

Gwynn_Gabriel_1_notebook_112024

November 21, 2024

PROJET 4 DATA ANALYST

Réalisez une étude de santé publique avec R ou Python

1 OBJECTIF DE CE NOTEBOOK

Bienvenue dans l'outil plébiscité par les analystes de données Jupyter.

Il s'agit d'un outil permettant de mixer et d'alterner codes, textes et graphique.

Cet outil est formidable pour plusieurs raisons:

- il permet de tester des lignes de codes au fur et à mesure de votre rédaction, de constater immédiatement le résultat d'une instruction, de la corriger si nécessaire.
- De rédiger du texte pour expliquer l'approche suivie ou les résultats d'une analyse et de le mettre en forme grâce à du code html ou plus simple avec **Markdown**
- d'agrémenter de graphiques

Pour vous aider dans vos premiers pas à l'usage de Jupyter et de Python, nous avons rédigé ce notebook en vous indiquant les instructions à suivre.

Il vous suffit pour cela de saisir le code Python répondant à l'instruction donnée.

Vous verrez de temps à autre le code Python répondant à une instruction donnée mais cela est fait pour vous aider à comprendre la nature du travail qui vous est demandée.

Et garder à l'esprit, qu'il n'y a pas de solution unique pour résoudre un problème et qu'il y a autant de résolutions de problèmes que de développeurs ;)...

Note Jeremy Est ce qu'il faut faire le calcul de la sous nutrition sur les pays qu'on a ? Est ce qu'il faut faire des graphiques ? Rajouter le soja La liste des céréales est difficile à trouver ...

Etape 1 - Importation des librairies et chargement des fichiers

1.1 - Importation des librairies

```
[6]: #Importation de la librairie Pandas
import pandas as pd
```

```
[7]: import matplotlib.pyplot as plt
```

1.2 - Chargement des fichiers Excel

```
[9]: #Importation du fichier population.csv
population = pd.read_csv('population.csv')

#Importation du fichier dispo_alimentaire.csv
dispo_alimentaire = pd.read_csv("dispo_alimentaire.csv")

#Importation du fichier aide_alimentaire.csv
aide_alimentaire = pd.read_csv("aide_alimentaire.csv")

#Importation du fichier sous_nutrition.csv
sous_nutrition = pd.read_csv("sous_nutrition.csv")
```

Etape 2 - Analyse exploratoire des fichiers

2.1 - Analyse exploratoire du fichier population

```
[12]: #Afficher les dimensions du dataset
print("Le tableau comporte {} observation(s) ou article(s)".format(population.
    ↪shape[0]))
print("Le tableau comporte {} colonne(s)".format(population.shape[1]))
```

Le tableau comporte 1416 observation(s) ou article(s)

Le tableau comporte 3 colonne(s)

```
[13]: #Consulter le nombre de colonnes
print("il y a", population.shape[1], "colonnes dans le fichier population")
#La nature des données dans chacune des colonnes
print(population.dtypes)
#Le nombre de valeurs présentes dans chacune des colonnes
print("il y a", population.shape[0], "lignes dans le fichier population")
```

il y a 3 colonnes dans le fichier population

Zone object

Année int64

Valeur float64

dtype: object

il y a 1416 lignes dans le fichier population

```
[14]: #Affichage les 5 premières lignes de la table
population.head()
```

```
[14]:
```

	Zone	Année	Valeur
0	Afghanistan	2013	32269.589
1	Afghanistan	2014	33370.794
2	Afghanistan	2015	34413.603
3	Afghanistan	2016	35383.032
4	Afghanistan	2017	36296.113

```
[15]: #Nous allons harmoniser les unités. Pour cela, nous avons décidé de multiplier
      ↪ la population par 1000
      #Multiplication de la colonne valeur par 1000
      population["Valeur"] = population["Valeur"]*1000
```

```
[16]: #changement du nom de la colonne Valeur par Population
      population = population.rename(columns={"Valeur":"Population"})
```

```
[17]: #Affichage les 5 premières lignes de la table pour voir les modifications
      population.head()
```

```
[17]:
```

	Zone	Année	Population
0	Afghanistan	2013	32269589.0
1	Afghanistan	2014	33370794.0
2	Afghanistan	2015	34413603.0
3	Afghanistan	2016	35383032.0
4	Afghanistan	2017	36296113.0

2.2 - Analyse exploratoire du fichier disponibilité alimentaire

```
[19]: #Afficher les dimensions du dataset
      print("lignes et colonnes")
      dispo_alimentaire.shape
```

lignes et colonnes

```
[19]: (15605, 18)
```

```
[20]: #Consulter le nombre de colonnes
      print("il y a", dispo_alimentaire.shape[1], "colonnes")
```

il y a 18 colonnes

```
[21]: #Affichage les 5 premières lignes de la table
      dispo_alimentaire.head()
```

```
[21]:
```

	Zone	Produit	Origine	Aliments pour animaux	\
0	Afghanistan	Abats Comestible	animale	NaN	
1	Afghanistan	Agrumes, Autres	vegetale	NaN	
2	Afghanistan	Aliments pour enfants	vegetale	NaN	
3	Afghanistan	Ananas	vegetale	NaN	
4	Afghanistan	Bananes	vegetale	NaN	

	Autres Utilisations	Disponibilité alimentaire (Kcal/personne/jour)	\
0	NaN	5.0	
1	NaN	1.0	
2	NaN	1.0	
3	NaN	0.0	
4	NaN	4.0	

	Disponibilité alimentaire en quantité (kg/personne/an) \
0	1.72
1	1.29
2	0.06
3	0.00
4	2.70

	Disponibilité de matière grasse en quantité (g/personne/jour) \
0	0.20
1	0.01
2	0.01
3	NaN
4	0.02

	Disponibilité de protéines en quantité (g/personne/jour) \
0	0.77
1	0.02
2	0.03
3	NaN
4	0.05

	Disponibilité intérieure	Exportations - Quantité	Importations - Quantité \
0	53.0	NaN	NaN
1	41.0	2.0	40.0
2	2.0	NaN	2.0
3	0.0	NaN	0.0
4	82.0	NaN	82.0

	Nourriture	Pertes	Production	Semences	Traitement	Variation de stock
0	53.0	NaN	53.0	NaN	NaN	NaN
1	39.0	2.0	3.0	NaN	NaN	NaN
2	2.0	NaN	NaN	NaN	NaN	NaN
3	0.0	NaN	NaN	NaN	NaN	NaN
4	82.0	NaN	NaN	NaN	NaN	NaN

```
[22]: #remplacement des NaN dans le dataset par des 0
dispo_alimentaire = dispo_alimentaire.fillna(0)
```

```
[23]: #multiplication de toutes les lignes contenant des milliers de tonnes en Kg
dispo_alimentaire.iloc[:,-9:] = dispo_alimentaire.iloc[:,-9:]*1000000
dispo_alimentaire.iloc[:, 3:5] = dispo_alimentaire.iloc[:, 3:5]*1000000
```

```
[24]: #Affichage les 5 premières lignes de la table
dispo_alimentaire.head()
```

[24] :

	Zone	Produit	Origine	Aliments pour animaux	\
0	Afghanistan	Abats Comestible	animale	0.0	
1	Afghanistan	Agrumes, Autres	vegetale	0.0	
2	Afghanistan	Aliments pour enfants	vegetale	0.0	
3	Afghanistan	Ananas	vegetale	0.0	
4	Afghanistan	Bananes	vegetale	0.0	

	Autres Utilisations	Disponibilité alimentaire (Kcal/personne/jour)	\
0	0.0	5.0	
1	0.0	1.0	
2	0.0	1.0	
3	0.0	0.0	
4	0.0	4.0	

	Disponibilité alimentaire en quantité (kg/personne/an)	\
0	1.72	
1	1.29	
2	0.06	
3	0.00	
4	2.70	

	Disponibilité de matière grasse en quantité (g/personne/jour)	\
0	0.20	
1	0.01	
2	0.01	
3	0.00	
4	0.02	

	Disponibilité de protéines en quantité (g/personne/jour)	\
0	0.77	
1	0.02	
2	0.03	
3	0.00	
4	0.05	

	Disponibilité intérieure	Exportations - Quantité	Importations - Quantité	\
0	53000000.0	0.0	0.0	
1	41000000.0	2000000.0	40000000.0	
2	2000000.0	0.0	2000000.0	
3	0.0	0.0	0.0	
4	82000000.0	0.0	82000000.0	

	Nourriture	Pertes	Production	Semences	Traitement	Variation de stock
0	53000000.0	0.0	53000000.0	0.0	0.0	0.0
1	39000000.0	2000000.0	3000000.0	0.0	0.0	0.0
2	2000000.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0

```
4 82000000.0      0.0      0.0      0.0      0.0      0.0
```

2.3 - Analyse exploratoire du fichier aide alimentaire

```
[26]: #Afficher les dimensions du dataset
print("lignes et colonnes")
aide_alimentaire.shape
```

lignes et colonnes

```
[26]: (1475, 4)
```

```
[27]: #Consulter le nombre de colonnes
print("il y a", aide_alimentaire.shape[1], "colonnes")
```

il y a 4 colonnes

```
[28]: #Affichage les 5 premières lignes de la table
aide_alimentaire.head()
```

```
[28]:
```

	Pays bénéficiaire	Année	Produit	Valeur
0	Afghanistan	2013	Autres non-céréales	682
1	Afghanistan	2014	Autres non-céréales	335
2	Afghanistan	2013	Blé et Farin	39224
3	Afghanistan	2014	Blé et Farin	15160
4	Afghanistan	2013	Céréales	40504

```
[29]: #changement du nom de la colonne Pays bénéficiaire par Zone
aide_alimentaire = aide_alimentaire.rename(columns={"Pays bénéficiaire": "Zone"})
```

```
[30]: #Multiplication de la colonne Aide_alimentaire qui contient des tonnes par 1000
      ↪pour avoir des kg
aide_alimentaire["Valeur"] = aide_alimentaire["Valeur"]*1000
```

```
[31]: #Affichage les 5 premières lignes de la table
aide_alimentaire.head()
```

```
[31]:
```

	Zone	Année	Produit	Valeur
0	Afghanistan	2013	Autres non-céréales	682000
1	Afghanistan	2014	Autres non-céréales	335000
2	Afghanistan	2013	Blé et Farin	39224000
3	Afghanistan	2014	Blé et Farin	15160000
4	Afghanistan	2013	Céréales	40504000

