

# Projet 3 :

## Réalisez une étude de santé publique

# Qu'est-ce que la faim ?

Faim ( Nom féminin ) : Sensation qui, normalement, accompagne le besoin de manger.



# Sous-alimentation



# Mal-nutrition



# Sur-alimentation



# 12 CSV différents


- Population.csv
- question\_2.csv
- Bilan\_alimentaire.csv
- bilan\_alimentaire\_vegetaux.csv
- sous\_nutrition\_2013.csv
- sous\_nutrition\_2015.csv
- cereales\_animaux.csv
- prot\_monde.csv
- question\_11.csv
- dispo\_alim\_cereales\_monde.csv
- dispo\_produit\_animaux.csv
- viande.csv

# Population.csv

PAYS

RÉGIONS

GROUPES SPÉCIAUX



Q

Filtrer les résultats

☐ Monde + (Total)

☒ Monde > (Liste)

☐ Afrique + (Total)

☐ Afrique > (Liste)

☐ -- Afrique de l'Est + (Total)

☐ -- Afrique de l'Est > (Liste)

Tout sélectionner

Tout effacer

Monde > (Liste) ×

PRODUITS

Q

Filtrer les résultats par ex. population-estimations

☒ Population-Estimations

Tout sélectionner

Tout effacer

Population-Estimations ×

ÉLÉMENTS

Q

Filtrer les résultats par ex. population totale

☒ Population totale

☐ Hommes

☐ Femmes

☐ Population rurale

☐ Population urbaine

Tout sélectionner

Tout effacer

Population totale ×

ANNÉES

ANNÉES PRÉVUES

Q

Filtrer les résultats

☒ 2017

☒ 2016

☒ 2015

☒ 2014

☒ 2013

☐ 2012

Tout sélectionner

Tout effacer

2013 × 2014 × 2015 × 2016 × 2017 ×

# question\_2.csv

Bilans Alimentaires

TÉLÉCHARGER LES DONNÉES

VISUALISER LES DONNÉES

MÉTADONNÉES

RAPPORT

PAYS

RÉGIONS

GROUPES SPÉCIAUX

Q fra

☒ France

☐ Polynésie française

Tout sélectionner

Tout effacer

France x

PRODUITS

GROUPES DE PRODUITS

Q Filtrer les résultats

☐ Total General + (Total)

☒ Total General > (Liste)

☐ Produits Vegetaux + (Total)

☐ Produits Vegetaux > (Liste)

☐ Produits Animaux + (Total)

☐ Produits Animaux > (Liste)

Tout sélectionner

Tout effacer

Total General > (Liste) x

ÉLÉMENTS

Q Filtrer les résultats par ex. population totale

☒ Pertes

☒ Transformés

☒ Autres Utilisations

☒ Nourriture

☐ Disponibilité alimentaire en quantité (kg/personne...

☐ Disponibilité alimentaire (Kcal/personne/jour)

Tout sélectionner

Tout effacer

Aliments pour animaux x

Autres Utilisations x

Disponibilité intérieure x

Exportations - Quantité x

Importations - Quantité x

Nourriture x

Pertes x

Production - Quantité x

Semences x

ANNÉES

Q Filtrer les résultats par ex. 2013

☒ 2013

☐ 2012

☐ 2011

☐ 2010

☐ 2009

☐ 2008

Tout sélectionner

Tout effacer

2013 x



# Bilan\_alimentaire.csv

Bilans Alimentaires

TÉLÉCHARGER LES DONNÉES

VISUALISER LES DONNÉES

MÉTADONNÉES

RAPPORT

PAYS

RÉGIONS

GROUPES SPÉCIAUX

⚙️

Q Filtre les résultats

☐ Monde + (Total)

☒ Monde > (Liste)

☐ Afrique + (Total)

☐ Afrique > (Liste)

☐ -- Afrique de l'Est + (Total)

☐ -- Afrique de l'Est > (Liste)

Tout sélectionner

Tout effacer

Monde > (Liste) x

PRODUITS

GROUPES DE PRODUITS

Q Filtre les résultats

☐ Total General + (Total)

☒ Total General > (Liste)

☐ Produits Vegetaux + (Total)

☐ Produits Vegetaux > (Liste)

☐ Produits Animaux + (Total)

☐ Produits Animaux > (Liste)

Tout sélectionner

Tout effacer

Total General > (Liste) x

ÉLÉMENTS

Q Filtre les résultats par ex. population totale

☐ Pertes

☐ Transformés

☐ Autres Utilisations

☐ Nourriture

☐ Disponibilité alimentaire en quantité (kg/personne...

☒ Disponibilité alimentaire (Kcal/personne/jour)

Tout sélectionner

Tout effacer

Disponibilité alimentaire (Kcal/personne/jour) x

ANNÉES

Q Filtre les résultats par ex. 2013

☒ 2013

☐ 2012

☐ 2011

☐ 2010

☐ 2009

☐ 2008

Tout sélectionner

Tout effacer

2013 x

Type d'output

Séparateur milles (montrer données)

Options de formatage de l'output

☒ Table ☐ Pivot

☒ Rien ☐ Virgule ☐ Période

☒ Symboles ☒ Codes ☒ Unités ☐ Values nulles

## bilan\_alimentaire\_vegetaux.csv

# Bilans Alimentaires

TÉLÉCHARGER LES DONNÉES    VISUALISER LES DONNÉES    MÉTADONNÉES    RAPPORT

PAYS RÉGIONS GROUPE SPÉCIAUX

Filtrer les résultats

- ☐ Monde + (Total)
- ☒ Monde > (Liste)
- ☐ Afrique + (Total)
- ☐ Afrique > (Liste)
- ☐ -- Afrique de l'Est + (Total)
- ☐ -- Afrique de l'Est > (Liste)

Tout sélectionner    Tout effacer

Monde > (Liste) x

ÉLÉMENTS

Filtrer les résultats par ex. population totale

- ☒ Autres Utilisations
- ☒ Nourriture
- ☒ Disponibilité alimentaire en quantité (kg/personne...)
- ☒ Disponibilité alimentaire (Kcal/personne/jour)
- ☒ Disponibilité de protéines en quantité (g/personne...)
- ☒ Disponibilité de matière grasse en quantité (g/per...)

Tout sélectionner    Tout effacer

Aliments pour animaux x    Autres Utilisations x

Disponibilité alimentaire (Kcal/personne/jour) x

Disponibilité alimentaire en quantité (kg/personne/an) x

Disponibilité de matière grasse en quantité

PRODUITS GROUPE DE PRODUITS

Filtrer les résultats

- ☐ Total General + (Total)
- ☐ Total General > (Liste)
- ☐ Produits Vegetaux + (Total)
- ☒ Produits Vegetaux > (Liste)
- ☐ Produits Animaux + (Total)
- ☐ Produits Animaux > (Liste)

Tout sélectionner    Tout effacer

Produits Vegetaux > (Liste) x

ANNÉES


Filtrer les résultats par ex. 2013

- ☒ 2013
- ☐ 2012
- ☐ 2011
- ☐ 2010
- ☐ 2009
- ☐ 2008


Tout sélectionner    Tout effacer

2013 x

# sous\_nutrition\_2013.csv

 Données de la sécurité alimentaire

[TÉLÉCHARGER LES DONNÉES](#) [VISUALISER LES DONNÉES](#) [MÉTADONNÉES](#)

PAYS RÉGIONS GROUPES SPÉCIAUX 

Q Filtrer les résultats

☐ Monde + (Total)

☒ Monde > (Liste)

☐ Afrique + (Total)

☐ Afrique > (Liste)

☐ -- Afrique de l'Est + (Total)

☐ -- Afrique de l'Est > (Liste)

Tout sélectionner

Tout effacer

Monde > (Liste) ×

PRODUITS

Q Filtrer les résultats par ex. disponibilité > (liste)

☐ -- Prévalence de la sous-alimentation (%) (moyenne...)

