PROJET 4 DATA ANALYST

Réalisez une étude de santé publique avec R ou Python

Note jeremy Est ce qu'il faut faire le calcul de la sous nutrition sur les pays qu'on a ? Est ce qu'il faut faire des graphiques ? Rajouter le soja La liste des céréales est difficile a trouver ...

Etape 1 - Importation des librairies et chargement des fichiers

1.1 - Importation des librairies

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

1.2 - Chargement des fichiers Excel

In [ ]:
```
#Importation du fichier population.csv
population = pd.read_csv('population.csv')

#Importation du fichier dispo_alimentaire.csv
dispoAlimentaire = pd.read_csv('dispo_alimentaire.csv')

#Importation du fichier aide_alimentaire.csv
aideAlimentaire = pd.read_csv('aide_alimentaire.csv')

#Importation du fichier sous_nutrition.csv
sousNutrition = pd.read_csv('sous_nutrition.csv')
```

Etape 2 - Analyse exploratoire des fichiers

2.1 - Analyse exploratoire du fichier population

In [ ]:
```
#Afficher les dimensions du dataset
print("Le tableau comporte {} observation(s) ou article(s)".format(popula
```

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

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

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

#La nature des données dans chacune des colonnes
# Convertir la série des types de données en un DataFrame
dtypes_df = population.dtypes.to_frame()
# Formater le DataFrame en une chaîne
dtypes_str = dtypes_df.to_string()
# Afficher le résultat
print("La nature des colonnes est :\n{}".format(dtypes_str))

#Le nombre de valeurs présentes dans chacune des colonnes
print("Le nombre de valeurs présentes dans chacune des colonnes est :\n{}
```

```
Le tableau comporte 3 colonne(s)
La nature des colonnes est :
              0
Zone     object
Année     int64
Valeur  float64
Le nombre de valeurs présentes dans chacune des colonnes est :
Zone      1416
Année     1416
Valeur    1416
dtype: int64
```

Le tableau comporte 3 colonne(s)

La nature des colonnes est : Zone object Année int64 Valeur float64

Le nombre de valeurs présentes dans chacune des colonnes est : Zone 1416 Année 1416 Valeur 1416

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

```
Les 5 premières lignes de la table population
          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
```

Les 5 premières lignes de la table population 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 [ ]:
```
#Nous allons harmoniser les unités. Pour cela, nous avons décidé de multi
#Multiplication de la colonne valeur par 1000

print("Nous allons harmoniser les unités en multipliant la population par
```

```
# Multiplication de la colonne "Valeur" par 1000
population['Population'] = population['Valeur'] * 1000

# Convertir les valeurs en entiers
population['Population'] = population['Population'].astype(int)

# Afficher le résultat
print("Nous avons harmonisé les unités en multipliant la population par 1
```

```
Nous allons harmoniser les unités en multipliant la population par 1000.
Nous avons harmonisé les unités en multipliant la population par 1000 :
             Zone  Année     Valeur  Population
0     Afghanistan   2013  32269.589    32269589
1     Afghanistan   2014  33370.794    33370794
2     Afghanistan   2015  34413.603    34413603
3     Afghanistan   2016  35383.032    35383032
4     Afghanistan   2017  36296.113    36296113
...           ...    ...        ...         ...
1411     Zimbabwe   2014  13586.707    13586707
1412     Zimbabwe   2015  13814.629    13814629
1413     Zimbabwe   2016  14030.331    14030331
1414     Zimbabwe   2017  14236.595    14236595
1415     Zimbabwe   2018  14438.802    14438802

[1416 rows x 4 columns]
```

Nous avons harmonisé les unités en multipliant la population par 1000 : 0 32269589 1 33370794 2 34413603 3 35383032 4 36296113 ... 1411 13586707 1412 13814629 1413 14030331 1414 14236595 1415 14438802

In [ ]:
```
# Supprimer la colonne "Valeur" originale
population.drop(columns=['Valeur'], inplace=True)

# Afficher le résultat
print(population.head())
```

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

Zone Année Population 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

2.2 - Analyse exploratoire du fichier disponibilité alimentaire

In [ ]:
```
#Afficher les dimensions du dataset

print("Le tableau comporte {} observation(s) ou article(s)".format(dispoA
```

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

Le tableau comporte 15605 observation(s) ou article(s)

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

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

Le tableau comporte 18 colonne(s)

In [ ]:
```
#Affichage les 5 premières lignes de la table
print("Les 5 premières lignes de la table dispositions alimentaires sont
print(dispoAlimentaire.head())
```

Les 5 premières lignes de la table dispositions alimentaires sont :

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

