

Etude de santé publique

Panorama de l'état de la malnutrition dans le monde

Importation des librairies :

```
In [1]: # Manipulation et calcul de DataFrames
import pandas as pd
import numpy as np
# Visualisation des données
import matplotlib.pyplot as plt
import seaborn as sns
import plotly
import plotly.express as px
```

Importation des données :

```
In [2]: disponibilite = pd.read_csv('dispo_alimentaire.csv')
aide_alimentaire = pd.read_csv('aide_alimentaire.csv')
population = pd.read_csv('population.csv')
sous_nutrition = pd.read_csv('sous_nutrition.csv')
disponibilite_2017 = pd.read_excel('dispo2017.xlsx') # Disponibilité des produits en 2017
iso = pd.read_excel('iso.xlsx') # Liste des codes iso-3 des pays (source FAO)
```

Constantes :

```
In [3]: besoin_calorie = 2400 # besoins journaliers moyen en kilocalories
```

Partie 1 : Nettoyage et modification des données :

1.1. Sous-nutrition :

```
In [4]: sous_nutrition.tail()
```

Out[4]:

	Zone	Année	Valeur
--	------	-------	--------

1213	Zimbabwe	2013-2015	NaN
1214	Zimbabwe	2014-2016	NaN
1215	Zimbabwe	2015-2017	NaN
1216	Zimbabwe	2016-2018	NaN
1217	Zimbabwe	2017-2019	NaN

1.1.1. Vérification des valeurs nulles et du bon formatage des données :

```
In [5]: sous_nutrition.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1218 entries, 0 to 1217
Data columns (total 3 columns):
#   Column   Non-Null Count  Dtype
---  -
0   Zone     1218 non-null   object
1   Année    1218 non-null   object
2   Valeur   624 non-null    object
dtypes: object(3)
memory usage: 28.7+ KB
```

```
In [6]: sous_nutrition.Valeur.value_counts()
```

```
Out[6]:
<0.1      120
0.3        48
0.4        32
0.2        28
1.5        20
...
3.7         1
24.1        1
189.2       1
190.1       1
8.3         1
Name: Valeur, Length: 139, dtype: int64
```

```
In [7]: #Remplacement des valeurs non standards par zéro :
sous_nutrition['Valeur'] = sous_nutrition['Valeur'].replace('<0.1', 0)
```

```
In [8]: # formatage de la colonne 'Valeur' :
sous_nutrition.Valeur = sous_nutrition.Valeur.astype(float)
```

1.1.2. Supression des valeurs nulles :

```
In [9]: sous_nutrition.dropna(axis = 0, inplace = True) # Supression des lignes avec valeurs nul
```

```
In [10]: sous_nutrition.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 624 entries, 0 to 1199
Data columns (total 3 columns):
#   Column   Non-Null Count  Dtype
---  -
0   Zone     624 non-null    object
1   Année    624 non-null    object
2   Valeur   624 non-null    float64
dtypes: float64(1), object(2)
memory usage: 19.5+ KB
```

1.1.3. Conversion et renommage de la colonnes :

```
In [11]: sous_nutrition['Valeur'] = sous_nutrition['Valeur'] * 1000
sous_nutrition.rename(columns = {'Valeur' : 'Sous_nourris_en_milliers'}, inplace = True)
sous_nutrition.tail()
```

```
Out[11]:
```

	Zone	Année	Sous_nourris_en_milliers
--	------	-------	--------------------------

1195	Viet Nam	2013-2015	8300.0
1196	Viet Nam	2014-2016	7600.0
1197	Viet Nam	2015-2017	7100.0
1198	Viet Nam	2016-2018	6500.0
1199	Viet Nam	2017-2019	6100.0

```
In [12]: # Transformation des élément de la colonne "Année" au format 'AAAA':
sous_nutrition['Année'] = sous_nutrition['Année'].str.split('-', n=1, expand=True)[0]
# Formatage des années :
sous_nutrition['Année'] = sous_nutrition['Année'].astype(int)
```

```
In [13]: sous_nutrition['Année'] = sous_nutrition['Année'] + 1
sous_nutrition.head()
```

```
Out[13]:
```

	Zone	Année	Sous_nourris_en_milliers
0	Afghanistan	2013	8600.0
1	Afghanistan	2014	8800.0
2	Afghanistan	2015	8900.0
3	Afghanistan	2016	9700.0
4	Afghanistan	2017	10500.0

1.2. Population :

1.2.1. Vérification des valeurs nulles et du bon formatage des données :

```
In [14]: population.head()
```

```
Out[14]:
```

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

```
In [15]: population.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1416 entries, 0 to 1415
Data columns (total 3 columns):
#   Column  Non-Null Count  Dtype
---  -
0    Zone    1416 non-null     object
1   Année    1416 non-null     int64
2   Valeur   1416 non-null     float64
dtypes: float64(1), int64(1), object(1)
memory usage: 33.3+ KB
```

1.2.2. Renommage des colonnes :

```
In [16]: population.rename(columns = {'Valeur' : 'Population_en_milliers'}, inplace = True)
population.head(3)
```

```
Out[16]:
```

	Zone	Année	Population_en_milliers
0	Afghanistan	2013	32269.589
1	Afghanistan	2014	33370.794
2	Afghanistan	2015	34413.603

1.3. Aide alimentaire :

1.3.1. Vérification des valeurs nulles et du bon formatage des données :

```
In [17]: aide_alimentaire.head()
```

```
Out[17]:
```

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

```
In [18]: aide_alimentaire.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1475 entries, 0 to 1474
Data columns (total 4 columns):
#   Column                Non-Null Count  Dtype
---  ---
0   Pays bénéficiaire     1475 non-null   object
1   Année                 1475 non-null   int64
2   Produit               1475 non-null   object
3   Valeur                1475 non-null   int64
dtypes: int64(2), object(2)
memory usage: 46.2+ KB
```

1.3.2. Renommage des colonnes :

```
In [19]: aide_alimentaire.rename(columns = {'Pays bénéficiaire' : 'Zone', 'Valeur' : 'Aide_en_tonnes'}, inplace = True)
aide_alimentaire.tail()
```

```
Out[19]:
```

	Zone	Année	Produit	Aide_en_tonnes
1470	Zimbabwe	2015	Mélanges et préparations	96
1471	Zimbabwe	2013	Non-céréales	5022
1472	Zimbabwe	2014	Non-céréales	2310
1473	Zimbabwe	2015	Non-céréales	306
1474	Zimbabwe	2013	Riz, total	64

1.4.1. Vérification des valeurs nulles et du bon formatage des données :

Out[20]:

```
In [21]: disponibilite.info()
```

[illegible]

```

t64
15  Semences                                2091 non-null    floa
t64
16  Traitement                              2292 non-null    floa
t64
17  Variation de stock                      6776 non-null    floa
t64
dtypes: float64(15), object(3)
memory usage: 2.1+ MB

```

1.4.2. Vérification des valeurs aberrantes :

In [22]: `disponibilite.describe()`

Out[22]:

	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)
count	2720.000000	5496.000000	14241.000000	14015.000000	11794.000000	11561.000000
mean	479.501838	157.391376	34.789832	8.719368	1.283111	1.223608
std	4240.119637	5076.785816	107.287655	24.618223	3.680399	3.598686
min	0.000000	0.000000	-21.000000	-1.930000	-0.030000	-0.370000
25%	0.000000	0.000000	0.000000	0.060000	0.010000	0.010000
50%	4.000000	0.000000	4.000000	0.830000	0.080000	0.100000
75%	74.000000	4.000000	21.000000	5.190000	0.630000	0.660000
max	150000.000000	347309.000000	1711.000000	430.760000	60.760000	54.970000

1.4.3. Suppression des valeurs négatives

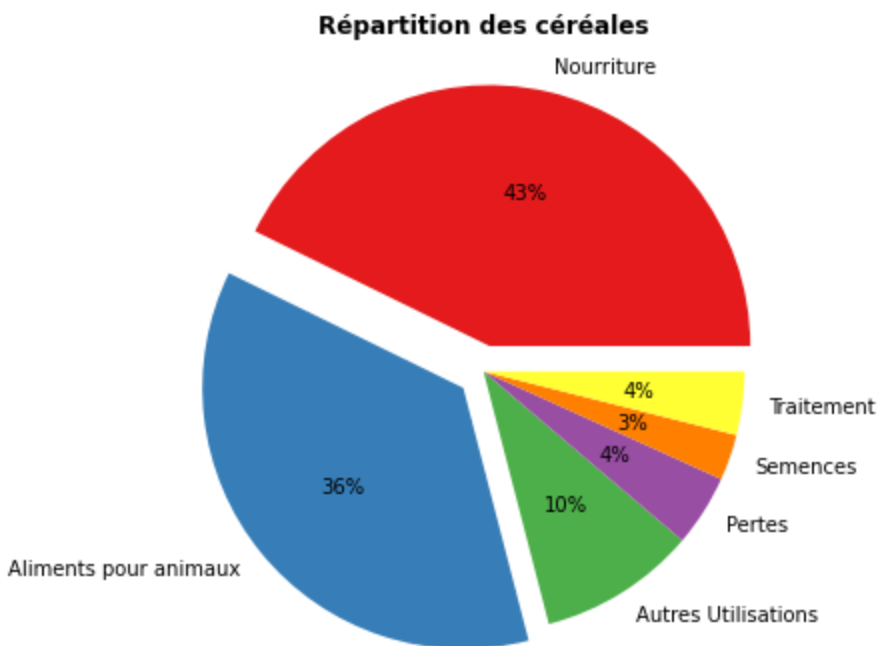
Suppression des valeurs "Disponibilité intérieure" négatives

In [23]: `list_index_dispo_neg = disponibilite[disponibilite['Disponibilité intérieure'] < 0].index`
`disponibilite.drop(list_index_dispo_neg, inplace = True)`

In [24]: `# Vérification des valeurs négatives restantes:`
`disponibilite.min()`

Out[24]:

Zone	Afghanistan
Produit	Abats Comestible
Origine	animale
Aliments pour animaux	0.0
Autres Utilisations	0.0
Disponibilité alimentaire (Kcal/personne/jour)	0.0
Disponibilité alimentaire en quantité (kg/personne/an)	0.0
Disponibilité de matière grasse en quantité (g/personne/jour)	0.0
Disponibilité de protéines en quantité (g/personne/jour)	0.0
Disponibilité intérieure	0.0
Exportations - Quantité	0.0
Importations - Quantité	0.0
Nourriture	0.0
Pertes	0.0
Production	0.0
Semences	0.0
Traitement	-19.0
Variation de stock	-39863.0
dtype:	object



2.3. Utilisation du manioc par la Thaïlande aux égards de la proportion de personnes en sous-nutrition (exportation par rapport à la production)

```
In [29]: # fusion de sous_nutrition et de population :
fusion_sous_nutri_popu = pd.merge(sous_nutrition, population, on= ['Zone', 'Année'])
# fusion de sous_nutrition, de population et de disponibilité :
fusion = pd.merge(disponibilite, fusion_sous_nutri_popu, on = 'Zone')
# Sélection des données sur le manioc :
manioc_data_thai_2013 = fusion.loc[(fusion['Produit'] == 'Manioc')
                                   & (fusion['Zone'] == 'Thaïlande')
                                   & (fusion['Année'] == 2013),
                                   ['Zone', 'Sous_nourris_en_milliers', 'Population_en_milliers',
                                   'Importations - Quantité', 'Production']]
# Pourcentage que représentent les exportations par rapport à la production de manioc :
A= round(manioc_data_thai_2013['Exportations - Quantité'] * 100 / manioc_data_thai_2013['Production'], 2)
# Quantités exportées de manioc en 2013 :
B = round(manioc_data_thai_2013['Exportations - Quantité']/1000, 2)
# Nombre de sous-nourris en 2013 :
C = list(manioc_data_thai_2013['Sous_nourris_en_milliers'] / 1000)

manioc_data_thai_2013
```

```
Out[29]:
```

	Zone	Sous_nourris_en_milliers	Population_en_milliers	Exportations - Quantité	Importations - Quantité	Production
48299	Thaïlande	6200.0	68144.518	25214.0	1250.0	30228.0

```
In [30]: print('En 2013, la Thaïlande a exporté',list(B)[0], ' millions de tonnes de manioc (' ,list(C)[0], ' millions de personnes souffraient de sous-nutrition.')
```

En 2013, la Thaïlande a exporté 25.21 millions de tonnes de manioc (83.41 % de sa production) alors que 6.2 millions de personnes souffraient de sous-nutrition.

2.4. Proportion de personnes en état de sous-nutrition

```
In [31]: # Sélection des données sur la populations des sous-nourris en 2017 :
sous_nutri_popu_2017 = fusion_sous_nutri_popu.loc[fusion_sous_nutri_popu['Année'] == 2017]
```



```

# calcul du pourcentage de sous-nourris par zone :
sous_nutri_popu_2017['Proportion_sous_nourris_en_%'] = round(sous_nutri_popu_2017['Sous_
                                                             /sous_nutri_popu_2017['Populatio

# Les 10 pays à forte proportion de sous-nourris :
sous_nutri_popu_2017[['Zone', 'Proportion_sous_nourris_en_%']].nlargest(10, 'Proportion_

```

Out[31]:

	Zone	Proportion_sous_nourris_en_%
249	Haïti	48.26
496	République populaire démocratique de Corée	47.19
346	Madagascar	41.06
334	Libéria	38.28
322	Lesotho	38.25
569	Tchad	37.96
508	Rwanda	35.06
400	Mozambique	32.81
581	Timor-Leste	32.17
4	Afghanistan	28.93

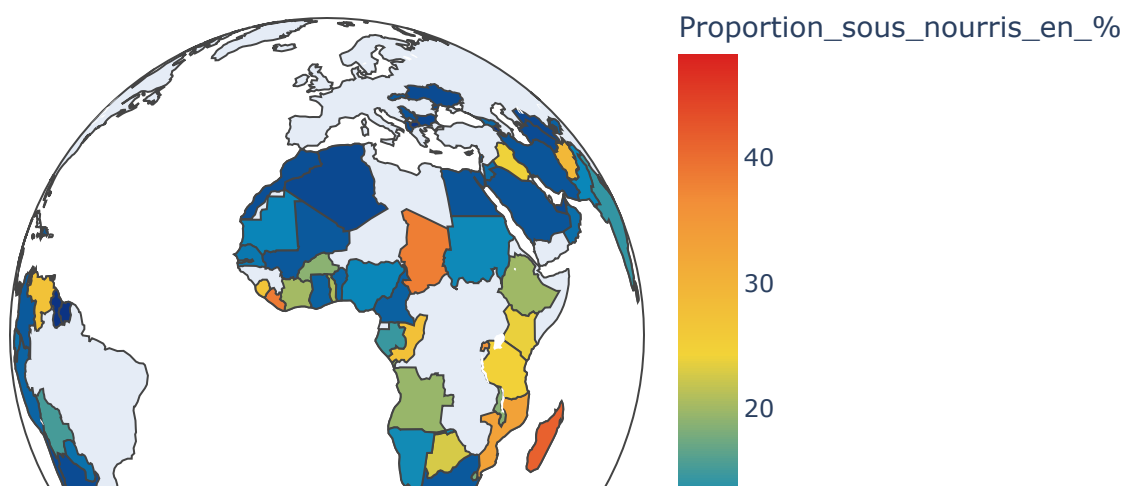
```

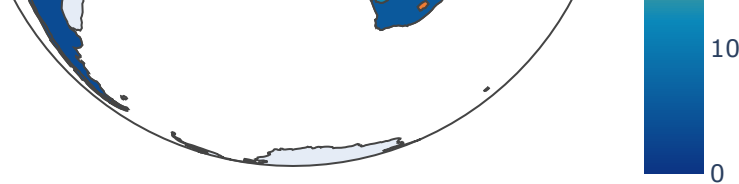
In [32]: # Visualisation des pays en proportion de sous-nourris :
df=pd.merge(sous_nutri_popu_2017, iso, on='Zone',how='inner')
fig = px.choropleth(df,
                    title = 'les pays et leur proportion de personnes en état de sous-nu
                    projection = 'orthographic',
                    locations = 'Iso3',
                    locationmode= 'ISO-3',
                    color = 'Proportion_sous_nourris_en_%',
                    hover_name = 'Zone',
                    hover_data = ['Proportion_sous_nourris_en_%'],
                    color_continuous_scale="Portland"

fig.show()

```

les pays et leur proportion de personnes en état de sous-nutrition





2.5. Nombre théorique de personnes qui pourraient être nourries :

```
In [33]: disponibilite_2017.head() # Disponibilité des produits en 2017
```

```
Out[33]:
```

	Zone	Produit	Valeur
0	Afghanistan	Produits Vegetaux	2108
1	Afghanistan	Produits Animaux	195
2	Afrique du Sud	Produits Vegetaux	2437
3	Afrique du Sud	Produits Animaux	466
4	Albanie	Produits Vegetaux	2292

```
In [34]: # Renommage de la colonne 'Valeur' par 'Disponibilité alimentaire (Kcal/personne/jour)':
disponibilite_2017.rename(columns = {'Valeur' : 'Disponibilité alimentaire (Kcal/personn
# disponibilité totale en (Kcal/personne/jour) par zone:
disponibilite_2017_zone = disponibilite_2017.groupby('Zone').sum()
# La population en 2017
population_2017 = population.loc[population['Année'] == 2017, ['Zone', 'Population_en_mi
```

```
In [35]: # Fusion de la disponibilité totale en (Kcal/personne/jour) par zone et de la population
fusion_dispo_popu_2017 = pd.merge(disponibilite_2017_zone, population_2017, on ='Zone',
# Nombre théorique de personnes qu'on pourrait nourrir avec ce qui est disponible par zo
fusion_dispo_popu_2017['disponibilite_zone'] = fusion_dispo_popu_2017['Disponibilité ali
# Nombre théorique total des personnes qu'on pourrait nourrir avec ce qui est disponible
D = round(fusion_dispo_popu_2017['disponibilite_zone'].sum())
```

```
In [36]: print(D, 'personnes pourraient être théoriquement nourries.')

9191834865 personnes pourraient être théoriquement nourries.
```

2.6. Nombre théorique de personnes qui pourraient être nourries (Produits végétaux)

```
In [37]: ## disponibilité en (Kcal/personne/jour) et en produits végétaux (année 2017):
disponibilite_2017_vegetal = disponibilite_2017.loc[disponibilite_2017['Produit'] == 'Pr
['Zone', 'Disponibilité alimentaire
```

```
In [38]: # Fusion de la disponibilité végétale et population en 2017:
fusion_vegetal = pd.merge(disponibilite_2017_vegetal, population_2017, on ='Zone', how =
# Nombre théorique de personnes qu'on pourrait nourrir avec ce qui est disponible en pr
fusion_vegetal['disponibilite_zone'] = fusion_vegetal['Disponibilité alimentaire (Kcal/p
# Nombre théorique total des personnes qu'on pourrait nourrir avec ce qui est disponible
E = round(fusion_vegetal['disponibilite_zone'].sum())
```

```
In [39]: print(E, 'personnes pourraient être théoriquement nourries avec des produits issus de la  
7562108935 personnes pourraient être théoriquement nourries avec des produits issus de l  
a terre.
```

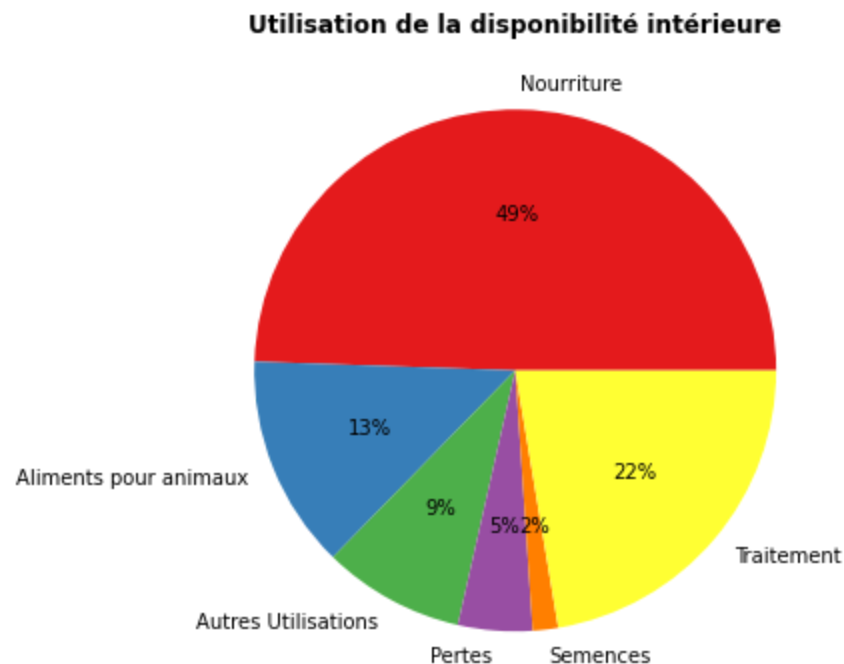
2.7. Utilisation de la disponibilité intérieure :

Disponibilité intérieure = Nourriture + perte + semences + Aliments pour animaux + Autres Utilisations + Traitement

```
In [40]: # Sélection des éléments qui composent la disponibilité intérieure et leur zone :  
dispo = disponibilite[['Zone', 'Nourriture', 'Aliments pour animaux', 'Autres Utilisatio  
                'Pertres', 'Semences', 'Traitement']]  
# Somme totale des éléments qui composent la disponibilité intérieure  
repartition_dispo = dispo.groupby('Zone').sum().sum()  
repartition_dispo
```

```
Out[40]: Nourriture                4875206.0  
Aliments pour animaux    1304176.0  
Autres Utilisations      864950.0  
Pertres                  453619.0  
Semences                 154661.0  
Traitement               2204215.0  
dtype: float64
```

```
In [41]: # Représentation de la part de chaque élément qui compose la disponibilité intérieure (d  
plt.figure(figsize = (6,6))  
colors = sns.color_palette("Set1")  
labels = ['Nourriture', 'Aliments pour animaux', 'Autres Utilisations', 'Pertres', 'Seme  
plt.pie(repartition_dispo, labels = labels, colors = colors, autopct = '%0.0f%')  
plt.title("Utilisation de la disponibilité intérieure", fontweight = "bold")  
plt.show()
```



2.8. Pays pour lesquels la proportion de personnes sous-alimentées est la plus forte en 2017

```
In [42]: # Nombre de sous-nourris par zone (en 2017) :  
sous_nutrition_2017 = sous_nutrition.loc[sous_nutrition['Année'] == 2017, ['Zone', 'Sous
```

```
In [43]: # Création de la colonne "Porportion_sous_nourris":
sous_nutrition_2017['Porportion_sous_nourris'] = round(sous_nutrition_2017['Sous_nourris_
/ sous_nutrition_2017['Sous_nourri
```

```
In [44]: # Liste des 10 pays pour lesquels la proportion de personnes sous-alimentées est la plus
sous_nutrition_2017[['Zone', 'Porportion_sous_nourris']].nlargest(10, 'Porportion_sous_nour
```

```
Out[44]:
```

	Zone	Porportion_sous_nourris
508	Inde	35.0
820	Pakistan	5.0
100	Bangladesh	4.0
370	Éthiopie	4.0
514	Indonésie	4.0
772	Nigéria	4.0
868	Philippines	3.0
952	République-Unie de Tanzanie	3.0
4	Afghanistan	2.0
526	Iraq	2.0

2.9. Liste des pays ayant le plus bénéficié d'aide depuis 2013

```
In [45]: # Les aides alimentaires attribuée depuis 2013 par zone :
aide_alimentaire_pays = aide_alimentaire.loc[aide_alimentaire['Année'] >= 2013,
['Zone', 'Aide_en_tonnes']].groupby('Zone')

#Liste des 10 pays ayant le plus bénéficié d'aide depuis 2013
aide_alimentaire_pays.nlargest(10, 'Aide_en_tonnes')
```

```
Out[45]:
```

	Aide_en_tonnes
Zone	
République arabe syrienne	1858943
Éthiopie	1381294
Yémen	1206484
Soudan du Sud	695248
Soudan	669784
Kenya	552836
Bangladesh	348188
Somalie	292678
République démocratique du Congo	288502
Niger	276344

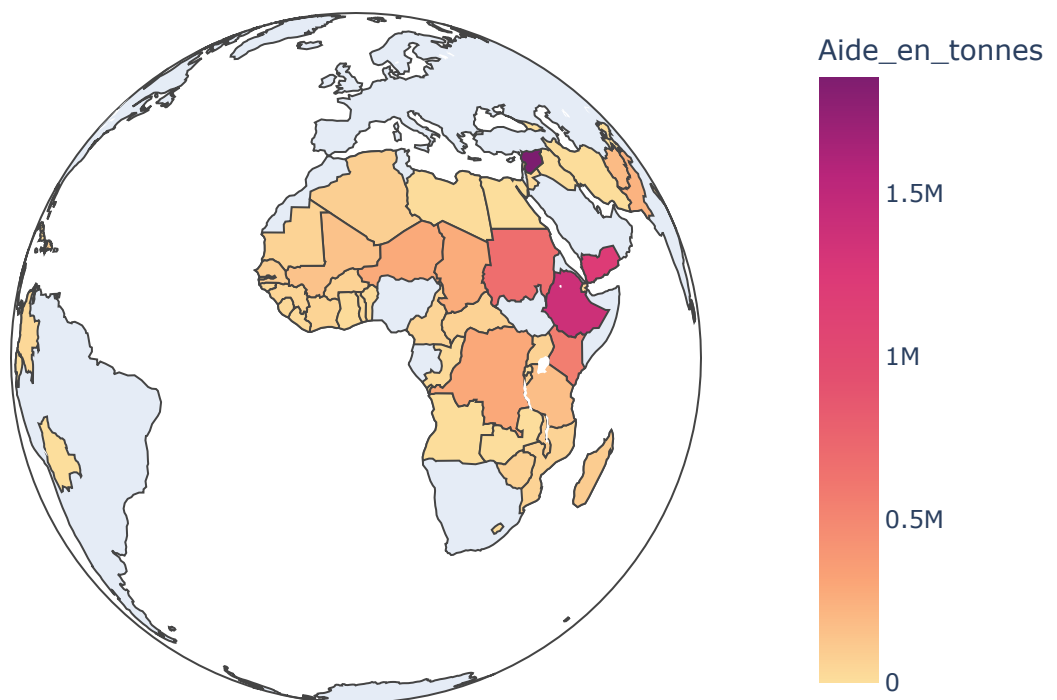
```
In [46]: # Visualisation des pays ayant le plus bénéficié d'aide depuis 2013 :

df1=pd.merge(aide_alimentaire_pays, iso, on='Zone',how='inner')
```

```
fig = px.choropleth(df1,
                    title = 'Pays bénéficiaires d'aide depuis 2013',
                    projection = 'orthographic',
                    locations = 'Iso3',
                    locationmode = 'ISO-3',
                    color = 'Aide_en_tonnes',
                    #hover_name = 'Zone',
                    hover_data = ['Aide_en_tonnes'],
                    color_continuous_scale = "Sunsetdark"
                    )

fig.show()
```

Pays bénéficiaires d'aide depuis 2013



2.10. Liste des pays ayant le plus de disponibilité/habitant:

```
In [47]: # La disponibilité en (kg/personne/an) par zone :
disponibilite_hab_zone = disponibilite[['Zone', 'Disponibilité alimentaire en quantité (
# Liste des 10 pays ayant le plus de disponibilité/habitant :
disponibilite_hab_zone.nlargest(10, 'Disponibilité alimentaire en quantité (kg/personne/
```

Out[47]:

Disponibilité alimentaire en quantité (kg/personne/an)	
Zone	
Monténégro	1210.41
Luxembourg	1150.17
Irlande	1147.12
Albanie	1095.54

Zone	
Monténégro	1210.41
Luxembourg	1150.17
Irlande	1147.12
Albanie	1095.54

Finlande	1093.77
Grèce	1070.90
Pays-Bas	1069.22
Autriche	1069.20
Lituanie	1050.07
Danemark	1030.46

2.11. Liste des pays ayant le moins de disponibilité/habitant:

```
In [48]: # Liste des 10 pays ayant le moins de disponibilité/habitant :
disponibilite_hab_zone.nsmallest(10, 'Disponibilité alimentaire en quantité (kg/personne
```

```
Out[48]:
```

Disponibilité alimentaire en quantité (kg/personne/an)	
Zone	
Tchad	319.45
Zimbabwe	336.59
Zambie	350.08
Afghanistan	351.41
Éthiopie	354.55
Guinée-Bissau	358.48
Bangladesh	364.54
Timor-Leste	369.10
Yémen	375.25
Sénégal	376.72

```
In [ ]:
```