belhadj-olfa-1-notebook-112023

January 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'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

[1]:
```
#Importation de la librairie Pandas
import pandas as pd
```

[2]:
```
import numpy as np
```

[3]:
```
import matplotlib.pyplot as plt
```

[4]:
```
import seaborn as sns
```

[5]:
```
import openpyxl
```

1.2 - Chargement des fichiers Excel

[6]:
```
#Importation du fichier population.csv
population_csv = pd.read_csv('/Users/helmisaddem/Downloads/DAN-P4-FAO/
 ↪population.csv')
```

[7]:
```
#Importation du fichier dispo_alimentaire.csv
dispo_alimentaire_csv = pd.read_csv('/Users/helmisaddem/Downloads/DAN-P4-FAO/
 ↪dispo_alimentaire.csv')
```

[8]:
```
#Importation du fichier aide_alimentaire.csv
aide_alimentaire_csv = pd.read_csv('/Users/helmisaddem/Downloads/DAN-P4-FAO/
 ↪aide_alimentaire.csv')
```

[9]:
```
#Importation du fichier sous_nutrition.csv
sous_nutrition_csv = pd.read_csv('/Users/helmisaddem/Downloads/DAN-P4-FAO/␣
 ↪sous_nutrition.csv')
```

Etape 2 - Analyse exploratoire des fichiers

2.1 - Analyse exploratoire du fichier population

[10]:
```
#Afficher les dimensions du dataset
print("Le tableau comporte {} observation(s) ou article(s)".
 ↪format(population_csv.shape[0]))
print("Le tableau comporte {} colonne(s)".format(population_csv.shape[1]))
```

```
Le tableau comporte 1416 observation(s) ou article(s)
Le tableau comporte 3 colonne(s)
```

[11]:
```
#Affichage les 5 premières lignes de la table
population_csv.head()
```

[11]:

	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

[12]:
```
dispo_alimentaire_csv.head()
```

[12]:

	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
```

[13]:
```
aide_alimentaire_csv.head()
```

[13]:

	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

[14]:
```
sous_nutrition_csv.head()
```

[14]:

	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

[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_csv['Valeur'] = population_csv['Valeur']*1000
```

[16]:
```
#changement du nom de la colonne Valeur par Population
population_csv.rename(columns = {'Valeur':'Population'}, inplace = True)
```

[17]:
```
#Affichage les 5 premières lignes de la table pour voir les modifications
population_csv.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

[18]:
```
#Afficher les dimensions du dataset
print("Le tableau comporte {} observation(s) ou article(s)".
 ↪format(dispo_alimentaire_csv.shape[0]))
print("Le tableau comporte {} colonne(s)".format(dispo_alimentaire_csv.
 ↪shape[1]))
```

```
Le tableau comporte 15605 observation(s) ou article(s)
Le tableau comporte 18 colonne(s)
```

[19]:
```
#Consulter le nombre de colonnes
print("Le tableau comporte {} colonne(s)".format(dispo_alimentaire_csv.
 ↪shape[1]))
```

```
Le tableau comporte 18 colonne(s)
```

[20]:
```
#Affichage les 5 premières lignes de la table
dispo_alimentaire_csv.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
0	Afghanistan	Abats Comestible	animale	NaN	NaN	5.0	1.72	0.20	0.77	53.0	NaN	NaN
1	Afghanistan	Agrumes, Autres	vegetale	NaN	NaN	1.0	1.29	0.01	0.02	41.0	2.0	40.0
2	Afghanistan	Aliments pour enfants	vegetale	NaN	NaN	1.0	0.06	0.01	0.03			
3	Afghanistan	Ananas	vegetale	NaN	NaN	0.0	0.00	NaN	NaN			
4	Afghanistan	Bananes	vegetale	NaN	NaN	4.0	2.70	0.02	0.05			

```
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
```

[21]:
```
#remplacement des NaN dans le dataset par des 0
dispo_alimentaire_csv = dispo_alimentaire_csv.replace(np.nan,0)
```

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

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

[23]:
```
          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

[24]:
```
#Afficher les dimensions du dataset
print("Le tableau comporte {} observation(s) ou article(s)".
  ↪format(aide_alimentaire_csv.shape[0]))
```

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

```
Le tableau comporte 1475 observation(s) ou article(s)
Le tableau comporte 4 colonne(s)
```

[25]:
```
#Consulter le nombre de colonnes
print("Le tableau comporte {} colonne(s)".format(aide_alimentaire_csv.shape[1]))
```

```
Le tableau comporte 4 colonne(s)
```

[26]:
```
#Affichage les 5 premières lignes de la table
aide_alimentaire_csv.head()
```

[26]:

	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

[27]:
```
#changement du nom de la colonne Pays bénéficiaire par Zone
aide_alimentaire_csv.rename(columns = {'Pays bénéficiaire':'Zone'}, inplace =␣
↪True)
```

[28]:
```
#Multiplication de la colonne Aide_alimentaire qui contient des tonnes par 1000␣
↪pour avoir des kg
aide_alimentaire_csv['Valeur'] = aide_alimentaire_csv['Valeur']*1000
```

[29]:
```
#Affichage les 5 premières lignes de la table
aide_alimentaire_csv.head()
```

[29]:

	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

[30]:
```
#Afficher les dimensions du dataset
print("Le tableau comporte {} observation(s) ou article(s)".
↪format(sous_nutrition_csv.shape[0]))
print("Le tableau comporte {} colonne(s)".format(sous_nutrition_csv.shape[1]))
```

```
Le tableau comporte 1218 observation(s) ou article(s)
Le tableau comporte 3 colonne(s)
```

[31]:
```
#Consulter le nombre de colonnes
print("Le tableau comporte {} colonne(s)".format(sous_nutrition_csv.shape[1]))
```

```
Le tableau comporte 3 colonne(s)
```

[32]:
```
#Afficher les 5 premières lignes de la table
sous_nutrition_csv.head()
```

[32]:

	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

[33]:
```
#Conversion de la colonne sous nutrition en numérique: problèmes des valeurs␣
↪<0,1 ==> remplacée par 0
sous_nutrition_csv['Valeur'] = sous_nutrition_csv['Valeur'].replace('<0.1','0')
```

[34]:
```
#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_csv['Valeur'] = pd.to_numeric(sous_nutrition_csv['Valeur'],␣
↪errors='coerce')
```

[35]:
```
sous_nutrition_csv = sous_nutrition_csv.replace(np.nan,0)
```

[36]:
```
#changement du nom de la colonne Valeur par sous_nutrition
sous_nutrition_csv.rename(columns = {'Valeur':'sous_nutrition'}, inplace = True)
```

[37]:
```
#Multiplication de la colonne sous_nutrition par 1000000
sous_nutrition_csv['sous_nutrition'] =␣
↪sous_nutrition_csv['sous_nutrition']*1000000
```

[38]:
```
#Afficher les 5 premières lignes de la table
sous_nutrition_csv.head()
```

[38]:

	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

[39]:
```
# Il faut tout d'abord faire une jointure entre la table population et la table␣
↪sous nutrition, en ciblant l'année 2017
sous_nutrition_2017 = sous_nutrition_csv.loc[sous_nutrition_csv['Année'] ==␣
↪'2016-2018', ['Zone', 'Année','sous_nutrition']]
```

[40]:
```
population_2017 = population_csv.loc[population_csv['Année'] == 2017, ['Zone',␣
↪'Année','Population']]
```

[41]:
```
sous_alimentation = pd.merge(sous_nutrition_2017, population_2017, on = 'Zone',␣
↪how = 'inner')
```

[42]:
```
sous_alimentation = sous_alimentation.rename(columns={'Année_y':'Annee'})
sous_alimentation = sous_alimentation.drop(columns=['Année_x'])
```

[43]:
```
#Affichage du dataset
sous_alimentation.head()
```

[43]:

	Zone	sous_nutrition	Annee	Population
0	Afghanistan	10500000.0	2017	36296113.0
1	Afrique du Sud	3100000.0	2017	57009756.0
2	Albanie	100000.0	2017	2884169.0
3	Algérie	1300000.0	2017	41389189.0
4	Allemagne	0.0	2017	82658409.0

[44]:
```
#Calcul et affichage du nombre de personnes en état de sous nutrition: on ne␣
↪dispose pas de données pour tous les pays, on ne fera de calcul que sur les␣
↪pays dont on a les chiffres
sous_alimentation['pourcentage_sous_nutrition'] =␣
↪round(((sous_alimentation['sous_nutrition'].div␣
↪(sous_alimentation['Population']))*100),2)
pays_documentes = sous_alimentation.
↪loc[sous_alimentation['pourcentage_sous_nutrition'] != 0]
pourcentage_sous_nutrition_monde = round(((pays_documentes['sous_nutrition'].
↪sum() / pays_documentes['Population'].sum())*100),2)
```

[45]:
```
print('nombre de personnes en état de sous nutrition dans les pays dont on␣
↪dispose des chiffres (84) est', pays_documentes['sous_nutrition'].sum())
```

```
nombre de personnes en état de sous nutrition dans les pays dont on dispose des
chiffres (84) est 535700000.0
```

[46]:
```
print ('le pourcentage de personnes dénutris en 2017 selon les données dont␣
↪dispose est', pourcentage_sous_nutrition_monde,'%' )
```

```
le pourcentage de personnes dénutris en 2017 selon les données dont dispose est
12.85 %
```

3.2 - Nombre théorique de personne qui pourrait être nourries

[47]:
```
#Combien mange en moyenne un être humain ? Source =>  2200 Kcal/jour
```

[48]:
```
#On commence par faire une jointure entre le data frame population et␣
 ↪Dispo_alimentaire afin d'ajouter dans ce dernier la population : problème␣
 ↪noms de 2 pays : royaume  uni et tchequie
population_2017['Zone'] = population_2017['Zone'].replace(["Royaume-Uni de␣
 ↪Grande-Bretagne et d'Irlande du Nord", "Tchéquie"], ["Royaume-Uni",␣
 ↪"Tchéquie (la)"])
df_dispo = pd.merge(dispo_alimentaire_csv, population_2017, on = 'Zone', how =␣
 ↪'inner')
del df_dispo['Année']
```

[49]:
```
df_dispo.head()
```

[49]:

	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)
0	Afghanistan	Abats Comestible	animale	0.0	0.0	5.0	1.72	0.20	0.77
1	Afghanistan	Agrumes, Autres	vegetale	0.0	0.0	1.0	1.29	0.01	0.02
2	Afghanistan	Aliments pour enfants	vegetale	0.0	0.0	1.0	0.06	0.01	0.03
3	Afghanistan	Ananas	vegetale	0.0	0.0	0.0	0.00	0.00	0.00
4	Afghanistan	Bananes	vegetale	0.0	0.0	4.0	2.70	0.02	

```
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  \
0  53000000.0        0.0  53000000.0       0.0         0.0
1  39000000.0  2000000.0   3000000.0       0.0         0.0
2   2000000.0        0.0         0.0       0.0         0.0
3         0.0        0.0         0.0       0.0         0.0
4  82000000.0        0.0         0.0       0.0         0.0

   Variation de stock  Population
0                 0.0  36296113.0
1                 0.0  36296113.0
2                 0.0  36296113.0
3                 0.0  36296113.0
4                 0.0  36296113.0
```

[50]:
```
disponibilite = df_dispo.drop(columns=["Origine",
                                       "Aliments pour animaux",
                                       "Autres Utilisations",
                                       "Disponibilité alimentaire en quantité␣
 ↪(kg/personne/an)",
                                       "Disponibilité de matière grasse en␣
 ↪quantité (g/personne/jour)",
                                       "Disponibilité de protéines en quantité␣
 ↪(g/personne/jour)",
                                       "Exportations - Quantité", "Importations␣
 ↪- Quantité",
                                       "Nourriture", "Pertes", "Production",␣
 ↪"Semences", "Traitement",
                                       "Variation de stock"])
```

[51]:
```
#Affichage du nouveau dataframe
disponibilite.head()
```

[51]:
```
          Zone                Produit  \
0  Afghanistan       Abats Comestible
1  Afghanistan        Agrumes, Autres
2  Afghanistan  Aliments pour enfants
3  Afghanistan                Ananas
4  Afghanistan                Bananes
```

```
   Disponibilité alimentaire (Kcal/personne/jour)  Disponibilité intérieure  \
0                                             5.0                53000000.0
1                                             1.0                41000000.0
2                                             1.0                 2000000.0
3                                             0.0                       0.0
4                                             4.0                82000000.0

   Population
0  36296113.0
1  36296113.0
2  36296113.0
3  36296113.0
4  36296113.0
```

[52]:
```
dispo_agg = disponibilite.groupby('Zone').agg({'Disponibilité alimentaire (Kcal/
↪personne/jour)':'sum', 'Disponibilité intérieure':'sum'})
```

[53]:
```
dispo_agg.reset_index()
```

[53]:
```
                     Zone  Disponibilité alimentaire (Kcal/personne/jour)  \
0              Afghanistan                                          2087.0
1            Afrique du Sud                                          3020.0
2                   Albanie                                          3188.0
3                   Algérie                                          3293.0
4                 Allemagne                                          3503.0
..                      …                                             …
169    Émirats arabes unis                                          3275.0
170               Équateur                                          2346.0
171  États-Unis d'Amérique                                          3682.0
172               Éthiopie                                          2129.0
173           Îles Salomon                                          2383.0

     Disponibilité intérieure
0                1.351500e+10
1                6.125600e+10
2                4.758000e+09
3                4.263000e+10
4                1.622750e+11
..                        …
169              1.185000e+10
170              1.861400e+10
171              7.779920e+11
172              4.401400e+10
173              6.950000e+08

[174 rows x 3 columns]
```

[54]:
```
dispo_agg_pop = pd.merge(dispo_agg, population_2017, on = 'Zone', how = 'inner')
```

[55]:
```
#Création de la colonne dispo_kcal avec calcul des kcal disponibles mondialement
dispo_agg_pop['Kcal_dispo_pays'] = round((dispo_agg_pop['Disponibilité␣
 ↪alimentaire (Kcal/personne/jour)'] * dispo_agg_pop['Population']), 2)
```

[56]:
```
dispo_agg_pop.head()
```

[56]:

	Zone	Disponibilité alimentaire (Kcal/personne/jour)	Disponibilité intérieure	Année	Population	Kcal_dispo_pays
0	Afghanistan	2087.0	1.351500e+10	2017	36296113.0	7.574999e+10
1	Afrique du Sud	3020.0	6.125600e+10	2017	57009756.0	1.721695e+11
2	Albanie	3188.0	4.758000e+09	2017	2884169.0	9.194731e+09
3	Algérie	3293.0	4.263000e+10	2017	41389189.0	1.362946e+11
4	Allemagne	3503.0	1.622750e+11	2017	82658409.0	2.895524e+11

[57]:
```
Kcal_dispo_mondial = dispo_agg_pop['Kcal_dispo_pays'].sum()
```

[58]:
```
print('nombre de calories disponibles mondialement',Kcal_dispo_mondial)
```

```
nombre de calories disponibles mondialement 21182162746926.0
```

[59]:
```
#Calcul du nombre d'humains pouvant être nourris sur la base de 2200 Kcal/jour␣
 ↪besoin jounalier moyen : 21182162746926.0/2200
nbre_humains_nourris = round((Kcal_dispo_mondial / 2200), 2)
print(nbre_humains_nourris,'pourraient être nourris en 2017 soit',␣
 ↪round((nbre_humains_nourris / (dispo_agg_pop['Population'].sum()))*100,␣
 ↪2),'% de la population mondiale en 2017')
```

```
9628255794.06 pourraient être nourris en 2017 soit 130.65 % de la population
mondiale en 2017
```

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

[60]:
```
#Transfert des données avec les végétaux dans un nouveau dataframe
df_vegetaux = df_dispo.loc[df_dispo['Origine'] == 'vegetale']
```

[61]:
```
#Calcul du nombre de kcal disponible pour les végétaux
disponibilite_vegetaux = df_vegetaux.drop(columns=["Origine",
                                        "Aliments pour animaux",
                                        "Autres Utilisations",
```

```
                                        "Disponibilité alimentaire en quantité␣
↪(kg/personne/an)",
                                        "Disponibilité de matière grasse en␣
↪quantité (g/personne/jour)",
                                        "Disponibilité de protéines en quantité␣
↪(g/personne/jour)",
                                        "Exportations - Quantité", "Importations␣
↪- Quantité",
                                        "Nourriture", "Pertes", "Production",␣
↪"Semences", "Traitement",
                                        "Variation de stock"])
```

[62]:
```
dispo_vegetaux_agg = disponibilite_vegetaux.groupby('Zone').agg({'Disponibilité␣
↪alimentaire (Kcal/personne/jour)':'sum', 'Disponibilité intérieure':'sum'})
```

[63]:
```
dispo_vegetaux_agg.reset_index()
```

[63]:

	Zone	Disponibilité alimentaire (Kcal/personne/jour)	Disponibilité intérieure
0	Afghanistan	1871.0	1.088400e+10
1	Afrique du Sud	2533.0	5.343900e+10
2	Albanie	2203.0	3.326000e+09
3	Algérie	2915.0	3.551000e+10
4	Allemagne	2461.0	1.268660e+11
..	…	…	…
169	Émirats arabes unis	2718.0	9.623000e+09
170	Équateur	1732.0	1.492600e+10
171	États-Unis d'Amérique	2698.0	6.320230e+11
172	Éthiopie	2005.0	3.882400e+10
173	Îles Salomon	2187.0	6.510000e+08

[174 rows x 3 columns]

[64]:
```
dispo_vegetaux_agg_pop = pd.merge(dispo_vegetaux_agg, population_2017, on =␣
↪'Zone', how = 'inner')
dispo_vegetaux_agg_pop.head()
```

[64]:

	Zone	Disponibilité alimentaire (Kcal/personne/jour)	Disponibilité intérieure	Année	Population
0	Afghanistan	1871.0	1.088400e+10	2017	36296113.0
1	Afrique du Sud	2533.0	5.343900e+10	2017	57009756.0
2	Albanie	2203.0	3.326000e+09	2017	2884169.0
3	Algérie	2915.0	3.551000e+10	2017	41389189.0
4	Allemagne	2461.0	1.268660e+11	2017	82658409.0

[65]:
```
dispo_vegetaux_agg_pop['Kcal_dispo_vegetaux_pays'] =␣
↪round((dispo_vegetaux_agg_pop['Disponibilité alimentaire (Kcal/personne/
↪jour)'] * dispo_vegetaux_agg_pop['Population']), 2)
dispo_vegetaux_agg_pop.head()
```

[65]:

	Zone	Disponibilité alimentaire (Kcal/personne/jour)	Disponibilité intérieure	Année	Population	Kcal_dispo_vegetaux_pays
0	Afghanistan	1871.0	1.088400e+10	2017	36296113.0	6.791003e+10
1	Afrique du Sud	2533.0	5.343900e+10	2017	57009756.0	1.444057e+11
2	Albanie	2203.0	3.326000e+09	2017	2884169.0	6.353824e+09
3	Algérie	2915.0	3.551000e+10	2017	41389189.0	1.206495e+11
4	Allemagne	2461.0	1.268660e+11	2017	82658409.0	2.034223e+11

[66]:
```
Kcal_vegetaux_dispo_mondial =␣
↪dispo_vegetaux_agg_pop['Kcal_dispo_vegetaux_pays'].sum()
print(Kcal_vegetaux_dispo_mondial)
```

```
17449509418936.0
```

[67]:
```
#Calcul du nombre d'humains pouvant être nourris avec les végétaux
nbre_humains_nourris_vegetaux = round((Kcal_vegetaux_dispo_mondial / 2200), 2)
print(nbre_humains_nourris_vegetaux,"pourraient être nourris uniquement par les␣
↪produits d'origine végétale en 2017 soit",
```

```
    round((nbre_humains_nourris_vegetaux /␣
↪(dispo_vegetaux_agg_pop['Population'].sum()))*100, 2),'% de la population␣
↪mondiale en 2017')
```

```
7931595190.43 pourraient être nourris uniquement par les produits d'origine
végétale en 2017 soit 107.63 % de la population mondiale en 2017
```

3.4 - Utilisation de la disponibilité intérieure

[68]:
```
#Calcul de la disponibilité totale
dispo_totale_mondiale = round((dispo_agg_pop['Disponibilité intérieure'].
↪sum()),2)
print('La disponibilité intérieure mondiale totale pour 2017',␣
↪dispo_totale_mondiale)
```

```
La disponibilité intérieure mondiale totale pour 2017 9848994000000.0
```

[69]:
```
distribution_dispo_int = df_dispo.drop(columns=["Origine",
                                                "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)",
                                                "Production",
                                                "Exportations - Quantité",␣
↪"Importations - Quantité",
                                                "Population",
                                                "Variation de stock"])
```

[70]:
```
distribution_dispo_int.head()
```

[70]:

	Zone	Produit	Aliments pour animaux	Autres Utilisations	Disponibilité intérieure	Nourriture	Pertes
0	Afghanistan	Abats Comestible	0.0	0.0	53000000.0	53000000.0	0.0
1	Afghanistan	Agrumes, Autres	0.0	0.0	41000000.0	39000000.0	2000000.0
2	Afghanistan	Aliments pour enfants	0.0	0.0	2000000.0	2000000.0	0.0
3	Afghanistan	Ananas	0.0	0.0	0.0	0.0	0.0
4	Afghanistan	Bananes	0.0	0.0	82000000.0	82000000.0	0.0