```
4         82.0     NaN          NaN        NaN          NaN                  Na
N
```

Les 5 premières lignes de la table dispositions alimentaires sont : Zone Produit Origine Aliments pour animaux Autres Utilisations ... Pertes Production Semences Traitement Variation de stock 0 Afghanistan Abats Comestible animale NaN NaN ... NaN 53.0 NaN NaN NaN 1 Afghanistan Agrumes, Autres vegetale NaN NaN ... 2.0 3.0 NaN NaN NaN 2 Afghanistan Aliments pour enfants vegetale NaN NaN ... NaN NaN NaN NaN NaN 3 Afghanistan Ananas vegetale NaN NaN ... NaN NaN NaN NaN NaN 4 Afghanistan Bananes vegetale NaN NaN ... NaN NaN NaN NaN NaN

In [ ]:
```
#remplacement des NaN dans le dataset par des 0
dispoAlimentaire = dispoAlimentaire.fillna(0)
print(dispoAlimentaire.head())
```

```
          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 - Quant
ité  \
0                      53.0                       0.0
0.0
1                      41.0                       2.0                     4
0.0
2                       2.0                       0.0
2.0
3                       0.0                       0.0
0.0
4                      82.0                       0.0                     8
2.0

   Nourriture  Pertes  Production  Semences  Traitement  Variation de stoc
k
0        53.0     0.0        53.0       0.0         0.0                 0.
0
1        39.0     2.0         3.0       0.0         0.0                 0.
0
2         2.0     0.0         0.0       0.0         0.0                 0.
0
3         0.0     0.0         0.0       0.0         0.0                 0.
0
4        82.0     0.0         0.0       0.0         0.0                 0.
0
```

```
 Zone                Produit   Origine  Aliments pour
animaux  Autres Utilisations  Disponibilité alimentaire
(Kcal/personne/jour)  Disponibilité alimentaire en
quantité (kg/personne/an)  ...  Importations - Quantité
Nourriture  Pertes  Production  Semences  Traitement
Variation de stock
```

0 Afghanistan Abats Comestible animale 0.0 0.0 5.0 1.72 ... 0.0 53.0 0.0 53.0 0.0 0.0 0.0 1 Afghanistan Agrumes, Autres vegetale 0.0 0.0 1.0 1.29 ... 40.0 39.0 2.0 3.0 0.0 0.0 0.0 2 Afghanistan Aliments pour enfants vegetale 0.0 0.0 1.0 0.06 ... 2.0 2.0 0.0 0.0 0.0 0.0 0.0 3 Afghanistan Ananas vegetale 0.0 0.0 0.0 0.00 ... 0.0 0.0 0.0 0.0 0.0 0.0 0.0 4 Afghanistan Bananes vegetale 0.0 0.0 4.0 2.70 ... 82.0 82.0 0.0 0.0 0.0 0.0 0.0

In [ ]:
```
#multiplication de toutes les lignes contenant des milliers de tonnes en
colonnes_a_convertir = ['Autres Utilisations', 'Disponibilité intérieure'
                        'Importations - Quantité', 'Nourriture', 'Pertes'
                        'Traitement', 'Variation de stock', 'Aliments pou

for colonne in colonnes_a_convertir:
    dispoAlimentaire[colonne] *= 1000  # Convertir de milliers de tonnes
```

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

```
          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 - Quant
ité  \
0                   53000.0                     0.0
0.0
1                   41000.0                  2000.0                  4000
0.0
2                    2000.0                     0.0                   200
0.0
3                       0.0                     0.0
0.0
4                   82000.0                     0.0                  8200
0.0

   Nourriture  Pertes  Production  Semences  Traitement  Variation de stoc
k
0     53000.0     0.0     53000.0       0.0         0.0                 0.
0
1     39000.0  2000.0      3000.0       0.0         0.0                 0.
0
2      2000.0     0.0         0.0       0.0         0.0                 0.
0
3         0.0     0.0         0.0       0.0         0.0                 0.
0
4     82000.0     0.0         0.0       0.0         0.0                 0.
0
```

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) ... Nourriture Pertes Production Semences Traitement Variation de stock 0 0 0 5000000 1720000 200000 770000 ... 53000000 0 53000000 0 0 0 1 0 0 1000000 1290000 10000 20000 ... 39000000 2000000 3000000 0 0 0 2 0 0 1000000 60000 10000 30000 ... 2000000 0 0 0 0 0 3 0 0 0 0 0 0 ... 0 0 0 0 0 0 4 0 0 4000000 2700000 20000 50000 ... 82000000 0 0 0 0 0

2.3 - Analyse exploratoire du fichier aide alimentaire

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

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

In [ ]:
```
#Consulter le nombre de colonnes
print("Le nombre de colonnes est : {}".format(aideAlimentaire.shape[1]))
```

```
Le nombre de colonnes est : 4
```

In [ ]:
```
#Affichage les 5 premières lignes de la table
print("les 5 premières lignes de la table aideAlimentaire sont : ")
print(aideAlimentaire.head())
```

```
les 5 premières lignes de la table aideAlimentaire sont :
  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 [ ]:
```
#changement du nom de la colonne Pays bénéficiaire par Zone
aideAlimentaire.rename(columns={'Pays bénéficiaire':'Zone'}, inplace=True
print(aideAlimentaire.head())
```

```
          Zone  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 [ ]:
```
#Multiplication de la colonne Aide_alimentaire qui contient des tonnes pa
aideAlimentaire['Valeur'] = aideAlimentaire['Valeur']*1000
```

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

```
          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

In [ ]:
```
#Afficher les dimensions du dataset
print("Le tableau comporte {} observation(s) ou article(s)".format(sousNu
```

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

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

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

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

```
          Zone      Année Valeur
0  Afghanistan  2012-2014    8.6
1  Afghanistan  2013-2015    8.8
2  Afghanistan  2014-2016    8.9
3  Afghanistan  2015-2017    9.7
4  Afghanistan  2016-2018   10.5
```

In [ ]:
```
#Conversion de la colonne sous nutrition en numérique
sousNutrition['Valeur'] = pd.to_numeric(sousNutrition['Valeur'], errors='
```

In [ ]:
```
#Conversion de la colonne (avec l'argument errors=coerce qui permet de co
#Puis remplacement des NaN en 0
sousNutrition['Valeur'] = sousNutrition['Valeur'].fillna(0)
print(sousNutrition['Valeur'])
```

```
0        8.6
1        8.8
2        8.9
3        9.7
4       10.5
       ...
1213     0.0
1214     0.0
1215     0.0
1216     0.0
1217     0.0
Name: Valeur, Length: 1218, dtype: float64
```

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

```
              Zone      Année  sous_nutrition
0      Afghanistan  2012-2014             8.6
1      Afghanistan  2013-2015             8.8
2      Afghanistan  2014-2016             8.9
3      Afghanistan  2015-2017             9.7
4      Afghanistan  2016-2018            10.5
...            ...        ...             ...
1213      Zimbabwe  2013-2015             0.0
1214      Zimbabwe  2014-2016             0.0
1215      Zimbabwe  2015-2017             0.0
1216      Zimbabwe  2016-2018             0.0
1217      Zimbabwe  2017-2019             0.0

[1218 rows x 3 columns]
```

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

# Conversion en chiffres entiers
sousNutrition['sous_nutrition'] = sousNutrition['sous_nutrition'].astype(
```

In [ ]:
```python
#Afficher les 5 premières lignes de la table
print(sousNutrition.head())
```

```
          Zone      Année  sous_nutrition
0  Afghanistan  2012-2014         8600000
1  Afghanistan  2013-2015         8800000
2  Afghanistan  2014-2016         8900000
3  Afghanistan  2015-2017         9700000
4  Afghanistan  2016-2018        10500000
```

3.1 - Proportion de personnes en sous nutrition

In [ ]:
```python
# TRAVAIL SUR LA TABLE SOUS NUTRITION

# Définition de la fonction create_year_list (créer une liste d'années)
def create_year_list(year_range):
    start_year, end_year = map(int, year_range.split('-'))
    return list(range(start_year, end_year + 1))

# Utilisation de la fonction pour spliter les groupes d'années
sousNutrition['Année'] = sousNutrition['Année'].apply(create_year_list)

# Déplier la liste d'années pour chaque ligne (une ligne par année)
sousNutrition = sousNutrition.explode('Année')

# Réinitialiser les index /garantit que chaque ligne a un index unique et
sousNutrition = sousNutrition.reset_index(drop=True)


# ************************************************************


# Jointure entre les tables population et sousNutrition sur les colonnes
'''Inner join par défaut car on veut les données qui sont dans les deux t
fichier population contient plus de pays que fichier sous-nutrition et re
donc on ne peut pas convertir en entier la colonne sous_nutrition'''
```

```
jointure = pd.merge(population, sousNutrition, on=['Zone', 'Année'])

# Filtrer les résultats pour ne conserver que les lignes avec l'année 201
resultats_2017 = jointure[jointure['Année'] == 2017]

#Regrouper les données en fonction de la colonne Zone
#puis retenir la première valeur de la colonne Population et la dernière
# les données de la FAO en sous-nutrition sont déjà une moyenne sur trois
# Trouver la valeur de personnes en sous-nutrition pour chaque pays en 20
resultats_2017 = resultats_2017.groupby('Zone').agg({'Population': 'first

# Convertir la colonne 'sous_nutrition' en type entier
resultats_2017['sous_nutrition'] = resultats_2017['sous_nutrition'].astyp

# Calculer la proportion de sous-nutrition
resultats_2017['proportion_sous_nutrition'] = resultats_2017['sous_nutrit

# Arrondir les valeurs à deux chiffres après la virgule et rajouter le si
resultats_2017['proportion_sous_nutrition'] = resultats_2017['proportion_
```

In [ ]:
```
#Affichage du dataset
print(resultats_2017)
```

