Devoir

BERETE Mamady I

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", "foreign")  
  
  
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 ()  
}

## Section I : Importation et analyse des bases

### Importation des bases

## Base ménage  
library(haven)  
base\_menage <- read\_dta("C:/Users/LENO/Desktop/ENSAE/R/Test\_BERETE\_Mamady\_I/Données/base\_menage.dta")  
View(base\_menage)  
  
## Base food\_comp\_child\_baseline  
base\_child\_baseline <- read\_dta("C:/Users/LENO/Desktop/ENSAE/R/Test\_BERETE\_Mamady\_I/Données/food\_comp\_child\_baseline.dta")  
View(base\_child\_baseline)  
  
## Base \_child\_endline  
base\_child\_endline <- read\_dta("C:/Users/LENO/Desktop/ENSAE/R/Test\_BERETE\_Mamady\_I/Données/food\_comp\_child\_endline.dta")  
View(base\_child\_endline)  
  
## Base food\_comp\_mother\_baseline  
base\_mother\_baseline <- read\_dta("C:/Users/LENO/Desktop/ENSAE/R/Test\_BERETE\_Mamady\_I/Données/food\_comp\_mother\_baseline.dta")  
View(base\_mother\_baseline)  
  
  
## Base food\_comp\_mother\_endline  
base\_mother\_endline <- read\_dta("C:/Users/LENO/Desktop/ENSAE/R/Test\_BERETE\_Mamady\_I/Données/food\_comp\_mother\_endline.dta")  
View(base\_mother\_endline)

### Structure des bases\*\*

print("Structure de la Base ménage")

## [1] "Structure de la Base ménage"

str(base\_menage)

## tibble [1,065 × 21] (S3: tbl\_df/tbl/data.frame)  
## $ regionid : num [1:1065] 2 2 2 2 2 2 2 2 2 2 ...  
## ..- attr(\*, "label")= chr "Region ID"  
## ..- attr(\*, "format.stata")= chr "%8.0g"  
## $ communeid : num [1:1065] 25 25 25 25 25 25 25 25 25 25 ...  
## ..- attr(\*, "label")= chr "Commune ID"  
## ..- attr(\*, "format.stata")= chr "%8.0g"  
## $ villageid : num [1:1065] 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 ...  
## ..- attr(\*, "label")= chr "Village ID"  
## ..- attr(\*, "format.stata")= chr "%8.0g"  
## $ hhid : chr [1:1065] "4948484848535052" "4948484848535053" "4948484848535055" "4948484848535056" ...  
## ..- attr(\*, "label")= chr "Household ID"  
## ..- attr(\*, "format.stata")= chr "%45s"  
## $ hhsize : num [1:1065] 4 8 11 9 16 6 31 8 23 5 ...  
## ..- attr(\*, "label")= chr "Household size"  
## ..- attr(\*, "format.stata")= chr "%10.0g"  
## $ poly : dbl+lbl [1:1065] 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 1, 1, 0,...  
## ..@ label : chr "Polygamous household?"  
## ..@ format.stata: chr "%8.0g"  
## ..@ labels : Named num [1:2] 0 1  
## .. ..- attr(\*, "names")= chr [1:2] "Non" "Oui"  
## $ hh\_primary : dbl+lbl [1:1065] 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...  
## ..@ label : chr "Household head completed primary education"  
## ..@ format.stata: chr "%9.0g"  
## ..@ labels : Named num [1:2] 0 1  
## .. ..- attr(\*, "names")= chr [1:2] "Non" "Oui"  
## $ s1\_q2 : dbl+lbl [1:1065] 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,...  
## ..@ label : chr "Male head of household"  
## ..@ format.stata: chr "%8.0g"  
## ..@ labels : Named num [1:2] 0 1  
## .. ..- attr(\*, "names")= chr [1:2] "Female" "Male"  
## $ s1\_q4a : num [1:1065] 37 52 67 38 75 41 56 52 63 26 ...  
## ..- attr(\*, "label")= chr "Age head of household"  
## ..- attr(\*, "format.stata")= chr "%8.0g"  
## $ s2\_q1 : dbl+lbl [1:1065] 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0,...  
## ..@ label : chr "Is head of household literate in local language"  
## ..@ format.stata: chr "%8.0g"  
## ..@ labels : Named num [1:2] 0 1  
## .. ..- attr(\*, "names")= chr [1:2] "Non" "Oui"  
## $ s2\_q2 : dbl+lbl [1:1065] 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...  
## ..@ label : chr "Is head of household literate in French?"  
## ..@ format.stata: chr "%8.0g"  
## ..@ labels : Named num [1:2] 0 1  
## .. ..- attr(\*, "names")= chr [1:2] "Non" "Oui"  
## $ s2\_q4 : dbl+lbl [1:1065] 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0,...  
## ..@ label : chr "Has head of household been to school (formal or informal)?"  
## ..@ format.stata: chr "%8.0g"  
## ..@ labels : Named num [1:2] 0 1  
## .. ..- attr(\*, "names")= chr [1:2] "Non" "Oui"  
## $ s29\_q1 : dbl+lbl [1:1065] 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0,...  
## ..@ label : chr "Est-ce qu'un membre de votre ménage a pris un prêt ou fait un emprunt en argent"  
## ..@ format.stata: chr "%8.0g"  
## ..@ labels : Named num [1:2] 0 1  
## .. ..- attr(\*, "names")= chr [1:2] "Non" "Oui"  
## $ demgrp1 : num [1:1065] 1 0 0 0 1 0 5 1 1 1 ...  
## ..- attr(\*, "label")= chr "Number of children 0-36 months"  
## ..- attr(\*, "format.stata")= chr "%9.0g"  
## $ demgrp2 : num [1:1065] 1 2 2 1 1 1 5 0 2 1 ...  
## ..- attr(\*, "label")= chr "Number of children 36-72 months"  
## ..- attr(\*, "format.stata")= chr "%9.0g"  
## $ demgrp3 : num [1:1065] 0 3 3 3 7 2 11 4 9 1 ...  
## ..- attr(\*, "label")= chr "Number of adults 6-14 years"  
## ..- attr(\*, "format.stata")= chr "%9.0g"  
## $ demgrp4 : num [1:1065] 1 2 5 5 6 3 10 3 11 2 ...  
## ..- attr(\*, "label")= chr "Number of adults 14-65 years"  
## ..- attr(\*, "format.stata")= chr "%9.0g"  
## $ demgrp5 : num [1:1065] 0 1 1 0 1 0 0 0 0 0 ...  
## ..- attr(\*, "label")= chr "Number of elders 65+ years"  
## ..- attr(\*, "format.stata")= chr "%9.0g"  
## $ dependencyratio: num [1:1065] 2 3 1.2 0.8 1.67 ...  
## ..- attr(\*, "label")= chr "Dependency ratio"  
## ..- attr(\*, "format.stata")= chr "%9.0g"  
## $ hfias\_score : num [1:1065] 18 21 0 4 1 3 12 0 14 0 ...  
## ..- attr(\*, "label")= chr "HFIAS Score (0-27)"  
## ..- attr(\*, "format.stata")= chr "%9.0g"  
## $ T1 : dbl+lbl [1:1065] 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1,...  
## ..@ label : chr "Treatment: 1st level comparison"  
## ..@ format.stata: chr "%12.0g"  
## ..@ labels : Named num [1:2] 0 1  
## .. ..- attr(\*, "names")= chr [1:2] "Controle" "Intervention"  
## - attr(\*, "label")= chr "One row per household"

print("Structure de la Base food\_comp\_child\_baseline")

## [1] "Structure de la Base food\_comp\_child\_baseline"

str(base\_child\_baseline)

