



PROJET FAO PRÉSENTATION

Powered by Jérémy and Matthieu

PLAN

Introduction

Présentation du projet FAO

01



Etude de la population

Etude de la base de données
population

02



Etude de la disponibilité alimentaire

Etude des bases de données cereal
et vegetal

03



Retour d'expérience

Retour sur une expérience épique !

04



01

Introduction :
Présentation du projet

01



Contexte du projet

Au sein du service Data Analytics and Prospective de la FAO, vous êtes chargé d'analyser les données produites par les Nations Unies.

Ces analyses conduisent à des rapports de synthèse à destination des décideurs.

Pour être également en charge de la partie Prospective, c'est-à-dire donner des tendances et des évolutions probables quant à la production agricole, à l'alimentation et la nutrition mondiales.

—DATA ANALYST AU SEIN DE LA FAO





" qu'il y ait du pain "

FOOD AND AGRICULTURE OF THE UNITED NATIONS

- Organisation spécialisée du système des Nations Unies, créée en 1945 au Québec.
- Son siège est à Rome depuis 1951 au Palazzo FAO.
- La FAO compte 197 membres avec 194 pays membres, une organisation membre (UE) et 2 pays associés.
- La mission de la FAO est d'assister les pays qui en sont membres dans la gestion des ressources liées à l'alimentation et l'agriculture, de donner une vision prospective.
- La FAO a la plus grande base de données, aujourd'hui gratuite, sur l'alimentation, l'agriculture et la faim.

01

Introduction :
Présentation du projet

02

Étude de la population :
Etude la base de données population

02

RECUPERATION DES DONNEES

```
#permet de voir le nom du répertoire courant
path = os.getcwd()
print(f"Le répertoire courant est : {path} \n")
```



Le répertoire courant est : /home/matthieu

```
#----- Déclaration d'un dataframe pour récupérer les données
fns=glob.glob('fao_2013/*.csv', recursive=True)
```

```
df_bigTab = pd.DataFrame(columns=['Code Domaine', 'Domaine', 'Code Pays', 'Pays', 'Code Élément',
    'Élément', 'Code Produit', 'Produit', 'Code Année', 'Année', 'Unité',
    'Valeur', 'Symbole', 'Description du Symbole'])
nameModificateur = ['Code Domaine', 'Domaine', 'Code Pays', 'Pays', 'Code Élément',
    'Élément', 'Code Produit', 'Produit', 'Code Année', 'Année', 'Unité',
    'Valeur', 'Symbole', 'Description du Symbole']
```

```
#----- Construction d'un dataframe réunissant les csv
for fn in fns:
    fns = fn[13:-4]
    df=pd.read_csv(fn, names=nameModificateur, header=0)
    print("\n")
    print(f"{Fore.YELLOW}----- Récupération du fichier {fns} -----{Style.RESET_ALL}\n")
    print(f"- {fn:50s} ({df.shape[0]:6d}, {df.shape[1]:2d}) - {Fore.BLACK}{Back.CYAN}{fns}{Style.RESET_ALL} \n")
    print(f"{Fore.YELLOW}-----{Style.RESET_ALL}\n")
    print("\n")
    print(f"{Fore.YELLOW}----- Analyse du fichier {fns} -----{Style.RESET_ALL}\n")
    print(f"{df.describe()}")
    print("\n")
    print(f"{Fore.YELLOW}-----{Style.RESET_ALL}\n")
    df_bigTab=df_bigTab.append(df)
```

RECUPERATION DES DONNEES

Exemple de résultat du code précédent

ADRESSE
RÉCUPÉRER

```
----- Récupération du fichier TAT_2013_cereal -----
- fao_2013/FAOSTAT_2013_cereal.csv                ( 16057, 14) - TAT_2013_cereal
-----
```

NOM DU
FICHIER

```
----- Analyse du fichier TAT_2013_cereal -----
```

	Code Pays	Code Élément	Code Produit	Code Année	Année \
count	16057.000000	16057.000000	16057.000000	16057.0	16057.0
mean	128.528243	3944.553030	2554.817836	2013.0	2013.0
std	75.298386	2191.371293	99.697147	0.0	0.0
min	1.000000	645.000000	2511.000000	2013.0	2013.0
25%	66.000000	684.000000	2513.000000	2013.0	2013.0
50%	122.000000	5142.000000	2516.000000	2013.0	2013.0
75%	191.000000	5521.000000	2520.000000	2013.0	2013.0
max	351.000000	5911.000000	2805.000000	2013.0	2013.0

