

TP1

GADO Seman Giovanni Jocelyn ISEP 3

2024-04-06

Importation et mise en forme

Importation

```
#Nos données étant sous format xlsx, nous allons utiliser la fonction  
#read_excel du package readxl  
projet <- readxl::read_excel("Base_Projet.xlsx")
```

Nombre de lignes et de colonnes

```
# Nombre de lignes  
nb_lignes <- nrow(projet)  
# Nombre de colonnes (variables)  
nb_colonnes <- ncol(projet)
```

La base comporte 250 lignes et 33 colonnes

Vérification des valeurs manquantes et identification

```
# Vérifions s'il y a des valeurs manquantes pour la variable "key"  
val_manquantes <- is.na(projet$key)  
  
# Identifions les valeurs manquantes existantes dans la base  
# Pour ce faire nous allons utiliser la fonction which du package base qui  
#renvoie la position des PME pour laquelle l'identifiant est une valeur  
#manquante  
indices_lignes_manquantes <- which(val_manquantes)
```

Création des variables

Renommer les variables

```
library(magrittr) # Package nécessaire à l'utilisation du pipe %>%  
  
## Warning: le package 'magrittr' a été compilé avec la version R 4.2.3  
projet <- projet %>%  
  dplyr::rename(region = q1, departement = q2, sexe = q23)
```

Création

```
# Création de la variable sexe_2
projet <- projet %>%
  dplyr::mutate(sexe_2 = if_else(sexe == "Femme", 1, 0))

# Création du dataframe langues
langues <- projet %>%
  select(key, starts_with("q24a_"))

# Création de la variable parle
langues <- langues %>%
  dplyr::mutate(parle = rowSums(select(., starts_with("q24a_"))))

# Sélection de key et parle
langues <- langues %>%
  select(key, parle)

# Merger les bases projet et parle
projet = left_join(projet,langues,by = "key")
```

Analyse descriptive

Analyse univariée

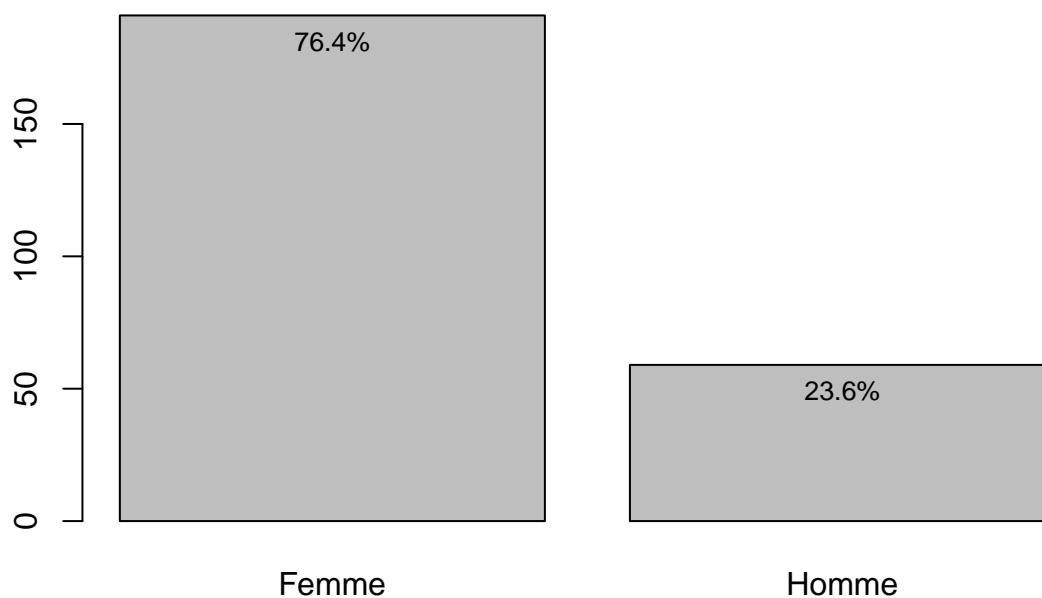
```
fonction_uni <- function(donnees, var) {
  cat(paste0("\nRépartition des PME suivant le ",var, "\n"))
  # Récupération des modalités uniques de la variable
  modalites <- unique(donnees[[var]])
  # Boucle sur les modalités
  for (mod in modalites) {
    cat(mod, ":", table(donnees[[var]])[mod], "\n")
  }

  cat("\nGraphiques:\n")
  freq <- table(donnees[[var]])
  bp = barplot(freq, main=paste("Histogramme de la répartition des PME suivant le", var))
  proportions <- prop.table(freq)
  text(x = bp, y = freq, labels = paste0(round(proportions * 100, 1), "%"), pos = 1, cex = 0.8)
}
```

```
# Répartition des PME suivant le sexe
fonction_uni(projet,"sexe")
```

```
##
## Répartition des PME suivant le sexe
## Femme : 191
## Homme : 59
##
## Graphiques:
```