## tibble [4,256 × 17] (S3: tbl\_df/tbl/data.frame)  
## $ regionid : num [1:4256] 2 2 2 2 2 2 2 2 2 2 ...  
## ..- attr(\*, "label")= chr "Region ID"  
## ..- attr(\*, "format.stata")= chr "%8.0g"  
## $ communeid : num [1:4256] 25 25 25 25 25 25 25 25 25 25 ...  
## ..- attr(\*, "label")= chr "Commune ID"  
## ..- attr(\*, "format.stata")= chr "%8.0g"  
## $ villageid : num [1:4256] 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 ...  
## ..- attr(\*, "label")= chr "Village ID"  
## ..- attr(\*, "format.stata")= chr "%8.0g"  
## $ hhid : chr [1:4256] "4948484848535052" "4948484848535052" "4948484848535052" "4948484848535052" ...  
## ..- attr(\*, "label")= chr "Household ID"  
## ..- attr(\*, "format.stata")= chr "%45s"  
## $ round : dbl+lbl [1:4256] 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,...  
## ..@ label : chr "Survey round : Baseline, Endline"  
## ..@ format.stata: chr "%10.0g"  
## ..@ labels : Named num [1:2] 1 2  
## .. ..- attr(\*, "names")= chr [1:2] "Baseline" "Endline"  
## $ s1\_q0 : dbl+lbl [1:4256] 1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4, 1, 2,...  
## ..@ label : chr "eating occasion"  
## ..@ format.stata: chr "%27.0g"  
## ..@ labels : Named num [1:4] 1 2 3 4  
## .. ..- attr(\*, "names")= chr [1:4] "Breakfast" "Lunch" "Dinner" "Snacks"  
## $ s1\_q1 : dbl+lbl [1:4256] 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,...  
## ..@ label : chr "Meal consumed? Y/N"  
## ..@ format.stata: chr "%9.0g"  
## ..@ labels : Named num [1:2] 0 1  
## .. ..- attr(\*, "names")= chr [1:2] "No" "Yes"  
## $ s1\_q2 : dbl+lbl [1:4256] 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,...  
## ..@ label : chr "Who consummed?"  
## ..@ format.stata: chr "%16.0g"  
## ..@ labels : Named num [1:3] 1 2 3  
## .. ..- attr(\*, "names")= chr [1:3] "mother" "child" "mother and child"  
## $ energ\_kcal : num [1:4256] 355 224 334 494 235 ...  
## ..- attr(\*, "label")= chr "Consommation en ernergie (kcal)"  
## ..- attr(\*, "format.stata")= chr "%10.0g"  
## $ protein\_g : num [1:4256] 12.74 8.01 12.72 16.65 4.45 ...  
## ..- attr(\*, "label")= chr "Consommation en proteine (g)"  
## ..- attr(\*, "format.stata")= chr "%10.0g"  
## $ lipid\_tot\_g: num [1:4256] 3.14 1.96 3.42 8.84 4.76 ...  
## ..- attr(\*, "label")= chr "Consommation en lipide (g)"  
## ..- attr(\*, "format.stata")= chr "%10.0g"  
## $ calcium\_mg : num [1:4256] 40.1 24.8 47 60.1 93.3 ...  
## ..- attr(\*, "label")= chr "Consommation en calcium (mg)"  
## ..- attr(\*, "format.stata")= chr "%10.0g"  
## $ iron\_mg : num [1:4256] 5.39 3.42 4.9 6.83 1.54 ...  
## ..- attr(\*, "label")= chr "Consommation en fer (mg)"  
## ..- attr(\*, "format.stata")= chr "%10.0g"  
## $ zinc\_mg : num [1:4256] 2.41 1.52 2.35 3.13 1.63 ...  
## ..- attr(\*, "label")= chr "Consommation en zinc (mg)"  
## ..- attr(\*, "format.stata")= chr "%10.0g"  
## $ vit\_b6\_mg : num [1:4256] 0.167 0.105 0.168 0.207 0.116 ...  
## ..- attr(\*, "label")= chr "Consommation en vitamine B6 (mg)"  
## ..- attr(\*, "format.stata")= chr "%10.0g"  
## $ vit\_b12\_mcg: num [1:4256] 0.00616 0.00377 0.00785 0.00959 0.00688 ...  
## ..- attr(\*, "label")= chr "Consommation en vitamine B12 (mcg)"  
## ..- attr(\*, "format.stata")= chr "%10.0g"  
## $ vit\_c\_mg : num [1:4256] 0.031282 0.019173 0.03986 0.048689 0.000167 ...  
## ..- attr(\*, "label")= chr "Consommation en vitamine C (mcg)"  
## ..- attr(\*, "format.stata")= chr "%10.0g"

print("Structure de la Base child\_endline")

## [1] "Structure de la Base child\_endline"

str(base\_child\_endline)

## tibble [4,256 × 17] (S3: tbl\_df/tbl/data.frame)  
## $ regionid : num [1:4256] 2 2 2 2 2 2 2 2 2 2 ...  
## ..- attr(\*, "label")= chr "Region ID"  
## ..- attr(\*, "format.stata")= chr "%8.0g"  
## $ communeid : num [1:4256] 25 25 25 25 25 25 25 25 25 25 ...  
## ..- attr(\*, "label")= chr "Commune ID"  
## ..- attr(\*, "format.stata")= chr "%8.0g"  
## $ villageid : num [1:4256] 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 ...  
## ..- attr(\*, "label")= chr "Village ID"  
## ..- attr(\*, "format.stata")= chr "%8.0g"  
## $ hhid : chr [1:4256] "4948484848535052" "4948484848535052" "4948484848535052" "4948484848535052" ...  
## ..- attr(\*, "label")= chr "Household ID"  
## ..- attr(\*, "format.stata")= chr "%45s"  
## $ round : dbl+lbl [1:4256] 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,...  
## ..@ label : chr "Survey round : Baseline, Endline"  
## ..@ format.stata: chr "%10.0g"  
## ..@ labels : Named num [1:2] 1 2  
## .. ..- attr(\*, "names")= chr [1:2] "Baseline" "Endline"  
## $ s1\_q0 : dbl+lbl [1:4256] 1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4, 1, 2,...  
## ..@ label : chr "eating occasion"  
## ..@ format.stata: chr "%27.0g"  
## ..@ labels : Named num [1:4] 1 2 3 4  
## .. ..- attr(\*, "names")= chr [1:4] "Breakfast" "Lunch" "Dinner" "Snacks"  
## $ s1\_q1 : dbl+lbl [1:4256] 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,...  
## ..@ label : chr "Meal consumed? Y/N"  
## ..@ format.stata: chr "%9.0g"  
## ..@ labels : Named num [1:2] 0 1  
## .. ..- attr(\*, "names")= chr [1:2] "No" "Yes"  
## $ s1\_q2 : dbl+lbl [1:4256] 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,...  
## ..@ label : chr "Who consummed?"  
## ..@ format.stata: chr "%16.0g"  
## ..@ labels : Named num [1:3] 1 2 3  
## .. ..- attr(\*, "names")= chr [1:3] "mother" "child" "mother and child"  
## $ energ\_kcal : num [1:4256] 1193 535 1701 531 188 ...  
## ..- attr(\*, "label")= chr "Consommation en ernergie (kcal)"  
## ..- attr(\*, "format.stata")= chr "%10.0g"  
## $ protein\_g : num [1:4256] 24.94 18.38 58.84 13.89 5.04 ...  
## ..- attr(\*, "label")= chr "Consommation en proteine (g)"  
## ..- attr(\*, "format.stata")= chr "%10.0g"  
## $ lipid\_tot\_g: num [1:4256] 51.915 3.205 90.268 3.917 0.496 ...  
## ..- attr(\*, "label")= chr "Consommation en lipide (g)"  
## ..- attr(\*, "format.stata")= chr "%10.0g"  
## $ calcium\_mg : num [1:4256] 479.73 41.32 1122.63 74.44 3.29 ...  
## ..- attr(\*, "label")= chr "Consommation en calcium (mg)"  
## ..- attr(\*, "format.stata")= chr "%10.0g"  
## $ iron\_mg : num [1:4256] 11.5857 8.709 7.2277 6.4328 0.0321 ...  
## ..- attr(\*, "label")= chr "Consommation en fer (mg)"  
## ..- attr(\*, "format.stata")= chr "%10.0g"  
## $ zinc\_mg : num [1:4256] 4.9181 3.3562 1.4184 2.7147 0.0102 ...  
## ..- attr(\*, "label")= chr "Consommation en zinc (mg)"  
## ..- attr(\*, "format.stata")= chr "%10.0g"  
## $ vit\_b6\_mg : num [1:4256] 0.4678 0.2251 0.2028 0.3523 0.0018 ...  
## ..- attr(\*, "label")= chr "Consommation en vitamine B6 (mg)"  
## ..- attr(\*, "format.stata")= chr "%10.0g"  
## $ vit\_b12\_mcg: num [1:4256] 0.00396 0.01364 0 0.00736 0.00309 ...  
## ..- attr(\*, "label")= chr "Consommation en vitamine B12 (mcg)"  
## ..- attr(\*, "format.stata")= chr "%10.0g"  
## $ vit\_c\_mg : num [1:4256] 6.8299 0.1201 16.904 0.0764 0 ...  
## ..- attr(\*, "label")= chr "Consommation en vitamine C (mcg)"  
## ..- attr(\*, "format.stata")= chr "%10.0g"

print("Structure de la Base food\_comp\_mother\_baseline")

## [1] "Structure de la Base food\_comp\_mother\_baseline"

str(base\_mother\_baseline)

## tibble [4,256 × 17] (S3: tbl\_df/tbl/data.frame)  
## $ regionid : num [1:4256] 2 2 2 2 2 2 2 2 2 2 ...  
## ..- attr(\*, "label")= chr "Region ID"  
## ..- attr(\*, "format.stata")= chr "%8.0g"  
## $ communeid : num [1:4256] 25 25 25 25 25 25 25 25 25 25 ...  
## ..- attr(\*, "label")= chr "Commune ID"  
## ..- attr(\*, "format.stata")= chr "%8.0g"  
## $ villageid : num [1:4256] 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 ...  
## ..- attr(\*, "label")= chr "Village ID"  
## ..- attr(\*, "format.stata")= chr "%8.0g"  
## $ hhid : chr [1:4256] "4948484848535052" "4948484848535052" "4948484848535052" "4948484848535052" ...  
## ..- attr(\*, "label")= chr "Household ID"  
## ..- attr(\*, "format.stata")= chr "%45s"  
## $ round : dbl+lbl [1:4256] 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,...  
## ..@ label : chr "Survey round : Baseline, Endline"  
## ..@ format.stata: chr "%10.0g"  
## ..@ labels : Named num [1:2] 1 2  
## .. ..- attr(\*, "names")= chr [1:2] "Baseline" "Endline"  
## $ s1\_q0 : dbl+lbl [1:4256] 1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4, 1, 2,...  
## ..@ label : chr "eating occasion"  
## ..@ format.stata: chr "%27.0g"  
## ..@ labels : Named num [1:4] 1 2 3 4  
## .. ..- attr(\*, "names")= chr [1:4] "Breakfast" "Lunch" "Dinner" "Snacks"  
## $ s1\_q1 : dbl+lbl [1:4256] 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1,...  
## ..@ label : chr "Meal consumed? Y/N"  
## ..@ format.stata: chr "%9.0g"  
## ..@ labels : Named num [1:2] 0 1  
## .. ..- attr(\*, "names")= chr [1:2] "No" "Yes"  
## $ s1\_q2 : dbl+lbl [1:4256] 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,...  
## ..@ label : chr "Who consummed?"  
## ..@ format.stata: chr "%16.0g"  
## ..@ labels : Named num [1:3] 1 2 3  
## .. ..- attr(\*, "names")= chr [1:3] "mother" "child" "mother and child"  
## $ V1 : num [1:4256] 680 NA 634 256 563 ...  
## ..- attr(\*, "label")= chr "Consommation en ernergie (kcal)"  
## ..- attr(\*, "format.stata")= chr "%10.0g"  
## $ protein\_g : num [1:4256] 23.31 NA 21.62 8.25 11.7 ...  
## ..- attr(\*, "label")= chr "Consommation en proteine (g)"  
## ..- attr(\*, "format.stata")= chr "%10.0g"  
## $ lipid\_tot\_g: num [1:4256] 5.3 NA 4.87 8.06 6.49 ...  
## ..- attr(\*, "label")= chr "Consommation en lipide (g)"  
## ..- attr(\*, "format.stata")= chr "%10.0g"  
## $ calcium\_mg : num [1:4256] 62.7 NA 57.1 22 116.3 ...  
## ..- attr(\*, "label")= chr "Consommation en calcium (mg)"  
## ..- attr(\*, "format.stata")= chr "%10.0g"  
## $ iron\_mg : num [1:4256] 10.591 NA 9.897 0.912 2.716 ...  
## ..- attr(\*, "label")= chr "Consommation en fer (mg)"  
## ..- attr(\*, "format.stata")= chr "%10.0g"  
## $ V9 : num [1:4256] 4.507 NA 4.19 0.456 3.382 ...  
## ..- attr(\*, "label")= chr "Consommation en zinc (mg)"  
## ..- attr(\*, "format.stata")= chr "%10.0g"  
## $ vit\_b6\_mg : num [1:4256] 0.3058 NA 0.2835 0.0456 0.2248 ...  
## ..- attr(\*, "label")= chr "Consommation en vitamine B6 (mg)"  
## ..- attr(\*, "format.stata")= chr "%10.0g"  
## $ vit\_b12\_mcg: num [1:4256] 0.00869 NA 0.0078 0 0.00823 ...  
## ..- attr(\*, "label")= chr "Consommation en vitamine B12 (mcg)"  
## ..- attr(\*, "format.stata")= chr "%10.0g"  
## $ vit\_c\_mg : num [1:4256] 0.0441 NA 0.0396 0 0.0002 ...  
## ..- attr(\*, "label")= chr "Consommation en vitamine C (mcg)"  
## ..- attr(\*, "format.stata")= chr "%10.0g"

print("Structure de la Base food\_comp\_mother\_endline")

## [1] "Structure de la Base food\_comp\_mother\_endline"

str(base\_mother\_endline)

## tibble [4,256 × 17] (S3: tbl\_df/tbl/data.frame)  
## $ regionid : num [1:4256] 2 2 2 2 2 2 2 2 2 2 ...  
## ..- attr(\*, "label")= chr "Region ID"  
## ..- attr(\*, "format.stata")= chr "%8.0g"  
## $ communeid : num [1:4256] 25 25 25 25 25 25 25 25 25 25 ...  
## ..- attr(\*, "label")= chr "Commune ID"  
## ..- attr(\*, "format.stata")= chr "%8.0g"  
## $ villageid : num [1:4256] 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 ...  
## ..- attr(\*, "label")= chr "Village ID"  
## ..- attr(\*, "format.stata")= chr "%8.0g"  
## $ hhid : chr [1:4256] "4948484848535052" "4948484848535052" "4948484848535052" "4948484848535052" ...  
## ..- attr(\*, "label")= chr "Household ID"  
## ..- attr(\*, "format.stata")= chr "%45s"  
## $ round : dbl+lbl [1:4256] 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,...  
## ..@ label : chr "Survey round : Baseline, Endline"  
## ..@ format.stata: chr "%10.0g"  
## ..@ labels : Named num [1:2] 1 2  
## .. ..- attr(\*, "names")= chr [1:2] "Baseline" "Endline"  
## $ s1\_q0 : dbl+lbl [1:4256] 1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4, 1, 2,...  
## ..@ label : chr "eating occasion"  
## ..@ format.stata: chr "%27.0g"  
## ..@ labels : Named num [1:4] 1 2 3 4  
## .. ..- attr(\*, "names")= chr [1:4] "Breakfast" "Lunch" "Dinner" "Snacks"  
## $ s1\_q1 : dbl+lbl [1:4256] 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1,...  
## ..@ label : chr "Meal consumed? Y/N"  
## ..@ format.stata: chr "%9.0g"  
## ..@ labels : Named num [1:2] 0 1  
## .. ..- attr(\*, "names")= chr [1:2] "No" "Yes"  
## $ s1\_q2 : dbl+lbl [1:4256] 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,...  
## ..@ label : chr "Who consummed?"  
## ..@ format.stata: chr "%16.0g"  
## ..@ labels : Named num [1:3] 1 2 3  
## .. ..- attr(\*, "names")= chr [1:3] "mother" "child" "mother and child"  
## $ energ\_kcal : num [1:4256] 2217 1021 3038 716 618 ...  
## ..- attr(\*, "label")= chr "Consommation en ernergie (kcal)"  
## ..- attr(\*, "format.stata")= chr "%10.0g"  
## $ protein\_g : num [1:4256] 48.7 34.8 92 19.1 16.5 ...  
## ..- attr(\*, "label")= chr "Consommation en proteine (g)"  
## ..- attr(\*, "format.stata")= chr "%10.0g"  
## $ lipid\_tot\_g: num [1:4256] 89.6 5.94 183.6 4.73 1.63 ...  
## ..- attr(\*, "label")= chr "Consommation en lipide (g)"  
## ..- attr(\*, "format.stata")= chr "%10.0g"  
## $ calcium\_mg : num [1:4256] 818.8 72.1 1781.3 131.6 10.8 ...  
## ..- attr(\*, "label")= chr "Consommation en calcium (mg)"  
## ..- attr(\*, "format.stata")= chr "%10.0g"  
## $ iron\_mg : num [1:4256] 22.659 16.558 11.345 8.877 0.105 ...  
## ..- attr(\*, "label")= chr "Consommation en fer (mg)"  
## ..- attr(\*, "format.stata")= chr "%10.0g"  
## $ zinc\_mg : num [1:4256] 9.5691 6.4618 2.2278 3.5867 0.0334 ...  
## ..- attr(\*, "label")= chr "Consommation en zinc (mg)"  
## ..- attr(\*, "format.stata")= chr "%10.0g"  
## $ vit\_b6\_mg : num [1:4256] 0.8748 0.4327 0.314 0.5277 0.0059 ...  
## ..- attr(\*, "label")= chr "Consommation en vitamine B6 (mg)"  
## ..- attr(\*, "format.stata")= chr "%10.0g"  
## $ vit\_b12\_mcg: num [1:4256] 0.00948 0.02232 0 0.01281 0.01016 ...  
## ..- attr(\*, "label")= chr "Consommation en vitamine B12 (mcg)"  
## ..- attr(\*, "format.stata")= chr "%10.0g"  
## $ vit\_c\_mg : num [1:4256] 11.547 0.183 26.738 0.133 0 ...  
## ..- attr(\*, "label")= chr "Consommation en vitamine C (mcg)"  
## ..- attr(\*, "format.stata")= chr "%10.0g"