RESULTAT DE
LA MÉTHODE
DESCRIBE

	Valeur
count	16057.000000
mean	608.890108
std	6947.381722
min	-39863.000000
25%	0.000000
50%	3.000000
75%	54.620000
max	353699.000000

CALCUL DE LA POPULATION

```
df_population=df_bigTab.query('Produit ==  
"Population").loc[:,["Code  
Pays","Pays","Valeur","Année"]].sort_values(by="Va  
leur", ascending=False)  
df_population=df_population.set_index('Code Pays')  
df_population
```



Création d'un Dataframe population par pays

```
pop_planetaire = df_population['Valeur'].sum()  
  
print (f"La population mondiale en 2013 est de :  
{pop_planetaire}")
```



Calcul de la population mondiale

La population mondiale en 2013 est de : 8413993.0



Résultat de la population mondiale

CORRECTION DU CALCUL DE LA POPULATION

```
#récupération du fichier population et calcul de la population mondiale
```

```
pop_planetaire = df_pop['Valeur'].sum()  
print (f"La population mondiale en 2013 est de : {pop_planetaire}")
```

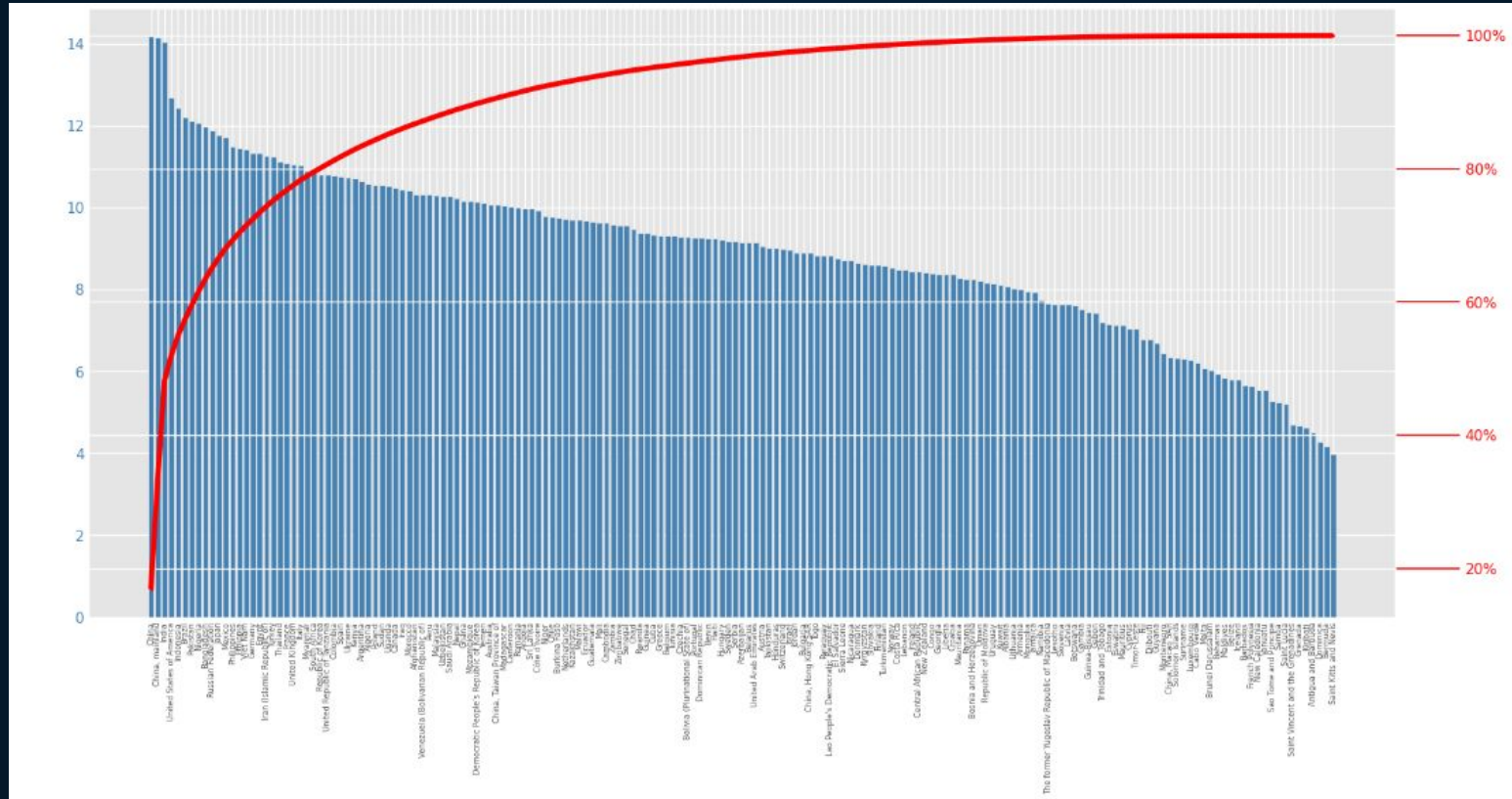
```
#résultat incohérent
```

```
#Le résultat peut changer lorsque l'on enlève les doublons
```

```
df_pop = df_pop.loc[df_pop["Pays"] != "China"]  
pop_planetaire = df_pop['Valeur'].sum()  
print (f"La vraie population mondiale en 2013 est de : {pop_planetaire}")
```

```
La population mondiale en 2013 est de : 8413993.0
```

```
La vraie population mondiale en 2013 est de : 6997326.0
```



01

Introduction :
Présentation du projet

02

Étude de la population :
Etude la base de données population

03

**Étude de la disponibilité
alimentaire :**
Étude des bases de données cereal
et vegetal

03

EQUATION DES REDONDANCES

**PRODUCTION
ET COMMERCE**

Production+Importation
s + Variation de stock -
Exportation



**DISPONIBILITÉS
INTÉRIEUR**



ENTRETIEN

Aliments pour animaux
+ Semences + Pertes +
Traitement + Autres
Utilisations + Nourriture

TENTATIVE D'UTILISATION DU GRAND TABLEAU

```
#-----Construction d'un tableau de la disponibilité alimentaire (question 3)-----

#SA  => Sous alimentation
#DA  => Disponibilité alimentaire (kg/personne/an)
#DAP => Disponibilité alimentaire en protéines (g/personne/jour)

df_SA=df_bigTab.query('Produit == "Nombre de personnes sous-alimentées (millions) (moyenne sur 3 ans)"').loc[:,["Code
Pays","Pays","Produit", "Valeur", "Unité"]]
df_DA=df_bigTab.query('Élément == "Disponibilité alimentaire en quantité (kg/personne/an)"').loc[:,["Code
Pays","Pays","Produit", "Valeur", "Unité"]]
df_DAP=df_bigTab.query('Élément == "Disponibilité de protéines en quantité (g/personne/jour)"').loc[:,["Code
Pays","Pays","Produit", "Valeur", "Unité"]]

df_aliment=pd.merge(df_pop,df_DA,on="Code Pays")
df_aliment=df_aliment.pivot_table('Valeur_x', index=['Code Pays','Pays_x', 'Produit_y'], columns='Valeur_y')
df_aliment

#Abandon de la méthode ici
```

Léger échec

Code Pays	Pays_x	Valeur_y	-1.93	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	...	288.12	291.86	295.46	303.72	308.07	318.69	341.23	341
		Produit_y																			
1	Armenia	Abats Comestible	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
		Agrumes, Autres	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN
		Aliments pour enfants	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN
		Ananas	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN
		Animaux Aqueiques Autre	NaN	2977.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN
...	
276	Sudan	Viande d'Ovins/Caprins	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN
		Viande de Bovins	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN
		Viande de Volailles	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN
		Viande, Autre	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN
		Épices, Autres	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	37964.0	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN

NOUVEAU DÉPART

```
#-----Retour à la question 3 -----  
  
#Récupération des trois fichiers pour le df aliments  
fn_vegetaux ="fao_2013/FAOSTAT_2013_vegetal.csv"  
vegetaux = pd.read_csv(fn_vegetaux)  
  
fn_population ="fao_2013/FAOSTAT_2013_population.csv"  
population = pd.read_csv(fn_population)  
  
fn_animaux ="fao_2013/FAOSTAT_2013_animal.csv"  
animaux = pd.read_csv(fn_animaux)
```


NOUVELLE PIVOT TABLE

```
animaux['Origine'] = 'Animale'
vegetaux['Origine'] = 'Végétale'

# Union de Végétaux et Animaux
aliments = pd.concat([vegetaux, animaux])

# Projection
aliments = aliments[[
    'Pays',
    'Code Pays',
    'Produit',
    'Code Produit',
    'Origine',
    'Élément',
    'Valeur',
    'Année'
]]

# Table pivot sur aliments
aliments = aliments.pivot_table('Valeur', index=['Code Pays', 'Pays', 'Produit', 'Code
Produit', 'Année', 'Origine'], columns='Élément')

# Reset index
aliments.reset_index(inplace=True)

# Joindre population et aliments
aliments = pd.merge(aliments, population[['Country', 'Value']], left_on='Pays', right_on='Country')
```

NOUVELLE PIVOT TABLE

Code Pays	Pays	Produit	Code Produit	Année	Origine	alim_an	Autres Utilisations	dispo_alim_kcal_p_j
2	Afghanistan	Abats Comestible	2736	2013	Animale	0.0	0.0	5.0
2	Afghanistan	Agrumes, Autres	2614	2013	Végétale	0.0	0.0	1.0
2	Afghanistan	Aliments pour enfants	2680	2013	Végétale	0.0	0.0	1.0
2	Afghanistan	Ananas	2618	2013	Végétale	0.0	0.0	0.0
2	Afghanistan	Bananes	2615	2013	Végétale	0.0	0.0	4.0
...
256	Luxembourg	Viande de	2722	2013	Animale	0.0	0.0	228.0

C'est mieux

DISPONIBILITÉS ALIMENTAIRES (KCAL ET KG PROT)

```
print("La disponibilité alimentaire en kcal pour chaque pays")
dispo_kcal=pd.DataFrame()
dispo_kcal["Pays"] = aliments["Pays"]
dispo_kcal["Produit"]=aliments["Produit"]
dispo_kcal["dispo alim en kcal/an"]=aliments["dispo_alim_kcal_p_j"]*aliments["population"]*365
print(dispo_kcal)
```

```
print("La disponibilité alimentaire en kg pour les protéines pour chaque Pays et chaque produit ")
```

```
dispo_prot=pd.DataFrame()

dispo_prot["Pays"] = aliments["Pays"]
dispo_prot["Produit"]=aliments["Produit"]
dispo_prot["dispo alim prot en kg"]=aliments['dispo_prot']*aliments["population"]*0.365
dispo_prot
```

DISPONIBILITÉS ALIMENTAIRES (KCAL ET KG PROT)

DISPONIBILITÉS ALIMENTAIRES EN KCAL:

La disponibilité alimentaire en kcal pour chaque pays

	Pays	Produit	dispo alim en kcal/an
0	Afghanistan	Abats Comestible	55757400.0
1	Afghanistan	Agrumes, Autres	11151480.0
2	Afghanistan	Aliments pour enfants	11151480.0
3	Afghanistan	Ananas	0.0
4	Afghanistan	Bananes	44605920.0
...
4759	Luxembourg	Viande de Suides	63451600.0
4760	Luxembourg	Viande de Volailles	14508750.0
4761	Luxembourg	Viande, Autre	1354150.0
4762	Luxembourg	Vin	19345000.0
4763	Luxembourg	Épices, Autres	1741050.0

DISPONIBILITÉS ALIMENTAIRES (KCAL ET KG PROT)

DISPONIBILITÉS ALIMENTAIRES DE PROT EN KG:

La disponibilité alimentaire en kg pour les protéines pour chaque Pays et chaque produit

	Pays	Produit	dispo alim prot en kg
0	Afghanistan	Abats Comestible	8586.6396
1	Afghanistan	Agrumes, Autres	223.0296
2	Afghanistan	Aliments pour enfants	334.5444
3	Afghanistan	Ananas	0.0000
4	Afghanistan	Bananes	557.5740
...
4759	Luxembourg	Viande de Suides	2331.0725
4760	Luxembourg	Viande de Volailles	1479.8925
4761	Luxembourg	Viande, Autre	183.7775
4762	Luxembourg	Vin	1.9345
4763	Luxembourg	Épices, Autres	58.0350

