TEST R EVALUATION

FALL Ndeye Ramatoulaye Ndoye

2025-02-25

# Installation et chargement des packages nécessaires  
packages <- c("haven", "utils", "dplyr", "tidyverse", "gtsummary", "survey", "knitr", "kableExtra", "labelled")  
  
for (pkg in packages) {  
 if (!require(pkg, character.only = TRUE)) install.packages(pkg, dependencies = TRUE)  
 library(pkg, character.only = TRUE)  
}

library(haven)  
  
# Chargement des bases  
base\_menage <- read\_dta("data/base\_menage.dta")  
mother\_baseline <- read\_dta("data/food\_comp\_mother\_baseline.dta")  
mother\_endline <- read\_dta("data/food\_comp\_mother\_endline.dta")  
child\_baseline <- read\_dta("data/food\_comp\_child\_baseline.dta")  
child\_endline <- read\_dta("data/food\_comp\_child\_endline.dta")

#PARTIE 1: GESTION ET NETTOYAGE DES BASES DE DONNEES

# Chargeons les packages nécessaires pour la gestion et le nettoyage des bases de données  
library(haven)  
library(dplyr)  
library(tidyverse)  
  
# 1. Vérification de la présence ou non de doublons et suppression si oui  
  
base\_menage <- base\_menage %>% distinct()  
mother\_baseline <- mother\_baseline %>% distinct()  
mother\_endline <- mother\_endline %>% distinct()  
child\_baseline <- child\_baseline %>% distinct()  
child\_endline <- child\_endline %>% distinct()  
  
# 2. Vérification et harmonisation des noms des variables entre les bases endline et baseline  
# Affichons les noms pour comparer  
names(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"

names(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"

names(child\_baseline)

## [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"

names(child\_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"

###Visiblement il y a des differeces our le zinc par exemple et l'energie calorique, alors nous allons attribuer aux variables de mother endline les noms dans baseline, Ce sont les mêmes pour les bases enfants donc labas il ne sera pas nécessaire de modifier

# Harmonisation des noms différents des variables pour les bases baseline et endline  
mother\_baseline <- mother\_baseline %>%  
 rename(  
 energ\_kcal = V1,  
 zinc\_mg = V9  
 )

# Maintenant vérifions que les changements ont bien été prises en compte  
setdiff(names(mother\_endline), names(mother\_baseline))

## character(0)

# 3. Corrigeons si possibles les données manquantes  
traiter\_na <- function(df, vars){  
 df %>% mutate(across(all\_of(vars), ~ if\_else(is.na(.), median(., na.rm = TRUE), .)))  
}  
  
vars\_nutrition <- c("energ\_kcal", "protein\_g", "lipid\_tot\_g", "calcium\_mg", "iron\_mg", "zinc\_mg", "vit\_b6\_mg", "vit\_b12\_mcg", "vit\_c\_mg")  
  
# Appliquons la correction sur les bases  
mother\_baseline <- mother\_baseline %>% replace\_na(as.list(colMeans(select(., all\_of(vars\_nutrition)), na.rm = TRUE)))  
mother\_endline <- mother\_endline %>%  
 mutate(across(all\_of(vars\_nutrition), ~if\_else(is.na(.), median(., na.rm=TRUE), .)))  
  
child\_baseline <- child\_baseline %>%  
 mutate(across(all\_of(vars\_nutrition), ~if\_else(is.na(.), median(., na.rm=TRUE), .)))  
  
child\_endline <- child\_endline %>%  
 mutate(across(all\_of(vars\_nutrition), ~if\_else(is.na(.), median(., na.rm=TRUE), .)))

Grâce à la fonction replace\_na\_median on a pu remplacer les valeurs manquantes par la médiane qui est une caractéristique de dispersion. Cette action permettra de corriger les manquements sans risquer de biaiser les résultats

# 4. Calculons l'apport énergétique moyenne à chaque repas pour l'ensemble des mères  
mother\_baseline %>%  
 mutate(Type\_de\_repas = case\_when(  
 s1\_q2 == 1 ~ "Petit-déjeuner",  
 s1\_q2 == 2 ~ "Déjeuner",  
 s1\_q2 == 3 ~ "Dîner",  
 s1\_q2 == 4 ~ "Snacks/Goûter"  
 )) %>%  
 group\_by(Type\_de\_repas) %>%  
 summarise(Energie\_moyenne\_kcal = mean(energ\_kcal, na.rm = TRUE)) %>%  
 kable("html", caption = "Consommation moyenne d'énergie par repas chez les mères (Baseline)")

Consommation moyenne d’énergie par repas chez les mères (Baseline)

Type\_de\_repas

Energie\_moyenne\_kcal

Petit-déjeuner

714.7165

La consommation énergétique moyenne des mères au cours des repas est d’environ 700 kcal par repas, ce qui indique un apport énergétique modéré. Cette valeur est cohérente avec les recommandations générales pour un adulte en bonne santé, dont l’apport calorique quotidien recommandé se situe entre 1800 et 2200 kcal réparties sur trois repas principaux et une collation éventuelle

# 5. Sauvegarde des nouvelles bases traitées et corrigées  
  
write\_dta(mother\_baseline, "mother\_baseline\_v1.dta")  
write\_dta(mother\_endline, "mother\_endline\_v1.dta")  
write\_dta(child\_baseline, "child\_baseline\_v1.dta")  
write\_dta(child\_endline, "child\_endline\_v1.dta")  
write\_dta(base\_menage, "base\_menage\_final.dta")

#PARTIE 2: EMPILEMENT ET FUSION DES DONNEES

##1) BASELINE

###Importation des packages nécessaires  
library(dplyr)  
library(haven)  
  
##Nomenclature des vecteurs pour les nouvelles bases   
mother\_baseline <- read\_dta("mother\_baseline\_v1.dta")  
child\_baseline <- read\_dta("child\_baseline\_v1.dta")  
base\_menage <- read\_dta("base\_menage\_final.dta")  
  
# Renommons les variables nutritionnelles avec \_b à la fin  
mother\_baseline <- mother\_baseline %>% rename\_with(~paste0(., "\_b"), energ\_kcal:vit\_c\_mg)  
child\_baseline <- child\_baseline %>% rename\_with(~paste0(., "\_b"), energ\_kcal:vit\_c\_mg)  
  
# Empilement des bases mère et enfant baselines  
baseline\_combined <- bind\_rows(mother\_baseline, child\_baseline)  
  
# Résumé des consommations journalières totales par individu  
baseline\_total <- baseline\_combined %>%  
 group\_by(hhid, s1\_q2) %>%  
 summarise(across(c(energ\_kcal\_b, protein\_g\_b, lipid\_tot\_g\_b, calcium\_mg\_b, iron\_mg\_b, zinc\_mg\_b), sum, na.rm = TRUE), .groups = "drop")  
  
# Vérifions les variables communes  
intersect(names(baseline\_total), names(base\_menage))  
  
# Ce code va faire l'intersection entre les bases pour trouver les réferences communes afin d'avoir une bonne jointure finale   
  
# Supprime ou renomme la variable gênante dans base\_menage avant la jointure  
base\_menage <- base\_menage %>% select(-s1\_q2)  
  
# Ensuite, effectue clairement ta jointure left\_join :  
baseline\_final <- baseline\_total %>%  
 left\_join(base\_menage, by = "hhid")  
  
# Sauvegarde la base finale baseline  
write\_dta(baseline\_final, "baseline\_final.dta")

##1) ENDLINE

# Mettons les packages nécessaires   
library(dplyr)  
library(haven)  
  
mother\_endline <- read\_dta("mother\_endline\_v1.dta")  
child\_endline <- read\_dta("child\_endline\_v1.dta")  
  
  
# Renommer les variables nutritionnelles pour endline  
mother\_endline <- mother\_endline %>% rename\_with(~paste0(., "\_e"), energ\_kcal:vit\_c\_mg)  
child\_endline <- child\_endline %>% rename\_with(~paste0(., "\_e"), energ\_kcal:vit\_c\_mg)  
  
# Empilement Endline  
endline\_combined <- bind\_rows(mother\_endline, child\_endline)  
  
# Calcul de la consommation journalière totales par individu (endline)  
endline\_total <- endline\_combined %>%  
 group\_by(hhid) %>%  
 summarise(across(c(energ\_kcal\_e, protein\_g\_e, lipid\_tot\_g\_e, calcium\_mg\_e, iron\_mg\_e, zinc\_mg\_e), sum, na.rm = TRUE), .groups = 'drop')

## Warning: There was 1 warning in `summarise()`.  
## ℹ In argument: `across(...)`.  
## ℹ In group 1: `hhid = "4948484848535052"`.  
## Caused by warning:  
## ! The `...` argument of `across()` is deprecated as of dplyr 1.1.0.  
## Supply arguments directly to `.fns` through an anonymous function instead.  
##   
## # Previously  
## across(a:b, mean, na.rm = TRUE)  
##   
## # Now  
## across(a:b, \(x) mean(x, na.rm = TRUE))

# Jointure Endline avec base ménage  
endline\_final <- endline\_total %>% left\_join(base\_menage, by = "hhid")  
  
write\_dta(endline\_final, "endline\_final.dta")

#SUITE PARTIE 2 APRES ENDLINE

library(dplyr)  
library(haven)  
  
  
# Importons les bases créées précédemment  
base\_menage <- read\_dta("base\_menage\_final.dta")  
baseline\_final <- read\_dta("baseline\_final.dta")  
endline\_final <- read\_dta("endline\_final.dta")  
  
  
# 3. Fusion des deux bases finales endline et baseline  
base\_finale <- inner\_join(baseline\_final, endline\_final, by = c("hhid"))  
  
# 4. Fusion avec la base ménage pour intégrer les caractéristiques socio-économiques  
base\_menage <- read\_dta("base\_menage\_final.dta")  
base\_finale <- left\_join(base\_finale, base\_menage, by = "hhid")

#PARTIE3: STATISTIQUES DESCRIPTIVES

Dans cette partie nous allons porter l’attention sur la base child baseline et nous allons nous intéresser à la nutrition infantile

library(gtsummary)  
library(dplyr)  
library(haven)  
  
# Importer la base des enfants baseline  
child\_baseline <- read\_dta("child\_baseline\_v1.dta")  
  
# Sélection des variables pertinentes pour l'analyse nutritionnelle  
child\_baseline <- child\_baseline %>%   
 select(energ\_kcal, protein\_g, lipid\_tot\_g, calcium\_mg, iron\_mg, zinc\_mg, vit\_b6\_mg, vit\_b12\_mcg, vit\_c\_mg)  
  
# Création du tableau de statistiques descriptives  
nutrition\_summary <- child\_baseline %>%  
 tbl\_summary(  
 statistic = list(all\_continuous() ~ "{mean} ({sd})"),  
 label = list(  
 energ\_kcal ~ "Énergie (kcal)",  
 protein\_g ~ "Protéines (g)",  
 lipid\_tot\_g ~ "Lipides totaux (g)",  
 calcium\_mg ~ "Calcium (mg)",  
 iron\_mg ~ "Fer (mg)",  
 zinc\_mg ~ "Zinc (mg)",  
 vit\_b6\_mg ~ "Vitamine B6 (mg)",  
 vit\_b12\_mcg ~ "Vitamine B12 (mcg)",  
 vit\_c\_mg ~ "Vitamine C (mg)"  
 ),  
 digits = everything() ~ c(0),   
 missing = "always",  
 missing\_text = "valeurs manquantes") %>%  
 modify\_header(label = "Caractéristiques nutritionnelles")  
  
# Affichage du tableau  
nutrition\_summary

| Caractéristiques nutritionnelles | **N = 4,256***1* |
| --- | --- |
| Énergie (kcal) | 405 (307) |
| valeurs manquantes | 0 |
| Protéines (g) | 13 (16) |
| valeurs manquantes | 0 |
| Lipides totaux (g) | 7 (14) |
| valeurs manquantes | 0 |
| Calcium (mg) | 76 (120) |
| valeurs manquantes | 0 |
| Fer (mg) | 5 (6) |
| valeurs manquantes | 0 |
| Zinc (mg) | 2 (2) |
| valeurs manquantes | 0 |
| Vitamine B6 (mg) | 0 (0) |
| valeurs manquantes | 0 |
| Vitamine B12 (mcg) | 0 (1) |
| valeurs manquantes | 0 |
| Vitamine C (mg) | 21 (67) |
| valeurs manquantes | 0 |
| *1*Mean (SD) | |