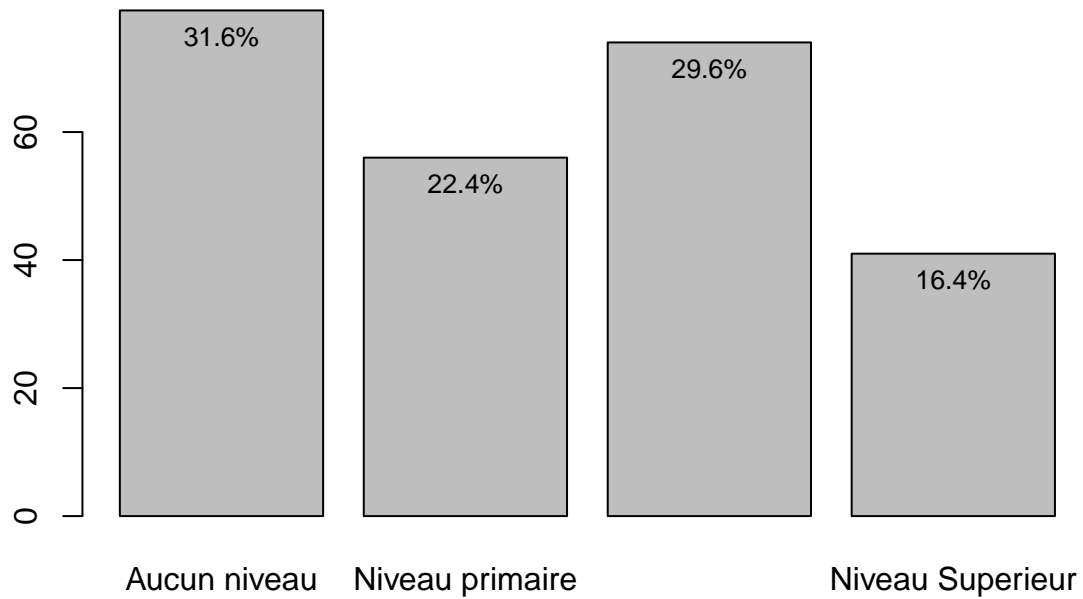
Histogramme de la répartition des PME suivant le sexe



```
# Répartition des PME suivant le niveau d'instruction  
fonction_uni(projet, "q25")
```

```
##  
## Répartition des PME suivant le q25  
## Aucun niveau : 79  
## Niveau primaire : 56  
## Niveau secondaire : 74  
## Niveau Supérieur : 41  
##  
## Graphiques:
```

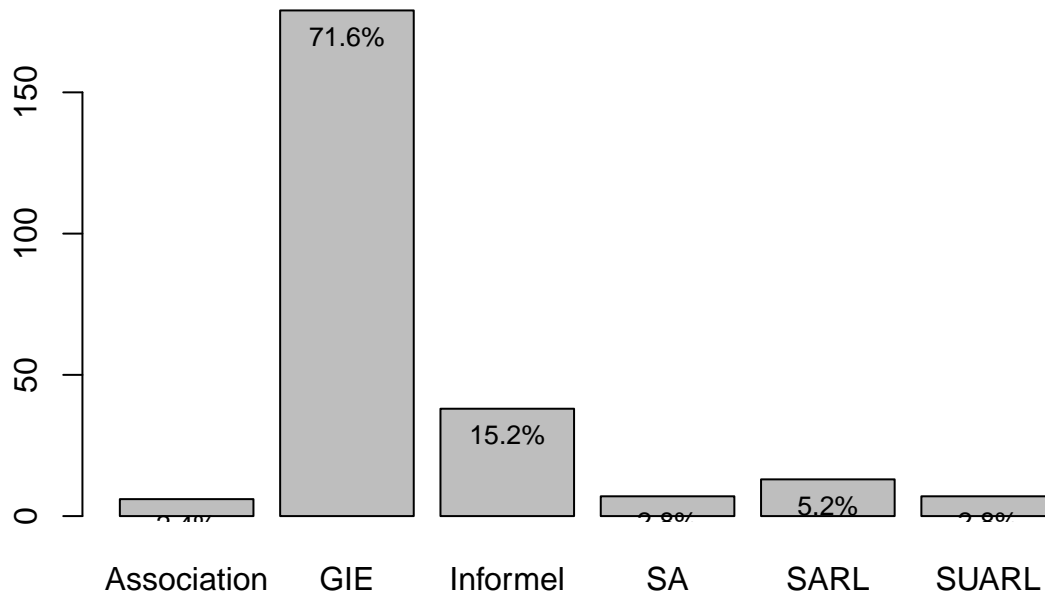
Histogramme de la répartition des PME suivant le q25



```
# Répartition des PME suivant le statut juridique  
fonction_uni(projet,"q12")
```

```
##  
## Répartition des PME suivant le q12  
## GIE : 179  
## Informel : 38  
## SUARL : 7  
## SARL : 13  
## Association : 6  
## SA : 7  
##  
## Graphiques:
```

Histogramme de la répartition des PME suivant le q12



Analyse bivariée

```

fonction_bi <- function(donnees, var1, var2) {
  cat(paste0("\nRépartition des PME suivant le ", var1, " et ", var2, "\n"))
  # Récupération des modalités uniques de la variable var1
  modalites_var1 <- unique(donnees[[var1]])

  # Boucle sur les modalités de var1
  for (mod_var1 in modalites_var1) {
    cat(paste("\n", mod_var1, ":\n"))

    # Sous-ensemble des données correspondant à la modalité de var1
    donnees_sous_ensemble <- donnees[donnees[[var1]] == mod_var1, ]

    # Récupération des modalités uniques de la variable var2 dans le sous-ensemble de données
    modalites_var2 <- unique(donnees_sous_ensemble[[var2]])

    # Boucle sur les modalités de var2 dans le sous-ensemble de données
    for (mod_var2 in modalites_var2) {
      cat(mod_var2, ":", sum(donnees_sous_ensemble[[var2]] == mod_var2), "\n")
    }
  }

  cat("\nGraphiques:\n")
}

