

# Projet3\_FA0\_Python SQL

Parcours Data Analyst

Xuefei ZHANG\_2021

# Sommaire

- ❏ Mise en contexte
    - Chiffres clés
    - Causes de la faim
  - ❏ Décorticage des causes et prévision à l'appui de calculs
  - ❏ Tables principales
    - Dataframes Python
    - Opérations d'algèbre relationnelle
  - ❏ Q19: requêtes SQL et résultats
  - ❏ Q20: exploration autres utilisations
-

# Contexte - chiffres clés

- En 2013, la population du monde entier est de **6,997,326,000**
- En 2013, la population en **sous-nutrition** est 743,800,000  
soit **10.63** % de la population mondiale de l'année

Nourriture  
Aliments pour animaux  
Pertes  
Semences  
Traitement  
Autres utilisations (non alimentaire)

**Disponibilité  
intérieure**

= Production + Importations  
- Exportations - Variation de stock

# Contexte - chiffres clés et causes

9. (en théorie) combien d'humains pourraient être nourris avec la **disponibilité alimentaire mondiale** ?

Niveau calorique (2500kcal/per/jour): 8070616278, soit **115.34 %**

Niveau protéine(50g/per/jour) : 11356117507, soit **162.29 %**

Basé sur la Q9, on peut en conclure que en 2013, la masse de **disponibilité alimentaire mondiale** dépasse le besoin calorique et protéine de la population mondiale par **15% et 62%**.

Ainsi on peut supposer que le problème sous-nutrition résulte plutôt d'un **problème commercial** que d'un manque de production.

# Contexte - chiffres clés et causes

Toute la disponibilité intérieure mondiale de  
produits végétaux à la nourriture

216%

273%

disponibilité intérieure mondiale  
- (Alimentation animale + Pertes)

148%

168%

Calorie  
Protéine

disponibilité intérieure mondiale  
- Alimentation animale

140%

157%



Malgré la perte, alimentation animale joue un rôle important en occupant **une importante partie des utilités de produits végétaux**, cela résonne avec Q11 où l'alimentation animale prend **45.9%** de céréales.

Et “**pertes**” joue un rôle relativement minable par rapport à alimentation animaux.

# Contexte - chiffres clés et causes

11. En prenant en compte seulement la production céréale alimentaire (humaine et animale), **45.9%** de céréales sont pour alimentation des animaux, d'où **54.1%** pour alimentation humaine.

13. Si les USA diminuaient leur production de produits animaux de 10% , on pourrait libérer: 14,009,600 tonnes de céréales

14. **83.41 %** de la production manioc en Thaïlande est exportée, alors que **8.36 %** de la population Thaïlandaise est en sous nutrition

=> sous-nutrition et pays exporteur de manioc

=> R1 pour faim:

surproduction des produits animaux, qui consomment bcp de produits végétaux (dont céréales)

# déséquilibre des usages

(=> solution: on pourrait **diminuer la production animale** pour libérer les produits végétaux (ou céréales) à la nourriture)

# [Décorticage de cause commerciale]

## A.Produits exportés par pays sous-nutri vs Importations mondiales

Les 15 produits les plus exportés par les pays en sous-nutrition:

Maïs  
Huile de Palme  
Manioc  
Blé  
Riz (Eq Blanchi)  
Sucre Eq Brut  
Légumes, Autres  
Bananes  
Soja  
Lait - Excl Beurre  
Fruits, Autres  
Poissons Pelagiques  
Tomates  
Orge  
Pommes

Les 15 plus grandes importations du monde

'Blé',  
'Maïs',  
'Soja',  
'Lait - Excl Beurre',  
'Sucre Eq Brut',  
'Manioc',  
'Huile de Palme',  
'Légumes, Autres',  
'Orge',  
'Fruits, Autres',  
'Riz (Eq Blanchi)',  
'Poissons Pelagiques',  
'Bananes',  
'Pommes',  
'Tomates'

### Observation:

**superposition** des produits exportés par pays sous-nutrition et produits les plus importés du monde.

Autrement dit, les pays sous-nutrition exportent vers d'autres pays du monde les produits alimentaires.

### Supposition:

produits exportés sont sous forme de **matière première**

**R2.1: Export important des produits alimentaires par pays sous-nutrition**

#commerciale

## B. Produits exportés vs Produits riches en calorie et protéine

15 produits les plus exportés par pays en sous-nutrition:	Q5. les 15 aliments les plus caloriques (kcal/kg) :	les 5 aliments les plus riches en protéines:
Maïs Huile de Palme Manioc Blé Riz (Eq Blanchi) Sucre Eq Brut Légumes, Autres Bananes Soja Lait - Excl Beurre Fruits, Autres Poissons Pelagiques Tomates Orge Pommes	Huiles de Foie de Poisson Huile de Son de Riz Huile de Sésame Huiles de Poissons Huile d'Arachide Huile de Germe de Maïs Huile d'Olive Huile Plantes Oleif Autr Huile de Palmistes Huile Graines de Coton Huile de Palme Huile de Tournesol Huile de Soja Huile de Colza&Moutarde Huile de Coco	Graines Colza/Moutarde Soja Arachides Decortiquees Pois Légumineuses Autres Haricots Sésame Abats Comestible Viande, Autre Plantes Oleiferes, Autre Aliments pour enfants Viande de Bovins Cephalopodes Viande d'Ovins/Caprins Viande de Volailles

### Observation:

au moins 5 des 15 produits les plus exportés par pays sous-nutrition sont aussi classés les aliments les plus caloriques.