```
                      Zone  Population  sous_nutrition  \
0              Afghanistan    36296113        11100000
1          Afrique du Sud     57009756         3300000
2                  Albanie     2884169          100000
3                  Algérie    41389189         1200000
4                Allemagne    82658409               0
..                     ...         ...             ...
198  États-Unis d'Amérique   325084756               0
199               Éthiopie   106399924        21500000
200              Îles Cook       17507               0
201          Îles Marshall       58058               0
202           Îles Salomon      636039               0

    proportion_sous_nutrition
0                      30.58%
1                       5.79%
2                       3.47%
3                        2.9%
4                        0.0%
..                        ...
198                      0.0%
199                     20.21%
200                      0.0%
201                      0.0%
202                      0.0%

[203 rows x 4 columns]
```

In [ ]:
```
# Calcul et affichage du nombre de personnes en état de sous-nutrition en
total_sous_nutrition = round(resultats_2017['sous_nutrition'].sum())

# Utiliser la méthode format avec le spécificateur de format {:,} pour aj
formatted_total_sous_nutrition = "{:,.0f}".format(total_sous_nutrition)

print("Le nombre de personnes en état de sous-nutrition en 2017 est de :

#calcul du nombre de personnes dans la population mondiale en 2017
```

```
total_population = round(resultats_2017['Population'].sum())
formatted_total_population = "{:,.0f}".format(total_population)
print("Le nombre de personnes dans la population mondiale en 2017 est de

# Calcul et affichage du pourcentage de personnes en état de sous-nutriti
pourcentage_sous_nutrition = total_sous_nutrition / total_population * 10
pourcentage_sous_nutrition_arrondi = round(pourcentage_sous_nutrition, 2)
print("Le pourcentage de personnes en état de sous-nutrition dans le mond
```

```
Le nombre de personnes en état de sous-nutrition en 2017 est de : 544,200,
000 personnes
Le nombre de personnes dans la population mondiale en 2017 est de : 7,543,
798,769 personnes
Le pourcentage de personnes en état de sous-nutrition dans le monde en 201
7 est de : 7.21%
```

FOCUS SUR MANQUE DE DONNEES DE SOUS-NUTRITION

Afficher la somme de Population dont sous_nutrition = 0

```
print('Nombre de personnes dont nous n"avons pas de
données de sous-nutrition :')
print(resultats_2017[resultats_2017['sous_nutrition'] ==
0]['Population'].sum())
```

Nombre de personnes total du fichier

```
print('Nombre total de personnes :')
print(resultats_2017['Population'].sum())
```

Calculer la proportion de la population ayant une sous-nutrition égale à zéro

```
proportion_sous_nutrition_0 =
(resultats_2017[resultats_2017['sous_nutrition'] == 0]
['Population'].sum() / resultats_2017['Population'].sum())
* 100
#print("Proportion de la population sans données de sous-
nutrition: {:.2f}%".format(proportion_sous_nutrition_0))
```

liste des pays sans données de sous-nutrition

```
print('Pays sans données de sous-nutrition :')
print(resultats_2017[resultats_2017['sous_nutrition'] ==
0]['Zone'])
```

Les 118 pays sans données ou valeur nulle de sous nutrition sont des pays riches ou des pays en guerre. Ils me semblent important de les conserver pour ne pas fausser les résultats. les exclures revient à exclure presque la moitié de la population mondiale, sachant que dans cette liste il y a un bon nombre qui ne sont pas en sous-nutrition.

3.2 - Nombre théorique de personnes qui pourraient être nourries

In [ ]:
```
#Combien mange en moyenne un être humain ? Source => dispoAlimentaire
# Calculer la disponibilité alimentaire moyenne par personne par produit

disponibilite_moyenne_par_personne = dispoAlimentaire[['Produit', 'Dispon
print(disponibilite_moyenne_par_personne)
```

```
                        Disponibilité alimentaire en quantité (kg/personn
e/an)
Produit
Abats Comestible                                                 2.730345
Agrumes, Autres                                                  1.306688
Alcool, non Comestible                                           0.000000
Aliments pour enfants                                            0.380936
Ananas                                                           3.890536
...                                                                   ...
Viande de Suides                                                14.407616
Viande de Volailles                                             20.561322
Viande, Autre                                                    1.800977
Vin                                                              6.161337
Épices, Autres                                                   0.622616

[98 rows x 1 columns]
```

In [ ]:
```
#Filtrer la table population sur l'année 2017
population_2017 = population[population['Année'] == 2017]

#compter le nombre de Pays dans le DataFrame population_2017
nombre_pays = population_2017['Zone'].nunique()
print("Le nombre de pays dans le DataFrame population_2017 est de : {}".f
#compter le nombre de pays dans dispoAlimentaire
nombre_pays_dispoAlimentaire = dispoAlimentaire['Zone'].nunique()
print("Le nombre de pays dans le DataFrame dispoAlimentaire est de : {}".