```

```

# Calcul des fréquences pour les catégories de var1
freq_croisees <- table(donnees[[var1]], donnees[[var2]])

# Affichage des fréquences croisées
print(freq_croisees)

cat("\nGraphiques:\n")

# Création du graphique à barres pour var1
bp_var1 <- barplot(freq_croisees, main=paste("Histogramme de la répartition des PME suivant le", var1))

# Calcul des proportions pour les barres du graphique
proportions <- prop.table(freq_croisees, margin = 1)

# Affichage des proportions au-dessus des barres pour chaque modalité de var1
for (i in 1:nrow(proportions)) {
  text(x = bp_var1[i], y = freq_croisees[i, ], labels = paste0(round(proportions[i, ] * 100, 1), "%"))
}

# Répartition des PME suivant le sexe et le statut juridique
fonction_bi(projet, "sexe", "q12")

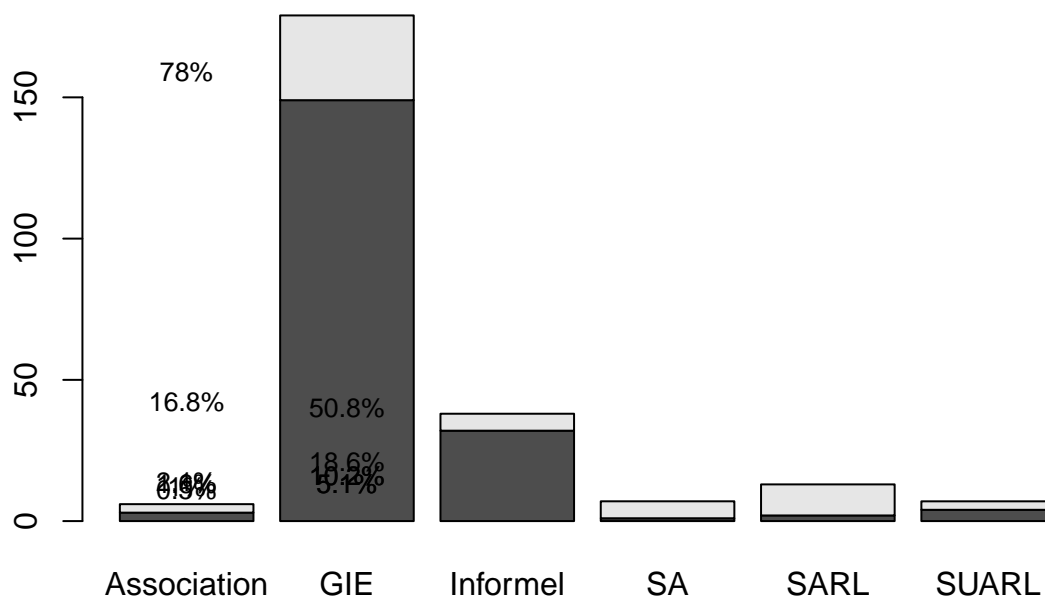
```

```

##
## Répartition des PME suivant le sexe et q12
##
## Femme :
## GIE : 149
## Informel : 32
## Association : 3
## SARL : 2
## SA : 1
## SUARL : 4
##
## Homme :
## GIE : 30
## SUARL : 3
## SARL : 11
## Informel : 6
## Association : 3
## SA : 6
##
## Graphiques:
##
##      Association GIE Informel  SA SARL SUARL
##  Femme          3 149        32   1   2    4
##  Homme          3  30         6   6  11    3
##
## Graphiques:

```

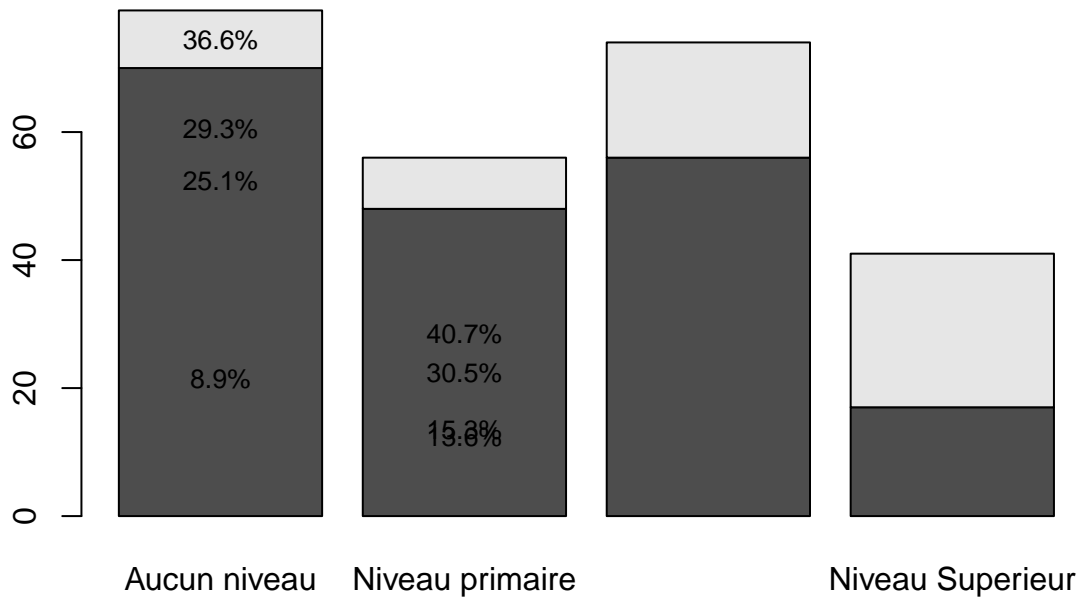
Histogramme de la répartition des PME suivant le sexe et q12



```
# Répartition des PME suivant le sexe et le niveau d'instruction
fonction_bi(projet,"sexe","q25")
```

```
##
## Répartition des PME suivant le sexe et q25
##
## Femme :
## Aucun niveau : 70
## Niveau primaire : 48
## Niveau secondaire : 56
## Niveau Supérieur : 17
##
## Homme :
## Niveau secondaire : 18
## Niveau Supérieur : 24
## Aucun niveau : 9
## Niveau primaire : 8
##
## Graphiques:
##
##      Aucun niveau Niveau primaire Niveau secondaire Niveau Supérieur
## Femme           70           48           56           17
## Homme            9            8           18           24
##
## Graphiques:
```

