Fernandes_David_1_notebook_112024

November 11, 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'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

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

1.2 - Chargement des fichiers Excel

[8]:
```
#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

[11]:
```
#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)
```

[12]:
```
#Consulter le nombre de colonnes
print("Le tableau comporte {} colonne(s)".format(population.shape[1]))

#La nature des données dans chacune des colonnes
print("\nNature des données dans chaque colonne :")
print(population.dtypes)

#Le nombre de valeurs présentes dans chacune des colonnes
print("\nNombre de valeurs présentes dans chaque colonne :")
population.count()
```

```
Le tableau comporte 3 colonne(s)

Nature des données dans chaque colonne :
Zone       object
Année       int64
Valeur    float64
dtype: object

Nombre de valeurs présentes dans chaque colonne :
```

[12]:
```
Zone      1416
Année     1416
Valeur    1416
dtype: int64
```

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

[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

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

#Multiplication de la colonne valeur par 1000
population.head()
```

[14]:

	Zone	Année	Valeur
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

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

[16]:
```
#Affichage les 5 premières lignes de la table pour voir les modifications
population.head()
```

[16]:

	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

[18]:
```
#Afficher les dimensions du dataset
print("Dimensions du dataset \'Dispo alimentaire\' :", dispo_alimentaire.shape)
```

```
Dimensions du dataset 'Dispo alimentaire' : (15605, 18)
```

[19]:
```
#Consulter le nombre de colonnes
print("Nombre de colonnes \'Dispo alimentaire\' :", dispo_alimentaire.shape[1])
```

```
Nombre de colonnes 'Dispo alimentaire' : 18
```

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

[20]:

	Zone	Produit	Origine	Aliments pour animaux	Autres Utilisations	Disponibilité alimentaire (Kcal/personne/jour)	Disponibilité alimentaire en quantité (kg/personne/an)	Disponibilité de matière grasse en quantité (g/personne/jour)	Disponibilité de protéines en quantité (g/personne/jour)	Disponibilité intérieure	Exportations - Quantité	Importations - Quantité	Nourriture	Pertes	Production	Semences	Traitement	Variation de stock
0	Afghanistan	Abats Comestible	animale	NaN	NaN	5.0	1.72	0.20	0.77	53.0	NaN	NaN	53.0	NaN	53.0	NaN	NaN	NaN
1	Afghanistan	Agrumes, Autres	vegetale	NaN	NaN	1.0	1.29	0.01	0.02	41.0	2.0	40.0	39.0	2.0	3.0	NaN	NaN	NaN
2	Afghanistan	Aliments pour enfants	vegetale	NaN	NaN	1.0	0.06	0.01	0.03	2.0	NaN	2.0	2.0	NaN	NaN	NaN	NaN	NaN
3	Afghanistan	Ananas	vegetale	NaN	NaN	0.0	0.00	NaN	NaN	0.0	NaN	0.0	0.0	NaN	NaN	NaN	NaN	NaN
4	Afghanistan	Bananes	vegetale	NaN	NaN	4.0	2.70	0.02	0.05	82.0	NaN	82.0						

```
4        82.0      NaN          NaN        NaN          NaN                   NaN
```

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

[22]:
```
#multiplication de toutes les lignes contenant des milliers de tonnes en Kg
dispo_alimentaire['Disponibilité intérieure'] =␣
 ↪dispo_alimentaire['Disponibilité intérieure'] * 1000000
dispo_alimentaire['Exportations - Quantité'] = dispo_alimentaire['Exportations␣
 ↪- Quantité'] * 1000000
dispo_alimentaire['Importations - Quantité'] = dispo_alimentaire['Importations␣
 ↪- Quantité'] * 1000000
dispo_alimentaire['Nourriture'] = dispo_alimentaire['Nourriture'] * 1000000
dispo_alimentaire['Pertes'] = dispo_alimentaire['Pertes'] * 1000000
dispo_alimentaire['Production'] = dispo_alimentaire['Production'] * 1000000
dispo_alimentaire['Semences'] = dispo_alimentaire['Semences'] * 1000000
dispo_alimentaire['Variation de stock'] = dispo_alimentaire['Variation de␣
 ↪stock'] * 1000000
dispo_alimentaire['Traitement'] = dispo_alimentaire['Traitement'] * 1000000
dispo_alimentaire['Aliments pour animaux'] = dispo_alimentaire['Aliments pour␣
 ↪animaux'] * 1000000
dispo_alimentaire['Autres Utilisations'] = dispo_alimentaire['Autres␣
 ↪Utilisations'] * 1000000
```