```
   Semences  Traitement
0       0.0         0.0
1       0.0         0.0
2       0.0         0.0
3       0.0         0.0
4       0.0         0.0
```

[71]:
```
distribution_dispo_int_agg = distribution_dispo_int.groupby('Zone').
↪agg({'Aliments pour animaux': 'sum',
                                                                    ␣
↪'Autres Utilisations': 'sum',
                                                                    ␣
↪'Nourriture':'sum',
                                                                    ␣
↪'Pertes': 'sum',
                                                                    ␣
↪'Semences': 'sum',
                                                                    ␣
↪'Traitement': 'sum',
                                                                    ␣
↪'Disponibilité intérieure': 'sum'})
```

[72]:
```
distribution_dispo_int_agg.reset_index()
```

[72]:
```
                     Zone  Aliments pour animaux  Autres Utilisations  \
0              Afghanistan           7.680000e+08         4.150000e+08
1            Afrique du Sud           5.309000e+09         8.760000e+08
2                  Albanie           6.600000e+08         1.740000e+08
3                  Algérie           4.352000e+09         1.707000e+09
4                Allemagne           3.020900e+10         7.711000e+09
..                     …                    …                    …
169    Émirats arabes unis           1.174000e+09         2.689000e+09
170               Équateur           1.200000e+09         1.909000e+09
171  États-Unis d'Amérique           1.484320e+11         1.546990e+11
172               Éthiopie           6.850000e+08         3.641000e+09
173           Îles Salomon           0.000000e+00         2.370000e+08

       Nourriture        Pertes      Semences    Traitement  \
0    1.073500e+10  1.135000e+09  3.950000e+08  6.300000e+07
1    2.981200e+10  2.193000e+09  2.530000e+08  2.281900e+10
2    3.476000e+09  2.760000e+08  5.500000e+07  1.130000e+08
3    3.172900e+10  3.753000e+09  4.490000e+08  6.370000e+08
4    7.923800e+10  3.781000e+09  1.551000e+09  3.984200e+10
..            …             …             …             …
169  6.548000e+09  7.050000e+08  3.000000e+06  8.040000e+08
```

```
170  8.574000e+09  7.070000e+08  1.380000e+08  6.200000e+09
171  3.208890e+11  7.162000e+09  1.040300e+10  1.410100e+11
172  3.336100e+10  2.256000e+09  6.400000e+08  3.443000e+09
173  3.620000e+08  6.000000e+06  0.000000e+00  9.000000e+07

     Disponibilité intérieure
0                1.351500e+10
1                6.125600e+10
2                4.758000e+09
3                4.263000e+10
4                1.622750e+11
..                        …
169              1.185000e+10
170              1.861400e+10
171              7.779920e+11
172              4.401400e+10
173              6.950000e+08

[174 rows x 8 columns]
```

[73]:
```
distribution_dispo_int_agg['prop_animaux'] =␣
 ↪round(((distribution_dispo_int_agg['Aliments pour animaux']/
 ↪distribution_dispo_int_agg['Disponibilité intérieure'])*100),2)
distribution_dispo_int_agg['prop_utilisations'] =␣
 ↪round(((distribution_dispo_int_agg['Autres Utilisations']/
 ↪distribution_dispo_int_agg['Disponibilité intérieure'])*100),2)
distribution_dispo_int_agg['prop_nourriture'] =␣
 ↪round(((distribution_dispo_int_agg['Nourriture']/
 ↪distribution_dispo_int_agg['Disponibilité intérieure'])*100),2)
distribution_dispo_int_agg['prop_pertes'] =␣
 ↪round(((distribution_dispo_int_agg['Pertes']/
 ↪distribution_dispo_int_agg['Disponibilité intérieure'])*100),2)
distribution_dispo_int_agg['prop_semences'] =␣
 ↪round(((distribution_dispo_int_agg['Semences']/
 ↪distribution_dispo_int_agg['Disponibilité intérieure'])*100),2)
distribution_dispo_int_agg['prop_traitement'] =␣
 ↪round(((distribution_dispo_int_agg['Traitement']/
 ↪distribution_dispo_int_agg['Disponibilité intérieure'])*100),2)
```

[74]:
```
distribution_dispo_int_agg.head()
```

[74]:
```
                Aliments pour animaux  Autres Utilisations    Nourriture  \
Zone
Afghanistan              7.680000e+08         4.150000e+08  1.073500e+10
Afrique du Sud           5.309000e+09         8.760000e+08  2.981200e+10
Albanie                  6.600000e+08         1.740000e+08  3.476000e+09
Algérie                  4.352000e+09         1.707000e+09  3.172900e+10
```

```
Allemagne                3.020900e+10         7.711000e+09  7.923800e+10

                     Pertes      Semences    Traitement  \
Zone
Afghanistan     1.135000e+09  3.950000e+08  6.300000e+07
Afrique du Sud  2.193000e+09  2.530000e+08  2.281900e+10
Albanie         2.760000e+08  5.500000e+07  1.130000e+08
Algérie         3.753000e+09  4.490000e+08  6.370000e+08
Allemagne       3.781000e+09  1.551000e+09  3.984200e+10

                Disponibilité intérieure  prop_animaux  prop_utilisations  \
Zone
Afghanistan                 1.351500e+10          5.68               3.07
Afrique du Sud              6.125600e+10          8.67               1.43
Albanie                     4.758000e+09         13.87               3.66
Algérie                     4.263000e+10         10.21               4.00
Allemagne                   1.622750e+11         18.62               4.75

                prop_nourriture  prop_pertes  prop_semences  prop_traitement
Zone
Afghanistan               79.43         8.40           2.92             0.47
Afrique du Sud            48.67         3.58           0.41            37.25
Albanie                   73.06         5.80           1.16             2.37
Algérie                   74.43         8.80           1.05             1.49
Allemagne                 48.83         2.33           0.96            24.55
```

[75]:
```
distribution_dispo_int_agg = distribution_dispo_int_agg.reset_index()
```

[76]:
```
distribution_dispo_int_agg.head()
```

[76]:
```
             Zone  Aliments pour animaux  Autres Utilisations    Nourriture  \
0     Afghanistan           7.680000e+08         4.150000e+08  1.073500e+10
1  Afrique du Sud           5.309000e+09         8.760000e+08  2.981200e+10
2         Albanie           6.600000e+08         1.740000e+08  3.476000e+09
3         Algérie           4.352000e+09         1.707000e+09  3.172900e+10
4       Allemagne           3.020900e+10         7.711000e+09  7.923800e+10

         Pertes      Semences    Traitement  Disponibilité intérieure  \
0  1.135000e+09  3.950000e+08  6.300000e+07              1.351500e+10
1  2.193000e+09  2.530000e+08  2.281900e+10              6.125600e+10
2  2.760000e+08  5.500000e+07  1.130000e+08              4.758000e+09
3  3.753000e+09  4.490000e+08  6.370000e+08              4.263000e+10
4  3.781000e+09  1.551000e+09  3.984200e+10              1.622750e+11

   prop_animaux  prop_utilisations  prop_nourriture  prop_pertes  \
0          5.68               3.07            79.43         8.40
1          8.67               1.43            48.67         3.58
```

```
2          13.87                3.66            73.06          5.80
3          10.21                4.00            74.43          8.80
4          18.62                4.75            48.83          2.33

   prop_semences  prop_traitement
0           2.92             0.47
1           0.41            37.25
2           1.16             2.37
3           1.05             1.49
4           0.96            24.55
```

[92]:
```
#création d'une boucle for pour afficher les différentes valeurs en fonction
 ↪des colonnes aliments pour animaux, pertes, nourritures,
for column in distribution_dispo_int_agg.columns[1:7]:
    print (column, round(((distribution_dispo_int_agg[column].sum()/
 ↪dispo_totale_mondiale)*100),2),'%')
```

```
Aliments pour animaux 13.24 %
Autres Utilisations 8.78 %
Nourriture 49.51 %
Pertes 4.61 %
Semences 1.57 %
Traitement 22.38 %
```

[93]:
```
value =[]
for column in distribution_dispo_int_agg.columns[1:7]:
    value.append(round(((distribution_dispo_int_agg[column].sum()/
 ↪dispo_totale_mondiale)*100),2))
```

[94]:
```
print(value)
```

