REPUBLIQUE DU SENEGAL

Un Peuple-Un But-Une Foi

Ministère de l'Economie du Plan et de la Coopération

ANSD
Agence Nationale de la Statistique et de la Démographie

ensae

PROJET R : Seconde partie

24 JUILLET 2023

By KABASSINA Gnimdou Ange
Elève Ingénieure Statisticienne Economiste

Sommaire

Importation des données

```
# Importation de la base sous format data.frame dans l'objet projet
projet2 = data.frame(readxl::read_excel("Base_Partie 2.xlsx"))
```

Nettoyage et gestion des données

Renommons “country_destination” en “destination”

```
# renommons "country_destination" en "destination"
library(dplyr)
# Importons `magrittr` pour acceder aux pipes
library(magrittr)
projet2 = projet2  %>% dplyr::rename("destination" = "country_destination")
```

Definir les vleurs négatives de “destination” comme manquantes

```
projet2 = projet2  %>% dplyr::mutate(destination = ifelse(destination<0, NA, destination))
```

Création une nouvelle variable contenant des tranches d’âge de 5 ans en utilisant la variable “age”.

Tout d’abord voyons comment se présente la variable “âge” pour definir le nombre de classe d’âge à créer, par où commencer et où s’arrêter.

```
summary(projet2$age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   15.00   21.00   24.00   35.61   29.00  999.00
```

On remarque que le maximum est 999 ce qui est impossible car aucune personne ne peut présenter ce âge. Cette valeur représente en réamité une valeur manquante que nous allons remplacer par la médiane pour conserver la tendance générale et particulièrement centrale de la distribution car la médiane est robuste (ou peu sensible) aux valeurs extrêmes et aberrantes.

```
# remplacer les 999 par la médiane
projet2 = projet2  %>% dplyr::mutate(age = ifelse(age==999, median(age), age))

# voyons la nouvelle distribution
summary(projet2$age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   15.00   21.00   24.00   25.56   29.00   43.00
```

Imputation réuissie !
Nous pouvons donc créer nos classes d'âge qui seront: "entre 15 et 20" equivalent à [15,20], "entre 20 et 25" equivalent à]20,25], "entre 25 et 30" equivalent à]25,30], "entre 30 et 35" equivalent à]30,35], "entre 35 et 40" equivalent à]35,40] et "entre 40 et 45" equivalent à]40,45].

```
# création des tranches d'âges
bornes = c(14, 20, 25, 30, 35, 40, 45)# bornes des intervales
etiquettes = c("entre 15 et 20", "entre 20 et 25", "entre 25 et 30",
               "entre 30 et 35","entre 35 et 40", "entre 40 et 45") # valeurs de la nouvelle variable c
projet2 = projet2 %>% dplyr::mutate(tranche_age=cut(age, breaks = bornes, labels = etiquettes))
```

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

Pour connaître le nombre d'entrétiens réalisé par chaque agent, nous allons séparé la base en sous base suivant l'identifiant de l'agent (groupeby) afin d'obtenir les individus interviewés par chaque agent. Ensuite pour connaître le nombre d'interview pour un agent, il suffira de compter le nombre de lignes de la sous base correspondant à l'agent et de le stocker dans chacune des lignes de la sous base.

```
# Création d'une nouvelle variable contenant le nombre d'entretiens
# réalisés par chaque agent recenseur
projet2 <- projet2 %>%
  dplyr::group_by(enumerator) %>% # séparer en sous groupes suivant l'identifiant de l'agent
  dplyr::mutate(nb_entretiens = dplyr::n()) # calculer le nombre de ligne de chaque sous groupe et l'aj
```

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

Pour créer une telle variable aléatoire, nous ferons appel à la fonction sample() puis à rowwise() pour faire un choix ligne par ligne c'est à dire pour chaque individu, on choisis aléatoirement entre 0 ou 1 puis on stocke le choix dans une nouvelle variable.Après création, les valeurs de cette nouvvelle variables seront labellisées comme suit : 1 = groupe de traitement et 0 = groupe de controle.

```
# Création d'une nouvelle variable qui affecte aléatoirement chaque répondant à un groupe de traitement
set.seed(100) # fixer l'aléa
projet2 <- projet2 %>% dplyr:: rowwise()  %>%  # pour faire l'aléa ligne par ligne
  mutate(g_traitement = sample(c(0,1),size=1))

# labelisation des valeurs de la nouvelle variable créée
# 1 = groupe de traitement et 0 = groupe de controle.
projet2 <- projet2 %>%
  mutate(g_traitement = factor(g_traitement, levels = c(0, 1),
                               labels = c("groupe de contrôle",
                                          "roupe de traitement")))
```

Fusion de la taille de la population de chaque district (feuille 2) avec l'ensemble de données (feuille 1)

Avant la fusion, nous allons d'abord importer la deuxième base contenant les informations sur la population des districts.

Importation de la base sur la population des districts Nous allons utiliser read_excel comme précédement mais spécifier le nom de la feuille contenant les données qui nous interesse dans le paramètre "sheet=".

```
## importation de la base sur la population des districts
district = data.frame(readxl::read_excel("Base_Partie 2.xlsx",sheet = "district"))
```

Fusion Pour la fusion, Nous allons utiliser merge comme dans la partie 1 avec pour clé de fusion cette fois-ci la variable "district".

```
## Fusion
projet2 = merge(projet2, district, by="district")
```

Calcul de la durée de l'entretien et de la durée moyenne de l'entretien par enquêteur.

Pour la durée de l'entretien, nous créerons une nouvelle variable qui sera la soustration de l'heure de début de l'heure de fin.

```
## calcul durée de l'entretien
projet2 <- projet2 %>%
  mutate(dure_entretien = endtime - starttime)
```

Pour la durée moyenne de l'entretien par agent, nous allont utiliser la fonction "aggregate()" qui est une fonction de base de R permettant de regrouper les individus d'une base suivant un critère (comme avec groupeby), d'y appliquer des fonction et de sortir un dataframe contenant le resultat de la fonction pour chaque groupe. Dans notre cas cela nous renverra donc un dataframe contenant la durée moyenne de l'entretien suivant l'identifiant des enquêteurs.

```
# durée moyenne de l'entretien par agent
# Calculer la durée moyenne de l'entretien par enquêteur
dure_moyenne <- projet2 %>% aggregate(dure_entretien ~ enumerator, FUN = mean)

# Afficher la durée moyenne de l'entretien par enquêteur sous forme de tableau de format Markdown
knitr::kable(dure_moyenne, format = "markdown")
```

enumerator	dure_entretien
1	68.14667 mins
4	36.48333 mins
5	33.55833 mins
6	25.84667 mins
7	37.16429 mins
8	40.13056 mins
9	114.76667 mins
10	55.27667 mins
11	33.48333 mins
12	48.16667 mins
13	31.59583 mins
14	25.56111 mins
15	28.65000 mins
17	29.28611 mins

enumerator	dure_entretien
18	36.85833 mins
20	28.76852 mins

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

Pour cette partie, nous avons plusieurs possibilités:

- Méthode1 : Nous allons d'abord recupérer les noms de toutes les variables dans un vecteur old_names. Ensuite on fera une boucle qui parcourt les élément de ce vecteur et qui à chaque fois remplace cet nom dans la base par la concatenation du préfixe "endline_" et de l'ancien nom, la concatenation étant assurée par la fonction paste().

Méthode 2: Utiliser la fonction lapply() ou list apply qui applique une fonction sur une liste d'éléments et renvoie le résultats sous forme de liste.

La méthode 2 étant plus simple et directe, nous l'utiliserons plutôt que la prémière.

```
# Renommons toutes les variables de l'ensemble de données en ajoutant le préfixe "endline_" à l'aide d'

colnames(projet2) <- lapply(colnames(projet2), function(x) paste("endline", x, sep = "_"))
```

Analyse et visualisation des données

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

```
## Tableau récapitulatif contenant l'âge moyen et le nombre moyen d'enfants par district.
library(gtsummary) # charger le package gtsummary

# faire le tableau avec gtsummary
projet2 %>%
  dplyr::select(endline_district, endline_age,endline_children_num) %>%
  gtsummary::tbl_summary(
    ## paramètres de tbl_summary
    by = endline_district,
    ## variables qui forme les groupes: sexe
    label = list(
      endline_district = "District",
      endline_age = "Age du répondant",
      endline_children_num = "Nombre d'enfants du répondant"
    ),
    ## ajouter les étiquettes des variables
    percent = "column",
    type = list(endline_children_num ~ 'continuous'),
    ## Type de pourcentage affichés dans le tableau
    statistic = c(endline_age,endline_children_num) ~ "{mean}",
    ## statistiques à calculer pour l'âge du dirigeant (moyenne)
```

```
  ) %>%
  add_overall() %>%
  ## ajouter les statistiques sur la base totale (non par groupe)
  add_stat_label() %>%
  ## afficher une colonne qui signifie les statistiques calculées et leur format d'affichage. Ex: mean
  bold_labels()
```