[23]:
```
liste = ['Disponibilité intérieure', 'Exportations - Quantité', 'Importations -␣
 ↪Quantité', 'Nourriture', 'Pertes', 'Production', 'Semences', 'Variation de␣
 ↪stock', 'Traitement',  'Aliments pour animaux', 'Autres Utilisations']
for element in liste:
    dispo_alimentaire[element] *= 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              5.300000e+13              0.000000e+00              0.000000e+00
1              4.100000e+13              2.000000e+12              4.000000e+13
2              2.000000e+12              0.000000e+00              2.000000e+12
3              0.000000e+00              0.000000e+00              0.000000e+00
4              8.200000e+13              0.000000e+00              8.200000e+13

     Nourriture        Pertes    Production  Semences  Traitement  \
0  5.300000e+13  0.000000e+00  5.300000e+13       0.0         0.0
1  3.900000e+13  2.000000e+12  3.000000e+12       0.0         0.0
2  2.000000e+12  0.000000e+00  0.000000e+00       0.0         0.0
3  0.000000e+00  0.000000e+00  0.000000e+00       0.0         0.0
4  8.200000e+13  0.000000e+00  0.000000e+00       0.0         0.0

   Variation de stock
0                 0.0
1                 0.0
2                 0.0
3                 0.0
4                 0.0
```

2.3 - Analyse exploratoire du fichier aide alimentaire

[26]:
```
#Afficher les dimensions du dataset
print("Dimensions du dataset \'Aide alimentaire\' :", aide_alimentaire.shape)
```

```
Dimensions du dataset 'Aide alimentaire' : (1475, 4)
```

[27]:
```
#Consulter le nombre de colonnes
print("Nombre de colonnes \'Aide alimentaire\' :", aide_alimentaire.shape[1])
```

```
Nombre de colonnes 'Aide alimentaire' : 4
```

[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.rename(columns={'Pays bénéficiaire': 'Zone'}, inplace=True)
```

[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("Dimensions du dataset \'Sous nutrition\' :", sous_nutrition.shape)
```

```
Dimensions du dataset 'Sous nutrition' : (1218, 3)
```

[34]:
```
#Consulter le nombre de colonnes
print("Nombre de colonnes \'Sous nutrition\' :", sous_nutrition.shape[1])
```

```
Nombre de colonnes 'Sous nutrition' : 3
```

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

[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
sous_nutrition['Valeur'] = pd.to_numeric(sous_nutrition['Valeur'],␣
 ↪errors='coerce')
sous_nutrition.head()
```

[36]:
```
          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
```

[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)


#Puis remplacement des NaN en 0
sous_nutrition.fillna(0, inplace=True)
sous_nutrition
```

[37]:
```
             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
…             …         …     …
1213     Zimbabwe  2013-2015     0.0
1214     Zimbabwe  2014-2016     0.0
1215     Zimbabwe  2015-2017     0.0
1216     Zimbabwe  2016-2018     0.0
1217     Zimbabwe  2017-2019     0.0

[1218 rows x 3 columns]
```

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

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

[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
…              …          …              …
1213      Zimbabwe  2013-2015             0.0
1214      Zimbabwe  2014-2016             0.0
1215      Zimbabwe  2015-2017             0.0
1216      Zimbabwe  2016-2018             0.0
1217      Zimbabwe  2017-2019             0.0

[1218 rows x 3 columns]
```

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

[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]:
```
population_tmp = population.copy()
sous_nutrition_tmp = sous_nutrition.copy()

population_2017 = population_tmp[population_tmp['Année'] == 2017]
sous_nutrition_2017 = sous_nutrition_tmp[sous_nutrition_tmp['Année'] ==␣
 ↪'2016-2018']
sous_nutrition_2017.loc[:, 'Année'] = 2017

merged_df_2017 = pd.merge(population_2017, sous_nutrition_2017, on=['Zone',␣
 ↪'Année'])
merged_df_2017
```

[42]:
```
                                    Zone Année  Population  sous_nutrition
0                              Afghanistan  2017  36296113.0      10500000.0
1                         Afrique du Sud  2017  57009756.0       3100000.0
2                                Albanie  2017   2884169.0        100000.0
3                                Algérie  2017  41389189.0       1300000.0
4                              Allemagne  2017  82658409.0             0.0
..                                   …   …         …              …
198  Venezuela (République bolivarienne du)  2017  29402484.0       8000000.0
199                            Viet Nam  2017  94600648.0       6500000.0
```

200	Yémen	2017	27834819.0	0.0
201	Zambie	2017	16853599.0	0.0
202	Zimbabwe	2017	14236595.0	0.0

[203 rows x 4 columns]

[43]:
```
#Affichage du dataset
print("Dataset joint pour l'année 2017 :")
merged_df_2017.head(10)
```

Dataset joint pour l'année 2017 :

[43]:

	Zone	Année	Population	sous_nutrition
0	Afghanistan	2017	36296113.0	10500000.0
1	Afrique du Sud	2017	57009756.0	3100000.0
2	Albanie	2017	2884169.0	100000.0
3	Algérie	2017	41389189.0	1300000.0
4	Allemagne	2017	82658409.0	0.0
5	Andorre	2017	77001.0	0.0
6	Angola	2017	29816766.0	5800000.0
7	Antigua-et-Barbuda	2017	95426.0	0.0
8	Arabie saoudite	2017	33101179.0	1600000.0
9	Argentine	2017	43937140.0	1500000.0

[44]:
```
#Calcul et affichage du nombre de personnes en état de sous nutrition mondiale␣
↪+ calcul de % de la population en sous nutrition par rapport à mondiale␣
↪(enrion 7%)
somme_sous_nutri = merged_df_2017['sous_nutrition'].sum()
print("Somme personne en sous nutrition dans le monde :")
somme_sous_nutri
```

Somme personne en sous nutrition dans le monde :

[44]: 535700000.0

[45]:
```
#Calcul du nombre d'humains pouvant être nourris
total_pop = merged_df_2017['Population'].sum()
Nourris_monde = somme_sous_nutri * 100 / total_pop
Nourris_monde
```

[45]: 7.1011968332354165

3.2 - Nombre théorique de personne qui pourrait être nourries

[47]:
```
#Combien mange en moyenne un être humain ? Source =>

# 2000 kcal : Source -> https://fr.wikipedia.org/wiki/Alimentation_humaine
```

[48]:
```
#On commence par faire une jointure entre le data frame population et␣
↪Dispo_alimentaire afin d'ajouter dans ce dernier la population (population␣
↪2017 /!\)


merged_df_Popu_ali = dispo_alimentaire[['Zone', 'Disponibilité alimentaire␣
↪(Kcal/personne/jour)']]
merged_df_Popu_ali = merged_df_Popu_ali.groupby('Zone').sum()
merged_df_Popu_ali = pd.merge(merged_df_Popu_ali, merged_df_2017, on=("Zone"),␣
↪how='inner')
merged_df_Popu_ali = merged_df_Popu_ali[['Zone', 'Disponibilité alimentaire␣
↪(Kcal/personne/jour)','Population']]
```

[49]:
```
#Affichage du nouveau dataframe
merged_df_Popu_ali.head()
```

[49]:

	Zone	Disponibilité alimentaire (Kcal/personne/jour)	Population
0	Afghanistan	2087.0	36296113.0
1	Afrique du Sud	3020.0	57009756.0
2	Albanie	3188.0	2884169.0
3	Algérie	3293.0	41389189.0
4	Allemagne	3503.0	82658409.0

[50]:
```
#Création de la colonne dispo_kcal avec calcul des kcal disponibles␣
↪mondialement (faire la somme de la colonne total calories disponible)

merged_df_Popu_ali['dispo kcal'] = merged_df_Popu_ali['Disponibilité␣
↪alimentaire (Kcal/personne/jour)'] * merged_df_Popu_ali['Population'] * 365
merged_df_Popu_ali.head()
```

[50]:

	Zone	Disponibilité alimentaire (Kcal/personne/jour)	Population	\
0	Afghanistan	2087.0	36296113.0	
1	Afrique du Sud	3020.0	57009756.0	
2	Albanie	3188.0	2884169.0	
3	Algérie	3293.0	41389189.0	
4	Allemagne	3503.0	82658409.0	

	dispo kcal
0	2.764875e+13
1	6.284185e+13
2	3.356077e+12
3	4.974753e+13
4	1.056866e+14

[51]:
```
dispo_kcal_monde = merged_df_Popu_ali['dispo kcal'].sum()
dispo_kcal_monde
```

[51]:
```
7635429388975815.0
```

[52]:
```
#Calcul du nombre d'humains pouvant être nourris (%)
kcalDispo = round(((merged_df_Popu_ali["Disponibilité alimentaire (Kcal/
 ↪personne/jour)"] * merged_df_Popu_ali["Population"])).sum())
# 2000 : nombre de calorie moyen pour un adulte
nb_pers_nourri = round(kcalDispo/2000)

print("Le nombre théorique de personne qui pourrais être nourri est␣
 ↪de",nb_pers_nourri,"personnes")
proportion_nourri = round(nb_pers_nourri/ total_pop *100)

print("Grace à la disponibilité de l'année 2017 nous pourrions␣
 ↪nourrire",proportion_nourri,"% de la population mondiale en 2017")
```

```
Le nombre théorique de personne qui pourrais être nourri est de 10459492314
personnes
Grace à la disponibilité de l'année 2017 nous pourrions nourrire 139 % de la
population mondiale en 2017
```

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_vege = dispo_alimentaire[['Zone', 'Origine', 'Disponibilité alimentaire␣
 ↪(Kcal/personne/jour)']]
dispo_vege = dispo_alimentaire[dispo_alimentaire['Origine'] == 'vegetale']

pop_dispo_veg = pd.merge(dispo_vege, merged_df_Popu_ali, on=("Zone"),␣
 ↪how='inner')
pop_dispo_veg = pop_dispo_veg[['Zone', 'Disponibilité alimentaire (Kcal/
 ↪personne/jour)_x','Population']]
pop_dispo_veg = pop_dispo_veg.rename(columns={'Disponibilité alimentaire (Kcal/
 ↪personne/jour)':'Disponibilité alimentaire vegetale (Kcal/personne/jour)'})
pop_dispo_veg.head()
```

[54]:

	Zone	Disponibilité alimentaire (Kcal/personne/jour)_x	Population
0	Afghanistan	1.0	36296113.0
1	Afghanistan	1.0	36296113.0
2	Afghanistan	0.0	36296113.0
3	Afghanistan	4.0	36296113.0
4	Afghanistan	0.0	36296113.0

[55]:
```
#Calcul du nombre de kcal disponible pour les végétaux
pop_dispo_veg['dispo kcal'] = pop_dispo_veg['Disponibilité alimentaire (Kcal/
 ↪personne/jour)_x'] * pop_dispo_veg['Population'] * 365
pop_dispo_veg
```

[55]:
```
                Zone  Disponibilité alimentaire (Kcal/personne/jour)_x  \
0        Afghanistan                                               1.0
1        Afghanistan                                               1.0
2        Afghanistan                                               0.0
3        Afghanistan                                               4.0
4        Afghanistan                                               0.0
…                ...                                               …
11746  Îles Salomon                                                0.0
11747  Îles Salomon                                                0.0
11748  Îles Salomon                                                0.0
11749  Îles Salomon                                                0.0
11750  Îles Salomon                                                4.0

       Population    dispo kcal
0      36296113.0  1.324808e+10
1      36296113.0  1.324808e+10
2      36296113.0  0.000000e+00
3      36296113.0  5.299232e+10
4      36296113.0  0.000000e+00
…             ...           ...
11746    636039.0  0.000000e+00
11747    636039.0  0.000000e+00
11748    636039.0  0.000000e+00
11749    636039.0  0.000000e+00
11750    636039.0  9.286169e+08

[11751 rows x 4 columns]
```

[56]:
```
#Calcul du nombre d'humains pouvant être nourris avec les végétaux
kcalDispo_veg = round(((pop_dispo_veg['Disponibilité alimentaire (Kcal/personne/
 ↪jour)_x'] * pop_dispo_veg["Population"])).sum())
# 2500 : nombre de calorie moyen pour un adulte
nb_pers_nourri_veg = round(kcalDispo_veg/2500)

print("Le nombre théorique de personne qui pourrais etre nourri, seulement␣
 ↪grace à la disponibilité alimentaire des produits végétaux, est␣
 ↪de",nb_pers_nourri_veg,"personnes")

proportion_nourri_veg = round((nb_pers_nourri_veg/total_pop)*100)

print("Grace à la disponibilité alimentaire des produits végétaux de l'année␣
 ↪2017 nous pourrions nourrir {:.2f} % de la population mondiale en 2017".
 ↪format(proportion_nourri_veg))
```

```
Le nombre théorique de personne qui pourrais etre nourri, seulement grace à la
disponibilité alimentaire des produits végétaux, est de 6904305685 personnes
Grace à la disponibilité alimentaire des produits végétaux de l'année 2017 nous
pourrions nourrir 92.00 % de la population mondiale en 2017
```

3.4 - Utilisation de la disponibilité intérieure

[58]:
```
#Calcul de la disponibilité totale
dispo_totale = round(dispo_alimentaire['Disponibilité intérieure'].sum())
dispo_totale
```

[58]: 9848994000000000000

[59]:
```
#création d'une boucle for pour afficher les différentes valeurs en fonction␣
 ↪des colonnes aliments pour animaux, pertes, nourritures,


liste_col = ['Aliments pour animaux', 'Pertes', 'Nourriture']
for element in liste_col:
    print('Proportion de',element, "{:.2f}".format(dispo_alimentaire[element].
 ↪sum() * 100 / dispo_totale),'%')
```

```
Proportion de Aliments pour animaux 13.24 %
Proportion de Pertes 4.61 %
Proportion de Nourriture 49.51 %
```

3.5 - Utilisation des céréales

[61]:
```
#Création d'une liste avec toutes les variables
cereales = ['Blé', 'Riz (Eq Blanchi)', 'Orge', 'Maïs' , 'Seigle', 'Avoine',␣
 ↪'Millet' ,'Sorgho', 'Céréales, Autres']
print(cereales)
```

```
['Blé', 'Riz (Eq Blanchi)', 'Orge', 'Maïs', 'Seigle', 'Avoine', 'Millet',
'Sorgho', 'Céréales, Autres']
```

[62]:
```
#Création d'un dataframe avec les informations uniquement pour ces céréales
cereales_dispo_ali = dispo_alimentaire.loc[dispo_alimentaire["Produit"].
 ↪isin(cereales)]
cereales_dispo_ali
```

[62]:

	Zone	Produit	Origine	Aliments pour animaux
7	Afghanistan	Blé	vegetale	0.000000e+00
12	Afghanistan	Céréales, Autres	vegetale	0.000000e+00
32	Afghanistan	Maïs	vegetale	2.000000e+14
34	Afghanistan	Millet	vegetale	0.000000e+00
40	Afghanistan	Orge	vegetale	3.600000e+14
…	…	…	…	…
15545	Îles Salomon	Céréales, Autres	vegetale	0.000000e+00
15568	Îles Salomon	Maïs	vegetale	0.000000e+00
15575	Îles Salomon	Orge	vegetale	0.000000e+00
15591	Îles Salomon	Riz (Eq Blanchi)	vegetale	0.000000e+00
15593	Îles Salomon	Sorgho	vegetale	0.000000e+00

```
       Autres Utilisations  Disponibilité alimentaire (Kcal/personne/jour)  \
7             0.000000e+00                                          1369.0
12            0.000000e+00                                             0.0
32            0.000000e+00                                            21.0
34            0.000000e+00                                             3.0
40            0.000000e+00                                            26.0
…                    …                                                …
15545         0.000000e+00                                             0.0
15568         0.000000e+00                                             1.0
15575         0.000000e+00                                             0.0
15591         1.200000e+13                                           623.0
15593         0.000000e+00                                             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+15                       0.0
12                 0.000000e+00                       0.0
32                 3.130000e+14                       0.0
34                 1.300000e+13                       0.0
40                 5.240000e+14                       0.0
…                           …                         …
15545              0.000000e+00                       0.0
15568              0.000000e+00                       0.0
15575              1.000000e+12                       0.0
15591              4.900000e+13                       0.0
15593              0.000000e+00                       0.0

       Importations - Quantité    Nourriture        Pertes    Production  \
7                 1.173000e+15  4.895000e+15  7.750000e+14  5.169000e+15
12                0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
32                1.000000e+12  7.600000e+13  3.100000e+13  3.120000e+14
34                0.000000e+00  1.200000e+13  1.000000e+12  1.300000e+13
40                1.000000e+13  8.900000e+13  5.200000e+13  5.140000e+14
…                          …             …             …             …
15545             0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
15568             0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
15575             1.000000e+12  0.000000e+00  0.000000e+00  0.000000e+00
15591             4.700000e+13  3.600000e+13  1.000000e+12  3.000000e+12
15593             0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00

           Semences    Traitement  Variation de stock
7      3.220000e+14  0.000000e+00       -3.500000e+14
12     0.000000e+00  0.000000e+00        0.000000e+00
32     5.000000e+12  0.000000e+00        0.000000e+00
34     0.000000e+00  0.000000e+00        0.000000e+00
40     2.200000e+13  0.000000e+00        0.000000e+00
…               …             …                   …
15545  0.000000e+00  0.000000e+00        0.000000e+00
15568  0.000000e+00  0.000000e+00        0.000000e+00
15575  0.000000e+00  1.000000e+12        0.000000e+00
15591  0.000000e+00  0.000000e+00        0.000000e+00
15593  0.000000e+00  0.000000e+00        0.000000e+00

[1497 rows x 18 columns]
```

[63]:
```
#Affichage de la proportion d'alimentation humaine
alim_animale = round(cereales_dispo_ali['Aliments pour animaux'].sum())
alim_hum = round(cereales_dispo_ali['Autres Utilisations'].sum())
alim_total = alim_hum + alim_animale
pourc_alim_hum = alim_hum / (alim_total) * 100
print("La proportion de céréale visant à l'alimentation humaine est de ",␣
 ↪pourc_alim_hum , "% de la population mondiale en 2017")
```

```
La proportion de céréale visant à l'alimentation humaine est de
21.18400609209237 % de la population mondiale en 2017
```

[64]:
```
#Affichage de la proportion d'alimentation animale
pourc_alim_animal = alim_animale / (alim_total) * 100

print("La proportion de céréale visant à l'alimentation animale est de",␣
 ↪pourc_alim_animal, "% de la population mondiale en 2017")
```

```
La proportion de céréale visant à l'alimentation animale est de
78.81599390790764 % de la population mondiale en 2017
```

