texte

Kerencia SEUNKAM

2025-02-25

## Installation et importation des packages

Dans cette section, nous installons tous les packages qui servirons dans la suite.

packages <- c("readr","cardx","haven","utils","dplyr","gtsummary","gt","labelled","survey")  
  
  
for (package in packages) {  
 if (!requireNamespace(package, quietly = TRUE)) { # Vérifie si le package n'est pas encore installé  
 install.packages(package)  
 }  
 library(package, character.only = TRUE) # nom du package en nom ou chaine de caractère ()  
}

##   
## 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

## Le chargement a nécessité le package : grid

## Le chargement a nécessité le package : Matrix

## Le chargement a nécessité le package : survival

##   
## Attachement du package : 'survey'

## L'objet suivant est masqué depuis 'package:graphics':  
##   
## dotchart

## chargement des bases

base\_men <- haven::read\_dta("Données/base\_menage.dta")%>%  
 labelled::to\_factor()  
  
 food\_comp\_child\_baseline <- haven::read\_dta("Données/food\_comp\_child\_baseline.dta")%>%  
 labelled::to\_factor()  
  
food\_comp\_child\_endline <- haven::read\_dta("Données/food\_comp\_child\_endline.dta")%>%  
 labelled::to\_factor()  
  
food\_comp\_mother\_baseline <- haven::read\_dta("Données/food\_comp\_mother\_baseline.dta")%>%  
 labelled::to\_factor()  
  
food\_comp\_mother\_endline <- haven::read\_dta("Données/food\_comp\_mother\_endline.dta")%>%  
labelled::to\_factor()

sum(duplicated(food\_comp\_child\_baseline))

## [1] 0

sum(duplicated(food\_comp\_mother\_baseline))

## [1] 0

sum(duplicated(food\_comp\_child\_endline))

## [1] 0

sum(duplicated(food\_comp\_mother\_endline))

## [1] 0

sum(duplicated(base\_men))

## [1] 1

base\_men <- base\_men %>% distinct()

ls()

## [1] "base\_men" "food\_comp\_child\_baseline"   
## [3] "food\_comp\_child\_endline" "food\_comp\_mother\_baseline"  
## [5] "food\_comp\_mother\_endline" "package"   
## [7] "packages"

child\_baseline\_var <- colnames(food\_comp\_child\_baseline)  
child\_endline\_var<-colnames(food\_comp\_child\_endline)  
mother\_baseline\_var<-colnames(food\_comp\_mother\_baseline)  
mother\_endline\_var<-colnames(food\_comp\_mother\_endline)  
  
setdiff(child\_baseline\_var, child\_endline\_var)

## character(0)

setdiff(mother\_baseline\_var, mother\_endline\_var)

## [1] "V1" "V9"

colnames(food\_comp\_mother\_endline)

## [1] "regionid" "communeid" "villageid" "hhid" "round"   
## [6] "s1\_q0" "s1\_q1" "s1\_q2" "energ\_kcal" "protein\_g"   
## [11] "lipid\_tot\_g" "calcium\_mg" "iron\_mg" "zinc\_mg" "vit\_b6\_mg"   
## [16] "vit\_b12\_mcg" "vit\_c\_mg"

colnames(food\_comp\_mother\_baseline)

## [1] "regionid" "communeid" "villageid" "hhid" "round"   
## [6] "s1\_q0" "s1\_q1" "s1\_q2" "V1" "protein\_g"   
## [11] "lipid\_tot\_g" "calcium\_mg" "iron\_mg" "V9" "vit\_b6\_mg"   
## [16] "vit\_b12\_mcg" "vit\_c\_mg"

food\_comp\_mother\_baseline <- food\_comp\_mother\_baseline %>%  
 rename(zinc\_mg = V9, energ\_kcal = V1)

summary(food\_comp\_mother\_baseline$round)

## Baseline Endline   
## 4256 0

summary(food\_comp\_mother\_baseline$s1\_q0)

## Breakfast Lunch Dinner Snacks   
## 1064 1064 1064 1064

summary(food\_comp\_mother\_baseline$s1\_q1)

## No Yes   
## 673 3583

summary(food\_comp\_mother\_baseline$s1\_q2)