Histogramme de la répartition des PME suivant le sexe et q25



Cartographie

Transformation du dataframe en données spatiales

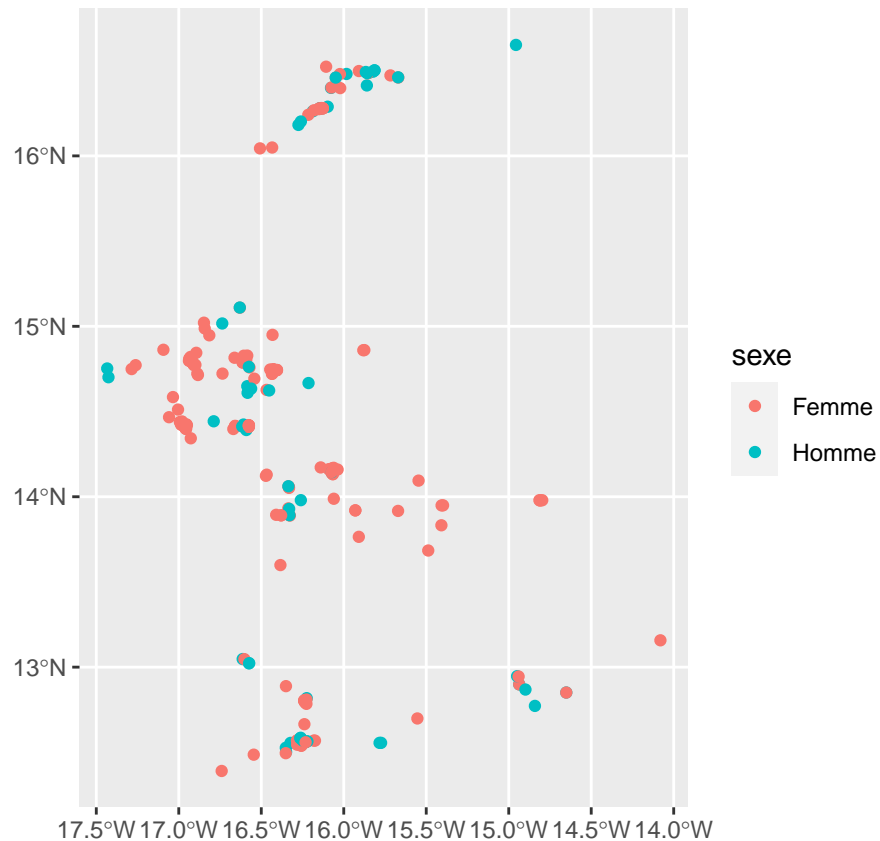
```
# Pour ce faire, nous allons utiliser le ppackage sf utilisé dans le traitement
#des données spatiales.
# L'argument coords regroupe la longitude et la latitude de la base de données
#initiale et l'argument crs représente le système de coordonnées

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

Création de la carte

```
carte <- ggplot2::ggplot() +
  ggspatial::geom_sf(data = projet_map, ggspatial::aes(col = sexe), size = 1.5)
# Nous avons d'abord créer une première couche vide ggplot2::ggplot() à laquelle
#on superpose
#la couche contenant les informations spatiales

carte
```

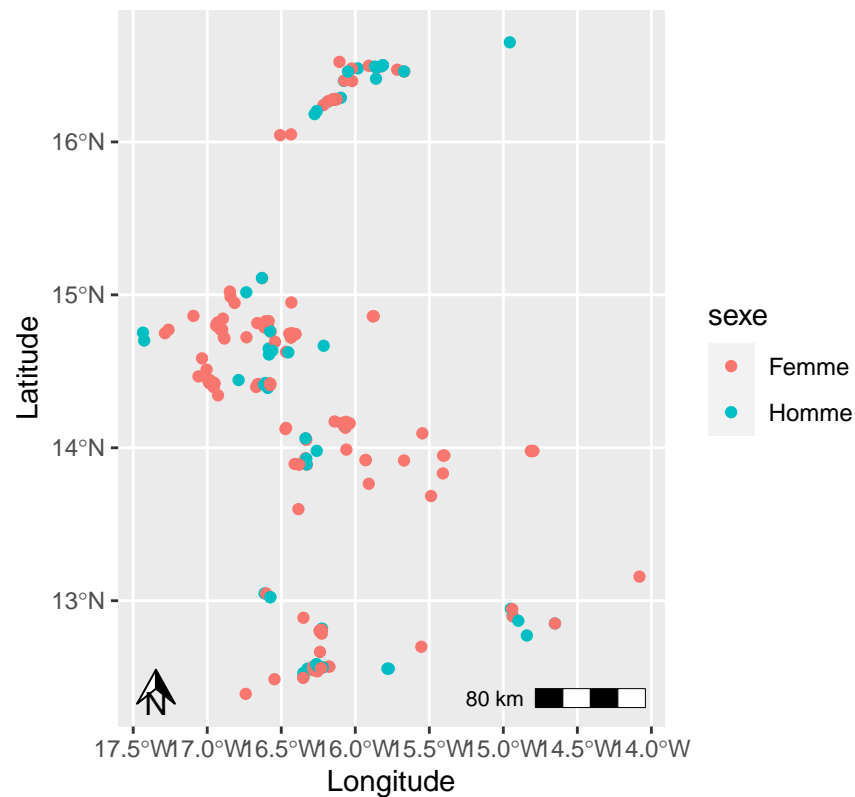



Ajout des axes, du titre, de la légende, de l'orientation et de l'échelle

```
carte_2 <- ggplot2::ggplot() +
  ggspatial::geom_sf(data = projet_map, ggspatial::aes(col = sexe), size = 1.5)+
  ggplot2::labs(title = "Répartition des PME par sexe", x = "Longitude",
    y = "Latitude") + # Ajout des axes et du titre
  ggspatial::annotation_scale(location = "br", line_width = .2) + # Ajout de
    #l'échelle
  ggspatial::annotation_north_arrow(location = "bl",
    height = grid::unit(0.5, "cm"), width = grid::unit(0.5, "cm")) # Ajout de
    #l'orientation

print(carte_2)
```

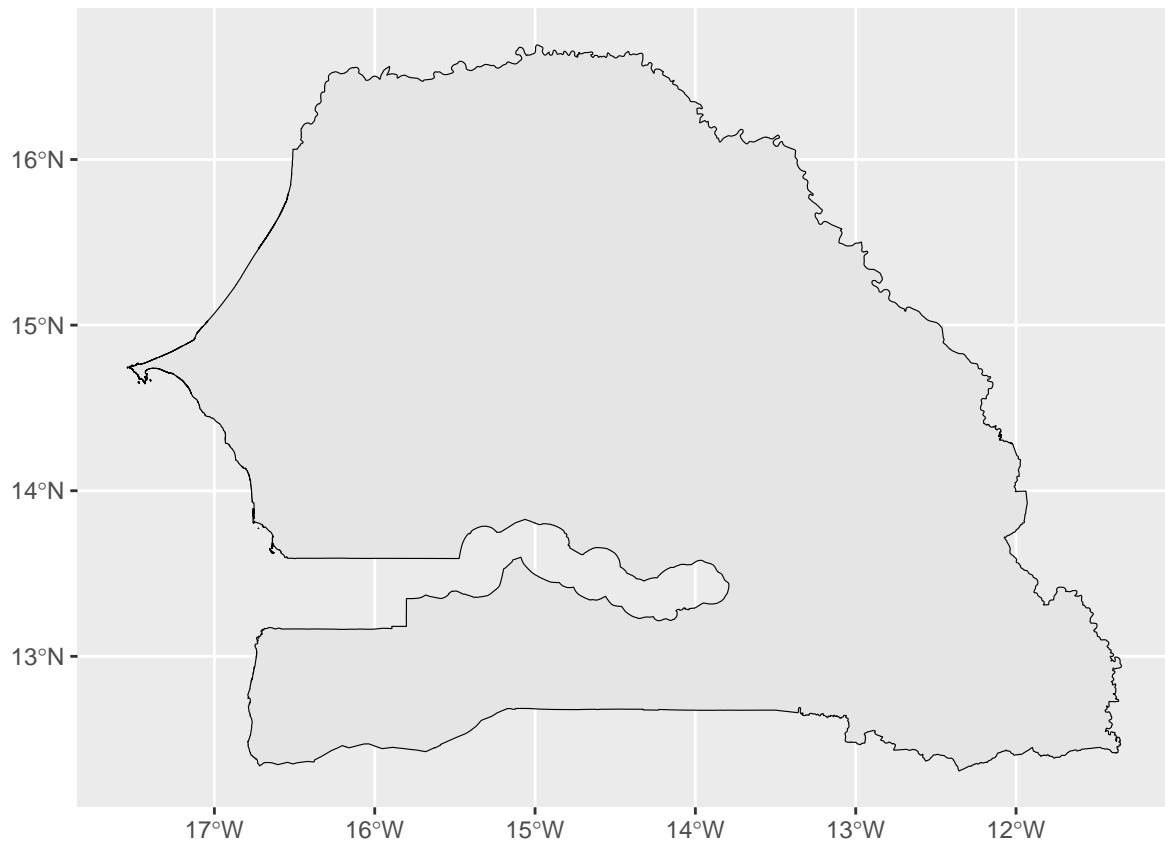
Répartition des PME par sexe



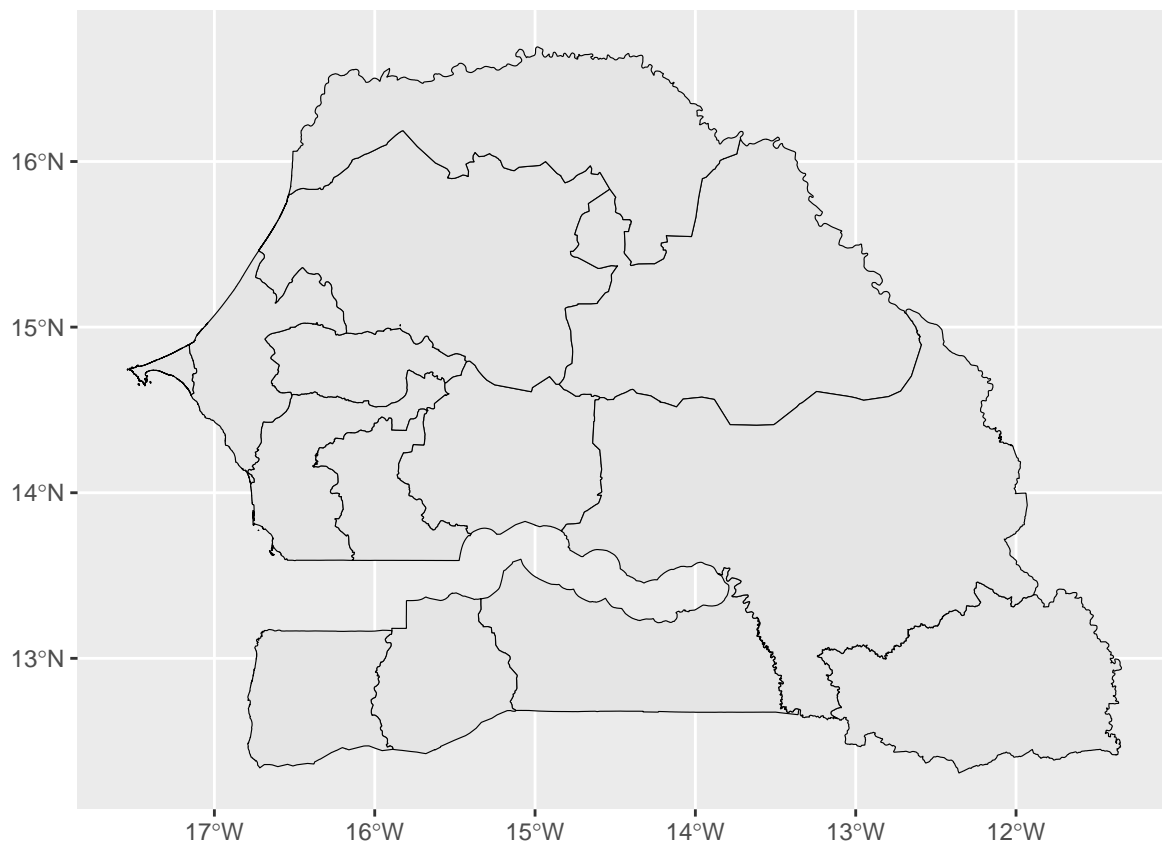
Importation des différents niveaux et affichage des différentes cartes

```
# Importation
niveau_0 = sf::read_sf("gadm41_SEN_shp/gadm41_SEN_0.shp")
niveau_1 = sf::read_sf("gadm41_SEN_shp/gadm41_SEN_1.shp")
niveau_2 = sf::read_sf("gadm41_SEN_shp/gadm41_SEN_2.shp")
niveau_3 = sf::read_sf("gadm41_SEN_shp/gadm41_SEN_3.shp")

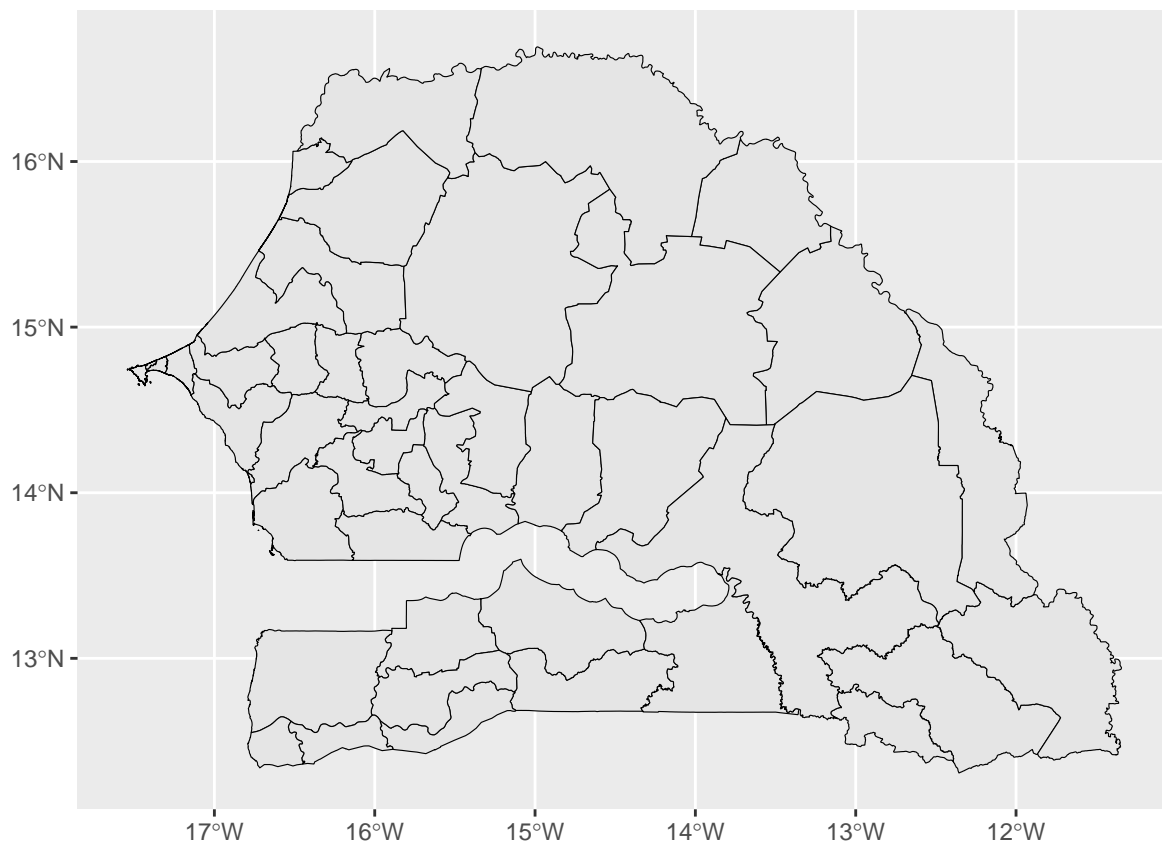
# Affichage niveau_0
carte_niveau_0 <- ggplot2::ggplot() +
  ggspatial::geom_sf(data=niveau_0, color="black",size=0.3)
carte_niveau_0
```



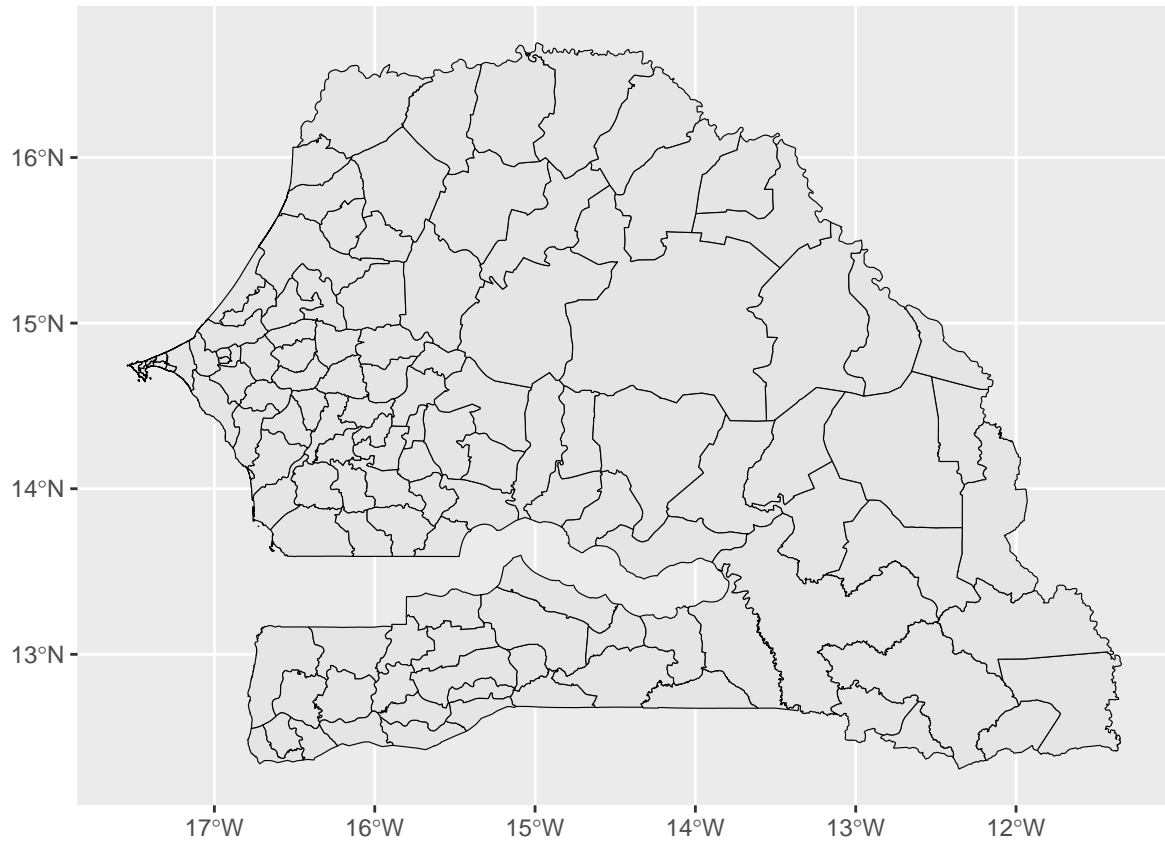
```
# Affichage niveau_1
carte_niveau_1 <- ggplot2::ggplot()+
  ggspatial::geom_sf(data=niveau_1, color="black",size=0.3)
carte_niveau_1
```



```
# Affichage niveau_2
carte_niveau_2 <- ggplot2::ggplot()+
  ggspatial::geom_sf(data=niveau_2,color="black",size=0.3)
carte_niveau_2
```



```
# Affichage niveau_3
carte_niveau_3 <- ggplot2::ggplot()+
  ggspatial::geom_sf(data=niveau_3,color="black",size=0.3)
carte_niveau_3
```

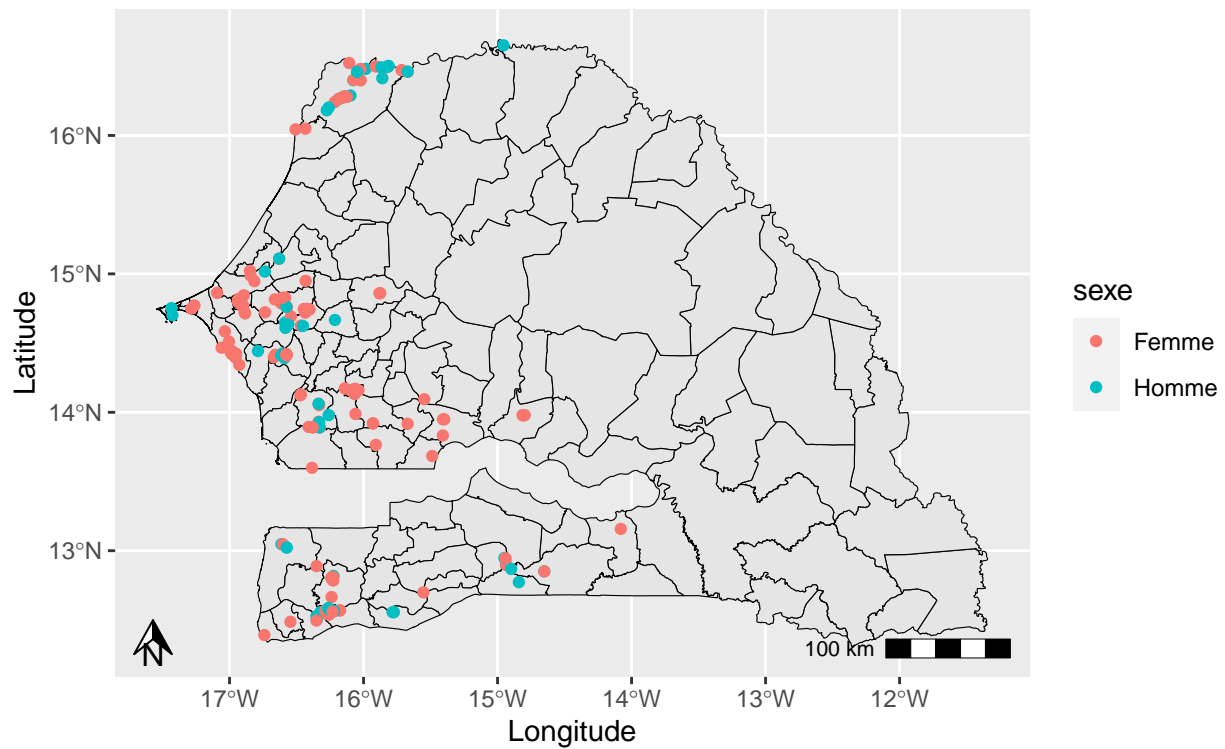


Représentation spatiale des PME suivant le sexe

