

# P4\_FAO

SIMON SAGNA

18/07/2021

## Réalisez une étude de santé publique avec R ou Python

Il existe un lien logique entre l'ensemble des 10 variables exprimées en milliers de tonnes sur les utilisations de la disponibilité intérieure, qui peut s'exprimer avec l'équation suivante :

-Production + Importations - Exportations + Variation de stock = Disponibilité intérieure = Semences + Pertes + Nourriture + Aliments pour animaux + Traitement + Autres utilisations

## Importation de la librairie

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.0.5
```

```
## -- Attaching packages ----- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.3      v purrr   0.3.4
## v tibble  3.1.0      v dplyr   1.0.5
## v tidyr   1.1.3      v stringr 1.4.0
## v readr   1.4.0      v forcats 0.5.1
```

```
## Warning: package 'ggplot2' was built under R version 4.0.5
```

```
## Warning: package 'tidyr' was built under R version 4.0.5
```

```
## Warning: package 'readr' was built under R version 4.0.5
```

```
## Warning: package 'stringr' was built under R version 4.0.5
```

```
## Warning: package 'forcats' was built under R version 4.0.5
```

```
## -- Conflicts ----- tidyverse_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag()    masks stats::lag()
```

```
library(dplyr)  
library(data.table)
```

```
## Warning: package 'data.table' was built under R version 4.0.5
```

```
##  
## Attaching package: 'data.table'
```

```
## The following objects are masked from 'package:dplyr':  
##  
##   between, first, last
```

```
## The following object is masked from 'package:purrr':  
##  
##   transpose
```

## Importation des données

```
setwd("C:/Users/stars/OneDrive/")

mydir = "Documents"

myfiles = list.files(path=mydir, pattern="*.csv", full.names=TRUE)

myfiles
```

```
## [1] "Documents/ADRESSE TEST CSV.csv"
## [2] "Documents/aide_alimentaire.csv"
## [3] "Documents/BASE_IDE-DESKTOP-L22R443.csv"
## [4] "Documents/BASE_IDE.csv"
## [5] "Documents/BASE_IDE_csv.csv"
## [6] "Documents/dispo alimentaire off.csv"
## [7] "Documents/immobilier.csv"
## [8] "Documents/population.csv"
## [9] "Documents/sous_nutrition.csv"
## [10] "Documents/webCSV.csv"
```

```
population <- read.csv("Documents/population.csv")
aide_alimentaire<-read.csv("Documents/aide_alimentaire.csv")
dispo_alimentaire.off<-read.csv("Documents/dispo alimentaire off.csv")
sous_nutrition<-read.csv("Documents/sous_nutrition.csv")
```

## Découverte des dataframes

```
glimpse(population)
```

```
## Rows: 1,416
## Columns: 3
## $ Zone   <chr> "Afghanistan", "Afghanistan", "Afghanistan", "Afghanistan", "Af~
## $ Année  <int> 2013, 2014, 2015, 2016, 2017, 2018, 2013, 2014, 2015, 2016, 201~
## $ Valeur <dbl> 32269.589, 33370.794, 34413.603, 35383.032, 36296.113, 37171.92~
```

```
glimpse(aide_alimentaire)
```

```
## Rows: 1,475
## Columns: 4
## $ Pays.bénéficiaire <chr> "Afghanistan", "Afghanistan", "Afghanistan", "Afghan~
## $ Année              <int> 2013, 2014, 2013, 2014, 2013, 2014, 2013, 2013, 2014~
## $ Produit            <chr> "Autres non-céréales", "Autres non-céréales", "Blé e~
## $ Valeur             <int> 682, 335, 39224, 15160, 40504, 15989, 85, 11087, 818~
```