#Faire une jointure entre les tables disponibilité alimentaire et populat
'''Inner join par défaut car on veut les données qui sont dans les deux t
dispoAlimentaire contient moins de pays que population et retourne des Na
dispoAlimentaire = pd.merge(dispoAlimentaire, population_2017, on='Zone')
```

```
Le nombre de pays dans le DataFrame population_2017 est de : 236
Le nombre de pays dans le DataFrame dispoAlimentaire est de : 174
```

In [ ]:
```
#Affichage du nouveau dataframe
# Afficher les premières lignes du DataFrame
print(dispoAlimentaire.head())
```

```
          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 - Quant
ité  \
0                   53000.0                     0.0
0.0
1                   41000.0                  2000.0                  4000
0.0
2                    2000.0                     0.0                   200
0.0
3                       0.0                     0.0
0.0
4                   82000.0                     0.0                  8200
0.0

   Nourriture  Pertes  Production  Semences  Traitement  Variation de stoc
k  \
0     53000.0     0.0     53000.0       0.0         0.0                 0.
0
1     39000.0  2000.0      3000.0       0.0         0.0                 0.
0
2      2000.0     0.0         0.0       0.0         0.0                 0.
0
3         0.0     0.0         0.0       0.0         0.0                 0.
0
4     82000.0     0.0         0.0       0.0         0.0                 0.
0
```

```
   Année  Population
0   2017    36296113
1   2017    36296113
2   2017    36296113
3   2017    36296113
4   2017    36296113
```

In [ ]:
```python
# Calculer la somme de la disponibilité alimentaire en calories par perso
Nbre_cal_journaliere_par_personne_disponibles = dispoAlimentaire.groupby(
Nbre_cal_journaliere_par_personne_disponibles.rename(columns={'Disponibil

# Fusionner les DataFrames sur la colonne 'Zone'
dispoAlimentaire = pd.merge(dispoAlimentaire, Nbre_cal_journaliere_par_pe
'''outer join pour prendre en compte l'entiereté des données de dispoAlim

# Grouper par pays et obtenir la première ligne de chaque groupe
NourritureDispo = dispoAlimentaire.groupby('Zone').first().reset_index()[
'''le fichier disopAlimentaire duplique les zones en fonction des produit
pour éviter de cumuler les calories'''

# Calculer la colonne Nbre_cal_dispo_totale
NourritureDispo['Nbre_cal_dispo_totale'] = NourritureDispo['Population']

# Afficher les premières lignes du DataFrame mis à jour
print(NourritureDispo.head())
```

```
             Zone  Population  Nbre_cal_journaliere_par_personne_disponibl
es  \
0     Afghanistan    36296113                                          208
7.0
1  Afrique du Sud    57009756                                          302
0.0
2         Albanie     2884169                                          318
8.0
3         Algérie    41389189                                          329
3.0
4       Allemagne    82658409                                          350
3.0

   Nbre_cal_dispo_totale
0           7.574999e+10
1           1.721695e+11
2           9.194731e+09
3           1.362946e+11
4           2.895524e+11
```

In [ ]:
```python
# Nombre moyen de calories nécessaires par personne par jour (valeur fict
calories_necessaires_par_personne = 2350  # en calories

# Calculer le nombre d'humains pouvant être nourris
NourritureDispo['Nb_humains_nourris'] = NourritureDispo['Nbre_cal_dispo_t

# Calculer le nombre total d'humains nourris
nombre_total_humains_nourris = NourritureDispo['Nb_humains_nourris'].sum(

print("Le nombre total d'humains pouvant être nourris est d'environ : {:.
```

```
Le nombre total d'humains pouvant être nourris est d'environ : 8.90 millia
rds d'humains
```

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

In [ ]:
```
#Transfert des données avec les végétaux dans un nouveau dataframe
dispoAlimentaire_vegetaux = dispoAlimentaire[dispoAlimentaire['Origine']
print(dispoAlimentaire_vegetaux.head())
```

```
          Zone                Produit   Origine  Aliments pour animaux  \
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
6  Afghanistan                  Bière  vegetale                    0.0

   Autres Utilisations  Disponibilité alimentaire (Kcal/personne/jour)  \
1                  0.0                                             1.0
2                  0.0                                             1.0
3                  0.0                                             0.0
4                  0.0                                             4.0
6                  0.0                                             0.0

   Disponibilité alimentaire en quantité (kg/personne/an)  \
1                                               1.29
2                                               0.06
3                                               0.00
4                                               2.70
6                                               0.09

   Disponibilité de matière grasse en quantité (g/personne/jour)  \
1                                               0.01
2                                               0.01
3                                               0.00
4                                               0.02
6                                               0.00

   Disponibilité de protéines en quantité (g/personne/jour)  \
1                                               0.02
2                                               0.03
3                                               0.00
4                                               0.05
6                                               0.00

   Disponibilité intérieure  ...  Importations - Quantité  Nourriture  Per
tes  \
1                   41000.0  ...                  40000.0     39000.0  200
0.0
2                    2000.0  ...                   2000.0      2000.0
0.0
3                       0.0  ...                      0.0         0.0
0.0
4                   82000.0  ...                  82000.0     82000.0
0.0
6                    3000.0  ...                   3000.0      3000.0
0.0

   Production  Semences  Traitement  Variation de stock  Année  Population
\
1      3000.0       0.0         0.0                 0.0   2017    36296113
2         0.0       0.0         0.0                 0.0   2017    36296113
3         0.0       0.0         0.0                 0.0   2017    36296113
4         0.0       0.0         0.0                 0.0   2017    36296113
6         0.0       0.0         0.0                 0.0   2017    36296113

   Nbre_cal_journaliere_par_personne_disponibles
1                                         2087.0
2                                         2087.0
3                                         2087.0
```

```
4                                                2087.0
6                                                2087.0

[5 rows x 21 columns]
```

In [ ]:
```
#Calcul du nombre de kcal disponible pour les végétaux

# Supprimer la colonne 'Nbre_cal_journaliere_par_personne_disponibles' du
dispoAlimentaire_vegetaux = dispoAlimentaire_vegetaux.drop('Nbre_cal_jour

#calculer le Nbre_cal_vege_journaliere_par_personne_disponibles pour les
Nbre_cal_vege_journaliere_par_personne_disponibles = dispoAlimentaire_veg
Nbre_cal_vege_journaliere_par_personne_disponibles.rename(columns={'Dispo

#Ajouter cette colonne au DataFrame dispoAlimentaire_vegetaux
dispoAlimentaire_vegetaux = pd.merge(dispoAlimentaire_vegetaux, Nbre_cal_
'''outer join pour prendre en compte l'entiereté des données de dispoAlim

#Grouper par pays et obtenir la première ligne de chaque groupe
NourritureDispo_vegetaux = dispoAlimentaire_vegetaux.groupby('Zone').firs
print(NourritureDispo_vegetaux.head())
```

```
             Zone  Population  \
0     Afghanistan    36296113
1  Afrique du Sud    57009756
2         Albanie     2884169
3         Algérie    41389189
4       Allemagne    82658409

   Nbre_cal_vege_journaliere_par_personne_disponibles
0                                             1871.0
1                                             2533.0
2                                             2203.0
3                                             2915.0
4                                             2461.0
```

In [ ]:
```
#Calculer la colonne Nbre_cal_vege_dispo_totale
NourritureDispo_vegetaux['Nbre_cal_vege_dispo_totale'] = NourritureDispo_

#Calculer le nombre d'humains pouvant être nourris avec les produits végé
NourritureDispo_vegetaux['Nb_humains_nourris_vegetaux'] = NourritureDispo

#Calculer le nombre total d'humains nourris avec les produits végétaux
nombre_total_humains_nourris_vegetaux = NourritureDispo_vegetaux['Nb_huma
print("Le nombre total d'humains pouvant être nourris avec les produits v
```

```
Le nombre total d'humains pouvant être nourris avec les produits végétaux
est d'environ : 7.35 milliards d'humains
```

3.4 - Utilisation de la disponibilité intérieure

In [ ]:
```
#Calcul de la disponibilité totale
dispo_int = dispoAlimentaire['Disponibilité intérieure'].sum()
print("La disponibilité intérieure mondiale est de : {:.2f} kilos".format
print("La disponibilité intérieure mondiale est de : {:.2f} tonnes".forma
```

```
La disponibilité intérieure mondiale est de : 9733927000.00 kilos
La disponibilité intérieure mondiale est de : 9733927.00 tonnes
```

In [ ]:
```
#création d'une boucle for pour afficher les différentes valeurs en fonct
colonnes = ['Aliments pour animaux', 'Pertes', 'Nourriture']
for colonne in colonnes:
    dispo_int_colonne = dispoAlimentaire[colonne].sum()
    print("La disponibilité intérieure mondiale en {} est de : {:.2f} kil
```

```
La disponibilité intérieure mondiale en Aliments pour animaux est de : 128
8002000.00 kilos
La disponibilité intérieure mondiale en Pertes est de : 452283000.00 kilos
La disponibilité intérieure mondiale en Nourriture est de : 4805525000.00
kilos
```

3.5 - Utilisation des céréales

In [ ]:
```
#Création d'une liste avec toutes les variables du fichier DispoAlimentai
variables = ['Autres Utilisations',
             'Aliments pour animaux',
             'Nourriture',
             'Pertes',
             'Semences',
             'Traitement',
             'Exportations - Quantité',
             'Importations - Quantité',
             'Production',
             'Variation de stock',
             'Disponibilité intérieure'
             ]
```

In [ ]:
```
#création d'un dataframe qui filtre sur toutes les céréales ['Blé', 'Riz
cereales = dispoAlimentaire[dispoAlimentaire['Produit'].isin(['Blé', 'Riz
```

In [ ]:
```
#Calcul de la quantité totale de céréales utilisée pour les animaux
quantite_totale_cereales_animaux = cereales['Aliments pour animaux'].sum(
print("La quantité totale de céréales utilisée pour les animaux est de :

# Calculer la proportion totale de 'Aliments pour animaux' par rapport à
quantite_totale_dispo_int = cereales['Disponibilité intérieure'].sum()
print("La quantité totale de céréales pour la disponibilité intérieure es