```
[13.24, 8.78, 49.51, 4.61, 1.57, 22.38]
```

[82]:
```
data = pd.DataFrame({'Utilisation':['Aliments pour animaux',
                                    'Autres Utilisations', 'Nourriture',
                                    'Pertes', 'Semences', 'Traitement'],
                     'Valeur':[1, 2, 3, 4, 5, 6]})
```

[91]:
```
data.head(7)
```

[91]:
```
             Utilisation  Valeur
0  Aliments pour animaux       1
1    Autres Utilisations       2
2             Nourriture       3
3                 Pertes       4
4               Semences       5
5             Traitement       6
```

[95]:
```
data.Valeur = value
```

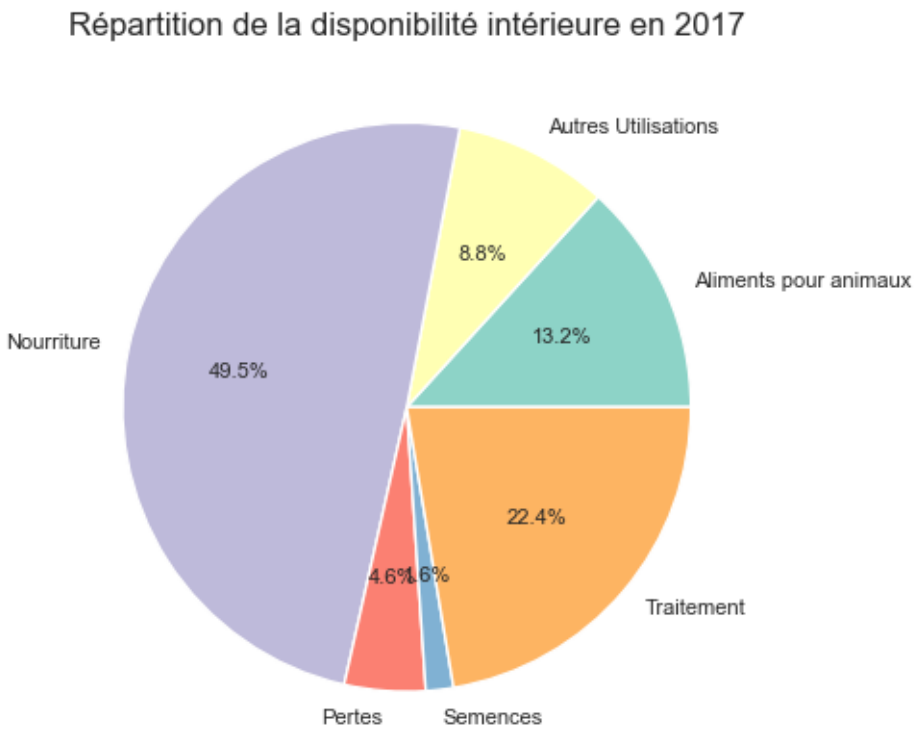
[96]:
```
data.head(10)
```

[96]:

	Utilisation	Valeur
0	Aliments pour animaux	13.24
1	Autres Utilisations	8.78
2	Nourriture	49.51
3	Pertes	4.61
4	Semences	1.57
5	Traitement	22.38

[ ]:
```
sns.set_style("darkgrid")
sns.set_palette("muted")
custom_palette = sns.color_palette("Set3", 9)
plt.pie(x = data['Valeur'], labels = data['Utilisation'],␣
 ↪colors=custom_palette, autopct='%.1f%%', textprops={'fontsize': 8})
plt.Figure(figsize=(4,4))
plt.title("Répartition de la disponibilité intérieure en 2017", loc = 'center')
```

[ ]:
```
Text(0.5, 1.0, 'Répartition de la disponibilité intérieure en 2017')
```



3.5 - Utilisation des céréales

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

[ ]:
```
dispo_cereales = dispo_alimentaire_csv[dispo_alimentaire_csv['Produit'].
 ↪isin(cereales)]
```

[ ]:
```
dispo_cereales["Disponibilité intérieure"].sum()
```

[ ]:
```
2406999000000.0
```

[ ]:
```
dispo_cereales = dispo_cereales.drop(columns=["Nourriture",
                              "Pertes",
                              "Production",
                              "Semences",
                              "Traitement",
                              "Exportations - Quantité", "Importations -␣
 ↪Quantité",
                              "Variation de stock"])
```

[ ]:
```
dispo_cereales_pays = pd.merge(dispo_cereales, population_2017, on = 'Zone',␣
 ↪how = 'inner')
```

[ ]:
```
dispo_cereales_pays['cereal_pays'] = dispo_cereales_pays['Disponibilité␣
 ↪alimentaire en quantité (kg/personne/an)'] *␣
 ↪dispo_cereales_pays['Population']
```

[ ]:
```
dispo_cereales_pays.head(40)
```

[ ]:
```
#Affichage de la proportion d'alimentation animale :
prop_animaux_cereales = round(((dispo_cereales_pays['Aliments pour animaux'].
 ↪sum()/dispo_cereales_pays['Disponibilité intérieure'].sum())*100),2)
print("Part de l'alimentation animale du total des céréales documentées",␣
 ↪prop_animaux_cereales,'%')
part_humaine_cereales = round(((dispo_cereales_pays['cereal_pays'].sum()/
 ↪dispo_cereales_pays['Disponibilité intérieure'].sum())*100),2)
print("Part de l'alimentation humaine du total des céréales documentées",␣
 ↪part_humaine_cereales,'%')
```

```
Part de l'alimentation animale du total des céréales documentées 36.29 %
Part de l'alimentation humaine du total des céréales documentées 45.1 %
```

[ ]:
```
cereales_utilisations = dispo_cereales_pays.groupby('Produit').agg({'Aliments␣
 ↪pour animaux': 'sum',
                                                                    ␣
 ↪'Disponibilité alimentaire (Kcal/personne/jour)': 'sum',
```

```
 ↪'Disponibilité alimentaire en quantité (kg/personne/an)': 'sum',␣
 ↪'Disponibilité de matière grasse en quantité (g/personne/jour)':'sum',␣
 ↪'Disponibilité de protéines en quantité (g/personne/jour)':'sum',␣
 ↪'Disponibilité intérieure':'sum',␣
 ↪'cereal_pays':'sum'})
```

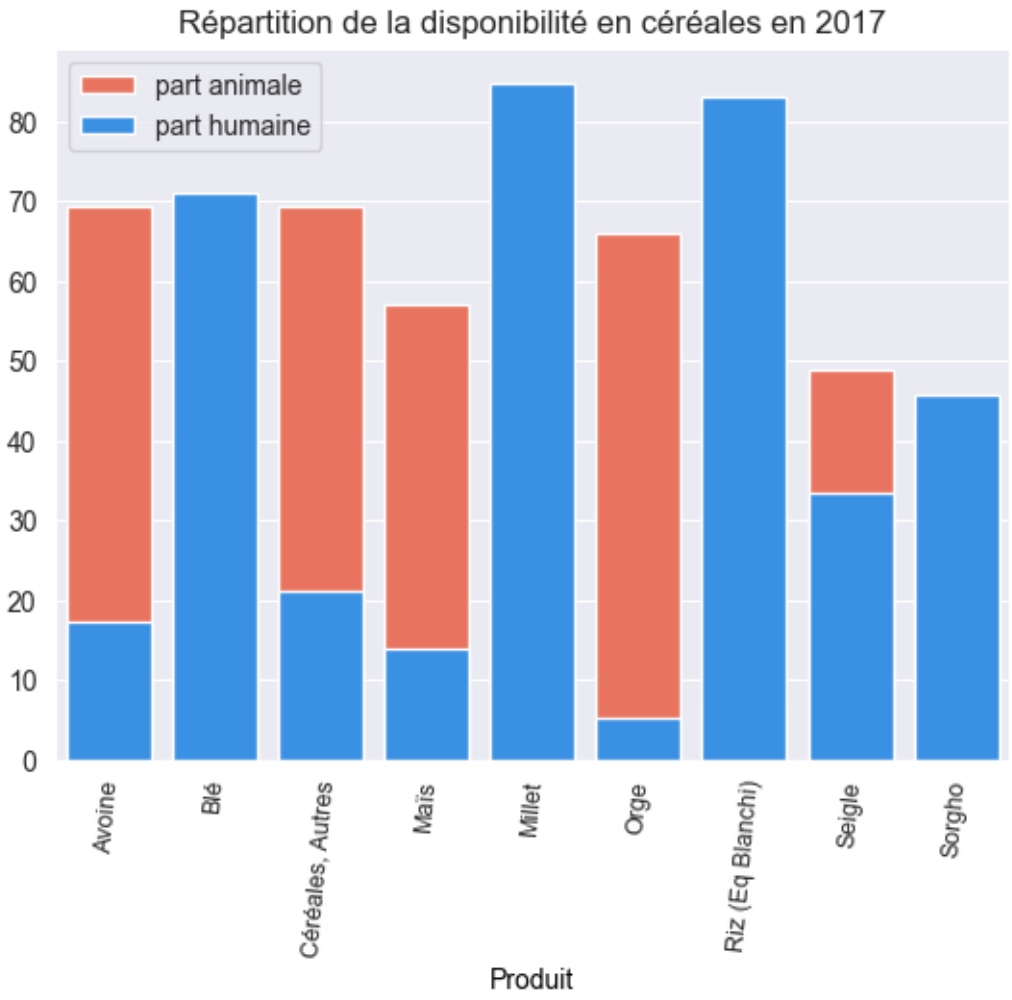
[ ]:
```
cereales_utilisations = cereales_utilisations.reset_index()
```

[ ]:
```
cereales_utilisations.head(30)
```

[ ]:
```
cereales_utilisations['prop_animaux %'] =␣
 ↪round(((cereales_utilisations['Aliments pour animaux']/
 ↪cereales_utilisations['Disponibilité intérieure'])*100),2)
cereales_utilisations['prop_humaine %'] =␣
 ↪round(((cereales_utilisations['cereal_pays']/
 ↪cereales_utilisations['Disponibilité intérieure'])*100),2)
```

[ ]:
```
sns.barplot(data=cereales_utilisations, x=cereales_utilisations['Produit'],␣
 ↪y=cereales_utilisations['prop_animaux %'], color='tomato', label='part␣
 ↪animale')
sns.barplot(data=cereales_utilisations, x=cereales_utilisations['Produit'],␣
 ↪y=cereales_utilisations['prop_humaine %'], color='dodgerblue', label = 'part␣
 ↪humaine')
plt.ylabel("")
plt.xlabel("")
plt.xlabel('Produit', fontsize=10, color='black')
plt.xticks(rotation = 85, fontsize=9)
plt.legend(loc='best')
plt.title("Répartition de la disponibilité en céréales en 2017", loc='center')
```

[ ]:
```
Text(0.5, 1.0, 'Répartition de la disponibilité en céréales en 2017')
```



[ ]:
```
#Analyse supplémentaire : production céréalière mondiale:

etude_cereales = dispo_alimentaire_csv[dispo_alimentaire_csv['Produit'].
 ↪isin(cereales)]
etude_cereales = etude_cereales.drop(columns=["Nourriture","Aliments pour␣
 ↪animaux","Origine", "Autres Utilisations",
                                               "Pertes", "Disponibilité␣
 ↪alimentaire (Kcal/personne/jour)",
                                               "Semences","Disponibilité␣
 ↪alimentaire en quantité (kg/personne/an)",
                                               "Traitement", "Disponibilité de␣
 ↪matière grasse en quantité (g/personne/jour)",
                                               "Exportations - Quantité",␣
 ↪"Importations - Quantité",
```

```
                                                    "Disponibilité de protéines en␣
 ↪quantité (g/personne/jour)",
                                                    "Variation de stock",␣
 ↪"Disponibilité intérieure"])
etude_cereales.head(50)
```

[ ]:
```
product_cerealiere = dispo_alimentaire_csv[dispo_alimentaire_csv['Produit'].
 ↪isin(cereales)].drop(columns=["Nourriture","Aliments pour␣
 ↪animaux","Origine", "Autres Utilisations",
                                                    "Pertes", "Disponibilité␣
 ↪alimentaire (Kcal/personne/jour)",
                                                    "Semences","Disponibilité␣
 ↪alimentaire en quantité (kg/personne/an)",
                                                    "Traitement", "Disponibilité de␣
 ↪matière grasse en quantité (g/personne/jour)",
                                                    "Exportations - Quantité",␣
 ↪"Importations - Quantité",
                                                    "Disponibilité de protéines en␣
 ↪quantité (g/personne/jour)",
                                                    "Variation de stock",␣
 ↪"Disponibilité intérieure"])
```

[ ]:
```
product_cerealiere = product_cerealiere.pivot_table(index='Zone',␣
 ↪values='Production', aggfunc='sum').reset_index().sort_values('Production',␣
 ↪ascending=False).reset_index()
```

[ ]:
```
product_cerealiere.head(50)
```

[ ]:
```
print(round(((product_cerealiere.iloc[0, 2]/(product_cerealiere['Production'].
 ↪sum()))*100),2))
```

```
19.22
```

[ ]:
```
etude_cereales = etude_cereales.pivot_table(index='Zone', columns='Produit',␣
 ↪values='Production', aggfunc='sum')
```

[ ]:
```
etude_cereales = etude_cereales.reset_index()
```

[ ]:
```
etude_cereales.head()
```

[ ]:
```
product_ble = etude_cereales.loc[:, ['Zone', 'Blé']].sort_values('Blé',␣
 ↪ascending=False).reset_index()
```

[ ]:
```
product_ble.head(10)
```

[ ]:
```
print(round(((product_ble.iloc[0, 2]/(product_ble['Blé'].sum()))*100),2))
```

```
17.21
```

[ ]:
```
print("La chine, continentale est le premier producteur céréalier :
 ↪",round(((product_cerealiere.iloc[0, 2]/(product_cerealiere['Production'].
 ↪sum()))*100),2), "% de la production céréalière dans le monde et le premier␣
 ↪producteur du blé en 2017 :",round(((product_ble.iloc[0, 2]/
 ↪(product_ble['Blé'].sum()))*100),2),"% de la production mondiale du blé")
```

```
La chine, continentale est le premier producteur céréalier : 19.22 % de la
production céréalière dans le monde et le premier producteur du blé en 2017 :
17.21 % de la production mondiale du blé
```

[ ]:
```
top_10_product_cerealiere = product_cerealiere.head(10)
```

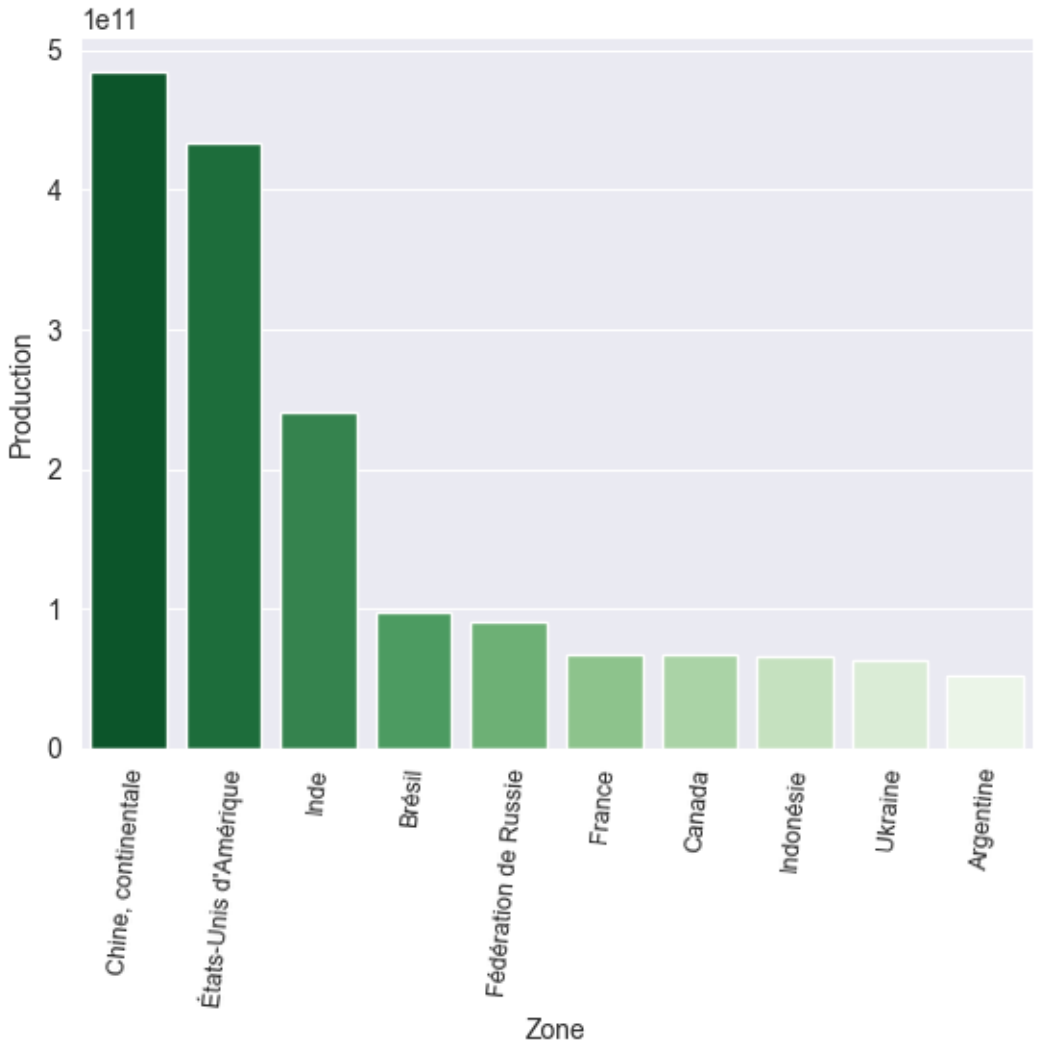
[ ]:
```
sns.barplot(data=top_10_product_cerealiere, x =␣
 ↪top_10_product_cerealiere['Zone'], y =␣
 ↪top_10_product_cerealiere['Production'], palette="Greens_r")
plt.xticks(rotation=85, fontsize=9)
```

```
/var/folders/r_/fd0gwkxn6n995_hk_5lc9f540000gn/T/ipykernel_95323/1411112225.py:1
: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.barplot(data=top_10_product_cerealiere, x =
top_10_product_cerealiere['Zone'], y = top_10_product_cerealiere['Production'],
palette="Greens_r")
```

[ ]:
```
([0, 1, 2, 3, 4, 5, 6, 7, 8, 9],
 [Text(0, 0, 'Chine, continentale'),
  Text(1, 0, "États-Unis d'Amérique"),
  Text(2, 0, 'Inde'),
  Text(3, 0, 'Brésil'),
  Text(4, 0, 'Fédération de Russie'),
  Text(5, 0, 'France'),
  Text(6, 0, 'Canada'),
  Text(7, 0, 'Indonésie'),
  Text(8, 0, 'Ukraine'),
  Text(9, 0, 'Argentine')])
```



[ ]:
```
etude_cereales_exportation =␣
 ↪dispo_alimentaire_csv[dispo_alimentaire_csv['Produit'].isin(cereales)]
etude_cereales_exportation = etude_cereales_exportation.
 ↪drop(columns=["Nourriture","Aliments pour animaux","Origine", "Autres␣
 ↪Utilisations",
                                                    "Pertes", "Disponibilité␣
 ↪alimentaire (Kcal/personne/jour)",
                                                    "Semences","Disponibilité␣
 ↪alimentaire en quantité (kg/personne/an)",
                                                    "Traitement", "Disponibilité de␣
 ↪matière grasse en quantité (g/personne/jour)",
                                                    "Production", "Importations -␣
 ↪Quantité",
```

```
                                                    "Disponibilité de protéines en␣
 ↪quantité (g/personne/jour)",
                                                    "Variation de stock",␣
 ↪"Disponibilité intérieure"])
etude_cereales_exportation.head(50)
```

[ ]:
```
plus_grand_export = etude_cereales_exportation.pivot_table(index='Zone',␣
 ↪values='Exportations - Quantité', aggfunc='sum').sort_values('Exportations -␣
 ↪Quantité', ascending=False).reset_index()
```

[ ]:
```
plus_grand_export = plus_grand_export.head(5)
```

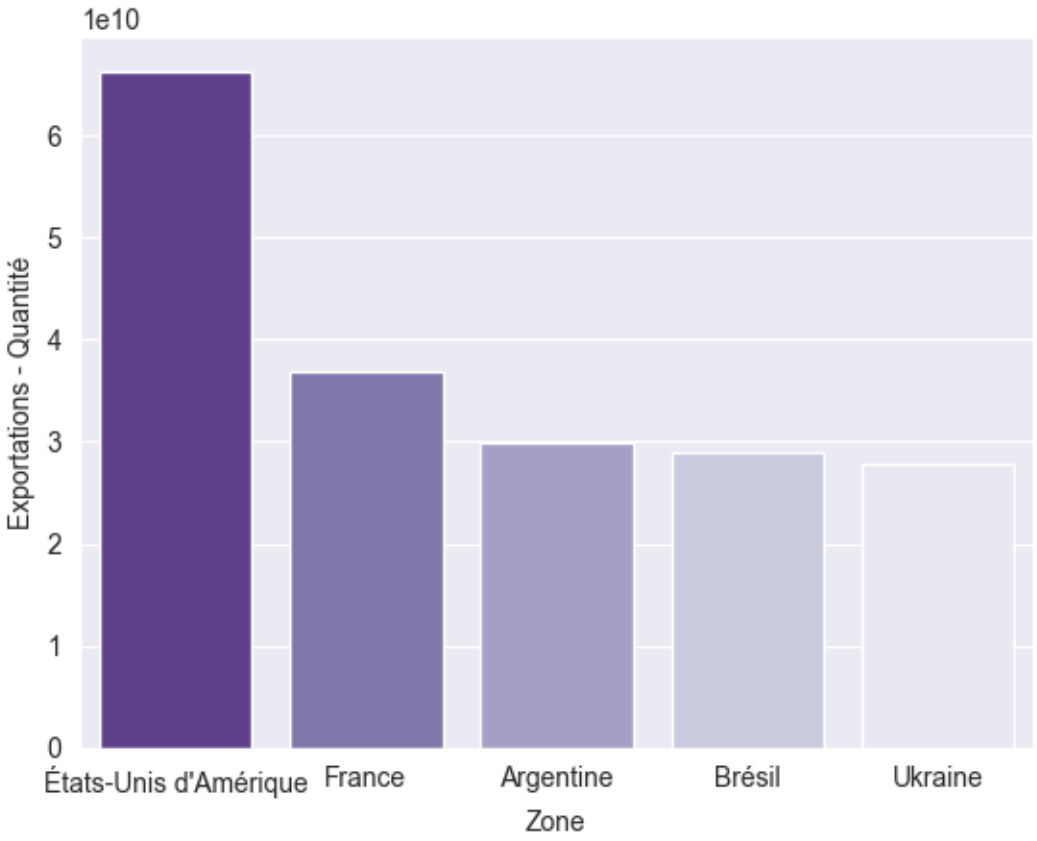
[ ]:
```
sns.barplot(data=plus_grand_export, x='Zone', y='Exportations - Quantité',␣
 ↪palette='Purples_r')
```

