PROJET 4 DATA ANALYST

Réalisez une étude de santé publique avec R ou Python

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'un 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 a trouver ...

Etape 1 - Importation des librairies et chargement des fichiers

1.1 - Importation des librairies

In [1]:
```
#Importation de la librairie Pandas
```

```
import pandas as pd
```

1.2 - Chargement des fichiers Excel

In [2]:
```
#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

In [259…
```
#Afficher les dimensions du dataset
print("Le tableau comporte {} observation(s) ou article(s)".format(population.sh
```

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

Consulter le nombre de colonnes

print("Le tableau comporte {} colonne(s)".format(population.shape[1]))

La nature des données dans chacune des colonnes

print("Type de données des colonnes {}".format(population.dtypes))

Le nombre de valeurs présentes dans chacune des colonnes

nb_colonnes = population.count()

population.info()

In [13]:
```
#Affichage les 5 premières lignes de la table
population.head()
```

Out[13]:

	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

In [14]:
```
#Nous allons harmoniser les unités. Pour cela, nous avons décidé de multiplier l
#Multiplication de la colonne valeur par 1000
population['Valeur'] = population['Valeur'] * 1000
```

In [15]:
```
#changement du nom de la colonne Valeur par Population
population = population.rename(columns={'Valeur': 'Population'})
```

In [262…
```
#Affichage les 5 premières lignes de la table pour voir les modifications
population.head()
```

Out[262…

	Zone	Année	Population
0	Afghanistan	2013	32269589
1	Afghanistan	2014	33370794
2	Afghanistan	2015	34413603
3	Afghanistan	2016	35383032
4	Afghanistan	2017	36296113

2.2 - Analyse exploratoire du fichier disponibilité alimentaire

In [18]:
```
#Afficher les dimensions du dataset
print(f"Le tableau contient {dispo_alimentaire.shape[0]} lignes et {dispo_alimen
```

```
Le tableau contient 15605 lignes et 18 colonnes
```

In [19]:
```
#Consulter le nombre de colonnes
print(f"Le tableau contient {dispo_alimentaire.shape[1]} colonnes")
```

```
Le tableau contient 18 colonnes
```

In [20]:
```
#Affichage les 5 premières lignes de la table
dispo_alimentaire.head()
```

Out[20]:

	Zone	Produit	Origine	Aliments pour animaux	Autres Utilisations	Disponibilité alimentaire (Kcal/personne/jour)	D alir (kg/pe
0	Afghanistan	Abats Comestible	animale	NaN	NaN	5.0	
1	Afghanistan	Agrumes, Autres	vegetale	NaN	NaN	1.0	
2	Afghanistan	Aliments pour enfants	vegetale	NaN	NaN	1.0	
3	Afghanistan	Ananas	vegetale	NaN	NaN	0.0	
4	Afghanistan	Bananes	vegetale	NaN	NaN	4.0	

In [21]:
```
#remplacement des NaN dans le dataset par des 0
dispo_alimentaire = dispo_alimentaire.fillna(0)
```

In [22]:
```
#multiplication de toutes les lignes contenant des milliers de tonnes en Kg
dispo_alimentaire[['Autres Utilisations','Disponibilité intérieure', 'Exportatio
```

In [23]:
```
#Affichage les 5 premières lignes de la table
dispo_alimentaire.head()
```

Out[23]:

	Zone	Produit	Origine	Aliments pour animaux	Autres Utilisations	Disponibilité alimentaire (Kcal/personne/jour)	D alir (kg/pe
0	Afghanistan	Abats Comestible	animale	0.0	0.0	5.0	
1	Afghanistan	Agrumes, Autres	vegetale	0.0	0.0	1.0	
2	Afghanistan	Aliments pour enfants	vegetale	0.0	0.0	1.0	
3	Afghanistan	Ananas	vegetale	0.0	0.0	0.0	
4	Afghanistan	Bananes	vegetale	0.0	0.0	4.0	

2.3 - Analyse exploratoire du fichier aide alimentaire

In [25]:
```
#Afficher les dimensions du dataset
print(f"Le tableau contient {aide_alimentaire.shape[0]} lignes et {aide_alimenta
```

```
Le tableau contient 1475 lignes et 4 colonnes
```

In [26]:
```
#Consulter le nombre de colonnes
print(f"Le tableau contient {aide_alimentaire.shape[1]} colonnes")
```

```
Le tableau contient 4 colonnes
```

In [27]:
```
#Affichage les 5 premières lignes de la table
aide_alimentaire.head()
```

Out[27]:

	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

In [28]:
```
#changement du nom de la colonne Pays bénéficiaire par Zone
aide_alimentaire = aide_alimentaire.rename({'Pays bénéficiaire': 'Zone'})
```

In [29]:
```
#Multiplication de la colonne Aide_alimentaire qui contient des tonnes par 1000
aide_alimentaire['Valeur'] = aide_alimentaire['Valeur'] * 1000
```

In [30]:
```
#Affichage les 5 premières lignes de la table
aide_alimentaire.head()
```

Out[30]:

	Pays bénéficiaire	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

In [32]:
```
#Afficher les dimensions du dataset
print(f"Le tableau contient {sous_nutrition.shape[0]} lignes et {sous_nutrition.
```

```
Le tableau contient 1218 lignes et 3 colonnes
```

In [33]:
```
#Consulter le nombre de colonnes
print(f"Le tableau contient {sous_nutrition.shape[0]} colonnes")
```

```
Le tableau contient 1218 colonnes
```

In [34]:
```
#Afficher les 5 premières lignes de la table
sous_nutrition.head()
```

Out[34]:

	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

In [35]:
```
#Conversion de la colonne sous nutrition en numérique
```

In [36]:
```
#Conversion de la colonne (avec l'argument errors=coerce qui permet de convertir
#Puis remplacement des NaN en 0
sous_nutrition['Valeur'] = pd.to_numeric(sous_nutrition['Valeur'], errors='coerc
sous_nutrition = sous_nutrition.fillna(0)
```

In [37]:
```
#changement du nom de la colonne Valeur par sous_nutrition
sous_nutrition = sous_nutrition.rename(columns={'Valeur': 'sous_nutrition'})
```

In [38]:
```
#Multiplication de la colonne sous_nutrition par 1000000
sous_nutrition['sous_nutrition'] = sous_nutrition['sous_nutrition'] * 1000000
```

In [39]:
```
#Afficher les 5 premières lignes de la table
sous_nutrition.head()
```

Out[39]:

	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

In [41]:
```
pop_filter_2017 = population.loc[population['Année'] == 2017]
print(pop_filter_2017)
```

```
                                      Zone  Année  Population
4                                Afghanistan   2017  36296113.0
10                           Afrique du Sud   2017  57009756.0
16                                  Albanie   2017   2884169.0
22                                  Algérie   2017  41389189.0
28                                 Allemagne   2017  82658409.0
...                                      ...    ...         ...
1390  Venezuela (République bolivarienne du)   2017  29402484.0
1396                               Viet Nam   2017  94600648.0
1402                                  Yémen   2017  27834819.0
1408                                 Zambie   2017  16853599.0
1414                               Zimbabwe   2017  14236595.0

[236 rows x 3 columns]
```

In [42]:
```
# Il faut tout d'abord faire une jointure entre la table population et la table
sous_nutrition_filter_2017 = sous_nutrition.loc[
    (sous_nutrition['Année'] == '2015-2017') |
    (sous_nutrition['Année'] == '2017-2019') |
    (sous_nutrition['Année'] == '2016-2018')
]
sous_nutrition_filter_2017_groupby = sous_nutrition_filter_2017.groupby('Zone',
pd.set_option('display.float_format', lambda x: '%.0f' % x)
sous_nutrition_filter_2017_groupby['Année'] = 2017
sous_nutrition_filter_2017_groupby = sous_nutrition_filter_2017_groupby.fillna(0
sous_nutrition_filter_2017_groupby.tail()
```

Out[42]:

	Zone	sous_nutrition	Année
198	États-Unis d'Amérique	0	2017
199	Éthiopie	21300000	2017
200	Îles Cook	0	2017
201	Îles Marshall	0	2017
202	Îles Salomon	0	2017

In [43]:
```
pop_filter_2017_without_zone = pop_filter_2017.drop(columns=['Zone'])

population_sous_nutrition = pd.merge(pop_filter_2017, sous_nutrition_filter_2017
```

In [44]:
```
population_sous_nutrition.head()
```

Out[44]:

	Zone	Année	Population	sous_nutrition
0	Afghanistan	2017	36296113	10433333
1	Afrique du Sud	2017	57009756	3133333
2	Albanie	2017	2884169	100000
3	Algérie	2017	41389189	1266667
4	Allemagne	2017	82658409	0

In [45]:
```
#Affichage du dataset
```

```
population_sous_nutrition.tail()
```

Out[45]:

	Zone	Année	Population	sous_nutrition
198	Venezuela (République bolivarienne du)	2017	29402484	7766667
199	Viet Nam	2017	94600648	6566667
200	Yémen	2017	27834819	0
201	Zambie	2017	16853599	0
202	Zimbabwe	2017	14236595	0

In [46]:
```
#Calcul et affichage du nombre de personnes en état de sous nutrition
population_sous_nutrition['pourcentage_de_sous_nutrition'] = (
    (population_sous_nutrition['sous_nutrition'] * 100) / population_sous_nutrit
).fillna(0).round(2)
population_sous_nutrition.head()
```

Out[46]:

	Zone	Année	Population	sous_nutrition	pourcentage_de_sous_nutrition
0	Afghanistan	2017	36296113	10433333	29
1	Afrique du Sud	2017	57009756	3133333	6
2	Albanie	2017	2884169	100000	3
3	Algérie	2017	41389189	1266667	3
4	Allemagne	2017	82658409	0	0

In [ ]:
```
```

In [47]:
```
nb_humain = population_sous_nutrition['Population'].sum()
```

In [48]:
```
pourcentage_population_sous_nutrition = (
    (sum(population_sous_nutrition['sous_nutrition']) * 100) / sum(population_so
)

pourcentage_population_sous_nutrition = round(pourcentage_population_sous_nutrit

print(f"le pourcentage de sous nutrition dans le monde est de {pourcentage_popul
```

```
le pourcentage de sous nutrition dans le monde est de 7.107 % pour un total de 75
43798779.0 humains
```

3.2 - Nombre théorique de personne qui pourrait être nourries

In [50]:
```
#Combien mange en moyenne un être humain ? Source =>
#https://www.vidal.fr/sante/nutrition/equilibre-alimentaire-adulte/recommandatio
#https://nutriandco.com/fr/pages/calcul-apport-calorique-journalier#:~:text=En%2
# environs 2500 pour un homme et 1800 pour une femme
l = {'calorie_homme': 2500, 'calorie_femme': 1800}
mean_kcal_per_human = sum(l.values()) / len(l)
print("La moyenne des calories est:", mean_kcal_per_human)
```