2.3 - Analyse exploratoire du fichier sous nutrition

```
[33]: #Afficher les dimensions du dataset
print("lignes et colonnes")
sous_nutrition.shape
```

lignes et colonnes

[33]: (1218, 3)

```
[34]: #Consulter le nombre de colonnes
print("il y a", sous_nutrition.shape[1], "colonnes")
```

il y a 3 colonnes

```
[35]: #Afficher les 5 premières lignes de la table
sous_nutrition.head()
```

```
[35]:
```

	Zone	Année	Valeur
0	Afghanistan	2012-2014	8.6
1	Afghanistan	2013-2015	8.8
2	Afghanistan	2014-2016	8.9
3	Afghanistan	2015-2017	9.7
4	Afghanistan	2016-2018	10.5

```
[36]: #Conversion de la colonne sous nutrition en numérique
#erreur volontaire ==> sous_nutrition['Valeur'] = pd.
      ↪to_numeric(sous_nutrition['Valeur'])
```

```
[37]: #Conversion de la colonne (avec l'argument errors=coerce qui permet de
      ↪convertir automatiquement les lignes qui ne sont pas des nombres en NaN)
sous_nutrition['Valeur'] = pd.
      ↪to_numeric(sous_nutrition['Valeur'],errors="coerce")
#Puis remplacement des NaN en 0
sous_nutrition["Valeur"] = sous_nutrition["Valeur"].fillna(0)
```

```
[38]: #changement du nom de la colonne Valeur par sous_nutrition
sous_nutrition = sous_nutrition.rename(columns={"Valeur":"sous_nutrition"})
```

```
[39]: #Multiplication de la colonne sous_nutrition par 1000000
sous_nutrition["sous_nutrition"] = sous_nutrition["sous_nutrition"]*1000000
```

```
[40]: #Afficher les 5 premières lignes de la table
sous_nutrition.head()
```

```
[40]:
```

	Zone	Année	sous_nutrition
0	Afghanistan	2012-2014	8600000.0
1	Afghanistan	2013-2015	8800000.0
2	Afghanistan	2014-2016	8900000.0
3	Afghanistan	2015-2017	9700000.0
4	Afghanistan	2016-2018	10500000.0

3.1 - Proportion de personnes en sous nutrition

```
[42]: # Il faut tout d'abord faire une jointure entre la table population et la table
      ↪ sous nutrition, en ciblant l'année 2017
population_2017 = population.loc[population["Année"] == 2017, :]
population_2017 = population_2017.drop(columns="Année")
sous_nutrition_2017 = sous_nutrition.loc[sous_nutrition["Année"] ==
      ↪ "2016-2018", :]
sous_nutrition_2017 = sous_nutrition_2017.drop(columns="Année")
population_sous_nutrition_2017 = pd.merge(population_2017, sous_nutrition_2017,
      ↪ on="Zone")
population_sous_nutrition_2017.head()
```

```
[42]:
```

	Zone	Population	sous_nutrition
0	Afghanistan	36296113.0	10500000.0
1	Afrique du Sud	57009756.0	3100000.0
2	Albanie	2884169.0	100000.0
3	Algérie	41389189.0	1300000.0
4	Allemagne	82658409.0	0.0

```
[43]: #Affichage du dataset
display(population_sous_nutrition_2017)
```

	Zone	Population	sous_nutrition
0	Afghanistan	36296113.0	10500000.0
1	Afrique du Sud	57009756.0	3100000.0
2	Albanie	2884169.0	100000.0
3	Algérie	41389189.0	1300000.0
4	Allemagne	82658409.0	0.0
..
198	Venezuela (République bolivarienne du)	29402484.0	8000000.0
199	Viet Nam	94600648.0	6500000.0
200	Yémen	27834819.0	0.0
201	Zambie	16853599.0	0.0
202	Zimbabwe	14236595.0	0.0

[203 rows x 3 columns]

```
[44]: #Calcul et affichage du nombre de personnes en état de sous nutrition
print("Il y a", round(population_sous_nutrition_2017["sous_nutrition"].sum()/
      ↪ 1000000), "M de personnes en état de sous nutrition")
```

Il y a 536 M de personnes en état de sous nutrition

```
[45]: print("Ce qui représente", ((population_sous_nutrition_2017["sous_nutrition"].
      ↪ sum())
                                     /(population_sous_nutrition_2017["Population"].
      ↪ sum())*100).round(2),
      ↪ "% de la population mondiale")
```

Ce qui représente 7.1 % de la population mondiale


```
[46]: #Population mondiale
print("La population mondiale est_
↳de",round(population_sous_nutrition_2017["Population"].sum()/1000000),"M en_
↳2017.")
```

La population mondiale est de 7544 M en 2017.

3.2 - Nombre théorique de personne qui pourrait être nourries

```
[48]: #Combien mange en moyenne un être humain ? Source => https://www.fao.org/4/
↳y3557f/y3557f15.htm
# La consommation moyenne humaine mondiale est de 2940 Calories/personne/jour_
↳(pour l'année 2015 source FAO)
#Source => https://www.anses.fr/fr/system/files/NUT2012SA0103Ra-2.pdf
# L'ANSES estime un besoin énergétique de 2600 kcal/j pour les hommes et de_
↳2100 kcal/j pour les femmes ce qui fait une moyenne de 2350kcal/j
```

```
[49]: #On commence par faire une jointure entre le data frame population et_
↳Dispo_alimentaire afin d'ajouter dans ce dernier la population
dispo_alimentaire_sum = dispo_alimentaire.groupby("Zone").sum()
dispo_alimentaire_sum.drop(["Produit","Origine"], axis=1, inplace=True)
population_dispo_alimentaire = pd.merge(population.loc[population['Année'] ==_
↳2017,["Zone", "Population"]],
dispo_alimentaire_sum,on='Zone')
```

```
[50]: #Affichage du nouveau dataframe
display(population_dispo_alimentaire)
```

	Zone	Population \
0	Afghanistan	36296113.0
1	Afrique du Sud	57009756.0
2	Albanie	2884169.0
3	Algérie	41389189.0
4	Allemagne	82658409.0
..
167	Venezuela (République bolivarienne du)	29402484.0
168	Viet Nam	94600648.0
169	Yémen	27834819.0
170	Zambie	16853599.0
171	Zimbabwe	14236595.0

	Aliments pour animaux	Autres Utilisations \
0	7.680000e+08	4.150000e+08
1	5.309000e+09	8.760000e+08
2	6.600000e+08	1.740000e+08
3	4.352000e+09	1.707000e+09
4	3.020900e+10	7.711000e+09
..
167	2.878000e+09	3.350000e+08

168	1.135600e+10	3.815000e+09
169	2.100000e+08	7.200000e+07
170	4.080000e+08	6.900000e+07
171	1.310000e+08	2.300000e+07

	Disponibilité alimentaire (Kcal/personne/jour) \
0	2087.0
1	3020.0
2	3188.0
3	3293.0
4	3503.0
..	...
167	2633.0
168	2744.0
169	2217.0
170	1924.0
171	2113.0

	Disponibilité alimentaire en quantité (kg/personne/an) \
0	351.41
1	564.80
2	1095.54
3	809.13
4	957.52
..	...
167	652.86
168	574.84
169	375.41
170	350.12
171	336.59

	Disponibilité de matière grasse en quantité (g/personne/jour) \
0	33.50
1	83.58
2	106.48
3	76.48
4	141.62
..	...
167	82.44
168	72.11
169	47.81
170	42.05
171	57.18

	Disponibilité de protéines en quantité (g/personne/jour) \
0	58.26
1	85.33
2	111.37

3	91.92
4	101.39
..	...
167	72.96
168	81.61
169	58.90
170	55.22
171	48.32

	Disponibilité intérieure	Exportations - Quantité \
0	1.351500e+10	2.780000e+08
1	6.125600e+10	1.167400e+10
2	4.758000e+09	9.400000e+07
3	4.263000e+10	6.550000e+08
4	1.622750e+11	5.743100e+10
..
167	3.228700e+10	6.200000e+07
168	9.513400e+10	2.263800e+10
169	9.782000e+09	7.190000e+08
170	1.004100e+10	6.750000e+08
171	9.507000e+09	2.780000e+08

	Importations - Quantité	Nourriture	Pertes	Production \
0	2.988000e+09	1.073500e+10	1.135000e+09	1.117100e+10
1	7.517000e+09	2.981200e+10	2.193000e+09	6.326300e+10
2	9.330000e+08	3.476000e+09	2.760000e+08	3.964000e+09
3	1.740300e+10	3.172900e+10	3.753000e+09	2.635900e+10
4	6.535300e+10	7.923800e+10	3.781000e+09	1.545470e+11
..
167	9.604000e+09	1.985300e+10	1.190000e+09	2.386800e+10
168	9.217000e+09	5.270000e+10	6.743000e+09	1.082750e+11
169	6.786000e+09	9.162000e+09	2.690000e+08	4.125000e+09
170	3.800000e+08	5.088000e+09	2.420000e+08	1.052600e+10
171	1.342000e+09	4.762000e+09	1.880000e+08	7.914000e+09

	Semences	Traitement	Variation de stock
0	3.950000e+08	6.300000e+07	-3.680000e+08
1	2.530000e+08	2.281900e+10	2.153000e+09
2	5.500000e+07	1.130000e+08	-4.500000e+07
3	4.490000e+08	6.370000e+08	-4.790000e+08
4	1.551000e+09	3.984200e+10	-1.920000e+08
..
167	9.100000e+07	7.957000e+09	-1.130000e+09
168	1.211000e+09	1.931100e+10	2.780000e+08
169	6.700000e+07	3.900000e+07	-4.080000e+08
170	7.600000e+07	4.178000e+09	-1.820000e+08
171	7.600000e+07	4.330000e+09	5.310000e+08

[172 rows x 17 columns]

```
[51]: #Création de la colonne dispo_kcal avec calcul des kcal disponibles mondialement
population_dispo_alimentaire = population_dispo_alimentaire.
      ↪rename(columns={"Disponibilité alimentaire (Kcal/personne/jour)":
      ↪"dispo_kcal"})
dispo_kcal_monde =
      ↪population_dispo_alimentaire["dispo_kcal"]*population_dispo_alimentaire["Population"]
print("La disponibilité alimentaire est de", round(dispo_kcal_monde.sum()/
      ↪10000000000),"Tcal/jour (milliards de calories par jour) dans le monde")
```

La disponibilité alimentaire est de 20919 Tcal/jour (milliards de calories par jour) dans le monde

```
[52]: #Calcul du nombre d'humains pouvant être nourris
print("Le nombre théorique d'humains pouvant être nourris est de",
      ↪(dispo_kcal_monde.sum()/2350/10000000000).round(2),"milliards.")
```

Le nombre théorique d'humains pouvant être nourris est de 8.9 milliards.

3.3 - Nombre théorique de personne qui pourrait être nourrie avec les produits végétaux

```
[54]: #Transfert des données avec les végétaux dans un nouveau dataframe
dispo_alimentaire_vegetal_sum = (dispo_alimentaire.
      ↪loc[dispo_alimentaire["Origine"]=="vegetale"]).groupby("Zone").sum()
dispo_alimentaire_vegetal_sum.drop(["Produit","Origine"], axis=1, inplace=True)
dispo_alimentaire_vegetal_sum = pd.merge(population.loc[population['Année'] ==
      ↪2017,["Zone", "Population"]],
      ↪dispo_alimentaire_vegetal_sum,on='Zone')
dispo_alimentaire_vegetal_sum
```

```
[54]:
```

	Zone	Population \
0	Afghanistan	36296113.0
1	Afrique du Sud	57009756.0
2	Albanie	2884169.0
3	Algérie	41389189.0
4	Allemagne	82658409.0
..
167	Venezuela (République bolivarienne du)	29402484.0
168	Viet Nam	94600648.0
169	Yémen	27834819.0
170	Zambie	16853599.0
171	Zimbabwe	14236595.0

	Aliments pour animaux	Autres Utilisations \
0	6.450000e+08	4.150000e+08
1	5.122000e+09	7.610000e+08
2	5.590000e+08	1.720000e+08
3	4.352000e+09	1.706000e+09

4	2.812200e+10	6.842000e+09
..
167	2.778000e+09	2.780000e+08
168	1.064700e+10	3.595000e+09
169	2.100000e+08	6.500000e+07
170	4.020000e+08	6.800000e+07
171	8.500000e+07	2.000000e+07

	Disponibilité alimentaire (Kcal/personne/jour) \
0	1871.0
1	2533.0
2	2203.0
3	2915.0
4	2461.0
..	...
167	2157.0
168	2169.0
169	2028.0
170	1818.0
171	1935.0

	Disponibilité alimentaire en quantité (kg/personne/an) \
0	271.49
1	425.17
2	706.49
3	632.35
4	568.82
..	...
167	442.56
168	459.79
169	303.73
170	316.47
171	278.92

	Disponibilité de matière grasse en quantité (g/personne/jour) \
0	18.27
1	48.92
2	38.76
3	54.12
4	63.63
..	...
167	51.20
168	24.62
169	35.22
170	35.06
171	44.64

	Disponibilité de protéines en quantité (g/personne/jour) \
0	46.05
1	48.94
2	51.96
3	66.94
4	39.90
..	...
167	34.83
168	50.56
169	45.05
170	45.99
171	36.56

	Disponibilité intérieure	Exportations - Quantité \
0	1.088400e+10	2.780000e+08
1	5.343900e+10	1.058600e+10
2	3.326000e+09	8.400000e+07
3	3.551000e+10	6.490000e+08
4	1.268660e+11	3.445400e+10
..
167	2.563700e+10	4.500000e+07
168	8.361700e+10	2.006900e+10
169	8.017000e+09	5.420000e+08
170	9.537000e+09	6.630000e+08
171	8.615000e+09	2.700000e+08

	Importations - Quantité	Nourriture	Pertes	Production \
0	2.638000e+09	8.293000e+09	1.072000e+09	8.891000e+09
1	6.468000e+09	2.244100e+10	2.110000e+09	5.564900e+10
2	8.460000e+08	2.241000e+09	1.820000e+08	2.609000e+09
3	1.467200e+10	2.479700e+10	3.587000e+09	2.196400e+10
4	5.064200e+10	4.707900e+10	3.717000e+09	1.109260e+11
..
167	7.243000e+09	1.345800e+10	1.109000e+09	1.956500e+10
168	6.534000e+09	4.215300e+10	6.728000e+09	9.697200e+10
169	5.900000e+09	7.413000e+09	2.510000e+08	3.078000e+09
170	2.930000e+08	4.600000e+09	2.350000e+08	1.009600e+10
171	1.180000e+09	3.947000e+09	1.630000e+08	7.174000e+09

	Semences	Traitement	Variation de stock
0	3.930000e+08	6.300000e+07	-3.680000e+08
1	1.860000e+08	2.281900e+10	1.912000e+09
2	5.500000e+07	1.130000e+08	-4.500000e+07
3	4.280000e+08	6.370000e+08	-4.790000e+08
4	1.475000e+09	3.968600e+10	-2.470000e+08
..
167	6.000000e+07	7.957000e+09	-1.130000e+09

```

168  1.187000e+09  1.931100e+10          1.780000e+08
169  5.600000e+07  3.900000e+07          -4.180000e+08
170  7.300000e+07  4.178000e+09          -1.820000e+08
171  7.200000e+07  4.330000e+09          5.310000e+08

```

[172 rows x 17 columns]

```

[55]: #Calcul du nombre de kcal disponible pour les végétaux
dispo_alimentaire_vegetal_sum = dispo_alimentaire_vegetal_sum.
    ↪rename(columns={"Disponibilité alimentaire (Kcal/personne/jour)":
    ↪"dispo_kcal"})
dispo_kcal_monde_vegetal =
    ↪dispo_alimentaire_vegetal_sum["dispo_kcal"]*dispo_alimentaire_vegetal_sum["Population"]
print("La disponibilité alimentaire végétale est de",
    ↪round(dispo_kcal_monde_vegetal.sum()/1000000000),
    ↪"Tcal/jour (milliards de calories par jour) dans le monde")

```

La disponibilité alimentaire végétale est de 17261 Tcal/jour (milliards de calories par jour) dans le monde

```

[56]: #Calcul du nombre d'humains pouvant être nourris avec les végétaux
print("Le nombre théorique d'humains pouvant être nourris avec des végétaux est,
    ↪de", (dispo_kcal_monde_vegetal.sum()/2350/1000000000).round(2),"milliards.")

```

Le nombre théorique d'humains pouvant être nourris avec des végétaux est de 7.35 milliards.

[]:

3.4 - Utilisation de la disponibilité intérieure

```

[58]: #Calcul de la disponibilité totale
dispo_interieur_sum = dispo_alimentaire["Disponibilité intérieure"].sum()
print("La disponibilité totale est de", round(dispo_interieur_sum/
    ↪1000000000),"Mt.")

```

La disponibilité totale est de 9849 Mt.

```

[59]: #création d'une boucle for pour afficher les différentes valeurs en fonction
    ↪des colonnes aliments pour animaux, pertes, nourritures,
dispo_interieur_detail = ["Aliments pour animaux", "Autres
    ↪Utilisations", "Nourriture", "Pertes", "Semences", "Traitement"]
for col in dispo_interieur_detail:
    sum_col = (dispo_alimentaire[col].sum()/1000000000).round()
    print(f"La disponibilité de '{col}' est : {sum_col} Mt")

```

La disponibilité de 'Aliments pour animaux' est : 1304.0 Mt

La disponibilité de 'Autres Utilisations' est : 865.0 Mt

La disponibilité de 'Nourriture' est : 4876.0 Mt

La disponibilité de 'Pertes' est : 454.0 Mt

La disponibilité de 'Semences' est : 155.0 Mt
 La disponibilité de 'Traitement' est : 2205.0 Mt

```
[60]: for col in dispo_interieur_detail:
        sum_col = (dispo_alimentaire[col].sum()/10000000000).round()
        print(f"La disponibilité de '{col}' représente : {(sum_col/
        ↪(dispo_interieur_sum/10000000000)*100).round(1)} %")
```

La disponibilité de 'Aliments pour animaux' représente : 13.2 %
 La disponibilité de 'Autres Utilisations' représente : 8.8 %
 La disponibilité de 'Nourriture' représente : 49.5 %
 La disponibilité de 'Pertes' représente : 4.6 %
 La disponibilité de 'Semences' représente : 1.6 %
 La disponibilité de 'Traitement' représente : 22.4 %

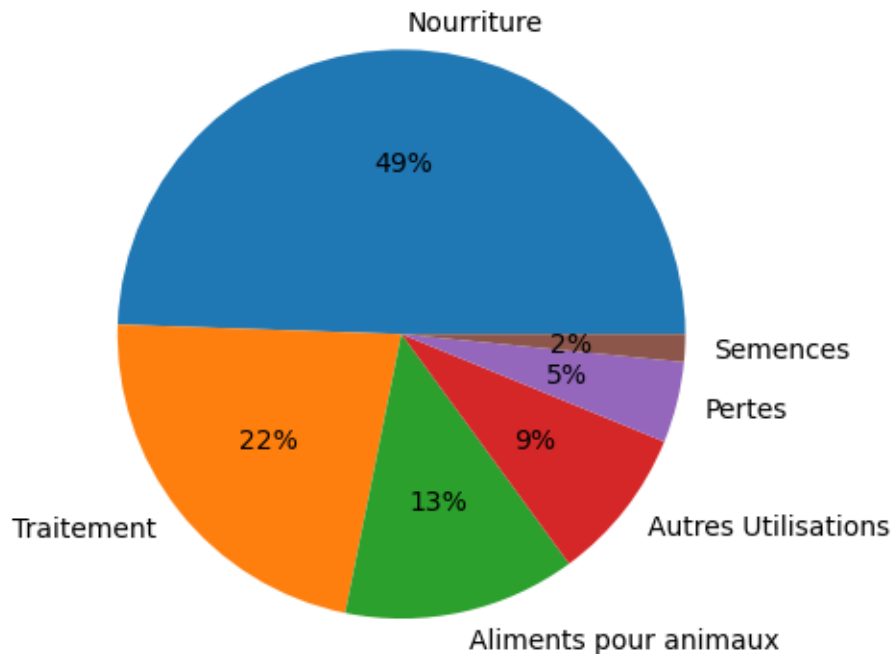
```
[61]: #Calculs pour graphiques 4)
dispo_interieur_detail_data = dispo_alimentaire[dispo_interieur_detail]
dispo_interieur_detail_data=dispo_interieur_detail_data.sum().reset_index().
    ↪sort_values(by=0, ascending=False)
dispo_interieur_detail_data
```

```
[61]:
```

	index	0
2	Nourriture	4.876258e+12
5	Traitement	2.204687e+12
0	Aliments pour animaux	1.304245e+12
1	Autres Utilisations	8.650230e+11
3	Pertes	4.536980e+11
4	Semences	1.546810e+11

```
[62]: plt.
    ↪pie(x=dispo_interieur_detail_data[0],labels=dispo_interieur_detail_data["index"],autopct='%
    ↪f%%')
plt.title("Répartition de la disponibilité intérieure en 2017")
plt.show()
```


Répartition de la disponibilité intérieure en 2017



3.5 - Utilisation des céréales

```
[64]: dispo_alimentaire_vegetale = (dispo_alimentaire.
      ↪loc[dispo_alimentaire["Origine"]=="vegetale"])
liste_produit_vegetal = dispo_alimentaire_vegetale["Produit"].unique()
print(liste_produit_vegetal)
```

```
['Agrumes, Autres' 'Aliments pour enfants' 'Ananas' 'Bananes' 'Bière'
 'Blé' 'Boissons Alcooliques' 'Café' 'Coco (Incl Coprah)'
 'Céréales, Autres' 'Dattes' 'Edulcorants Autres' 'Fève de Cacao'
 'Fruits, Autres' 'Graines de coton' 'Graines de tournesol'
 'Huile Plantes Oleif Autr' 'Huile Graines de Coton' 'Huile d'Arachide'
 'Huile d'Olive' 'Huile de Colza&Moutarde' 'Huile de Palme'
 'Huile de Soja' 'Huile de Sésame' 'Huile de Tournesol' 'Légumes, Autres'
 'Légumineuses Autres' 'Maïs' 'Miel' 'Millet' 'Miscellanees' 'Noix'
 'Olives' 'Oranges, Mandarines' 'Orge' 'Plantes Oleiferes, Autre' 'Poivre'
 'Pommes' 'Pommes de Terre' 'Raisin' 'Riz (Eq Blanchi)' 'Sucre Eq Brut'
 'Sucre, betterave' 'Sucre, canne' 'Sésame' 'Thé' 'Tomates' 'Vin'
 'Épices, Autres' 'Alcool, non Comestible' 'Arachides Decortiquees'
 'Avoine' 'Bananes plantains' 'Boissons Fermentés' 'Citrons & Limes'
 'Girofles' 'Graines Colza/Moutarde' 'Haricots' 'Huile de Coco'
 'Huile de Germe de Maïs' 'Huile de Palmistes' 'Ignames' 'Manioc'
 'Oignons' 'Palmistes' 'Pamplemousse' 'Patates douces' 'Piments' 'Pois']
```

```
'Racines nda' 'Seigle' 'Soja' 'Sorgho' 'Huile de Son de Riz'
'Sucre non centrifugé']
```

```
[65]: #Création d'une liste avec toutes les variables
liste_cereales = ["Blé", "Céréales, Autres", 'Maïs', 'Millet', 'Orge', 'Riz (Eq
↳Blanchi)', 'Avoine', 'Seigle', 'Sorgho']
```

```
[66]: #Création d'un dataframe avec les informations uniquement pour ces céréales
dispo_alimentaire_cereales = dispo_alimentaire[dispo_alimentaire["Produit"].
↳isin(liste_cereales)]
dispo_alimentaire_cereales
```

```
[66]:
```

	Zone	Produit	Origine	Aliments pour animaux	\
7	Afghanistan	Blé	vegetale	0.0	
12	Afghanistan	Céréales, Autres	vegetale	0.0	
32	Afghanistan	Maïs	vegetale	200000000.0	
34	Afghanistan	Millet	vegetale	0.0	
40	Afghanistan	Orge	vegetale	360000000.0	
...	
15545	Îles Salomon	Céréales, Autres	vegetale	0.0	
15568	Îles Salomon	Maïs	vegetale	0.0	
15575	Îles Salomon	Orge	vegetale	0.0	
15591	Îles Salomon	Riz (Eq Blanchi)	vegetale	0.0	
15593	Îles Salomon	Sorgho	vegetale	0.0	

	Autres Utilisations	Disponibilité alimentaire (Kcal/personne/jour)	\
7	0.0	1369.0	
12	0.0	0.0	
32	0.0	21.0	
34	0.0	3.0	
40	0.0	26.0	
...	
15545	0.0	0.0	
15568	0.0	1.0	
15575	0.0	0.0	
15591	12000000.0	623.0	
15593	0.0	0.0	

	Disponibilité alimentaire en quantité (kg/personne/an)	\
7	160.23	
12	0.00	
32	2.50	
34	0.40	
40	2.92	
...	...	
15545	0.00	
15568	0.15	

15575	0.07
15591	63.76
15593	0.00

	Disponibilité de matière grasse en quantité (g/personne/jour) \
7	4.69
12	0.00
32	0.30
34	0.02
40	0.24
...	...
15545	0.00
15568	0.01
15575	0.00
15591	1.36
15593	0.00

	Disponibilité de protéines en quantité (g/personne/jour) \
7	36.91
12	0.00
32	0.56
34	0.08
40	0.79
...	...
15545	0.00
15568	0.03
15575	0.01
15591	10.90
15593	0.00

	Disponibilité intérieure	Exportations - Quantité \
7	5.992000e+09	0.0
12	0.000000e+00	0.0
32	3.130000e+08	0.0
34	1.300000e+07	0.0
40	5.240000e+08	0.0
...
15545	0.000000e+00	0.0
15568	0.000000e+00	0.0
15575	1.000000e+06	0.0
15591	4.900000e+07	0.0
15593	0.000000e+00	0.0

	Importations - Quantité	Nourriture	Pertes	Production \
7	1.173000e+09	4.895000e+09	775000000.0	5.169000e+09
12	0.000000e+00	0.000000e+00	0.0	0.000000e+00
32	1.000000e+06	7.600000e+07	31000000.0	3.120000e+08

34	0.000000e+00	1.200000e+07	1000000.0	1.300000e+07
40	1.000000e+07	8.900000e+07	52000000.0	5.140000e+08
...
15545	0.000000e+00	0.000000e+00	0.0	0.000000e+00
15568	0.000000e+00	0.000000e+00	0.0	0.000000e+00
15575	1.000000e+06	0.000000e+00	0.0	0.000000e+00
15591	4.700000e+07	3.600000e+07	1000000.0	3.000000e+06
15593	0.000000e+00	0.000000e+00	0.0	0.000000e+00

	Semences	Traitement	Variation de stock
7	322000000.0	0.0	-350000000.0
12	0.0	0.0	0.0
32	5000000.0	0.0	0.0
34	0.0	0.0	0.0
40	22000000.0	0.0	0.0
...
15545	0.0	0.0	0.0
15568	0.0	0.0	0.0
15575	0.0	1000000.0	0.0
15591	0.0	0.0	0.0
15593	0.0	0.0	0.0

[1497 rows x 18 columns]

```
[67]: #Affichage de la proportion d'alimentation animale
dispo_interieur_cereal_sum = dispo_alimentaire_cereales["Disponibilité_
↳intérieure"].sum()
cereales_alimentation_animale_sum = dispo_alimentaire_cereales["Aliments pour_
↳animaux"].sum()
print("La proportion de céréales dans l'alimentation animale est de :",
      round(cereales_alimentation_animale_sum/dispo_interieur_cereal_sum*100),_
↳"%")
```

La proportion de céréales dans l'alimentation animale est de : 36 %

```
[68]: #Affichage de la proportion d'alimentation animale
cereales_nourriture_sum = dispo_alimentaire_cereales["Nourriture"].sum()
print("La proportion decéréales dans la nourriture à destionantion humaine est_
↳de :",
      round(cereales_nourriture_sum/dispo_interieur_cereal_sum*100), "%")
```

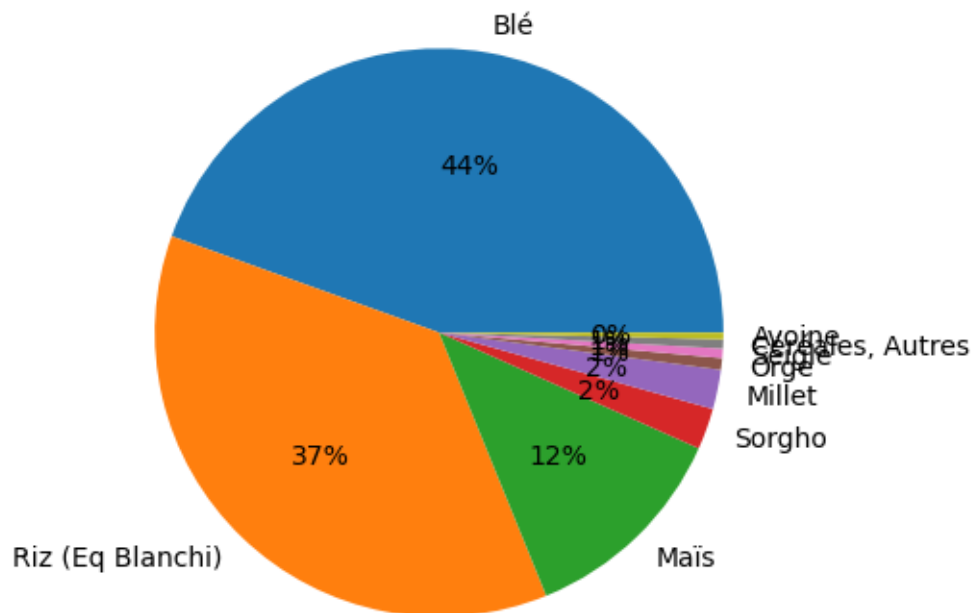
La proportion decéréales dans la nourriture à destionantion humaine est de : 43 %

```
[69]: #graphique pour powerpoint
cereales_nourriture_sum_detail = dispo_alimentaire_cereales.
↳groupby(["Produit"])["Nourriture"].sum()
```

```
cereales_nourriture_sum_detail_index = cereales_nourriture_sum_detail.  
    ↪reset_index().sort_values(by="Nourriture", ascending=False)
```

```
[70]: plt.  
    ↪pie(x=cereales_nourriture_sum_detail_index["Nourriture"],labels=cereales_nourriture_sum_det  
    ↪f'%')  
plt.title("Répartition des céréales dans l'alimentaion humain")  
plt.show()
```

Répartition des céréales dans l'alimentaion humain



```
[71]: #Tableau pour power point  
tableau_cereales_animaux_nourriture_sum = (dispo_alimentaire_cereales.  
    ↪groupby(["Produit"])[["Aliments pour animaux", "Nourriture"]].sum()/  
    ↪10000000000).round(1)  
tableau_cereales_animaux_nourriture_sum
```

```
[71]:
```

	Aliments pour animaux	Nourriture
Produit		
Avoine	16.3	3.9
Blé	129.7	457.8
Céréales, Autres	19.0	5.3
Maïs	546.1	125.2
Millet	3.3	23.0

Orge	92.7	6.8
Riz (Eq Blanche)	33.6	377.3
Seigle	8.1	5.5
Sorgho	24.8	24.2

3.6 - Pays avec la proportion de personnes sous-alimentée la plus forte en 2017

```
[73]: population_sous_nutrition_2017
```

```
[73]:
```

	Zone	Population	sous_nutrition
0	Afghanistan	36296113.0	10500000.0
1	Afrique du Sud	57009756.0	3100000.0
2	Albanie	2884169.0	100000.0
3	Algérie	41389189.0	1300000.0
4	Allemagne	82658409.0	0.0
..
198	Venezuela (République bolivarienne du)	29402484.0	8000000.0
199	Viet Nam	94600648.0	6500000.0
200	Yémen	27834819.0	0.0
201	Zambie	16853599.0	0.0
202	Zimbabwe	14236595.0	0.0

[203 rows x 3 columns]

```
[74]: #Création de la colonne proportion par pays
population_sous_nutrition_2017["Proportion"] =_
    (population_sous_nutrition_2017["sous_nutrition"]
     /
     population_sous_nutrition_2017["Population"]*100).round(2)
#Attention, 119 pays sans problème de sous nutrition, on peut se poser des_
    questions sur la qualité des données (ex: Yémen...)
print(population_sous_nutrition_2017[population_sous_nutrition_2017["sous_nutrition"]==0])
```

	Zone	Population	sous_nutrition	Proportion
4	Allemagne	82658409.0	0.0	0.0
5	Andorre	77001.0	0.0	0.0
7	Antigua-et-Barbuda	95426.0	0.0	0.0
10	Arménie	2944791.0	0.0	0.0
11	Australie	24584620.0	0.0	0.0
..
196	Uruguay	3436641.0	0.0	0.0
197	Vanuatu	285510.0	0.0	0.0
200	Yémen	27834819.0	0.0	0.0
201	Zambie	16853599.0	0.0	0.0
202	Zimbabwe	14236595.0	0.0	0.0

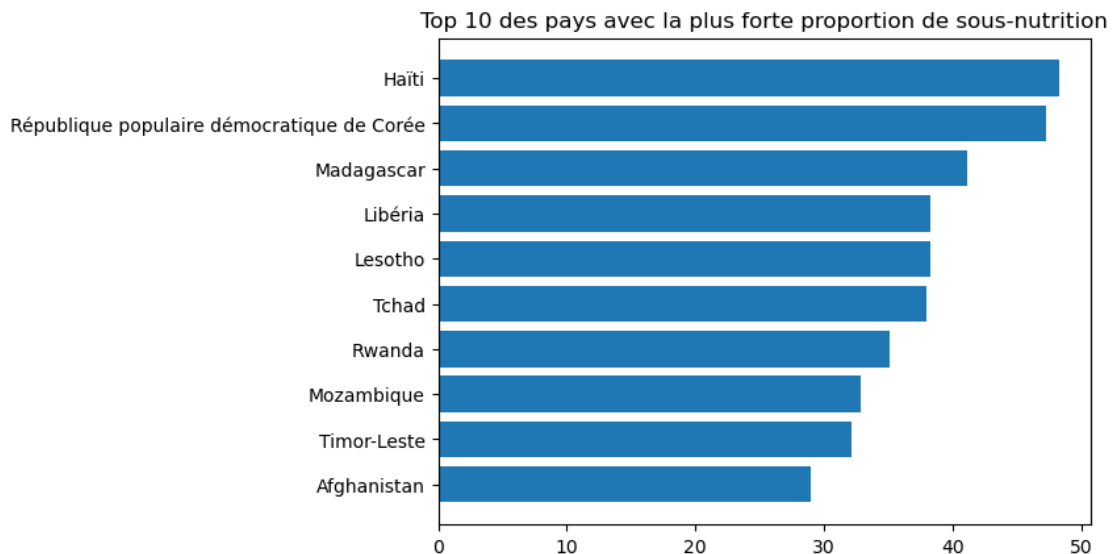
[119 rows x 4 columns]

```
[75]: #affichage après trie des 10 pires pays
classement_pays_sous_nutrition = population_sous_nutrition_2017.
↳sort_values(by="Proportion", ascending=False)
print(classement_pays_sous_nutrition[["Zone", "Proportion"]].head(10))
```

	Zone	Proportion
78	Haïti	48.26
157	République populaire démocratique de Corée	47.19
108	Madagascar	41.06
103	Libéria	38.28
100	Lesotho	38.25
183	Tchad	37.96
161	Rwanda	35.06
121	Mozambique	32.81
186	Timor-Leste	32.17
0	Afghanistan	28.93

```
[ ]:
```

```
[76]: #Graphique pour power point
classement_pays_sous_nutrition_inverse = classement_pays_sous_nutrition.
↳head(10).sort_values(by="Proportion", ascending=True)
plt.barh(width=classement_pays_sous_nutrition_inverse['Proportion'],
         y=classement_pays_sous_nutrition_inverse['Zone'])
plt.title("Top 10 des pays avec la plus forte proportion de sous-nutrition")
plt.show()
```



3.7 - Pays qui ont le plus bénéficié d'aide alimentaire depuis 2013

```
[78]: aide_alimentaire
```

```
[78]:
```

	Zone	Année	Produit	Valeur
0	Afghanistan	2013	Autres non-céréales	682000
1	Afghanistan	2014	Autres non-céréales	335000
2	Afghanistan	2013	Blé et Farin	39224000
3	Afghanistan	2014	Blé et Farin	15160000
4	Afghanistan	2013	Céréales	40504000
...
1470	Zimbabwe	2015	Mélanges et préparations	96000
1471	Zimbabwe	2013	Non-céréales	5022000
1472	Zimbabwe	2014	Non-céréales	2310000
1473	Zimbabwe	2015	Non-céréales	306000
1474	Zimbabwe	2013	Riz, total	64000

[1475 rows x 4 columns]

```
[79]: #calcul du total de l'aide alimentaire par pays
aide_alimentaire_pays = aide_alimentaire.groupby("Zone").sum()
aide_alimentaire_pays_reset = aide_alimentaire_pays.reset_index()
aide_alimentaire_zetv = aide_alimentaire_pays_reset.loc[:,["Zone", "Valeur"]]
print(aide_alimentaire_zetv)
```

	Zone	Valeur
0	Afghanistan	185452000
1	Algérie	81114000
2	Angola	5014000
3	Bangladesh	348188000
4	Bhoutan	2666000
..
71	Zambie	3026000
72	Zimbabwe	62570000
73	Égypte	1122000
74	Équateur	1362000
75	Éthiopie	1381294000

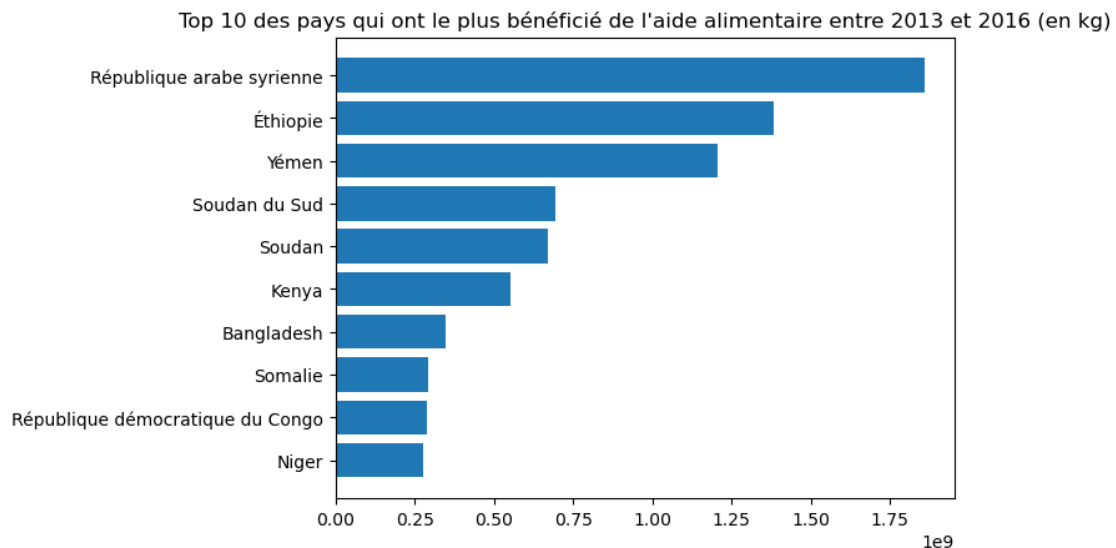
[76 rows x 2 columns]

```
[80]: #affichage après trie des 10 pays qui ont bénéficié le plus de l'aide_
      ↪ alimentaire
top_aide_alimentaire = aide_alimentaire_zetv.sort_values(by="Valeur",
      ↪ ascending=False)
print(top_aide_alimentaire.head(10))
```

	Zone	Valeur
50	République arabe syrienne	1858943000
75	Éthiopie	1381294000
70	Yémen	1206484000

61	Soudan du Sud	695248000
60	Soudan	669784000
30	Kenya	552836000
3	Bangladesh	348188000
59	Somalie	292678000
53	République démocratique du Congo	288502000
43	Niger	276344000

```
[81]: #graphique pour power point
top_aide_alimentaire_inverse = top_aide_alimentaire.head(10).
    ↪sort_values(by="Valeur", ascending=True)
plt.barh(width=top_aide_alimentaire_inverse['Valeur'],
        y=top_aide_alimentaire_inverse['Zone'])
plt.title("Top 10 des pays qui ont le plus bénéficié de l'aide alimentaire_
    ↪entre 2013 et 2016 (en kg)")
plt.show()
```



3.8 - Evolution des 5 pays qui ont le plus bénéficiés de l'aide alimentaire entre 2013 et 2016

```
[83]: #Création d'un dataframe avec la zone, l'année et l'aide alimentaire puis_
    ↪groupby sur zone et année
aide_alimentaire_pays_année = aide_alimentaire.loc[:,["Zone","Année","Valeur"]].
    ↪groupby(["Zone","Année"]).sum()

aide_alimentaire_pays_année
```

```
[83]:
Zone      Année      Valeur
Afghanistan 2013  128238000
```

	2014	57214000
Algérie	2013	35234000
	2014	18980000
	2015	17424000
...	...	
Égypte	2013	1122000
Équateur	2013	1362000
Éthiopie	2013	591404000
	2014	586624000
	2015	203266000

[228 rows x 1 columns]

```
[84]: #Création d'une liste contenant les 5 pays qui ont le plus bénéficiées de
      ↪ l'aide alimentaire
```

```
liste_top5_aide_alimentaire = ["République arabe
      ↪ syrienne", "Éthiopie", "Yémen", "Soudan du Sud", "Soudan"]
```

```
[85]: #On filtre sur le dataframe avec notre liste
```

```
aide_alimentaire_pays_année_reset = aide_alimentaire_pays_année.reset_index()
aide_alimentaire_pays_année_top5 =
      ↪ aide_alimentaire_pays_année_reset[aide_alimentaire_pays_année_reset["Zone"]
      ↪ isin(liste_top5_aide_alimentaire)]
aide_alimentaire_pays_année_top5
```

```
[85]:
```

	Zone	Année	Valeur
157	République arabe syrienne	2013	563566000
158	République arabe syrienne	2014	651870000
159	République arabe syrienne	2015	524949000
160	République arabe syrienne	2016	118558000
189	Soudan	2013	330230000
190	Soudan	2014	321904000
191	Soudan	2015	17650000
192	Soudan du Sud	2013	196330000
193	Soudan du Sud	2014	450610000
194	Soudan du Sud	2015	48308000
214	Yémen	2013	264764000
215	Yémen	2014	103840000
216	Yémen	2015	372306000
217	Yémen	2016	465574000
225	Éthiopie	2013	591404000
226	Éthiopie	2014	586624000
227	Éthiopie	2015	203266000

```
[86]: # Affichage des pays avec l'aide alimentaire par année
```

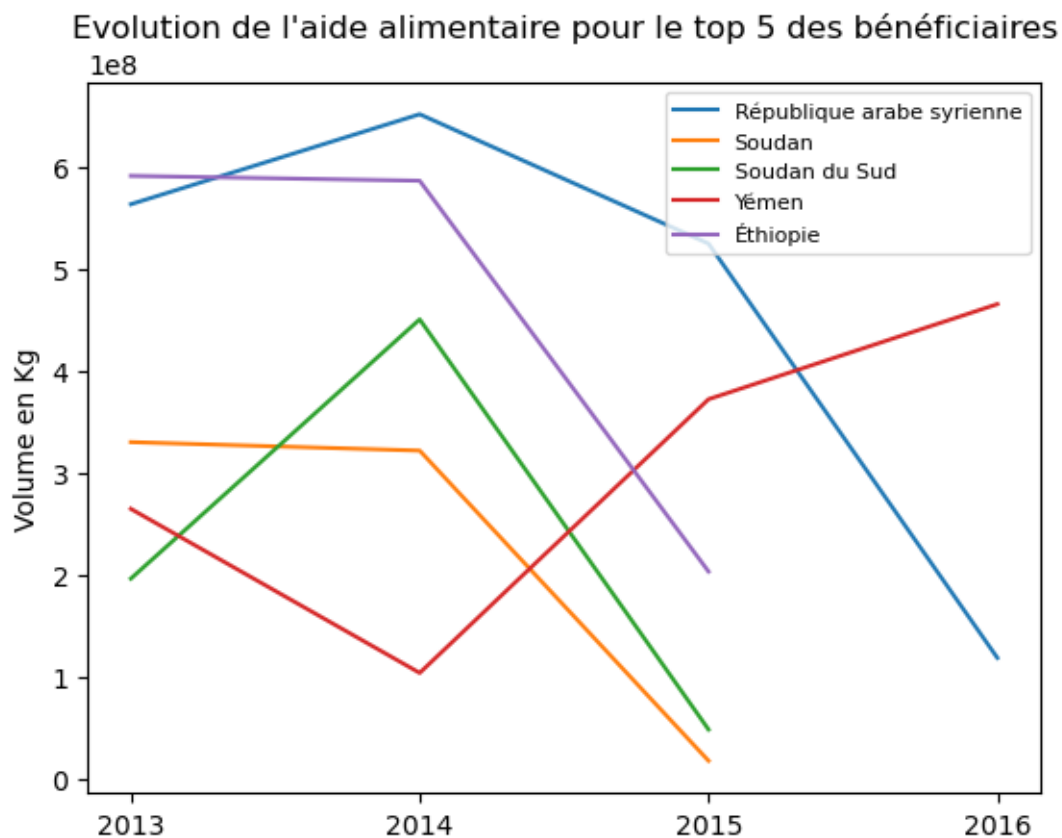
```
aide_alimentaire_pays_année_pivot_top5 = aide_alimentaire_pays_année_top5.
      ↪ pivot_table(index="Zone", columns="Année", values="Valeur")
```

```
aide_alimentaire_pays_année_pivot_top5
```

```
[86]: Année                2013                2014                2015                2016
Zone
République arabe syrienne  563566000.0  651870000.0  524949000.0  118558000.0
Soudan                    330230000.0  321904000.0   17650000.0         NaN
Soudan du Sud             196330000.0  450610000.0  48308000.0         NaN
Yémen                     264764000.0  103840000.0  372306000.0  465574000.0
Éthiopie                   591404000.0  586624000.0  203266000.0         NaN
```

```
[87]: #graphique pour power point
for zone in aide_alimentaire_pays_année_top5['Zone'].unique():
    subset_zone =
    aide_alimentaire_pays_année_top5[aide_alimentaire_pays_année_top5['Zone'] ==
    zone]
    plt.plot(subset_zone['Année'], subset_zone['Valeur'],label=zone)

plt.title("Evolution de l'aide alimentaire pour le top 5 des bénéficiaires")
plt.legend(fontsize=8)
plt.xticks([2013,2014,2015,2016])
plt.ylabel("Volume en Kg")
plt.show()
```



3.9 - Pays avec le moins de disponibilité par habitant

[89]: population_dispo_alimentaire

```
[89]:
```

	Zone	Population	\
0	Afghanistan	36296113.0	
1	Afrique du Sud	57009756.0	
2	Albanie	2884169.0	
3	Algérie	41389189.0	
4	Allemagne	82658409.0	
..	
167	Venezuela (République bolivarienne du)	29402484.0	
168	Viet Nam	94600648.0	
169	Yémen	27834819.0	
170	Zambie	16853599.0	
171	Zimbabwe	14236595.0	

	Aliments pour animaux	Autres Utilisations	dispo_kcal	\
0	7.680000e+08	4.150000e+08	2087.0	
1	5.309000e+09	8.760000e+08	3020.0	
2	6.600000e+08	1.740000e+08	3188.0	
3	4.352000e+09	1.707000e+09	3293.0	
4	3.020900e+10	7.711000e+09	3503.0	
..	
167	2.878000e+09	3.350000e+08	2633.0	
168	1.135600e+10	3.815000e+09	2744.0	
169	2.100000e+08	7.200000e+07	2217.0	
170	4.080000e+08	6.900000e+07	1924.0	
171	1.310000e+08	2.300000e+07	2113.0	

	Disponibilité alimentaire en quantité (kg/personne/an)	\
0	351.41	
1	564.80	
2	1095.54	
3	809.13	
4	957.52	
..	...	
167	652.86	
168	574.84	
169	375.41	
170	350.12	
171	336.59	

	Disponibilité de matière grasse en quantité (g/personne/jour)	\
0	33.50	

1	83.58
2	106.48
3	76.48
4	141.62
..	...
167	82.44
168	72.11
169	47.81
170	42.05
171	57.18

Disponibilité de protéines en quantité (g/personne/jour) \	
0	58.26
1	85.33
2	111.37
3	91.92
4	101.39
..	...
167	72.96
168	81.61
169	58.90
170	55.22
171	48.32

Disponibilité intérieure Exportations - Quantité \		
0	1.351500e+10	2.780000e+08
1	6.125600e+10	1.167400e+10
2	4.758000e+09	9.400000e+07
3	4.263000e+10	6.550000e+08
4	1.622750e+11	5.743100e+10
..
167	3.228700e+10	6.200000e+07
168	9.513400e+10	2.263800e+10
169	9.782000e+09	7.190000e+08
170	1.004100e+10	6.750000e+08
171	9.507000e+09	2.780000e+08

Importations - Quantité	Nourriture	Pertes	Production \
0	2.988000e+09	1.073500e+10	1.135000e+09 1.117100e+10
1	7.517000e+09	2.981200e+10	2.193000e+09 6.326300e+10
2	9.330000e+08	3.476000e+09	2.760000e+08 3.964000e+09
3	1.740300e+10	3.172900e+10	3.753000e+09 2.635900e+10
4	6.535300e+10	7.923800e+10	3.781000e+09 1.545470e+11
..
167	9.604000e+09	1.985300e+10	1.190000e+09 2.386800e+10
168	9.217000e+09	5.270000e+10	6.743000e+09 1.082750e+11
169	6.786000e+09	9.162000e+09	2.690000e+08 4.125000e+09

170	3.800000e+08	5.088000e+09	2.420000e+08	1.052600e+10
171	1.342000e+09	4.762000e+09	1.880000e+08	7.914000e+09

	Semences	Traitement	Variation de stock
0	3.950000e+08	6.300000e+07	-3.680000e+08
1	2.530000e+08	2.281900e+10	2.153000e+09
2	5.500000e+07	1.130000e+08	-4.500000e+07
3	4.490000e+08	6.370000e+08	-4.790000e+08
4	1.551000e+09	3.984200e+10	-1.920000e+08
..
167	9.100000e+07	7.957000e+09	-1.130000e+09
168	1.211000e+09	1.931100e+10	2.780000e+08
169	6.700000e+07	3.900000e+07	-4.080000e+08
170	7.600000e+07	4.178000e+09	-1.820000e+08
171	7.600000e+07	4.330000e+09	5.310000e+08

[172 rows x 17 columns]

```
[90]: #Calcul de la disponibilité en kcal par personne par jour par pays
print(population_dispo_alimentaire[["Zone","dispo_kcal"]])
```

	Zone	dispo_kcal
0	Afghanistan	2087.0
1	Afrique du Sud	3020.0
2	Albanie	3188.0
3	Algérie	3293.0
4	Allemagne	3503.0
..
167	Venezuela (République bolivarienne du)	2633.0
168	Viet Nam	2744.0
169	Yémen	2217.0
170	Zambie	1924.0
171	Zimbabwe	2113.0

[172 rows x 2 columns]

```
[91]: #Affichage des 10 pays qui ont le moins de dispo alimentaire par personne
top_min_dispo_alimentaire = population_dispo_alimentaire.
    ↪ sort_values(by="dispo_kcal", ascending=True)
print(top_min_dispo_alimentaire[["Zone","dispo_kcal"]].head(10))
```

	Zone	dispo_kcal
131	République centrafricaine	1879.0
170	Zambie	1924.0
97	Madagascar	2056.0
0	Afghanistan	2087.0
70	Haïti	2089.0
136	République populaire démocratique de Corée	2093.0

156	Tchad	2109.0
171	Zimbabwe	2113.0
120	Ouganda	2126.0
55	Éthiopie	2129.0

```
[92]: #Graphique pour power point
plt.barh(width=top_min_dispo_alimentaire[["Zone","dispo_kcal"]].
↳head(10)['dispo_kcal'],
        y=top_min_dispo_alimentaire[["Zone","dispo_kcal"]].head(10)['Zone'])
plt.title("Top 10 des pays avec la plus faible disponibilité alimentaire par_
↳habitant")
plt.show()
```

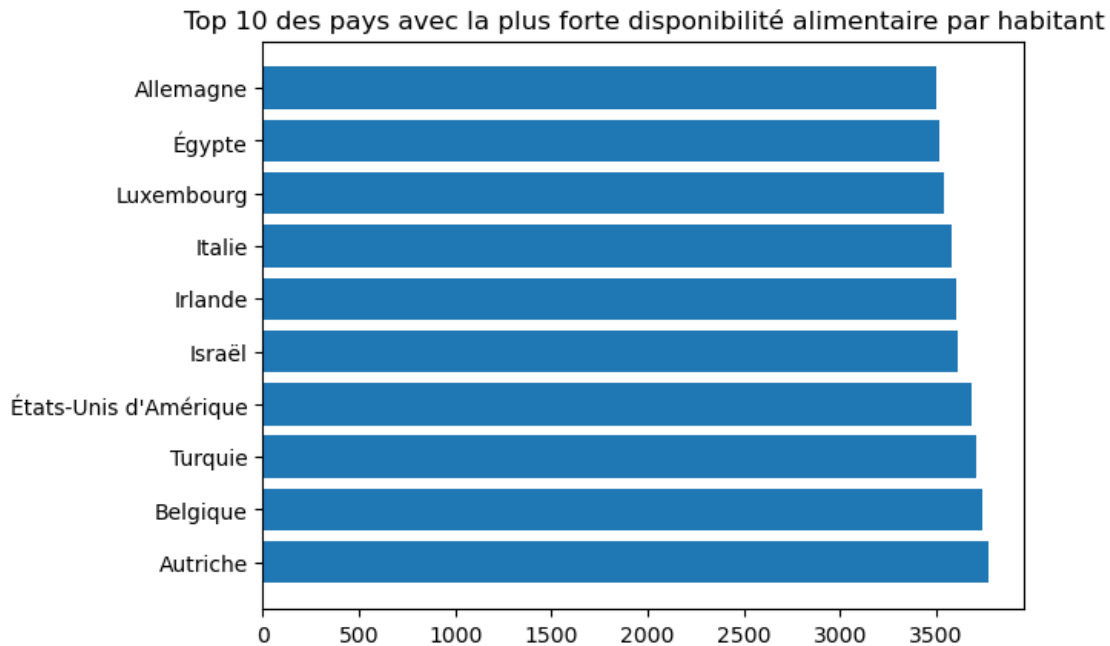


3.10 - Pays avec le plus de disponibilité par habitant

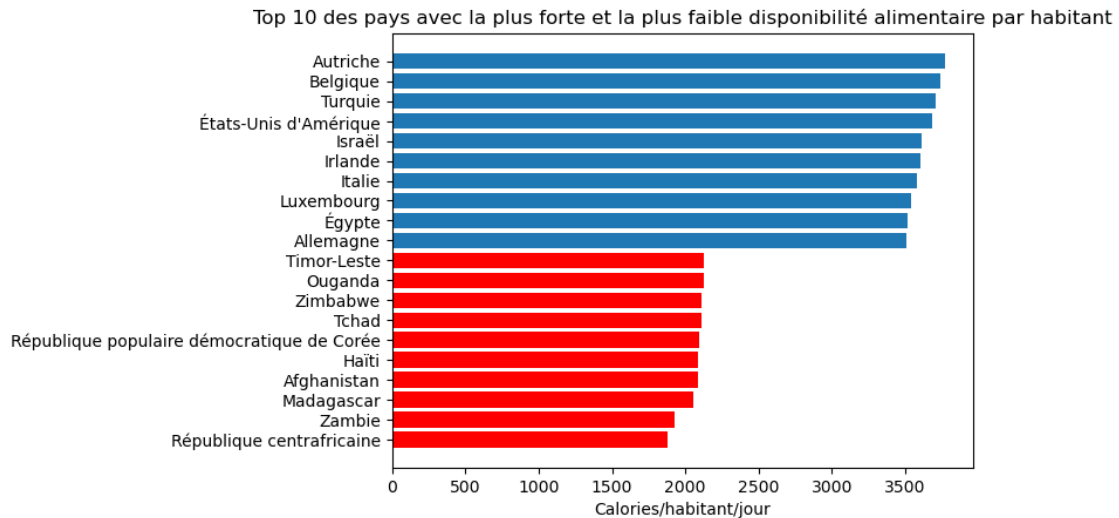
```
[94]: #Affichage des 10 pays qui ont le plus de dispo alimentaire par personne
top_dispo_alimentaire = population_dispo_alimentaire.
↳sort_values(by="dispo_kcal", ascending=False)
print(top_dispo_alimentaire[["Zone","dispo_kcal"]].head(10))
```

	Zone	dispo_kcal
11	Autriche	3770.0
17	Belgique	3737.0
163	Turquie	3708.0
54	États-Unis d'Amérique	3682.0
80	Israël	3610.0
78	Irlande	3602.0
81	Italie	3578.0
95	Luxembourg	3540.0
47	Égypte	3518.0

```
[95]: #Graphique pour power point
plt.barh(width=top_dispo_alimentaire[["Zone","dispo_kcal"]].
    ↪head(10)['dispo_kcal'],
        y=top_dispo_alimentaire[["Zone","dispo_kcal"]].head(10)['Zone'])
plt.title("Top 10 des pays avec la plus forte disponibilité alimentaire par_
    ↪habitant")
plt.show()
```



```
[96]: plt.barh(width=top_min_dispo_alimentaire[["Zone","dispo_kcal"]].head(10).
    ↪sort_values(by="dispo_kcal", ascending=True)['dispo_kcal'],
        y=top_min_dispo_alimentaire.sort_values(by="dispo_kcal",
    ↪ascending=True)[["Zone","dispo_kcal"]].head(10)['Zone'],color="red")
plt.barh(width=top_dispo_alimentaire[["Zone","dispo_kcal"]].head(10).
    ↪sort_values(by="dispo_kcal", ascending=True)['dispo_kcal'],
        y=top_dispo_alimentaire[["Zone","dispo_kcal"]].head(10).
    ↪sort_values(by="dispo_kcal", ascending=True)['Zone'])
plt.title("Top 10 des pays avec la plus forte et la plus faible disponibilité_
    ↪alimentaire par habitant")
plt.xlabel("Calories/habitant/jour")
plt.show()
```

3.11 - Exemple de la Thaïlande pour le Manioc

```
[98]: #création d'un dataframe avec uniquement la Thaïlande
dataframe_Thaïlande = pd.merge(population_sous_nutrition_2017,
    ↳dispo_alimentaire, on="Zone")
dataframe_Thaïlande = dataframe_Thaïlande[dataframe_Thaïlande["Zone"]=="Thaïlande"]
dataframe_Thaïlande
```

```
[98]:
```

	Zone	Population	sous_nutrition	Proportion \
14116	Thaïlande	69209810.0	6200000.0	8.96
14117	Thaïlande	69209810.0	6200000.0	8.96
14118	Thaïlande	69209810.0	6200000.0	8.96
14119	Thaïlande	69209810.0	6200000.0	8.96
14120	Thaïlande	69209810.0	6200000.0	8.96
...
14206	Thaïlande	69209810.0	6200000.0	8.96
14207	Thaïlande	69209810.0	6200000.0	8.96
14208	Thaïlande	69209810.0	6200000.0	8.96
14209	Thaïlande	69209810.0	6200000.0	8.96
14210	Thaïlande	69209810.0	6200000.0	8.96

	Produit	Origine	Aliments pour animaux \
14116	Abats Comestible	animale	0.0
14117	Agrumes, Autres	vegetale	0.0
14118	Alcool, non Comestible	vegetale	0.0
14119	Aliments pour enfants	vegetale	0.0
14120	Ananas	vegetale	0.0
...
14206	Viande de Suides	animale	0.0

14207	Viande de Volailles	animale	0.0
14208	Viande, Autre	animale	0.0
14209	Vin	vegetale	0.0
14210	Épices, Autres	vegetale	0.0

	Autres Utilisations	Disponibilité alimentaire (Kcal/personne/jour)	\
14116	0.0		3.0
14117	0.0		0.0
14118	358000000.0		0.0
14119	0.0		2.0
14120	0.0		10.0
...
14206	0.0		124.0
14207	0.0		52.0
14208	0.0		0.0
14209	0.0		0.0
14210	0.0		16.0

	Disponibilité alimentaire en quantité (kg/personne/an)	...	\
14116	1.11	...	
14117	0.09	...	
14118	0.00	...	
14119	0.18	...	
14120	10.02	...	
...	
14206	13.00	...	
14207	13.69	...	
14208	0.03	...	
14209	0.12	...	
14210	1.70	...	

	Disponibilité de protéines en quantité (g/personne/jour)	\
14116	0.56	
14117	0.00	
14118	0.00	
14119	0.08	
14120	0.08	
...	...	
14206	3.92	
14207	4.49	
14208	0.02	
14209	0.00	
14210	0.43	

	Disponibilité intérieure	Exportations - Quantité	\
14116	74000000.0	5.000000e+06	
14117	8000000.0	6.000000e+06	

14118	358000000.0	1.100000e+08
14119	12000000.0	7.000000e+06
14120	782000000.0	1.449000e+09
...
14206	871000000.0	2.200000e+07
14207	945000000.0	5.360000e+08
14208	-92000000.0	9.600000e+07
14209	8000000.0	8.000000e+06
14210	114000000.0	4.200000e+07

	Importations	- Quantité	Nourriture	Pertes	Production \
14116	33000000.0	75000000.0	0.0	4.500000e+07	
14117	2000000.0	6000000.0	0.0	1.200000e+07	
14118	21000000.0	0.0	0.0	4.470000e+08	
14119	19000000.0	12000000.0	0.0	0.000000e+00	
14120	9000000.0	671000000.0	110000000.0	2.209000e+09	
...	
14206	1000000.0	871000000.0	0.0	8.910000e+08	
14207	11000000.0	917000000.0	28000000.0	1.470000e+09	
14208	4000000.0	2000000.0	0.0	0.000000e+00	
14209	16000000.0	8000000.0	0.0	0.000000e+00	
14210	13000000.0	114000000.0	0.0	1.430000e+08	

	Semences	Traitement	Variation de stock
14116	0.0	0.0	0.0
14117	0.0	2000000.0	0.0
14118	0.0	0.0	0.0
14119	0.0	0.0	0.0
14120	0.0	0.0	13000000.0
...
14206	0.0	0.0	0.0
14207	0.0	0.0	0.0
14208	0.0	0.0	0.0
14209	0.0	0.0	0.0
14210	0.0	0.0	0.0

[95 rows x 21 columns]

```
[99]: #Calcul de la sous nutrition en Thaïlande
print("La sous nutrition en Thaïlande représente",
      dataframe_Thailand["Proportion"].unique(), "% de la population")
```

La sous nutrition en Thaïlande représente [8.96] % de la population

```
[100]: dataframe_Thailand_manioc =
      dataframe_Thailand[dataframe_Thailand["Produit"]=="Manioc"]
dataframe_Thailand_manioc
```



```
[104]: dataframe_Thailand_manioc.iloc[0,-9:] = dataframe_Thailand_manioc.iloc[0,-9:]/
↳1000000
dataframe_Thailand_manioc.iloc[0, 6:8] = dataframe_Thailand_manioc.iloc[0, 6:8]/
↳1000000
dataframe_Thailand_manioc
```

```
[104]:
```

	Zone	Population	sous_nutrition	Proportion	Produit	Origine	\
14166	Thaïlande	69209810.0	6200000.0	8.96	Manioc	vegetale	
	Aliments pour animaux		Autres Utilisations				\
14166		1800.0		2081.0			
	Disponibilité alimentaire (Kcal/personne/jour)						\
14166				40.0			
	Disponibilité alimentaire en quantité (kg/personne/an)			...			\
14166				13.0		...	
	Disponibilité de protéines en quantité (g/personne/jour)						\
14166				0.14			
	Disponibilité intérieure		Exportations - Quantité				\
14166		6264.0		25214.0			
	Importations - Quantité		Nourriture	Pertes	Production	Semences	\
14166		1250.0	871.0	1511.0	30228.0	0.0	
	Traitement		Variation de stock				
14166		0.0		0.0			

[1 rows x 21 columns]

```
[105]: nourriture_manioc = dataframe_Thailand_manioc.loc[14166, "Nourriture"]
exportations_manioc = dataframe_Thailand_manioc.loc[14166, "Exportations - Quantit  "]
ratio_nourriture_exportation = exportations_manioc/nourriture_manioc
print("Il y a", ratio_nourriture_exportation.round(), "fois plus de manioc  
  export   que de manioc qui sert    nourrir la population tha  landaise")

dispo_alimentaire_manioc = dataframe_Thailand_manioc.loc[14166, "Disponibilit  
  alimentaire (Kcal/personne/jour)"]
equivalent_exportation_manioc_dispo_alimentaire =   
  dispo_alimentaire_manioc*ratio_nourriture_exportation
proportion_besoin_alimentaire = equivalent_exportation_manioc_dispo_alimentaire/
  2940*100
print("Les exportations repr  sentent l'  quivalent de",   
  equivalent_exportation_manioc_dispo_alimentaire.round(),
```

```
"Kcal/personne/jour, c'est à dire,",proportion_besoin_alimentaire.round(),
"% des besoins alimentaires (en calories) de la population thaïlandaise")
```

Il y a 29.0 fois plus de manioc exporté que de manioc qui sert à nourrir la population thaïlandaise

Les exportations représentent l'équivalent de 1158.0 Kcal/personne/jour, c'est à dire, 39.0 % des besoins alimentaires (en calories) de la population thaïlandaise

Étape 6 - Analyse complémentaires

```
[107]: #Rajouter en dessous toutes les analyses complémentaires suite à la demande de
      ↪mélanie :
      #"et toutes les infos que tu trouverais utiles pour mettre en relief les pays
      ↪qui semblent être
      #le plus en difficulté au niveau alimentaire"

      #Recherche sur le rapport exportation/nourriture et sous alimentation
      top_min_dispo_alimentaire["Proportion exp/nourriture"] =
      ↪(top_min_dispo_alimentaire["Exportations - Quantité"/
      ↪top_min_dispo_alimentaire["Nourriture"]*100).round(1)
      Tableau_exp_nourriture = top_min_dispo_alimentaire[["Zone","Proportion exp/
      ↪nourriture","dispo_kcal"]]

      Tableau_exp_nourriture_reset = Tableau_exp_nourriture.reset_index()
```

```
[108]: Tableau_exp_nourriture_reset[Tableau_exp_nourriture_reset["Proportion exp/
      ↪nourriture"]>10].head(20)
```

```
[108]:
```

	index	Zone	Proportion exp/nourriture \
1	170	Zambie	13.3
11	111	Namibie	55.9
19	21	Bolivie (État plurinational de)	32.5
21	68	Guinée-Bissau	33.0
23	53	Eswatini	107.1
25	50	Équateur	93.3
27	73	Îles Salomon	32.6
30	66	Guatemala	96.5
32	122	Pakistan	10.2
35	145	Sénégal	10.3
39	29	Cambodge	12.8
48	127	Philippines	15.7
49	110	Myanmar	10.3
50	48	El Salvador	34.3
51	124	Paraguay	241.9
52	140	Sainte-Lucie	17.9
54	45	Djibouti	28.8
55	135	République dominicaine	11.2

59	113	Nicaragua	55.4
60	71	Honduras	53.0

	dispo_kcal
1	1924.0
11	2166.0
19	2250.0
21	2288.0
23	2326.0
25	2346.0
27	2383.0
30	2416.0
32	2438.0
35	2453.0
39	2473.0
48	2568.0
49	2574.0
50	2576.0
51	2584.0
52	2589.0
54	2609.0
55	2611.0
59	2638.0
60	2639.0

```
[109]: Tableau_exp_nourriture_reset[Tableau_exp_nourriture_reset["Proportion exp/
↳nourriture"]>100].head(10)
```

```
[109]:
```

	index	Zone	Proportion exp/nourriture	dispo_kcal
23	53	Eswatini	107.1	2326.0
51	124	Paraguay	241.9	2584.0
76	18	Belize	216.6	2746.0
81	157	Thaïlande	128.4	2785.0
86	26	Bulgarie	185.2	2828.0
89	40	Costa Rica	189.5	2846.0
94	98	Malaisie	158.5	2914.0
101	148	Slovaquie	104.4	2943.0
110	72	Hongrie	150.8	3034.0
113	165	Uruguay	307.7	3052.0

```
[110]: ajout_sous_nutrition = pd.
↳merge(Tableau_exp_nourriture_reset,population_sous_nutrition_2017[["Zone","Proportion"]],on
```

```
[111]: tableau_proportion_exp_nourriture_sous_nutrition =ajout_sous_nutrition.
↳rename(columns={"Proportion":"Proportion sous alimentation"})
```

```
[112]: tableau_proportion_expnourriture_sousnutrition[tableau_proportion_expnourriture_sousnutrition[
↳exp/nourriture"]>10].head(10)
```

```
[112]:      index      Zone  Proportion exp/nourriture  \
1      170      Zambie      13.3
11     111      Namibie      55.9
19     21  Bolivie (État plurinational de)      32.5
21     68      Guinée-Bissau      33.0
23     53      Eswatini      107.1
25     50      Équateur      93.3
27     73      Îles Salomon      32.6
30     66      Guatemala      96.5
32    122      Pakistan      10.2
35    145      Sénégal      10.3
```

```
      dispo_kcal  Proportion sous alimentation
1      1924.0      0.00
11     2166.0      12.49
19     2250.0      15.19
21     2288.0      0.00
23     2326.0      17.78
25     2346.0      8.94
27     2383.0      0.00
30     2416.0      15.96
32     2438.0      11.93
35     2453.0      9.73
```

```
[113]: tableau_proportion_expnourriture_sousnutrition[tableau_proportion_expnourriture_sousnutrition[
↳exp/nourriture"]>100].head(10)
```

```
[113]:      index      Zone  Proportion exp/nourriture  dispo_kcal  \
23     53      Eswatini      107.1      2326.0
51    124      Paraguay      241.9      2584.0
76     18      Belize      216.6      2746.0
81    157      Thaïlande      128.4      2785.0
86     26      Bulgarie      185.2      2828.0
89     40      Costa Rica      189.5      2846.0
94     98      Malaisie      158.5      2914.0
101   148      Slovaquie      104.4      2943.0
110    72      Hongrie      150.8      3034.0
113   165      Uruguay      307.7      3052.0
```

```
      Proportion sous alimentation
23      17.78
51      8.74
76      0.00
81      8.96
```


86	2.82
89	4.04
94	3.21
101	5.51
110	0.00
113	0.00