```
/var/folders/r_/fd0gwkxn6n995_hk_5lc9f540000gn/T/ipykernel_95323/1146881413.py:1
: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.barplot(data=plus_grand_export, x='Zone', y='Exportations - Quantité',
palette='Purples_r')
```

[ ]:
```
<Axes: xlabel='Zone', ylabel='Exportations - Quantité'>
```



[ ]:
```
plus_grand_export_ble = etude_cereales_exportation.
 ↪loc[etude_cereales_exportation['Produit']=='Blé', :].
 ↪sort_values('Exportations - Quantité', ascending=False).reset_index()
```

[ ]:
```
plus_grand_export_ble = plus_grand_export_ble.head(10)
plus_grand_export_ble.head(10)
```

[ ]:
```
chine_info = dispo_alimentaire_csv[dispo_alimentaire_csv['Produit'].
 ↪isin(cereales)]
chine = chine_info.loc[chine_info['Zone'] == "Chine, continentale", ['Produit',␣
 ↪'Production', 'Nourriture', 'Exportations - Quantité', 'Importations -␣
 ↪Quantité']].reset_index()
chine.head(20)
```

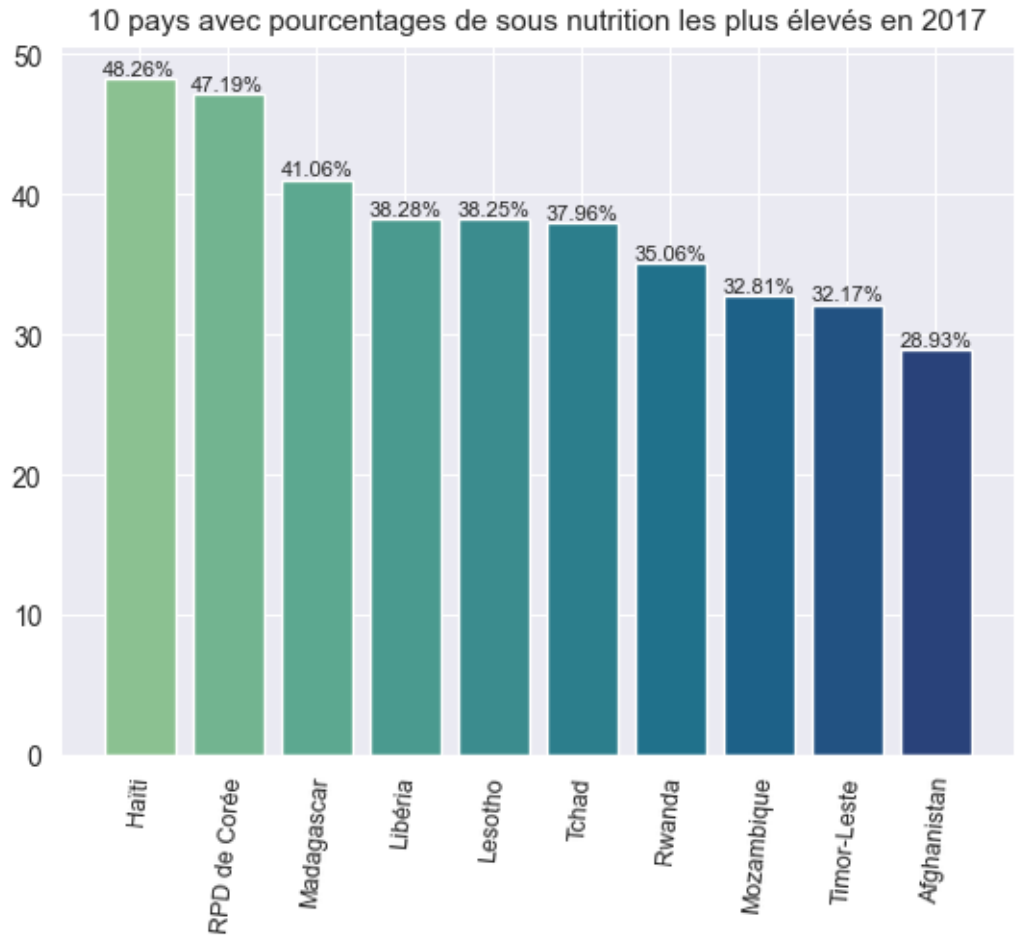
3.6 - Pays avec la proportion de personnes sous-alimentée la plus forte en 2017

[ ]:
```
#affichage après trie des 10 pires pays
pays_documentes= pays_documentes.sort_values('pourcentage_sous_nutrition',␣
 ↪ascending=False)
pays_denutris = pays_documentes.head(10)
```

```
[ ]: pays_denutris = pays_denutris.sort_values('pourcentage_sous_nutrition',␣
      ↪ascending=False).reset_index()
```

```
[ ]: sequential_colors = sns.color_palette("crest", 10)
     pays_denutris.iloc[1,1] = "RPD de Corée"
```

```
[ ]: graph = plt.bar(pays_denutris.Zone, pays_denutris.pourcentage_sous_nutrition,␣
      ↪color=sequential_colors)
     plt.xticks(rotation = 85, fontsize=9)
     plt.title("10 pays avec pourcentages de sous nutrition les plus élevés en␣
      ↪2017", loc = 'center', fontsize = 11)
     for rect in graph:
         height = rect.get_height()
         plt.text(rect.get_x() + rect.get_width() / 2.0, height, f'{height:.2f}%',␣
      ↪ha='center', va='bottom', fontsize=8)
```



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

[ ]:
```
#calcul du total de l'aide alimentaire par pays (entre 2013 et 2016):
aide_alimentaire_csv.head(50)
```

[ ]:
```
#affichage après tri des 10 pays qui ont bénéficié le plus de l'aide alimentaire
top_10_aide = aide_alimentaire_csv.groupby('Zone').agg({'Valeur':'sum'}).
 ↪sort_values(by='Valeur', ascending=False)
```

[ ]:
```
top_10 = top_10_aide.head(10)
```

[ ]:
```
top_10 = top_10.reset_index()
```

[ ]:
```
top_10.head(10)
```

[ ]:

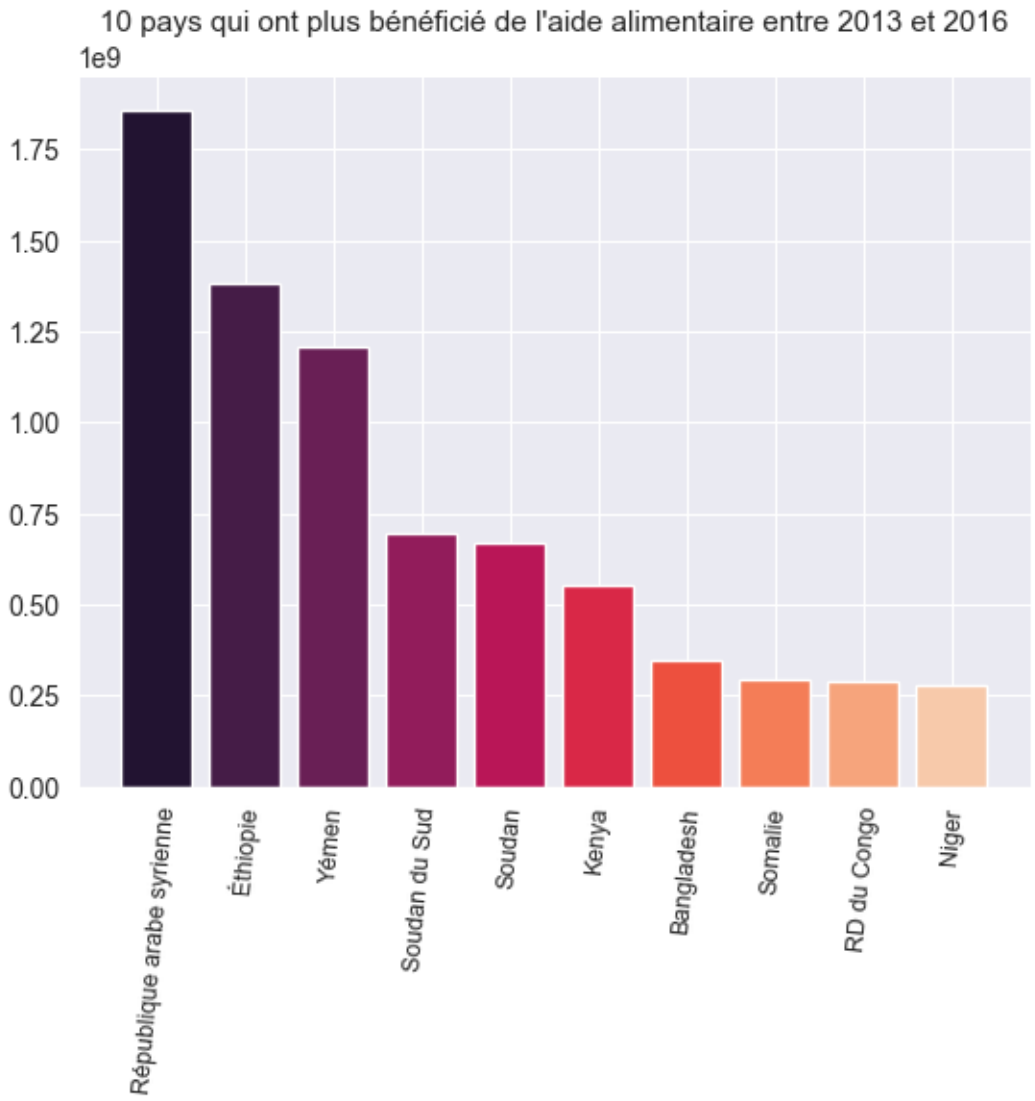
	Zone	Valeur
0	République arabe syrienne	1858943000
1	Éthiopie	1381294000
2	Yémen	1206484000
3	Soudan du Sud	695248000
4	Soudan	669784000
5	Kenya	552836000
6	Bangladesh	348188000
7	Somalie	292678000
8	République démocratique du Congo	288502000
9	Niger	276344000

[ ]:
```
top_10.iloc[8,0]
```

```
République démocratique du Congo
```

[ ]:
```
top_10.iloc[8,0] = "RD du Congo"
sequential = sns.color_palette("rocket", 10)
graph2 = plt.bar(top_10.Zone, top_10.Valeur, color=sequential)
plt.xticks(rotation = 85, fontsize=9)
plt.title("10 pays qui ont plus bénéficié de l'aide alimentaire entre 2013 et␣
 ↪2016", loc='center', fontsize = 11)
```

[ ]:
```
Text(0.5, 1.0, "10 pays qui ont plus bénéficié de l'aide alimentaire entre 2013
et 2016")
```



[ ]:
```
sns.barplot(x = top_10['Valeur'], y = top_10['Zone'], palette= sequential)
plt.title("10 pays qui ont plus bénéficié de l'aide alimentaire entre 2013 et␣
↪2016", loc='center', fontsize = 11)
```

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

[99]:
```
#Création d'un dataframe avec la zone, l'année et l'aide alimentaire puis␣
↪groupby sur zone et année

evolution = aide_alimentaire_csv.groupby(['Zone', 'Année']).agg({'Valeur':
↪'sum'})
```

[100]:
```
evolution_pays = aide_alimentaire_csv.pivot_table(index='Zone',␣
↪columns='Année', values='Valeur', aggfunc='sum')
```

[101]:
```
evolution_pays_3 = aide_alimentaire_csv.pivot_table(index='Année',␣
↪columns='Zone', values='Valeur', aggfunc='sum')
```

[102]:
```
evolution_pays_3 = evolution_pays_3.reset_index()
```

[103]:
```
evolution_pays_3.head()
```

[103]:
```
Zone  Année  Afghanistan     Algérie     Angola   Bangladesh    Bhoutan  \
0      2013  128238000.0  35234000.0  5000000.0  131018000.0  1724000.0
1      2014   57214000.0  18980000.0    14000.0  194628000.0   146000.0
2      2015          NaN  17424000.0        NaN   22542000.0   578000.0
3      2016          NaN   9476000.0        NaN          NaN   218000.0

Zone  Bolivie (État plurinational de)  Burkina Faso     Burundi       Bénin  \
0                                 NaN    18620000.0  53372000.0  17622000.0
1                              6000.0    22938000.0  11010000.0    672000.0
2                                 NaN    23182000.0  12936000.0   3786000.0
3                                 NaN       72000.0         NaN    144000.0

Zone  …       Tchad  Timor-Leste        Togo   Vanuatu        Yémen  \
0     …  93930000.0     116000.0  24804000.0       NaN  264764000.0
1     …  97926000.0          NaN         NaN       NaN  103840000.0
2     …  73678000.0          NaN         NaN  802000.0  372306000.0
3     …   2432000.0          NaN         NaN       NaN  465574000.0

Zone     Zambie    Zimbabwe     Égypte   Équateur     Éthiopie
0      328000.0  21252000.0  1122000.0  1362000.0  591404000.0
1     2698000.0  26600000.0        NaN        NaN  586624000.0
2           NaN  14718000.0        NaN        NaN  203266000.0
3           NaN         NaN        NaN        NaN          NaN

[4 rows x 77 columns]
```

[116]:
```
evolut = evolution_pays_3.loc[:, ['Année', 'République arabe syrienne',␣
↪'Éthiopie', 'Yémen', 'Soudan du Sud', 'Soudan']]
```

[117]:
```
evolut.head()
```

[117]:
```
Zone  Année  République arabe syrienne     Éthiopie        Yémen  \
0      2013                563566000.0  591404000.0  264764000.0
1      2014                651870000.0  586624000.0  103840000.0
2      2015                524949000.0  203266000.0  372306000.0
3      2016                118558000.0          NaN  465574000.0
```

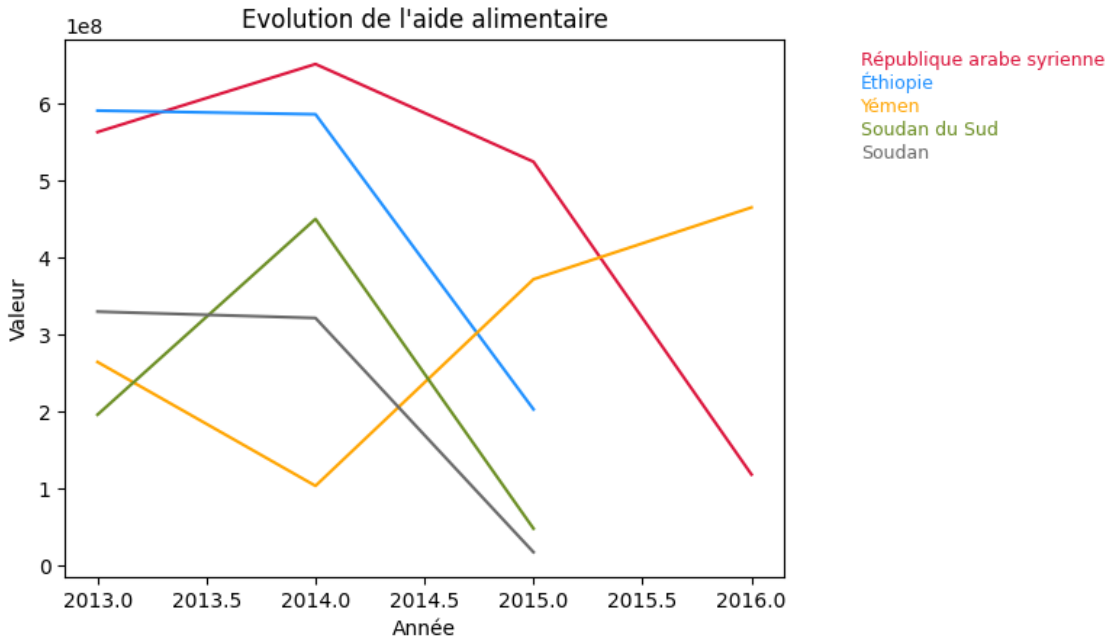
```
Zone  Soudan du Sud       Soudan
0       196330000.0  330230000.0
1       450610000.0  321904000.0
2        48308000.0   17650000.0
3               NaN          NaN
```

[118]:
```
df_republique_arabe_syrienne = evolut.loc[:, ['Année', 'République arabe␣
↪syrienne']]
df_ethiopie = evolut.loc[:, ['Année', 'Éthiopie']]
df_yemen = evolut.loc[:, ['Année', 'Yémen']]
df_soudan_du_sud = evolut.loc[:, ['Année', 'Soudan du Sud']]
df_soudan = evolut.loc[:, ['Année', 'Soudan']]
```

[119]:
```
df_republique_arabe_syrienne.rename(columns={'République arabe syrienne' :␣
↪'Valeur'}, inplace=True)
df_ethiopie.rename(columns={'Éthiopie' : 'Valeur'}, inplace=True)
df_yemen.rename(columns={'Yémen' : 'Valeur'}, inplace=True)
df_soudan_du_sud.rename(columns={'Soudan du Sud' : 'Valeur'}, inplace=True)
df_soudan.rename(columns={'Soudan' : 'Valeur'}, inplace=True)
```

[120]:
```
sns.lineplot(x = df_republique_arabe_syrienne['Année'], y =␣
↪df_republique_arabe_syrienne['Valeur'], color='crimson').text(s='République␣
↪arabe syrienne', x=2016.5, y=650000000.0, color='crimson', fontsize=9)
sns.lineplot(x = df_ethiopie['Année'], y = df_ethiopie['Valeur'],␣
↪color='dodgerblue').text(s='Éthiopie', x=2016.5, y=620000000.0,␣
↪color='dodgerblue', fontsize=9)
sns.lineplot(x = df_yemen['Année'], y = df_yemen['Valeur'], color='orange').
↪text(s='Yémen', x=2016.5, y=590000000.0, color='orange', fontsize=9)
sns.lineplot(x = df_soudan_du_sud['Année'], y = df_soudan_du_sud['Valeur'],␣
↪color = 'olivedrab').text(s='Soudan du Sud', x=2016.5, y=560000000.0,␣
↪color='olivedrab', fontsize=9)
sns.lineplot(x = df_soudan['Année'], y = df_soudan['Valeur'], color =␣
↪'dimgrey').text(s='Soudan', x=2016.5, y=530000000.0, color='dimgrey',␣
↪fontsize=9)
plt.title("Evolution de l'aide alimentaire", loc='center')
```

[120]:
```
Text(0.5, 1.0, "Evolution de l'aide alimentaire")
```



[ ]:
```
evolution_pays.head()
```

[ ]:
```
evolution_pays = evolution_pays.reset_index()
```

[ ]:
```
evolution_pays.head()
```

[ ]:

Année	Zone	2013	2014	2015	2016
0	Afghanistan	128238000.0	57214000.0	NaN	NaN
1	Algérie	35234000.0	18980000.0	17424000.0	9476000.0
2	Angola	5000000.0	14000.0	NaN	NaN
3	Bangladesh	131018000.0	194628000.0	22542000.0	NaN
4	Bhoutan	1724000.0	146000.0	578000.0	218000.0

[ ]:
```
#Création d'une liste contenant les 5 pays qui ont le plus bénéficiées de␣
↪l'aide alimentaire
list_pays = top_10_aide.index.values.tolist()
list_pays_slice = list_pays[0:5]
print(list_pays_slice)
```

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

[ ]:
```
#On filtre sur le dataframe avec notre liste
evolution_5_pays = evolution_pays[evolution_pays['Zone'].isin(list_pays_slice)]
```

[ ]:
```
# Affichage des pays avec l'aide alimentaire par année
evolution_5_pays.head()
```

3.9 - Pays avec le moins de disponibilité par habitant

[ ]:
```
dispo_agg_pop.columns
```

[ ]:
```
Index(['Zone', 'Disponibilité alimentaire (Kcal/personne/jour)',
       'Disponibilité intérieure', 'Année', 'Population', 'Kcal_dispo_pays'],
      dtype='object')
```

[ ]:
```
#Calcul de la disponibilité en kcal par personne par jour par pays
dispo_par_habitant = dispo_agg_pop.drop(columns=["Disponibilité intérieure",␣
↪"Année", "Population"])
```

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

[ ]:
```
moins_dispo.head(10)
```

[ ]:

	index	Zone	Disponibilité alimentaire (Kcal/personne/jour)	Kcal_dispo_pays
0	128	République centrafricaine	1879.0	8.635927e+09
1	166	Zambie	1924.0	3.242632e+10
2	91	Madagascar	2056.0	5.257297e+10
3	0	Afghanistan	2087.0	7.574999e+10
4	65	Haïti	2089.0	2.294216e+10
5	133	République populaire démocratique de Corée	2093.0	5.322462e+10
6	151	Tchad	2109.0	3.167033e+10
7	167	Zimbabwe	2113.0	3.008193e+10
8	114	Ouganda	2126.0	8.752017e+10
9	172	Éthiopie	2129.0	2.265254e+11

[ ]:
```
moins_dispo.iloc[5,1] = "RPD de Corée"
```

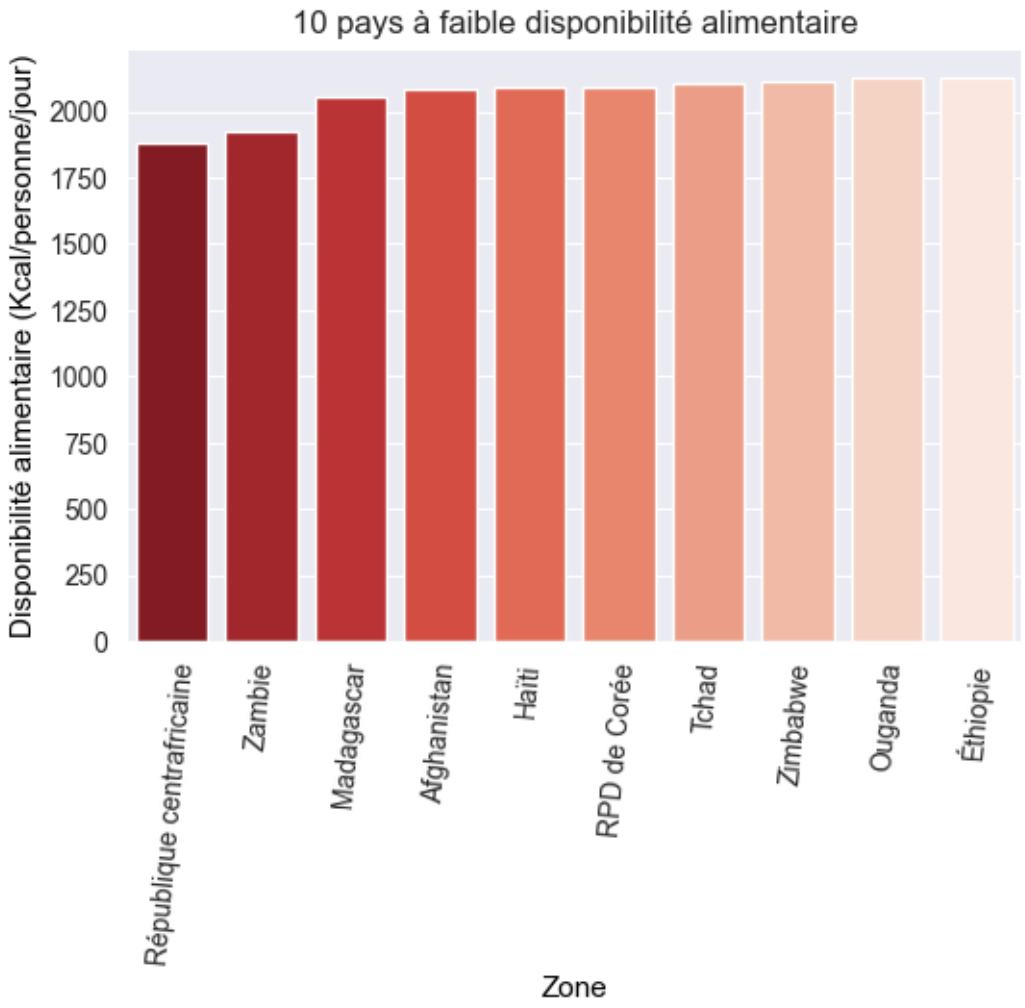
[ ]:
```
plt.figure(figsize=(6,4))
graph3= sns.barplot(x=moins_dispo['Zone'] ,y=moins_dispo['Disponibilité␣
 ↪alimentaire (Kcal/personne/jour)'], palette="Reds_r")
plt.xlabel('Zone', fontsize=11, color='black')
plt.ylabel("Disponibilité alimentaire (Kcal/personne/jour)", fontsize=11,␣
 ↪color='black')
plt.xticks(rotation = 85)
plt.title("10 pays à faible disponibilité alimentaire", loc='center')
```

```
/var/folders/r_/fd0gwkxn6n995_hk_5lc9f540000gn/T/ipykernel_95323/2604262571.py:2
: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  graph3= sns.barplot(x=moins_dispo['Zone'] ,y=moins_dispo['Disponibilité
alimentaire (Kcal/personne/jour)'], palette="Reds_r")
```

[ ]:
```
Text(0.5, 1.0, '10 pays à faible disponibilité alimentaire')
```



3.10 - Pays avec le plus de disponibilité par habitant

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

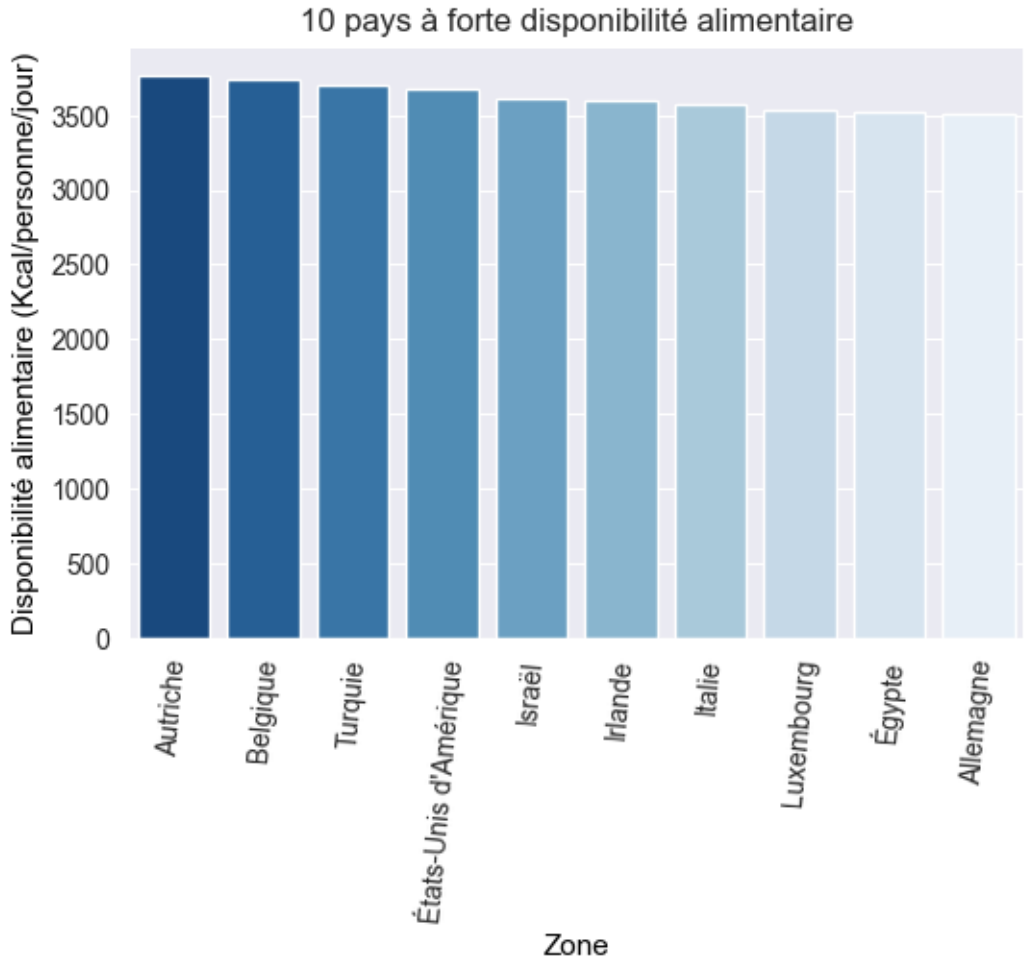
[ ]:
```
plt.figure(figsize=(6,4))
sns.barplot(x=plus_dispo['Zone'] ,y=plus_dispo['Disponibilité alimentaire (Kcal/
↪personne/jour)'], palette="Blues_r")
plt.xlabel('Zone', fontsize=11, color='black')
plt.ylabel("Disponibilité alimentaire (Kcal/personne/jour)", fontsize=11,␣
↪color='black')
plt.xticks(rotation = 85)
plt.title("10 pays à forte disponibilité alimentaire", loc='center')
```

```
/var/folders/r_/fd0gwkxn6n995_hk_5lc9f540000gn/T/ipykernel_95323/3969104232.py:2
: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.barplot(x=plus_dispo['Zone'] ,y=plus_dispo['Disponibilité alimentaire
(Kcal/personne/jour)'], palette="Blues_r")
```

[ ]:
```
Text(0.5, 1.0, '10 pays à forte disponibilité alimentaire')
```



3.11 - Exemple de la Thaïlande pour le Manioc

[ ]:
```
#création d'un dataframe avec uniquement la Thaïlande
pays_documentes.loc[pays_documentes["Zone"] == "Thaïlande"]
```

[ ]:
```
          Zone  sous_nutrition  Annee  Population  pourcentage_sous_nutrition
185  Thaïlande       6200000.0   2017  69209810.0                        8.96
```

[ ]:
```
#Calcul de la sous nutrition en Thaïlande : 8,96%
```

[ ]:
```
# On calcule la proportion exportée en fonction de la proportion
thailande_manioc = dispo_alimentaire_csv.loc[(dispo_alimentaire_csv['Zone'] ==␣
 ↪"Thaïlande") & (dispo_alimentaire_csv['Produit'] == "Manioc")]
```

[ ]:
```
thailande_manioc.head()
```

[ ]:
```
            Zone Produit   Origine  Aliments pour animaux  \
13809  Thaïlande  Manioc  vegetale           1.800000e+09

       Autres Utilisations  Disponibilité alimentaire (Kcal/personne/jour)  \
13809         2.081000e+09                                           40.0

       Disponibilité alimentaire en quantité (kg/personne/an)  \
13809                                               13.0

       Disponibilité de matière grasse en quantité (g/personne/jour)  \
13809                                               0.05

       Disponibilité de protéines en quantité (g/personne/jour)  \
13809                                               0.14

       Disponibilité intérieure  Exportations - Quantité  \
13809              6.264000e+09             2.521400e+10

       Importations - Quantité   Nourriture        Pertes    Production  \
13809             1.250000e+09  871000000.0  1.511000e+09  3.022800e+10

       Semences  Traitement  Variation de stock
13809       0.0         0.0                 0.0
```

[ ]:
```
export_manioc = round(((thailande_manioc['Exportations - Quantité']/
 ↪thailande_manioc['Production'])*100),2)
print(export_manioc)
```

```
13809    83.41
dtype: float64
```

[ ]:
```
nourriture_manioc = round(((thailande_manioc['Nourriture']/
 ↪thailande_manioc['Production'])*100),2)
print(nourriture_manioc)
```

```
13809    2.88
```

```
dtype: float64
```

Etape 6 - Analyse complémentaires