### Nom des colonnes dans chaque base

print("Structure de la Base ménage")

## [1] "Structure de la Base ménage"

colnames(base\_menage)

## [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"

print("Structure de la Base food\_comp\_child\_baseline")

## [1] "Structure de la Base food\_comp\_child\_baseline"

colnames(base\_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"

print("Structure de la Base child\_endline")

## [1] "Structure de la Base child\_endline"

colnames(base\_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"

print("Structure de la Base food\_comp\_mother\_baseline")

## [1] "Structure de la Base food\_comp\_mother\_baseline"

colnames(base\_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"

print("Structure de la Base food\_comp\_mother\_endline")

## [1] "Structure de la Base food\_comp\_mother\_endline"

colnames(base\_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"

# Partie 1 : Gestion et nettoyage des bases de données

## 1. Vérification de la présence de doublons dans les bases de données Baseline, Endline et ménage. Supprimez les doublons si nécessaire.

#Base ménage  
  
#Vérification des doublons  
doublons\_menage <- base\_menage[duplicated(base\_menage), ]  
  
# Suppression des doublons selon une clé spécifique  
base\_menage\_unique <- base\_menage %>%   
 distinct(hhid, .keep\_all = TRUE)  
  
# Vérification du résultat  
cat("Nombre de lignes avant suppression :", nrow(base\_menage), "\n")

## Nombre de lignes avant suppression : 1065

cat("Nombre de lignes après suppression des doublons :", nrow(base\_menage\_unique), "\n")

## Nombre de lignes après suppression des doublons : 1064

# Mettre à jour ta base initiale  
base\_menage <- base\_menage\_unique  
  
  
#Base Base child\_baseline  
  
#Vérification des doublons  
  
doublons\_base\_child\_baseline <- base\_child\_baseline[duplicated(base\_child\_baseline), ]  
  
# Suppression des doublons selon une clé spécifique  
base\_child\_baseline\_unique <- base\_child\_baseline %>%   
 distinct(hhid, .keep\_all = TRUE)  
  
# Vérification du résultat  
cat("Nombre de lignes avant suppression :", nrow(base\_child\_baseline), "\n")

## Nombre de lignes avant suppression : 4256

cat("Nombre de lignes après suppression des doublons :", nrow(base\_child\_baseline\_unique), "\n")

## Nombre de lignes après suppression des doublons : 1064

# Mettre à jour ta base initiale  
base\_child\_baseline <- base\_child\_baseline\_unique  
  
  
#Base Base child\_endline   
  
#Vérification des doublons  
  
doublons\_base\_child\_endline <- base\_child\_endline[duplicated(base\_child\_endline), ]  
  
# Suppression des doublons selon une clé spécifique  
base\_child\_endline\_unique <- base\_child\_endline %>%   
 distinct(hhid, .keep\_all = TRUE)  
  
# Vérification du résultat  
cat("Nombre de lignes avant suppression :", nrow(base\_child\_endline), "\n")

## Nombre de lignes avant suppression : 4256

cat("Nombre de lignes après suppression des doublons :", nrow(base\_child\_endline\_unique), "\n")

## Nombre de lignes après suppression des doublons : 1064

# Mettre à jour ta base initiale  
base\_child\_endline <- base\_child\_endline\_unique  
  
  
#Base Base food\_comp\_mother\_baseline  
  
#Vérification des doublons  
  
doublons\_base\_mother\_baseline <- base\_mother\_baseline[duplicated(base\_mother\_baseline), ]  
  
# Suppression des doublons selon une clé spécifique  
base\_mother\_baseline\_unique <- base\_mother\_baseline %>%   
 distinct(hhid, .keep\_all = TRUE)  
  
# Vérification du résultat  
cat("Nombre de lignes avant suppression :", nrow(base\_mother\_baseline), "\n")

## Nombre de lignes avant suppression : 4256

cat("Nombre de lignes après suppression des doublons :", nrow(base\_mother\_baseline\_unique), "\n")

## Nombre de lignes après suppression des doublons : 1064

# Mettre à jour ta base initiale  
base\_mother\_baseline <- base\_mother\_baseline\_unique  
  
  
#Base Base food\_comp\_mother\_endline   
  
#Vérification des doublons  
  
doublons\_base\_mother\_endline <- base\_mother\_endline[duplicated(base\_mother\_endline), ]  
  
# Suppression des doublons selon une clé spécifique  
base\_mother\_endline\_unique <- base\_mother\_endline %>%   
 distinct(hhid, .keep\_all = TRUE)  
  
# Vérification du résultat  
cat("Nombre de lignes avant suppression :", nrow(base\_mother\_endline), "\n")

## Nombre de lignes avant suppression : 4256

cat("Nombre de lignes après suppression des doublons :", nrow(base\_mother\_endline\_unique), "\n")

## Nombre de lignes après suppression des doublons : 1064

# Mettre à jour ta base initiale  
base\_mother\_endline <- base\_mother\_endline\_unique

## La consommation d’énergie moyenne à chaque repas pour l’ensemble des mères lors de l’enquête de base

## Consommation d'énergie moyenne des mères (baseline)  
# Assumant que la variable énergie s'appelle "energy\_baseline\_mother"  
  
V1 <- base\_mother\_baseline %>%  
 summarise(mean\_energy = mean(V1, na.rm = TRUE))

# La vérification et correction de chaque base de données

# Sauvegarde mother\_baseline et son exportation  
write\_dta(base\_mother\_baseline, "mother\_baseline\_v1.dta")  
  
write\_dta(base\_mother\_baseline, "C:/Users/LENO/Desktop/ENSAE/R/Test\_BERETE\_Mamady\_I/Données/mother\_baseline\_v1.dta")  
  
  
# Sauvegarde mother\_endline et son exportation  
write\_dta(base\_mother\_endline, "mother\_endline\_v1.dta")  
  
write\_dta(base\_mother\_endline, "C:/Users/LENO/Desktop/ENSAE/R/Test\_BERETE\_Mamady\_I/Données/mother\_endline\_v1.dta")  
  
# Sauvegarde child\_baseline  
write\_dta(base\_child\_baseline, "child\_baseline\_v1.dta")  
  
write\_dta(base\_child\_baseline, "C:/Users/LENO/Desktop/ENSAE/R/Test\_BERETE\_Mamady\_I/Données/child\_baseline\_v1.dta")  
  
  
# Sauvegarde child\_endline  
write\_dta(base\_child\_endline, "child\_endline\_v1.dta")  
  
write\_dta(base\_child\_endline, "C:/Users/LENO/Desktop/ENSAE/R/Test\_BERETE\_Mamady\_I/Données/child\_endline\_v1.dta")  
  
  
# Sauvegarde base\_menage  
write\_dta(base\_menage, "base\_menage\_final.dta")  
  
write\_dta(base\_menage, "C:/Users/LENO/Desktop/ENSAE/R/Test\_BERETE\_Mamady\_I/Données/base\_menage\_final.dta")

# Partie 2 : Empilement et Fusion des données

## Section I : Importation et analyse des bases

### Importation des nouvelles bases

## Base ménage  
  
base\_menage\_final <- read\_dta("C:/Users/LENO/Desktop/ENSAE/R/Test\_BERETE\_Mamady\_I/Données/base\_menage\_final.dta")  
  
  
## Base food\_comp\_child\_baseline  
child\_baseline\_v1 <- read\_dta("C:/Users/LENO/Desktop/ENSAE/R/Test\_BERETE\_Mamady\_I/Données/child\_baseline\_v1.dta")  
  
  
## Base \_child\_endline  
child\_endline\_v1 <- read\_dta("C:/Users/LENO/Desktop/ENSAE/R/Test\_BERETE\_Mamady\_I/Données/child\_endline\_v1.dta")  
  
  
## Base food\_comp\_mother\_baseline  
mother\_baseline\_v1 <- read\_dta("C:/Users/LENO/Desktop/ENSAE/R/Test\_BERETE\_Mamady\_I/Données/mother\_baseline\_v1.dta")  
  
  
  
## Base food\_comp\_mother\_endline  
mother\_endline\_v1.dta <- read\_dta("C:/Users/LENO/Desktop/ENSAE/R/Test\_BERETE\_Mamady\_I/Données/mother\_endline\_v1.dta")

# Baseline

### Empilez les bases de données des mères (mother\_baseline\_v1.dta) et des enfants (child\_baseline\_v1.dta) de sorte que pour chaque ménage de l’étude, on retrouve une ou plusieurs lignes concernant la mère et sa consommation journalière et une ou plusieurs lignes concernant la consommation de l’enfant.

## La fusion des trois bases  
  
base\_merge <- dplyr::left\_join(base\_menage\_final ,mother\_baseline\_v1,child\_baseline\_v1,by="hhid")

# Renommer les variables de consommation avec le suffixe \_b  
  
## Liste des anciennes variables  
variables <- c("V1", "protein\_g", "lipid\_tot\_g", "calcium\_mg", "iron\_mg", "V9", "vit\_b6\_mg", "vit\_b12\_mcg", "vit\_c\_mg")  
  
## Vérifier si ces variables existent dans base\_merge  
variables\_existantes <- variables[variables %in% names(base\_merge)]  
  
## Renommer les colonnes existantes en ajoutant "\_e"  
names(base\_merge)[names(base\_merge) %in% variables\_existantes] <- paste0(variables\_existantes, "\_e")  
  
## Vérifier les nouvelles colonnes  
print(names(base\_merge))

## [1] "regionid.x" "communeid.x" "villageid.x" "hhid"   
## [5] "hhsize" "poly" "hh\_primary" "s1\_q2.x"   
## [9] "s1\_q4a" "s2\_q1" "s2\_q2" "s2\_q4"   
## [13] "s29\_q1" "demgrp1" "demgrp2" "demgrp3"   
## [17] "demgrp4" "demgrp5" "dependencyratio" "hfias\_score"   
## [21] "T1" "regionid.y" "communeid.y" "villageid.y"   
## [25] "round" "s1\_q0" "s1\_q1" "s1\_q2.y"   
## [29] "V1\_e" "protein\_g\_e" "lipid\_tot\_g\_e" "calcium\_mg\_e"   
## [33] "iron\_mg\_e" "V9\_e" "vit\_b6\_mg\_e" "vit\_b12\_mcg\_e"   
## [37] "vit\_c\_mg\_e"

# Agréger les consommations journalières totales par individu  
print(names(base\_merge))

## [1] "regionid.x" "communeid.x" "villageid.x" "hhid"   
## [5] "hhsize" "poly" "hh\_primary" "s1\_q2.x"   
## [9] "s1\_q4a" "s2\_q1" "s2\_q2" "s2\_q4"   
## [13] "s29\_q1" "demgrp1" "demgrp2" "demgrp3"   
## [17] "demgrp4" "demgrp5" "dependencyratio" "hfias\_score"   
## [21] "T1" "regionid.y" "communeid.y" "villageid.y"   
## [25] "round" "s1\_q0" "s1\_q1" "s1\_q2.y"   
## [29] "V1\_e" "protein\_g\_e" "lipid\_tot\_g\_e" "calcium\_mg\_e"   
## [33] "iron\_mg\_e" "V9\_e" "vit\_b6\_mg\_e" "vit\_b12\_mcg\_e"   
## [37] "vit\_c\_mg\_e"

if (!"hhid" %in% names(base\_merge)) {  
 stop("La variable 'hhid' n'existe pas dans base\_merge")  
}  
base\_merge <- base\_merge %>%  
 mutate(across(c(V1\_e, protein\_g\_e,lipid\_tot\_g\_e, calcium\_mg\_e, iron\_mg\_e, V9\_e, vit\_b6\_mg\_e, vit\_b12\_mcg\_e, vit\_c\_mg\_e), as.numeric, .names = "clean\_{.col}"))  
  
 #Effectuer l'agrégation  
base\_journaliere <- base\_merge %>%  
 group\_by(hhid) %>%  
 summarise(across(c(V1\_e, protein\_g\_e,lipid\_tot\_g\_e, calcium\_mg\_e, iron\_mg\_e, V9\_e, vit\_b6\_mg\_e, vit\_b12\_mcg\_e, vit\_c\_mg\_e), sum, na.rm = TRUE))

## 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))

# Vérifier la base résultante  
print(head(base\_journaliere))

## # A tibble: 6 × 10  
## hhid V1\_e protein\_g\_e lipid\_tot\_g\_e calcium\_mg\_e iron\_mg\_e V9\_e vit\_b6\_mg\_e  
## <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 4948… 680. 23.3 5.30 62.7 10.6 4.51 0.306   
## 2 4948… 563. 11.7 6.49 116. 2.72 3.38 0.225   
## 3 4948… 237. 8.67 1.39 11.9 3.78 1.38 0.0891  
## 4 4948… 873. 9.60 4.33 26.6 12.6 2.94 0.103   
## 5 4948… 408. 20.8 1.30 60.6 3.53 2.95 0.181   
## 6 4948… 640. 20.5 2.71 20.6 2.15 3.52 0.234   
## # ℹ 2 more variables: vit\_b12\_mcg\_e <dbl>, vit\_c\_mg\_e <dbl>

# Sélectionner les variables d'intérêt  
  
baseline\_final <- base\_merge %>%  
 select(hhid, s1\_q2.x, V1\_e, protein\_g\_e, lipid\_tot\_g\_e, calcium\_mg\_e, iron\_mg\_e, V9\_e)  
   
# Vérification : Chaque ménage doit avoir une ligne pour la mère et une pour l'enfant  
  
table(baseline\_final$hhid) # Vérifier la répartition par hhid

##   
## 4948484848535052 4948484848535053 4948484848535055 4948484848535056   
## 1 1 1 1   
## 4948484848535151 4948484848535152 4948484848535154 4948484848535155   
## 1 1 1 1   
## 4948484848535253 4948484848535255 4948484848535256 4948484848535257   
## 1 1 1 1   
## 4948485048535349 4948485048535353 4948485048535448 4948485048535454   
## 1 1 1 1   
## 4948485048535455 4948485048535456 4948485048535550 4948485048535551   
## 1 1 1 1   
## 4948485048535552 4948485048535553 4948485048535554 4948485048535556   
## 1 1 1 1   
## 4948485648535649 4948485648535650 4948485648535655 4948485648535749   
## 1 1 1 1   
## 4948485648535750 4948485648535754 4948485648535755 4948485648535756   
## 1 1 1 1   
## 4948485648535757 4948485648544849 4948485648544854 4948485648544948   
## 1 1 1 1   
## 4948505748545551 4948505748545554 4948505748545555 4948505748545648   
## 1 1 1 1   
## 4948505748545652 4948505748545653 4948505748545655 4948505748545656   
## 1 1 1 1   
## 4948505748545657 4948505748545749 4948505748545751 4948505748545756   
## 1 1 1 1   
## 4948515548554849 4948515548554850 4948515548554851 4948515548554854   
## 1 1 1 1   
## 4948515548554856 4948515548554948 4948515548554949 4948515548554951   
## 1 1 1 1   
## 4948515548554952 4948515548554953 4948515548555051 4948515548555148   
## 1 1 1 1   
## 4948565648575551 4948565648575552 4948565648575555 4948565648575556   
## 1 1 1 1   
## 4948565648575648 4948565648575649 4948565648575650 4948565648575654   
## 1 1 1 1   
## 4948565648575749 4948565648575752 4948565648575756 494857574849484849   
## 1 1 1 1   
## 494857574849484855 494857574849484857 494857574849484951 494857574849484954   
## 1 1 1 1   
## 494857574849484955 494857574849484957 494857574849485049 494857574849485050   
## 1 1 1 1   
## 494857574849485052 494857574849485054 494857574849485057 49494848514957   
## 1 1 1 1   
## 49494848515049 49494848515050 49494848515051 49494848515056   
## 1 1 1 1   
## 49494848515057 49494848515148 49494848515152 49494848515153   
## 1 1 1 1   
## 49494848515154 49494848515156 49494848515248 494948524849485750   
## 1 1 1 1   
## 494948524849485751 494948524849485752 494948524849485754 494948524849485757   
## 1 1 1 1   
## 494948524849494851 494948524849494854 494948524849494948 494948524849494949   
## 1 1 1 1   
## 494948524849494950 494948524849494954 494948524849494957 494949534849495051   
## 1 1 1 1   
## 494949534849495052 494949534849495056 494949534849495057 494949534849495152   
## 1 1 1 1   
## 494949534849495153 494949534849495154 494949534849495156 494949534849495249   
## 1 1 1 1   
## 494949534849495254 494949534849495255 494949534849495257 494950514849504951   
## 1 1 1 1   
## 494950514849504952 494950514849504955 494950514849505048 494950514849505050   
## 1 1 1 1   
## 494950514849505051 494950514849505056 494950514849505148 494950514849505152   
## 1 1 1 1   
## 494950514849505153 494950514849505157 494950514849505248 494950524849505250   
## 1 1 1 1   
## 494950524849505251 494950524849505254 494950524849505255 494950524849505256   
## 1 1 1 1   
## 494950524849505257 494950524849505348 494950524849505351 494950524849505355   
## 1 1 1 1   
## 494950524849505452 494950524849505453 494950524849505455 494951484849505549   
## 1 1 1 1   
## 494951484849505550 494951484849505551 494951484849505554 494951484849505555   
## 1 1 1 1   
## 494951484849505557 494951484849505651 494951484849505653 494951484849505657   
## 1 1 1 1   
## 494951484849505753 494951484849505754 494951484849514848 494951504849514849   
## 1 1 1 1   
## 494951504849514850 494951504849514852 494951504849514853 494951504849514857   
## 1 1 1 1   
## 494951504849514948 494951504849514949 494951504849514951 494951504849514955   
## 1 1 1 1   
## 494951504849515049 494951504849515053 494951504849515054 494951534849515150   
## 1 1 1 1   
## 494951534849515151 494951534849515154 494951534849515157 494951534849515248   
## 1 1 1 1   
## 494951534849515249 494951534849515349 494951534849515350 494951534849515352   
## 1 1 1 1   
## 494951534849515355 494951534849515356 494951534849515357 494951544849515449   
## 1 1 1 1   
## 494951544849515450 494951544849515455 494951544849515456 494951544849515548   
## 1 1 1 1   
## 494951544849515555 494951544849515557 494951544849515650 494951544849515651   
## 1 1 1 1   
## 494951544849515653 494951544849515654 494951544849515748 49495248515255   
## 1 1 1 1   
## 49495248515257 49495248515349 49495248515350 49495248515351   
## 1 1 1 1   
## 49495248515353 49495248515354 49495248515357 49495248515453   
## 1 1 1 1   
## 49495248515456 49495248515548 49495248515553 4949525448535354   
## 1 1 1 1   
## 4949525448535450 4949525448535451 4949525448535454 4949525448535456   
## 1 1 1 1   
## 4949525448535457 4949525448535549 4949525448535550 4949525448535556   
## 1 1 1 1   
## 4949525448535649 4949525448535651 4949525448535652 4949525548535655   
## 1 1 1 1   
## 4949525548535749 4949525548535755 4949525548544849 4949525548544852   
## 1 1 1 1   
## 4949525548544853 4949525548544854 4949525548544855 4949525548544948   
## 1 1 1 1   
## 4949525548544950 4949525548544951 4949525548544952 4949565748545555   
## 1 1 1 1   
## 4949565748545557 4949565748545648 4949565748545650 4949565748545652   
## 1 1 1 1   
## 4949565748545655 4949565748545748 4949565748545751 4949565748545754   
## 1 1 1 1   
## 4949565748545757 4949565748554850 4949565748554851 4949575048554855   
## 1 1 1 1   
## 4949575048554857 4949575048554949 4949575048554953 4949575048554955   
## 1 1 1 1   
## 4949575048554956 4949575048554957 4949575048555053 4949575048555055   
## 1 1 1 1   
## 4949575048555056 4949575048555148 4949575048555153 4950484848555155   
## 1 1 1 1   
## 4950484848555251 4950484848555256 4950484848555257 4950484848555349   
## 1 1 1 1   
## 4950484848555351 4950484848555355 4950484848555357 4950484848555448   
## 1 1 1 1   
## 4950484848555451 4950484848555452 4950484948555455 4950484948555456   
## 1 1 1 1   
## 4950484948555457 4950484948555552 4950484948555649 4950484948555650   
## 1 1 1 1   
## 4950484948555652 4950484948555654 4950484948555656 4950484948555657   
## 1 1 1 1   
## 4950484948555748 4950484948555749 4950515548565355 4950515548565357   
## 1 1 1 1   
## 4950515548565449 4950515548565450 4950515548565452 4950515548565453   
## 1 1 1 1   
## 4950515548565455 4950515548565549 4950515548565552 4950515548565554   
## 1 1 1 1   
## 4950515548565555 4950515548565557 4950525448565654 4950525448565655   
## 1 1 1 1   
## 4950525448565656 4950525448565657 4950525448565755 4950525448565757   
## 1 1 1 1   
## 4950525448574849 4950525448574852 4950525448574855 4950525448574856   
## 1 1 1 1   
## 4950525448574952 4950535348574953 4950535348574954 4950535348574955   
## 1 1 1 1   
## 4950535348574956 4950535348574957 4950535348575054 4950535348575056   
## 1 1 1 1   
## 4950535348575149 4950535348575152 4950535348575153 4950535348575154   
## 1 1 1 1   
## 4950535348575157 4950535648575253 4950535648575254 4950535648575255   
## 1 1 1 1   
## 4950535648575350 4950535648575353 4950535648575354 4950535648575357   
## 1 1 1 1   
## 4950535648575450 4950535648575451 4950535648575457 4950535648575549   
## 1 1 1 1   
## 4950535648575550 495054524849484848 495054524849484850 495054524849484851   
## 1 1 1 1   
## 4950545248575648 4950545248575650 4950545248575652 4950545248575653   
## 1 1 1 1   
## 4950545248575656 4950545248575657 4950545248575755 4950545248575756   
## 1 1 1 1   
## 4950545248575757 49505548525757 49505548534848 49505548534849   
## 1 1 1 1   
## 49505548534950 49505548534951 49505548534952 49505548534953   
## 1 1 1 1   
## 49505548534956 49505548534957 49505548535051 49505548535052   
## 1 1 1 1   
## 49505548535053 495055504849484853 495055504849484855 495055504849484856   
## 1 1 1 1   
## 495055504849484857 495055504849484952 495055504849484953 495055504849484956   
## 1 1 1 1   
## 495055504849484957 495055504849485048 495055504849485050 495055504849485055   
## 1 1 1 1   
## 495055504849485057 49505556485549 49505556485550 49505556485554   
## 1 1 1 1   
## 49505556485555 49505556485556 49505556485649 49505556485651   
## 1 1 1 1   
## 49505556485653 49505556485654 49505556485656 49505556485757   
## 1 1 1 1   
## 4950575248494849 4950575248494850 4950575248494851 4950575248494852   
## 1 1 1 1   
## 4950575248494854 4950575248494951 4950575248494955 4950575248495049   
## 1 1 1 1   
## 4950575248495050 4950575248495051 4950575248495053 4950575248495057   
## 1 1 1 1   
## 4950575748505349 4950575748505350 4950575748505351 4950575748505354   
## 1 1 1 1   
## 4950575748505355 4950575748505449 4950575748505450 4950575748505452   
## 1 1 1 1   
## 4950575748505454 4950575748505455 4950575748505548 4950575748505554   
## 1 1 1 1   
## 4951485248505649 4951485248505653 4951485248505749 4951485248505750   
## 1 1 1 1   
## 4951485248505751 4951485248505752 4951485248505754 4951485248514850   
## 1 1 1 1   
## 4951485248514851 4951485248514854 4951485248514857 4951485248514948   
## 1 1 1 1   
## 49515148535055 49515148535057 49515148535148 49515148535153   
## 1 1 1 1   
## 49515148535155 49515148535157 49515148535251 49515148535253   
## 1 1 1 1   
## 49515148535254 49515148535349 49515148535352 495155564849495352   
## 1 1 1 1   
## 495155564849495357 495155564849495448 495155564849495450 495155564849495451   
## 1 1 1 1   
## 495155564849495453 495155564849495549 495155564849495551 495155564849495552   
## 1 1 1 1   
## 495155564849495553 495155564849495555 495155564849495557 495156574849495649   
## 1 1 1 1   
## 495156574849495651 495156574849495653 495156574849495656 495156574849495657   
## 1 1 1 1   
## 495156574849495750 495156574849495751 495156574849495756 495156574849504852   
## 1 1 1 1   
## 495156574849504853 495156574849504854 495156574849504855 4952484950   
## 1 1 1 1   
## 4952484951 4952484955 4952484956 4952484957   
## 1 1 1 1   
## 4952485051 4952485053 49524851 495248524849515750   
## 1 1 1 1   
## 495248524849515751 495248524849515756 495248524849515757 495248524849524850   
## 1 1 1 1   
## 495248524849524854 495248524849524856 495248524849524857 495248524849524950   
## 1 1 1 1   
## 495248524849524954 495248524849524955 495248524849524957 49524853   
## 1 1 1 1   
## 49524854 49524856 49524857 495249564849525049   
## 1 1 1 1   
## 495249564849525050 495249564849525054 495249564849525055 495249564849525149   
## 1 1 1 1   
## 495249564849525150 495249564849525153 495249564849525154 495249564849525156   
## 1 1 1 1   
## 495249564849525252 495249564849525253 495249564849525256 49525153484950   
## 1 1 1 1   
## 49525153484951 49525153484953 49525153485050 49525153485052   
## 1 1 1 1   
## 49525153485054 49525153485056 49525153485148 49525153485151   
## 1 1 1 1   
## 49525153485153 49525153485154 49525153485155 49525154485251   
## 1 1 1 1   
## 49525154485255 49525154485257 49525154485349 49525154485351   
## 1 1 1 1   
## 49525154485353 49525154485354 49525154485356 49525154485357   
## 1 1 1 1   
## 49525154485454 49525154485456 49525154485548 4952534848495749   
## 1 1 1 1   
## 4952534848495751 4952534848495752 4952534848495753 4952534848495754   
## 1 1 1 1   
## 4952534848495755 4952534848504848 4952534848504850 4952534848504851   
## 1 1 1 1   
## 4952534848504853 4952534848504854 4952534848504856 4952535148505050   
## 1 1 1 1   
## 4952535148505054 4952535148505057 4952535148505148 4952535148505150   
## 1 1 1 1   
## 4952535148505151 4952535148505152 4952535148505154 4952535148505155   
## 1 1 1 1   
## 4952535148505250 4952535148505348 4952545148514954 4952545148514955   
## 1 1 1 1   
## 4952545148514957 4952545148515049 4952545148515051 4952545148515052   
## 1 1 1 1   
## 4952545148515055 4952545148515148 4952545148515150 4952545148515152   
## 1 1 1 1   
## 4952545148515157 4952545148515248 4952555148515250 4952555148515251   
## 1 1 1 1   
## 4952555148515252 4952555148515350 4952555148515352 4952555148515353   
## 1 1 1 1   
## 4952555148515354 4952555148515356 4952555148515448 4952555148515449   
## 1 1 1 1   
## 4952555148515450 4952555148515453 4952574848515551 4952574848515555   
## 1 1 1 1   
## 4952574848515557 4952574848515649 4952574848515650 4952574848515651   
## 1 1 1 1   
## 4952574848515656 4952574848515657 4952574848515749 4952574848515754   
## 1 1 1 1   
## 4952574848515757 4952574848524848 4952575048524851 4952575048524856   
## 1 1 1 1   
## 4952575048524857 4952575048524948 4952575048524952 4952575048524954   
## 1 1 1 1   
## 4952575048524955 4952575048525049 4952575048525052 4952575048525055   
## 1 1 1 1   
## 4952575048525056 4952575048525148 4952575748525150 4952575748525151   
## 1 1 1 1   
## 4952575748525152 4952575748525153 4952575748525154 4952575748525156   
## 1 1 1 1   
## 4952575748525251 4952575748525255 4952575748525349 4952575748525351   
## 1 1 1 1   
## 4952575748525354 4952575748525356 4953484848525449 4953484848525453   
## 1 1 1 1   
## 4953484848525455 4953484848525457 4953484848525548 4953484848525554   
## 1 1 1 1   
## 4953484848525557 4953484848525648 4953484848525649 4953484848525651   
## 1 1 1 1   
## 4953484848525653 4953484848525656 4953484948 4953484952   
## 1 1 1 1   
## 4953484957 4953485049 4953485050 4953485053   
## 1 1 1 1   
## 4953485055 4953485056 49534851 49534854   
## 1 1 1 1   
## 49534855 49534856 495349484849485149 495349484849485150   
## 1 1 1 1   
## 495349484849485151 495349484849485152 495349484849485248 495349484849485249   
## 1 1 1 1   
## 495349484849485251 495349484849485252 495349484849485256 495349484849485257   
## 1 1 1 1   
## 495349484849485353 495349484849485357 495352564849485450 495352564849485451   
## 1 1 1 1   
## 495352564849485456 495352564849485548 495352564849485549 495352564849485556   
## 1 1 1 1   
## 495352564849485650 495352564849485651 495352564849485652 495352564849485654   
## 1 1 1 1   
## 495352564849485655 495352564849485657 495448534849504952 495448534849504953   
## 1 1 1 1   
## 495448534849504954 495448534849504956 495448534849504957 495448534849505050   
## 1 1 1 1   
## 495448534849505051 495448534849505056 495448534849505148 495448534849505157   
## 1 1 1 1   
## 495448534849505251 495448564849505252 495448564849505253 495448564849505255   
## 1 1 1 1   
## 495448564849505256 495448564849505257 495448564849505352 495448564849505353   
## 1 1 1 1   
## 495448564849505354 495448564849505355 495448564849505450 495448564849505452   
## 1 1 1 1   
## 495448564849505457 495449494849505552 495449494849505555 495449494849505557   
## 1 1 1 1   
## 495449494849505648 495449494849505652 495449494849505653 495449494849505654   
## 1 1 1 1   
## 495449494849505657 495449494849505753 495449494849505755 495449494849505757   
## 1 1 1 1   
## 495449494849514851 495449524849514854 495449524849514855 495449524849514856   
## 1 1 1 1   
## 495449524849514857 495449524849514949 495449524849514953 495449524849515049   
## 1 1 1 1   
## 495449524849515051 495449524849515052 495449524849515053 495449524849515056   
## 1 1 1 1   
## 495449524849515148 495449574849515153 495449574849515155 495449574849515156   
## 1 1 1 1   
## 495449574849515249 495449574849515250 495449574849515251 495449574849515253   
## 1 1 1 1   
## 495449574849515257 495449574849515352 495449574849515355 495449574849515449   
## 1 1 1 1   
## 495449574849515451 495450524849515453 495450524849515454 495450524849515548   
## 1 1 1 1   
## 495450524849515550 495450524849515552 495450524849515648 495450524849515649   
## 1 1 1 1   
## 495450524849515651 495450524849515652 495450524849515657 495450524849515748   
## 1 1 1 1   
## 495450524849515751 495452574849535656 495452574849535748 495452574849535750   
## 1 1 1 1   
## 495452574849535751 495452574849535753 495452574849535755 495452574849535757   
## 1 1 1 1   
## 495452574849544848 495452574849544849 495452574849544850 495454524849544852   
## 1 1 1 1   
## 495454524849544855 495454524849544856 495454524849544948 495454524849544950   
## 1 1 1 1   
## 495454524849544952 495454524849544953 495454524849544954 495454524849544955   
## 1 1 1 1   
## 495454524849545052 495454524849545055 495454524849545057 495455514849555352   
## 1 1 1 1   
## 495455514849555353 495455514849555356 495455514849555357 495455514849555448   
## 1 1 1 1   
## 495455514849555450 495455514849555456 495455514849555548 495455514849555551   
## 1 1 1 1   
## 495455514849555553 495455514849555555 495455514849555556 495455534849555648   
## 1 1 1 1   
## 495455534849555651 495455534849555654 495455534849555655 495455534849555656   
## 1 1 1 1   
## 495455534849555657 495455534849555749 495455534849555750 495455534849555753   
## 1 1 1 1   
## 495455534849555757 495455534849564852 495455534849564853 495456504849564856   
## 1 1 1 1   
## 495456504849564948 495456504849564949 495456504849564957 495456504849565051   
## 1 1 1 1   
## 495456504849565054 495456504849565055 495456504849565056 495456504849565149   
## 1 1 1 1   
## 495456504849565153 495456504849565154 495456504849565155 495456564849565253   
## 1 1 1 1   
## 495456564849565254 495456564849565257 495456564849565348 495456564849565350   
## 1 1 1 1   
## 495456564849565354 495456564849565355 495456564849565448 495456564849565449   
## 1 1 1 1   
## 495456564849565451 495456564849565452 495457484849565454 495457484849565455   
## 1 1 1 1   
## 495457484849565548 495457484849565550 495457484849565552 495457484849565650   
## 1 1 1 1   
## 495457484849565653 495457484849565654 495457484849565656 495457484849565749   
## 1 1 1 1   
## 495457484849565751 495457484849565752 495457574849565754 495457574849565755   
## 1 1 1 1   
## 495457574849565756 495457574849565757 495457574849574848 495457574849574851   
## 1 1 1 1   
## 495457574849574853 495457574849574857 495457574849574951 495457574849574955   
## 1 1 1 1   
## 495457574849574956 495457574849575049 495552544850485250 495552544850485251   
## 1 1 1 1   
## 495552544850485252 495552544850485256 495552544850485348 495552544850485349   
## 1 1 1 1   
## 495552544850485350 495552544850485353 495552544850485354 495552544850485356   
## 1 1 1 1   
## 495552544850485449 495552544850485453 495553484850485549 495553484850485551   
## 1 1 1 1   
## 495553484850485552 495553484850485554 495553484850485555 495553484850485649   
## 1 1 1 1   
## 495553484850485653 495553484850485655 495553484850485656 495553484850485657   
## 1 1 1 1   
## 495553484850485753 495553484850485755 495753574849495352 495753574849495355   
## 1 1 1 1   
## 495753574849495356 495753574849495451 495753574849495456 495753574849495548   
## 1 1 1 1   
## 495753574849495550 495753574849495551 495753574849495555 495753574849495556   
## 1 1 1 1   
## 495753574849495557 495753574849495650 495754494849495653 495754494849495654   
## 1 1 1 1   
## 495754494849495657 495754494849495751 495754494849495754 495754494849504852   
## 1 1 1 1   
## 495754494849504853 495754494849504855 495754494849504856 495754494849504948   
## 1 1 1 1   
## 495754494849504949 495754494849504950 495756524849525353 495756524849525355   
## 1 1 1 1   
## 495756524849525357 495756524849525449 495756524849525451 495756524849525453   
## 1 1 1 1   
## 495756524849525456 495756524849525551 495756524849525553 495756524849525556   
## 1 1 1 1   
## 495756524849525648 495757524849525652 495757524849525656 495757524849525748   
## 1 1 1 1   
## 495757524849525749 495757524849525750 495757524849525754 495757524849534849   
## 1 1 1 1   
## 495757524849534850 495757524849534856 495757524849534857 495757524849534951   
## 1 1 1 1   
## 504848514849545155 504848514849545156 504848514849545157 504848514849545250   
## 1 1 1 1   
## 504848514849545251 504848514849545255 504848514849545257 504848514849545349   
## 1 1 1 1   
## 504848514849545350 504848514849545353 504848514849545354 504848514849545355   
## 1 1 1 1   
## 504848564849545452 504848564849545453 504848564849545457 504848564849545549   
## 1 1 1 1   
## 504848564849545550 504848564849545554 504848564849545649 504848564849545655   
## 1 1 1 1   
## 504848564849545656 504848564849545749 504848564849545751 50505548515554   
## 1 1 1 1   
## 50505548515556 50505548515557 50505548515651 50505548515652   
## 1 1 1 1   
## 50505548515654 50505548515749 50505548515754 50505548515757   
## 1 1 1 1   
## 50505548524849 50505548524851 50505548524853 50515248524854   
## 1 1 1 1   
## 50515248524857 50515248524948 50515248524953 50515248524954   
## 1 1 1 1   
## 50515248525049 50515248525051 50515248525055 50515248525149   
## 1 1 1 1   
## 50515248525150 50515248525151 50515248525152 50534948494848   
## 1 1 1 1   
## 50534948494850 50534948494948 50534948494949 505349485651   
## 1 1 1 1   
## 505349485653 505349485656 505349485749 505349485751   
## 1 1 1 1   
## 505349485752 505349485755 505349485757 50555548494952   
## 1 1 1 1   
## 50555548494953 50555548494956 50555548495048 50555548495049   
## 1 1 1 1   
## 50555548495052 50555548495053 50555548495056 50555548495057   
## 1 1 1 1   
## 50555548495148 50555548495151 5151485150 5151485151   
## 1 1 1 1   
## 5151485156 5151485248 5151485252 5151485253   
## 1 1 1 1   
## 5151485255 5151485256 5151485351 5151485355   
## 1 1 1 1   
## 5151485356 5151485448 5154485450 5154485451   
## 1 1 1 1   
## 5154485454 5154485455 5154485456 5154485457   
## 1 1 1 1   
## 5154485552 5154485554 5154485555 5154485557   
## 1 1 1 1   
## 535348495154 535348495155 535348495251 535348495255   
## 1 1 1 1   
## 535348495256 535348495351 535348495352 535348495353   
## 1 1 1 1   
## 535348495357 535348495448 535348495450 535348495453   
## 1 1 1 1   
## 535548495455 535548495456 535548495549 535548495552   
## 1 1 1 1   
## 535548495554 535548495649 535548495653 535548495655   
## 1 1 1 1   
## 535548495657 535548495750 535548495752 535548495753   
## 1 1 1 1   
## 54485248495757 54485248504848 54485248504850 54485248504851   
## 1 1 1 1   
## 54485248504852 54485248504857 54485248504948 54485248504954   
## 1 1 1 1   
## 54485248504957 54485248505050 54485248505051 54485248505052   
## 1 1 1 1   
## 54495048505055 54495048505057 54495048505149 54495048505151   
## 1 1 1 1   
## 54495048505153 54495048505154 54495048505156 54495048505157   
## 1 1 1 1   
## 54495048505251 54495048505253 54495048505255 54495048505348   
## 1 1 1 1   
## 57555348495152 57555348495155 57555348495157 57555348495250   
## 1 1 1 1   
## 57555348495251 57555348495252 57555348495254 57555348495348   
## 1 1 1 1   
## 57555348495355 57555348495356 57555348495357 57555348495448   
## 1 1 1 1   
## 57555648495451 57555648495455 57555648495549 57555648495553   
## 1 1 1 1   
## 57555648495554 57555648495557 57555648495652 57555648495653   
## 1 1 1 1   
## 57555648495654 57555648495655 57555648495656 57555648495657   
## 1 1 1 1   
## 57574848544957 57574848545048 57574848545050 57574848545052   
## 1 1 1 1   
## 57574848545053 57574848545057 57574848545148 57574848545152   
## 1 1 1 1   
## 57574848545153 57574848545156 57574848545250 57574848545251   
## 1 1 1 1   
## 57575148545255 57575148545257 57575148545348 57575148545349   
## 1 1 1 1   
## 57575148545350 57575148545352 57575148545449 57575148545453   
## 1 1 1 1   
## 57575148545548 57575148545549 57575148545551 57575148545553   
## 1 1 1 1   
## 57575548525749 57575548525751 57575548525752 57575548525753   
## 1 1 1 1   
## 57575548525757 57575548534855 57575548534857 57575548534951   
## 1 1 1 1   
## 57575548534954 57575548534955 57575548534956 57575548534957   
## 1 1 1 1