Characteristic	**Overall**, N = 97	**1**, N = 8	**2**, N = 27	**3**, N = 8	**4**, N = 5	**5**, N = 6	**6**, N = 26	**7**, N = 6	**8**, N = 11
Age du répondant, Mean	26	30	27	26	26	24	23	28	25
Nombre d'enfants du répondant, Mean	0.58	1.50	0.85	0.00	0.00	0.50	0.12	0.17	1.27

```
  ## Mettre le nom des variables en gras
```

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

Commençons par visualiser la distribution de l'âge suivant le sexe.

```
# statistiques descriptives : distribution de l'âge suivant le sexe avec gtsummary

projet2 %>%
  dplyr::select(endline_sex, endline_age) %>%
  gtsummary::tbl_summary(
    ## paramètres de tbl_summary
    by = endline_sex,
    ## variables qui forme les groupes: sexe
    label = list(
      endline_sex = "Sexe du répondant",
      endline_age = "Age du répondant"
    ),
    ## ajouter les étiquettes des variables
    statistic = endline_age ~ "
    {median}
    {mean}
    {sd}",
    ## statistiques à calculer pour l'âge du dirigeant (médiane moyenne écart type)
    ## formatage et nomination de la variable "valeur manquante"
  ) %>%
  add_overall() %>%
  ## ajouter les statistiques sur la base totale (non par groupe)
  add_difference() %>%
  ## afficher la différence entre les groupes, le test de significativité de la différence
  add_stat_label()
```

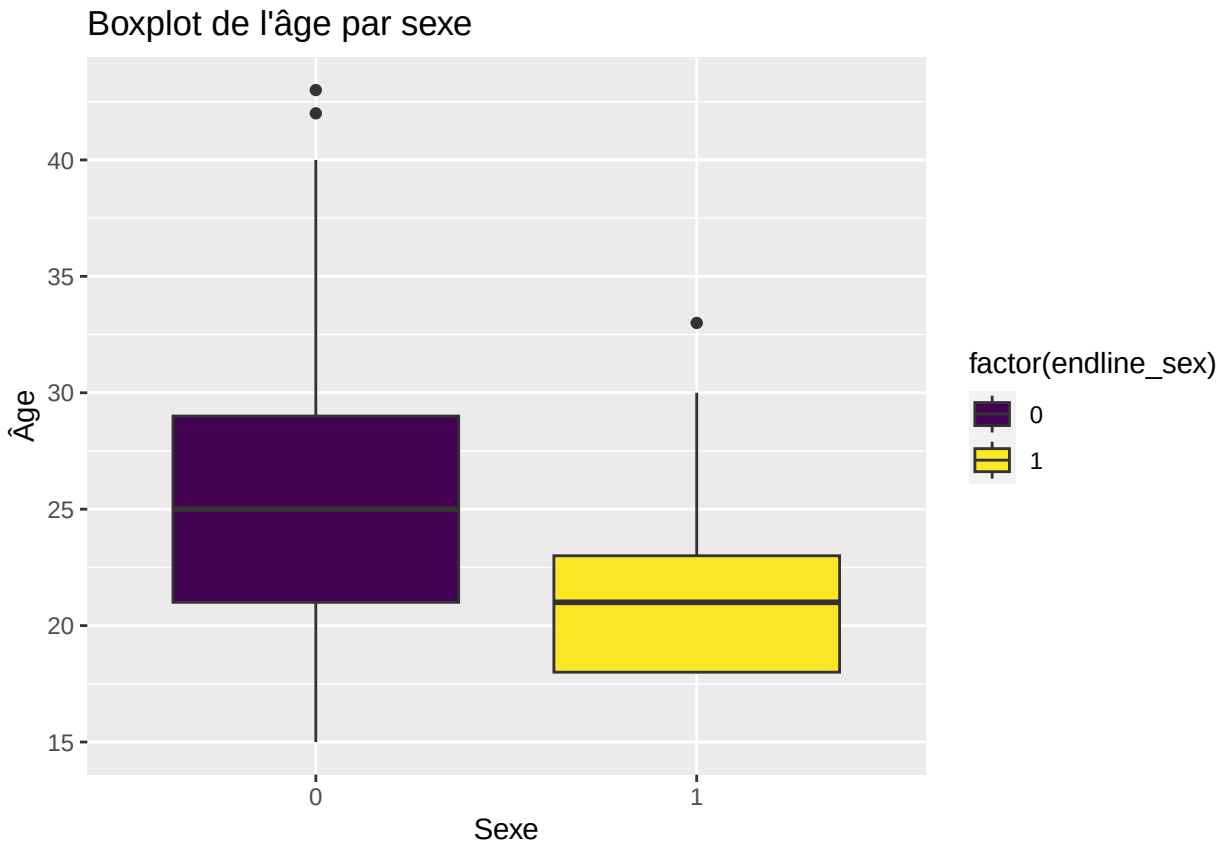
Characteristic	Overall, N = 97	0, N = 86	1, N = 11	Difference	95% CI	p-value
Age du répondant,						

```
Median
Mean
SD |         24
```

26 6 | 25 26 6 | 21 22 5 | 3.8 | 0.23, 7.4 | 0.039 |

```
## afficher une colonne qui signifie les statistiques calculées et leur format d'affichage. Ex: mean (s
```

```
# représentation graphique : âge suivant le sexe
library(ggplot2)
library(viridis)
projet2 %>%
  ggplot(aes(x = factor(endline_sex), y = endline_age, fill = factor(endline_sex))) +
  geom_boxplot() +
  scale_fill_viridis_d() +  # Utiliser la palette "viridis"
  labs(x = "Sexe", y = "Âge", title = "Boxplot de l'âge par sexe")
```



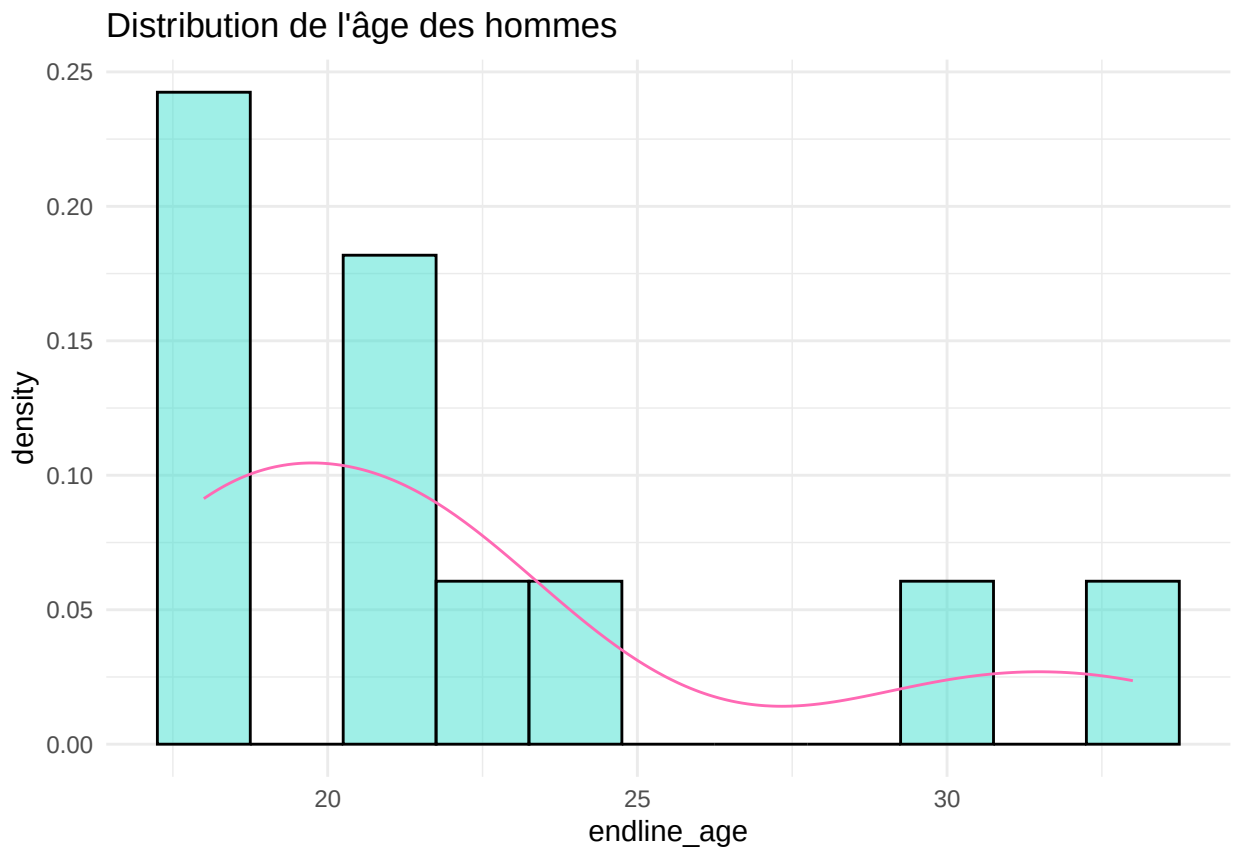
La différence observée entre les deux répartition est t-elle sstatistiquement significative ?

On pourrait se limiter aux résultats du tableau de gtsummary qui propose un test de student pour évaluer la significativité de différence entre la moyenne des deux groupe et conclure que puisque la p-value est inférieure à 5%, les moyennes sont différentes mais ce test est-il valide ?

En statistique, étant donné deux échantillons , il existent des tests dit paramétriques tels que le tests t de student pour déterminer si les moyennes de ces échantillons diffèrent l'une de l'autre significativement.

Nous utiliserons donc le test t de student pour mesurer la différence entre l'âge des hommes et celui des femmes. Mais d'abord notont que ce test est dit paramétrique ce qui signifie que nos échantillons sur lequels il sera appliqué doivent respecter certainnes conditions à savoir la normalité de la distribution. Nous allons donc tester si la distribution de l'âge suivant le sexe suit une loi normale à partir d'une représentation graphique et du test de normalité de shapiro.

```
# Tester si la différence d'âge entre les sexes est statistiquement significative au niveau de 5 %.
## extraire la distribution de lâge des femmes et celle des hommes
age_homme = projet2 %>% dplyr::filter(endline_sex==1)%>%
  select(endline_age) # distribution de l'âge des hommes
age_femme = projet2 %>% dplyr::filter(endline_sex==0) %>%
  select(endline_age) # distribution de l'âge des femmes
## Test de normalité
## test graphique : visualisation
ggplot(age_homme, aes(x = endline_age )) +
  geom_histogram(aes(y = ..density..), binwidth = 1.5, color = "black", fill = "turquoise", alpha = 0.5)
  geom_density(color = "hotpink") +
  labs(title = "Distribution de l'âge des hommes") +
  theme_minimal()
```



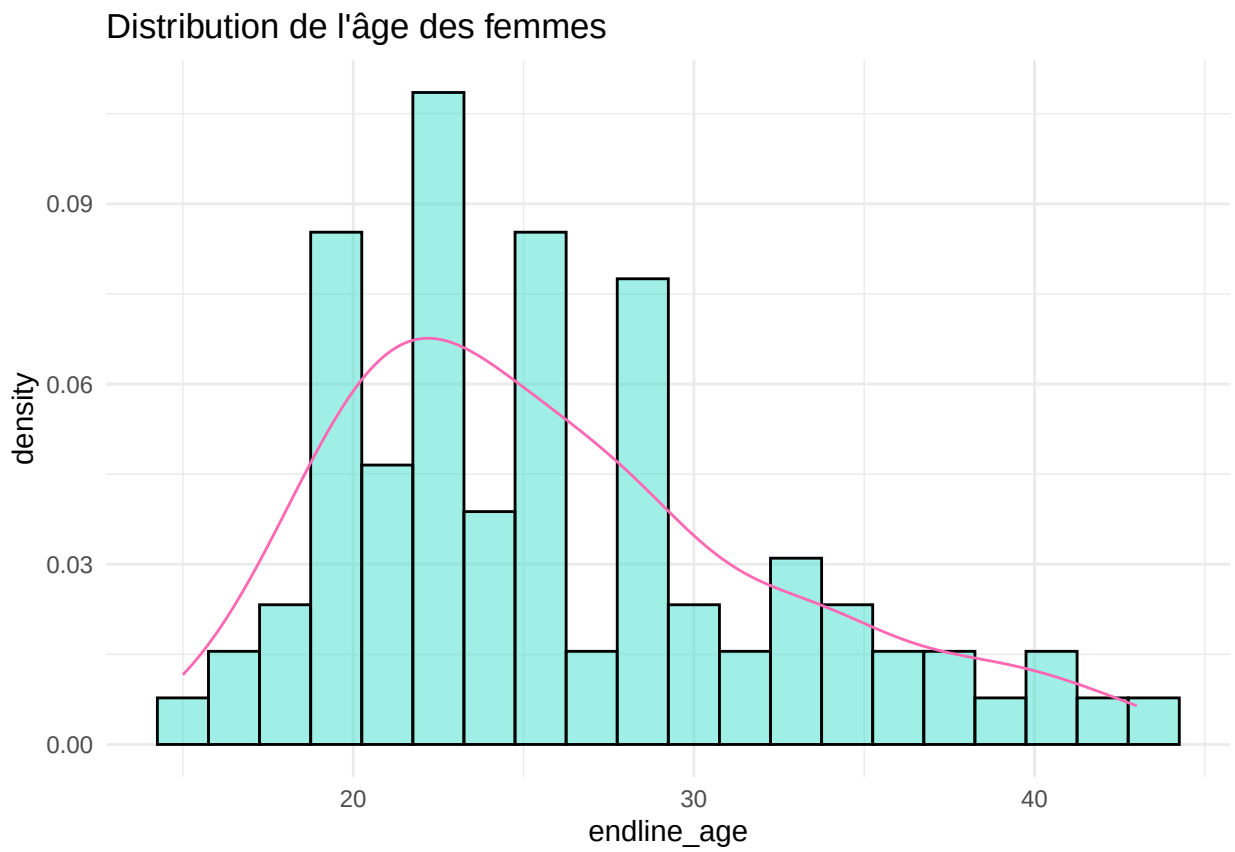
```r
## test de shapiro
print("Test de shapiro (hommes)")
```

```
## [1] "Test de shapiro (hommes)"
```

```r
shapiro.test (age_homme$endline_age)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  age_homme$endline_age
## W = 0.80078, p-value = 0.009631
```

```r
## test graphique : visualisation
ggplot(age_femme, aes(x = endline_age )) +
  geom_histogram(aes(y = ..density..),binwidth = 1.5, color = "black", fill = "turquoise", alpha = 0.5) +
  geom_density(color = "hotpink") +
  labs(title = "Distribution de l'âge des femmes") +
  theme_minimal()
```




```r
## test de shapiro
print("Test de shapiro (femmes)")
```

```
## [1] "Test de shapiro (femmes)"
```

```r
shapiro.test (age_femme$endline_age)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  age_femme$endline_age
## W = 0.94296, p-value = 0.0008633
```

Les graphiques de même que le resultat des tests montre que les distributions ne sont pas normales. Deux possibilités s'offrent alors à nous :

- Solution 1: Appliquer un test non paramétrique qui ne fait donc pas d'hypothèses sur les lois de distribution des variables testées, en l'occurrence le test des rangs de Wilcoxon, à l'aide de la fonction wilcox.test :

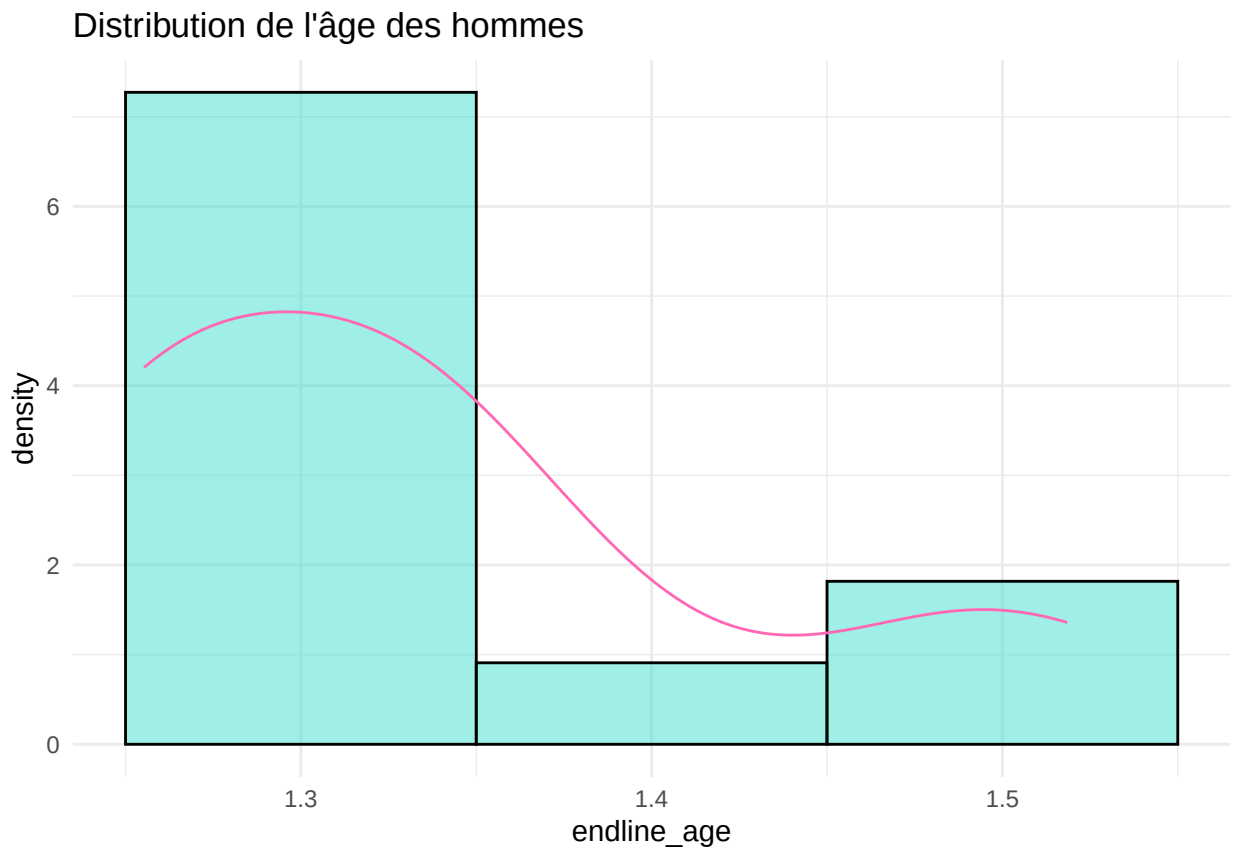
```r
## solution 1 : test non paramétrique (test de Wilcoxon)
wilcox.test(projet2$endline_age ~ projet2$endline_sex)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  projet2$endline_age by projet2$endline_sex
## W = 662.5, p-value = 0.03121
## alternative hypothesis: true location shift is not equal to 0
```