```
glimpse(dispo_alimentaire.off)
```

```
## Rows: 15,605
## Columns: 18
## $ Zone                                     <chr> "Afghani~
## $ Produit                                <chr> "Abats C~
## $ Origine                                <chr> "animale~
## $ Aliments.pour.animaux                  <int> NA, NA, ~
## $ Autres.Utilisations                    <int> NA, NA, ~
## $ Disponibilité.alimentaire..Kcal.personne.jour. <int> 5, 1, 1,~
## $ Disponibilité.alimentaire.en.quantité..kg.personne.an. <dbl> 1.72, 1.~
## $ Disponibilité.de.matière.grasse.en.quantité..g.personne.jour. <dbl> 0.20, 0.~
## $ Disponibilité.de.protéines.en.quantité..g.personne.jour. <dbl> 0.77, 0.~
## $ Disponibilité.intérieure                <int> 53, 41, ~
## $ Exportations...Quantité                <int> NA, 2, N~
## $ Importations...Quantité                <int> NA, 40, ~
## $ Nourriture                             <int> 53, 39, ~
## $ Pertes                                 <int> NA, 2, N~
## $ Production                             <int> 53, 3, N~
## $ Semences                               <int> NA, NA, ~
## $ Traitement                             <int> NA, NA, ~
## $ Variation.de.stock                     <int> NA, NA, ~
```

```
glimpse(sous_nutrition)
```

```
## Rows: 1,218
## Columns: 3
## $ Zone <chr> "Afghanistan", "Afghanistan", "Afghanistan", "Afghanistan", "Af~
## $ Année <chr> "2012-2014", "2013-2015", "2014-2016", "2015-2017", "2016-2018"~
## $ Valeur <chr> "8.6", "8.8", "8.9", "9.7", "10.5", "11.1", "2.2", "2.5", "2.8"~
```

```
summary(population)
```

```
##      Zone      Année      Valeur
## Length:1416   Min.   :2013   Min.    :    0.8
## Class :character 1st Qu.:2014 1st Qu.:   378.3
## Mode  :character Median :2016 Median :  5126.5
##              Mean  :2016 Mean  : 31447.9
##              3rd Qu.:2017 3rd Qu.: 19306.7
##              Max.   :2018 Max.   :1427647.8
```

```
summary(aide_alimentaire)
```

```
## Pays.bénéficiaire  Année      Produit      Valeur
## Length:1475       Min.   :2013 Length:1475   Min.    :    0.0
## Class :character  1st Qu.:2013 Class :character 1st Qu.:   287.5
## Mode  :character  Median :2014 Mode  :character Median :  1178.0
##              Mean  :2014              Mean  :  7482.0
##              3rd Qu.:2015              3rd Qu.:  4334.5
##              Max.   :2016              Max.   :265013.0
```

```
summary(sous_nutrition)
```

```
##      Zone      Année      Valeur
## Length:1218   Length:1218   Length:1218
## Class :character Class :character Class :character
## Mode  :character Mode  :character Mode  :character
```

```
summary(dispo_alimentaire.off)
```

```

##      Zone      Produit      Origine      Aliments.pour.animaux
## Length:15605   Length:15605   Length:15605   Min.    :    0.0
## Class :character Class :character Class :character 1st Qu.:    0.0
## Mode  :character Mode  :character Mode  :character Median :    4.0
##                                     Mean   :   479.5
##                                     3rd Qu.:    74.0
##                                     Max.   :150000.0
##                                     NA's    :12885
## Autres.Utilisations Disponibilité.alimentaire..Kcal.personne.jour.
## Min.    :    0.0   Min.    : -21.00
## 1st Qu.:    0.0   1st Qu.:  0.00
## Median :    0.0   Median :  4.00
## Mean   :   157.4   Mean   :  34.79
## 3rd Qu.:    4.0   3rd Qu.:  21.00
## Max.   :347309.0   Max.   :1711.00
## NA's   :10109     NA's   :1364
## Disponibilité.alimentaire.en.quantité..kg.personne.an.
## Min.    : -1.930
## 1st Qu.:  0.060
## Median :  0.830
## Mean   :  8.719
## 3rd Qu.:  5.190
## Max.   :430.760
## NA's   :1590
## Disponibilité.de.matière.grasse.en.quantité..g.personne.jour.
## Min.    :-0.030
## 1st Qu.:  0.010
## Median :  0.080
## Mean   :  1.283
## 3rd Qu.:  0.630
## Max.   :60.760
## NA's   :3811
## Disponibilité.de.protéines.en.quantité..g.personne.jour.
## Min.    :-0.370
## 1st Qu.:  0.010
## Median :  0.100
## Mean   :  1.224
## 3rd Qu.:  0.660
## Max.   :54.970

```

```
## NA's :4044
## Disponibilité.intérieure Exportations...Quantité Importations...Quantité
## Min. : -3430.0 Min. : -41.0 Min. : -201.00
## 1st Qu.: 0.0 1st Qu.: 0.0 1st Qu.: 0.00
## Median : 7.0 Median : 0.0 Median : 2.00
## Mean : 640.3 Mean : 110.6 Mean : 87.26
## 3rd Qu.: 76.8 3rd Qu.: 9.0 3rd Qu.: 18.00
## Max. :739267.0 Max. :42797.0 Max. :63381.00
## NA's :223 NA's :3379 NA's :753
## Nourriture Pertes Production Semences
## Min. : -246.0 Min. : 0 Min. : 0.0 Min. : 0.00
## 1st Qu.: 0.0 1st Qu.: 0 1st Qu.: 2.0 1st Qu.: 0.00
## Median : 5.0 Median : 4 Median : 22.0 Median : 2.00
## Mean : 347.9 Mean : 106 Mean : 1090.4 Mean : 73.97
## 3rd Qu.: 52.0 3rd Qu.: 26 3rd Qu.: 191.2 3rd Qu.: 17.00
## Max. :426850.0 Max. :55047 Max. :739267.0 Max. :17060.00
## NA's :1590 NA's :11327 NA's :6425 NA's :13514
## Traitement Variation.de.stock
## Min. : -19.0 Min. : -39863.00
## 1st Qu.: 0.0 1st Qu.: 0.00
## Median : 6.0 Median : 0.00
## Mean : 961.9 Mean : -15.41
## 3rd Qu.: 69.0 3rd Qu.: 0.00
## Max. :326711.0 Max. : 5284.00
## NA's :13313 NA's :8829
```

## Population

```
population[,3]<- 1000*population[,3]
```

J'ai multiplié par 1000 car la population est par milliers d'habitants

## Disponibilité alimentaire



```
dispo_alimentaire.off[is.na(dispo_alimentaire.off)] = 0
col_tonnes_kg = c('Aliments.pour.animaux', 'Disponibilité.intérieure', 'Exportations...Quantité', 'Importations...Quantité', 'No
urriture', 'Pertes', 'Production', 'Semences', 'Traitement', 'Variation.de.stock', 'Autres.Utilisations')
for (i in 1:length(col_tonnes_kg)){
  elt = col_tonnes_kg[i]
  dispo_alimentaire.off[elt] = dispo_alimentaire.off[elt] * 1000000
}
```

Remplacer les NA par 0 Multiplier par 1 000 000 toutes les colonnes dont l'unité est en milliers de tonnes afin de les convertir en kg

## Aide alimentaire

```
aide_alimentaire[,4]<- 1000*aide_alimentaire[,4]
```

Multiplier par 1000 pour avoir en kg

## Sous nutrition

```
sous_nutrition$Valeur = as.numeric(as.character(sous_nutrition$Valeur))
```

```
## Warning: NAs introduced by coercion
```

```
sous_nutrition[is.na(sous_nutrition)] = 0

sous_nutrition[,3]<-1000000*sous_nutrition[,3]
customf<-function(x){if(x=="<0,1"){return (0)}else{return (as.numeric(x))}}
```

Transformer les valeurs en numérique. Remplacer les NA par 0. Multiplier par 1000000 afin de l'avoir en habitant. Faire passer la colonne "valeur" en numérique car elle contient des valeurs non fixes (ex:<0,1 ). DE ce fait je remplace ces valeurs non fixes par 0.

## Population 2017

```
population2017<-population[population$Année==2017,]
```

Selectionner les données de la population en ciblant l'année 2017

## Supression duplicat pour population

```
population2017<-population2017[!duplicated(population2017$Zone),]
```

## Sous nutrition 2017

```
sousnutrition2017<-sous_nutrition[sous_nutrition$Année=="2016-2018",c("Zone","Valeur")]  
colnames(sousnutrition2017)<-c("Zone","Val")
```

Selectionner les données de la sous nutrition en ciblant l'année 2017 ("2016-2018")

## Jointures entre population 2017 et sous nutrition 2017

```
dfprop<-inner_join(population2017,sousnutrition2017)
```

```
## Joining, by = "Zone"
```

faire une jointure entre population2017 et sous nutrition2017

## Calcul de la proportion de personnes en état de sous-nutrition

```
newprop<-sapply(dfprop$Val,customf)  
dfprop<-cbind(dfprop,newprop)  
proportion = sum(dfprop$Val) * 100 / sum(dfprop$Valeur)  
print(paste("Proportion de personnes en état de sous nutrition :", proportion, "%"))
```

```
## [1] "Proportion de personnes en état de sous nutrition : 7.10119683323542 %"
```

## Le nombre théorique de personnes qui pourraient être nourries (Disponibilité alimentaire)

```
pop = population[population['Année'] == 2017, c("Zone", "Valeur")]

dispo_alimentaire.off = merge(dispo_alimentaire.off, pop, by='Zone')

dispo_kcal<- dispo_alimentaire.off$Disponibilité.alimentaire..Kcal.personne.jour.*dispo_alimentaire.off$Valeur * 365
Dispo_kcal_total<-sum(dispo_kcal)
total_kcal = round(sum(Dispo_kcal_total)/(2500*365))

print(paste("le nombre de personnes qui pourraient etre nourries :", total_kcal))
```

```
## [1] "le nombre de personnes qui pourraient etre nourries : 8367593851"
```

```
print(paste("Prop :", total_kcal*100/sum(population[population['Année'] == 2017,"Valeur"]), "%"))
```

```
## [1] "Prop : 110.856454428993 %"
```

Tout d'abord, j'ai choisi les kcal, ensuite j'ai converti la colonne de disponibilité alimentaire en kcal car elle était en kcal.pers.jour. De ce fait, j'ai multiplié par la population et par 365. A cela se rajoute une jointure entre la disponibilité alimentaire et la population. Au vu de mes recherches, le besoin alimentaire d'un être humain se situe 1800 et 2700 kcal/j. Dans notre cas, j'ai choisi 2500 kcal/j.

## Le nombre théorique de personnes qui pourraient être nourries (Produits végétaux)

```
vegetaux = dispo_alimentaire.off[dispo_alimentaire.off['Origine'] == "vegetale",]
dispo_kcal_veg<- vegetaux$Disponibilité.alimentaire..Kcal.personne.jour.*vegetaux$Valeur * 365
Veg_kcal_total<-sum(dispo_kcal_veg)
total_kcal_veg = round(sum(Veg_kcal_total)/(2500*365))

print(paste("le nombre de personnes qui pourraient etre nourries :", total_kcal_veg))
```

```
## [1] "le nombre de personnes qui pourraient etre nourries : 6904305685"
```

```
print(paste("Proportion :", total_kcal_veg*100/sum(population[population['Année'] == 2017,"Valeur"]), "%"))
```

```
## [1] "Proportion : 91.470363184701 %"
```

## Calculer la part qui est attribuée à l'alimentation animale, celle qui est perdue et celle qui est concrètement utilisée pour l'alimentation humaine

```
dispo_int<-sum(dispo_alimentaire.off$Disponibilité.intérieure)

for(i in 1:3){
  elt = c('Aliments.pour.animaux', 'Pertes', 'Nourriture')[i]
  print(paste("Proportion de", elt, ":", sum(dispo_alimentaire.off[elt])*100/dispo_int, "%"))
}
```

```
## [1] "Proportion de Aliments.pour.animaux : 13.2320901934029 %"
## [1] "Proportion de Pertes : 4.64645974846534 %"
## [1] "Proportion de Nourriture : 49.3688210318405 %"
```

## Creation d'une table contenant que les informations liés aux céréales

```
cereales = c("Blé et produits", "Riz et produits", "Orge et produits", "Maïs et produits", "Seigle et produits","Avoine", "Mill
et et produits", "Sorgho et produits", "Céréales, Autres")

base_cereales = dispo_alimentaire.off[dispo_alimentaire.off$Produit %in% cereales,]
```

D'après mes recherches, la liste des céréales est la suivante:Blé et produits, Riz et produits, Orge et produits, Maïs et produits,Seigle et produits, Avoine,Millet et produits, Sorgho et produits, Céréales, Autres.

## Calcul de la proportion d'alimentation animale (PAA)

```
PAA<-sum(base_cereales$Aliments.pour.animaux)*100/sum(base_cereales$Disponibilité.intérieure)  
  
print(paste("Proportion d'alimentation animale :", PAA, "%"))
```

```
## [1] "Proportion d'alimentation animale : 69.6702191835914 %"
```

## Calcul de la proportion d'alimentation humaine

```
PAH<-sum(base_cereales$Nourriture)*100/sum(base_cereales$Disponibilité.intérieure)  
  
print(paste("Proportion d'alimentation humaine :", PAH, "%"))
```

```
## [1] "Proportion d'alimentation humaine : 17.629197667404 %"
```

## Exemple d'utilisation du manioc par la thailande

```
thailande_manioc = dispo_alimentaire.off[(dispo_alimentaire.off['Produit'] == "Manioc") & (dispo_alimentaire.off['Zone'] == "Th  
aïlande"),]  
  
Proportion_exportée<-thailande_manioc[1, 'Exportations...Quantité']*100 / thailande_manioc[1,'Production']  
  
print(paste('Proportion de manioc exportée : ', Proportion_exportée, "%"))
```

```
## [1] "Proportion de manioc exportée : 83.4127299192801 %"
```

## Calcul de la proportion de personnes en sous-nutrition en thailande

```
thai_sous_nutrition = dfprop[dfprop['Zone'] == 'Thaïlande',]
Prop_sous_thai<-thai_sous_nutrition[1,'Val']*100/thai_sous_nutrition[1,'Valeur']
print(paste('Proportion de personne en sous nutrition en thaïlande :', Prop_sous_thai, "%"))
```

```
## [1] "Proportion de personne en sous nutrition en thaïlande : 8.95826762130975 %"
```

## Les pays pour lesquels la proportion de personnes sous-alimentées est la plus forte en 2017

```
proportion_sous_nutrition = (dfprop$Val) * 100 / (dfprop$Valeur)
dfprop<-cbind(dfprop,proportion_sous_nutrition)
order = order(-dfprop$proportion_sous_nutrition)
Propor_sous<-head(dfprop[order, c('Zone', "proportion_sous_nutrition")],10)
view(Propor_sous)
```

## Les pays qui ont le plus bénéficié depuis 2013

```
benefices<-aide_alimentaire[!is.na(aide_alimentaire$Valeur),]
aggregate(benefices$Valeur, by=list(benefices$Pays.bénéficiaire),FUN=sum)
```

Group.1 <chr>	x <dbl>
Afghanistan	185452000
Algérie	81114000
Angola	5014000
Bangladesh	348188000
Bénin	22224000
Bhoutan	2666000

**Group.1**

&lt;chr&gt;

**x**

&lt;dbl&gt;

Bolivie (État plurinational de)

6000

Burkina Faso

64812000

Burundi

77318000

Cambodge

25780000

1-10 of 76 rows

Previous 1 2 3 4 5 6 ... 8 Next

```

benef<-benefices %>%
  group_by(Pays.bénéficiaire) %>%
  summarize(aide_kg=sum(Valeur,na.rm = TRUE))
SORT_benef<- order(benef$aide_kg,decreasing = TRUE)
benef[SORT_benef,]

```

**Pays.bénéficiaire**

&lt;chr&gt;

**aide\_kg**

&lt;dbl&gt;

République arabe syrienne

1858943000

Éthiopie

1381294000

Yémen

1206484000

Soudan du Sud

695248000

Soudan

669784000

Kenya

552836000

Bangladesh

348188000

Somalie

292678000

République démocratique du Congo

288502000

Niger

276344000

1-10 of 76 rows

Previous 1 2 3 4 5 6 ... 8 Next

head(benef[*SORT\_benef*,],10)

<b>Pays.bénéficiaire</b> <chr>	<b>aide_kg</b> <dbl>
République arabe syrienne	1858943000
Éthiopie	1381294000
Yémen	1206484000
Soudan du Sud	695248000
Soudan	669784000
Kenya	552836000
Bangladesh	348188000
Somalie	292678000
République démocratique du Congo	288502000
Niger	276344000
1-10 of 10 rows	

## Les pays ayant le plus,le moins de disponibilité/habitant

```
Dispo_pays = dispo_alimentaire.off[c('Zone','Disponibilité.alimentaire..Kcal.personne.jour.')]
aggregate(Dispo_pays$Disponibilité.alimentaire..Kcal.personne.jour., by=list(Dispo_pays$Zone),FUN=sum)
```

<b>Group.1</b> <chr>	<b>x</b> <dbl>
Afghanistan	2087



<b>Group.1</b>	<b>x</b>
<chr>	<dbl>
Afrique du Sud	3020
Albanie	3188
Algérie	3293
Allemagne	3503
Angola	2474
Antigua-et-Barbuda	2416
Arabie saoudite	3255
Argentine	3226
Arménie	2924
1-10 of 172 rows	Previous 1 2 3 4 5 6 ... 18 Next

```
Dispo_total_pays<-Dispo_pays %>%
  group_by(Zone) %>%
  summarize(Disponibilité_alimentaire..Kcal.personne.jour=sum(Disponibilité.alimentaire..Kcal.personne.jour.,na.rm = TRUE))

SORT_dispo<- order(Dispo_total_pays$Disponibilité_alimentaire..Kcal.personne.jour,decreasing = TRUE)
Dispo_total_pays[SORT_dispo,]
```

<b>Zone</b>	<b>Disponibilité_alimentaire..Kcal.personne.jour</b>
<chr>	<dbl>
Autriche	3770
Belgique	3737
Turquie	3708
États-Unis d'Amérique	3682

Zone <chr>	Disponibilité_alimentaire..Kcal.personne.jour <dbl>
Israël	3610
Irlande	3602
Italie	3578
Luxembourg	3540
Égypte	3518
Allemagne	3503
1-10 of 172 rows	Previous 1 2 3 4 5 6 ... 18 Next

```
head(Dispo_total_pays[SORT_dispo,],10)
```

Zone <chr>	Disponibilité_alimentaire..Kcal.personne.jour <dbl>
Autriche	3770
Belgique	3737
Turquie	3708
États-Unis d'Amérique	3682
Israël	3610
Irlande	3602
Italie	3578
Luxembourg	3540
Égypte	3518
Allemagne	3503
1-10 of 10 rows	

```
tail(Dispo_total_pays[Sort_dispo,],10)
```

Zone <chr>	Disponibilité_alimentaire..Kcal.personne.jour <dbl>
Timor-Leste	2129
Ouganda	2126
Zimbabwe	2113
Tchad	2109
République populaire démocratique de Corée	2093
Haïti	2089
Afghanistan	2087
Madagascar	2056
Zambie	1924
République centrafricaine	1879
1-10 of 10 rows	