```
La moyenne des calories est: 2150.0
```

In [51]:
```
#On commence par faire une jointure entre le data frame population et Dispo_alim
dispo_alimentaire_population  = pd.merge(dispo_alimentaire, population_sous_nutr
```

In [52]:
```
#Affichage du nouveau dataframe
dispo_alimentaire_population = dispo_alimentaire_population.drop(columns=['Année
dispo_alimentaire_population.head()
```

Out[52]:

	Zone	Produit	Origine	Aliments pour animaux	Autres Utilisations	Disponibilité alimentaire (Kcal/personne/jour)	D alir (kg/pe
0	Afghanistan	Abats Comestible	animale	0	0	5	
1	Afghanistan	Agrumes, Autres	vegetale	0	0	1	
2	Afghanistan	Aliments pour enfants	vegetale	0	0	1	
3	Afghanistan	Ananas	vegetale	0	0	0	
4	Afghanistan	Bananes	vegetale	0	0	4	

In [53]:
```
#Création de la colonne dispo_kcal avec calcul des kcal disponibles mondialement
dispo_alimentaire_population['dispo_kcal'] = dispo_alimentaire_population['Dispo

dispo_alimentaire_population.head()
```

Out[53]:

	Zone	Produit	Origine	Aliments pour animaux	Autres Utilisations	Disponibilité alimentaire (Kcal/personne/jour)	D alir (kg/pe
0	Afghanistan	Abats Comestible	animale	0	0	5	
1	Afghanistan	Agrumes, Autres	vegetale	0	0	1	
2	Afghanistan	Aliments pour enfants	vegetale	0	0	1	
3	Afghanistan	Ananas	vegetale	0	0	0	
4	Afghanistan	Bananes	vegetale	0	0	4	

5 rows × 21 columns

In [54]:
```
#Calcul du nombre d'humains pouvant être nourris
somme_dispo_alimentaire_population = dispo_alimentaire_population['dispo_kcal'].
```

```
nb_humain_pouvant_etre_nourris = somme_dispo_alimentaire_population / mean_kcal_
print(f"Nombre d'humain pouvant être nourris {nb_humain_pouvant_etre_nourris}")
```

```
Nombre d'humain pouvant être nourris 9729760291.78186
```

In [55]:
```
pourcentage_humain_pouvant_etre_nourris = (nb_humain_pouvant_etre_nourris / nb_h
print(f"Nous pourrions nourrir {round(pourcentage_humain_pouvant_etre_nourris, 3
```

```
Nous pourrions nourrir 128.977 % de la population
```

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

In [57]:
```
#Transfert des données avec les végétaux dans un nouveau dataframe
vegetal = dispo_alimentaire_population.loc[dispo_alimentaire_population['Origine
```

In [58]:
```
#Calcul du nombre de kcal disponible pour les végétaux
vegetal.head()
```

Out[58]:

	Zone	Produit	Origine	Aliments pour animaux	Autres Utilisations	Disponibilité alimentaire (Kcal/personne/jour)	Dis alime (kg/pers
1	Afghanistan	Agrumes, Autres	vegetale	0	0	1	
2	Afghanistan	Aliments pour enfants	vegetale	0	0	1	
3	Afghanistan	Ananas	vegetale	0	0	0	
4	Afghanistan	Bananes	vegetale	0	0	4	
6	Afghanistan	Bière	vegetale	0	0	0	

5 rows × 21 columns

In [59]:
```
#Calcul du nombre d'humains pouvant être nourris avec les végétaux
somme_dispo_alimentaire_population_vegetal = vegetal['dispo_kcal'].sum()
nb_humain_nourris_vegetal = somme_dispo_alimentaire_population_vegetal / mean_kc
print(nb_humain_nourris_vegetal)
```

```
8028262423.953954
```

In [60]:
```
pourcentage_humain_nourris_vegetal = (nb_humain_nourris_vegetal / nb_humain) * 1
print(f"Nous pourrions nourrir {round(pourcentage_humain_nourris_vegetal, 3)} %
```

```
Nous pourrions nourrir 106.422 % de la population
```

3.4 - Utilisation de la disponibilité intérieure

In [62]:
```
#Calcul de la disponibilité totale
dispo_alimentaire.head()
```

```
dispo_alimentaire.shape
sum_disponibilité_intérieur = dispo_alimentaire['Disponibilité intérieure'].sum(
print(sum_disponibilité_intérieur)
```

```
9848994000.0
```

In [63]:
```
dispo_alimentaire.head(10)
```

Out[63]:

	Zone	Produit	Origine	Aliments pour animaux	Autres Utilisations	Disponibilité alimentaire (Kcal/personne/jour)	D ali (kg/pe
0	Afghanistan	Abats Comestible	animale	0	0	5	
1	Afghanistan	Agrumes, Autres	vegetale	0	0	1	
2	Afghanistan	Aliments pour enfants	vegetale	0	0	1	
3	Afghanistan	Ananas	vegetale	0	0	0	
4	Afghanistan	Bananes	vegetale	0	0	4	
5	Afghanistan	Beurre, Ghee	animale	0	0	23	
6	Afghanistan	Bière	vegetale	0	0	0	
7	Afghanistan	Blé	vegetale	0	0	1369	
8	Afghanistan	Boissons Alcooliques	vegetale	0	0	0	
9	Afghanistan	Café	vegetale	0	0	0	

In [64]:
```
#création d'une boucle for pour afficher les différentes valeurs en fonction des
l = ['Aliments pour animaux', 'Nourriture', 'Pertes', 'Autres Utilisations', 'Se
d = {}

for i in l:
    d[i] = dispo_alimentaire[i].sum()
for o in d:
    d[o] = round((d[o] / sum_disponibilité_intérieur) * 100, 3)

print(f"Affichage du pourcentage de chaque catégories par rapport à la disponibi
```