La p-value est inférieure à 5%, on peut considérer que les distributions des âges dans les deux sous-populations sont différentes.

- Solution 2 : Normaliser la distribution en appliquant la fonction logarithme et appliquer le test de student.

```r
## solution 2 : normaliser en appliquant le logarithme
### appliqation du logarithme
age_f_norm = log(age_femme, base = 10)
age_h_norm = log(age_homme, base = 10)

### testons pour voir si les distributions sont bien normalisées
## Test de normalité
## test graphique : visualisation
ggplot(age_h_norm, aes(x = endline_age )) +
  geom_histogram(aes(y = ..density..), binwidth = 0.1, color = "black", fill = "turquoise", alpha = 0.5)
  geom_density(color = "hotpink") +
  labs(title = "Distribution de l'âge des hommes") +
  theme_minimal()
```



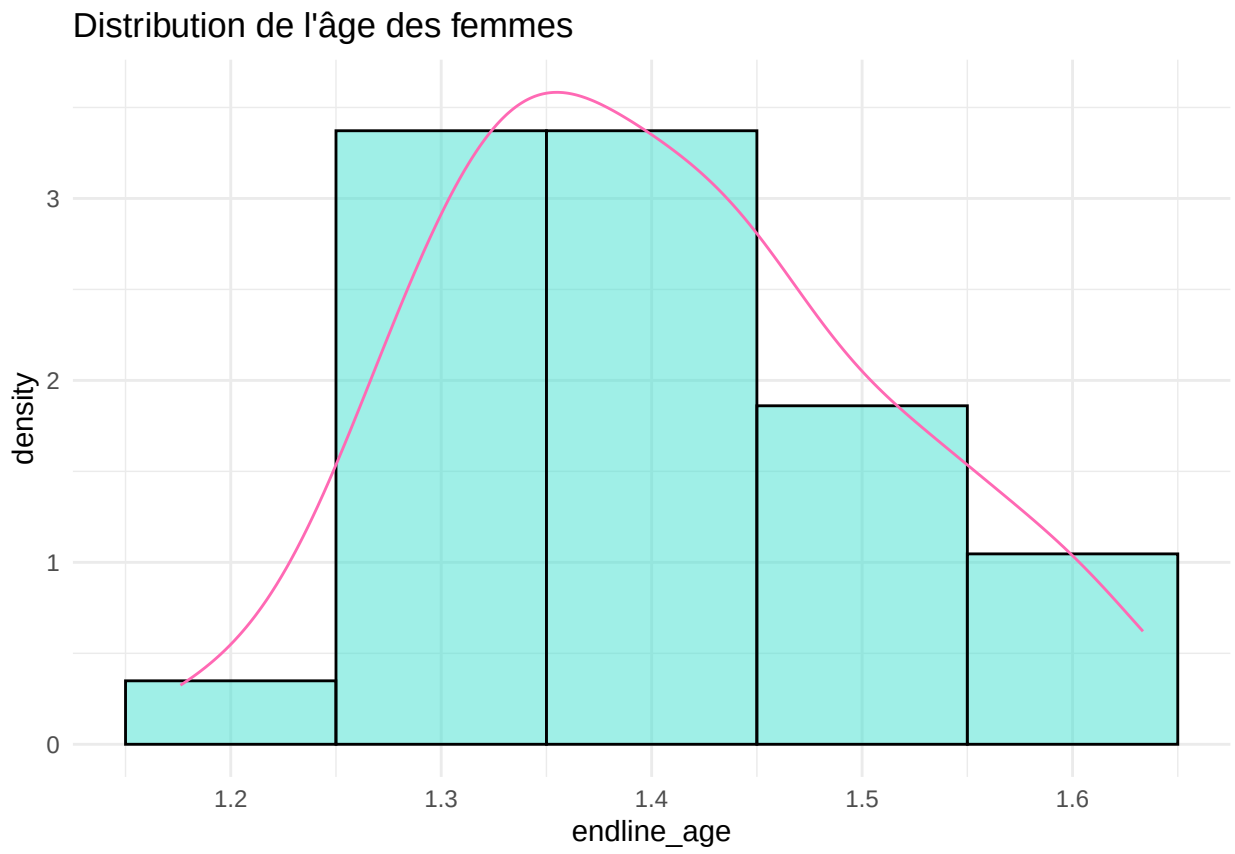

```
## test de shapiro
print("Test de shapiro (hommes)")
```

```
## [1] "Test de shapiro (hommes)"
```

```
shapiro.test (age_h_norm$endline_age)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  age_h_norm$endline_age
## W = 0.83666, p-value = 0.02856
```

```
## Test de normalité
## test graphique : visualisation
ggplot(age_f_norm, aes(x = endline_age )) +
  geom_histogram(aes(y = ..density..),binwidth = 0.1, color = "black", fill = "turquoise", alpha = 0.5)
  geom_density(color = "hotpink") +
  labs(title = "Distribution de l'âge des femmes") +
  theme_minimal()
```




```
## test de shapiro
print("Test de shapiro (femmes)")
```

```
## [1] "Test de shapiro (femmes)"
```

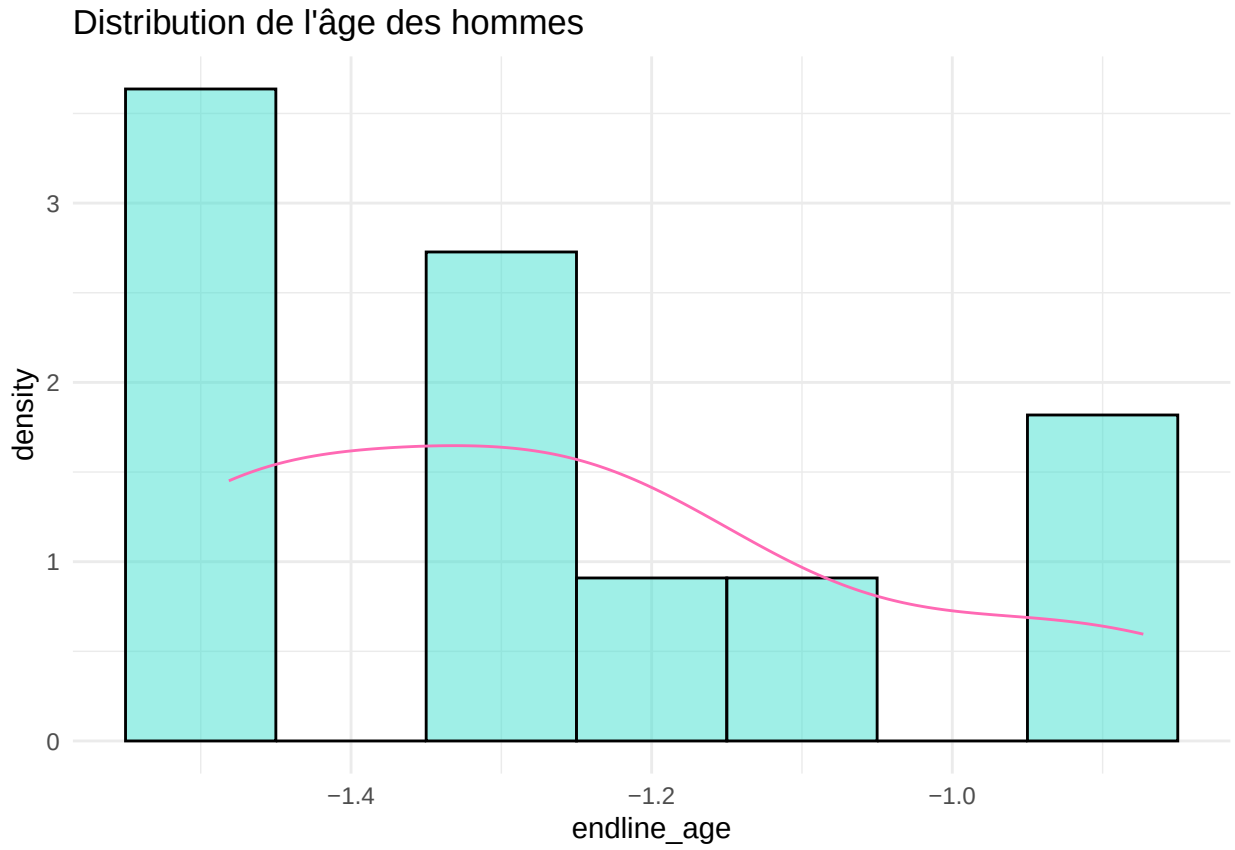
```
shapiro.test (age_f_norm$endline_age)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  age_f_norm$endline_age
## W = 0.9795, p-value = 0.1887
```

Les résultats nontrent que les nouvelles distributions obtenues ne sont pas toutes deux normales donc nous allons appliquer le logarithme une fois de plus.

```
### appliquer encore du logarithme
age_f_norm = log(log(age_f_norm))
age_h_norm = log(log(age_h_norm))

### testons pour voir si les distributions sont bien normalisées
## Test de normalité
## test graphique : visualisation
ggplot(age_h_norm, aes(x = endline_age )) +
```

```
  geom_histogram(aes(y = ..density..), binwidth = 0.1, color = "black", fill = "turquoise", alpha = 0.5)
  geom_density(color = "hotpink") +
  labs(title = "Distribution de l'âge des hommes") +
  theme_minimal()
```



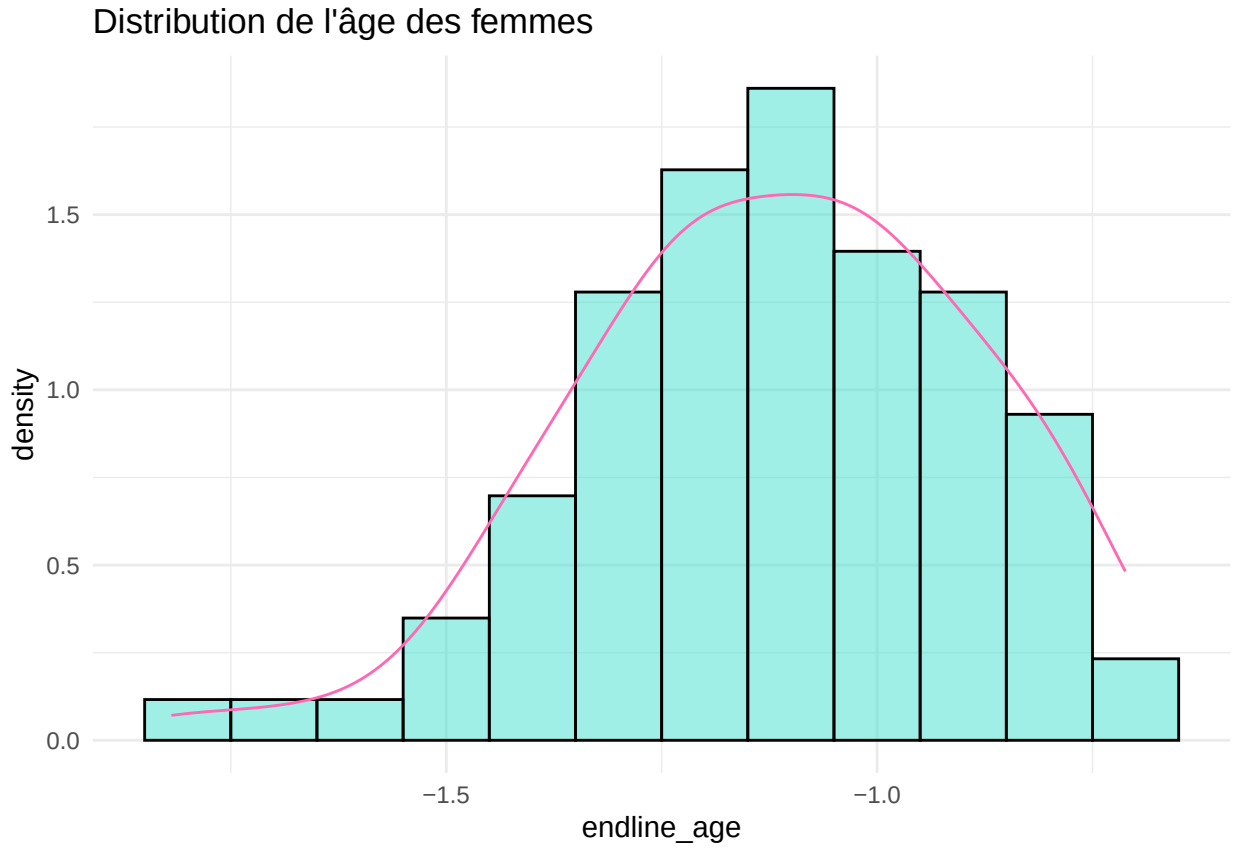

```
## test de shapiro
print("Test de shapiro (hommes)")
```

```
## [1] "Test de shapiro (hommes)"
```

```
shapiro.test (age_h_norm$endline_age)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  age_h_norm$endline_age
## W = 0.86825, p-value = 0.07361
```

```
## Test de normalité
## test graphique : visualisation
ggplot(age_f_norm, aes(x = endline_age )) +
  geom_histogram(aes(y = ..density..),binwidth = 0.1, color = "black", fill = "turquoise", alpha = 0.5)
  geom_density(color = "hotpink") +
  labs(title = "Distribution de l'âge des femmes") +
  theme_minimal()
```




```
## test de shapiro
print("Test de shapiro (femmes)")
```

```
## [1] "Test de shapiro (femmes)"
```

```
shapiro.test (age_f_norm$endline_age)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  age_f_norm$endline_age
## W = 0.97852, p-value = 0.1622
```

Distributions normales ! On peut donc appliquer le test de student

```
## test de student
t.test(age_h_norm$endline_age,age_f_norm$endline_age)
```

```
##
##  Welch Two Sample t-test
##
## data:  age_h_norm$endline_age and age_f_norm$endline_age
## t = -2.1365, df = 13.016, p-value = 0.0522
```

```
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.298631839  0.001639311
## sample estimates:
## mean of x mean of y
## -1.265509 -1.117013
```

La p-value étant supérieure à 5%, on admet que la différence d'âge entre les sexes est statistiquement significative au niveau de 5 %.

