Test\_R

Lawa Foumsou Prosper

# Chargement de packages nécessaires

# Chargement des bases

Nous chargeons en premier lieu les bases qui seront utilisées lors du travail pratique.

## Base ménage

base\_men <- haven::read\_dta("../data/base\_menage.dta")  
colnames(base\_men) # voir le nom des variables dans la base

## [1] "regionid" "communeid" "villageid" "hhid"   
## [5] "hhsize" "poly" "hh\_primary" "s1\_q2"   
## [9] "s1\_q4a" "s2\_q1" "s2\_q2" "s2\_q4"   
## [13] "s29\_q1" "demgrp1" "demgrp2" "demgrp3"   
## [17] "demgrp4" "demgrp5" "dependencyratio" "hfias\_score"   
## [21] "T1"

La base ménage est composée de 1065 individus et 21 variables.

dim(base\_men) # connaitre le nombre d'observations et de variables dans la base

## [1] 1065 21

## Base child baseline

base\_child\_baseline <- haven::read\_dta("../data/food\_comp\_child\_baseline.dta")  
colnames(base\_men) # voir le nom des variables dans la base

## [1] "regionid" "communeid" "villageid" "hhid"   
## [5] "hhsize" "poly" "hh\_primary" "s1\_q2"   
## [9] "s1\_q4a" "s2\_q1" "s2\_q2" "s2\_q4"   
## [13] "s29\_q1" "demgrp1" "demgrp2" "demgrp3"   
## [17] "demgrp4" "demgrp5" "dependencyratio" "hfias\_score"   
## [21] "T1"

La base child baseline est composée de 4256 individus et 17 variables.

dim(base\_child\_baseline) # connaitre le nombre d'observations et de variables dans la base

## [1] 4256 17

## Base child endline

base\_child\_endline <- haven::read\_dta("../data/food\_comp\_child\_endline.dta")  
colnames(base\_child\_endline) # voir le nom des variables dans la base

## [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"

La base child endline est composée de 4256 individus et 17 variables.

dim(base\_child\_endline) # connaitre le nombre d'observations et de variables dans la base

## [1] 4256 17

## Base mother baseline

base\_mother\_baseline <- haven::read\_dta("../data/food\_comp\_mother\_baseline.dta")  
colnames(base\_mother\_baseline) # voir le nom des variables dans la base

## [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"

La base mother baseline est composée de 4256 individus et 17 variables.

dim(base\_mother\_baseline) # connaitre le nombre d'observations et de variables dans la base

## [1] 4256 17

## Base mother endline

base\_mother\_endline <- haven::read\_dta("../data/food\_comp\_mother\_endline.dta")   
colnames(base\_mother\_endline) # voir le nom des variables dans la base

## [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"

La base mother endline est composée de 4256 individus et 17 variables.

dim(base\_mother\_baseline) # connaitre le nombre d'observations et de variables dans la base

## [1] 4256 17

# **Partie 1 : Gestion et nettoyage des bases de données**

## Q1.

Verifions s’il n’y a pas de valeurs manquantes dans les identifiants.

base\_men[is.na(base\_men$regionid) |   
 is.na(base\_men$communeid) |   
 is.na(base\_men$villageid) |   
 is.na(base\_men$hhid), ]

## # A tibble: 1 x 21  
## regionid communeid villageid hhid hhsize poly hh\_primary s1\_q2 s1\_q4a  
## <dbl> <dbl> <dbl> <chr> <dbl> <dbl+l> <dbl+lbl> <dbl+l> <dbl>  
## 1 1 NA 33 5151485~ 8 0 [Non] 0 [Non] 1 [Mal~ 49  
## # i 12 more variables: s2\_q1 <dbl+lbl>, s2\_q2 <dbl+lbl>, s2\_q4 <dbl+lbl>,  
## # s29\_q1 <dbl+lbl>, demgrp1 <dbl>, demgrp2 <dbl>, demgrp3 <dbl>,  
## # demgrp4 <dbl>, demgrp5 <dbl>, dependencyratio <dbl>, hfias\_score <dbl>,  
## # T1 <dbl+lbl>