☒ -- Nombre de personnes sous-alimentées (millions)...

☐ -- Prévalence de l'insécurité alimentaire grave - po...

☐ -- Nombre de personnes en situation d'insécurité a...

☐ -- Ampleur du déficit alimentaire (Kcal/personne/jo...

☐ Stabilité > (Liste)

☐ Taux de dépendance à l'égard des importations...

Tout sélectionner

Tout effacer

-- Nombre de personnes sous-alimentées (millions)  
(moyenne sur 3 ans) ×

ÉLÉMENTS

Q Filtrer les résultats par ex. valeur

☒ Valeur

Tout sélectionner

Tout effacer

Valeur ×

ANNÉES

Q Filtrer les résultats par ex. 2016 / 2015-2017

☐ 2016 / 2015-2017

☐ 2015 / 2014-2016

☐ 2014 / 2013-2015

☒ 2013 / 2012-2014

☐ 2012 / 2011-2013


☐ 2011 / 2010-2012

Tout sélectionner


Tout effacer

2013 / 2012-2014 ×

# sous\_nutrition\_2015.csv

 Données de la sécurité alimentaire

[TÉLÉCHARGER LES DONNÉES](#) [VISUALISER LES DONNÉES](#) [MÉTADONNÉES](#)

PAYS RÉGIONS GROUPES SPÉCIAUX 

🔍 Filtrer les résultats

☐ Monde + (Total)

☒ Monde > (Liste)

☐ Afrique + (Total)

☐ Afrique > (Liste)

☐ -- Afrique de l'Est + (Total)

☐ -- Afrique de l'Est > (Liste)

Tout sélectionner

Tout effacer

Monde > (Liste) ✕

PRODUITS

🔍 Filtrer les résultats par ex. disponibilité > (liste)

☐ -- Prévalence de la sous-alimentation (%) (moyenne...)

☒ -- Nombre de personnes sous-alimentées (millions)...

☐ -- Prévalence de l'insécurité alimentaire grave - po...

☐ -- Nombre de personnes en situation d'insécurité a...

☐ -- Ampleur du déficit alimentaire (Kcal/personne/jo...

☐ Stabilité > (Liste)

☐ Taux de dépendance à l'égard des importations...

Tout sélectionner

Tout effacer

-- Nombre de personnes sous-alimentées (millions)  
(moyenne sur 3 ans) ✕

ÉLÉMENTS

🔍 Filtrer les résultats par ex. valeur

☒ Valeur

Tout sélectionner

Tout effacer

Valeur ✕

ANNÉES

🔍 Filtrer les résultats par ex. 2016 / 2015-2017

☐ 2016 / 2015-2017

☒ 2015 / 2014-2016

☐ 2014 / 2013-2015

☐ 2013 / 2012-2014

☐ 2012 / 2011-2013

☐ 2011 / 2010-2012

Tout sélectionner

Tout effacer

2015 / 2014-2016 ✕

# cereales\_animaux.csv

Bilans Alimentaires

TÉLÉCHARGER LES DONNÉES

VISUALISER LES DONNÉES

MÉTADONNÉES

RAPPORT

PAYS

RÉGIONS

GROUPES SPÉCIAUX

Q

Filtrer les résultats par ex. afghanistan

☐ Monde + (Total)

☒ Monde > (Liste)

☐ Afrique + (Total)

☐ Afrique > (Liste)

☐ -- Afrique de l'Est + (Total)

☐ -- Afrique de l'Est > (Liste)

Tout sélectionner

Tout effacer

Monde > (Liste) x

PRODUITS

GROUPES DE PRODUITS

Q

Filtrer les résultats par ex. population

☐ Produits Vegetaux + (Total)

☐ Produits Vegetaux > (Liste)

☐ Produits Animaux + (Total)

☐ Produits Animaux > (Liste)

☐ Céréales - Excl Bière + (Total)

☒ Céréales - Excl Bière > (Liste)

Tout sélectionner

Tout effacer

Céréales - Excl Bière > (Liste) x

ÉLÉMENTS

Q

Filtrer les résultats par ex. population totale

☐ Variation de stock

☐ Exportations - Quantité

☐ Disponibilité intérieure

☒ Aliments pour animaux

☐ Semences

☐ Pertes

Tout sélectionner

Tout effacer

Aliments pour animaux x

Disponibilité de protéines en quantité (g/personne/jour) x

ANNÉES

Q

Filtrer les résultats par ex. 2013

☒ 2013

☐ 2012

☐ 2011

☐ 2010

☐ 2009

☐ 2008

Tout sélectionner

Tout effacer

2013 x

# prot\_monde.csv

Bilans Alimentaires

TÉLÉCHARGER LES DONNÉES

VISUALISER LES DONNÉES

MÉTADONNÉES

RAPPORT

PAYS

RÉGIONS

GROUPES SPÉCIAUX

⚙

Q Filtrer les résultats par ex. afghanistan

☐ Monde + (Total)

☒ Monde > (Liste)

☐ Afrique + (Total)

☐ Afrique > (Liste)

☐ -- Afrique de l'Est + (Total)

☐ -- Afrique de l'Est > (Liste)

Tout sélectionner

Tout effacer

Monde > (Liste) x

PRODUITS

GROUPES DE PRODUITS

Q Filtrer les résultats par ex. population

☐ Total General + (Total)

☒ Total General > (Liste)

☐ Produits Vegetaux + (Total)

☐ Produits Vegetaux > (Liste)

☐ Produits Animaux + (Total)

☐ Produits Animaux > (Liste)

Tout sélectionner

Tout effacer

Total General > (Liste) x

ÉLÉMENTS

Q Filtrer les résultats par ex. population totale

☐ Autres Utilisations

☐ Nourriture

☐ Disponibilité alimentaire en quantité (kg/personne...

☐ Disponibilité alimentaire (Kcal/personne/jour)

☒ Disponibilité de protéines en quantité (g/personne...

☐ Disponibilité de matière grasse en quantité (g/per...

Tout sélectionner

Tout effacer

Disponibilité de protéines en quantité (g/personne/jour) x

ANNÉES

Q Filtrer les résultats par ex. 2013

☒ 2013

☐ 2012

☐ 2011

☐ 2010

☐ 2009


☐ 2008

Tout sélectionner


Tout effacer

2013 x

# question\_11.csv

 Bilans Alimentaires

[TÉLÉCHARGER LES DONNÉES](#) [VISUALISER LES DONNÉES](#) [MÉTADONNÉES](#) [RAPPORT](#)

PAYS RÉGIONS GROUPES SPÉCIAUX 

Q Filtrer les résultats

☐ Monde + (Total)

☒ Monde > (Liste)

☐ Afrique + (Total)

☐ Afrique > (Liste)

☐ - Afrique de l'Est + (Total)

☐ - Afrique de l'Est > (Liste)

Tout sélectionner

Tout effacer

Monde > (Liste) ×

PRODUITS GROUPES DE PRODUITS

Q Filtrer les résultats

☐ Total General + (Total)

☒ Total General > (Liste)

☐ Produits Vegetaux + (Total)

☐ Produits Vegetaux > (Liste)

☐ Produits Animaux + (Total)

☐ Produits Animaux > (Liste)

Tout sélectionner

Tout effacer

Total General > (Liste) ×

ÉLÉMENTS

Q Filtrer les résultats par ex. population totale

☐ Autres Utilisations

☐ Nourriture

☒ Disponibilité alimentaire en quantité (kg/personne/...

☐ Disponibilité alimentaire (Kcal/personne/jour)

☐ Disponibilité de protéines en quantité (g/personne/...

☐ Disponibilité de matière grasse en quantité (g/pers...

Tout sélectionner

Tout effacer

Disponibilité alimentaire en quantité (kg/personne/an) ×

ANNÉES

Q Filtrer les résultats par ex. 2013

☒ 2013

☐ 2012

☐ 2011

☐ 2010

☐ 2009


☐ 2008

Tout sélectionner

Tout effacer

2013 ×

# dispo\_alim\_cereales\_monde.csv


 Bilans Alimentaires

[TÉLÉCHARGER LES DONNÉES](#) [VISUALISER LES DONNÉES](#) [MÉTADONNÉES](#) [RAPPORT](#)

PAYS

RÉGIONS

GROUPES SPÉCIAUX



Q

Filtrer les résultats par ex. afghanistan

☐ Monde + (Total)

☒ Monde > (Liste)

☐ Afrique + (Total)

☐ Afrique > (Liste)

☐ -- Afrique de l'Est + (Total)

☐ -- Afrique de l'Est > (Liste)

Tout sélectionner

Tout effacer

Monde > (Liste) ×

PRODUITS

GROUPES DE PRODUITS

Q

Filtrer les résultats par ex. population

☐ Produits Animaux > (Liste)

☐ Céréales - Excl Bière + (Total)

☒ Céréales - Excl Bière > (Liste)

☐ Racines Amyl + (Total)

☐ Racines Amyl > (Liste)

☐ Cultures Sucreries + (Total)

Tout sélectionner

Tout effacer

Céréales - Excl Bière > (Liste) ×

ÉLÉMENTS

Q

Filtrer les résultats par ex. population totale

☒ Disponibilité intérieure

☐ Aliments pour animaux

☐ Semences

☐ Pertes

☐ Transformés

☐ Autres Utilisations

Tout sélectionner

Tout effacer

Disponibilité intérieure ×

ANNÉES

Q

Filtrer les résultats par ex. 2013

☒ 2013

☐ 2012

☐ 2011

☐ 2010

☐ 2009

☐ 2008


Tout sélectionner

Tout effacer

2013 ×



# dispo\_produit\_animaux.csv


 Bilans Alimentaires

[TÉLÉCHARGER LES DONNÉES](#) [VISUALISER LES DONNÉES](#) [MÉTADONNÉES](#) [RAPPORT](#)

PAYS

RÉGIONS

GROUPES SPÉCIAUX



Q

Filtrer les résultats par ex. afghanistan

☐ Monde + (Total)

☒ Monde > (Liste)

☐ Afrique + (Total)

☐ Afrique > (Liste)

☐ -- Afrique de l'Est + (Total)

☐ -- Afrique de l'Est > (Liste)

Tout sélectionner

Tout effacer

Monde > (Liste) ×

PRODUITS

GROUPES DE PRODUITS

Q

Filtrer les résultats par ex. population

☒ Produits Animaux > (Liste)

☐ Céréales - Excl Bière + (Total)

☐ Céréales - Excl Bière > (Liste)

☐ Racines Amyl + (Total)

☐ Racines Amyl > (Liste)

☐ Cultures Sucrières + (Total)

Tout sélectionner

Tout effacer

Produits Animaux > (Liste) ×

ÉLÉMENTS

Q

Filtrer les résultats par ex. population totale

☐ Aliments pour animaux

☐ Semences

☐ Pertes

☐ Transformés

☐ Autres Utilisations

☐ Nourriture

Tout sélectionner

Tout effacer

Disponibilité intérieure ×

ANNÉES

Q

Filtrer les résultats par ex. 2013

☒ 2013

☐ 2012

☐ 2011

☐ 2010

☐ 2009


☐ 2008

Tout sélectionner


Tout effacer

2013 ×

# viande.csv

 Bilans Alimentaires

[TÉLÉCHARGER LES DONNÉES](#) [VISUALISER LES DONNÉES](#) [MÉTADONNÉES](#) [RAPPORT](#)

PAYS RÉGIONS GROUPES SPÉCIAUX 

Q

 Filtrer les résultats par ex. afghanistan

☐ Monde + (Total)

☒ Monde > (Liste)

☐ Afrique + (Total)

☐ Afrique > (Liste)

☐ -- Afrique de l'Est + (Total)

☐ -- Afrique de l'Est > (Liste)

Tout sélectionner

Tout effacer

Monde > (Liste) ×

PRODUITS GROUPES DE PRODUITS

Q

 Filtrer les résultats par ex. population

☒ Viande > (Liste)

☐ Abats + (Total)

☐ Abats > (Liste)

☐ Graisses animales + (Total)

☐ Graisses animales > (Liste)

☐ Oeufs + (Total)

☐ Oeufs > (Liste)

Tout sélectionner

Tout effacer

Viande > (Liste) ×

ÉLÉMENTS

Q

 Filtrer les résultats par ex. population totale

☐ Aliments pour animaux

☐ Semences

☐ Pertes

☐ Transformés

☐ Autres Utilisations

☐ Nourriture

Tout sélectionner

Tout effacer

Disponibilité intérieure ×

ANNÉES

Q

 Filtrer les résultats par ex. 2013

☒ 2013

☐ 2012

☐ 2011

☐ 2010

☐ 2009

☐ 2008

Tout sélectionner

Tout effacer

2013 ×

# Les opérations d'algèbre relationnelle utilisées

- Import de csv



```
import pandas as pd
import numpy as np
from math import *

data = pd.read_csv("C:/Users/KyRun69/Downloads/population.csv")
df=pd.DataFrame(data)

pop_2013=df[df["Année"]==2013]
pop_2013=pop_2013[pop_2013["Code zone"]!=351]
valeur_pop_2013= int(1000*(pop_2013["Valeur"].sum()))
print("En 2013 il y avait ", valeur_pop_2013, "habitants sur terre")

pop_2017=df[df["Année"]==2017]
pop_2017=pop_2017[pop_2017["Code zone"]!=351]
valeur_pop_2017= int(1000*(pop_2017["Valeur"].sum()))
print("En 2017 il y avait ", valeur_pop_2017, "habitants sur terre")
```

# Les opérations d'algèbre relationnelle utilisées

- Suppression de colonnes



```
aliments_animaux=pd.merge(aliments_animaux,proteines,how="left")
aliments_animaux=aliments_animaux.drop(columns="Disponibilité alimentaire en quantité")
aliments_animaux=aliments_animaux.drop(columns="Protéines")
aliments_animaux["Grammes de protéines"]=aliments_animaux.Valeur*aliments_animaux["Ratio Proteines/Poids"]

ratio_ultime_kg=aliments_animaux.Valeur.sum()/df_dispo_produit_animaux.Valeur.sum()
ratio_ultime_Kcal=aliments_animaux.Kcal.sum()/df_dispo_kcal.Valeur.sum()
ratio_ultime_protéines=aliments_animaux["Grammes de protéines"].sum()/df_dispo_prot.Valeur.sum()
print("Le ratio de Disponibilité de céréales destinée à l'alimentation animale / Disponibilité animale destinée à l'alimentation ")
print("\nLe ratio de Disponibilité de céréales destinée à l'alimentation animale / Disponibilité animale destinée à l'alimentation ")
print("\nLe ratio de Disponibilité de céréales destinée à l'alimentation animale / Disponibilité animale destinée à l'alimentation ")
```

# Les opérations d'algèbre relationnelle utilisées

- Création de colonnes




```
aliments_animaux=pd.merge(aliments_animaux,proteines,how="left")
aliments_animaux=aliments_animaux.drop(columns="Disponibilité alimentaire en quantité")
aliments_animaux=aliments_animaux.drop(columns="Protéines")
aliments_animaux["Grammes de protéines"]=aliments_animaux.Valeur*aliments_animaux["Ratio Proteines/Poids"]

ratio_ultime_kg=aliments_animaux.Valeur.sum()/df_dispo_produit_animaux.Valeur.sum()
ratio_ultime_Kcal=aliments_animaux.Kcal.sum()/df_dispo_kcal.Valeur.sum()
ratio_ultime_protéines=aliments_animaux["Grammes de protéines"].sum()/df_dispo_prot.Valeur.sum()
print("Le ratio de Disponibilité de céréales destinée à l'alimentation animale / Disponibilité animale destinée à l'alimentation ")
print("\nLe ratio de Disponibilité de céréales destinée à l'alimentation animale / Disponibilité animale destinée à l'alimentation ")
print("\nLe ratio de Disponibilité de céréales destinée à l'alimentation animale / Disponibilité animale destinée à l'alimentation ")
```

# Les opérations d'algèbre relationnelle utilisées

- Renommer les colonnes




```
prot=pd.read_csv("C:/Users/KyRun69/Downloads/prot_monde.csv")
proteines=pd.DataFrame(prot)
proteines=pd.merge(proteines,pop_2013,how="left")
print(proteines)
proteines["Valeur"]=(proteines.Valeur*proteines.Population)
proteines=proteines.drop(columns="Population")
proteines=proteines.groupby("Produit").Valeur.sum()
proteines=proteines.reset_index()
proteines=proteines.rename(index=str, columns={"Valeur": "Protéines"})

dispoo=pd.read_csv("C:/Users/KyRun69/Downloads/question_11.csv")
df_dispoo=pd.DataFrame(dispoo)
df_dispoo=pd.merge(df_dispoo,pop_2013,how="left")
df_dispoo["Valeur"]=df_dispoo.Valeur*df_dispoo.Population/365
df_dispoo=df_dispoo.groupby("Produit").Valeur.sum()
df_dispoo=df_dispoo.reset_index()
df_dispoo=df_dispoo.rename(columns={"Valeur": "Disponibilité alimentaire en quantité"})
```

# Les opérations d'algèbre relationnelle utilisées

- Restriction



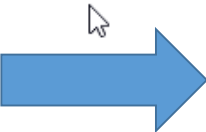
```
data = pd.read_csv("C:/Users/KyRun69/Downloads/population.csv")
df=pd.DataFrame(data)

pop_2013=df[df["Année"]==2013]
pop_2013=pop_2013[pop_2013["Code zone"]!=351]
valeur_pop_2013= int(1000*(pop_2013["Valeur"].sum()))
print("En 2013 il y avait ", valeur_pop_2013, "habitants sur terre")

pop_2017=df[df["Année"]==2017]
pop_2017=pop_2017[pop_2017["Code zone"]!=351]
valeur_pop_2017= int(1000*(pop_2017["Valeur"].sum()))
```

# Les opérations d'algèbre relationnelle utilisées

- Somme



```
data = pd.read_csv("C:/Users/KyRun69/Downloads/population.csv")
df=pd.DataFrame(data)


pop_2013=df[df["Année"]==2013]
pop_2013=pop_2013[pop_2013["Code zone"]!=351]
valeur_pop_2013= int(1000*(pop_2013["Valeur"].sum()))
print("En 2013 il y avait ", valeur_pop_2013, "habitants sur terre")

pop_2017=df[df["Année"]==2017]
pop_2017=pop_2017[pop_2017["Code zone"]!=351]
valeur_pop_2017= int(1000*(pop_2017["Valeur"].sum()))
```



# Les opérations d'algèbre relationnelle utilisées

- Agrégation + somme



```
In [17]: #Question 10 A l'aide de la table donnant les ratios énergie/poids de chaque aliment (que vous avez établie précédemment), conver

cereales_animaux=pd.read_csv("C:/Users/KyRun69/Downloads/cereales_animaux.csv")
df_cereales_animaux=pd.DataFrame(cereales_animaux)
aliments_animaux=df_cereales_animaux[df_cereales_animaux["Code Élément"]==5521]
aliments_animaux=aliments_animaux.groupby("Produit").Valeur.sum()
aliments_animaux=aliments_animaux.reset_index()
aliments_animaux=pd.merge(aliments_animaux,Table_ratio,how="left")
aliments_animaux.Valeur=aliments_animaux.Valeur*1000000
aliments_animaux["Kcal"]=aliments_animaux.Valeur*aliments_animaux.Ratio

#print(df_cereales_animaux)
print(aliments_animaux)
```

# Les opérations d'algèbre relationnelle utilisées

- Agrégation + moyenne

*#Question 4 calculez pour chaque produit le ratio "énergie/poids", que vous donnerez en kcal/kg.*

```
df_transition=df_bilan[df_bilan["Code Élément"]==645]
df_transition=df_transition.reset_index(drop=True)
valeur_transition={"Pays":df_transition.Pays.tolist(),"Code Produit":df_transition["Code Produit"].tolist(),"Disponibilité alimentaire":df_transition["Disponibilité alimentaire"].tolist()}
df_transition=pd.DataFrame(valeur_transition)

df_dispo=pd.merge(df_dispo,df_transition,how="left")
df_dispo=df_dispo.replace(0,np.nan)
df_dispo=df_dispo.dropna(axis=0, how='any')
df_dispo["Ratio"]=(df_dispo["Disponibilité alimentaire (Kcal)"])/((df_dispo["Disponibilité alimentaire en quantité"])*(df_dispo.Poids))
ratio={"Code produit":df_dispo["Code Produit"].tolist(),"Produit":df_dispo.Produit.tolist(),"Pays":df_dispo.Pays.tolist(),"Ratio":df_dispo.Ratio.tolist()}
df_ratio=pd.DataFrame(ratio)
df_ratio=df_ratio.replace(0,np.nan)
df_ratio=df_ratio.dropna(axis=0, how='any')
df_ratio=df_ratio.reset_index(drop=True)

Table_ratio=df_ratio.groupby("Produit").mean()
Table_ratio=Table_ratio.reset_index()

print(Table_ratio)
```



# Les opérations d'algèbre relationnelle utilisées

- Projection


In [10]: *# Question 6 Calculez, pour les produits végétaux uniquement, la disponibilité intérieure mondiale exprimée en kcal.*



```
data_2 = pd.read_csv("C:/Users/KyRun69/Downloads/bilan_alimentaire_vegetaux.csv")
df_vegetaux=pd.DataFrame(data_2)
df_dispo2=df_vegetaux[df_vegetaux["Code Élément"]==5301]
df_dispo2=df_dispo2.replace(0,np.nan)
df_dispo2=df_dispo2.dropna(axis=0, how='any')
df_dispo2=df_dispo2.reset_index(drop=True)
disponibilite={"Produit":df_dispo2.Produit.tolist(),"Valeur_kg":df_dispo2.Valeur.tolist()}
df_disponibilite=pd.DataFrame(disponibilite)
df_disponibilite.Valeur_kg=df_disponibilite.Valeur_kg*1000000
df_disponibilite=df_disponibilite.groupby("Produit").Valeur_kg.sum()
```

# Les opérations d'algèbre relationnelle utilisées

- Jointure



```
In [17]: #Question 10 A l'aide de la table donnant les ratios énergie/poids de chaque aliment (que vous avez établie précédemment), conver

cereales_animaux=pd.read_csv("C:/Users/KyRun69/Downloads/cereales_animaux.csv")
df_cereales_animaux=pd.DataFrame(cereales_animaux)
aliments_animaux=df_cereales_animaux[df_cereales_animaux["Code Élément"]==5521]
aliments_animaux=aliments_animaux.groupby("Produit").Valeur.sum()
aliments_animaux=aliments_animaux.reset_index()
aliments_animaux=pd.merge(aliments_animaux,Table_ratio,how="left")
aliments_animaux.Valeur=aliments_animaux.Valeur*1000000
aliments_animaux["Kcal"]=aliments_animaux.Valeur*aliments_animaux.Ratio

#print(df_cereales_animaux)
print(aliments_animaux)
```

# Les opérations d'algèbre relationnelle utilisées

- Replace, dropna, reset\_index

*#Question 4 calculez pour chaque produit le ratio "énergie/poids", que vous donnerez en kcal/kg.*

```
df_transition=df_bilan[df_bilan["Code Élément"]==645]
df_transition=df_transition.reset_index(drop=True)
valeur_transition={"Pays":df_transition.Pays.tolist(),"Code Produit":df_transition["Code Produit"].tolist(),"Disponibilité alimen"
df_transition=pd.DataFrame(valeur_transition)

df_dispo=pd.merge(df_dispo,df_transition,how="left")
df_dispo=df_dispo.replace(0,np.nan)
df_dispo=df_dispo.dropna(axis=0, how='any')
df_dispo["Ratio"]=(df_dispo["Disponibilité alimentaire (Kcal)"])/((df_dispo["Disponibilité alimentaire en quantité"])*(df_dispo.P
ratio={"Code produit":df_dispo["Code Produit"].tolist(),"Produit":df_dispo.Produit.tolist(),"Pays":df_dispo.Pays.tolist(),"Ratio"
df_ratio=pd.DataFrame(ratio)
df_ratio=df_ratio.replace(0,np.nan)
df_ratio=df_ratio.dropna(axis=0, how='any')
df_ratio=df_ratio.reset_index(drop=True)

Table_ratio=df_ratio.groupby("Produit").mean()
Table_ratio=Table_ratio.reset_index()

print(Table_ratio)
```

# Réponses aux questions

- En termes de disponibilité intérieure de produits végétaux, combien d'humains pourraient être nourris si toute la production végétale était utilisée pour de la nourriture ?

```
In [11]: #Question 7 : En termes de disponibilité intérieure de produits végétaux, combien d'humains pourraient être nourris si toute la p

besoin_pop_moyen=365*2550
#print(besoin_pop_moyen)
print(str(int(somme_dispo_vegetaux/besoin_pop_moyen)) + " humains pourraient être nourris si toute la production végétale était u

14761333091 humains pourraient être nourris si toute la production végétale était utilisée pour de la nourriture
```

# Réponses aux questions

- En termes de disponibilité intérieure de produits végétaux, combien d'humains pourraient être nourris si la production végétale, mis à part ses utilisations en *Semence*, *Traitement* et *Autres utilisation*, était utilisée pour de la nourriture ?

```
df_vegetaux_sauf_traitement=df_vegetaux[df_vegetaux["Code Élément"]==5131]
df_vegetaux_sauf_traitement.Valeur=df_vegetaux_sauf_traitement.Valeur*1000000
df_vegetaux_sauf_traitement=df_vegetaux_sauf_traitement.groupby("Produit").Valeur.sum()
df_vegetaux_sauf_traitement=df_vegetaux_sauf_traitement.reset_index()
df_vegetaux_sauf_traitement=pd.merge(df_vegetaux_sauf_traitement,Table_ratio,how="left")
df_vegetaux_sauf_traitement["Kcal"]=df_vegetaux_sauf_traitement.Valeur*df_vegetaux_sauf_traitement.Ratio
df_vegetaux_sauf_traitement=df_vegetaux_sauf_traitement.replace([np.inf], np.nan)
df_vegetaux_sauf_traitement=df_vegetaux_sauf_traitement.dropna(axis=0,how="any")

valeur_dispo_vegetaux=somme_dispo_vegetaux-(df_vegetaux_sauf_semences.Kcal.sum()+df_vegetaux_sauf_autre.Kcal.sum()+df_vegetaux_sauf_traitement.Kcal.sum())
print("La disponibilité intérieure mondiale des produits végétaux en omettant les catégories semences, traitement et autre est de",valeur_dispo_vegetaux)
print("Il serait alors possible de nourrir " + str(int(valeur_dispo_vegetaux/besoin_pop_moyen)) + " humains")
```

```
La disponibilité intérieure mondiale des produits végétaux en omettant les catégories semences, traitement et autre est de 9501
992609133666.0 Kcal
Il serait alors possible de nourrir 10208963318 humains
```

# Réponses aux questions

- Quelle proportion de la population mondiale est considérée comme étant en sous-nutrition ?

```
pop_2015=df[df["Année"]==2015]
tableau={"Pays":pop_2015.Zone.tolist(),"Population":pop_2015.Valeur.tolist()}
pop_2015=pd.DataFrame(tableau)
pop_2015.Population=pop_2015.Population*1000

nutrition_2015_2={"Pays":df_nutrition_2015.Zone.tolist(),"million de personnes sous-alimentées":df_nutrition_2015.Valeur.tolist()}
df_nutrition_2015_2=pd.DataFrame(nutrition_2015_2)
pourcentage_sous_nutrition_2015=pd.merge(df_nutrition_2015_2,pop_2015,how="left")
pourcentage_sous_nutrition_2015.fillna(0,inplace=True)
#print(pourcentage_sous_nutrition_2015)
pourcentage_2015=(pourcentage_sous_nutrition_2015.Population.sum())/(pourcentage_sous_nutrition_2015["million de personnes sous-a.
print("Il y avait " +str(pourcentage_2015)+ " % de sous-nutris en 2015")
```

```
Il y avait 9.948509841947393 % de sous-nutris en 2013
Il y avait 10.203172996987954 % de sous-nutris en 2015
```



# Réponses aux questions

- Quelle proportion (en terme de poids) de la disponibilité mondiale de céréales est destinée à l'alimentation animale ?

```
In [27]: # Question 12 : quelle proportion (en terme de poids) de la disponibilité mondiale de céréales est destinée à l'alimentation animale ?

cereales_monde=pd.read_csv("C:/Users/KyRun69/Downloads/dispo_alim_cereales_monde.csv")
df_cereales_monde=pd.DataFrame(cereales_monde)

df_cereales_monde["Kg"]=df_cereales_monde.Valeur*1000000
dispo_alim_cereales_monde=df_cereales_monde.Kg.sum()
dispo_animaux_cereales=aliments_animaux.Valeur.sum()
#print(dispo_animaux_cereales)
#print(dispo_alim_cereales_monde)

print(str((dispo_animaux_cereales/dispo_alim_cereales_monde)*100) + "% de la disponibilité mondiale de céréales est destinée à l'alimentation animale")
```

36.291456706047654% de la disponibilité mondiale de céréales est destinée à l'alimentation animale

# Réponses aux questions

- Combien de kg de céréales sont nécessaires pour produire 1kg de viande ?

In [25]: *#Question : Combien de kg de céréales sont nécessaires pour produire 1kg de viande ?*

```
viande=pd.read_csv("C:/Users/KyRun69/Downloads/viande.csv")
df_viande=pd.DataFrame(viande)
df_viande1=df_viande.Valeur.sum()*1000000
print("Pour produire 1kg de viande il faut " + str((aliments_animaux.Valeur.sum())/(df_viande.Valeur.sum()*1000000)) + "kg de céréales")
```

Pour produire 1kg de viande il faut 2.858921867994122kg de céréales.

# Export des données vers une base de données

- Lecture CSV



```
import pandas as pd
import numpy as np

data = pd.read_csv("C:/Users/KyRun69/Downloads/sql_population.csv")
df=pd.DataFrame(data)
pop={"Pays":df.Zone.tolist(),"Code Pays":df["Code zone"].tolist(),"Année":df.Année.tolist(),"Population":df.Valeur.tolist()}
df_pop=pd.DataFrame(pop)
df_pop.Population=df_pop.Population*1000

df_pop=df_pop.rename(columns={"Année":"Annee"})
df_pop.to_csv("population_sql.csv", sep='\t', encoding='utf-8', index = False)
print(df_pop)
```

# Export des données vers une base de données

- Opérations



```
import pandas as pd
import numpy as np

data = pd.read_csv("C:/Users/KyRun69/Downloads/sql_population.csv")
df=pd.DataFrame(data)
pop={"Pays":df.Zone.tolist(),"Code Pays":df["Code zone"].tolist(),"Année":df.Année.tolist(),"Population":df.Valeur.tolist()}
df_pop=pd.DataFrame(pop)
df_pop.Population=df_pop.Population*1000

df_pop=df_pop.rename(columns={"Année":"Annee"})
df_pop.to_csv("population_sql.csv", sep='\t', encoding='utf-8', index = False)
print(df_pop)
```

# Export des données vers une base de données

- Implantation des colonnes demandées



```
import pandas as pd
import numpy as np

data = pd.read_csv("C:/Users/KyRun69/Downloads/sql_population.csv")
df=pd.DataFrame(data)
pop={"Pays":df.Zone.tolist(),"Code Pays":df["Code zone"].tolist(),"Année":df.Année.tolist(),"Population":df.Valeur.tolist()}
df_pop=pd.DataFrame(pop)
df_pop.Population=df_pop.Population*1000

df_pop=df_pop.rename(columns={"Année":"Annee"})
df_pop.to_csv("population_sql.csv", sep='\t', encoding='utf-8', index = False)
print(df_pop)
```

# Export des données vers une base de données

- Export



```
import pandas as pd
import numpy as np

data = pd.read_csv("C:/Users/KyRun69/Downloads/sql_population.csv")
df=pd.DataFrame(data)
pop={"Pays":df.Zone.tolist(),"Code Pays":df["Code zone"].tolist(),"Année":df.Année.tolist(),"Population":df.Valeur.tolist()}
df_pop=pd.DataFrame(pop)
df_pop.Population=df_pop.Population*1000

df_pop=df_pop.rename(columns={"Année":"Annee"})
df_pop.to_csv("population_sql.csv", sep='\t', encoding='utf-8', index = False)
print(df_pop)
```

# Export des données vers une base de données

```
In [1]: import pandas as pd
import numpy as np


data = pd.read_csv("C:/Users/KyRun69/Downloads/sql_population.csv")
df=pd.DataFrame(data)
pop={"Pays":df.Zone.tolist(),"Code Pays":df["Code zone"].tolist(),"Année":df.Année.tolist(),"Population":df.Valeur.tolist()}
df_pop=pd.DataFrame(pop)
df_pop.Population=df_pop.Population*1000

df_pop=df_pop.rename(columns={"Année":"Annee"})
df_pop.to_csv("population_sql.csv", sep='\t', encoding='utf-8', index = False)
print(df_pop)
```


**Clé primaire** : code\_pays ; année

# Export des données vers une base de données


- Lecture csv



```
data2 = pd.read_csv("C:/Users/KyRun69/Downloads/pop10-13.csv")
df_pop10=pd.DataFrame(data2)
pop10={"Pays":df_pop10.Zone.tolist(),"Année":df_pop10.Année.tolist(),"Population":df_pop10.Valeur.tolist()}
df_pop10=pd.DataFrame(pop10)
df_pop10.Population=df_pop10.Population*1000
```



```
animaux=pd.read_csv("C:/Users/KyRun69/Downloads/sql_animaux.csv")
df_animaux=pd.DataFrame(animaux)
```



```
vegetaux=pd.read_csv("C:/Users/KyRun69/Downloads/sql_vegetaux.csv")
df_vegetaux=pd.DataFrame(vegetaux)
```



# Export des données vers une base de données

- Opérations (1)

```
df_animaux["nature"]="animal"  
df_vegetaux["nature"]="vegetal"  
df_concat1=pd.merge(df_animaux,df_vegetaux,how="outer")
```

Création de la nouvelle  
colonne exigée

```
df_concat=df_concat1[df_concat1["Code Élément"]==645]  
df_concat=df_concat.rename(columns = {"Valeur":"dispo_alim_tonnes"})  
df_concat=pd.merge(df_concat,df_pop10,how="left")  
df_concat["dispo_alim_tonnes"]=df_concat["dispo_alim_tonnes"]*df_concat["Population"]*365  
df_concat=df_concat.reset_index(drop=True)
```

```
df_011=df_concat1[df_concat1["Code Élément"]==664]  
test={"Pays":df_011.Pays.tolist(),"Produit":df_011.Produit.tolist(),"Année":df_011.Année.tolist(),"dispo_alim_kcal_p_j":df_011.Valeur.tolist()}  
df_01=pd.DataFrame(test)
```

```
df_022=df_concat1[df_concat1["Code Élément"]==674]  
test2={"Pays":df_022.Pays.tolist(),"Produit":df_022.Produit.tolist(),"Année":df_022.Année.tolist(),"dispo_prot":df_022.Valeur.tolist()}  
df_02=pd.DataFrame(test2)
```

```
df_033=df_concat1[df_concat1["Code Élément"]==684]  
test3={"Pays":df_033.Pays.tolist(),"Produit":df_033.Produit.tolist(),"Année":df_033.Année.tolist(),"dispo_matiere_grasse":df_033.Valeur.tolist()}  
df_03=pd.DataFrame(test3)
```

# Export des données vers une base de données

- Opérations (1)

```
df_animaux["nature"]="animal"
df_vegetaux["nature"]="vegetal"
df_concat1=pd.merge(df_animaux,df_vegetaux,how="outer")

df_concat=df_concat1[df_concat1["Code Élément"]==645]
df_concat=df_concat.rename(columns = {"Valeur":"dispo_alim_tonnes"})
df_concat=pd.merge(df_concat,df_pop10,how="left")
df_concat["dispo_alim_tonnes"]=df_concat["dispo_alim_tonnes"]*df_concat["Population"]*365
df_concat=df_concat.reset_index(drop=True)
```

Calcul de la  
colonne dispo\_alim\_tonnes

```
df_011=df_concat1[df_concat1["Code Élément"]==664]
test={"Pays":df_011.Pays.tolist(),"Produit":df_011.Produit.tolist(),"Année":df_011.Année.tolist(),"dispo_alim_kcal_p_j":df_011.Valeur.tolist()}
df_01=pd.DataFrame(test)

df_022=df_concat1[df_concat1["Code Élément"]==674]
test2={"Pays":df_022.Pays.tolist(),"Produit":df_022.Produit.tolist(),"Année":df_022.Année.tolist(),"dispo_prot":df_022.Valeur.tolist()}
df_02=pd.DataFrame(test2)

df_033=df_concat1[df_concat1["Code Élément"]==684]
test3={"Pays":df_033.Pays.tolist(),"Produit":df_033.Produit.tolist(),"Année":df_033.Année.tolist(),"dispo_matiere_grasse":df_033.Valeur.tolist()}
df_03=pd.DataFrame(test3)
```

# Export des données vers une base de données

- Opérations (1)

```
df_animaux["nature"]="animal"
df_vegetaux["nature"]="vegetal"
df_concat1=pd.merge(df_animaux,df_vegetaux,how="outer")

df_concat=df_concat1[df_concat1["Code Élément"]==645]
df_concat=df_concat.rename(columns = {"Valeur":"dispo_alim_tonnes"})
df_concat=pd.merge(df_concat,df_pop10,how="left")
df_concat["dispo_alim_tonnes"]=df_concat["dispo_alim_tonnes"]*df_concat["Population"]*365
df_concat=df_concat.reset_index(drop=True)

df_011=df_concat1[df_concat1["Code Élément"]==664]
test={"Pays":df_011.Pays.tolist(),"Produit":df_011.Produit.tolist(),"Année":df_011.Année.tolist(),"dispo_alim_kcal_p_j":df_011.Valeur.tolist()}
df_01=pd.DataFrame(test)

df_022=df_concat1[df_concat1["Code Élément"]==674]
test2={"Pays":df_022.Pays.tolist(),"Produit":df_022.Produit.tolist(),"Année":df_022.Année.tolist(),"dispo_prot":df_022.Valeur.tolist()}
df_02=pd.DataFrame(test2)

df_033=df_concat1[df_concat1["Code Élément"]==684]
test3={"Pays":df_033.Pays.tolist(),"Produit":df_033.Produit.tolist(),"Année":df_033.Année.tolist(),"dispo_matiere_grasse":df_033.Valeur.tolist()}
df_03=pd.DataFrame(test3)
```

Création de trois  
DataFrames  
contenant chacun  
une des colonnes  
exigées

# Export des données vers une base de données

- Opérations (2)

```
df_concat=pd.merge(df_concat,df_01,how="left")  
df_concat=pd.merge(df_concat,df_02,how="left")  
df_concat=pd.merge(df_concat,df_03,how="left")
```

} Jointure des 3 nouveaux  
DataFrames au premier

```
df_concat=df_concat.drop(columns=["Code Domaine","Domaine","Code Élément", "Élément", "Symbole","Code Année","Unité","Description"]  
df_concat=df_concat.rename(columns = {"Année":"Annee"})  
df_concat.to_csv("dispo_alim_sql.csv", sep='\t', encoding='utf-8', index = False)
```

# Export des données vers une base de données

- Implantation des colonnes demandées

```
df_concat=pd.merge(df_concat,df_01,how="left")  
df_concat=pd.merge(df_concat,df_02,how="left")  
df_concat=pd.merge(df_concat,df_03,how="left")  
  
df_concat=df_concat.drop(columns=["Code Domaine","Domaine","Code Élément", "Élément", "Symbole","Code Année","Unité","Description"])  
df_concat=df_concat.rename(columns = {"Année":"Annee"})  
df_concat.to_csv("dispo_alim_sql.csv", sep='\t', encoding='utf-8', index = False)
```



# Export des données vers une base de données

- Export

```
df_concat=pd.merge(df_concat,df_01,how="left")
df_concat=pd.merge(df_concat,df_02,how="left")
df_concat=pd.merge(df_concat,df_03,how="left")

df_concat=df_concat.drop(columns=["Code Domaine","Domaine","Code Élément", "Élément", "Symbole","Code Année","Unité","Description"])
df_concat=df_concat.rename(columns = {"Année":"Annee"})
df_concat.to_csv("dispo_alim_sql.csv", sep='\t', encoding='utf-8', index = False)
```



# Export des données vers une base de données

- Clé primaire

```
df_concat=pd.merge(df_concat,df_01,how="left")
df_concat=pd.merge(df_concat,df_02,how="left")
df_concat=pd.merge(df_concat,df_03,how="left")

df_concat=df_concat.drop(columns=["Code Domaine","Domaine","Code Élément", "Élément", "Symbole","Code Année","Unité","Description"])
df_concat=df_concat.rename(columns = {"Année":"Annee"})
df_concat.to_csv("dispo_alim_sql.csv", sep='\t', encoding='utf-8', index = False)
```

**Clé primaire** : code\_pays ; année ; code produit

# Requêtes SQL

```
SELECT Pays, Annee, SUM(dispo_prot)/1000 AS Dispo_Prot
FROM dispo_alim
WHERE Annee="2013"
GROUP BY Pays, Annee
ORDER BY Dispo_Prot DESC
LIMIT 10;
```

	Pays	Annee	Dispo_Prot
1	Islande	2013	0.13233
2	Chine - RAS de Hong-Kong	2013	0.128360000000000006
3	Israël	2013	0.128000000000000003
4	Lituanie	2013	0.124339999999999998
5	Maldives	2013	0.121850000000000003
6	Finlande	2013	0.117090000000000007
7	Luxembourg	2013	0.113340000000000001
8	Monténégro	2013	0.11147
9	Pays-Bas	2013	0.111460000000000002
10	Albanie	2013	0.111240000000000003



# Requêtes SQL

```
SELECT Pays, Annee, SUM(dispo_alim_kcal_p_j) AS Dispo_Kcal
FROM dispo_alim
WHERE Annee="2013"
GROUP BY Pays, Annee
ORDER BY Dispo_Kcal DESC
LIMIT 10;
```

	Pays	Annee	Dispo_Kcal
1	Autriche	2013	3767
2	Belgique	2013	3737
3	Turquie	2013	3708
4	États-Unis d'Amérique	2013	3682
5	Israël	2013	3610
6	Irlande	2013	3602
7	Italie	2013	3578
8	Luxembourg	2013	3530
9	Égypte	2013	3513
10	Allemagne	2013	3503

# Requêtes SQL

```
SELECT Pays, Annee, SUM(dispo_prot)/1000 AS Dispo_Prot
FROM dispo_alim
WHERE Annee="2013"
GROUP BY Pays, Annee
ORDER BY Dispo_Prot
LIMIT 10;
```

	Pays	Annee	Dispo_Prot
1	Libéria	2013	0.0376
2	Guinée-Bissau	2013	0.04395999999999999
3	Mozambique	2013	0.04567999999999999
4	République centrafricaine	2013	0.04602999999999999
5	Madagascar	2013	0.046679999999999985
6	Haïti	2013	0.047689999999999975
7	Zimbabwe	2013	0.04825
8	Congo	2013	0.05136
9	Ouganda	2013	0.052629999999999996
10	Sao Tomé-et-Principe	2013	0.052800000000000001

# Requêtes SQL

```
SELECT Pays, Annee, SUM(dispo_alim_kcal_p_j) AS Dispo_Kcal
FROM dispo_alim
WHERE Annee="2013"
GROUP BY Pays, Annee
ORDER BY Dispo_Kcal
LIMIT 10;
```

	Pays	Annee	Dispo_Kcal
1	République centrafricaine	2013	1879
2	Zambie	2013	1923
3	Madagascar	2013	2056
4	Afghanistan	2013	2086
5	Haïti	2013	2089
6	République populaire démocratique de Corée	2013	2093
7	Tchad	2013	2108
8	Zimbabwe	2013	2111
9	Timor-Leste	2013	2119
10	Ouganda	2013	2126

# Requêtes SQL

```
5 SELECT sous_nutrition.pays,(sous_nutrition.nb_personnes)/(population.Population)*100 AS Ratio
6 FROM sous_nutrition JOIN population
7 ON sous_nutrition.code_pays=population."Code Pays"
8 WHERE (sous_nutrition.Annee="2012-2014" AND population.Annee="2013")
9 ORDER BY (sous_nutrition.nb_personnes)/(population.Population)*100 DESC
10 LIMIT 10;
```

	pays	Ratio
1	République centrafricaine	51.11508672896781
2	Haïti	49.847581433693136
3	Zambie	47.51471636354038
4	République populaire démocratique de Corée	41.223084927559434
5	Libéria	39.6613388268176
6	Zimbabwe	37.198179946195495
7	Tchad	36.54748126941585
8	Madagascar	35.27705554176411
9	Rwanda	35.245634963669886
10	Ouganda	33.5519536024413

# Requêtes SQL

```
19 SELECT Produit, "Code Produit", (SUM(autres_utilisations)/SUM(dispo_int)) AS Ratio
20 FROM equilibre_prod
21 WHERE Annee="2013"
22 GROUP BY "Code Produit"
23 ORDER BY Ratio DESC
24 LIMIT 10
```

	Produit	Code Produit	Ratio
1	Alcool, non Comestible	2659	1.0012878892415251
2	Huil Plantes Oleif Autr	2586	0.7547440599585393
3	Huile de Palmistes	2576	0.7039573820395738
4	Huile de Palme	2577	0.6981094645797378
5	Girofles	2642	0.6461538461538462
6	Huile de Colza&Moutarde	2574	0.551289716433208
7	Graisses Animales Crue	2737	0.4702191625637227
8	Huiles de Poissons	2781	0.44806763285024154
9	Huile de Soja	2571	0.41617215986272965
10	Plantes Aquatiques	2775	0.3792986329285896

# Quelles sont les "autres utilisations" possibles ?

- Biocarburants / bioénergie

Exemple : huile de colza, huile de palme



# Quelles sont les "autres utilisations" possibles ?

- Biochimie

Produits ménagers ( exemple : citron )

Produit corporels ( beurre de charité )

Produits cosmétiques ( banane )

Alcool ( riz )

Produits pharmaceutiques ( eucalyptus )

-[...]





# Quelles sont les "autres utilisations" possibles ?

- Bio-matériaux

- construction ( soja )
- automobile ( colza )
- plastique ( maïs )
- [...]





# URL où les images ont été empruntées

- <http://www.bio-carburant.info/avantages-des-biocarburants>
- <https://pixabay.com/en/oilseed-rape-rape-blossom-2090139/>
- <https://www.lanutrition.fr/bien-dans-son-assiette/les-nutriments/lipides/la-verite-sur-lhuile-de-palme>
- [https://www.setal.net/44-Incroyables-utilisations-du-citron\\_a41558.html](https://www.setal.net/44-Incroyables-utilisations-du-citron_a41558.html)
- [https://www.bhphotovideo.com/c/product/1070383-REG/makerbot\\_mp06572\\_1\\_75mm\\_pla\\_filament\\_large.html](https://www.bhphotovideo.com/c/product/1070383-REG/makerbot_mp06572_1_75mm_pla_filament_large.html)
- <http://www.comment-perdredupoids.com/coupe-faim-naturels-efficaces-satiete/>
- <http://resistance-verte.over-blog.com/2017/10/une-personne-sur-neuf-a-faim.html>
- <http://guineezenith.com/pourquoi-malnutrition-a-sigui-le-directeur-prefectoral-de-la-sante-sexplique/>
- <http://www.e-sante.fr/plan-national-nutrition-sante-vaincre-malbouffe-obesite/actualite/1172>