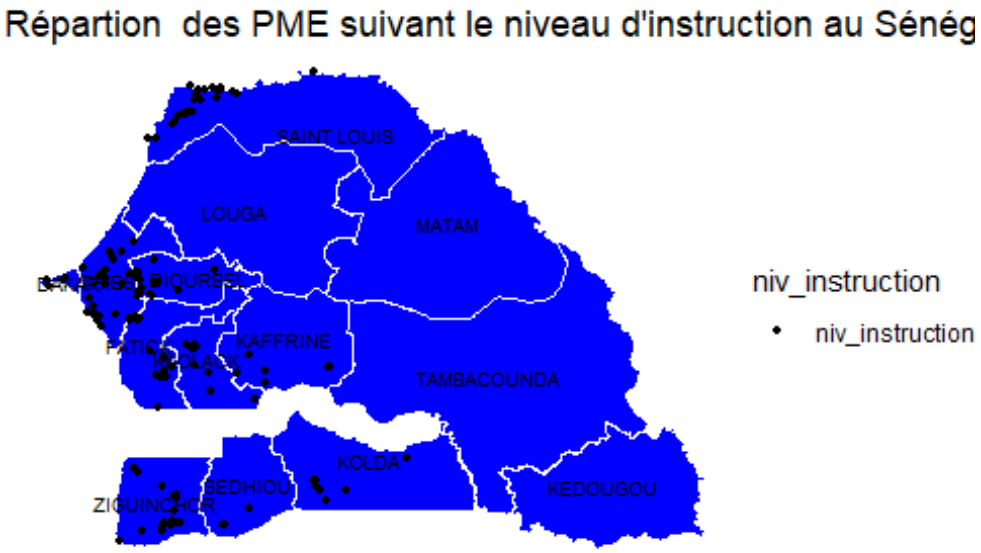
```
c("gps_menlongitude","gps_menlatitude"),
                            crs=4326)
##la fonction st_as_sf du package sf  permet  d'importer des données
géospatiales dans R et les convertir en objets sf

senegal <- sf::st_read("delimitation_SEN/Limite_Région.shp") #permet  de
lire les fichiers  shapefile

## Reading layer `Limite_Région' from data source
##   `C:\AENSAE\second se\projet sous
r\PROJET_RRRR\DIOP_Fatou_ISE\delimitation_SEN\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(senegal)[1] <- "region"
ggplot()+
  # la fonction geom_sf permet  d'ajouter des points, lignes ou polygones à
un graphique créé par ggplot2
  ggplot2::geom_sf(data=senegal,fill="pink",color="white")+
 ggplot2:: geom_sf(data=projet_map,aes(color=sexe),size=2)+
 ggplot2:: geom_sf_text(data=senegal,aes(label=region),size=2)+
  ggplot2::scale_color_manual(values = c("black", "red")) +
 ggplot2::theme_void()+# personnaliser  l'apparence du graphique
 theme(legend.position = "right")+
 labs(title="Répartion  des PME suivant sexe au Sénégal",color="sexe")
```



une réprésentation spatiale des PME suivant le niveau d'instruction

```
library(sp)
library(sf)
library(dplyr)
library(ggplot2)

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

senegal <- st_read("delimitation_SEN/Limite_Région.shp")

## Reading layer `Limite_Région' from data source
##   `C:\AENSAE\second se\projet sous
r\PROJET_RRRR\DIOP_Fatou_ISE\delimitation_SEN\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(senegal)[1] <- "region"
ggplot()+
  geom_sf(data=senegal,fill="blue",color="white")+
  geom_sf(data=projet_map,aes(color="niv_instruction"),size=1)+
  geom_sf_text(data=senegal,aes(label=region),size=2.5)+
```

```
  scale_color_manual(values = c("black", "violet")) +
 theme_void()+
 theme(legend.position = "right")+
 labs(title="Répartion  des PME suivant le niveau d'instruction au
Sénégal",color="niv_instruction")
```



Partie 2 :