# Calculer la proportion totale de 'Aliments pour animaux' par rapport à
proportion_aliments_animaux = quantite_totale_cereales_animaux/ quantite_
print("La proportion totale d'Aliments pour animaux par rapport à la disp
```

```
La quantité totale de céréales utilisée pour les animaux est de : 85961500
0.00 kilos
La quantité totale de céréales pour la disponibilité intérieure est de : 2
378371000.00 kilos
La proportion totale d'Aliments pour animaux par rapport à la disponibilit
é intérieure totale est de : 36.14%
```

In [ ]:
```
quantite_totale_cereales_humains = cereales['Nourriture'].sum()
print("La quantité totale de céréales utilisée pour les Humains est de :

# Calculer la proportion totale de 'Nourriture'humaine par rapport à la d
proportion_nourriture = cereales['Nourriture'].sum() / cereales['Disponib
print("La proportion totale de Nourriture humaine par rapport à la dispon
```

```
La quantité totale de céréales utilisée pour les Humains est de : 1020464000.00 kilos
La proportion totale de Nourriture humaine par rapport à la disponibilité intérieure totale est de : 42.91%
```

In [ ]:
```python
# Calculer la proportion totale par variable par rapport à la disponibili
proportions = cereales[variables].sum() / cereales['Disponibilité intérie
print(proportions)
```

```
Autres Utilisations           9.782830
Aliments pour animaux        36.143016
Nourriture                   42.906006
Pertes                        4.486516
Semences                      2.847285
Traitement                    3.858986
Exportations - Quantité      17.636483
Importations - Quantité      16.461814
Production                  104.934974
Variation de stock           -3.760053
Disponibilité intérieure    100.000000
dtype: float64
```

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

In [ ]:
```python
#trouver le type de données de chaque colonne du dataFrame resultats_2017
print(resultats_2017.dtypes)

#modifier le type de données proportion_sous_nutrition en float
resultats_2017['proportion_sous_nutrition'] = resultats_2017['proportion_

# Trier le DataFrame par proportion_sous_nutrition de manière décroissant
resultats_tries = resultats_2017.sort_values(by='proportion_sous_nutritio

# Afficher les 10 premières lignes du DataFrame trié
print(resultats_tries[['Zone', 'proportion_sous_nutrition']].head(10))
```

```
Zone                         object
Population                    int64
sous_nutrition                int64
proportion_sous_nutrition    object
dtype: object
                                          Zone  proportion_sous_nutrition
72                                       Haïti                      49.17
151  République populaire démocratique de Corée                      47.98
99                                  Madagascar                      43.02
174                                      Tchad                      40.62
95                                     Libéria                      38.28
143                                     Rwanda                      36.72
112                                  Mozambique                      33.51
91                                      Lesotho                      33.47
177                                Timor-Leste                      32.17
189      Venezuela (République bolivarienne du)                      30.95
```

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

depuis 2013

In [ ]:
```
#création d'une table pivot pour avoir le montant total de l'aide aliment
pivot_aideAlimentaire = aideAlimentaire.pivot_table(index='Zone', columns
pivot_aideAlimentaire['Valeur Totale'] = pivot_aideAlimentaire.sum(axis=1

pivot_aideAlimentaire.reset_index(inplace=True)
print('LA TABLE PIVOT : ')
print (pivot_aideAlimentaire.head())
```

```
LA TABLE PIVOT :
Année          Zone       2013       2014      2015     2016  Valeur Totale
0       Afghanistan  128238000   57214000         0        0      185452000
1           Algérie   35234000   18980000  17424000  9476000       81114000
2            Angola    5000000      14000         0        0        5014000
3        Bangladesh  131018000  194628000  22542000        0      348188000
4           Bhoutan    1724000     146000    578000   218000        2666000
```

In [ ]:
```
# Tri décroissant par la colonne 'Valeur Totale'
pivot_aideAlimentaire_sorted = pivot_aideAlimentaire.sort_values(by='Vale
top_10 = pivot_aideAlimentaire_sorted.head(10)
print('Les 10 premiers pays selon la valeur totale de l\'aide alimentaire
print(top_10)
```

```
Les 10 premiers pays selon la valeur totale de l'aide alimentaire :
Année                                Zone       2013       2014       2015
\
50             République arabe syrienne  563566000  651870000  524949000
75                               Éthiopie  591404000  586624000  203266000
70                                  Yémen  264764000  103840000  372306000
61                            Soudan du Sud  196330000  450610000   48308000
60                                  Soudan  330230000  321904000   17650000
30                                   Kenya  220966000  217418000  114452000
3                                Bangladesh  131018000  194628000   22542000
59                                  Somalie  139800000   81180000   71698000
53     République démocratique du Congo  150320000   70134000   68048000
43                                   Niger   62720000   66226000   54656000

Année       2016  Valeur Totale
50     118558000     1858943000
75             0     1381294000
70     465574000     1206484000
61             0      695248000
60             0      669784000
30             0      552836000
3              0      348188000
59             0      292678000
53             0      288502000
43      92742000      276344000
```

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

In [ ]:
```
#calcul des 5 pays qui ont reçu le plus d'aide alimentaire entre 2013 et
top_5 = pivot_aideAlimentaire_sorted.head(5)
```