```
re_1 = ggplot2::ggplot() + ggspatial::geom_sf(data=niveau_3,color="black",size=0.3) +
  ggspatial::geom_sf(data = projet_map, ggspatial::aes(col = sexe), size = 1.5) +
  ggplot2::labs(title = "Répartition des PME par sexe", x = "Longitude",
    y = "Latitude") +
  ggspatial::annotation_scale(location = "br", line_width = .2) +
  ggspatial::annotation_north_arrow(location = "bl",
    height = grid::unit(0.5, "cm"), width = grid::unit(0.5, "cm"))
```

re_1

Répartition des PME par sexe

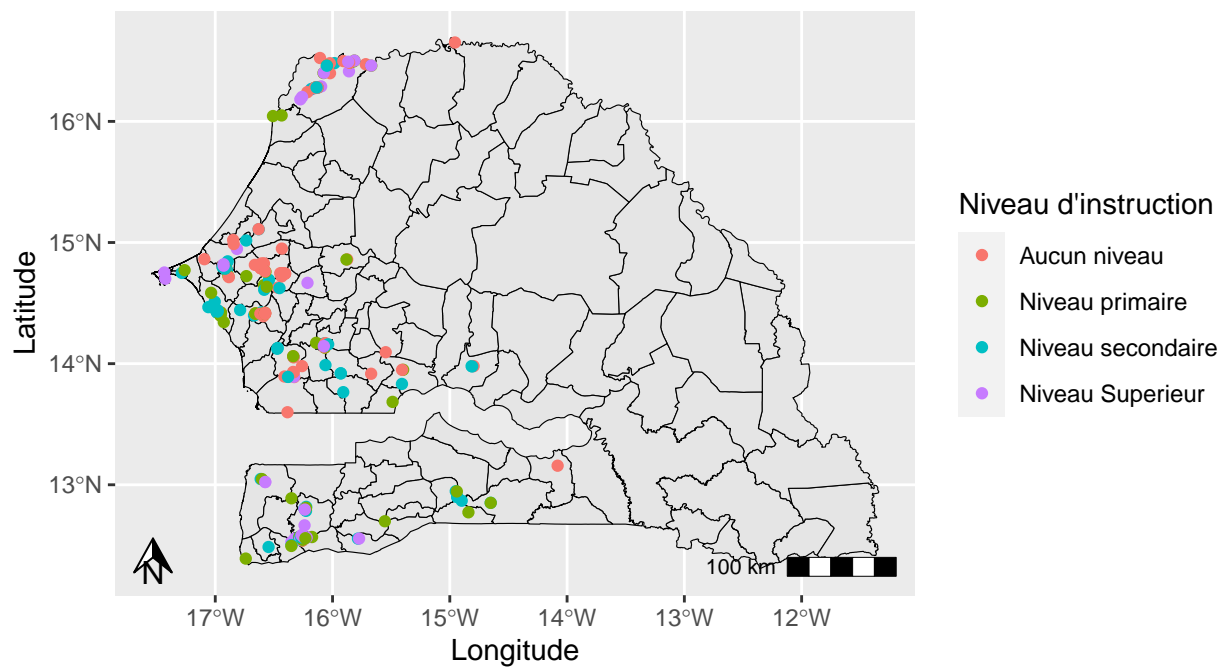


Représentation spatiale des PME suivant le niveau d'instruction

```
re_2 = ggplot2::ggplot() + ggspatial::geom_sf(data=niveau_3,color="black",size=0.3) +
  ggspatial::geom_sf(data = projet_map, ggspatial::aes(col = q25), size = 1.5) +
  ggplot2::labs(title = "Répartition des PME par niveau d'instruction",
    x = "Longitude", y = "Latitude", color = "Niveau d'instruction" ) +
  ggspatial::annotation_scale(location = "br", line_width = .2) +
  ggspatial::annotation_north_arrow(location = "bl",
    height = grid::unit(0.5, "cm"), width = grid::unit(0.5, "cm"))
```

re_2

Répartition des PME par niveau d'instruction



Représentation au choix (par statut juridique)

```
re_3 = ggplot2::ggplot() + ggspatial::geom_sf(data=niveau_3,color="black",size=0.3) +
  ggspatial::geom_sf(data = projet_map, ggspatial::aes(col = q12), size = 1.5) +
  ggplot2::labs(title = "Répartition des PME par statut juridique",
    x = "Longitude",y = "Latitude", color = "Statut juridique" ) +
  ggspatial::annotation_scale(location = "br", line_width = .2) +
  ggspatial::annotation_north_arrow(location = "bl",
    height = grid::unit(0.5, "cm"), width = grid::unit(0.5, "cm"))
```

re_3

Répartition des PME par statut juridique

