#Installation et importation des packages  
  
# Liste des packages nécessaires  
packages <- c(  
 "tidyverse", # Manipulation & visualisation de données : inclut : ggplot2, dplyr, tidyr, readr, purrr, tibble, stringr, forcats  
 "janitor", # Nettoyage des données  
 "gtsummary", # Tableaux statistiques formatés pour Word/HTML  
 "sf", # Données spatiales (cartographie, shapefiles)  
 "haven", # Lecture de fichiers stata(.dta)  
 "flextable", # Mise en forme avancée de tableaux Word  
 "officer", # Interaction avec Word (officedown)  
 "officedown", # Intégration R Markdown → Word enrichi  
 "sf"  
)  
  
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 ()  
}

flextable(data.frame(Contenu = "REPUBLIQUE DU SENEGAL")) %>% #Créer un tableau flextable  
 delete\_part(part = "header") %>% #supprimer l'en-tête  
 border\_remove() %>% #Supprimer les bordures  
   
 # Application de la police  
 font(fontname = "Times New Roman", part = "all") %>%   
 fontsize(size = 14, part = "all") %>%   
   
 # Mise en forme du texte  
 bold(i = 1, j = 1) %>%   
 align(align = "center") %>%   
   
 set\_table\_properties(layout = "autofit", width = 1) #Ajuste la largeur des colonnes automatiquements

|  |
| --- |
| **REPUBLIQUE DU SENEGAL** |

|  |
| --- |
|  |

flextable(data.frame(Contenu = c( "\*\*\*\*\*\*\*\*\*\*",  
 "Un Peuple - Un But - Une Foi",  
 "\*\*\*\*\*\*\*\*\*\*",  
 "Agence nationale de la Statistique et de la démographie"))) %>%   
 delete\_part(part = "header") %>%   
 border\_remove() %>%   
   
 # Application de la police  
 font(fontname = "Times New Roman", part = "all") %>%   
 fontsize(size = 14, part = "all") %>%  
   
 # Mise en forme du texte  
 bold(i = 1:4, j = 1) %>%   
 italic(i = 2, j = 1) %>%   
 align(align = "center") %>%   
   
 set\_table\_properties(layout = "autofit", width = 1)

|  |
| --- |
| **\*\*\*\*\*\*\*\*\*\*** |
| ***Un Peuple - Un But - Une Foi*** |
| **\*\*\*\*\*\*\*\*\*\*** |
| **Agence nationale de la Statistique et de la démographie** |

|  |
| --- |
|  |

flextable(data.frame(Contenu = c("\*\*\*\*\*\*\*\*\*\*",  
 "Ecole nationale de la Statistique et de l'Analyse économique Pierre Ndiaye"))) %>%  
 delete\_part(part = "header") %>%   
 border\_remove() %>%   
   
 # Application de la police  
 font(fontname = "Times New Roman", part = "all") %>%   
 fontsize(size = 14, part = "all") %>%   
   
 # Mise en forme du texte  
 bold(i = 1:2, j = 1) %>%   
 align(align = "center") %>%   
   
 set\_table\_properties(layout = "autofit", width = 1)

|  |
| --- |
| **\*\*\*\*\*\*\*\*\*\*** |
| **Ecole nationale de la Statistique et de l'Analyse économique Pierre Ndiaye** |

|  |
| --- |
|  |

##### Projet statistique sur R : Evaluation

donnees <- data.frame(  
 col1 = c("Rédigé par", "LAWA FOUMSOU Prosper", "Élève Ingénieur Statisticien Économiste"),  
 col2 = c("Sous la supervision de", "M. Aboubacar HEMA", "Data-scientist"),  
 stringsAsFactors = FALSE  
)  
flextable(donnees) %>%   
 delete\_part(part = "header") %>%   
 width(j = 1:2, width = 0.5) %>% # Répartition égale de la largeur  
 align(j = 1, align = "left") %>%   
 align(j = 2, align = "right") %>%   
 border\_remove() %>%   
 bold(i = 1, j = 1:2) %>%   
 italic(i = 3, j = 1:2) %>%   
 set\_table\_properties(layout = "autofit", width = 1) %>% # Force la largeur totale  
 fontsize(size = 11) # Ajuste la taille de police si nécessaire

|  |  |
| --- | --- |
| **Rédigé par** | **Sous la supervision de** |
| LAWA FOUMSOU Prosper | M. Aboubacar HEMA |
| *Élève Ingénieur Statisticien Économiste* | *Data-scientist* |

|  |
| --- |
| **Année académique 2024-2025** |

###### Sommaire

N’est pas généré automatiquement donc le faire directement avec word.

###### Liste des figures

N’est pas généré automatiquement donc le faire directement avec word.

###### Liste des tableaux

N’est pas généré automatiquement donc le faire directement avec word.

# Introduction

Cette étude s’inscrit dans le cadre de l’**Enquête sur les déplacements forcés au Sud-Soudan menée en 2023**, un pays affecté par des crises humanitaires prolongées dues à des conflits armés, des catastrophes naturelles et des déplacements massifs de population. Ces dynamiques ont eu un impact significatif sur la sécurité alimentaire des ménages, en particulier parmi les populations déplacées internes (IDPs).

L’objectif principal de cette analyse est d’évaluer la qualité de la consommation alimentaire des ménages à l’aide du **Score de Consommation Alimentaire (SCA)**, un indicateur développé par le **Programme Alimentaire Mondial (PAM)**. Ce score combine la diversité alimentaire, la fréquence de consommation et la valeur nutritionnelle des groupes alimentaires consommés durant les 7 derniers jours.

L’utilisation de **R** permet d’effectuer un traitement reproductible des données, de calculer le SCA selon les normes du PAM, et de classifier les ménages en fonction de leur niveau de sécurité alimentaire. Cette analyse repose sur les seuils adaptés au contexte du **Sud-Soudan**, en tenant compte des recommandations techniques du PAM en matière de classification.

Les résultats obtenus permettront d’identifier les ménages les plus vulnérables en matière d’accès à une alimentation adéquate et d’orienter les interventions humanitaires de manière ciblée et efficace.

# I. Importation et Analyse de consistance des bases

## 1. Importation des jeux de données

Commençons par avoir une idée du nombre de variables et d’observations dans chaque base.

### Base principale

La base principale a 3058 lignes et 1312 colonnes.

### Base MAD

La base individus a 22092 lignes et 22092 colonnes.

# II. Analyse de consistance des bases

# Suppression de colonnes vides  
principal\_dataset <- principal\_dataset %>%   
 janitor::remove\_empty(which = c("rows", "cols"))  
  
Individus\_dataset <- Individus\_dataset %>% janitor::remove\_empty(which = c("rows", "cols"))

La base principale a 3058 lignes et 1244 colonnes après suppression de colonnes vides.La base individus a 22092 lignes et 22092 colonnes après suppression de colonnes vides.

## Detection de doublons

On remarque qi’il n’ y a pas de doublons dans la base principale

# Identification du nombre de duplication dans la base principale.   
principal\_dataset %>%   
 group\_by(id) %>%   
 mutate(nb\_line=n()) %>% arrange(desc(nb\_line)) %>%   
 ungroup %>%  
 count(nb\_line)

Aussi on remarque qu’il n’y a pas de doublons dans la base individus. Passons donc au traitement des données manquantes

## Traitement de données manquantes

table\_manquant1 <- principal\_dataset %>%  
 naniar::miss\_var\_summary() %>% # Résumé du nombre et pourcentage de valeurs manquantes par variable  
 dplyr::filter(n\_miss != 0) %>% # On garde seulement les variables qui ont au moins une valeur manquante  
   
 dplyr::rename("manquant" = "n\_miss", # On renomme les colonnes pour plus de lisibilité  
 "pourcentage" = "pct\_miss")  
table\_manquant1

table\_manquant2 <- Individus\_dataset %>%  
 naniar::miss\_var\_summary() %>% # Résumé du nombre et pourcentage de valeurs manquantes par variable  
 dplyr::filter(n\_miss != 0) %>% # On garde seulement les variables qui ont au moins une valeur manquante  
   
 dplyr::rename("manquant" = "n\_miss", # On renomme les colonnes pour plus de lisibilité  
 "pourcentage" = "pct\_miss")  
table\_manquant2

# Analyse socio-economique des deplaces internes du Sudan en 2023

## 1) Pyramide des ages des individus

## **2) Caracteristiques du chef de menage**

# Appliquer le thème compact  
set\_gtsummary\_theme(theme\_gtsummary\_compact())  
  
# Convertir les colonnes "labelled" en facteurs et s'assurer que l'âge est numérique  
Individus\_dataset <- Individus\_dataset %>%  
 filter(!is.na(age\_years), !is.na(hh\_02), !is.na(hh\_08)) %>%  
 mutate(  
 intro\_07\_1 = haven::as\_factor(intro\_07\_1),  
 hh\_02 = haven::as\_factor(hh\_02),  
 hh\_08 = haven::as\_factor(hh\_08),  
 age\_years = as.numeric(age\_years)  
 )  
  
# Création du tableau sociodémographique  
tableau\_analyse\_sociodemo <- Individus\_dataset %>%  
 tbl\_summary(  
 include = c(age\_years, hh\_02, hh\_08),  
 by = intro\_07\_1,   
 statistic = list(  
 all\_continuous() ~ "{mean} ({sd})",  
 all\_categorical() ~ "{n} ({p}%)"  
 ),  
 label = list(  
 hh\_02 ~ "Sexe du chef de ménage",  
 age\_years ~ "Âge du chef de ménage",  
 hh\_08 ~ "Situation matrimoniale"  
 ),  
 missing = "no"  
 ) %>%  
 add\_n() %>%  
 modify\_header(label = "\*\*Variables socio-démographiques\*\*") %>%  
 bold\_labels() %>%  
 italicize\_levels()  
  
# Conversion en flextable  
as\_flex\_table(tableau\_analyse\_sociodemo)

| **Variables socio-démographiques** | **N** | **Refugees**, N = 9,2851 | **Asylum-seekers**, N = 01 | **Host community North**, N = 4,9631 | **Returnees**, N = 01 |
| --- | --- | --- | --- | --- | --- |
| **Âge du chef de ménage** | 14,248 | 27 (14) | NA (NA) | 29 (15) | NA (NA) |
| **Sexe du chef de ménage** | 14,248 |  |  |  |  |
| *Male* |  | 4,417 (48%) | 0 (NA%) | 2,405 (48%) | 0 (NA%) |
| *Female* |  | 4,868 (52%) | 0 (NA%) | 2,558 (52%) | 0 (NA%) |
| **Situation matrimoniale** | 14,248 |  |  |  |  |
| *monogamous/married* |  | 3,217 (35%) | 0 (NA%) | 2,176 (44%) | 0 (NA%) |
| *polygamous/married* |  | 438 (4.7%) | 0 (NA%) | 210 (4.2%) | 0 (NA%) |
| *non-formal union* |  | 29 (0.3%) | 0 (NA%) | 8 (0.2%) | 0 (NA%) |
| *separated* |  | 207 (2.2%) | 0 (NA%) | 53 (1.1%) | 0 (NA%) |
| *divorced* |  | 127 (1.4%) | 0 (NA%) | 30 (0.6%) | 0 (NA%) |
| *widow or widower* |  | 256 (2.8%) | 0 (NA%) | 165 (3.3%) | 0 (NA%) |
| *never married* |  | 5,011 (54%) | 0 (NA%) | 2,321 (47%) | 0 (NA%) |
| 1Mean (SD); n (%) | | | | | |

## **3) Crowding Index ou l’indice d’affluence**

# **Analyse de la securite alimentaire des deplaces internes**

## Score de consommation alimentaire (SCA)

### Faites une analyse descriptive des variables qui composent le SCA

fcs\_vars <- principal\_dataset %>%  
 dplyr::select("food\_div1","food\_div2","food\_div3","food\_div4","food\_div5","food\_div6","food\_div7","food\_div8")

Les variables concernées sont : food\_div1, food\_div2, food\_div3, food\_div4, food\_div5, food\_div6, food\_div7, food\_div8.

desc\_var\_sca <- principal\_dataset %>%  
 select(food\_div1, food\_div2, food\_div3, food\_div4, food\_div5, food\_div6, food\_div7, food\_div8) %>%  
 tbl\_summary(  
 statistic = list(all\_continuous() ~ "{mean} ({sd})\nmin: {min}, max: {max}\nP25: {p25}, P75: {p75}"),  
 digits = all\_continuous() ~ 2,  
 label = list(  
 food\_div1 ~ "Céréales & tubercules (jours)",  
 food\_div2 ~ "Légumineuses/noix (jours)",  
 food\_div3 ~ "Légumes (jours)",  
 food\_div4 ~ "Fruits (jours)",  
 food\_div5 ~ "Viande/poisson/œufs (jours)",  
 food\_div6 ~ "Produits laitiers (jours)",  
 food\_div7 ~ "Sucre (jours)",  
 food\_div8 ~ "Matières grasses (jours)"  
 ),  
 missing = "no"  
 ) %>%  
 as\_flex\_table() %>%  
 theme\_zebra()  
  
desc\_var\_sca

| **Characteristic** | **N = 3,0581** |
| --- | --- |
| Céréales & tubercules (jours) |  |
| 0 | 677 (22%) |
| 1 | 450 (15%) |
| 2 | 253 (8.3%) |
| 3 | 365 (12%) |
| 4 | 230 (7.5%) |
| 5 | 297 (9.7%) |
| 6 | 92 (3.0%) |
| 7 | 690 (23%) |
| Légumineuses/noix (jours) |  |
| 0 | 1,900 (62%) |
| 1 | 317 (10%) |
| 2 | 276 (9.0%) |
| 3 | 228 (7.5%) |
| 4 | 89 (2.9%) |
| 5 | 86 (2.8%) |
| 6 | 43 (1.4%) |
| 7 | 113 (3.7%) |
| Légumes (jours) |  |
| 0 | 2,479 (81%) |
| 1 | 228 (7.5%) |
| 2 | 113 (3.7%) |
| 3 | 79 (2.6%) |
| 4 | 27 (0.9%) |
| 5 | 38 (1.2%) |
| 6 | 15 (0.5%) |
| 7 | 72 (2.4%) |
| Fruits (jours) |  |
| 0 | 2,179 (71%) |
| 1 | 397 (13%) |
| 2 | 231 (7.6%) |
| 3 | 124 (4.1%) |
| 4 | 34 (1.1%) |
| 5 | 31 (1.0%) |
| 6 | 12 (0.4%) |
| 7 | 40 (1.3%) |
| Viande/poisson/œufs (jours) |  |
| 0 | 2,099 (69%) |
| 1 | 244 (8.0%) |
| 2 | 243 (8.0%) |
| 3 | 175 (5.8%) |
| 4 | 91 (3.0%) |
| 5 | 64 (2.1%) |
| 6 | 15 (0.5%) |
| 7 | 112 (3.7%) |
| Produits laitiers (jours) |  |
| 0 | 2,775 (91%) |
| 1 | 56 (1.8%) |
| 2 | 56 (1.8%) |
| 3 | 47 (1.5%) |
| 4 | 25 (0.8%) |
| 5 | 38 (1.2%) |
| 6 | 3 (<0.1%) |
| 7 | 42 (1.4%) |
| Sucre (jours) |  |
| 0 | 2,144 (70%) |
| 1 | 152 (5.0%) |
| 2 | 198 (6.5%) |
| 3 | 199 (6.5%) |
| 4 | 109 (3.6%) |
| 5 | 70 (2.3%) |
| 6 | 41 (1.3%) |
| 7 | 134 (4.4%) |
| Matières grasses (jours) |  |
| 0 | 2,280 (75%) |
| 1 | 194 (6.4%) |
| 2 | 161 (5.3%) |
| 3 | 129 (4.2%) |
| 4 | 73 (2.4%) |
| 5 | 66 (2.2%) |
| 6 | 33 (1.1%) |
| 7 | 106 (3.5%) |
| **1n (%)** | |

### Calculer le score de consommation alimentaire

# Étape 1 : liste des variables d'alimentation  
fcs\_vars <- c("food\_div1","food\_div2","food\_div3","food\_div4",  
 "food\_div5","food\_div6","food\_div7","food\_div8")  
  
# Étape 2 : poids dans l'ordre correspondant  
principal\_dataset <- principal\_dataset %>%  
 mutate(  
 # Calcul du score pondéré par groupe alimentaire avec food\_div  
   
 # food\_div1 : Céréales & tubercules – poids 2  
 food\_div1\_w = food\_div1 \* 2,  
  
 # food\_div2 : Légumineuses/noix – poids 3  
 food\_div2\_w = food\_div2 \* 3,  
  
 # food\_div3 : Légumes – poids 1  
 food\_div3\_w = food\_div3 \* 1,  
  
 # food\_div4 : Fruits – poids 1  
 food\_div4\_w = food\_div4 \* 1,  
  
 # food\_div5 : Viande/poisson/œufs – poids 4  
 food\_div5\_w = food\_div5 \* 4,  
  
 # food\_div6 : Produits laitiers – poids 4  
 food\_div6\_w = food\_div6 \* 4,  
  
 # food\_div7 : Sucre – poids 0.5  
 food\_div7\_w = food\_div7 \* 0.5,  
  
 # food\_div8 : Matières grasses – poids 0.5  
 food\_div8\_w = food\_div8 \* 0.5,  
  
 # Score total SCA  
 SCA = food\_div1\_w + food\_div2\_w + food\_div3\_w + food\_div4\_w +food\_div5\_w + food\_div6\_w + food\_div7\_w + food\_div8\_w  
 )

### Faites un tableau illustrant le poids attribue a chaque groupe alimentaire pour le calcul du SCA (la somme totale des poids doit etre egale a 16)

### Categoriser le SCA selon les seuil 21/35 et 28/42

# Classification des ménages selon les seuils standards (21/35 et 28/42)  
principal\_dataset <- principal\_dataset %>%   
 mutate(  
 # Catégorisation avec le seuil 21/35  
 fcs\_cat\_21\_35 = case\_when(  
 SCA <= 21 ~ "Pauvre",  
 SCA <= 35 ~ "Limite",  
 TRUE ~ "Acceptable"  
 ),  
 fcs\_cat\_21\_35 = factor(fcs\_cat\_21\_35, levels = c("Pauvre", "Limite", "Acceptable")),  
   
 # Catégorisation avec le seuil 28/42  
 fcs\_cat\_28\_42 = case\_when(  
 SCA <= 28 ~ "Pauvre",  
 SCA <= 42 ~ "Limite",  
 TRUE ~ "Acceptable"  
 ),  
 fcs\_cat\_28\_42 = factor(fcs\_cat\_28\_42, levels = c("Pauvre", "Limite", "Acceptable"))  
 )

# L’indice réduit des stratégies de survie (rCSI):

# Table des matières

###### Conclusion

Merci pour votre attention