```
Affichage du pourcentage de chaque catégories par rapport à la disponibilité tota
le:
{'Aliments pour animaux': 13.242, 'Nourriture': 49.51, 'Pertes': 4.607, 'Autres U
tilisations': 8.783, 'Semences': 1.571, 'Traitement': 22.385}
```

3.5 - Utilisation des céréales

In [66]:
```
#Création d'une liste avec toutes les variables
#source : https://www.fao.org/faostat/fr/#data/FBSH
céréales = ['Blé', 'Céréales, Autres', 'Feve de Cacao', 'Graines de coton', 'Gra
            'Maïs', 'Sésame', 'Millet', 'Avoine', 'Orge', 'Riz', 'Seigle', 'Sorg
```

In [67]:
```
#Création d'un dataframe avec les informations uniquement pour ces céréales
d = {}
for i in range(len(céréales)):
    d[f'var{i}'] = dispo_alimentaire.loc[dispo_alimentaire['Produit'] == céréale
    info_céréales = pd.concat(d.values()).reset_index()
info_céréales.tail()
```

Out[67]:

	index	Zone	Produit	Origine	Aliments pour animaux	Autres Utilisations	Disponibilité alimentaire (Kcal/personne/jour)
1940	15232	Émirats arabes unis	Sorgho	vegetale	59000	0	0
1941	15325	Équateur	Sorgho	vegetale	14000	0	0
1942	15421	États-Unis d'Amérique	Sorgho	vegetale	2351000	0	7
1943	15513	Éthiopie	Sorgho	vegetale	0	1500000	216
1944	15593	Îles Salomon	Sorgho	vegetale	0	0	0

In [68]:
```
aliment_dispo_intérieure = info_céréales['Disponibilité intérieure'].sum()
```

In [69]:
```
#Affichage de la proportion d'alimentation animale
aliment_pour_animaux = info_céréales['Aliments pour animaux'].sum()
pourcentage_alimentation_animale = round((aliment_pour_animaux / aliment_dispo_i
print(f"L'alimentation animale représente {pourcentage_alimentation_animale} % d
```

```
L'alimentation animale représente 42.123 % de l'utilisation des céréales
```

In [70]:
```
#Affichage de la proportion d'alimentation humaine
nourriture = info_céréales['Nourriture'].sum()
pourcentage_alimentation_humaine = round((nourriture / aliment_dispo_intérieure)
print(f"L'alimentation humaine représente {pourcentage_alimentation_humaine} % d
```

```
L'alimentation humaine représente 32.48 % de l'utilisation des céréales
```

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

In [72]:
```
#Création de la colonne proportion par pays
population_sous_nutrition.head()
```

Out[72]:

	Zone	Année	Population	sous_nutrition	pourcentage_de_sous_nutrition
0	Afghanistan	2017	36296113	10433333	29
1	Afrique du Sud	2017	57009756	3133333	6
2	Albanie	2017	2884169	100000	3
3	Algérie	2017	41389189	1266667	3
4	Allemagne	2017	82658409	0	0

In [73]:
```
#affichage après trie des 10 pires pays
sorted_population_sous_nutrition = population_sous_nutrition.sort_values(['pourc
sorted_population_sous_nutrition.to_csv('10_pays_sous_nutrition.csv', index=Fals
sorted_population_sous_nutrition.head(10)
```

Out[73]:

	Zone	Année	Population	sous_nutrition	pourcentage_de_sous_nutrition
78	Haïti	2017	10982366	5300000	48
157	République populaire démocratique de Corée	2017	25429825	11933333	47
108	Madagascar	2017	25570512	10600000	41
103	Libéria	2017	4702226	1800000	38
183	Tchad	2017	15016753	5700000	38
100	Lesotho	2017	2091534	766667	37
161	Rwanda	2017	11980961	4233333	35
121	Mozambique	2017	28649018	9400000	33
186	Timor-Leste	2017	1243258	400000	32
0	Afghanistan	2017	36296113	10433333	29

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

In [75]:
```
#calcul du total de l'aide alimentaire par pays


aide_alimentaire_groupby = aide_alimentaire.groupby('Pays bénéficiaire', as_inde
aide_alimentaire_groupby.head()
```

Out[75]:

	Pays bénéficiaire	Valeur
0	Afghanistan	185452000
1	Algérie	81114000
2	Angola	5014000
3	Bangladesh	348188000
4	Bhoutan	2666000

In [76]:
```
#affichage après trie des 10 pays qui ont bénéficié le plus de l'aide alimentair
aide_alimentaire_groupby_sorted = aide_alimentaire_groupby.sort_values(['Valeur'
aide_alimentaire_groupby_sorted.head(10)
```

Out[76]:

	Pays bénéficiaire	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

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

In [78]:
```
aide_alimentaire_rework = aide_alimentaire.rename(columns={'Pays bénéficiaire':
aide_alimentaire_2013_2016 = aide_alimentaire_rework.loc[(aide_alimentaire_rewor

aide_alimentaire_2013_2016.head()
```

Out[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

In [79]:
```
aide_alimentaire_rework_groupby_annee_zone = aide_alimentaire_2013_2016.groupby(
aide_alimentaire_rework_groupby_annee_zone.head()
aide_alimentaire_rework_groupby_annee_zone.to_csv('aide_alimentaire_rework_group
```

In [80]:
```
aide_alimentaire_rework_groupby_zone = aide_alimentaire_2013_2016.groupby(['Zone
aide_alimentaire_rework_groupby_zone = aide_alimentaire_rework_groupby_zone.drop
aide_alimentaire_rework_groupby_zone.head()
```

Out[80]:

	Zone	Valeur
0	Afghanistan	185452000
1	Algérie	81114000
2	Angola	5014000
3	Bangladesh	348188000
4	Bhoutan	2666000

In [81]:
```
#Création d'une liste contenant les 5 pays qui ont le plus bénéficiées de l'aide
top_5_most_beneficial_country_aide_alimentaire = aide_alimentaire_rework_groupby
top_5_most_beneficial_country_aide_alimentaire.head()
liste_top_5_most_beneficial_country_aide_alimentaire = []

for o in range(5):
    liste_top_5_most_beneficial_country_aide_alimentaire.append(top_5_most_benef
print(liste_top_5_most_beneficial_country_aide_alimentaire)
```

```
['République arabe syrienne', 'Éthiopie', 'Yémen', 'Soudan du Sud', 'Soudan']
```

In [82]:
```
sous_nutrition_2013_2016 = sous_nutrition.copy()
sous_nutrition_2013_2016['Année'] = sous_nutrition_2013_2016['Année'].replace({'
sous_nutrition_2013_2016_pop = pd.merge(sous_nutrition_2013_2016, population, on
sous_nutrition_2013_2016_pop['Pourcentage'] = (sous_nutrition_2013_2016_pop['sou
sous_nutrition_2013_2016_pop_top_5 = sous_nutrition_2013_2016_pop.loc[sous_nutri
sous_nutrition_2013_2016_pop_top_5.head()
```

Out[82]:

	Zone	Année	sous_nutrition	Population	Pourcentage
366	Éthiopie	2013	26200000	95385798	27
367	Éthiopie	2014	24300000	98094265	25
368	Éthiopie	2015	21700000	100835458	22
369	Éthiopie	2016	21300000	103603462	21
370	Éthiopie	2017	21100000	106399924	20

In [83]:
```
population.head()
```

Out[83]:

	Zone	Année	Population
0	Afghanistan	2013	32269589
1	Afghanistan	2014	33370794
2	Afghanistan	2015	34413603
3	Afghanistan	2016	35383032
4	Afghanistan	2017	36296113

In [84]:
```
#On filtre sur le dataframe avec notre liste
aide_alimentaire_top_5 = aide_alimentaire_2013_2016.loc[aide_alimentaire_2013_20
aide_alimentaire_top_5_groupby_annee = aide_alimentaire_top_5.groupby(['Année',
aide_alimentaire_top_5_groupby_annee = aide_alimentaire_top_5_groupby_annee.drop
aide_alimentaire_top_5_groupby_annee.head(20)
aide_alimentaire_top_5_groupby_annee.to_csv('aide_alimentaire_top_5_groupby_anne
```

In [85]:
```
# Affichage des pays avec l'aide alimentaire par année
aide_alimentaire_top_5_groupby = aide_alimentaire_top_5.groupby('Zone')['Valeur'
aide_alimentaire_top_5_groupby_sorted = aide_alimentaire_top_5_groupby.sort_valu

aide_alimentaire_top_5_groupby_sorted.head()
```

Out[85]:

	Zone	Valeur
0	République arabe syrienne	1858943000
4	Éthiopie	1381294000
3	Yémen	1206484000
2	Soudan du Sud	695248000
1	Soudan	669784000

3.9 - Pays avec le moins de disponibilité par habitant

In [87]:
```
population_filter_2018 = population.loc[population['Année'] == 2018]
population_filter_2018 = population_filter_2018.fillna(0)
dispo_alimentaire_population_2018 = pd.merge(dispo_alimentaire, population_filte
dispo_alimentaire_population_2018.shape
```

Out[87]: (15416, 20)

In [88]:
```
dispo_alimentaire_population_2018['dispo_kcal'] = dispo_alimentaire_population_2
```

In [89]:
```
#Calcul de la disponibilité en kcal par personne par jour par pays

dispo_alimentaire_population_groupby_dispo = dispo_alimentaire_population_2018.g
dispo_alimentaire_population_par_personne = pd.merge(dispo_alimentaire_populatio
dispo_alimentaire_population_par_personne.head()
```

Out[89]:

	Zone	dispo_kcal	Année	Population
0	Afghanistan	77577799127	2018	37171921
1	Afrique du Sud	174533404360	2018	57792518
2	Albanie	9190175120	2018	2882740
3	Algérie	139058147544	2018	42228408
4	Allemagne	291184836254	2018	83124418

In [90]:
```
calcul_dispo = dispo_alimentaire_population_par_personne['dispo_kcal'] / dispo_a
dispo_alimentaire_population_par_personne['dispo_kcal_personne_jour'] = calcul_d
dispo_alimentaire_population_par_personne.loc[dispo_alimentaire_population_par_p
```

Out[90]:

	Zone	dispo_kcal	Année	Population	dispo_kcal_personne_jour
151	Thaïlande	193358241605	2018	69428453	2785

In [91]:
```
#Affichage des 10 pays qui ont le moins de dispo alimentaire par personne
dispo_alimentaire_population_par_personne.sort_values(by='dispo_kcal_personne_jo
```

Out[91]:

	index	Zone	dispo_kcal	Année	Population	dispo_kcal_personne_jour
0	127	République centrafricaine	8768105472	2018	4666368	1879
1	164	Zambie	33384686192	2018	17351708	1924
2	91	Madagascar	53995315528	2018	26262313	2056
3	0	Afghanistan	77577799127	2018	37171921	2087
4	65	Haïti	23236318842	2018	11123178	2089
5	132	République populaire démocratique de Corée	53475321172	2018	25549604	2093
6	150	Tchad	32642530461	2018	15477729	2109
7	165	Zimbabwe	30509188626	2018	14438802	2113
8	114	Ouganda	90841930536	2018	42729036	2126
9	152	Timor-Leste	2699516646	2018	1267974	2129

3.10 - Pays avec le plus de disponibilité par habitant

In [93]:
```
#Affichage des 10 pays qui ont le plus de dispo alimentaire par personne

dispo_alimentaire_population_par_personne = dispo_alimentaire_population_par_per
dispo_alimentaire_population_par_personne.to_csv('top_10_dispo_alimentaire_popul
dispo_alimentaire_population_par_personne.head()
```

Out[93]:

	index	Zone	dispo_kcal	Année	Population	dispo_kcal_personne_jour
0	11	Autriche	33520532760	2018	8891388	3770
1	16	Belgique	42908899186	2018	11482178	3737
2	157	Turquie	305317046304	2018	82340088	3708
3	169	États-Unis d'Amérique	1204368447730	2018	327096265	3682
4	74	Israël	30257272760	2018	8381516	3610

3.11 - Exemple de la Thaïlande pour le Manioc

In [95]:
```
#création d'un dataframe avec uniquement la Thaïlande
dispo_alimentaire_thailande_manioc = dispo_alimentaire.loc[(dispo_alimentaire['Z
                                     (dispo_alimentaire['Produit'] == 'Manioc')]

dispo_alimentaire_thailande_manioc.head()
```

Out[95]:

	Zone	Produit	Origine	Aliments pour animaux	Autres Utilisations	Disponibilité alimentaire (Kcal/personne/jour)	Di alin (kg/pe
13809	Thaïlande	Manioc	vegetale	1800000	2081000	40	

In [96]:
```
#Calcul de la sous nutrition en Thaïlande
population_thailande = population.loc[population['Zone'] == 'Thaïlande']
sous_nutrition_thailande = sous_nutrition.loc[sous_nutrition['Zone'] == 'Thaïlan
sous_nutrition_thailande.loc[:, 'Année'] = sous_nutrition_thailande['Année'].rep
    '2012-2014': '2013',
    '2013-2015': '2014',
    '2014-2016': '2015',
    '2015-2017': '2016',
    '2016-2018': '2017',
    '2017-2019': '2018'
})

sous_nutrition_thailande.loc[:, 'Année'] = pd.to_numeric(sous_nutrition_thailand
sous_nutrition_thailande.head(6)
```

Out[96]:

	Zone	Année	sous_nutrition
1110	Thaïlande	2013	6200000
1111	Thaïlande	2014	6000000
1112	Thaïlande	2015	5900000
1113	Thaïlande	2016	6000000
1114	Thaïlande	2017	6200000
1115	Thaïlande	2018	6500000

In [97]:
```
sous_nutrition_pop_thailande = pd.merge(population_thailande, sous_nutrition_tha
pourcentage_sous_nutrition_thailande = (((sous_nutrition_pop_thailande['sous_nut
sous_nutrition_pop_thailande['pourcentage_sous_nutrition'] = pourcentage_sous_nu
sous_nutrition_pop_thailande.head()
```

Out[97]:

	Zone	Année	Population	sous_nutrition	pourcentage_sous_nutrition
0	Thaïlande	2013	68144518	6200000	9
1	Thaïlande	2014	68438746	6000000	9
2	Thaïlande	2015	68714511	5900000	9
3	Thaïlande	2016	68971308	6000000	9
4	Thaïlande	2017	69209810	6200000	9

In [98]:
```
moyenne_pourcentage_sous_nutrition_thailande = (((sum(sous_nutrition_pop_thailan
population_thailande_2018 = sous_nutrition_pop_thailande.loc[sous_nutrition_pop_
print(f'La moyenne de sous_nutrition sur les 5 dernières années est de {round(mo
```

```
La moyenne de sous_nutrition sur les 5 dernières années est de 8.912 % pour 69428
453.0 humain
```

In [99]:
```
# On calcule la proportion exportée en fonction de la proportion
proportion_exporter_production = ((dispo_alimentaire_thailande_manioc['Exportati

print(f"L'exportation de manioc représente {round(proportion_exporter_production
```

```
L'exportation de manioc représente 83.413 % de la production
```

Etape 6 - Analyse complémentaires

In [101…
```
#Rajouter en dessous toutes les analyses complémtaires suite à la demande de mél
#"et toutes les infos que tu trouverais utiles pour mettre en relief les pays qu
#le plus en difficulté au niveau alimentaire"
```

In [102…
```
dispo_alimentaire_produit = dispo_alimentaire.iloc[:, 1].reset_index()
dispo_alimentaire_produit_unique = dispo_alimentaire_produit.drop_duplicates(sub
```

In [103…
```
dispo_alimentaire.head()
```

Out[103...

	Zone	Produit	Origine	Aliments pour animaux	Autres Utilisations	Disponibilité alimentaire (Kcal/personne/jour)	D alir (kg/pe
0	Afghanistan	Abats Comestible	animale	0	0	5	
1	Afghanistan	Agrumes, Autres	vegetale	0	0	1	
2	Afghanistan	Aliments pour enfants	vegetale	0	0	1	
3	Afghanistan	Ananas	vegetale	0	0	0	
4	Afghanistan	Bananes	vegetale	0	0	4	

In [104...

```
dispo_alimentaire_kcal = dispo_alimentaire[['Zone', 'Produit', 'Disponibilité al
dispo_alimentaire_kcal.head()
```

Out[104...

	Zone	Produit	Disponibilité alimentaire (Kcal/personne/jour)	Nourriture	Pertes
0	Afghanistan	Abats Comestible	5	53000	0
1	Afghanistan	Agrumes, Autres	1	39000	2000
2	Afghanistan	Aliments pour enfants	1	2000	0
3	Afghanistan	Ananas	0	0	0
4	Afghanistan	Bananes	4	82000	0

In [105...

```
population_dispo_alimentaire = pd.merge(dispo_alimentaire_kcal, pop_filter_2017,
population_dispo_alimentaire = population_dispo_alimentaire.drop(columns='Année'
population_dispo_alimentaire.head(10)
```

Out[105…

	Zone	Produit	Disponibilité alimentaire (Kcal/personne/jour)	Nourriture	Pertes	Population
0	Afghanistan	Abats Comestible	5	53000	0	36296113
1	Afghanistan	Agrumes, Autres	1	39000	2000	36296113
2	Afghanistan	Aliments pour enfants	1	2000	0	36296113
3	Afghanistan	Ananas	0	0	0	36296113
4	Afghanistan	Bananes	4	82000	0	36296113
5	Afghanistan	Beurre, Ghee	23	36000	0	36296113
6	Afghanistan	Bière	0	3000	0	36296113
7	Afghanistan	Blé	1369	4895000	775000	36296113
8	Afghanistan	Boissons Alcooliques	0	0	0	36296113
9	Afghanistan	Café	0	0	0	36296113

In [106…

```
population_dispo_alimentaire['Valeur_kcal/kg'] = (365 * population_dispo_aliment
population_dispo_alimentaire = population_dispo_alimentaire.fillna(0)
population_dispo_alimentaire.head(10)
```

Out[106…

	Zone	Produit	Disponibilité alimentaire (Kcal/personne/jour)	Nourriture	Pertes	Population	Valeu
0	Afghanistan	Abats Comestible	5	53000	0	36296113	
1	Afghanistan	Agrumes, Autres	1	39000	2000	36296113	
2	Afghanistan	Aliments pour enfants	1	2000	0	36296113	
3	Afghanistan	Ananas	0	0	0	36296113	
4	Afghanistan	Bananes	4	82000	0	36296113	
5	Afghanistan	Beurre, Ghee	23	36000	0	36296113	
6	Afghanistan	Bière	0	3000	0	36296113	
7	Afghanistan	Blé	1369	4895000	775000	36296113	
8	Afghanistan	Boissons Alcooliques	0	0	0	36296113	
9	Afghanistan	Café	0	0	0	36296113	

In [107…

```
population_dispo_alimentaire['Kcal_pertes'] = population_dispo_alimentaire['Pert
population_dispo_alimentaire.head()
```

Out[107…

	Zone	Produit	Disponibilité alimentaire (Kcal/personne/jour)	Nourriture	Pertes	Population	Valeur_
0	Afghanistan	Abats Comestible	5	53000	0	36296113	
1	Afghanistan	Agrumes, Autres	1	39000	2000	36296113	
2	Afghanistan	Aliments pour enfants	1	2000	0	36296113	
3	Afghanistan	Ananas	0	0	0	36296113	
4	Afghanistan	Bananes	4	82000	0	36296113	

In [108…

```
population_dispo_alimentaire_sous_nutrition = pd.merge(population_sous_nutrition
population_dispo_alimentaire_sous_nutrition = population_dispo_alimentaire_sous_
population_dispo_alimentaire_sous_nutrition.head()
```

Out[108…

	Zone	Population	sous_nutrition	pourcentage_de_sous_nutrition	Produit	(I
0	Afghanistan	36296113	10433333	29	Abats Comestible	
1	Afghanistan	36296113	10433333	29	Agrumes, Autres	
2	Afghanistan	36296113	10433333	29	Aliments pour enfants	
3	Afghanistan	36296113	10433333	29	Ananas	
4	Afghanistan	36296113	10433333	29	Bananes	

In [109…

```
population_dispo_alimentaire_sous_nutrition_groupby_zone = population_dispo_alim
population_dispo_alimentaire_sous_nutrition_groupby_zone = population_dispo_alim
population_dispo_alimentaire_sous_nutrition_groupby_zone.head()
```

Out[109…

	Zone	Kcal_pertes/an
0	Afghanistan	3439850926
1	Afrique du Sud	2144373735
2	Albanie	NaN
3	Algérie	4819549451
4	Allemagne	5255788411

In [110…

```
pertes = dispo_alimentaire[['Zone', 'Pertes']].groupby('Zone').sum()
pertes_sorted = pertes.sort_values('Pertes', ascending=False)
pertes_sorted.head(10)
```

Out[110…

Zone	Pertes
Chine, continentale	89575000
Brésil	75914000
Inde	55930000
Nigéria	19854000
Indonésie	13081000
Turquie	12036000
Mexique	8289000
Égypte	7608000
Ghana	7442000
États-Unis d'Amérique	7162000

In [111…

```
sous_nutrition_pertes = pd.merge(population_sous_nutrition, pertes, on='Zone')
sous_nutrition_pertes_sorted = sous_nutrition_pertes.sort_values('Pertes', ascen
sous_nutrition_pertes_sorted.head(10)
```

Out[111…

	Zone	Année	Population	sous_nutrition	pourcentage_de_sous_nutrition	
35	Chine, continentale	2017	1421021791	0	0	89
24	Brésil	2017	207833823	0	0	75
74	Inde	2017	1338676785	190066667	14	55
115	Nigéria	2017	190873244	23200000	12	19
75	Indonésie	2017	264650963	23900000	9	13
163	Turquie	2017	81116450	0	0	12
106	Mexique	2017	124777324	8433333	7	8
47	Égypte	2017	96442591	4566667	5	7
63	Ghana	2017	29121465	2000000	7	7
54	États-Unis d'Amérique	2017	325084756	0	0	7

In [112…

```
sous_nutrition_pertes['Kcal_pertes'] = population_dispo_alimentaire_sous_nutriti
sous_nutrition_pertes['kcal_besoin/an'] = sous_nutrition_pertes['Population'] *
sous_nutrition_pertes['pourcentage_de_sous_nutrition_perte'] = (sous_nutrition_p
sous_nutrition_pertes.head()
```

Out[112...

	Zone	Année	Population	sous_nutrition	pourcentage_de_sous_nutrition	Pert
0	Afghanistan	2017	36296113	10433333	29	11350
1	Afrique du Sud	2017	57009756	3133333	6	21930
2	Albanie	2017	2884169	100000	3	2760
3	Algérie	2017	41389189	1266667	3	37530
4	Allemagne	2017	82658409	0	0	37810

In [245...

```
sous_nutrition_pertes.sort_values('pourcentage_de_sous_nutrition_perte', ascendi
```

Out[245...

	Zone	Année	Population	sous_nutrition	pourcentage_de_sous_nutrition	F
85	Kazakhstan	2017	18080019	0	0	16
73	Îles Salomon	2017	636039	0	0	
82	Jamaïque	2017	2920848	300000	10	
89	Koweït	2017	4056099	0	0	1
20	Bermudes	2017	63049	0	0	
...	...	...	...	...	...	
23	Botswana	2017	2205080	466667	21	
69	Guyana	2017	775222	0	0	1
68	Guinée-Bissau	2017	1828145	0	0	
108	Monténégro	2017	627563	0	0	
166	Vanuatu	2017	285510	0	0	

172 rows × 9 columns

In [249...

```
sous_nutrition_cause_perte = pd.merge(population_dispo_alimentaire_sous_nutritio
sous_nutrition_cause_perte = sous_nutrition_cause_perte.fillna(0)
sous_nutrition_cause_perte.head()
```

Out[249...

	Zone	Kcal_pertes/an	Année	Population	sous_nutrition	pourcentage_de_sous_
0	Afghanistan	3439850926	2017	36296113	10433333	
1	Afrique du Sud	2144373735	2017	57009756	3133333	
2	Albanie	0	2017	2884169	100000	
3	Algérie	4819549451	2017	41389189	1266667	
4	Allemagne	5255788411	2017	82658409	0	

In [251...

```
sous_nutrition_cause_perte['pourcentage_de_sous_nutrition_perte'] = (sous_nutrit
sous_nutrition_cause_perte = sous_nutrition_cause_perte.drop(columns='Kcal_perte
sous_nutrition_cause_perte.sort_values('pourcentage_de_sous_nutrition_perte', as
```

Out[251...

	Zone	Kcal_pertes/an	Année	Population	sous_nutrition	pourcentage_de_sou
140	Sierra Leone	2906650755	2017	7488423	2000000	
24	Bulgarie	2562783247	2017	7102444	200000	
57	Ghana	9649737397	2017	29121465	2000000	
29	Cambodge	4853720282	2017	16009409	2400000	
93	Malawi	4927552897	2017	17670196	3200000	
...	...	...	...	...	...	
6	Antigua-et-Barbuda	0	2017	95426	0	
73	Islande	0	2017	334393	0	
134	Saint-Kitts-et-Nevis	0	2017	52045	0	
2	Albanie	0	2017	2884169	100000	
82	Kiribati	0	2017	114158	0	

172 rows × 9 columns

In [253...

```
sous_nutrition_cause_perte.head()
```

Out[253…

	Zone	Kcal_pertes/an	Année	Population	sous_nutrition	pourcentage_de_sous_
0	Afghanistan	3439850926	2017	36296113	10433333	
1	Afrique du Sud	2144373735	2017	57009756	3133333	
2	Albanie	0	2017	2884169	100000	
3	Algérie	4819549451	2017	41389189	1266667	
4	Allemagne	5255788411	2017	82658409	0	

In [255…
```
sous_nutrition_cause_perte['nb_humain_nourrissable'] = sous_nutrition_cause_pert
```

In [257…
```
sous_nutrition_cause_perte.head()
```

Out[257…

	Zone	Kcal_pertes/an	Année	Population	sous_nutrition	pourcentage_de_sous_
0	Afghanistan	3439850926	2017	36296113	10433333	
1	Afrique du Sud	2144373735	2017	57009756	3133333	
2	Albanie	0	2017	2884169	100000	
3	Algérie	4819549451	2017	41389189	1266667	
4	Allemagne	5255788411	2017	82658409	0	

In [243…
```
sous_nutrition_cause_perte.to_csv('sous_nutrition_cause_perte.csv', sep=',')
```

In [ ]: