

PROJET 4 DATA ANALYST

Réalisez une étude de santé publique avec R ou Python

PARTIE ANALYSE DES DONNEES

```
In [96]: import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
```

```
In [97]: sous_nutrition= pd.read_csv('sous_nutrition_clean.csv')
population= pd.read_csv('population_clean.csv')
dispo_alimentaire= pd.read_csv('dispo_alimentaire_clean.csv')
aide_alimentaire= pd.read_csv('aide_alimentaire_clean.csv')
```

1 - Proportion de personnes en sous nutrition

```
In [102]: # Il faut tout d'abord faire une jointure entre la table population et la table sous nutrition
popu_sous_nutri = pd.merge(population, sous_nutrition, on=['Zone', 'Année'], how='left')
popu_sous_nutri.head()
```

```
Out[102]:
```

	Zone	Année	Population	Sous_nutrition
0	Afghanistan	2013	32269589.0	8600000.0
1	Afghanistan	2014	33370794.0	8800000.0
2	Afghanistan	2015	34413603.0	8900000.0
3	Afghanistan	2016	35383032.0	9700000.0
4	Afghanistan	2017	36296113.0	10500000.0

```
In [38]: popu_sous_nutri.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 1176 entries, 0 to 1175
Data columns (total 4 columns):
#   Column          Non-Null Count  Dtype
---  -
0   Zone             1176 non-null   object
1   Année            1176 non-null   int64
2   Population       1176 non-null   float64
3   Sous_nutrition   1080 non-null   float64
```

dtypes: float64(2), int64(1), object(1)
memory usage: 45.9+ KB

```
In [39]: #Calcul et affichage de la population total et du nombre de personnes en état de sous
population_mondiale=round(popu_sous_nutri[popu_sous_nutri.Année==2017].Population.sum()
popusousNutrition_2017=round(popu_sous_nutri[popu_sous_nutri.Année==2017].Sous_nutriti
print(f"la population mondial : {population_mondiale} personnes")
print(f"la population en sous nutrition 2017 : {popusousNutrition_2017} personnes")

la population mondial : 7510681110 personnes
la population en sous nutrition 2017 : 529100000 personnes
```

```
In [40]: proportion=round((popusousNutrition_2017/population_mondiale)*100,2)
print(f"la proportion de personnes en sous nutrition dans le monde en 2017 est de {pro

la proportion de personnes en sous nutrition dans le monde en 2017 est de 7.04 %
```

2 - Nombre théorique de personne qui pourrait être nourries 2017

</div>

```
In [41]: # un être humain prend en moyenne 2400 kcal/jour (Source => google)
print(f"un être humain consomme en moyenne ")

un être humain consomme en moyenne
```

```
In [42]: #une jointure entre le data frame population et Dispo_alimentaire afin d'ajouter dan
data_theorique = pd.merge(dispo_alimentaire,population[population.Année==2017],on='Z
```

```
In [43]: data_theorique.head()
```

```
Out[43]:
```

	Zone	Produit	Origine	Aliments pour animaux	Autres Utilisations	Disponibilité alimentaire (Kcal/personne/jour)	Disponibilité alimentaire en quantité (kg/personne/an)	Dispon matière c
0	Afghanistan	Abats Comestible	animale	0.0	0.0	5.0	1.72	
1	Afghanistan	Agrumes, Autres	vegetale	0.0	0.0	1.0	1.29	
2	Afghanistan	Aliments pour enfants	vegetale	0.0	0.0	1.0	0.06	
3	Afghanistan	Ananas	vegetale	0.0	0.0	0.0	0.00	
4	Afghanistan	Bananes	vegetale	0.0	0.0	4.0	2.70	

```
In [44]: data_theorique.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 14442 entries, 0 to 14441
Data columns (total 20 columns):
#   Column                                     Non-Null Count
Dtype
---  ---
0    Zone                                     14442 non-null
object
1    Produit                                 14442 non-null
```

```

object
  2   Origine                                     14442 non-null
object
  3   Aliments pour animaux                     14442 non-null
float64
  4   Autres Utilisations                       14442 non-null
float64
  5   Disponibilité alimentaire (Kcal/personne/jour) 14442 non-null
float64
  6   Disponibilité alimentaire en quantité (kg/personne/an) 14442 non-null
float64
  7   Disponibilité de matière grasse en quantité (g/personne/jour) 14442 non-null
float64
  8   Disponibilité de protéines en quantité (g/personne/jour) 14442 non-null
float64
  9   Disponibilité intérieure                  14442 non-null
float64
 10   Exportations - Quantité                  14442 non-null
float64
 11   Importations - Quantité                  14442 non-null
float64
 12   Nourriture                              14442 non-null
float64
 13   Pertes                                 14442 non-null
float64
 14   Production                             14442 non-null
float64
 15   Semences                              14442 non-null
float64
 16   Traitement                            14442 non-null
float64
 17   Variation de stock                     14442 non-null
float64
 18   Année                                 14189 non-null
float64
 19   Population                            14189 non-null
float64
dtypes: float64(17), object(3)
memory usage: 2.3+ MB

```

```

In [45]: #Création de la colonne dispo_kcal avec calcul des kcal disponibles mondialement
data_theorique['dispo_kcal']=data_theorique['Disponibilité alimentaire (Kcal/personne/jour)']
data_theorique.head()

```

```

Out[45]:

```

	Zone	Produit	Origine	Aliments pour animaux	Autres Utilisations	Disponibilité alimentaire (Kcal/personne/jour)	Disponibilité alimentaire en quantité (kg/personne/an)	Dispon matière g
0	Afghanistan	Abats Comestible	animale	0.0	0.0	5.0	1.72	
1	Afghanistan	Agrumes, Autres	vegetale	0.0	0.0	1.0	1.29	
2	Afghanistan	Aliments pour enfants	vegetale	0.0	0.0	1.0	0.06	
3	Afghanistan	Ananas	vegetale	0.0	0.0	0.0	0.00	
4	Afghanistan	Bananes	vegetale	0.0	0.0	4.0	2.70	

5 rows × 21 columns

```
In [46]: #Calcul du nombre d'humains pouvant être nourris
nbre_theorique=round(data_theorique['dispo_kcal'].sum()/2400)
print(f"Le nombre théorique de personnes qui pourraient être nourries en 2017 est {n
```

Le nombre théorique de personnes qui pourraient être nourries en 2017 est 8781586896 personnes

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

</div>

```
In [103]: #Calcul du nombre d'humains pouvant être nourris avec les végétaux
nbre_theorique=round(data_theorique[data_theorique['Origine'] == 'vegetale']['dispo
print(f"Le nombre théorique de personnes qui pourraient être nourries avec les prod
```

Le nombre théorique de personnes qui pourraient être nourries avec les produits végétaux en 2017 est 7237749736 personnes

