Partie 2

DIAKITE Moussa

#-------------------------------------------------------------  
# NETTOYAGE ET GESTION DES DONNEES  
#-------------------------------------------------------------  
  
library(readxl)  
library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.2 ✔ readr 2.1.4  
## ✔ forcats 1.0.0 ✔ stringr 1.5.0  
## ✔ ggplot2 3.4.2 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.2 ✔ tidyr 1.3.0  
## ✔ purrr 1.0.1   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

# chargement des packages nécessaires  
  
  
data <- read\_excel("Base\_Partie 2.xlsx", sheet = 1)  
# importer la base  
  
head(data)

## # 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>

# lecture des premières lignes  
  
data <- data %>%  
 rename(destination = country\_destination) %>%  
 mutate(destination = ifelse(destination < 0, NA, destination))  
# renom de country\_destination en destination  
# definition des valeurs negatives en valeur manquante  
  
  
# Créer une nouvelle variable contenant des tranches d’âge de 5 ans en utilisant la variable “age”  
data <- data %>%  
 mutate(age\_group = cut(age, breaks = seq(0, 100, 5), labels = FALSE, include.lowest = TRUE))  
# Création des tranches d'âge de 5ans pour la variable âge  
  
data <- data %>%  
 group\_by(id) %>%  
 mutate(nombre\_interviews = n())  
# création de variable contenant le nombre d'interview par agent  
  
  
# Avant l'application de sample(), on vérifie la taille de chaque groupe  
group\_sizes <- data %>%  
 group\_by(id) %>%  
 summarize(group\_size = n())  
  
print(group\_sizes)

## # A tibble: 97 × 2  
## id group\_size  
## <dbl> <int>  
## 1 1 1  
## 2 2 1  
## 3 3 1  
## 4 4 1  
## 5 5 1  
## 6 6 1  
## 7 7 1  
## 8 8 1  
## 9 9 1  
## 10 10 1  
## # ℹ 87 more rows

set.seed(42)   
# Réproduction des mêmes résultats aléatoires  
data <- data %>%  
 mutate(group\_assignment = sample(0:1, n(), replace = TRUE))  
# Créer une nouvelle variable qui affecte aléatoirement chaque répondant à un groupe de traitement (1) ou de controle (0).  
  
  
  
district\_population <- read\_excel("Base\_Partie 2.xlsx", sheet = 2)  
  
# Chargement des données de la feuille 2  
# La base contient la taille de la population de chaque district  
  
head(district\_population)

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

colnames(data)

## [1] "id" "starttime" "endtime"   
## [4] "enumerator" "district" "age"   
## [7] "sex" "children\_num" "intention"   
## [10] "destination" "age\_group" "nombre\_interviews"  
## [13] "group\_assignment"

colnames(district\_population)

## [1] "district" "population"

#data <- left\_join(data, district\_population, by = c("endline\_district" = "district"))  
data <- left\_join(data, district\_population, by = "district")  
# Fusion de la feuille 2 avec la feuille 1  
  
  
data <- data %>%  
 mutate(duree = endtime - starttime) %>%  
 group\_by(id) %>%  
 mutate(moy\_duree = mean(duree, na.rm = TRUE))  
# Calcule de la durée de l’entretien   
# Calcule de la durée moyenne de l’entretien par enquêteur  
  
  
names(data) <- lapply(names(data), function(x) paste("endline\_", x, sep = ""))  
# renomçage des variables par une boucle  
  
names(data)

## [1] "endline\_id" "endline\_starttime"   
## [3] "endline\_endtime" "endline\_enumerator"   
## [5] "endline\_district" "endline\_age"   
## [7] "endline\_sex" "endline\_children\_num"   
## [9] "endline\_intention" "endline\_destination"   
## [11] "endline\_age\_group" "endline\_nombre\_interviews"  
## [13] "endline\_group\_assignment" "endline\_population"   
## [15] "endline\_duree" "endline\_moy\_duree"

# affichage des données après renom.  
  
  
#---------------------------------------------------------------  
# ANALYSE ET VISUALISATION DES DONNEES  
#---------------------------------------------------------------  
  
  
library(dplyr)  
library(ggplot2)  
library(knitr)  
library(gt)  
library(stargazer)

##   
## Please cite as:   
##   
## Hlavac, Marek (2022). stargazer: Well-Formatted Regression and Summary Statistics Tables.  
## R package version 5.2.3. https://CRAN.R-project.org/package=stargazer

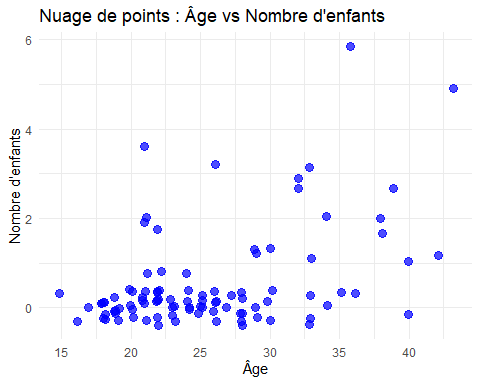
# chargement des packages nécessaires  
  
  
  
names(data)

## [1] "endline\_id" "endline\_starttime"   
## [3] "endline\_endtime" "endline\_enumerator"   
## [5] "endline\_district" "endline\_age"   
## [7] "endline\_sex" "endline\_children\_num"   
## [9] "endline\_intention" "endline\_destination"   
## [11] "endline\_age\_group" "endline\_nombre\_interviews"  
## [13] "endline\_group\_assignment" "endline\_population"   
## [15] "endline\_duree" "endline\_moy\_duree"

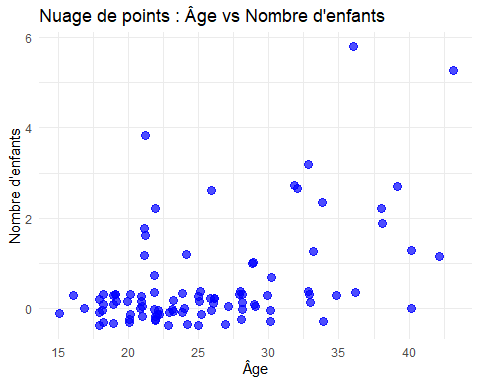
tableau\_recap <- data %>%  
 group\_by(endline\_district) %>%  
 summarise(age\_moyen = mean(endline\_age, na.rm = TRUE),  
 enfants\_moyen = mean(endline\_children\_num, na.rm = TRUE))  
# tableau recapitulatif de l'âge moyen   
# et du nombre d'enfant moyen par district  
  
  
  
# nous allons effectuer un test de student   
test\_age\_sex <- t.test(endline\_age ~ endline\_sex, data = data)  
  
# Obtenir la valeur p du test  
p\_value <- test\_age\_sex$p.value  
  
if (p\_value < 0.05) {  
 message("La différence d'âge entre les sexes est statistiquement significative.")  
} else {  
 message("Il n'y a pas de différence d'âge statistiquement significative entre les sexes.")  
}

## Il n'y a pas de différence d'âge statistiquement significative entre les sexes.

# Filtrer les données pour exclure les âges au-delà de 100 ans  
data\_filtered <- data %>%  
 filter(endline\_age <= 100)  
  
# Créer le nuage de points en utilisant les données filtrées  
ggplot(data\_filtered, aes(x = endline\_age, y = endline\_children\_num)) +  
 geom\_point(position = position\_jitter(width = 0.2), color = "blue", size = 3, alpha = 0.7) +  
 labs(x = "Âge", y = "Nombre d'enfants", title = "Nuage de points : Âge vs Nombre d'enfants") +  
 theme\_minimal()



graphique <- ggplot(data\_filtered, aes(x = endline\_age, y = endline\_children\_num)) +  
 geom\_point(position = position\_jitter(width = 0.2), color = "blue", size = 3, alpha = 0.7) +  
 labs(x = "Âge", y = "Nombre d'enfants", title = "Nuage de points : Âge vs Nombre d'enfants") +  
 theme\_minimal()  
  
graphique



#ggplot(data, aes(x = age, y = children\_num)) +  
#geom\_point(position = position\_jitter(width = 0.2), color = "blue", size = 3, alpha = 0.7) +  
#labs(x = "Âge", y = "Nombre d'enfants", title = "Nuage de points : Âge vs Nombre d'enfants") +  
#theme\_minimal()  
  
# Création du groupe de traitement et celui du controle  
  
# Calculer la moyenne de la variable "intention"  
mean\_intention <- mean(data$endline\_intention)  
print(mean\_intention)

## [1] 2.092784

# Créer les variables "groupe\_traitement" et "groupe\_controle"  
data$groupe\_traitement <- ifelse(data$endline\_intention > mean\_intention, 1, 0)  
data$groupe\_controle <- ifelse(data$endline\_intention <= mean\_intention, 1, 0)  
  
  
modele\_intention <- lm(endline\_intention ~ groupe\_traitement, data = data)  
summary(modele\_intention)

##   
## Call:  
## lm(formula = endline\_intention ~ groupe\_traitement, data = data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.4000 -0.0597 -0.0597 -0.0597 2.6000   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.05970 0.09122 11.62 <2e-16 \*\*\*  
## groupe\_traitement 3.34030 0.16402 20.36 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.7467 on 95 degrees of freedom  
## Multiple R-squared: 0.8136, Adjusted R-squared: 0.8117   
## F-statistic: 414.7 on 1 and 95 DF, p-value: < 2.2e-16

# Fonction pour extraire les coefficients et erreurs standard en gérant les cas où le modèle n'a pas de coefficient spécifique  
get\_coefficients <- function(model) {  
 coefficients <- tryCatch(coef(model), error = function(e) rep(NA, length(coef(model))))  
 std\_errors <- tryCatch(summary(model)$coefficients[, "Std. Error"], error = function(e) rep(NA, length(coef(model))))  
 return(list(coefficients, std\_errors))  
}  
  
modele\_A <- lm(endline\_intention ~ groupe\_traitement, data = data)  
modele\_B <- lm(endline\_intention ~ groupe\_traitement + endline\_age + endline\_sex, data = data)  
modele\_C <- lm(endline\_intention ~ groupe\_traitement + endline\_age + endline\_sex + endline\_district, data = data)  
  
# Extraire les coefficients et erreurs standard pour chaque modèle  
coefficients\_A <- get\_coefficients(modele\_A)  
coefficients\_B <- get\_coefficients(modele\_B)  
coefficients\_C <- get\_coefficients(modele\_C)  
  
  
  
  
# Créer le tableau de régression en utilisant data.frame avec des vecteurs de même longueur  
  
#tableau\_regression <- data.frame(  
 # Modèle = c("Modèle A", "Modèle B", "Modèle C"),  
 #Estimateur = unlist(get\_coefficients(modele\_A)[1], get\_coefficients(modele\_B)[1], get\_coefficients(modele\_C)[1]),  
 #Erreur\_standard = unlist(get\_coefficients(modele\_A)[2], get\_coefficients(modele\_B)[2], get\_coefficients(modele\_C)[2]),  
 #P\_value = c(summary(modele\_A)$coefficients[, "Pr(>|t|)"],  
 # summary(modele\_B)$coefficients[1, "Pr(>|t|)"],  
 #summary(modele\_C)$coefficients[1, "Pr(>|t|)"])  
#)  
  
#tableau\_regression