4764 rows × 3 columns

RATIO ÉNERGIE/POIDS

#-----Question 4-----

```
dispo_alim4=pd.DataFrame()  
dispo_alim4["Pays"] = aliments["Pays"]  
dispo_alim4["Produit"]=aliments["Produit"]  
dispo_alim4["Dispo alimentaire kg/an"]=aliments["Disponibilité alimentaire en quantité  
(kg/personne/an) "]*aliments["population"]  
dispo_alim4["dispo alim en kcal/an"]=dispo_kcal["dispo alim en kcal/an"]  
dispo_alim4["Ratio énergie/poids"]=dispo_alim4["dispo alim en kcal/an"]/(dispo_alim4["Dispo alimentaire  
kg/an"])  
  
dispo_alim4[dispo_alim4.Produit=="Oeufs"]
```

	Pays	Produit	Dispo alimentaire kg/an	dispo alim en kcal/an	Ratio énergie/poids
37	Afghanistan	Oeufs	39717.60	5.575740e+07	1403.846154
107	Angola	Oeufs	23833.92	3.134912e+07	1315.315315
195	Bahamas	Oeufs	2337.40	3.164915e+06	1354.032258
286	Bangladesh	Oeufs	325717.60	5.144146e+08	1579.326923

ALIMENTS LES PLUS CALORIQUES

#-----Question 5-----

```
dispo_alim5 = dispo_alim4.loc[dispo_alim4["Dispo alimentaire kg/an"] != 0]
dispo_alim5=dispo_alim5.sort_values('Ratio énergie/poids', ascending = False)
dispo_alim5.head(20)
```

	Pays	Produit	Dispo alimentaire kg/an	dispo alim en kcal/an	Ratio énergie/poids
4182	Oman	Huile d'Arachide	72.64	1325680.0	18250.0
3507	Portugal	Huiles de Foie de Poisso	212.16	3871920.0	18250.0
2586	Maldives	Huile de Colza&Moutarde	6.90	125925.0	18250.0
1540	Ghana	Huile d'Olive	518.10	9455325.0	18250.0
725	Cabo Verde	Huile de Tournesol	9.98	182135.0	18250.0

DISPONIBILITÉS INTÉRIEURES MONDIALES VÉGÉTAUX

```
dispo_alim6 = aliments.loc[aliments["Origine"] == "Végétale"]
dispo_alim6 = dispo_alim6[['Pays', 'Produit', 'Origine', 'dispo_int']]
dispo_alim6["Ratio énergie/poids"] = dispo_alim6["Ratio énergie/poids"]
dispo_alim6["Dispo int en kcal"] = dispo_alim6["Ratio énergie/poids"] * dispo_alim6["dispo_int"] * 1000000

dispo_alim6
```

	Pays	Produit	Origine	dispo_int	Ratio énergie/poids	Dispo int en kcal
1	Afghanistan	Agrumes, Autres	Végétale	41.0	282.945736	1.160078e+10
2	Afghanistan	Aliments pour enfants	Végétale	2.0	6083.333333	1.216667e+10
3	Afghanistan	Ananas	Végétale	0.0	NaN	NaN
4	Afghanistan	Bananes	Végétale	82.0	540.740741	4.434074e+10
6	Afghanistan	Bière	Végétale	3.0	0.000000	0.000000e+00
...

01

Introduction :
Présentation du projet

02

Étude de la population :
Etude la base de données population

04

Retour d'expérience :
Retour sur une expérience épique

**Étude de la disponibilité
alimentaire :**
Étude des bases de données cereal
et vegetal

04

03

RÉUSSITES SUR LE PROJET



MANIPULATION DE DATAFRAMES



UTILISATION DU SQL AVEC SQLITE



UTILISATION DU SHELL LINUX ET
CONNEXION SSH



CRÉATION DE GRAPHIQUE



ÉCHECS SUR LE PROJET



MANQUE DE TEMPS



MANQUE D'EXPERTISE



POURSUITE D'UNE VERSION DU
PROGRAMME TROP COMPLEXE



Commentaires



Avec en guest : Matthieu ERNOUX

" Le ridicule ne tue pas, ce qui ne tue pas vous rend plus fort, donc le ridicule vous rend plus fort "