R2.2 : malgré la production intérieure élevée, les pays en sous-nutrition exportent vers d'autres pays leurs **produits riches en calorie**  
#commerciale



## C. Produits exportés vs proportion des autres utilisations

### 15 produits les plus exportés par les pays en sous-nutrition:

Maïs  
Huile de Palme  
Manioc  
Blé  
Riz (Eq Blanchi)  
Sucre Eq Brut  
Légumes, Autres  
Bananes  
Soja  
Lait - Excl Beurre  
Fruits, Autres  
Poissons Pelagiques  
Tomates  
Orge  
Pommes

	10 produits	Ratio Autres utilisations/Dispo intérieure, %
1	Alcool, non Comestible	96.46
2	Plantes Aquatiques	91.89
3	Huile de Palme	65.03
4	Huil Plantes Oleif Austr	54.96
5	Huile de Palmistes	53.5
6	Huile de Colza&Moutarde	45.84
7	Huiles de Poissons	40.37
8	Huile de Coco	36.84
9	Graisses Animales Crue	30.58
10	Manioc	23.06

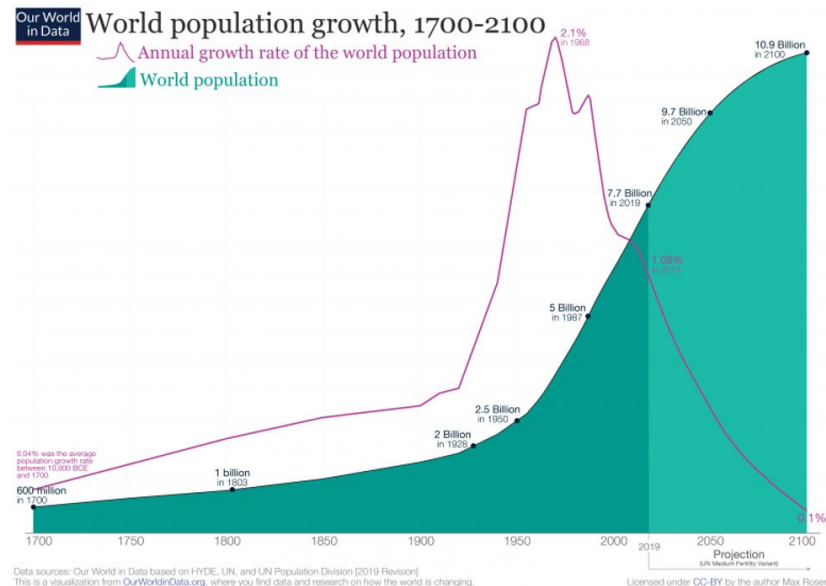
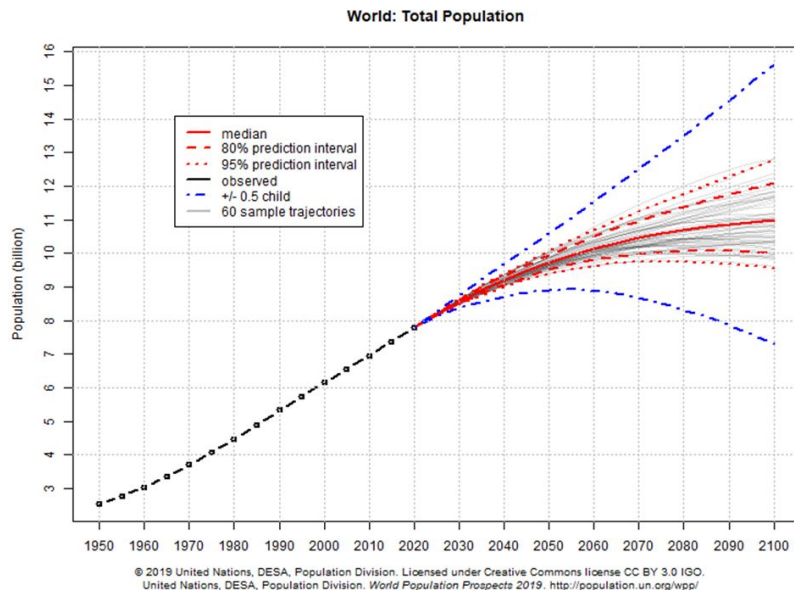
### Observation:

3 parmi les 15 produits les plus exportés par pays sous-nutrition connaissent des **proportions élevées en Autres utilisations**/Disponibilité intérieure: 65%, 40%, 23% (usage non-alimentaire)

R3: les pays sous-nutrition exportent leur produits agricoles **qui devraient servir à nourrir leur peuple vers d'autres pays pour d'autres utilisations.**

#utilisation #commerciale

# Prévision de la population 2050



2050: population mondiale prévu **9,7 milliards**, soit **138,76%** fois de la population en 2013.

# 2050: Population vs Production

9. disponibilité alimentaire mondiale en 2013?

Niveau calorique (2500kcal/per/jour): **115%**

Niveau protéine(50g/per/jour) : **162 %**

2050: population mondiale prévu **9,7 milliards**, soit **138%** fois de la population en 2013.

Au niveau de protéine, la disponibilité alimentaire mondiale en 2013 peut couvrir le besoin protéine de la population en 2050 en dépassant le besoin par (138% - 115% = **23%**) ;

Au niveau calorique, il existe un fossé de (**162% - 138% = 24%**) à rattraper pour combler le besoin en 2050.

**Mais ça sous-entend pas l'obligation d'une augmentation drastique de la production alimentaire.**

1. La faim résulte des **problèmes commerciaux et de la politique d'exportation**: exportation de produits agricoles par pays sous-nutrition (répartition mondiale), manque d'industrie agro-alimentaire, déséquilibre de production animale par pays industrialisés ie.Etats-Unis
2. D'ailleurs, on envisage aussi une **croissance de production** au fur et à mesure, et en parallèle, l'**amélioration de la répartition des utilisations** grâce au progrès et innovations technologiques agricole.
3. Disponibilité protéine a la possibilité d'**équibrer** la disponibilité calorique.
4. À noter que 2500 Kcal/jour est un standard pour les hommes de 100kg ( à la différence des enfants, femmes....)

# Tables et opérations

- Dataframes Python
- Détaillez quelques opération d'algèbre relationnelle utilisées sur les dataframes:
  - une agrégation
  - une jointure
  - une restriction

# 5 fichiers CSV en dataframe

```
pop = pd.read_csv("fr_population.csv")
pop.head(1)
```

Code Domaine	Domaine	Code zone	Zone	Code Élément	Élément	Code Produit	Produit	Code année	Année	Unité	Valeur	Symbole	Description du Symbole	
0	FBSH	Bilans Alimentaire (Ancienne méthodologie et p...	2	Afghanistan	511	Population totale	2501	Population	2013	2013	1000 personnes	30552	NaN	Donnée officielle

```
sousnutri = pd.read_csv("fr_sousalimentation.csv")
sousnutri.head(1)
```

Code Domaine	Domaine	Code zone	Zone	Code Élément	Élément	Code Produit	Produit	Code année	Année	Unité	Valeur	Symbole	Description du Symbole	Note	
0	FS	Données de la sécurité alimentaire	2	Afghanistan	6132	Valeur	210011	Nombre de personnes sous-alimentées (millions)...	2012-2014	2012-2014	millions	7.9	F	Estimation FAO	NaN

```
vege = pd.read_csv("fr_vegetaux.csv")
vege.head(1)
```

Code Domaine	Domaine	Code zone	Zone	Code Élément	Élément	Code Produit	Produit	Code année	Année	Unité	Valeur	Symbole	Description du Symbole	
0	FBSH	Bilans Alimentaire (Ancienne méthodologie et p...	2	Afghanistan	5511	Production	2511	Blé	2013	2013	Milliers de tonnes	5169.0	S	Données standardisées

```
animaux = pd.read_csv("fr_animaux.csv")
animaux.head(1)
```

Code Domaine	Domaine	Code zone	Zone	Code Élément	Élément	Code Produit	Produit	Code année	Année	Unité	Valeur	Symbole	Description du Symbole	
0	FBSH	Bilans Alimentaire (Ancienne méthodologie et p...	2	Afghanistan	5511	Production	2731	Viande de Bovins	2013	2013	Milliers de tonnes	134.0	S	Données standardisées

```
cereal = pd.read_csv("fr_céréales.csv")
cereal.head(1)
```

Code Domaine	Domaine	Code zone	Zone	Code Élément	Élément	Code Produit	Produit	Code année	Année	Unité	Valeur	Symbole	Description du Symbole	
0	FBSH	Bilans Alimentaire (Ancienne méthodologie et p...	2	Afghanistan	5511	Production	2511	Blé	2013	2013	Milliers de tonnes	5169	S	Données standardisées

# Population #projection

## Question 1

Calculez le nombre total d'humains sur la planète. Critiquez votre résultat. En cas d'anomalie, analysez et effectuez les corrections nécessaires sur le résultat de votre calcul pour l'année 2013.

```
In [16]: # Importation et traitement de la table population
pop = pd.read_csv("fr_population.csv")

#drop doublons
#pop=pop.drop([pop.index[34],pop.index[35],pop.index[36],pop.index[37], pop.index[118],pop.index[34]])
pop = pop.loc[~pop['Code zone'].isin([96, 128, 41, 214]),:]

# sélection des colonnes pertinentes pour pop et sousalim
pop = pop[['Zone', 'Code zone', 'Année', 'Valeur']]

# Renommage de la colonne Valeur et mise à l'unité d'habitant, au lieu de milliers d'habitants
pop.rename(columns={"Valeur": "Population", 'Année': 'Annee'}, inplace=True)
pop['Population'] *= 1000

pop.head()
```

Out[16]:

	Zone	Code zone	Annee	Population
0	Afghanistan	2	2013	30552000
1	Afrique du Sud	202	2013	52776000
2	Albanie	3	2013	3173000
3	Algérie	4	2013	39208000
4	Allemagne	79	2013	82727000

- Projection
- Renommage

Zone | Zone

# animaux, vege #Concaténation

```
In [20]: animaux = pd.read_csv("fr_animaux.csv")
animaux = animaux.loc[~animaux['Code zone'].isin([96, 128, 41, 214]).]
animaux['origine'] = 'animal'
#to add a column called "origine", with all values of this column "animal"

animaux.head()
```

Out[20]:

	Code Domaine	Domaine	Code zone	Zone	Code Élément	Élément	Code Produit	Produit	Code année	Année	Unité	Valeur	Symbole	Description du Symbole	origine
0	FBSH	Bilans Alimentaire (Ancienne méthodologie et p...	2	Afghanistan	5511	Production	2731	Viande de Bovins	2013	2013	Milliers de tonnes	134.00	S	Données standardisées	animal
1	FBSH	Bilans Alimentaire (Ancienne méthodologie et p...	2	Afghanistan	5611	Importations - Quantité	2731	Viande de Bovins	2013	2013	Milliers de tonnes	6.00	S	Données standardisées	animal
2	FBSH	Bilans Alimentaire (Ancienne méthodologie et p...	2	Afghanistan	5301	Disponibilité intérieure	2731	Viande de Bovins	2013	2013	Milliers de tonnes	140.00	S	Données standardisées	animal
3	FBSH	Bilans Alimentaire (Ancienne méthodologie et p...	2	Afghanistan	5142	Nourriture	2731	Viande de Bovins	2013	2013	Milliers de tonnes	140.00	S	Données standardisées	animal
4	FBSH	Bilans Alimentaire (Ancienne méthodologie et p...	2	Afghanistan	645	Disponibilité alimentaire en quantité (kg/pers...	2731	Viande de Bovins	2013	2013	kg	4.59	Fc	Donnée calculée	animal

```
In [21]: vegeani = pd.concat([animaux, vege])
vegeani.head()
```

Out[21]:

	Code Domaine	Domaine	Code zone	Zone	Code Élément	Élément	Code Produit	Produit	Code année	Année	Unité	Valeur	Symbole	Description du Symbole	origine
--	-----------------	---------	--------------	------	-----------------	---------	-----------------	---------	---------------	-------	-------	--------	---------	---------------------------	---------

- Restriction
- Suppression de redondance
- Ajout de colonne

Zone+Element

Même structure animaux  
et vege => concaténation  
dans un même dataframe

# Ratios: Kcal/kg par produit par pays

## Question 4

A partir de ces dernières informations, et à partir du poids de la disponibilité alimentaire (pour chaque pays et chaque produit), calculez pour chaque produit le ratio "énergie/poids", que vous donnerez en Kcal/kg.

```
In [26]: #Q4 Ratio energie/poids (Kcal/kg)
#!!! Disponibilité alimentaire (Kcal/per/j) = QuantityNourri * Ratio kcal/kg / pop pays / 365
# => Ratio kcal/kg = Kcal/per/j * 365 * pop pays / QuantityNourriture

DispoAliN = vapvtpop[["Zone", "Produit", "origine", "Disponibilité alimentaire (Kcal/personne/jour)", "Nourriture", "Population"]]
DispoAliN.head()

DispoAliN["Ratio kcal/kg"] = DispoAliN["Disponibilité alimentaire (Kcal/personne/jour)"] / (DispoAliN["Nourriture"] * 1000000) * 365 * DispoAliN["Population"]
DispoAliN.head()
```

Out[26]:

	Zone	Produit	origine	Disponibilité alimentaire (Kcal/personne/jour)	Nourriture	Population	Ratio kcal/kg
0	Afghanistan	Abats Comestible	animal	5.0	53.0	30552000	1052.026415
1	Afghanistan	Agrumes, Autres	vegetal	1.0	39.0	30552000	285.935385
2	Afghanistan	Aliments pour enfants	vegetal	1.0	2.0	30552000	5575.740000
3	Afghanistan	Ananas	vegetal	0.0	0.0	30552000	NaN
4	Afghanistan	Bananes	vegetal	4.0	82.0	30552000	543.974634

- Projection
- Ajout de colonnes

Zone+Produit



# #Pivot table #Agrégation #Projection

## Question 6

Calculez, pour les produits végétaux uniquement, la Disponibilité intérieure mondiale exprimée en Kcal.

```
In [30]: vapvtpopVegi = vapvtpop[vapvtpop["origine"]=="vegetal"]
vapvtpopVegi = vapvtpopVegi[["origine","Produit","Disponibilité intérieure"]]
vapvtpopVegi

Dispon = vapvtpopVegi.pivot_table(index=["Produit"], values="Disponibilité intérieure", aggfunc=sum)
Dispon
```

Out[30]:

Disponibilité intérieure	
Produit	
Agrumes, Autres	12177.0
Alcool, non Comestible	21740.0
Aliments pour enfants	86.0
Ananas	23282.0
Arachides Decortiquees	29694.0
Avoine	23407.0
Bananes	103991.0
Bananes plantains	29233.0
Bière	187348.0
Blé	679497.0

Clé après agrégation: Produit

# Q8 perte et nourriture

```
In [34]: #select several columns from vapvtpop

Q8pre=vapvtpop[vapvtpop["origine"]=="vegetal"]
Q8 =Q8pre[["Zone", "Produit", "Aliments pour animaux", "Pertes", "Nourriture"]].groupby("Produit").sum()
Q8.head(20)

# unité pour les 3 : mille tonnes
```

Out[34]:

	Aliments pour animaux	Pertes	Nourriture
Produit			
Agrumes, Autres	0.0	810.0	11360.0
Alcool, non Comestible	0.0	0.0	0.0
Aliments pour enfants	0.0	1.0	762.0
Ananas	0.0	2745.0	20757.0
Arachides Decortiquees	1.0	1307.0	12203.0
Avoine	16251.0	514.0	3904.0
Bananes	871.0	15295.0	85953.0
Bananes plantains	562.0	3222.0	22863.0
Bière	0.0	43.0	187051.0
Blé	129668.0	27530.0	457824.0
Boissons Alcooliques	0.0	0.0	21248.0
Boissons Fermentés	0.0	1117.0	29404.0
Café	0.0	114.0	8101.0
Citrons & Limes	0.0	1417.0	14190.0
Coco (Incl Coprah)	46.0	2846.0	22231.0
Céréales, Autres	19036.0	785.0	5324.0
Dattes	283.0	565.0	6497.0
Edulcorants Autres	32.0	0.0	20818.0
Fève de Cacao	2.0	154.0	4757.0
Fruits, Autres	61.0	18045.0	185474.0

- Restriction
- Agrégation (SUM)

Clé après agrégation: Produit

# #Merge(jointure)

Que how soit 'inner', 'left', 'right' ou 'outer', dans ce cas le résultat dataframe vapvtpop est le même, puisqu'on a déjà cadré les colonnes de pop

```
In [22]: vapvt = vegeani.pivot_table(index=['Zone','Code zone','Produit', 'Code Produit', 'origine','Année'], values='Valeur', columns='Élément')
vapvt = vapvt.rename_axis(None, axis=1)
vapvt.reset_index(inplace=True)
vapvtpop = pd.merge(vapvt, pop[['Zone','Population']], on='Zone', how='inner')
vapvtpop.head(20)
```

	Zone	Code zone	Produit	Code Produit	origine	Année	Aliments pour animaux	Autres utilisations (non alimentaire)	Disponibilité alimentaire (Kcal/personne/jour)	Disponibilité alimentaire en quantité (kg/personne/an)	...	Disponibilité intérieure	Exportations - Quantité	Importations - Quantité	Nourriture
0	Afghanistan	2	Abats Comestible	2736	animal	2013	NaN	NaN	5.0	1.72	...	53.0	NaN	NaN	53.0
1	Afghanistan	2	Agrumes, Autres	2614	vegetal	2013	NaN	NaN	1.0	1.29	...	41.0	2.0	40.0	39.0
2	Afghanistan	2	Aliments pour enfants	2680	vegetal	2013	NaN	NaN	1.0	0.06	...	2.0	NaN	2.0	2.0
3	Afghanistan	2	Ananas	2618	vegetal	2013	NaN	NaN	0.0	0.00	...	0.0	NaN	0.0	0.0
4	Afghanistan	2	Bananes	2615	vegetal	2013	NaN	NaN	4.0	2.70	...	82.0	NaN	82.0	82.0
5	Afghanistan	2	Beurre, Ghee	2740	animal	2013	NaN	NaN	23.0	1.17	...	36.0	NaN	0.0	36.0
6	Afghanistan	2	Bières	2656	vegetal	2013	NaN	NaN	0.0	0.00	...	0.0	NaN	0.0	0.0

Zone + Produit | Zone + Produit

# Q10 sousnutri

## Q10

A partir des données téléchargées qui concernent la sous-nutrition, Quelle proportion de la population mondiale est considérée comme étant en sous-nutrition ?

```
In [148]: sousnutri=pd.read_csv("fr_sousalimentation.csv")
# transformation de la colonne Valeur :
sousnutri["Valeur"] = pd.to_numeric(sousnutri["Valeur"], errors = 'coerce')

# transformation de la colonne Année :
sousnutri["Année"] = pd.to_numeric(sousnutri["Année"].str[0:4]) + 1

sousnutri.head()
```

Out[148]:

	Code Domaine	Domaine	Code zone	Zone	Code Élément	Élément	Code Produit	Produit	Code année	Année	Unité	Valeur	Symbole	Description du Symbole	Note
0	FS	Données de la sécurité alimentaire	2	Afghanistan	6132	Valeur	210011	Nombre de personnes sous-alimentées (millions)...	20122014	2013	millions	7.9	F	Estimation FAO	NaN
1	FS	Données de la sécurité alimentaire	2	Afghanistan	6132	Valeur	210011	Nombre de personnes sous-alimentées (millions)...	20132015	2014	millions	8.8	F	Estimation FAO	NaN
2	FS	Données de la sécurité alimentaire	2	Afghanistan	6132	Valeur	210011	Nombre de personnes sous-alimentées (millions)...	20142016	2015	millions	9.6	F	Estimation FAO	NaN
3	FS	Données de la sécurité alimentaire	2	Afghanistan	6132	Valeur	210011	Nombre de personnes sous-alimentées (millions)...	20152017	2016	millions	10.2	F	Estimation FAO	NaN
4	FS	Données de la sécurité alimentaire	2	Afghanistan	6132	Valeur	210011	Nombre de personnes sous-alimentées (millions)...	20162018	2017	millions	10.6	F	Estimation FAO	NaN

Transformation de types de donnés

Zone + Année | Zone + Année

# Q12.1: ratio animaux/animaux et humains

## Question 12

Parmi les données des bilans alimentaires au niveau mondial, sélectionnez les 200 plus grandes importations de ces produits (1 importation = une quantité d'un produit donné importée par un pays donné)

Groupez ces importations par produit, afin d'avoir une table contenant 1 ligne pour chacun des 15 produits. Ensuite, calculez pour chaque produit les 2 quantités suivantes :

le ratio entre la quantité destinées aux "Autres utilisations" (Other uses) et la disponibilité intérieure.

le ratio entre la quantité destinée à la nourriture animale et la quantité destinée à la nourriture (animale + humaine)

```
In [150]: temp = vapvtpop.loc[vapvtpop['Produit'].isin(list_produitexpo),:]
temp = temp.sort_values('Importations - Quantité', ascending = False)[: 200]

temp = temp[['Produit','Importations - Quantité', 'Nourriture', 'Aliments pour animaux', 'Autres utilisations (non alimentaire)', 'Disponibilité intérieure']]
byproduct = temp.groupby("Produit").sum()

byproduct['Ratio_autres/interieur'] = byproduct['Autres utilisations (non alimentaire)']/byproduct['Disponibilité intérieure']
byproduct['Ratio_animal/human&animal'] = byproduct['Aliments pour animaux']/(byproduct['Nourriture']+byproduct['Aliments pour animaux'])

byproduct.round({'Ratio_autres/interieur':3})

#15plus grandes importations de ces produits
byproduct_impo=byproduct.sort_values('Importations - Quantité', ascending = False)[: 15]
print(byproduct_impo.index)
byproduct_impo
```

```
Index(['Blé', 'Maïs', 'Soja', 'Lait - Excl Beurre', 'Sucre Eq Brut', 'Manioc',
      'Huile de Palme', 'Légumes, Autres', 'Orge', 'Fruits, Autres',
      'Riz (Eq Blanchi)', 'Poissons Pelagiques', 'Bananes', 'Pommes',
      'Tomates'],
      dtype='object', name='Produit')
```

Out[150]:

Produit	Importations - Quantité	Nourriture	Aliments pour animaux	Autres utilisations (non alimentaire)	Disponibilité intérieure	Ratio_autres/interieur	Ratio_animal/human&animal
Blé	150140.0	303720.0	98715.0	15125.0	460327.0	0.032857	0.245294
Maïs	102489.0	55950.0	402887.0	182118.0	701104.0	0.259759	0.878061
Soja	93497.0	7532.0	13710.0	525.0	159593.0	0.003290	0.645419
Lait - Excl Beurre	91023.0	332010.0	29133.0	12479.0	378021.0	0.033011	0.080669

- Sort values
- Restriction
- Projection
- Ajout de colonnes
- Arrondissement

*Produit* + Importations-Quantité

# Q12.2 Ratios calorique et prot

## Question 12

Donnez les 3 produits qui ont la plus grande valeur pour chacun des 2 ratios (vous aurez donc 6 produits à citer)

In [152]: `byproduct.sort_values('Ratio_autres/interieur', ascending = False)[ : 3]`

*#3 produits avec les plus grands ratios autres utilisations/dispo intérieure*

Out[152]:

	Importations - Quantité	Nourriture	Aliments pour animaux	Autres utilisations (non alimentaire)	Disponibilité intérieure	Ratio_autres/interieur	Ratio_animal/human&animal
Produit							
Huile de Palme	31055.0	6922.0	0.0	19747.0	26886.0	0.734471	0.000000
Manioc	34751.0	15268.0	25153.0	20355.0	67341.0	0.302268	0.622276
Maïs	102489.0	55950.0	402887.0	182118.0	701104.0	0.259759	0.878061

Sort values

In [153]: `byproduct.sort_values('Ratio_animal/human&animal', ascending = False)[ : 3]`

Out[153]:

	Importations - Quantité	Nourriture	Aliments pour animaux	Autres utilisations (non alimentaire)	Disponibilité intérieure	Ratio_autres/interieur	Ratio_animal/human&animal
Produit							
Orge	22515.0	365.0	20160.0	12.0	29600.0	0.000405	0.982217
Maïs	102489.0	55950.0	402887.0	182118.0	701104.0	0.259759	0.878061
Poissons Pelagiques	14371.0	4052.0	13097.0	40.0	17190.0	0.002327	0.763718

# Q15

## Q15

Une table appelée population, contenant la population de chaque pays pour 2013. Elle devra contenir 4 colonnes : pays, code\_pays, annee, population. Proposez une clé primaire pertinente pour cette table.

```
In [162]: pop['idx'] = np.arange(len(pop)) #to add a new column in sequential numbers
          cols = pop.columns.tolist()
          cols = cols[-1:] + cols[:-1] # move the last element to the first position/to move element to a later next one
          pop = pop[cols] #reorder the dataframe

          pop.rename(columns={'Zone':'pays','Code zone':'code_pays','Valeur':'population', 'Annee':'annee'}, inplace=True)
          pop.to_csv("pop.csv", index = False)
          pop.head()

          # PK: pays
```

Out[162]:

	idx	pays	code_pays	annee	Population
0	0	Afghanistan	2	2013	30552000
1	1	Afrique du Sud	202	2013	52776000
2	2	Albanie	3	2013	3173000
3	3	Algérie	4	2013	39208000
4	4	Allemagne	79	2013	82727000

pays / code\_pays / idx

```
In [157]: def check_possible_primary_key(pop, columns_pk):
          if len(pop) != len(pop[columns_pk].drop_duplicates()):
              raise Exception("{} can't be a primary key!".format(columns_pk))
```

# Q16

## Q16

Une table appelée `dispo_alim` contenant pour chaque pays et pour chaque produit en 2013, les informations suivantes: la nature du produit (deux valeurs possibles : "animal" ou "végétal") disponibilité alimentaire en tonnes disponibilité alimentaire en Kcal/personne/jour disponibilité alimentaire de protéines en g/personne/jour disponibilité alimentaire de matières grasses en g/personne/jour

Elle devra contenir ces colonnes : `pays`, `code_pays`, `année`, `produit`, `code_produit`, `origin`, `dispo_alim_tonnes`, `dispo_alim_kcal_p_j`, `dispo_prot`, `dispo_mat_gr`

```
In [65]: # sélection des colonnes pertinentes pour pop et sousalim
dispo_alim = vaptvpop[['Zone', 'Code zone', 'Année', 'Produit', 'Code Produit', 'origine', 'Disponibilité alimentaire en quantité (kg/personne/an)', 'Disponibilité alime
dispo_alim['dispo_alim_tonnes'] = dispo_alim['Disponibilité alimentaire en quantité (kg/personne/an)'] * dispo_alim['Population'] / 1000

dispo_alim['idx'] = np.arange(len(dispo_alim)) #to add a new column in sequential numbers

cols = dispo_alim.columns.tolist()
cols = cols[-1:] + cols[:-1] # move the last element to the first position
dispo_alim = dispo_alim[cols] #reorder the dataframe

# Renommage des colonnes
dispo_alim.rename(columns={'Zone': 'pays', 'Code zone': 'code_pays', 'Produit': 'produit', 'Code Produit': 'code_produit', 'Année': 'année', 'origine': 'origin', 'Disponi
# Drop unnecessary columns
dispo_alim = dispo_alim.drop(['Population', 'Disponibilité alimentaire en quantité (kg/personne/an)'], axis=1)

dispo_alim.to_csv("dispo_alim.csv", index = False)
dispo_alim.head()

#PK : pays + produit
```

```
Out[65]:
```

	idx	pays	code_pays	année	produit	code_produit	origin	dispo_alim_kcal_p_j	dispo_prot	dispo_mat_gr	dispo_alim_tonnes
0	0	Afghanistan	2	2013	Abats Comestible	2736	animal	5.0	0.77	0.20	52549.44
1	1	Afghanistan	2	2013	Agrumes, Autres	2614	vegetal	1.0	0.02	0.01	39412.08
2	2	Afghanistan	2	2013	Aliments pour enfants	2680	vegetal	1.0	0.03	0.01	1833.12
3	3	Afghanistan	2	2013	Ananas	2618	vegetal	0.0	NaN	NaN	0.00
4	4	Afghanistan	2	2013	Bananes	2615	vegetal	4.0	0.05	0.02	82490.40

Clé primaire après agrégation:  
Pays ou `code_pays` + produit ou  
`code_produit`



# Q17

## Q17

Une table appelée `equilibre_prod` contenant pour chaque pays et pour chaque produit en 2013, les quantités suivantes : disponibilité intérieure aliments pour animaux semences pertes transformés nourriture autres utilisations

Elle devra contenir ces colonnes : `pays`, `code_pays`, `année`, `produit`, `code_produit`, `dispo_int`, `alim_an`, `semences`, `pertes`, `transfo`, `nourriture`, `autres_utilisations`.

In [66]:

```
equilibre_prod = vapvtpop[["Zone", "Code zone", "Année", "Produit", "Code Produit", "Disponibilité intérieure", "Aliments pour animaux", "Semences", "Pertes", "Traitement"]]

equilibre_prod["idx"] = np.arange(len(equilibre_prod)) #to add a new column in sequential numbers

cols = equilibre_prod.columns.tolist()
cols = cols[-1:] + cols[:-1] # move the last element to the first position
equilibre_prod = equilibre_prod[cols] #reorder the dataframe

equilibre_prod.rename(columns={"idx": "idx", "Zone": "pays", "Code zone": "code_pays", "Année": "année", "Produit": "produit", "Code Produit": "code_produit", "Disponibilité intérieure": "dispo_int", "Aliments pour animaux": "alim_an", "Semences": "semences", "Pertes": "pertes", "Transformation": "transfo", "Nourriture": "nourriture", "Autres utilisations": "autres_utilisations"})

equilibre_prod.to_csv("equilibre_prod.csv", index = False)

equilibre_prod.head()

# PK: pays + produit
```

Out[66]:

	idx	pays	code_pays	année	produit	code_produit	dispo_int	alim_an	semences	pertes	transfo	nourriture	autres_utilisations
0	0	Afghanistan	2	2013	Abats Comestible	2736	53.0	NaN	NaN	NaN	NaN	53.0	NaN
1	1	Afghanistan	2	2013	Agrumes, Autres	2614	41.0	NaN	NaN	2.0	NaN	39.0	NaN
2	2	Afghanistan	2	2013	Aliments pour enfants	2680	2.0	NaN	NaN	NaN	NaN	2.0	NaN
3	3	Afghanistan	2	2013	Ananas	2618	0.0	NaN	NaN	NaN	NaN	0.0	NaN
4	4	Afghanistan	2	2013	Bananes	2615	82.0	NaN	NaN	NaN	NaN	82.0	NaN

Clé primaire après agrégation:  
Pays ou code\_pays + produit ou  
code\_produit

# Q18

## Q18

Une table appelée sous\_nutrition, contenant le nombre de personnes en sous-alimentation pour chaque pays en 2013. Elle devra contenir 4 colonnes : pays, code\_pays, année, nb\_personnes.

```
In [76]: # Q18

sous_nutrition = sousnutri[['Zone', 'Code zone', 'Année', 'Valeur']]
sous_nutrition = sous_nutrition[sous_nutrition['Année'] == 2013]

sous_nutrition['idx'] = np.arange(len(sous_nutrition)) #to add a new column in sequential numbers

cols = sous_nutrition.columns.tolist()
cols = cols[-1:] + cols[:-1] # move the last element to the first position
sous_nutrition = sous_nutrition[cols] #reorder the dataframe

sous_nutrition.rename(columns = {'Zone': 'pays', 'Code zone': 'code_pays', 'Année': 'année', 'Valeur': 'nb_personnes'}, inplace = True)
sous_nutrition['nb_personnes'] = sous_nutrition['nb_personnes']*1000000

sous_nutrition.to_csv("sous_nutrition.csv", index = False)
sous_nutrition.head(10)

#PK: pays
# transformation Année - suppression des années surplus - CSV, SQL
```

Out[76]:

	idx	pays	code_pays	année	nb_personnes
0	0	Afghanistan	2	2013	7900000.0
5	1	Afrique du Sud	202	2013	2600000.0
10	2	Albanie	3	2013	200000.0
15	3	Algérie	4	2013	1700000.0
20	4	Allemagne	79	2013	NaN
25	5	Andorre	6	2013	NaN
30	6	Angola	7	2013	8100000.0
35	7	Antigua-et-Barbuda	8	2013	NaN
40	8	Arabie saoudite	194	2013	1600000.0
45	9	Argentine	9	2013	1500000.0

- Projection
- Restriction

Clé primaire après agrégation:  
Pays ou code\_pays

# Q19.1

Les 10 pays ayant le plus **haut** ratio disponibilité alimentaire en termes de **protéines** (en kg) par habitant

```
SELECT DISTINCT pays AS "10 first countries with  
highest protein disponibility rate (kg)",  
round(SUM(dispo_prot),2) AS "DispoProtKg"  
  
FROM dispo_alim  
  
GROUP BY pays  
ORDER by "DispoProtKg" DESC  
LIMIT 10;
```

	10 first countries	DispoProtKg
1	Islande	133.06
2	Israël	128
3	Lituanie	124.36
4	Maldives	122.32
5	Finlande	117.56
6	Luxembourg	113.64
7	Monténégro	111.9
8	Pays-Bas	111.46
9	Albanie	111.37
10	Portugal	110.87

## Q19.2

Les 10 pays ayant le plus **haut** ratio disponibilité alimentaire en termes de **kcal** par habitant.

```
SELECT DISTINCT pays AS "10 first  
countries with highest calorie disponibility  
rate by habitant",  
round(sum(dispo_alim_kcal_p_j),2) AS  
DispoAlimKcal  
  
from dispo_alim  
  
group by pays  
order by DispoAlimKcal desc  
limit 10;
```

	10 first countries with highest calorie disponibility	DispoAlimKcal
1	Autriche	3770
2	Belgique	3737
3	Turquie	3708
4	États-Unis d'Amérique	3682
5	Israël	3610
6	Irlande	3602
7	Italie	3578
8	Luxembourg	3540
9	Égypte	3518
10	Allemagne	3503

## Q19.3 les 10 pays ayant le plus **faible** ratio disponibilité alimentaire en termes de **protéine** (en kg) par habitant.

```
Select dispo_alim.pays,  
round(sum(365*Pop.population*dispo_alim.d  
spo_prot/1000),0)  
AS "10 countries with lowest protein dispo ratio"  
FROM dispo_alim  
  
LEFT JOIN Pop on Pop.pays= dispo_alim.pays  
  
Group by dispo_alim.pays  
Order by "10 countries with lowest protein dispo  
ratio" ASC  
LIMIT 10;
```

	pays	10 countries with lowest protein
1	Saint-Kitts-et-Nevis	1425427
2	Dominique	2004638
3	Bermudes	2143791
4	Grenade	2677348
5	Kiribati	2730448
6	Antigua-et-Barbuda	2742647
7	Saint-Vincent-et-les Grenadines	3432252
8	Sao Tomé-et-Principe	3740630
9	Sainte-Lucie	5695708
10	Samoa	5931506

## Q19.4 La quantité totale (en kg) de produits perdus par pays en 2013.

```
SELECT pays, sum(pertes*1000000) AS "Perte  
de quantité totale en kg/pays"
```

```
FROM equilibre_prod
```

```
GROUP BY pays;
```

	pays	Perte de quantité totale en kg/pays
1	Afghanistan	1135000000.
2	Afrique du Sud	2193000000.
3	Albanie	276000000.
4	Algérie	3753000000.
5	Allemagne	3781000000.
6	Angola	4799000000.
7	Antigua-et-Barbuda	0
8	Arabie saoudite	1040000000.
9	Argentine	3522000000.
10	Arménie	228000000.
11	Australie	520000000.

## Q19.5

Les 10 pays pour lesquels la proportion de **personnes sous-alimentées** est la plus forte

```
SELECT
sous_nutrition.pays,sous_nutrition.nb_per
sonnes,Pop.population,
round((round(sous_nutrition.nb_personne
s,0)/Pop.population)*100,2) AS
"Pourcentage de sous nutrition, %"
```

From Pop, sous\_nutrition

```
WHERE Pop.pays = sous_nutrition.pays
Group by sous_nutrition.pays
Order by "Pourcentage de sous nutrition,
%" DESC
Limit 10;
```

	pays	nb_personnes	population	Pourcentage
1	Haïti	5200000	10317000	50.4
2	Zambie	7000000	14539000	48.15
3	Zimbabwe	6600000	14150000	46.64
4	République centrafricaine	2000000	4616000	43.33
5	République populaire démocratique de Corée	10600000	24895000	42.58
6	Congo	1800000	4448000	40.47
7	Tchad	4900000	12825000	38.21
8	Angola	8100000	21472000	37.72
9	Libéria	1600000	4294000	37.26
10	Madagascar	8200000	22925000	35.77

## 19.6 Les 10 produits pour lesquels le ratio **Autres utilisations/Disponibilité intérieure** est le plus **élevé**.

```
SELECT produit AS "10 produits",  
round(100*avg(autres_utilisations/dispo_i  
nt),2) AS "Ratio Autres utilisations/Dispo  
intérieure,%"
```

```
FROM equilibre_prod
```

```
Group by produit
```

```
Order by "Ratio Autres utilisations/Dispo  
intérieure,%" DESC
```

```
Limit 10;
```

	10 produits	Ratio Autres utilisations/Dispo intérieure, %
1	Alcool, non Comestible	96.46
2	Plantes Aquatiques	91.89
3	Huile de Palme	65.03
4	Huil Plantes Oleif Autr	54.96
5	Huile de Palmistes	53.5
6	Huile de Colza&Moutarde	45.84
7	Huiles de Poissons	40.37
8	Huile de Coco	36.84
9	Graisses Animales Crue	30.58
10	Manioc	23.06



## Q20. Quelles sont ces "autres utilisations" possibles?

### Alcool

- ingrédient de produits alimentaires et pharmaceutiques, cosmétologiques
- solvant
- biocide
- matière première de la chimie, de la pharmacie
- combustible

### Huile de palme

- Agro-carburants
- produits cosmétiques (oléochimie)

### Huile de colza

- Huile alimentaire
- Fabrication de margarine
- Directe ou indirecte dans biocarburant
- Utilisation dans des thermomètres

### Huile de poissons

- Alimentation humaine
- Industrie pharmaceutique
- Alimentation des animaux d'élevage
- Alimentation des animaux de compagnie
- Aquaculture
- Travail du cuir (tannerie, chamoiserie)
- Industrie chimique de spécialités

### Huile de coco

Cosmétologiques

(dentifrice, soin nourrissant des cheveux et de la peau)



**Q & A**



# Merci

@Xuefei ZHANG 2021