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

[66]:
```
#Création de la colonne proportion par pays
population_tmp = population.copy()
sous_nutrition_tmp = sous_nutrition.copy()

population_2017 = population_tmp[population_tmp['Année'] == 2017]
sous_nutrition_2017 = sous_nutrition_tmp[sous_nutrition_tmp['Année'] ==␣
 ↪'2016-2018']
sous_nutrition_2017.loc[:, 'Année'] = 2017

merged_df_2017 = pd.merge(population_2017, sous_nutrition_2017, on=['Zone',␣
 ↪'Année'])

merged_df_2017['proportion sous nutrition'] = merged_df_2017['sous_nutrition'] /
 ↪ merged_df_2017['Population'] * 100
merged_df_2017
```

[66]:

	Zone	Année	Population	sous_nutrition	\
0	Afghanistan	2017	36296113.0	10500000.0	
1	Afrique du Sud	2017	57009756.0	3100000.0	
2	Albanie	2017	2884169.0	100000.0	
3	Algérie	2017	41389189.0	1300000.0	
4	Allemagne	2017	82658409.0	0.0	
..	…	…	…	…	
198	Venezuela (République bolivarienne du)	2017	29402484.0	8000000.0	
199	Viet Nam	2017	94600648.0	6500000.0	
200	Yémen	2017	27834819.0	0.0	

```
201                                Zambie  2017  16853599.0            0.0
202                              Zimbabwe  2017  14236595.0            0.0

     proportion sous nutrition
0                    28.928718
1                     5.437666
2                     3.467203
3                     3.140917
4                     0.000000
..                         …
198                  27.208586
199                   6.870989
200                   0.000000
201                   0.000000
202                   0.000000

[203 rows x 5 columns]
```

[67]:
```
#affichage après trie des 10 pires pays
Pire_pays = merged_df_2017.sort_values(by='proportion sous nutrition',␣
 ↪ascending=False)
Pire_pays.head(10)
```

[67]:
```
                                        Zone Année  Population  \
78                                     Haïti  2017  10982366.0
157  République populaire démocratique de Corée  2017  25429825.0
108                                Madagascar  2017  25570512.0
103                                   Libéria  2017   4702226.0
100                                   Lesotho  2017   2091534.0
183                                     Tchad  2017  15016753.0
161                                    Rwanda  2017  11980961.0
121                                 Mozambique  2017  28649018.0
186                                Timor-Leste  2017   1243258.0
0                                  Afghanistan  2017  36296113.0

     sous_nutrition  proportion sous nutrition
78        5300000.0                  48.259182
157      12000000.0                  47.188685
108      10500000.0                  41.062924
103       1800000.0                  38.279742
100        800000.0                  38.249438
183       5700000.0                  37.957606
161       4200000.0                  35.055619
121       9400000.0                  32.810898
186        400000.0                  32.173531
0        10500000.0                  28.928718
```

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

[69]:
```
#calcul du total de l'aide alimentaire par pays
aide = aide_alimentaire[['Zone','Valeur']].groupby('Zone').sum()
aide
```

[69]:

Zone	Valeur
Afghanistan	185452000
Algérie	81114000
Angola	5014000
Bangladesh	348188000
Bhoutan	2666000
…	…
Zambie	3026000
Zimbabwe	62570000
Égypte	1122000
Équateur	1362000
Éthiopie	1381294000

[76 rows x 1 columns]

[70]:
```
#affichage après trie des 10 pays qui ont bénéficié le plus de l'aide␣
↪alimentaire
aide.sort_values(by=['Valeur'], ascending=False).head(10)
```

[70]:

Zone	Valeur
République arabe syrienne	1858943000
Éthiopie	1381294000
Yémen	1206484000
Soudan du Sud	695248000
Soudan	669784000
Kenya	552836000
Bangladesh	348188000
Somalie	292678000
République démocratique du Congo	288502000
Niger	276344000

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

[72]:
```
#Création d'un dataframe avec la zone, l'année et l'aide alimentaire puis␣
↪groupby sur zone et année

aide2 = aide_alimentaire[['Zone','Année','Valeur']].groupby(["Zone","Année"]).
↪sum().reset_index()
aide2
```

[72]:

	Zone	Année	Valeur
0	Afghanistan	2013	128238000

```
1     Afghanistan   2014   57214000
2         Algérie   2013   35234000
3         Algérie   2014   18980000
4         Algérie   2015   17424000
..            …    …         …
223       Égypte   2013    1122000
224     Équateur   2013    1362000
225     Éthiopie   2013  591404000
226     Éthiopie   2014  586624000
227     Éthiopie   2015  203266000

[228 rows x 3 columns]
```

[73]:
```
Top5Aide_liste = ['République arabe syrienne', 'Éthiopie', 'Yémen', 'Soudan du␣
↪Sud', 'Soudan']
Top5Aide_liste
```

[73]:
```
['République arabe syrienne', 'Éthiopie', 'Yémen', 'Soudan du Sud', 'Soudan']
```

[74]:
```
#On filtre sur le dataframe avec notre liste
aide_filtre = aide2.loc[aide2['Zone'].isin(Top5Aide_liste),:]

aide_filtre
```

[74]:
```
                          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
```

[75]:
```
# Affichage des pays avec l'aide alimentaire par année
```

3.9 - Pays avec le moins de disponibilité par habitant

[77]:
```
#Calcul de la disponibilité en kcal par personne par jour par pays
dispo = dispo_alimentaire.groupby('Zone')['Disponibilité alimentaire (Kcal/
 ↪personne/jour)'].sum()
DFdispo = pd.DataFrame(dispo, columns = ['Disponibilité alimentaire (Kcal/
 ↪personne/jour)'])
DFdispo
```

[77]:

Zone	Disponibilité alimentaire (Kcal/personne/jour)
Afghanistan	2087.0
Afrique du Sud	3020.0
Albanie	3188.0
Algérie	3293.0
Allemagne	3503.0
…	…
Émirats arabes unis	3275.0
Équateur	2346.0
États-Unis d'Amérique	3682.0
Éthiopie	2129.0
Îles Salomon	2383.0

[174 rows x 1 columns]

[78]:
```
#Affichage des 10 pays qui ont le moins de dispo alimentaire par personne
pays_moins_dispo = DFdispo.sort_values(by=['Disponibilité alimentaire (Kcal/
 ↪personne/jour)'], ascending=True).head(10)
pays_moins_dispo
```

[78]:

Zone	Disponibilité alimentaire (Kcal/personne/jour)
République centrafricaine	1879.0
Zambie	1924.0
Madagascar	2056.0
Afghanistan	2087.0
Haïti	2089.0
République populaire démocratique de Corée	2093.0
Tchad	2109.0
Zimbabwe	2113.0

```
Ouganda
2126.0
Timor-Leste
2129.0
```

3.10 - Pays avec le plus de disponibilité par habitant

[80]:
```
#Affichage des 10 pays qui ont le plus de dispo alimentaire par personne
pays_plus_dispo = DFdispo.sort_values(by=['Disponibilité alimentaire (Kcal/
 ↪personne/jour)'], ascending=False).head(10)
pays_plus_dispo
```

[80]:

Zone	Disponibilité alimentaire (Kcal/personne/jour)
Autriche	3770.0
Belgique	3737.0
Turquie	3708.0
États-Unis d'Amérique	3682.0
Israël	3610.0
Irlande	3602.0
Italie	3578.0
Luxembourg	3540.0
Égypte	3518.0
Allemagne	3503.0

3.11 - Exemple de la Thaïlande pour le Manioc

[82]:
```
#création d'un dataframe avec uniquement la Thaïlande

dispo_thaï = dispo_alimentaire[dispo_alimentaire['Zone'] == 'Thaïlande']
dispo_thaï_produit = dispo_thaï.groupby('Produit').sum().reset_index()
dispo_thaï_produit
```

[82]:

	Produit	Zone	Origine	Aliments pour animaux	\
0	Abats Comestible	Thaïlande	animale	0.0	
1	Agrumes, Autres	Thaïlande	vegetale	0.0	
2	Alcool, non Comestible	Thaïlande	vegetale	0.0	
3	Aliments pour enfants	Thaïlande	vegetale	0.0	
4	Ananas	Thaïlande	vegetale	0.0	
..	…	…	…	…	
90	Viande de Suides	Thaïlande	animale	0.0	
91	Viande de Volailles	Thaïlande	animale	0.0	
92	Viande, Autre	Thaïlande	animale	0.0	
93	Vin	Thaïlande	vegetale	0.0	
94	Épices, Autres	Thaïlande	vegetale	0.0	

	Autres Utilisations	Disponibilité alimentaire (Kcal/personne/jour)	\
0	0.000000e+00	3.0	

```
1          0.000000e+00                                                  0.0
2          3.580000e+14                                                  0.0
3          0.000000e+00                                                  2.0
4          0.000000e+00                                                 10.0
..                  …                                                    …
90         0.000000e+00                                                124.0
91         0.000000e+00                                                 52.0
92         0.000000e+00                                                  0.0
93         0.000000e+00                                                  0.0
94         0.000000e+00                                                 16.0

    Disponibilité alimentaire en quantité (kg/personne/an)  \
0                                                    1.11
1                                                    0.09
2                                                    0.00
3                                                    0.18
4                                                   10.02
..                                                    …
90                                                  13.00
91                                                  13.69
92                                                   0.03
93                                                   0.12
94                                                   1.70

    Disponibilité de matière grasse en quantité (g/personne/jour)  \
0                                                    0.09
1                                                    0.00
2                                                    0.00
3                                                    0.01
4                                                    0.04
..                                                    …
90                                                  11.83
91                                                   3.62
92                                                   0.01
93                                                   0.00
94                                                   0.30

    Disponibilité de protéines en quantité (g/personne/jour)  \
0                                                    0.56
1                                                    0.00
2                                                    0.00
3                                                    0.08
4                                                    0.08
..                                                    …
90                                                   3.92
91                                                   4.49
92                                                   0.02
```

```
93                                                0.00
94                                                0.43

    Disponibilité intérieure  Exportations - Quantité  \
0               7.400000e+13             5.000000e+12
1               8.000000e+12             6.000000e+12
2               3.580000e+14             1.100000e+14
3               1.200000e+13             7.000000e+12
4               7.820000e+14             1.449000e+15
..                       …                        …
90              8.710000e+14             2.200000e+13
91              9.450000e+14             5.360000e+14
92             -9.200000e+13             9.600000e+13
93              8.000000e+12             8.000000e+12
94              1.140000e+14             4.200000e+13

    Importations - Quantité    Nourriture        Pertes    Production  \
0              3.300000e+13  7.500000e+13  0.000000e+00  4.500000e+13
1              2.000000e+12  6.000000e+12  0.000000e+00  1.200000e+13
2              2.100000e+13  0.000000e+00  0.000000e+00  4.470000e+14
3              1.900000e+13  1.200000e+13  0.000000e+00  0.000000e+00
4              9.000000e+12  6.710000e+14  1.100000e+14  2.209000e+15
..                      …           …           …           …
90             1.000000e+12  8.710000e+14  0.000000e+00  8.910000e+14
91             1.100000e+13  9.170000e+14  2.800000e+13  1.470000e+15
92             4.000000e+12  2.000000e+12  0.000000e+00  0.000000e+00
93             1.600000e+13  8.000000e+12  0.000000e+00  0.000000e+00
94             1.300000e+13  1.140000e+14  0.000000e+00  1.430000e+14

    Semences    Traitement  Variation de stock
0        0.0  0.000000e+00        0.000000e+00
1        0.0  2.000000e+12        0.000000e+00
2        0.0  0.000000e+00        0.000000e+00
3        0.0  0.000000e+00        0.000000e+00
4        0.0  0.000000e+00        1.300000e+13
..       …           …                   …
90       0.0  0.000000e+00        0.000000e+00
91       0.0  0.000000e+00        0.000000e+00
92       0.0  0.000000e+00        0.000000e+00
93       0.0  0.000000e+00        0.000000e+00
94       0.0  0.000000e+00        0.000000e+00

[95 rows x 18 columns]
```

[83]:
```
#Calcul de la sous nutrition en Thaïlande
sous_nutrition_thaï = merged_df_2017[merged_df_2017['Zone'] == 'Thaïlande']
sous_nutrition_thaï
```

[83]:
```
          Zone Année  Population  sous_nutrition  proportion sous nutrition
185  Thaïlande  2017  69209810.0       6200000.0                   8.958268
```

[84]:
```
# On calcule la proportion exportée en fonction de la proportion
Thai_manioc = dispo_thaï_produit[dispo_thaï_produit['Produit'] == 'Manioc']
export_thaï = Thai_manioc['Exportations - Quantité'].sum() * 100 /␣
 ↪(Thai_manioc['Production'].sum())
print("La proportion de manioc exportée par la Thaïlande en fonction de sa␣
 ↪production est de {:.2f} %".format(export_thaï))
```

```
La proportion de manioc exportée par la Thaïlande en fonction de sa production
est de 83.41 %
```

Etape 6 - Analyse complémentaires

[86]:
```
#Rajouter en dessous toutes les analyses complémtaires 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"
```

[87]:
```
Thai_manioc
```

[87]:
```
   Produit       Zone   Origine  Aliments pour animaux  Autres Utilisations  \
50  Manioc  Thaïlande  vegetale           1.800000e+15         2.081000e+15

    Disponibilité alimentaire (Kcal/personne/jour)  \
50                                            40.0

    Disponibilité alimentaire en quantité (kg/personne/an)  \
50                                                13.0

    Disponibilité de matière grasse en quantité (g/personne/jour)  \
50                                                0.05

    Disponibilité de protéines en quantité (g/personne/jour)  \
50                                                0.14

    Disponibilité intérieure  Exportations - Quantité  \
50              6.264000e+15             2.521400e+16

    Importations - Quantité    Nourriture        Pertes    Production  \
50             1.250000e+15  8.710000e+14  1.511000e+15  3.022800e+16

    Semences  Traitement  Variation de stock
50       0.0         0.0                 0.0
```

[ ]:
```

```