## mother child mother and child   
## 4256 0 0

#La consommation moyenne d’énergie par repas pour les mères lors de l’enquête de base est 714.72 kcal. Cela a été calculé à partir de la division de energ\_kcal par nb\_repas pour chaque observation comme ci dessous:

mother\_baseline <- food\_comp\_mother\_baseline %>%  
 mutate(nb\_repas = (s1\_q0 == "Yes") + (s1\_q1 == "Yes") + (s1\_q2 == "Yes"))

mother\_baseline <- mother\_baseline %>%  
 mutate(energ\_par\_repas = energ\_kcal / nb\_repas)  
  
mean\_energie\_par\_repas <- mean(mother\_baseline$energ\_par\_repas, na.rm = TRUE)  
print(mean\_energie\_par\_repas)

## [1] 714.7165

# Sauvegarde des chandement des bases

# Sauvegarde de food\_comp\_mother\_baseline corrigée  
haven::write\_dta(food\_comp\_mother\_baseline, "mother\_baseline\_v1.dta")  
  
# Sauvegarde de food\_comp\_mother\_endline corrigée  
haven::write\_dta(food\_comp\_mother\_endline, "mother\_endline\_v1.dta")  
  
# Sauvegarde de food\_comp\_child\_baseline corrigée  
haven::write\_dta(food\_comp\_child\_baseline, "child\_baseline\_v1.dta")  
  
# Sauvegarde de food\_comp\_child\_endline corrigée  
haven::write\_dta(food\_comp\_child\_endline, "child\_endline\_v1.dta")  
  
# Sauvegarde de base\_menage corrigée  
haven::write\_dta(base\_men, "base\_menage\_final.dta")

## partie 2

###Baseline

child\_baseline\_V1 <- haven::read\_dta("child\_baseline\_v1.dta")%>%  
 labelled::to\_factor()  
child\_endline\_V1 <- haven::read\_dta("child\_endline\_v1.dta")%>%  
 labelled::to\_factor()  
mother\_baseline\_V1 <- haven::read\_dta("mother\_baseline\_v1.dta")%>%  
 labelled::to\_factor()  
mother\_endline\_V1 <- haven::read\_dta("mother\_endline\_v1.dta")%>%  
 labelled::to\_factor()

# Empilement des données mères et enfants (Baseline)  
baseline\_combined <- bind\_rows(  
 mother\_baseline\_V1 %>% mutate(type = "mother"),   
 child\_baseline\_V1 %>% mutate(type = "child")  
)

# Vérifier les noms des colonnes dans baseline\_combined  
colnames(baseline\_combined)

## [1] "regionid" "communeid" "villageid" "hhid" "round"   
## [6] "s1\_q0" "s1\_q1" "s1\_q2" "energ\_kcal" "protein\_g"   
## [11] "lipid\_tot\_g" "calcium\_mg" "iron\_mg" "zinc\_mg" "vit\_b6\_mg"   
## [16] "vit\_b12\_mcg" "vit\_c\_mg" "type"

# Liste des variables de consommation  
consumption\_vars <- c("energ\_kcal", "protein\_g", "lipid\_tot\_g",   
 "calcium\_mg", "iron\_mg", "zinc\_mg",   
 "vit\_b6\_mg", "vit\_b12\_mcg", "vit\_c\_mg")  
  
# Renommer les variables de consommation en ajoutant le suffixe "\_b"  
baseline\_combined <- baseline\_combined %>%  
 rename\_with(~paste0(.,"\_b"), all\_of(consumption\_vars))

# Calcul de la somme des repas pour chaque individu  
baseline\_combined <- baseline\_combined %>%  
 mutate(total\_energ\_kcal\_b = rowSums(select(., energ\_kcal\_b), na.rm = TRUE),  
 total\_protein\_g\_b = rowSums(select(., protein\_g\_b), na.rm = TRUE),  
 total\_lipid\_tot\_g\_b = rowSums(select(., lipid\_tot\_g\_b), na.rm = TRUE),  
 total\_calcium\_mg\_b = rowSums(select(., calcium\_mg\_b), na.rm = TRUE),  
 total\_iron\_mg\_b = rowSums(select(., iron\_mg\_b), na.rm = TRUE),  
 total\_zinc\_mg\_b = rowSums(select(., zinc\_mg\_b), na.rm = TRUE),  
 total\_vit\_b6\_mg\_b = rowSums(select(., vit\_b6\_mg\_b), na.rm = TRUE),  
 total\_vit\_b12\_mcg\_b = rowSums(select(., vit\_b12\_mcg\_b), na.rm = TRUE),  
 total\_vit\_c\_mg\_b = rowSums(select(., vit\_c\_mg\_b), na.rm = TRUE))

# Sauvegarder la base de données finale  
baseline\_final <- baseline\_combined %>%  
 select(hhid, s1\_q2, energ\_kcal\_b, protein\_g\_b, lipid\_tot\_g\_b,   
 calcium\_mg\_b, iron\_mg\_b, zinc\_mg\_b, vit\_b6\_mg\_b,   
 vit\_b12\_mcg\_b, vit\_c\_mg\_b)  
  
# Sauvegarder en fichier .dta  
haven::write\_dta(baseline\_final, "baseline\_final.dta")

###Endline

# Chargement des bases de données Endline  
mother\_endline\_v1 <- haven::read\_dta("mother\_endline\_v1.dta") %>% labelled::to\_factor()  
child\_endline\_v1 <- haven::read\_dta("child\_endline\_v1.dta") %>% labelled::to\_factor()  
  
# Empilement des données mères et enfants  
endline\_combined <- bind\_rows(  
 mother\_endline\_v1 %>% mutate(type = "mother"),   
 child\_endline\_v1 %>% mutate(type = "child")  
)  
  
# Liste des variables de consommation  
consumption\_vars <- c("energ\_kcal", "protein\_g", "lipid\_tot\_g",   
 "calcium\_mg", "iron\_mg", "zinc\_mg",   
 "vit\_b6\_mg", "vit\_b12\_mcg", "vit\_c\_mg")  
  
# Renommer les variables de consommation en ajoutant le suffixe "\_e"  
endline\_combined <- endline\_combined %>%  
 rename\_with(~paste0(.,"\_e"), all\_of(consumption\_vars))

# Vérifier les noms des colonnes dans le jeu de données  
colnames(endline\_combined)

## [1] "regionid" "communeid" "villageid" "hhid"   
## [5] "round" "s1\_q0" "s1\_q1" "s1\_q2"   
## [9] "energ\_kcal\_e" "protein\_g\_e" "lipid\_tot\_g\_e" "calcium\_mg\_e"   
## [13] "iron\_mg\_e" "zinc\_mg\_e" "vit\_b6\_mg\_e" "vit\_b12\_mcg\_e"  
## [17] "vit\_c\_mg\_e" "type"

# Calculer la somme des 4 repas pour chaque individu  
endline\_summary <- endline\_combined %>%  
 group\_by(hhid, type) %>% # Regrouper par hhid et type (mère ou enfant)  
 summarize(  
 total\_energ\_kcal = sum(energ\_kcal\_e, na.rm = TRUE),  
 total\_protein\_g = sum(protein\_g\_e, na.rm = TRUE),  
 total\_lipid\_tot\_g = sum(lipid\_tot\_g\_e, na.rm = TRUE),  
 total\_calcium\_mg = sum(calcium\_mg\_e, na.rm = TRUE),  
 total\_iron\_mg = sum(iron\_mg\_e, na.rm = TRUE),  
 total\_zinc\_mg = sum(zinc\_mg\_e, na.rm = TRUE),  
 total\_vit\_b6\_mg = sum(vit\_b6\_mg\_e, na.rm = TRUE),  
 total\_vit\_b12\_mcg = sum(vit\_b12\_mcg\_e, na.rm = TRUE),  
 total\_vit\_c\_mg = sum(vit\_c\_mg\_e, na.rm = TRUE)  
 )

## `summarise()` has grouped output by 'hhid'. You can override using the  
## `.groups` argument.

# Vérification de la nouvelle base résumée  
head(endline\_summary)

## # A tibble: 6 × 11  
## # Groups: hhid [3]  
## hhid type total\_energ\_kcal total\_protein\_g total\_lipid\_tot\_g  
## <chr> <chr> <dbl> <dbl> <dbl>  
## 1 4948484848535052 child 3960. 116. 149.   
## 2 4948484848535052 mother 6992. 195. 284.   
## 3 4948484848535053 child 364. 9.81 1.13  
## 4 4948484848535053 mother 1292. 35.0 4.09  
## 5 4948484848535055 child 3149. 107. 97.8   
## 6 4948484848535055 mother 3112. 73.9 124.   
## # ℹ 6 more variables: total\_calcium\_mg <dbl>, total\_iron\_mg <dbl>,  
## # total\_zinc\_mg <dbl>, total\_vit\_b6\_mg <dbl>, total\_vit\_b12\_mcg <dbl>,  
## # total\_vit\_c\_mg <dbl>

# Sauvegarder la base de données finale sous "endline\_final.dta"  
haven::write\_dta(endline\_summary, "endline\_final.dta")

colnames(endline\_combined)

## [1] "regionid" "communeid" "villageid" "hhid"   
## [5] "round" "s1\_q0" "s1\_q1" "s1\_q2"   
## [9] "energ\_kcal\_e" "protein\_g\_e" "lipid\_tot\_g\_e" "calcium\_mg\_e"   
## [13] "iron\_mg\_e" "zinc\_mg\_e" "vit\_b6\_mg\_e" "vit\_b12\_mcg\_e"  
## [17] "vit\_c\_mg\_e" "type"

# Sélectionner les colonnes nécessaires  
endline\_final <- endline\_combined %>%  
 select(hhid, s1\_q2, energ\_kcal\_e, protein\_g\_e, lipid\_tot\_g\_e, calcium\_mg\_e, iron\_mg\_e, zinc\_mg\_e)  
  
# Vérification du contenu de la base avant de sauvegarder  
head(endline\_final)

## # A tibble: 6 × 8  
## hhid s1\_q2 energ\_kcal\_e protein\_g\_e lipid\_tot\_g\_e calcium\_mg\_e iron\_mg\_e  
## <chr> <fct> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 494848484… moth… 2217. 48.7 89.6 819. 22.7   
## 2 494848484… moth… 1021. 34.8 5.94 72.1 16.6   
## 3 494848484… moth… 3038. 92.0 184. 1781. 11.3   
## 4 494848484… moth… 716. 19.1 4.73 132. 8.88   
## 5 494848484… moth… 618. 16.5 1.63 10.8 0.105  
## 6 494848484… moth… NA NA NA NA NA   
## # ℹ 1 more variable: zinc\_mg\_e <dbl>

# Sauvegarder la base de données finale sous "endline\_final.dta"  
haven::write\_dta(endline\_final, "endline\_final.dta")

# Charger les données  
baseline\_final <- haven::read\_dta("baseline\_final.dta")  
endline\_final <- haven::read\_dta("endline\_final.dta")  
  
# Fusionner les deux bases de données  
combined\_data <- merge(baseline\_final, endline\_final, by = "hhid", suffixes = c("\_b", "\_e"))  
  
# Vérifier la fusion  
head(combined\_data)

## hhid s1\_q2\_b energ\_kcal\_b protein\_g\_b lipid\_tot\_g\_b calcium\_mg\_b  
## 1 4948484848535052 1 680.4425 23.31366 5.301461 62.73322  
## 2 4948484848535052 1 680.4425 23.31366 5.301461 62.73322  
## 3 4948484848535052 1 680.4425 23.31366 5.301461 62.73322  
## 4 4948484848535052 1 680.4425 23.31366 5.301461 62.73322  
## 5 4948484848535052 1 680.4425 23.31366 5.301461 62.73322  
## 6 4948484848535052 1 680.4425 23.31366 5.301461 62.73322  
## iron\_mg\_b zinc\_mg\_b vit\_b6\_mg\_b vit\_b12\_mcg\_b vit\_c\_mg\_b s1\_q2\_e energ\_kcal\_e  
## 1 10.59063 4.507263 0.3057863 0.008691675 0.04414822 1 2217.1804  
## 2 10.59063 4.507263 0.3057863 0.008691675 0.04414822 1 1021.0393  
## 3 10.59063 4.507263 0.3057863 0.008691675 0.04414822 1 3037.6642  
## 4 10.59063 4.507263 0.3057863 0.008691675 0.04414822 1 716.1742  
## 5 10.59063 4.507263 0.3057863 0.008691675 0.04414822 2 534.8655  
## 6 10.59063 4.507263 0.3057863 0.008691675 0.04414822 2 1700.6784  
## protein\_g\_e lipid\_tot\_g\_e calcium\_mg\_e iron\_mg\_e zinc\_mg\_e  
## 1 48.68229 89.597996 818.78014 22.659099 9.569128  
## 2 34.78248 5.944220 72.12158 16.558456 6.461785  
## 3 92.04654 183.596783 1781.30971 11.344701 2.227778  
## 4 19.11306 4.728846 131.59349 8.876951 3.586713  
## 5 18.37790 3.205405 41.32309 8.709022 3.356234  
## 6 58.83699 90.268128 1122.62760 7.227706 1.418428

# Charger la base de données des caractéristiques du ménage  
base\_menage\_final <- haven::read\_dta("base\_menage\_final.dta")  
  
# Fusionner les informations des caractéristiques du ménage  
final\_data <- merge(combined\_data, base\_menage\_final, by = "hhid")  
  
# Vérifier la fusion  
head(final\_data)

## hhid s1\_q2\_b energ\_kcal\_b protein\_g\_b lipid\_tot\_g\_b calcium\_mg\_b  
## 1 4948484848535052 1 680.4425 23.31366 5.301461 62.73322  
## 2 4948484848535052 1 680.4425 23.31366 5.301461 62.73322  
## 3 4948484848535052 1 680.4425 23.31366 5.301461 62.73322  
## 4 4948484848535052 1 680.4425 23.31366 5.301461 62.73322  
## 5 4948484848535052 1 680.4425 23.31366 5.301461 62.73322  
## 6 4948484848535052 1 680.4425 23.31366 5.301461 62.73322  
## iron\_mg\_b zinc\_mg\_b vit\_b6\_mg\_b vit\_b12\_mcg\_b vit\_c\_mg\_b s1\_q2\_e energ\_kcal\_e  
## 1 10.59063 4.507263 0.3057863 0.008691675 0.04414822 1 2217.1804  
## 2 10.59063 4.507263 0.3057863 0.008691675 0.04414822 1 1021.0393  
## 3 10.59063 4.507263 0.3057863 0.008691675 0.04414822 1 3037.6642  
## 4 10.59063 4.507263 0.3057863 0.008691675 0.04414822 1 716.1742  
## 5 10.59063 4.507263 0.3057863 0.008691675 0.04414822 2 534.8655  
## 6 10.59063 4.507263 0.3057863 0.008691675 0.04414822 2 1700.6784  
## protein\_g\_e lipid\_tot\_g\_e calcium\_mg\_e iron\_mg\_e zinc\_mg\_e regionid communeid  
## 1 48.68229 89.597996 818.78014 22.659099 9.569128 2 25  
## 2 34.78248 5.944220 72.12158 16.558456 6.461785 2 25  
## 3 92.04654 183.596783 1781.30971 11.344701 2.227778 2 25  
## 4 19.11306 4.728846 131.59349 8.876951 3.586713 2 25  
## 5 18.37790 3.205405 41.32309 8.709022 3.356234 2 25  
## 6 58.83699 90.268128 1122.62760 7.227706 1.418428 2 25  
## villageid hhsize poly hh\_primary s1\_q2 s1\_q4a s2\_q1 s2\_q2 s2\_q4 s29\_q1  
## 1 1000 4 1 1 1 37 1 1 1 1  
## 2 1000 4 1 1 1 37 1 1 1 1  
## 3 1000 4 1 1 1 37 1 1 1 1  
## 4 1000 4 1 1 1 37 1 1 1 1  
## 5 1000 4 1 1 1 37 1 1 1 1  
## 6 1000 4 1 1 1 37 1 1 1 1  
## demgrp1 demgrp2 demgrp3 demgrp4 demgrp5 dependencyratio hfias\_score T1  
## 1 1 1 0 1 0 2 18 1  
## 2 1 1 0 1 0 2 18 1  
## 3 1 1 0 1 0 2 18 1  
## 4 1 1 0 1 0 2 18 1  
## 5 1 1 0 1 0 2 18 1  
## 6 1 1 0 1 0 2 18 1

final\_data<- haven::read\_dta("base\_finale.dta")%>%  
 labelled::to\_factor()  
colnames(final\_data)

## [1] "hhid" "s1\_q2\_b" "energ\_kcal\_b" "protein\_g\_b"   
## [5] "lipid\_tot\_g\_b" "calcium\_mg\_b" "iron\_mg\_b" "zinc\_mg\_b"   
## [9] "vit\_b6\_mg\_b" "vit\_b12\_mcg\_b" "vit\_c\_mg\_b" "s1\_q2\_e"   
## [13] "energ\_kcal\_e" "protein\_g\_e" "lipid\_tot\_g\_e" "calcium\_mg\_e"   
## [17] "iron\_mg\_e" "zinc\_mg\_e" "regionid" "communeid"   
## [21] "villageid" "hhsize" "poly" "hh\_primary"   
## [25] "s1\_q2" "s1\_q4a" "s2\_q1" "s2\_q2"   
## [29] "s2\_q4" "s29\_q1" "demgrp1" "demgrp2"   
## [33] "demgrp3" "demgrp4" "demgrp5" "dependencyratio"  
## [37] "hfias\_score" "T1"

# Sauvegarder la base de données finale sous "base\_finale.dta"  
haven::write\_dta(final\_data, "base\_finale.dta")