###Nettoyage et gestion des données

importation de la base

nous allons importer en première lieu la premiere feuille du fichier excel nommé “Base_Partie 2” qui est une base

```
library(readxl)
library(dplyr)
library(gtsummary)
Base2 <- readxl::read_excel("Base_Partie 2.xlsx",sheet = 1) #
head(Base2)
```

```
## # A tibble: 6 × 10
##      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
## # ℹ 3 more variables: children_num <dbl>, intention <dbl>,
## #   country_destination <dbl>
```

renommons la variable “country destination”en “destination”

```
Base2 =dplyr::rename(Base2,destination=country_destination)
View(Base2)
```

definissons les valeurs négatifs de la variable destination en valeur manquante cette ligne de code permet de sélectionner toutes les valeurs négatives de la variable destination et de les considérer comme des données manquantes

```
Base2$destination[Base2$destination<0] <- NA
head(Base2)
```

```
## # A tibble: 6 × 10
##      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
## # ℹ 3 more variables: children_num <dbl>, intention <dbl>, destination
<dbl>
```

creation d’une nouvelle variable qui va nous permettre regrouper les ages en classe on utilise la fonction mutate du package dplyr et la fonction case_when pour créer des classes suivant des conditions

```
Base2$age <- as.integer(Base2$age)
Base2=dplyr::mutate(Base2, tran_age = case_when(
  Base2$age < 5 ~ "[0,5[",
  Base2$age >= 5 & Base2$age < 10 ~ "[5,10[",
  Base2$age >= 10 & Base2$age < 15 ~ "[10,15[",
  Base2$age >= 15 & Base2$age < 20 ~ "[15,20[",
  Base2$age >= 20 & Base2$age < 25 ~ "[20,25[",
  Base2$age >= 25 & Base2$age < 30 ~ "[25,30[",
  Base2$age >= 30 & Base2$age < 35 ~ "[30,35[",
  Base2$age >= 35 & Base2$age < 40 ~ "[35,40[",
  TRUE ~ "40 et plus"
))
View(Base2)
```

calculons le nombre d'interview fait par chaque enqueteur on utlise la fonction count qui permet de compter pour chaque modalité de la variable enumerator le nombre de fois dont il apparait ce qui vas nous donner le nombre d'interview fait par chaque enqueteur

```
nombre_interview=count(Base2,Base2$enumerator,sort = TRUE)
nombre_interview =as_tibble(nombre_interview )
nombre_interview
```

```
## # A tibble: 16 × 2
##    `Base2$enumerator`     n
##                 <dbl> <int>
##  1                  4     9
##  2                 20     9
##  3                 13     8
##  4                  7     7
##  5                 11     7
##  6                  5     6
##  7                  8     6
##  8                  9     6
##  9                 14     6
## 10                 17     6
## 11                 18     6
## 12                  1     5
## 13                  6     5
## 14                 10     5
## 15                 12     5
## 16                 15     1
```

la creation une nouvelle variable qui affecte aléatoirement chaque répondant à un groupe de traitement (1) ou de controle (0). on crée une nouvelle variable en utilisant la fonction sample dont les modalités seront 0 et 1 affecté aleatoire qau PME

```
set.seed(154) # nous permet de fixer l'alea
Base2 <- dplyr::mutate(Base2, groupe = sample(c(0, 1), size= nrow(Base2),
replace = TRUE))
View(Base2)
```

• 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.

```
library(readxl)
library(dplyr)
library(gtsummary)
taille <- readxl::read_excel("Base_Partie 2.xlsx",sheet = 2)
##on importe la deuxieme feuille du fichier excel Base_Partie 2
taille
```

```
## # A tibble: 8 × 2
##   district population
##      <dbl>      <dbl>
## 1        1      10000
## 2        2       5000
## 3        3       3000
## 4        4       2000
## 5        5       1500
## 6        6      15000
## 7        7      50000
## 8        8       1000
```

```
# on crée une nouvelle variable nommé taille_pop de la base 2 afin que toutes les personnes interrogées aient une valeur correspondante représentant la taille de la population du district dans lequel elles vivent.
Base2=dplyr::mutate(Base2,taille_Pop=case_when(
  Base2$district==1 ~ taille$population[which(taille$district == 1)],
  Base2$district==2 ~ taille$population[which(taille$district == 2)],
  Base2$district==3 ~ taille$population[which(taille$district == 3)],
  Base2$district==4 ~ taille$population[which(taille$district == 4)],
  Base2$district==5 ~ taille$population[which(taille$district == 5)],
  Base2$district==6 ~ taille$population[which(taille$district == 6)],
  Base2$district==7 ~ taille$population[which(taille$district == 7)],
  Base2$district==8 ~ taille$population[which(taille$district == 8)],

 ))
head(Base2)
```

```
## # A tibble: 6 × 13
##      id starttime           endtime             enumerator district   age
sex
##   <dbl> <dttm>              <dttm>                   <dbl>    <dbl> <int>
<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
## # ℹ 6 more variables: children_num <dbl>, intention <dbl>, destination
<dbl>,
## #   tran_age <chr>, groupe <dbl>, taille_Pop <dbl>
```

Calculer la durée de l'entretien et indiquer la durée moyenne de l'entretien par enquête

```
library(lubridate) # permet de manipuler les dates et les heures

##
## Attachement du package : 'lubridate'

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

library(dplyr)


# on convertit les temps en format "POSIXct"
Base2$debut_intev <- ymd_hms(Base2$starttime)
Base2$fin_interv<- ymd_hms(Base2$endtime
)

# en suite  la durée de l'entretien en secondes pour chaque enquêteur
Base2 <- Base2 %>%
  mutate(duree_entretien = as.numeric(difftime(fin_interv, debut_intev, units
= "secs")))
head(Base2)

## # A tibble: 6 × 16
##      id starttime           endtime             enumerator district   age
sex
##   <dbl> <dttm>              <dttm>                   <dbl>    <dbl> <int>
<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
## # ℹ 9 more variables: children_num <dbl>, intention <dbl>, destination
<dbl>,
## #   tran_age <chr>, groupe <dbl>, taille_Pop <dbl>, debut_intev <dttm>,
## #   fin_interv <dttm>, duree_entretien <dbl>
```

```
# nous calculons  la durée moyenne de l'entretien par enquêteur
duree_moyenne_par_enqueteur <- Base2 %>%
  group_by(enumerator)%>%
  summarize(duree_moyenne_entretien = mean(duree_entretien))

head(Base2)
```

```
## # A tibble: 6 × 16
##      id starttime           endtime             enumerator district   age
sex
##   <dbl> <dttm>              <dttm>                   <dbl>    <dbl> <int>
<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
## # ℹ 9 more variables: children_num <dbl>, intention <dbl>, destination
<dbl>,
## #   tran_age <chr>, groupe <dbl>, taille_Pop <dbl>, debut_intev <dttm>,
## #   fin_interv <dttm>, duree_entretien <dbl>
```

```
head(duree_moyenne_par_enqueteur)  # Afficher la durée moyenne par enquêteur
```

```
## # A tibble: 6 × 2
##   enumerator duree_moyenne_entretien
##        <dbl>                   <dbl>
## 1          1                   4089.
## 2          4                   2189
## 3          5                   2014.
## 4          6                   1551.
## 5          7                   2230.
## 6          8                   2408.
```

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

```
# Utiliser apply() pour renommer les variables avec le préfixe "endline_"
new_colnames <- apply(Base2, 2, function(col) {
  col <- paste("endline_", colnames(Base2)[which(colnames(Base2) == col)],
sep = "")
})
```

```
## Warning in colnames(Base2) == col: la taille d'un objet plus long n'est
pas
## multiple de la taille d'un objet plus court

## Warning in colnames(Base2) == col: la taille d'un objet plus long n'est
pas
## multiple de la taille d'un objet plus court

## Warning in colnames(Base2) == col: la taille d'un objet plus long n'est
pas
## multiple de la taille d'un objet plus court

## Warning in colnames(Base2) == col: la taille d'un objet plus long n'est
pas
## multiple de la taille d'un objet plus court

## Warning in colnames(Base2) == col: la taille d'un objet plus long n'est
pas
## multiple de la taille d'un objet plus court

## Warning in colnames(Base2) == col: la taille d'un objet plus long n'est
pas
## multiple de la taille d'un objet plus court

## Warning in colnames(Base2) == col: la taille d'un objet plus long n'est
pas
## multiple de la taille d'un objet plus court

## Warning in colnames(Base2) == col: la taille d'un objet plus long n'est
pas
## multiple de la taille d'un objet plus court

## Warning in colnames(Base2) == col: la taille d'un objet plus long n'est
pas
## multiple de la taille d'un objet plus court

## Warning in colnames(Base2) == col: la taille d'un objet plus long n'est
pas
## multiple de la taille d'un objet plus court

## Warning in colnames(Base2) == col: la taille d'un objet plus long n'est
pas
## multiple de la taille d'un objet plus court
```

```
## Warning in colnames(Base2) == col: la taille d'un objet plus long n'est
pas
## multiple de la taille d'un objet plus court

## Warning in colnames(Base2) == col: la taille d'un objet plus long n'est
pas
## multiple de la taille d'un objet plus court

## Warning in colnames(Base2) == col: la taille d'un objet plus long n'est
pas
## multiple de la taille d'un objet plus court

## Warning in colnames(Base2) == col: la taille d'un objet plus long n'est
pas
## multiple de la taille d'un objet plus court

## Warning in colnames(Base2) == col: la taille d'un objet plus long n'est
pas
## multiple de la taille d'un objet plus court
```

```
# Attribuer les nouveaux noms de colonnes à l'ensemble de données
#colnames(Base2) <- new_colnames

# Afficher le dataframe avec les nouvelles variables renommées
head(Base2)
```

```
## # A tibble: 6 × 16
##      id starttime           endtime             enumerator district   age
sex
##   <dbl> <dttm>              <dttm>                   <dbl>    <dbl> <int>
<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
## # ℹ 9 more variables: children_num <dbl>, intention <dbl>, destination
<dbl>,
## #   tran_age <chr>, groupe <dbl>, taille_Pop <dbl>, debut_intev <dttm>,
## #   fin_interv <dttm>, duree_entretien <dbl>
```

Analyse et visualisation des données

• Crée=z un tableau récapitulatif contenant l'âge moyen et le nombre moyen d'enfants par district.

```
library(readxl)
library(dplyr)
library(gtsummary)
library(flextable)
#Base2 <- readxl::read_excel("Base_Partie 2.xlsx",sheet = 1)
head(Base2)
```

```
## # A tibble: 6 × 16
##      id starttime           endtime             enumerator district   age
sex
##   <dbl> <dttm>              <dttm>                   <dbl>    <dbl> <int>
<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
## # ℹ 9 more variables: children_num <dbl>, intention <dbl>, destination
<dbl>,
## #   tran_age <chr>, groupe <dbl>, taille_Pop <dbl>, debut_intev <dttm>,
## #   fin_interv <dttm>, duree_entretien <dbl>
```

```
print(Base2$age)
```

```
##  [1]  33  43  28  24  29  22  21  20  21  20  24  17  23  19  24  18  18
20  38
## [20]  18  23  19  25  26  42  16  22  40  21  36  33  22  19  30  28  22
27  36
## [39]  22  34  32  25  18  21  26 999  15  28  25  40  18  20  35  19  22
23  22
## [58]  26  32  18  24  28  22  21  30  28  21  33  23  21  21  21  33  29
39  22
## [77]  33  30  26  27  26  38  29  18  22  25  30  26  29  20  34  25  19
23  24
## [96]  19  28
```

```
Base2$age[Base2$age>200] <- NA

head(Base2)
```

```
## # A tibble: 6 × 16
##      id starttime           endtime             enumerator district   age
sex
##   <dbl> <dttm>              <dttm>                   <dbl>    <dbl> <int>
<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
## # ℹ 9 more variables: children_num <dbl>, intention <dbl>, destination
<dbl>,
## #   tran_age <chr>, groupe <dbl>, taille_Pop <dbl>, debut_intev <dttm>,
## #   fin_interv <dttm>, duree_entretien <dbl>
```

```
tbl <- Base2 %>% tbl_summary(include =c(age,children_num,district),statistic
= list(age~"{mean}",
children_num ~ "{mean}"), by= district
,type=c(age,children_num)~"continuous",
label = list(age ~ "age_moyen" ,
children_num~"nombre_enfants_moyen"))%>%add_overall()
#%>%bold_labels() %>% italicize_levels()%>%as_flex_table()
tbl
```

Characteristic	**Overall**, N = 97[1]	**1**, N = 8[1]	**2**, N = 27[1]	**3**, N = 8[1]	**4**, N = 5[1]	**5**, N = 6[1]	**6**, N = 26[1]	**7**, N = 6[1]	**8**, N = 11[1]
age_moyen	26	30	27	26	26	24	23	28	25
Unknown	1	0	1	0	0	0	0	0	0
nombre_enfants_moyen	0.58	1.50	0.85	0.00	0.00	0.50	0.12	0.17	1.27

[1]Mean

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

```
Base2 %>% tbl_summary(include = c(age, sex), by = sex, statistic =
list(all_continuous() ~ "{mean}",
      all_categorical() ~ "{p}%"
    ),
    digits = list(
      all_continuous() ~ 2,
      all_categorical() ~ 1
    )
  ) %>%
  add_difference()
```

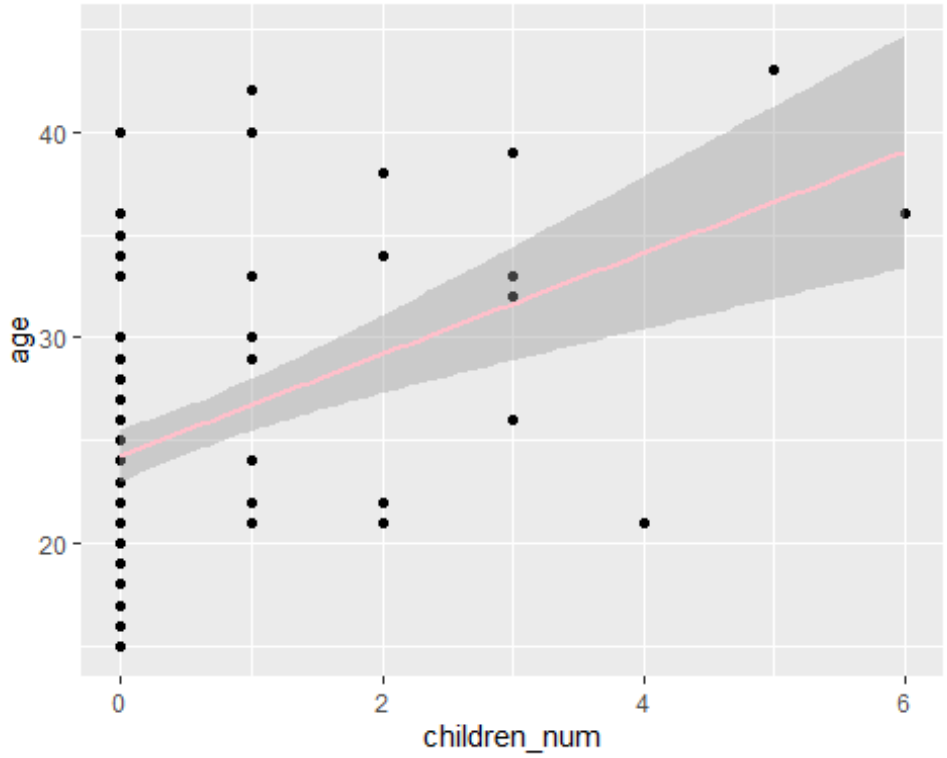
Characteristic	**0**, N = 86[1]	**1**, N = 11[1]	**Difference**[2]	**95% CI**[2,3]	**p-value**[2]

Characteristic	0, N = 86[1]	1, N = 11[1]	Difference[2]	95% CI[23]	p-value[2]
age	25.99	22.00	4.0	0.05, 7.9	0.047
Unknown	0	1			

[1]Mean
[2]Welch Two Sample t-test
[3]CI = Confidence Interval

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

```
library(ggplot2)
ggplot(Base2) +
aes(x = children_num, y = age) + # on ajoute le nom d'une variable à color
geom_point()+ # on ajoute une droite de rég. non paramétrique
geom_smooth(method = "lm", # on ajoute une droite de régression linéaire
col = "pink")

## `geom_smooth()` using formula = 'y ~ x'

## Warning: Removed 1 rows containing non-finite values (`stat_smooth()`).

## Warning: Removed 1 rows containing missing values (`geom_point()`).
```



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

• Créez un tableau de régression avec 3 modèles. La variable de résultat est toujours "intention". Modèle A : Modèle vide - Effet du traitement sur les intentions. Modèle B : Effet du traitement sur les intentions en tenant compte de l'âge et du sexe. Modèle C : Identique

au modèle B mais en contrôlant le district. Les résultats des trois modèles doivent être affichés dans un seul tableau.