On remarque qu’il y a une valeur manquantes dans l’identifiant **communeid**.

base\_men %>%  
 drop\_na(regionid, communeid, villageid, hhid) %>% # Ignore les NA  
 group\_by(regionid, communeid, villageid, hhid) %>%  
 mutate(nb\_line = n()) %>%  
 ungroup() %>%  
 filter(nb\_line > 1) %>%  
 nrow()

## [1] 2

On remarque qu’il y a deux observations avec doublons.

# Identification du nombre de duplication des ménages.   
base\_men %>%   
 group\_by(regionid, communeid, villageid, hhid) %>%   
 mutate(nb\_line=n()) %>% arrange(desc(nb\_line)) %>%   
 ungroup %>%  
 count(nb\_line)

## # A tibble: 2 x 2  
## nb\_line n  
## <int> <int>  
## 1 1 1063  
## 2 2 2

Dans la suite nous allons directement procéder à la suppression des doublons s’il existent.

# Suppression des doublons dans la base ménage  
base\_men<-base\_men %>%   
 distinct(regionid, communeid, villageid, hhid, .keep\_all = T)

# Suppression des doublons dans la base base\_child\_baseline  
base\_child\_baseline<-base\_child\_baseline %>%   
 distinct(regionid, communeid, villageid, hhid, .keep\_all = T)

# Suppression des doublons dans la base base\_child\_endline  
base\_child\_endline<-base\_child\_endline %>%   
 distinct(regionid, communeid, villageid, hhid, .keep\_all = T)

# Suppression des doublons dans la base mother\_baseline  
base\_mother\_baseline<-base\_mother\_baseline %>%   
 distinct(regionid, communeid, villageid, hhid, .keep\_all = T)

# Suppression des doublons dans la base mother\_endline  
base\_mother\_endline<-base\_mother\_endline %>%   
 distinct(regionid, communeid, villageid, hhid, .keep\_all = T)

## Q2

# Récupérer les noms des variables  
var\_base\_child\_baseline <- names(base\_child\_baseline)  
var\_base\_child\_endline <- names(base\_child\_endline)  
var\_base\_mother\_baseline <- names(base\_mother\_baseline)  
var\_base\_mother\_endline <- names(base\_mother\_endline)  
  
# Comparer les noms des variables pour les enfants  
diff\_child\_baseline\_endline <- setdiff(var\_base\_child\_baseline, var\_base\_child\_endline)  
diff\_child\_endline\_baseline <- setdiff(var\_base\_child\_endline, var\_base\_child\_baseline)  
  
# Comparer les noms des variables pour les mères  
diff\_mother\_baseline\_endline <- setdiff(var\_base\_mother\_baseline, var\_base\_mother\_endline)  
diff\_mother\_endline\_baseline <- setdiff(var\_base\_mother\_endline, var\_base\_mother\_baseline)  
  
# Afficher les différences  
cat("Variables présentes dans base\_child\_baseline mais absentes de base\_child\_endline :\n")

## Variables présentes dans base\_child\_baseline mais absentes de base\_child\_endline :

print(diff\_child\_baseline\_endline)

## character(0)

cat("\nVariables présentes dans base\_child\_endline mais absentes de base\_child\_baseline :\n")

##   
## Variables présentes dans base\_child\_endline mais absentes de base\_child\_baseline :

print(diff\_child\_endline\_baseline)

## character(0)

cat("\nVariables présentes dans base\_mother\_baseline mais absentes de base\_mother\_endline :\n")

##   
## Variables présentes dans base\_mother\_baseline mais absentes de base\_mother\_endline :

print(diff\_mother\_baseline\_endline)

## [1] "V1" "V9"

cat("\nVariables présentes dans base\_mother\_endline mais absentes de base\_mother\_baseline :\n")

##   
## Variables présentes dans base\_mother\_endline mais absentes de base\_mother\_baseline :

print(diff\_mother\_endline\_baseline)

## [1] "energ\_kcal" "zinc\_mg"

On remarque qu’il y a des variables dont les noms sont différents dans les base des mères.

### Renommage des variables

# Renommer V1 en energ\_kcal et V9 en zinc\_mg dans base\_mother\_baseline  
base\_mother\_baseline <- base\_mother\_baseline %>%  
 rename(  
 energ\_kcal = V1,  
 zinc\_mg = V9  
 )  
  
# Vérifier si le renommage a bien été fait  
names(base\_mother\_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"

## Q3

Fonction pour compter les nombre de valeurs manquantes pour chaque variable dans les différentes bases.

count\_na <- function(df, df\_name = "Base") {  
 missing\_counts <- colSums(is.na(df)) # Compte le nombre de NA par colonne  
 missing\_df <- data.frame(Variable = names(df), Missing\_Count = missing\_counts)  
 missing\_df <- missing\_df[order(-missing\_df$Missing\_Count), ] # Trier par nombre de NA décroissant  
   
 cat("\nNombre de valeurs manquantes par variable pour", df\_name, ":\n")  
 print(missing\_df)  
   
 return(missing\_df)  
}

na\_base\_men <- count\_na(base\_men, "base\_men")

##   
## Nombre de valeurs manquantes par variable pour base\_men :  
## Variable Missing\_Count  
## communeid communeid 1  
## regionid regionid 0  
## villageid villageid 0  
## hhid hhid 0  
## hhsize hhsize 0  
## poly poly 0  
## hh\_primary hh\_primary 0  
## s1\_q2 s1\_q2 0  
## s1\_q4a s1\_q4a 0  
## s2\_q1 s2\_q1 0  
## s2\_q2 s2\_q2 0  
## s2\_q4 s2\_q4 0  
## s29\_q1 s29\_q1 0  
## demgrp1 demgrp1 0  
## demgrp2 demgrp2 0  
## demgrp3 demgrp3 0  
## demgrp4 demgrp4 0  
## demgrp5 demgrp5 0  
## dependencyratio dependencyratio 0  
## hfias\_score hfias\_score 0  
## T1 T1 0

na\_baseline\_mother <- count\_na(base\_mother\_baseline, "base\_mother\_baseline")

##   
## Nombre de valeurs manquantes par variable pour base\_mother\_baseline :  
## Variable Missing\_Count  
## energ\_kcal energ\_kcal 68  
## protein\_g protein\_g 68  
## lipid\_tot\_g lipid\_tot\_g 68  
## calcium\_mg calcium\_mg 68  
## iron\_mg iron\_mg 68  
## zinc\_mg zinc\_mg 68  
## vit\_b6\_mg vit\_b6\_mg 68  
## vit\_b12\_mcg vit\_b12\_mcg 68  
## vit\_c\_mg vit\_c\_mg 68  
## communeid communeid 2  
## regionid regionid 1  
## villageid villageid 0  
## hhid hhid 0  
## round round 0  
## s1\_q0 s1\_q0 0  
## s1\_q1 s1\_q1 0  
## s1\_q2 s1\_q2 0

na\_endline\_mother <- count\_na(base\_mother\_endline, "base\_mother\_endline")

##   
## Nombre de valeurs manquantes par variable pour base\_mother\_endline :  
## Variable Missing\_Count  
## energ\_kcal energ\_kcal 78  
## protein\_g protein\_g 78  
## lipid\_tot\_g lipid\_tot\_g 78  
## calcium\_mg calcium\_mg 78  
## iron\_mg iron\_mg 78  
## zinc\_mg zinc\_mg 78  
## vit\_b6\_mg vit\_b6\_mg 78  
## vit\_b12\_mcg vit\_b12\_mcg 78  
## vit\_c\_mg vit\_c\_mg 78  
## regionid regionid 0  
## communeid communeid 0  
## villageid villageid 0  
## hhid hhid 0  
## round round 0  
## s1\_q0 s1\_q0 0  
## s1\_q1 s1\_q1 0  
## s1\_q2 s1\_q2 0

na\_baseline\_child <- count\_na(base\_child\_baseline, "base\_child\_baseline")

##   
## Nombre de valeurs manquantes par variable pour base\_child\_baseline :  
## Variable Missing\_Count  
## energ\_kcal energ\_kcal 34  
## protein\_g protein\_g 34  
## lipid\_tot\_g lipid\_tot\_g 34  
## calcium\_mg calcium\_mg 34  
## iron\_mg iron\_mg 34  
## zinc\_mg zinc\_mg 34  
## vit\_b6\_mg vit\_b6\_mg 34  
## vit\_b12\_mcg vit\_b12\_mcg 34  
## vit\_c\_mg vit\_c\_mg 34  
## regionid regionid 0  
## communeid communeid 0  
## villageid villageid 0  
## hhid hhid 0  
## round round 0  
## s1\_q0 s1\_q0 0  
## s1\_q1 s1\_q1 0  
## s1\_q2 s1\_q2 0

na\_endline\_child <- count\_na(base\_child\_endline, "base\_child\_endline")

##   
## Nombre de valeurs manquantes par variable pour base\_child\_endline :  
## Variable Missing\_Count  
## energ\_kcal energ\_kcal 54  
## protein\_g protein\_g 54  
## lipid\_tot\_g lipid\_tot\_g 54  
## calcium\_mg calcium\_mg 54  
## iron\_mg iron\_mg 54  
## zinc\_mg zinc\_mg 54  
## vit\_b6\_mg vit\_b6\_mg 54  
## vit\_b12\_mcg vit\_b12\_mcg 54  
## vit\_c\_mg vit\_c\_mg 54  
## regionid regionid 0  
## communeid communeid 0  
## villageid villageid 0  
## hhid hhid 0  
## round round 0  
## s1\_q0 s1\_q0 0  
## s1\_q1 s1\_q1 0  
## s1\_q2 s1\_q2 0

On va juste supprimer les données manquantes dans chaque base. Vu qu’il n’y aura pas une extrapolation après et que le nombre de NA est relativement petit au vu du nombre total d’observation dans chaque base.

## Q4

# Recalculer la moyenne par groupe  
consommation\_moyenne\_par\_groupe <- base\_mother\_baseline %>%  
 group\_by(s1\_q0) %>%  
 summarise(consommation\_moyenne = mean(energ\_kcal, na.rm = TRUE))  
  
# Afficher le résultat  
print(consommation\_moyenne\_par\_groupe)

## # A tibble: 1 x 2  
## s1\_q0 consommation\_moyenne  
## <dbl+lbl> <dbl>  
## 1 1 [Breakfast] 637.

## Q5

On va sauvegarder les nouvelles bases en supprimant les données manquantes.

base\_menage\_final <- base\_men %>% drop\_na()  
mother\_baseline\_v1 <- base\_mother\_baseline %>% drop\_na()  
mother\_endline\_V1 <- base\_mother\_endline %>% drop\_na()  
child\_baseline\_v1 <- base\_child\_baseline %>% drop\_na()  
child\_endline\_v1 <- base\_child\_endline %>% drop\_na()

# **Partie 2 : Empilement et Fusion des données**

## Baseline

### Q1

# Empiler deux bases de données (base1 et base2)  
base1 <- bind\_rows(mother\_baseline\_v1, child\_baseline\_v1)  
colnames(base1)

## [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"

## Q2

# Identifier les indices des colonnes entre energ\_kcal et vit\_c\_mcg  
start\_col <- which(names(base1) == "energ\_kcal")  
end\_col <- which(names(base1) == "vit\_c\_mg")  
  
# Sélectionner les colonnes entre energ\_kcal et vit\_c\_mcg  
cols\_to\_rename <- names(base1)[start\_col:end\_col]  
  
# Renommer ces colonnes en ajoutant le suffixe \_b  
base1 <- base1 %>%  
 rename\_with(~paste0(., "\_b"), all\_of(cols\_to\_rename))  
  
# Vérifier les premières lignes pour vérifier les changements  
colnames(base1)

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

## Q3

base2 <- base1 %>%  
 group\_by(s1\_q0) %>% # Grouper par s1\_q0  
 mutate(  
 sum\_values = 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  
 ) %>%  
 ungroup() # Dé-grouper une fois le calcul effectué  
  
# Résultat  
colnames(base2)

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

## Q4

baseline\_final <- base2 %>%  
 select(hhid, 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, sum\_values)  
  
# Sauvegarder le fichier sous le nom 'baseline\_final.dta'  
write\_dta(baseline\_final, "baseline\_final.dta")  
dim(baseline\_final)

## [1] 2023 11

## Endline

### Q1

# Empiler deux bases de données (base1 et base2)  
base3 <- bind\_rows(mother\_endline\_V1, child\_endline\_v1)  
dim(base3)

## [1] 1996 17

colnames(base3)

## [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"

### Q2

# Identifier les indices des colonnes entre energ\_kcal et vit\_c\_mcg  
start\_col <- which(names(base3) == "energ\_kcal")  
end\_col <- which(names(base3) == "vit\_c\_mg")  
  
# Sélectionner les colonnes entre energ\_kcal et vit\_c\_mcg  
cols\_to\_rename <- names(base3)[start\_col:end\_col]  
  
# Renommer ces colonnes en ajoutant le suffixe \_b  
base3 <- base3 %>%  
 rename\_with(~paste0(., "\_e"), all\_of(cols\_to\_rename))  
  
# Vérifier les premières lignes pour vérifier les changements  
colnames(base3)

## [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"

### Q3

base4 <- base3 %>%  
 group\_by(s1\_q0) %>% # Grouper par s1\_q0  
 mutate(  
 sum\_values = energ\_kcal\_e + protein\_g\_e + lipid\_tot\_g\_e +   
 calcium\_mg\_e + iron\_mg\_e + zinc\_mg\_e +   
 vit\_b6\_mg\_e + vit\_b12\_mcg\_e + vit\_c\_mg\_e  
 ) %>%  
 ungroup() # Dé-grouper une fois le calcul effectué  
  
# Résultat : afficher les noms des colonnes du dataframe  
colnames(base4)

## [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" "sum\_values"

### Q4

endline\_final <- base4 %>%  
 select(hhid, energ\_kcal\_e, protein\_g\_e, lipid\_tot\_g\_e, calcium\_mg\_e,   
 iron\_mg\_e, zinc\_mg\_e, vit\_b6\_mg\_e, vit\_b12\_mcg\_e, vit\_c\_mg\_e, sum\_values)  
  
# Sauvegarder le fichier sous le nom 'baseline\_final.dta'  
write\_dta(endline\_final, "baseline\_final.dta")

colnames(baseline\_final)

## [1] "hhid" "energ\_kcal\_b" "protein\_g\_b" "lipid\_tot\_g\_b"  
## [5] "calcium\_mg\_b" "iron\_mg\_b" "zinc\_mg\_b" "vit\_b6\_mg\_b"   
## [9] "vit\_b12\_mcg\_b" "vit\_c\_mg\_b" "sum\_values"

base\_finale1 <- baseline\_final %>% left\_join(endline\_final, by="hhid")  
colnames(base\_finale1)

## [1] "hhid" "energ\_kcal\_b" "protein\_g\_b" "lipid\_tot\_g\_b"  
## [5] "calcium\_mg\_b" "iron\_mg\_b" "zinc\_mg\_b" "vit\_b6\_mg\_b"   
## [9] "vit\_b12\_mcg\_b" "vit\_c\_mg\_b" "sum\_values.x" "energ\_kcal\_e"   
## [13] "protein\_g\_e" "lipid\_tot\_g\_e" "calcium\_mg\_e" "iron\_mg\_e"   
## [17] "zinc\_mg\_e" "vit\_b6\_mg\_e" "vit\_b12\_mcg\_e" "vit\_c\_mg\_e"   
## [21] "sum\_values.y"

base\_finale <- base\_finale1 %>% left\_join(base\_menage\_final, by="hhid")  
colnames(base\_finale)

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

# **Partie 3 : Statistiques descriptives avec le package gtsummary**

base\_finale %>%  
 labelled::to\_factor() %>%  
 select(  
 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, sum\_values.x, energ\_kcal\_e,   
 protein\_g\_e, lipid\_tot\_g\_e, calcium\_mg\_e, iron\_mg\_e, zinc\_mg\_e, vit\_b6\_mg\_e,   
 vit\_b12\_mcg\_e, vit\_c\_mg\_e  
 ) %>%  
 tbl\_summary(  
 digits = everything() ~ c(0, 0, 0, 0),  
 missing = "always",  
 missing\_text = "Valeurs manquantes"  
 ) %>%  
 modify\_header(  
 label = "\*\*Taille de l'échantillon\*\*"  
 ) %>%  
 as\_gt() %>%  
 gt::tab\_header(  
 title = "Caractéristiques du logement"  
 ) %>%  
 gt::tab\_source\_note(  
 source\_note = "Source : Calcul des auteurs"  
 ) %>%  
 gt::tab\_options(  
 heading.title.font.size = 20,  
 heading.title.font.weight = "bold"  
 )

Table 1: Caractéristiques du logement

| **Taille de l'échantillon** | **N = 3,862***1* |
| --- | --- |
| energ\_kcal\_b | 407 (236, 658) |
| Valeurs manquantes | 0 |
| protein\_g\_b | 12 (6, 20) |
| Valeurs manquantes | 0 |
| lipid\_tot\_g\_b | 3 (1, 7) |
| Valeurs manquantes | 0 |
| calcium\_mg\_b | 30 (14, 88) |
| Valeurs manquantes | 0 |
| iron\_mg\_b | 4 (1, 9) |
| Valeurs manquantes | 0 |
| zinc\_mg\_b | 2 (1, 3) |
| Valeurs manquantes | 0 |
| vit\_b6\_mg\_b | 0 (0, 0) |
| Valeurs manquantes | 0 |
| vit\_b12\_mcg\_b | 0 (0, 0) |
| Valeurs manquantes | 0 |
| vit\_c\_mg\_b | 0 (0, 2) |
| Valeurs manquantes | 0 |
| sum\_values.x | 511 (286, 798) |
| Valeurs manquantes | 0 |
| energ\_kcal\_e | 371 (216, 618) |
| Valeurs manquantes | 72 |
| protein\_g\_e | 10 (5, 18) |
| Valeurs manquantes | 72 |
| lipid\_tot\_g\_e | 2 (1, 5) |
| Valeurs manquantes | 72 |
| calcium\_mg\_e | 23 (9, 82) |
| Valeurs manquantes | 72 |
| iron\_mg\_e | 3 (1, 7) |
| Valeurs manquantes | 72 |
| zinc\_mg\_e | 2 (1, 3) |
| Valeurs manquantes | 72 |
| vit\_b6\_mg\_e | 0 (0, 0) |
| Valeurs manquantes | 72 |
| vit\_b12\_mcg\_e | 0 (0, 0) |
| Valeurs manquantes | 72 |
| vit\_c\_mg\_e | 0 (0, 2) |
| Valeurs manquantes | 72 |
| *1*Median (IQR) | |
| Source : Calcul des auteurs | |