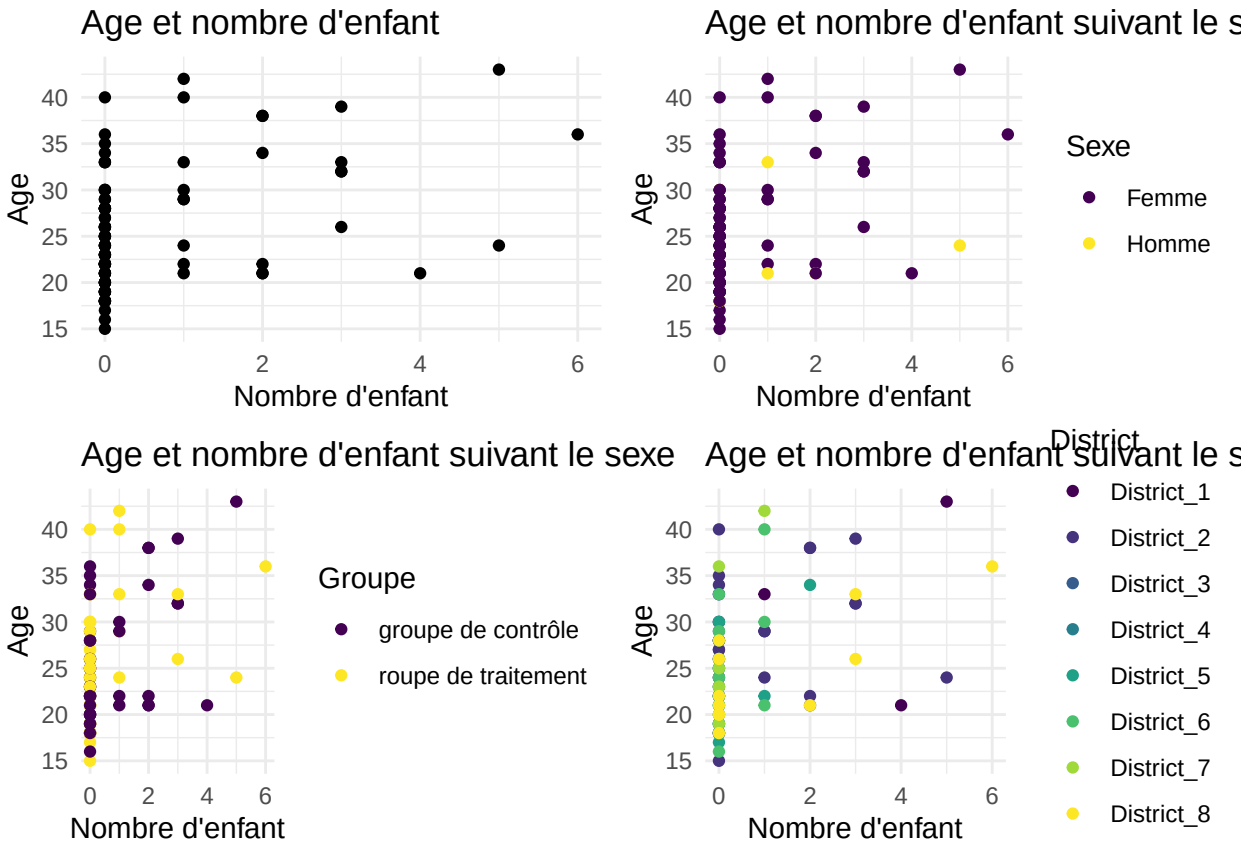
Nous avons là deux conclusions et résultats différents, est-ce normal ? Eh bien oui et cela s'rsplique par le fait que le test de Wilcoxon se base sur la différence des médianes alors que celui de student se base sur les moyennes. Par ailleurs, la médianne étant plus robuste par rapport aux valeurs abbérantes, nous retiendrons donc le test de Wilcoxon. D'où la différence d'âge entre les sexes est statistiquement significative au seuil de 5 %.

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

Ici, nous allons d'abord sortir le nuage de points pour l'ensemble de la population puis 3 autres nuages sur lesquels nous ferons apparaître une petite distinction suivant le sexe, le groupe (contrôle ou traitement) et le district.

```r
# Création du nuage de points de l'âge en fonction du nombre d'enfants
# premier nuage de points simple ou général
plot1 <- projet2 %>%
  ggplot(aes(x = endline_children_num, y = endline_age)) +
  geom_point() +
  labs(title = "Age et nombre d'enfant", x = "Nombre d'enfant", y = "Age") +
  theme_minimal()

# deuxième nuage de points : suivant le sexe
plot2 <- projet2 %>%
  ggplot(aes(x = endline_children_num, y = endline_age,
             color = factor(endline_sex, levels = c(0,1),labels= c("Femme","Homme")))) +
  geom_point() +
  labs(title = "Age et nombre d'enfant suivant le sexe",
       x = "Nombre d'enfant", y = "Age", color = "Sexe") +
  scale_color_viridis_d()+
  theme_minimal()

# troisième nuage de points : suivant le groupe
plot3 <- projet2 %>%
  ggplot(aes(x = endline_children_num, y = endline_age,
             color = factor(endline_g_traitement))) +
  geom_point() +
  labs(title = "Age et nombre d'enfant suivant le sexe",
       x = "Nombre d'enfant", y = "Age", color = "Groupe") +
  scale_color_viridis_d()+
  theme_minimal()

# quatrième nuage de points : suivant le district
plot4 <- projet2 %>%
  ggplot(aes(x = endline_children_num, y = endline_age,
```

```
            color = factor(endline_district, levels = c(1,2,3,4,5,6,7,8),
                           labels= paste("District", rep(1:8), sep="_")))) +
  geom_point() +
  labs(title = "Age et nombre d'enfant suivant le sexe",
       x = "Nombre d'enfant", y = "Age", color = "District") +
  scale_color_viridis_d()+
  theme_minimal()

# Afficher les graphiques côte à côte
library(gridExtra)
gridExtra::grid.arrange(plot1, plot2, plot3, plot4, ncol = 2)
```



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

Dans notre métier de statisticien nous somme souvent amenés dans le cadre de nos travaux à faire ce qu'on appele la regression statistique, une méthode qui permet de trouver une relation entre des données. Ainsi dans notre cas d'étude on pourrait s'interresser à comprendre ce qui pourrait et comment influencer l'intention de migrer à travers une regression sur une partie ou l'ensemble de nos donnée.

Nous avons plusieurs variantes de regression à savoir la régression linéaire et logistique qui une fois appliquées fournissnt en sortie des coefficients associés aux différentes variables représentant une estimation de l'effet moyen de ces dernières sur la variable à expliquer ou comprendre.

Par ailleurs dans le cas d'une regression logistique, il est possible de calculer des Odds ratio (exponentiel

des coeficient) qui losqu'ils sont significatifs permettent dans le cas de variables explicatives catégorielles de dire comment les différentes modalités influencent relativement (par rapport à une modalité de référence) la variable à expliquer. Dans notre exemple cela permettre de voir si par exemple un individu du groupe de contrôle a plus l'intention de migrer qu'un individu du groupe de traitement ou si c'est l'inverse.

Nous allons alors la régression la plus adaptée à nos donnée qui est une régression logistique ordinale (car la variable intention de migrer est une variable ordinale) et voir l'effet de l'appartenance au groupe de traitement sur l'intention de migrer.

Pour les autres variables explicatives (en dehors du groupe) à inclure dans le modèle en nous référants aux différentes études réalisées sur les déterminants de la migration[1], nous choisirons : l'âge, le sexe et le nombre d'enfant (charge familiale à supporter).

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

## selection des variables utiles au modèle : intention de migrer, âge, sexe, destination, nombre d'enf
base_rlo = projet2 %>% select(endline_intention,endline_district,
                              endline_sex, endline_children_num,
                              endline_g_traitement, endline_age)

  ## convertir les variables catégorielles en facteurs
base_rlo =  base_rlo %>% mutate(intention=factor(endline_intention, ordered = TRUE,
                                                 labels = c("1","2","3","4","5","6","7")),
                                district =factor(endline_district,levels = c(1,2,3,4,5,6,7,8), labels= p
                                sex=factor(endline_sex, levels = c(0,1),
                                           labels = c("Femme", "Homme")))


## Régression logistique ordinale
library(ordinal)
rl_ordinal = clm(intention~sex +endline_g_traitement + endline_age + endline_children_num, data=base_rl

## Tableau avec les coefficients estimés
print ("Tableau des coefficients estimés et Odd ratio")
```

```
## [1] "Tableau des coefficients estimés et Odd ratio"
```

```
t = data.frame(rl_ordinal$coefficients) # Coefficients
t = mutate(t, odd_ratio = exp(rl_ordinal.coefficients))
knitr::kable(t, format = "markdown")
```

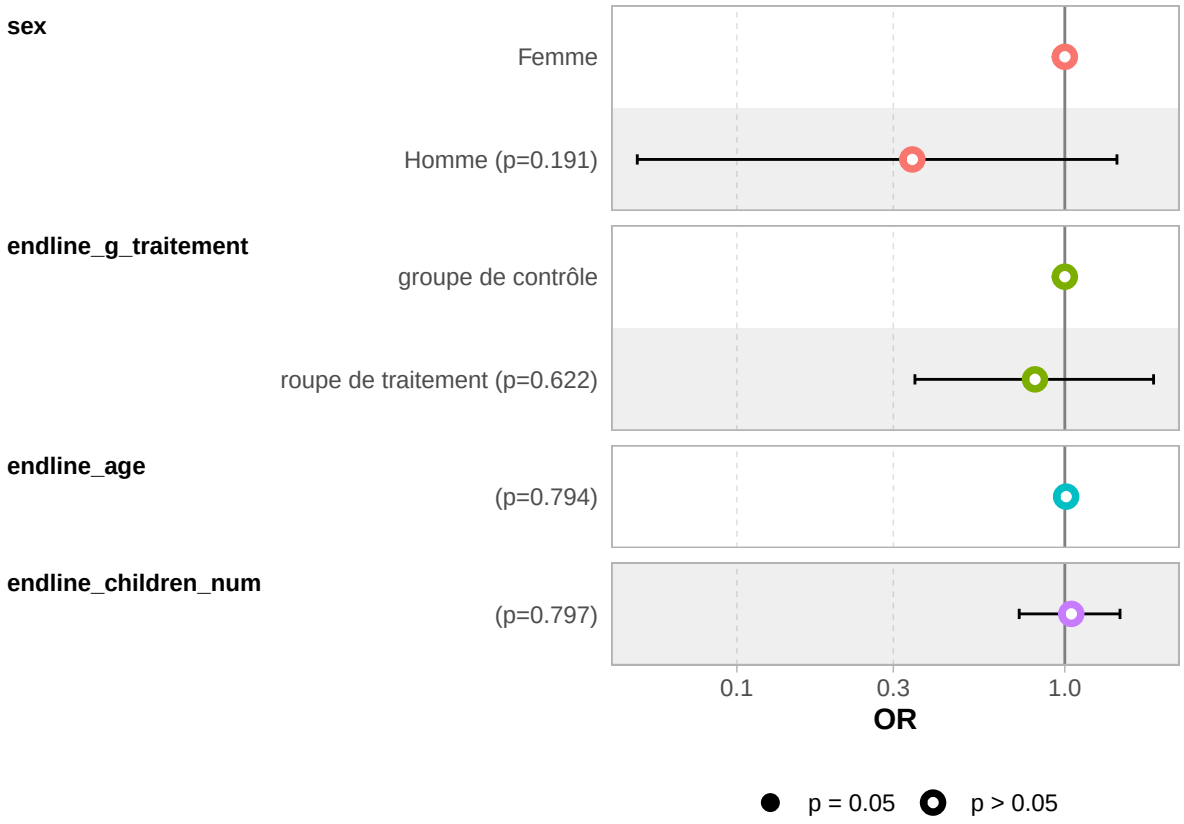
	rl_ordinal.coefficients	odd_ratio
1|2	0.6784328	1.9707868
2|3	0.8697437	2.3862993
3|4	1.4281685	4.1710531
4|5	1.9531012	7.0505189
5|6	2.6457055	14.0933849
6|7	3.9583514	52.3709168
sexHomme	-1.0707087	0.3427655
endline_g_traitementroupe de traitement	-0.2093486	0.8111124

[1]Document de synthèse: facteurs déterminants de la migration et processus de prise de décision d'Africains de l'Ouest et du Centre en déplacement - Processus de Rabat (rabat-process.org)

	rl_ordinal.coefficients	odd_ratio
endline_age	0.0093598	1.0094037
endline_children_num	0.0454031	1.0464496

Visualisation graphique des modèles : Odd ratio

```
library(GGally)
ggcoef_model(rl_ordinal, exponentiate = TRUE)
```



Il ressort de tout cela avec un seuil de significativité de 95 % que les personnes du groupe de contrôle sont 1.18 fois plus incités à migrer que ceux du groupe de traitement.

Créez un tableau de régression avec 3 modèles.

- 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.

```r
# Création d'un tableau de régression avec 3 modèles.
## modèle A : Modèle vide - Effet du traitement sur les intentions;
model.a = ordinal::clm(intention ~ endline_g_traitement,data = base_rlo)

## Modèle B : Effet du traitement sur les intentions en tenant compte de l'âge et du sexe;
model.b = ordinal::clm(intention ~ sex + endline_age, data = base_rlo)

## Modèle C : Identique au modèle B mais en contrôlant le district.
library(nnet)
model.c = nnet::multinom(intention ~ endline_sex +
                                        endline_age + district, data = base_rlo)
```

```
## # weights:  77 (60 variable)
## initial  value 188.753284
## iter  10 value 100.113054
## iter  20 value 92.905093
## iter  30 value 92.819597
## iter  40 value 92.803682
## iter  50 value 92.801422
## final  value 92.801392
## converged
```

```r
## sortir les résultats dans un seul tableau
### tabuler le modèle A
ta = model.a %>% gtsummary::tbl_regression()

### tabuler le modèle B
tb = model.b %>% gtsummary::tbl_regression()

### tabuler le modèle C
tc = model.c %>% gtsummary::tbl_regression()

### fusionner les 3 tableaux en un seul
gtsummary::tbl_stack(
  list(ta, tb, tc),
  group_header = c("Modèle A", "Modèle B", "Modèle C") ## intitulé des groupes de tableau associés
)
```

Group	Characteristic	log(OR)	95% CI	p-value
Modèle A	endline_g_traitement			
	groupe de contrôle	—	—	
	roupe de traitement	-0.28	-1.1, 0.53	0.5
Modèle B	sex			
	Femme	—	—	
	Homme	-1.1	-3.0, 0.33	0.2
	endline_age	0.01	-0.05, 0.08	0.7
Modèle C	endline_sex	2.2	-2.2, 6.6	0.3
	endline_age	-0.14	-0.44, 0.15	0.3
	district			
	District_1	—	—	
	District_2	-27	-27, -27	<0.001

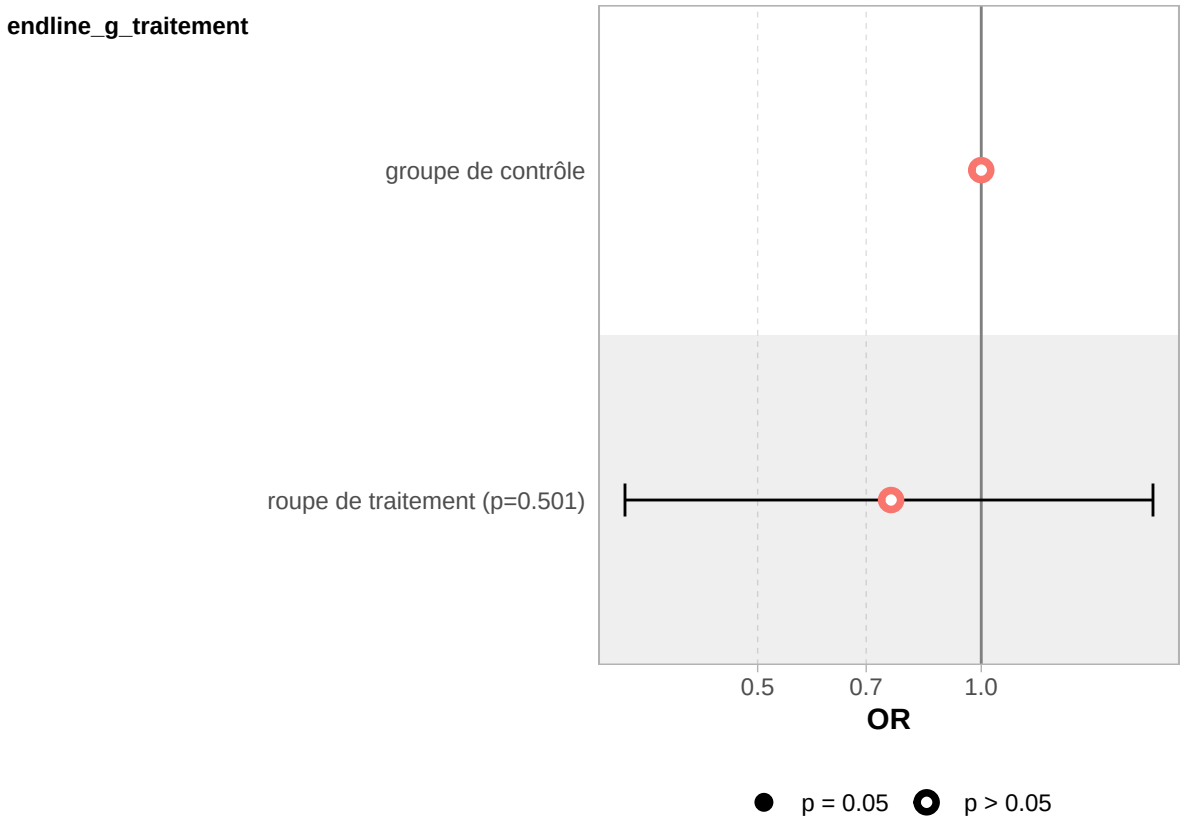
Group	Characteristic	log(OR)	95% CI	p-value
	District_3	-0.04	-3.3, 3.2	>0.9
	District_4	-20	-20, -20	<0.001
	District_5	-17	-17, -17	<0.001
	District_6	-26	-26, -26	<0.001
	District_7	-17	-17, -17	<0.001
	District_8	-0.69	-3.9, 2.5	0.7
	endline_sex	0.49	-2.0, 3.0	0.7
	endline_age	0.07	-0.04, 0.19	0.2
	district			
	District_1	—	—	
	District_2	16	15, 18	<0.001
	District_3	17	15, 19	<0.001
	District_4	-18	-18, -18	<0.001
	District_5	18	15, 20	<0.001
	District_6	16	15, 18	<0.001
	District_7	17	15, 19	<0.001
	District_8	17	16, 19	<0.001
	endline_sex	-17	-17, -17	<0.001
	endline_age	0.07	-0.06, 0.20	0.3
	district			
	District_1	—	—	
	District_2	17	15, 19	<0.001
	District_3	-12	-12, -12	<0.001
	District_4	-14	-14, -14	<0.001
	District_5	18	16, 20	<0.001
	District_6	16	14, 18	<0.001
	District_7	-12	-12, -12	<0.001
	District_8	18	16, 20	<0.001
	endline_sex	-18	-18, -18	<0.001
	endline_age	-0.01	-0.14, 0.13	>0.9
	district			
	District_1	—	—	
	District_2	-21	-21, -21	<0.001
	District_3	-19	-19, -19	<0.001
	District_4	-21	-21, -21	<0.001
	District_5	0.66	-2.7, 4.0	0.7
	District_6	-0.63	-3.4, 2.2	0.7
	District_7	0.28	-2.9, 3.4	0.9
	District_8	-0.25	-3.4, 2.9	0.9
	endline_sex	-19	-19, -19	<0.001
	endline_age	0.01	-0.17, 0.18	>0.9
	district			
	District_1	—	—	
	District_2	13	11, 16	<0.001
	District_3	15	13, 17	<0.001
	District_4	-14	-14, -14	<0.001
	District_5	16	13, 18	<0.001
	District_6	14	13, 16	<0.001
	District_7	-10	-10, -10	<0.001
	District_8	-12	-12, -12	<0.001
	endline_sex	-4.3	-4.3, -4.3	<0.001
	endline_age	-0.16	-0.59, 0.27	0.5

Group	Characteristic	log(OR)	95% CI	p-value
	district			
	District_1	—	—	
	District_2	-8.6	-8.6, -8.6	<0.001
	District_3	16	12, 20	<0.001
	District_4	-6.6	-6.6, -6.6	<0.001
	District_5	-4.7	-4.7, -4.7	<0.001
	District_6	-9.8	-9.8, -9.8	<0.001
	District_7	16	13, 20	<0.001
	District_8	-6.8	-6.8, -6.8	<0.001

Visualisation graphique des modèles

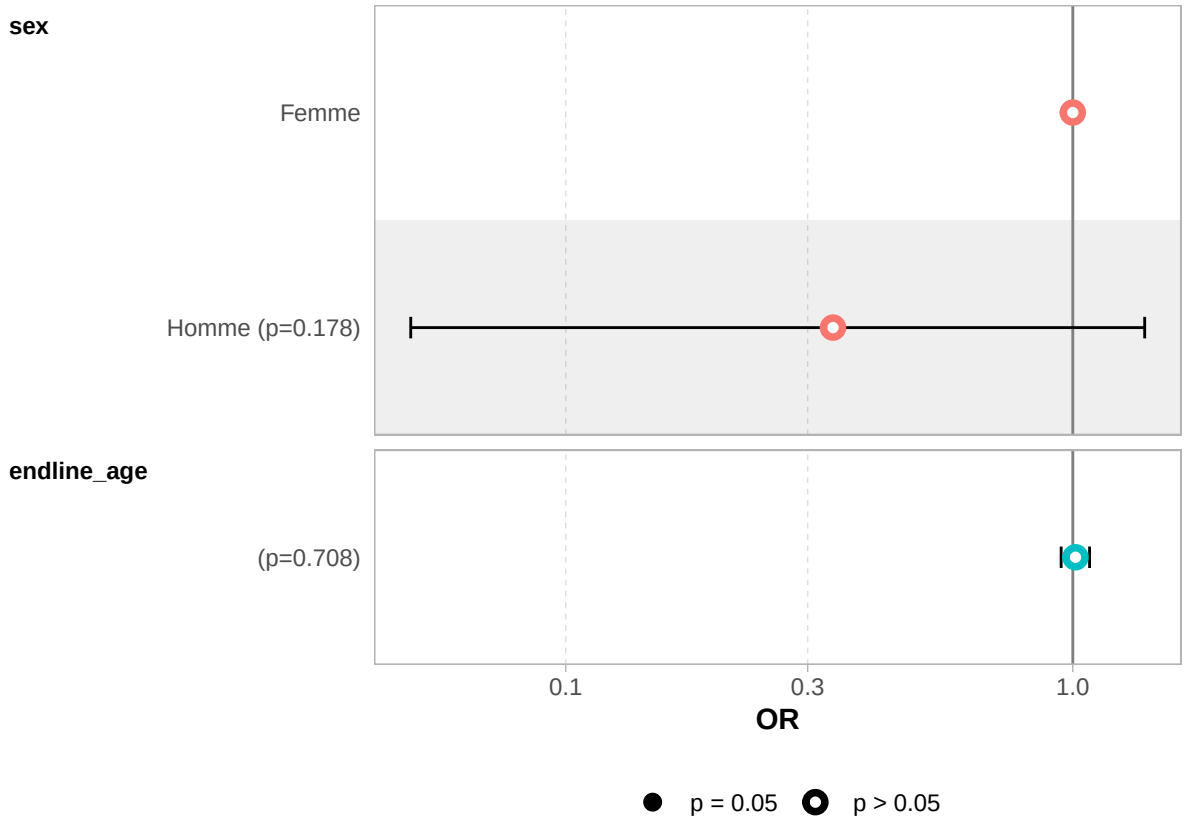
```
ggcoef_model(model.a, exponentiate = TRUE)
```

Modèle A



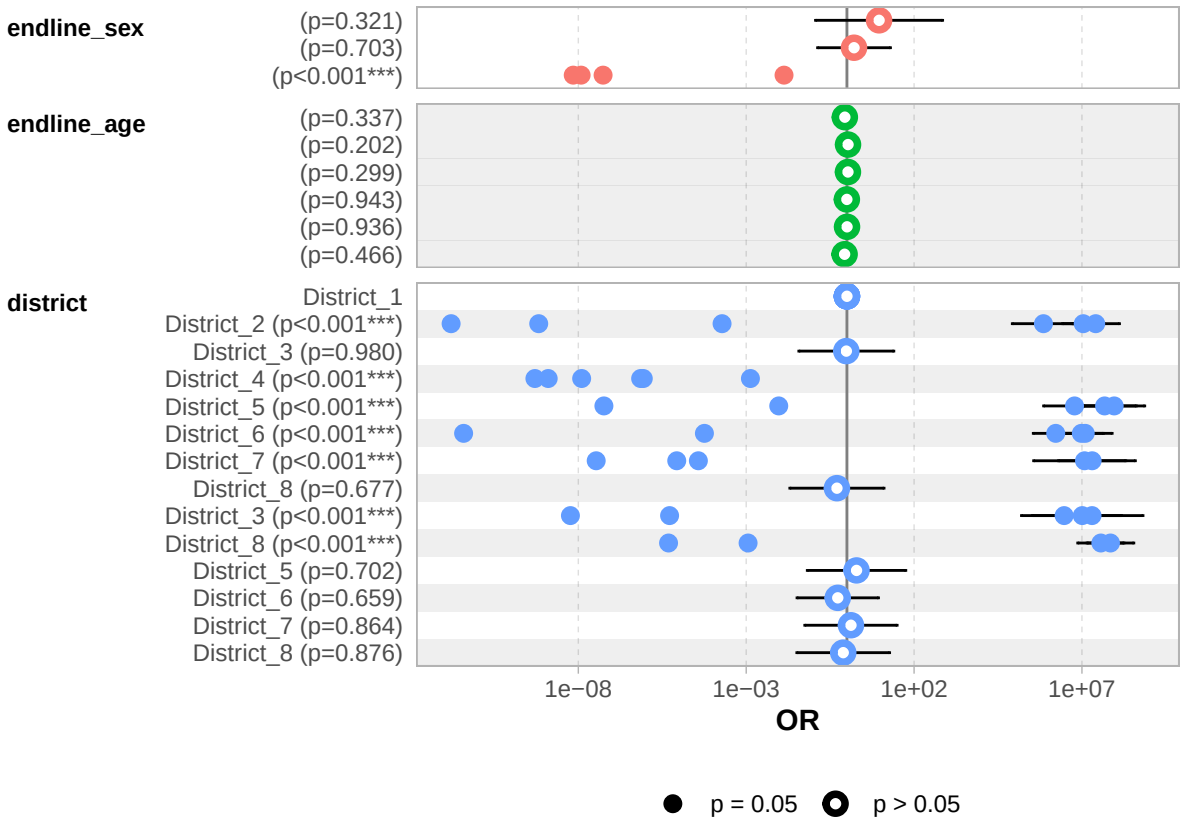

```
ggcoef_model(model.b, exponentiate = TRUE)
```

Modèle B




```
ggcoef_model(model.c, exponentiate = TRUE)
```

Modèle C


endline_sex
(p=0.321)
(p=0.703)
(p<0.001***)
endline_age
(p=0.337)
(p=0.202)
(p=0.299)
(p=0.943)
(p=0.936)
(p=0.466)
district
District_1
District_2 (p<0.001***)
District_3 (p=0.980)
District_4 (p<0.001***)
District_5 (p<0.001***)
District_6 (p<0.001***)
District_7 (p<0.001***)
District_8 (p=0.677)
District_3 (p<0.001***)
District_8 (p<0.001***)
District_5 (p=0.702)
District_6 (p=0.659)
District_7 (p=0.864)
District_8 (p=0.876)
1e−08
1e−03
1e+02
1e+07
OR
p = 0.05
p > 0.05