4 - Répartition de la disponibilité intérieure

</div>

```
In [48]: dispo_alimentaire.head()
```

Out[48]:

	Zone	Produit	Origine	Aliments pour animaux	Autres Utilisations	Disponibilité alimentaire (Kcal/personne/jour)	Disponibilité alimentaire en quantité (kg/personne/an)	Dispo
0	Afghanistan	Abats Comestible	animale	0.0	0.0	5.0	1.72	
1	Afghanistan	Agrumes, Autres	vegetale	0.0	0.0	1.0	1.29	
2	Afghanistan	Aliments pour enfants	vegetale	0.0	0.0	1.0	0.06	
3	Afghanistan	Ananas	vegetale	0.0	0.0	0.0	0.00	
4	Afghanistan	Bananes	vegetale	0.0	0.0	4.0	2.70	

```
In [49]: #Calcul de la disponibilité totale
disponibilite= dispo_alimentaire['Disponibilité intérieure'].sum()
```

```
In [50]: #création d'une boucle for pour afficher les différentes valeurs en fonction des c
nourriture = dispo_alimentaire['Nourriture'].sum()
pertes = dispo_alimentaire['Pertes'].sum()
aliments_animaux = dispo_alimentaire['Aliments pour animaux'].sum()
autre = disponibilite - (nourriture + pertes + aliments_animaux)

valeurs_a_analyser = [aliments_animaux, pertes, nourriture, autre]

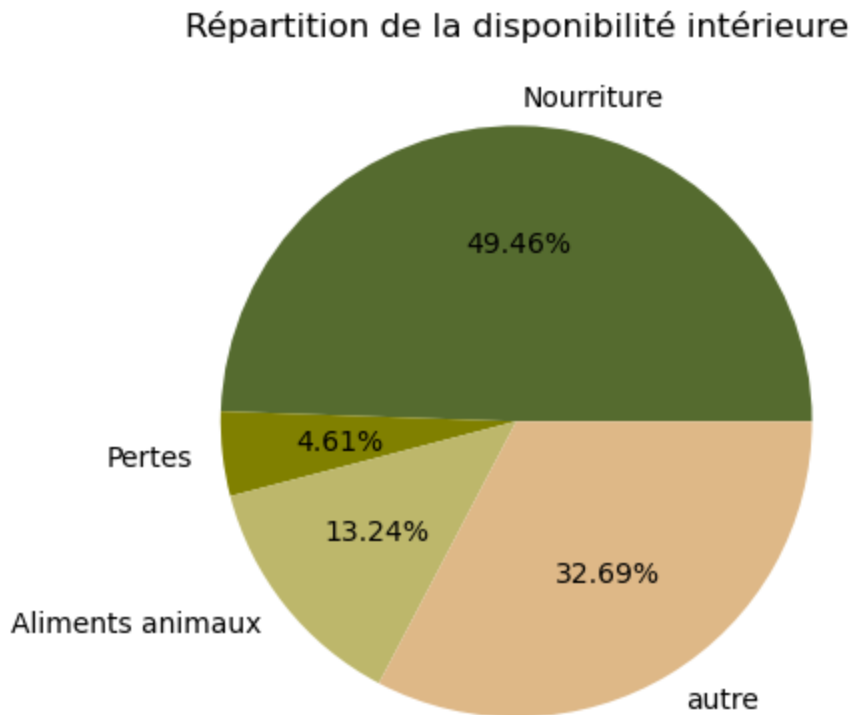
for valeur in valeurs_a_analyser:
```

```
pourcentage = round((valeur / disponibilite) * 100, 2)
print(f"{valeur}: {pourcentage} %")
```

```
1298786000.0: 13.24 %
452832000.0: 4.61 %
4853202000.0: 49.46 %
3208222000.0: 32.69 %
```

```
In [51]: labels = 'Nourriture', 'Pertes', 'Aliments animaux', 'autre'
         sizes=[nourriture, pertes, aliments_animaux, autre]

         fig, ax = plt.subplots()
         ax.pie(sizes, labels=labels, autopct='%1.2f%%', colors=['darkolivegreen', 'olive',
         ax.set_title("Répartition de la disponibilité intérieure")
         plt.show()
```



5 - Part de l'utilisation des principales céréales entre l'alimentation humaine et animale

</div>

```
In [52]: #Création d'une liste avec toutes les variables
         liste_cereal = ['Blé', 'Avoine', 'Céréales, Autres', 'Maïs', 'Orge', 'Riz (Eq Blé)']

In [53]: #Création d'un dataframe avec les informations uniquement pour ces céréales
         dispo_cereal = dispo_alimentaire[dispo_alimentaire['Produit'].isin(liste_cereal)]
         dispo_cereal.head()
```

```
Out[53]:
```

	Zone	Produit	Origine	Aliments pour animaux	Autres Utilisations	Disponibilité alimentaire (Kcal/personne/jour)	Disponibilité alimentaire en quantité (kg/personne/an)	Dispo alimentaire (g/personne/jour)
7	Afghanistan	Blé	vegetale	0.0	0.0	1369.0	160.23	

12	Afghanistan	Céréales, Autres	vegetale	0.0	0.0	0.0	0.00
32	Afghanistan	Maïs	vegetale	200000.0	0.0	21.0	2.50
34	Afghanistan	Millet	vegetale	0.0	0.0	3.0	0.40
40	Afghanistan	Orge	vegetale	360000.0	0.0	26.0	2.92

```
In [54]: #Affichage de la proportion d'alimentation animale
part_animale =round((dispo_cereal['Aliments pour animaux'].sum())*100/dispo_cereal['Disponib
print(f"la part_animale {part_animale} %")
```

la part_animale 36.25 %

```
In [55]: #Affichage de la proportion d'alimentation humain
part_humaine =round((dispo_cereal['Nourriture'].sum())*100/dispo_cereal['Disponib
print(f"la part_humaine {part_humaine} %")
```

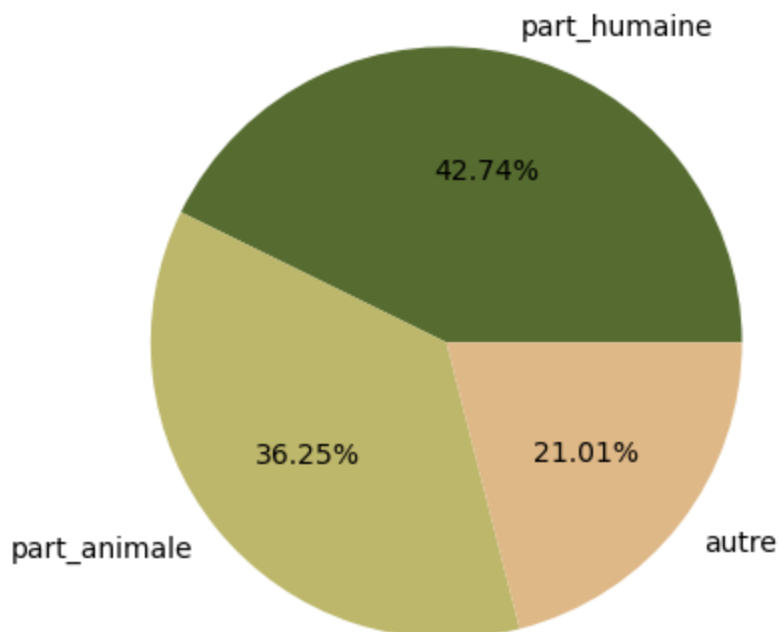
la part_humaine 42.74 %

```
In [56]: autre=100-(part_animale + part_humaine)
```

```
In [57]: labels = 'part_humaine', 'part_animale','autre'
sizes = [part_humaine,part_animale,autre]
colors=['darkolivegreen', 'darkkhaki','burlywood']

fig, ax = plt.subplots()
ax.pie(sizes,labels=labels,autopct='%1.2f%%',colors=colors)
plt.title("Répartition céréales pour humains et animaux")
plt.show()
```

Répartition céréales pour humains et animaux



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

</div>

```
In [58]: popu_sous_nutri.head()
```

```
Out[58]:
```

	Zone	Année	Population	Sous_nutrition
0	Afghanistan	2013	32269589.0	8600000.0
1	Afghanistan	2014	33370794.0	8800000.0
2	Afghanistan	2015	34413603.0	8900000.0
3	Afghanistan	2016	35383032.0	9700000.0
4	Afghanistan	2017	36296113.0	10500000.0

```
In [59]: #pour l'année 2017 on agrège les lignes suivant la colonne Zone pour faire la
sous_nutrition_pays=popu_sous_nutri[popu_sous_nutri['Année'] == 2017].groupby(['
sous_nutrition_pays.head()
```

```
Out[59]:
```

Zone	
Afghanistan	10500000.0
Afrique du Sud	3100000.0
Albanie	100000.0
Algérie	1300000.0
Allemagne	0.0

Name: Sous_nutrition, dtype: float64

```
In [60]: #pour l'année 2017 on agrège les lignes suivant la colonne Zone pour faire la
Population_pays=popu_sous_nutri[popu_sous_nutri['Année'] == 2017].groupby(['Zon
Population_pays.head()
```

```
Out[60]:
```

Zone	
Afghanistan	36296113.0
Afrique du Sud	57009756.0
Albanie	2884169.0
Algérie	41389189.0
Allemagne	82658409.0

Name: Population, dtype: float64

```
In [61]: #oncalcul proportion de sous nutrition par pays
proportion=round(sous_nutrition_pays*100/Population_pays,2)
proportion.head()
```

```
Out[61]:
```

Zone	
Afghanistan	28.93
Afrique du Sud	5.44
Albanie	3.47
Algérie	3.14
Allemagne	0.00

dtype: float64

```
In [62]: #affichage après trie des 10 pires pays
proportion.sort_values(ascending=False,inplace=True)
proportion.head(10)
```

```
Out[62]:
```

Zone	
République populaire démocratique de Corée	47.19
Madagascar	41.06
Libéria	38.28
Lesotho	38.25
Tchad	37.96
Rwanda	35.06
Mozambique	32.81
Afghanistan	28.93

```
Congo 27.39
Venezuela (République bolivarienne du) 27.21
dtype: float64
```

```
In [63]: #selection des noms des pays a mettre sur le code de graphique à bar
proportion.index[:10]
```

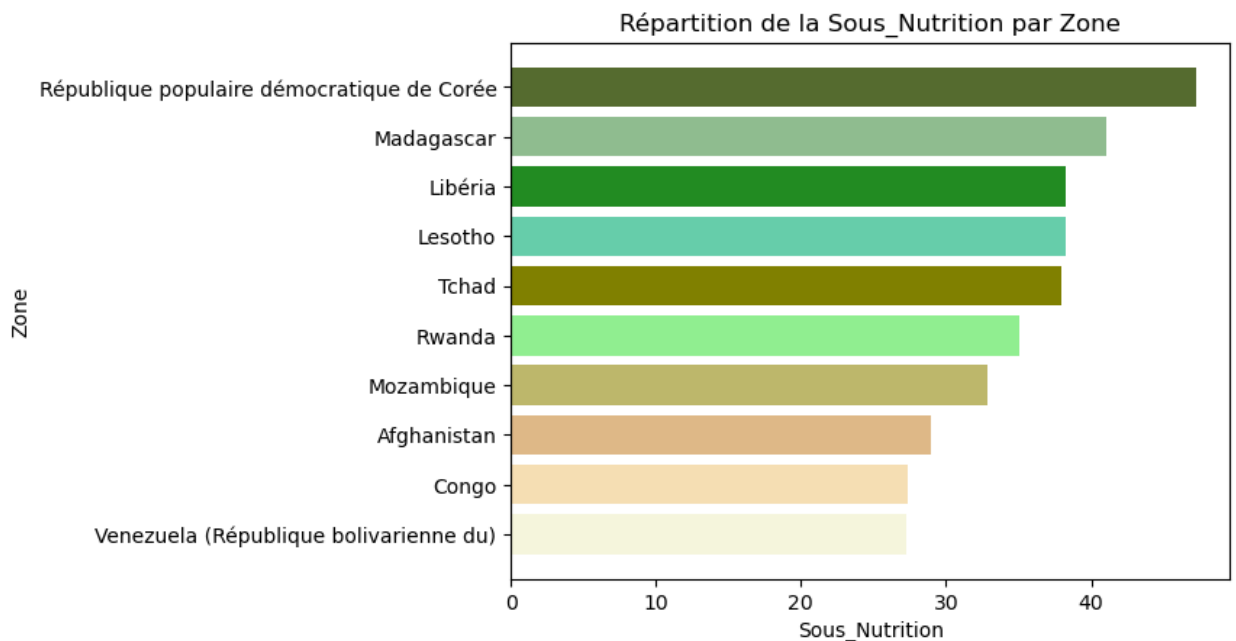
```
Out[63]: Index(['République populaire démocratique de Corée', 'Madagascar', 'Libéria',
        'Lesotho', 'Tchad', 'Rwanda', 'Mozambique', 'Afghanistan', 'Congo',
        'Venezuela (République bolivarienne du)'],
        dtype='object', name='Zone')
```

```
In [64]: #selection des valeurs a mettre sur le code de graphique à bar
proportion[:10]
```

```
Out[64]: Zone
République populaire démocratique de Corée 47.19
Madagascar 41.06
Libéria 38.28
Lesotho 38.25
Tchad 37.96
Rwanda 35.06
Mozambique 32.81
Afghanistan 28.93
Congo 27.39
Venezuela (République bolivarienne du) 27.21
dtype: float64
```

```
In [65]: fig, ax = plt.subplots()
colors=['darkolivegreen','darkseagreen','forestgreen','mediumaquamarine','olive']
zones =proportion.index[:10]
Sous_nutrition =proportion[:10]

ax.barh(zones, Sous_nutrition, color=colors)
plt.gca().invert_yaxis()
ax.set_xlabel('Sous_Nutrition')
ax.set_ylabel('Zone')
ax.set_title('Répartition de la Sous_Nutrition par Zone');
```



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

</div>

```
In [66]: aide_alimentaire.head()
```

```
Out[66]:
```

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

```
In [67]: aide_alimentaire.Année.unique()
```

```
Out[67]: array([2013, 2014, 2015, 2016], dtype=int64)
```

```
In [70]: #calcul du total de l'aide alimentaire par pays  
Top_aide=aide_alimentaire.groupby(['Zone'])['Aide_alimentaire'].sum()  
Top_aide.head()
```

```
Out[70]:
```

Zone	
Afghanistan	185452000
Algérie	81114000
Angola	5014000
Bangladesh	348188000
Bhoutan	2666000

Name: Aide_alimentaire, dtype: int64

```
In [71]: #affichage après trie des 10 pays qui ont bénéficié le plus de l'aide alimenta  
Top_aide=Top_aide.sort_values(ascending=False).head(10)  
Top_aide
```

```
Out[71]:
```

Zone	
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

Name: Aide_alimentaire, dtype: int64

```
In [72]: Top_aide.index[:10]
```

```
Out[72]:
```

Index(['République arabe syrienne', 'Éthiopie', 'Yémen', 'Soudan du Sud', 'Soudan', 'Kenya', 'Bangladesh', 'Somalie', 'République démocratique du Congo', 'Niger'], dtype='object', name='Zone')	

```
In [73]: Top_aide[:10]
```

```
Out[73]:
```

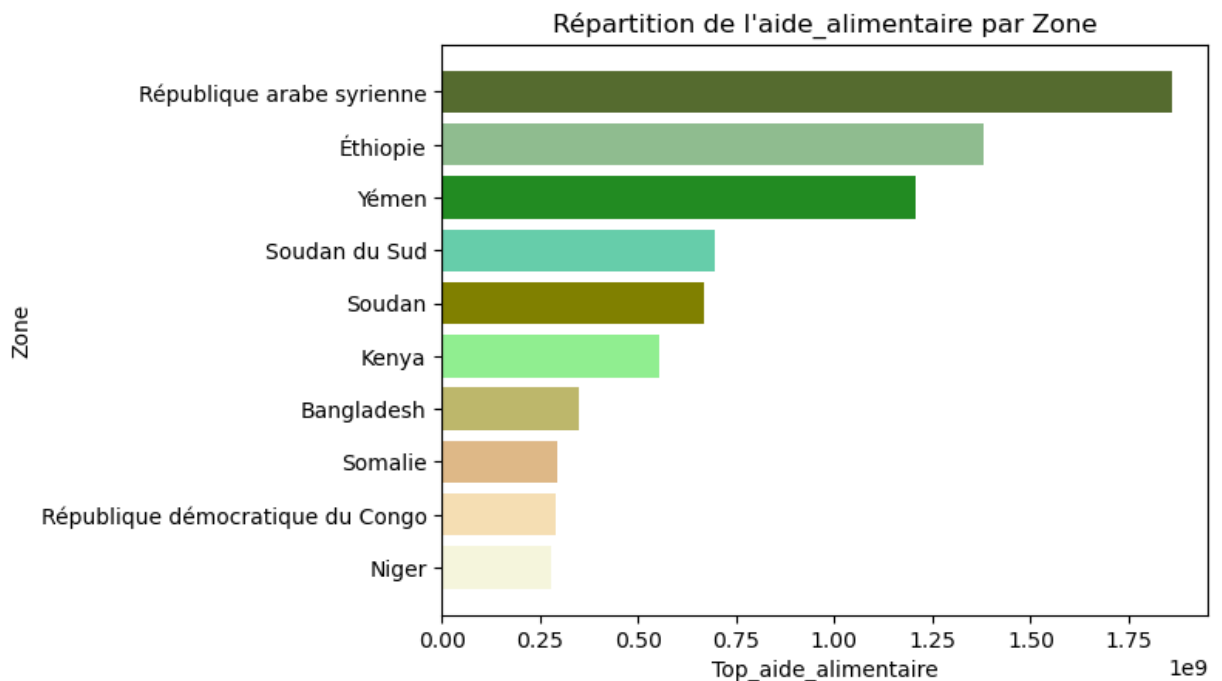
Zone	
République arabe syrienne	1858943000
Éthiopie	1381294000
Yémen	1206484000
Soudan du Sud	695248000
Soudan	669784000

Kenya	552836000
Bangladesh	348188000
Somalie	292678000
République démocratique du Congo	288502000
Niger	276344000

Name: Aide_alimentaire, dtype: int64

```
In [74]: fig, ax = plt.subplots()
zones = Top_aide.index[:10]
Top_aide = Top_aide[:10]
colors=['darkolivegreen','darkseagreen','forestgreen','mediumaquamarine','olivegreen','beige']

ax.barh(zones,Top_aide,color=colors,)
plt.gca().invert_yaxis()
ax.set_xlabel('Top_aide_alimentaire')
ax.set_ylabel('Zone')
ax.set_title("Répartition de l'aide_alimentaire par Zone");
```



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

</div>

```
In [75]: #Création d'un dataframe avec la zone et somme d'aide alimentaire en agrégat
df_result = aide_alimentaire.groupby(['Zone'])['Aide_alimentaire'].sum()
df_result.sort_values(ascending=False,inplace=True)
df_result.head()
```

```
Out[75]: Zone
République arabe syrienne    1858943000
Éthiopie                    1381294000
Yémen                      1206484000
Soudan du Sud               695248000
Soudan                      669784000
Name: Aide_alimentaire, dtype: int64
```

```
In [76]: df_result.index[:5]
```

```
Out[76]: Index(['République arabe syrienne', 'Éthiopie', 'Yémen', 'Soudan du Sud',  
         'Soudan'],  
        dtype='object', name='Zone')
```

```
In [77]: #creation dataframe avec les zones predefinie par df_result.index[:5]  
df=aide_alimentaire[aide_alimentaire.Zone.isin(df_result.index[:5])]  
df.head()
```

```
Out[77]:
```

	Zone	Année	Produit	Aide_alimentaire
354	Éthiopie	2013	Autres non-céréales	170000
355	Éthiopie	2014	Autres non-céréales	466000
356	Éthiopie	2015	Autres non-céréales	244000
357	Éthiopie	2013	Blé et Farin	181066000
358	Éthiopie	2014	Blé et Farin	178646000

```
In [78]: #aggregation de dfsuivant Zone,Année et la somme d'aide alimentaire pour avoir  
top_pays=df.groupby(['Zone', 'Année']).agg(Aide_alimentaire=('Aide_alimentaire', 'sum'))  
top_pays
```

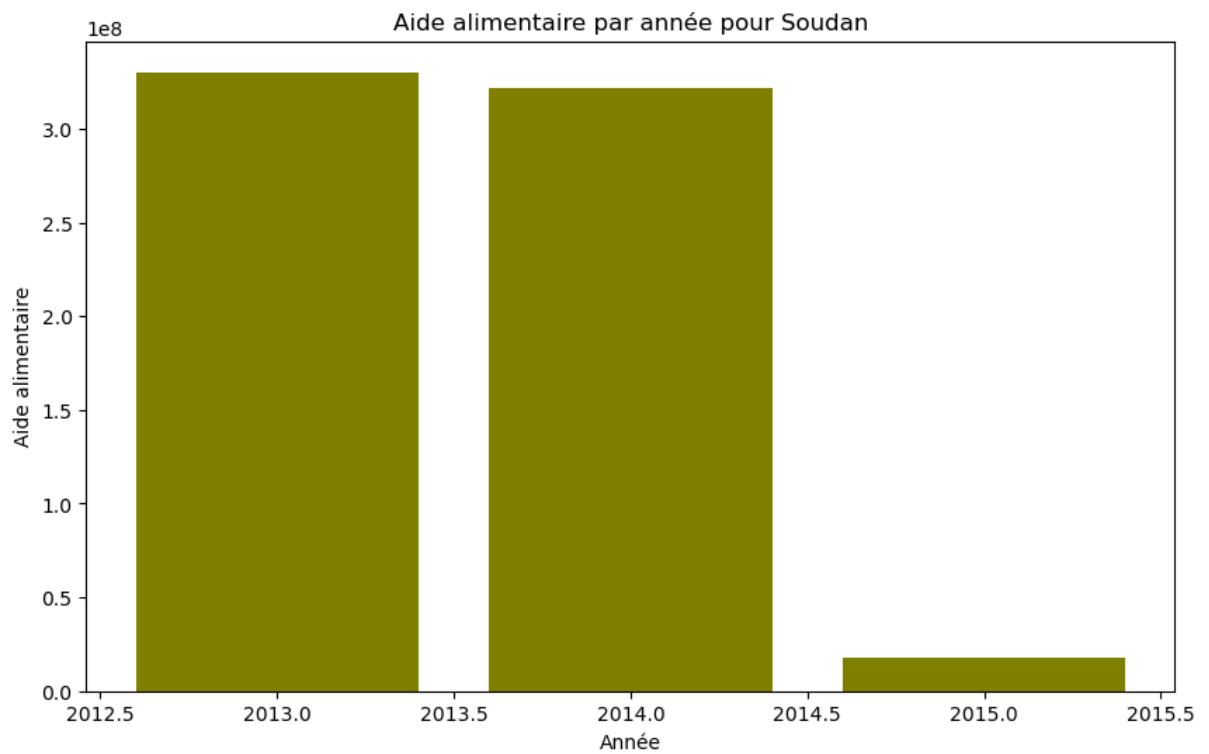
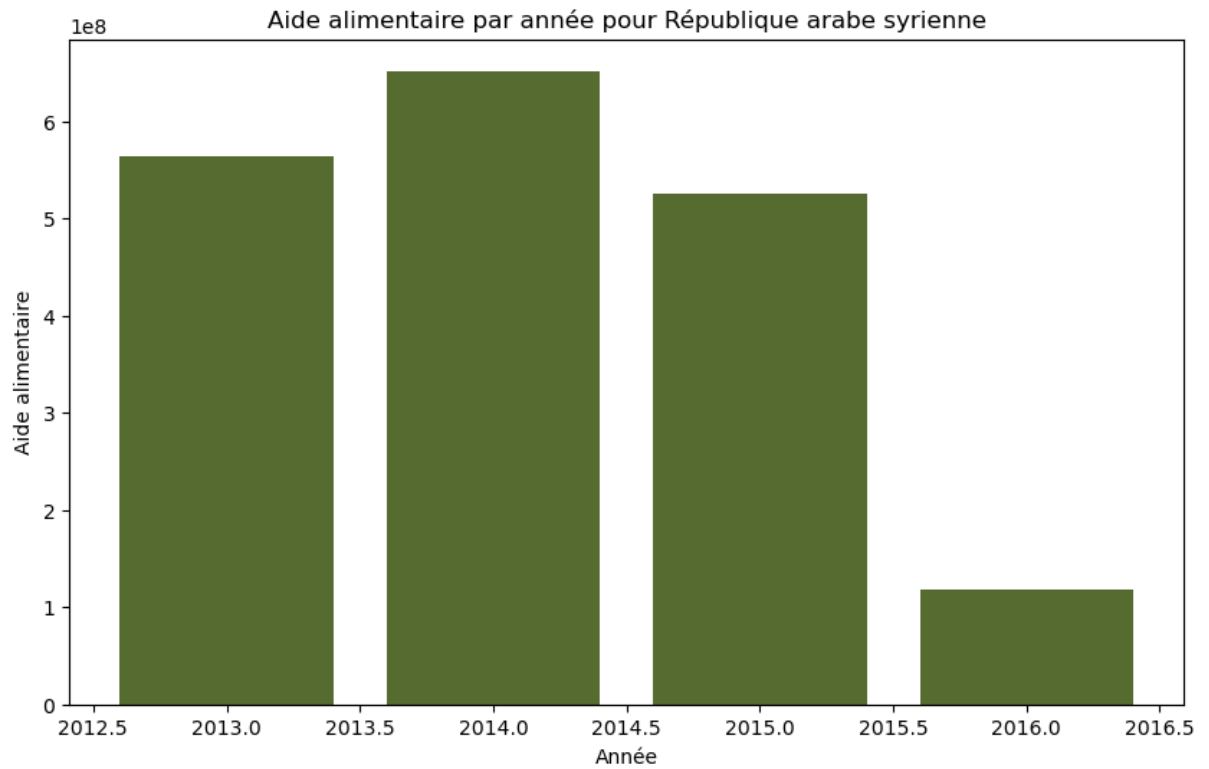
```
Out[78]:
```

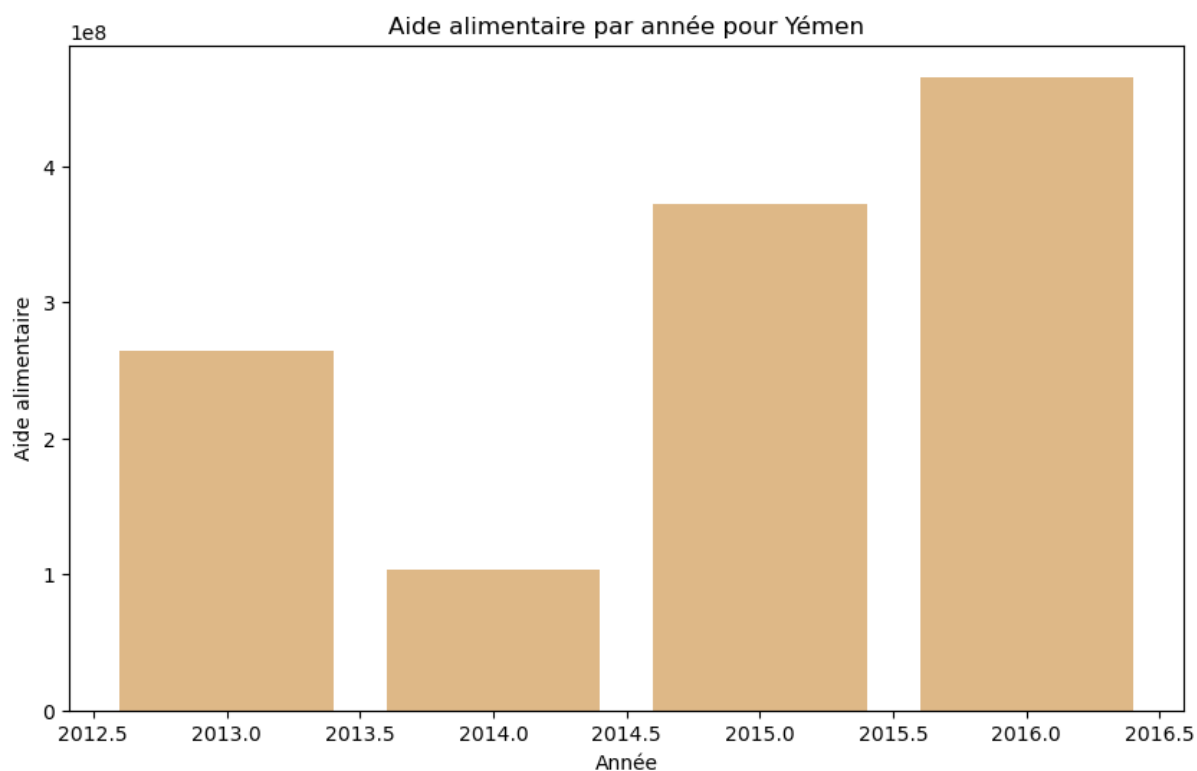
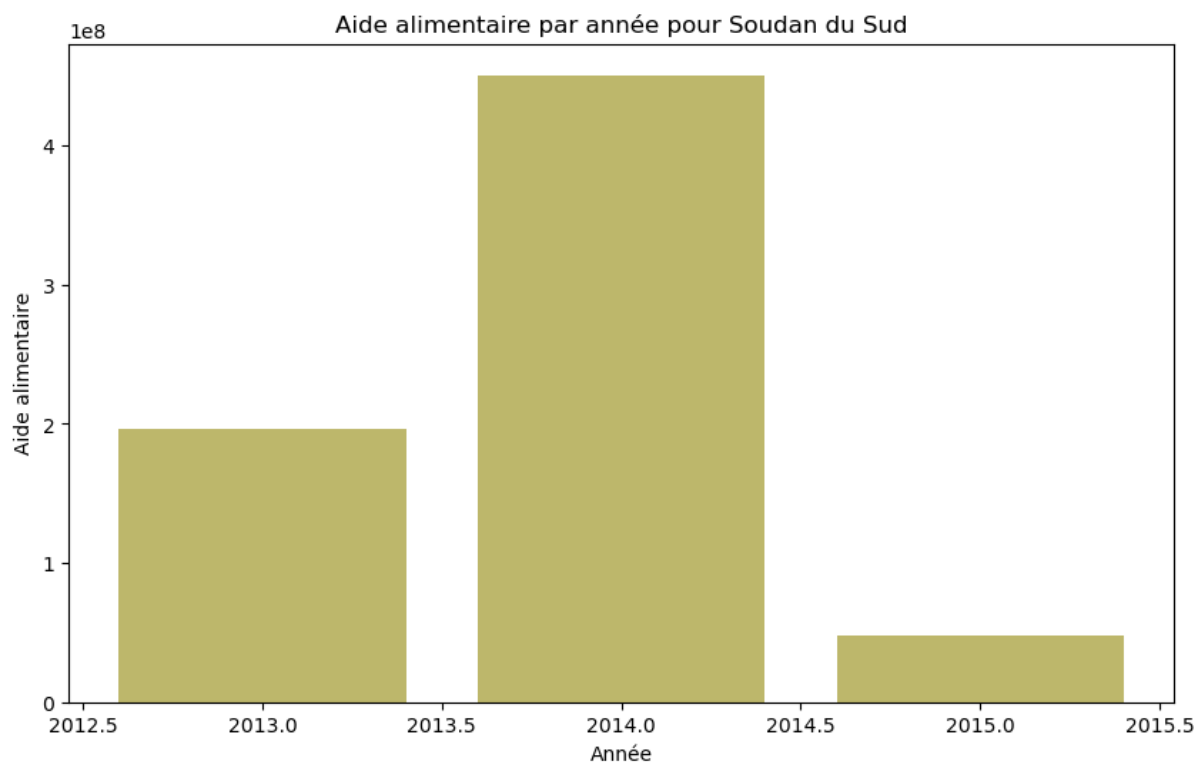
		Aide_alimentaire
	Zone	Année
République arabe syrienne	2013	563566000
	2014	651870000
	2015	524949000
	2016	118558000
Soudan	2013	330230000
	2014	321904000
	2015	17650000
Soudan du Sud	2013	196330000
	2014	450610000
	2015	48308000
Yémen	2013	264764000
	2014	103840000
	2015	372306000
	2016	465574000
Éthiopie	2013	591404000
	2014	586624000
	2015	203266000

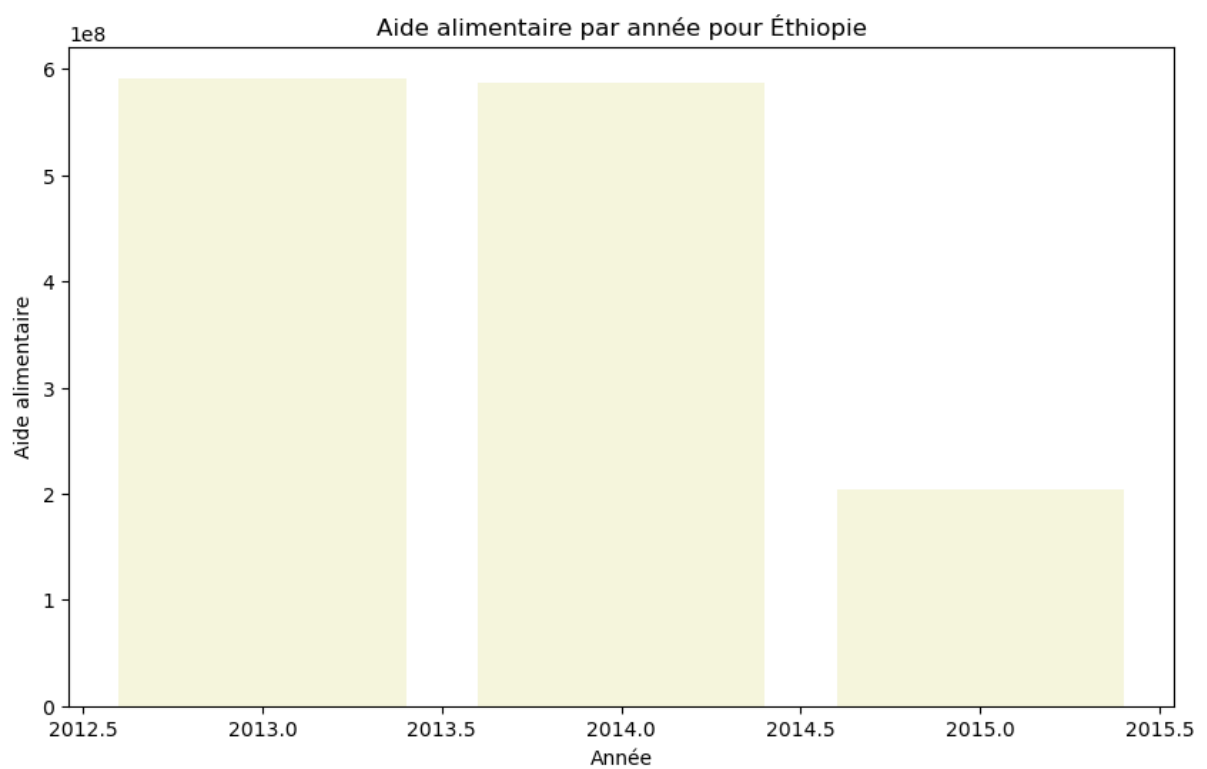
```
In [79]: top_pays.reset_index(inplace=True)
```

```
In [80]: zone=top_pays.Zone.unique()  
colors=['darkolivegreen', 'olive', 'darkkhaki', 'burlywood', 'beige']  
for i, zone in enumerate(zone):  
    zone_data = top_pays[top_pays['Zone'] == zone ]
```

```
plt.figure(figsize=(10, 6))
plt.bar(zone_data['Année'], zone_data['Aide_alimentaire'],color=colors[i])
plt.xlabel('Année')
plt.ylabel('Aide alimentaire')
plt.title(f'Aide alimentaire par année pour {zone}')
plt.show()
```







9 - 10 Pays avec le moins de disponibilité par habitant

</div>

```
In [81]: dispo_alimentaire.head()
```

```
Out[81]:
```

	Zone	Produit	Origine	Aliments pour animaux	Autres Utilisations	Disponibilité alimentaire (Kcal/personne/jour)	Disponibilité alimentaire en quantité (kg/personne/an)
0	Afghanistan	Abats Comestible	animale	0.0	0.0	5.0	1.7%
1	Afghanistan	Agrumes, Autres	vegetale	0.0	0.0	1.0	1.2%
2	Afghanistan	Aliments pour enfants	vegetale	0.0	0.0	1.0	0.0%
3	Afghanistan	Ananas	vegetale	0.0	0.0	0.0	0.0%
4	Afghanistan	Bananes	vegetale	0.0	0.0	4.0	2.7%

```
In [82]: #Calcul de la disponibilité en kcal par personne par jour par pays
data_parHabitant = dispo_alimentaire.groupby('Zone')['Disponibilité alimentaire'].head()
```

```
Out[82]:
```

Zone	
Afghanistan	2087.0
Afrique du Sud	3020.0
Albanie	3188.0
Algérie	3293.0

Allemagne 3503.0
Name: Disponibilité alimentaire (Kcal/personne/jour), dtype: float64

```
In [83]: #Affichage des 10 pays qui ont le moins de dispo alimentaire par personne
data_parHabitant_trie = data_parHabitant.sort_values(ascending=True).head(10)
data_parHabitant_trie
```

```
Out[83]: Zone
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
Éthiopie                      2129.0
Name: Disponibilité alimentaire (Kcal/personne/jour), dtype: float64
```

```
In [84]: data_parHabitant_trie.index[-10:]
```

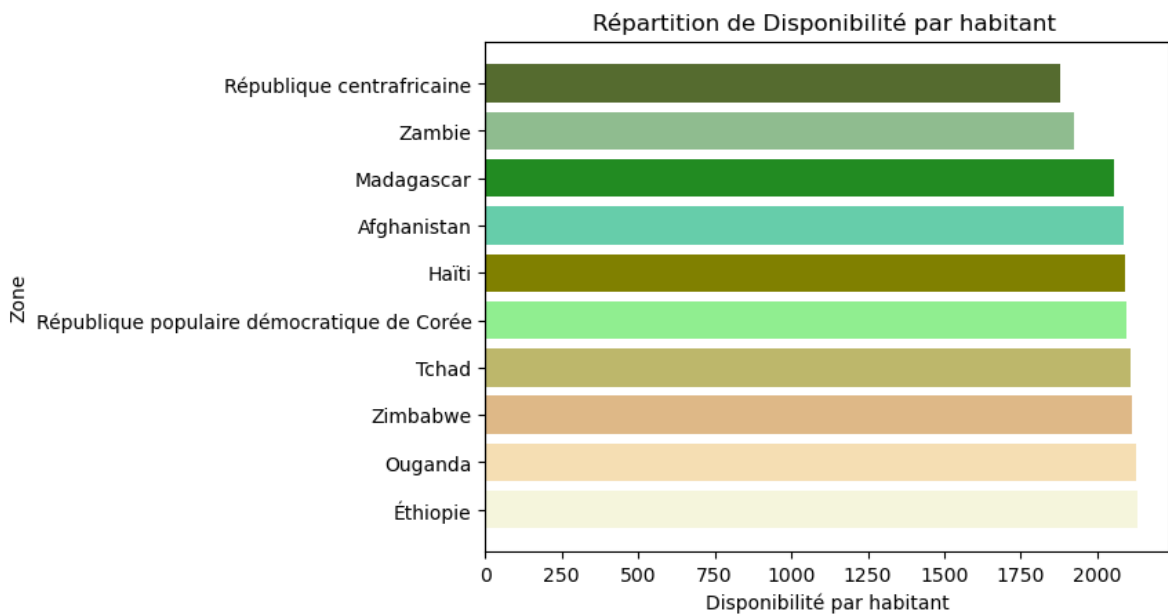
```
Out[84]: Index(['République centrafricaine', 'Zambie', 'Madagascar', 'Afghanistan',
               'Haïti', 'République populaire démocratique de Corée', 'Tchad',
               'Zimbabwe', 'Ouganda', 'Éthiopie'],
              dtype='object', name='Zone')
```

```
In [85]: data_parHabitant_trie[-10:]
```

```
Out[85]: Zone
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
Éthiopie                      2129.0
Name: Disponibilité alimentaire (Kcal/personne/jour), dtype: float64
```

```
In [86]: fig, ax = plt.subplots()
colors=['darkolivegreen','darkseagreen','forestgreen','mediumaquamarine','olivedrabgreen']
zones = data_parHabitant_trie.index[-10:]
Top_dispo_alimentaire = data_parHabitant_trie[-10:]

ax.barh(zones,Top_dispo_alimentaire,color=colors)
plt.gca().invert_yaxis()
ax.set_xlabel('Disponibilité par habitant')
ax.set_ylabel('Zone')
ax.set_title("Répartition de Disponibilité par habitant ");
```



10 - Pays avec le plus de disponibilité par habitant

</div>

```
In [87]: #Affichage des 10 pays qui ont le plus de dispo alimentaire par personne
data_parHabitant = dispo_alimentaire.groupby('Zone')['Disponibilité alimen
data_parHabitant_trie = data_parHabitant.sort_values(ascending=False).head
data_parHabitant_trie
```

```
Out[87]: Zone
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
Name: Disponibilité alimentaire (Kcal/personne/jour), dtype: float64
```

```
In [88]: data_parHabitant_trie.index[:10]
```

```
Out[88]: Index(['Autriche', 'Belgique', 'Turquie', 'États-Unis d'Amérique', 'Israël',
                'Irlande', 'Italie', 'Luxembourg', 'Égypte', 'Allemagne'],
              dtype='object', name='Zone')
```

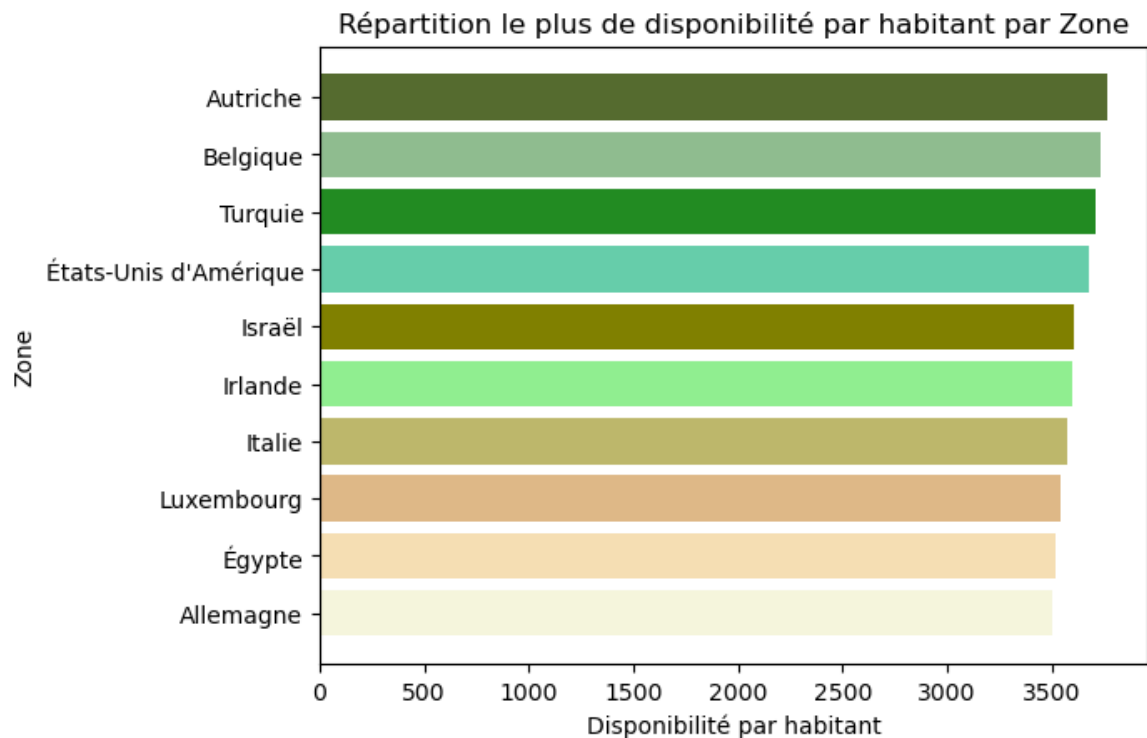
```
In [89]: data_parHabitant_trie[:10]
```

```
Out[89]: Zone
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
Name: Disponibilité alimentaire (Kcal/personne/jour), dtype: float64

```
In [90]: fig, ax = plt.subplots()
colors=['darkolivegreen','darkseagreen','forestgreen','mediumaquamarine',''
zones = data_parHabitant_trie.index[:10]
Top_aide_alimentaire = data_parHabitant_trie[:10]

ax.barh(zones,Top_aide_alimentaire,color=colors)
plt.gca().invert_yaxis()
ax.set_xlabel('Disponibilité par habitant')
ax.set_ylabel('Zone')
ax.set_title("Répartition le plus de disponibilité par habitant par Zone")
```



11 - Exemple de la Thaïlande pour le Manioc

</div>

```
In [91]: #création d'un dataframe avec uniquement la Thaïlande
sous_nutrition_Thaïlande = popu_sous_nutri[(popu_sous_nutri['Zone']=='Thaï')
sous_nutrition_Thaïlande
```

```
Out[91]:
```

	Zone	Année	Population	Sous_nutrition
1078	Thaïlande	2017	69209810.0	6200000.0

```
In [92]: #Calcul de la proportion sous nutrition en Thaïlande
proportion_Thaïlande=round(sous_nutrition_Thaïlande.Sous_nutrition*100/sc
proportion_Thaïlande
```

```
Out[92]:
```

1078	8.96
------	------

dtype: float64

```
In [93]: dispo_manioc= dispo_alimentaire[(dispo_alimentaire['Produit']=='Manioc')
dispo_manioc
```

Out[93]:

	Zone	Produit	Origine	Aliments pour animaux	Autres Utilisations	Disponibilité alimentaire (Kcal/personne/jour)	Disponi alimentai qua (kg/personne
12785	Thaïlande	Manioc	vegetale	1800000.0	2081.0	40.0	

In [94]: *#On calcule la proportion exportée en fonction de la Production*
proportion_exportee =round(dispo_manioc['Exportations - Quantité']*100/di
proportion_exportee

Out[94]: 12785 83.41
dtype: float64

In []:

In []: