Statistique descriptives

ILLY Jacques

2025-02-25

#Installation des packages

#listes des packages à installer  
packages<-c("dplyr","janitor", "readr", "cardx" ,"utils", "haven", "survey", "gtsummary", "labelled","gt")  
#installation des package  
for (pack in packages)  
 {if (!requireNamespace(pack, quietly=TRUE))  
 {install.packages(pack)  
 }  
 #chargement des packages  
 library(pack, character.only=TRUE)  
}

##   
## 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

##   
## Attachement du package : 'janitor'

## Les objets suivants sont masqués depuis 'package:stats':  
##   
## chisq.test, fisher.test

## 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

#Importation des bases

#Importation de la base menages  
base\_menage<-haven::read\_dta("..\\Données\\base\_menage.dta")  
  
#Importation de la base food\_comp\_child\_baseline  
base\_child\_baseline<-haven::read\_dta("..\\Données\\food\_comp\_child\_baseline.dta")  
  
#Importation de la base food\_comp\_child\_endline  
base\_child\_endline<-haven::read\_dta("..\\Données\\food\_comp\_child\_endline.dta")  
  
#Importation de la base food\_comp\_mother\_baseline  
base\_mother\_baseline<-haven::read\_dta("..\\Données\\food\_comp\_mother\_baseline.dta")  
  
#Importation de la base food\_comp\_child\_endline  
base\_mother\_endline<-haven::read\_dta("..\\Données\\food\_comp\_mother\_endline.dta")

View(base\_child\_baseline)

#Partie 1 : Gestion et nettoyage des bases de données

## Vérifieons la présence de doublons dans la base de données Baseline

### Dans la base food\_comp\_child\_baseline

# Vérification de la présence de doublons   
base\_doublons <- base\_child\_baseline %>%  
 duplicated()%>%  
 any()  
  
# Afficher le résultat  
if (base\_doublons) {  
 print("Des doublons sont présents dans les données.")  
 #supression des doublons  
base\_child\_baseline\_sans\_doublons <- base\_child\_baseline %>%  
 distinct()  
# Au contraire on affiche  
} else {  
 print("Aucun doublon trouvé.")  
   
 base\_child\_baseline\_sans\_doublons <- base\_child\_baseline   
}

## [1] "Aucun doublon trouvé."

### Dans la base food\_comp\_mother\_baseline

# Vérification de la présence de doublons   
base\_doublons <- base\_mother\_baseline %>%  
 duplicated()%>%  
 any()  
  
# Afficher le résultat  
if (base\_doublons) {  
 print("Des doublons sont présents dans les données.")  
 #supression des doublons  
base\_mother\_baseline\_sans\_doublons <- base\_child\_baseline %>%  
 distinct()  
# Au contraire on affiche  
} else {  
 print("Aucun doublon trouvé.")  
   
 base\_mother\_baseline\_sans\_doublons <- base\_child\_baseline  
}

## [1] "Aucun doublon trouvé."

## Vérifieons la présence de doublons dans la base de données endline

### Dans la base food\_comp\_child\_endline

# Vérification de la présence de doublons   
base\_doublons <- base\_child\_endline %>%  
 duplicated()%>%  
 any()  
  
# Afficher le résultat  
if (base\_doublons) {  
 print("Des doublons sont présents dans les données.")  
 #supression des doublons  
base\_child\_endline\_sans\_doublons <- base\_child\_endline %>%  
 distinct()  
# Au contraire on affiche  
} else {  
 print("Aucun doublon trouvé.")  
   
 base\_child\_endline\_sans\_doublons <- base\_child\_endline  
}

## [1] "Aucun doublon trouvé."

### Dans la base food\_comp\_mother\_endline

# Vérification de la présence de doublons   
base\_doublons <- base\_mother\_endline %>%  
 duplicated()%>%  
 any()  
  
# Afficher le résultat  
if (base\_doublons) {  
 print("Des doublons sont présents dans les données.")  
 #supression des doublons  
base\_mother\_endline\_sans\_doublons <- base\_child\_endline %>%  
 distinct()  
# Au contraire on affiche  
} else {  
 print("Aucun doublon trouvé.")  
   
 base\_mother\_endline\_sans\_doublons <- base\_child\_endline  
}

## [1] "Aucun doublon trouvé."

### Dans la base menage

# Vérification de la présence de doublons   
base\_doublons <- base\_menage %>%  
 duplicated()%>%  
 any()  
  
# Afficher le résultat  
if (base\_doublons) {  
 print("Des doublons sont présents dans les données.")  
 #supression des doublons  
base\_menage\_sans\_doublons <- base\_menage %>%  
 distinct()  
# Au contraire on affiche  
} else {  
 print("Aucun doublon trouvé.")  
   
 base\_menage\_sans\_doublons <- base\_menage  
}

## [1] "Des doublons sont présents dans les données."

View(base\_child\_baseline\_sans\_doublons)

## 2.les noms des variables

# Afficher les noms des variables dans la base\_child\_baseline\_sans\_doublons  
names(base\_child\_baseline\_sans\_doublons)

## [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"

# Afficher les noms des variables dans la base\_child\_endline\_sans\_doublons  
names(base\_child\_endline\_sans\_doublons)

## [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"

# Afficher les noms des variables dans la base\_mother\_baseline\_sans\_doublons  
names(base\_mother\_baseline\_sans\_doublons)

## [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"

# Afficher les noms des variables dans la base\_mother\_endline\_sans\_doublons  
names(base\_mother\_endline\_sans\_doublons)

## [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"

# Vérifions les noms des variables après   
setdiff(names(base\_child\_baseline\_sans\_doublons), names(base\_child\_endline\_sans\_doublons)) # Vérifier les variables manquantes dans Endline

## character(0)

setdiff(names(base\_child\_endline\_sans\_doublons), names(base\_child\_endline\_sans\_doublons)) # Vérifier les variables manquantes dans Baseline

## character(0)

On remarque que les noms des variables sont cohérent

# Vérifions les noms des variables après   
setdiff(names(base\_mother\_baseline\_sans\_doublons), names(base\_mother\_endline\_sans\_doublons)) # Vérifier les variables manquantes dans Endline

## character(0)

setdiff(names(base\_mother\_endline\_sans\_doublons), names(base\_mother\_endline\_sans\_doublons)) # Vérifier les variables manquantes dans Baseline

## character(0)

# Vérifions les noms des variables après   
base\_child\_baseline\_sans\_doublons <- base\_child\_baseline\_sans\_doublons %>% clean\_names()  
  
base\_child\_endline\_sans\_doublons <- base\_child\_endline\_sans\_doublons %>% clean\_names()  
  
base\_mother\_baseline\_sans\_doublons <- base\_mother\_baseline\_sans\_doublons %>% clean\_names()  
  
base\_mother\_endline\_sans\_doublons <- base\_mother\_endline\_sans\_doublons %>% clean\_names()

head(base\_mother\_baseline\_sans\_doublons)

## # A tibble: 6 × 17  
## regionid communeid villageid hhid round s1\_q0 s1\_q1 s1\_q2 energ\_kcal  
## <dbl> <dbl> <dbl> <chr> <dbl+l> <dbl+l> <dbl+l> <dbl+l> <dbl>  
## 1 2 25 1000 49484… 1 [Bas… 1 [Bre… 1 [Yes] 2 [chi… 355.  
## 2 2 25 1000 49484… 1 [Bas… 2 [Lun… 1 [Yes] 2 [chi… 224.  
## 3 2 25 1000 49484… 1 [Bas… 3 [Din… 1 [Yes] 2 [chi… 334.  
## 4 2 25 1000 49484… 1 [Bas… 4 [Sna… 1 [Yes] 2 [chi… 494.  
## 5 2 25 1000 49484… 1 [Bas… 1 [Bre… 1 [Yes] 2 [chi… 235.  
## 6 2 25 1000 49484… 1 [Bas… 2 [Lun… 1 [Yes] 2 [chi… 216.  
## # ℹ 8 more variables: protein\_g <dbl>, lipid\_tot\_g <dbl>, calcium\_mg <dbl>,  
## # iron\_mg <dbl>, zinc\_mg <dbl>, vit\_b6\_mg <dbl>, vit\_b12\_mcg <dbl>,  
## # vit\_c\_mg <dbl>

L’énergie moyenne est:

# Calculer la consommation d'énergie moyenne par repas pour l'ensemble des mères  
consommation\_moyenne\_energie <- base\_mother\_baseline\_sans\_doublons %>%  
 summarise(consommation\_moyenne = mean(energ\_kcal, na.rm = TRUE))  
  
# Afficher le résultat  
print(consommation\_moyenne\_energie)

## # A tibble: 1 × 1  
## consommation\_moyenne  
## <dbl>  
## 1 412.

1. Sauvegarde:

mother\_baseline\_v1.dta<-base\_mother\_baseline\_sans\_doublons  
mother\_endline\_v1.dta<-base\_child\_endline\_sans\_doublons  
child\_baseline\_v1.dta<-base\_child\_baseline\_sans\_doublons  
child\_endline\_v1.dta<-base\_child\_endline\_sans\_doublons  
base\_menage.dta <-base\_menage\_sans\_doublons

# Partie 2 : Empilement et Fusion des données

## 1) Baseline

#Nous ferrons une jointure gauche  
base1<-mother\_baseline\_v1.dta %>% left\_join(child\_baseline\_v1.dta, by= "hhid")

## Warning in left\_join(., child\_baseline\_v1.dta, by = "hhid"): Detected an unexpected many-to-many relationship between `x` and `y`.  
## ℹ Row 1 of `x` matches multiple rows in `y`.  
## ℹ Row 1 of `y` matches multiple rows in `x`.  
## ℹ If a many-to-many relationship is expected, set `relationship =  
## "many-to-many"` to silence this warning.

View(base1)

Renommons les variables

**je n’ai pas les compétences nécessaires pour traiter cette partie. je passe à la partie trois**

#Partie 3 : Statistiques descriptives avec le package gtsummary

Nous nous proposons de faire de statistiques descriptives sans ponderation sur quelques variables.

#Statistiques descriptive   
stat\_base<- mother\_baseline\_v1.dta %>% labelled::to\_factor() %>% select(regionid, communeid, villageid) %>% tbl\_summary(  
 label = list(  
 regionid ~ "La région",  
 communeid ~ "la commune",  
 villageid ~ "Milieu de résidence"  
 ),  
  
 missing = "always",  
 missing\_text="val manquant"  
) %>% modify\_header(label="statistiques descriptives sur la base individu") #%>% as\_gt() %>% gt:: tab\_source\_note(source\_note="souce: Donné de EHCVM 2018-2019 Togo")

#mise en forme du tableau avec la le package gt  
stat\_base\_ind1<- as\_gt(stat\_base)   
stat\_base\_ind1 %>%   
 tab\_header(title="Taille") %>%  
 #Mettons la bordure en vert pour toutes les colonnes  
 tab\_style(style=cell\_borders(color="green", weight = px(4)), locations = cells\_body(columns=everything())) %>%  
 #Mettons le fonf en jaune  
 tab\_style(style=cell\_fill(color="yellow"), locations = cells\_body(columns=everything())) %>%   
 #mettons le texte en marron et en gras  
 tab\_style(style=cell\_text(color="darkred", weight="bold" ), locations = cells\_body(columns=everything()))%>%  
 #Le titrage en bas   
 gt:: tab\_source\_note(source\_note="souce: Donné de EHCVM 2018-2019 Togo")

Table 1: Taille

| statistiques descriptives sur la base individu | **N = 4,256***1* |
| --- | --- |
| **La région** |  |
| **1** | **752 (18%)** |
| **2** | **2,476 (58%)** |
| **3** | **1,028 (24%)** |
| **val manquant** | **0** |
| **la commune** | **27 (14, 48)** |
| **val manquant** | **0** |
| **Milieu de résidence** | **1,255 (1,002, 1,548)** |
| **val manquant** | **0** |
| *1*n (%); Median (Q1, Q3) | |
| souce: Donné de EHCVM 2018-2019 Togo | |

stat\_base\_ind1

| statistiques descriptives sur la base individu | **N = 4,256***1* |
| --- | --- |
| La région |  |
| 1 | 752 (18%) |
| 2 | 2,476 (58%) |
| 3 | 1,028 (24%) |
| val manquant | 0 |
| la commune | 27 (14, 48) |
| val manquant | 0 |
| Milieu de résidence | 1,255 (1,002, 1,548) |
| val manquant | 0 |
| *1*n (%); Median (Q1, Q3) | |

head(base\_child\_baseline\_sans\_doublons)

## # A tibble: 6 × 17  
## regionid communeid villageid hhid round s1\_q0 s1\_q1 s1\_q2 energ\_kcal  
## <dbl> <dbl> <dbl> <chr> <dbl+l> <dbl+l> <dbl+l> <dbl+l> <dbl>  
## 1 2 25 1000 49484… 1 [Bas… 1 [Bre… 1 [Yes] 2 [chi… 355.  
## 2 2 25 1000 49484… 1 [Bas… 2 [Lun… 1 [Yes] 2 [chi… 224.  
## 3 2 25 1000 49484… 1 [Bas… 3 [Din… 1 [Yes] 2 [chi… 334.  
## 4 2 25 1000 49484… 1 [Bas… 4 [Sna… 1 [Yes] 2 [chi… 494.  
## 5 2 25 1000 49484… 1 [Bas… 1 [Bre… 1 [Yes] 2 [chi… 235.  
## 6 2 25 1000 49484… 1 [Bas… 2 [Lun… 1 [Yes] 2 [chi… 216.  
## # ℹ 8 more variables: protein\_g <dbl>, lipid\_tot\_g <dbl>, calcium\_mg <dbl>,  
## # iron\_mg <dbl>, zinc\_mg <dbl>, vit\_b6\_mg <dbl>, vit\_b12\_mcg <dbl>,  
## # vit\_c\_mg <dbl>

Base menage

#Statistiques descriptive   
stat\_base1 <- child\_baseline\_v1.dta %>% labelled::to\_factor() %>% select(regionid, communeid, villageid) %>% tbl\_summary(  
 label = list(  
 regionid ~ "La région",  
 communeid ~ "la commune",  
 villageid ~ "Milieu de résidence"  
 ),  
 missing = "always",  
 missing\_text="val manquant"  
) %>% modify\_header(label="statistiques descriptives sur la base individu") #%>% as\_gt() %>% gt:: tab\_source\_note(source\_note="souce: Donné de EHCVM 2018-2019 Togo")

#mise en forme du tableau avec la le package gt  
stat\_base1<- as\_gt(stat\_base1)   
stat\_base1 %>%   
 tab\_header(title="Statistique de la base menage") %>%  
 #Mettons la bordure en vert pour toutes les colonnes  
 tab\_style(style=cell\_borders(color="blue", weight = px(4)), locations = cells\_body(columns=everything())) %>%  
 #Mettons le fonf en jaune  
 tab\_style(style=cell\_fill(color="lightblue"), locations = cells\_column\_labels(columns=everything())) %>%   
 tab\_style(style=cell\_fill(color="lightblue"), locations = cells\_body(columns=everything())) %>%   
 #mettons le texte en marron et en gras  
 tab\_style(style=cell\_text(color="black", weight="bold" ), locations = cells\_body(columns=everything()))%>%  
   
 #tab\_source\_note(style=cell\_text(color="lightblue")) %>%   
 #Le titrage en bas   
 gt:: tab\_source\_note(source\_note="souce: Donné de EHCVM 2018-2019 Togo") #style=cell\_text(color="lightblue") )

Table 1: Statistique de la base menage

| statistiques descriptives sur la base individu | **N = 4,256***1* |
| --- | --- |
| **La région** |  |
| **1** | **752 (18%)** |
| **2** | **2,476 (58%)** |
| **3** | **1,028 (24%)** |
| **val manquant** | **0** |
| **la commune** | **27 (14, 48)** |
| **val manquant** | **0** |
| **Milieu de résidence** | **1,255 (1,002, 1,548)** |
| **val manquant** | **0** |
| *1*n (%); Median (Q1, Q3) | |
| souce: Donné de EHCVM 2018-2019 Togo | |

stat\_base\_ind1

| statistiques descriptives sur la base individu | **N = 4,256***1* |
| --- | --- |
| La région |  |
| 1 | 752 (18%) |
| 2 | 2,476 (58%) |
| 3 | 1,028 (24%) |
| val manquant | 0 |
| la commune | 27 (14, 48) |
| val manquant | 0 |
| Milieu de résidence | 1,255 (1,002, 1,548) |
| val manquant | 0 |
| *1*n (%); Median (Q1, Q3) | |

Ces tableau nous donnent quelques statistiques sur les variables choisits.