```
print('Les 5 premiers pays selon la valeur totale de l\'aide alimentaire
print(top_5)
```

```
Les 5 premiers pays selon la valeur totale de l'aide alimentaire :
Année                      Zone       2013       2014       2015       20
16  \
50     République arabe syrienne  563566000  651870000  524949000  1185580
00
75                       Éthiopie  591404000  586624000  203266000
0
70                          Yémen  264764000  103840000  372306000  4655740
00
61                  Soudan du Sud  196330000  450610000   48308000
0
60                         Soudan  330230000  321904000   17650000
0

Année  Valeur Totale
50        1858943000
75        1381294000
70        1206484000
61         695248000
60         669784000
```

3.9 - Pays avec le moins de disponibilité par habitant

In [ ]:
```
# Calculer la somme totale de Disponibilité alimentaire en quantité par p
dispo_total_par_pays = dispoAlimentaire.groupby('Zone')['Disponibilité al

# Trier les pays par ordre croissant selon leur Disponibilité alimentaire
pays_dispo_faible = dispo_total_par_pays.sort_values(by='Disponibilité al
print("Les 10 pays avec la disponibilité alimentaire la plus faible sont
print(pays_dispo_faible.head(10))
```

```
Les 10 pays avec la disponibilité alimentaire la plus faible sont :
                                          Zone  \
127                  République centrafricaine
164                                     Zambie
91                                   Madagascar
0                                    Afghanistan
65                                         Haïti
132  République populaire démocratique de Corée
150                                       Tchad
165                                    Zimbabwe
114                                     Ouganda
152                                Timor-Leste

     Disponibilité alimentaire (Kcal/personne/jour)
127                                          1879.0
164                                          1924.0
91                                           2056.0
0                                            2087.0
65                                           2089.0
132                                          2093.0
150                                          2109.0
165                                          2113.0
114                                          2126.0
152                                          2129.0
```

3.10 - Pays avec le plus de disponibilité par habitant

In [ ]:
```
# Trier les pays par ordre décroissant selon la Disponibilité alimentaire
pays_dispo_elevee = dispo_total_par_pays.sort_values(by='Disponibilité al

# Afficher les 10 premiers pays ayant la Disponibilité alimentaire en qua
print("Les 10 pays ayant la Disponibilité alimentaire en quantité la plus
print(pays_dispo_elevee.head(10))
```

```
Les 10 pays ayant la Disponibilité alimentaire en quantité la plus haute s
ont :
                      Zone  Disponibilité alimentaire (Kcal/personne/jour)
11                 Autriche                                          3770.0
16                 Belgique                                          3737.0
157                 Turquie                                          3708.0
169  États-Unis d'Amérique                                          3682.0
74                   Israël                                          3610.0
72                  Irlande                                          3602.0
75                   Italie                                          3578.0
89                Luxembourg                                          3540.0
166                   Égypte                                          3518.0
4                  Allemagne                                          3503.0
```

3.11 - Exemple de la Thaïlande pour le Manioc

In [ ]:
```
#création d'un dataframe avec uniquement la Thaïlande
# Filtrer les résultats pour ne conserver que les lignes avec l'année 201
resultats_thailande_2017 = resultats_2017[resultats_2017['Zone'] == 'Thaï
print(resultats_thailande_2017)
```

```
          Zone  Population  sous_nutrition  proportion_sous_nutrition
176  Thaïlande    69209810         6500000                       9.39
```

In [ ]:
```
# Calculer le nombre de personnes en état de sous-nutrition en Thaïlande
sous_nutrition_thailande_2017 = resultats_thailande_2017['sous_nutrition'
print("Le nombre de personnes en état de sous-nutrition en Thaïlande en 2

# Calculer le nombre de personnes dans la population en Thaïlande en 2017
population_thailande_2017 = population[(population['Zone'] == 'Thaïlande'
print("Le nombre de personnes dans la population en Thaïlande en 2017 est

# Calculer le pourcentage de sous-nutrition en Thaïlande en 2017
pourcentage_sous_nutrition_thailande = (sous_nutrition_thailande_2017 / p
print(f"Le pourcentage de sous-nutrition en Thaïlande en 2017 est de : {p
```

```
Le nombre de personnes en état de sous-nutrition en Thaïlande en 2017 est
de : 6,500,000 personnes
Le nombre de personnes dans la population en Thaïlande en 2017 est de : 6
9,209,810 personnes
Le pourcentage de sous-nutrition en Thaïlande en 2017 est de : 9.39%
```

In [ ]:
```
# Filtrer les données pour ne conserver que celles concernant le Manioc e
manioc_thailande_data = dispoAlimentaire[(dispoAlimentaire['Zone'] == 'Th

# Calculer la quantité de Manioc Thaïlandais exporté
```

```
exportations_manioc_thailande = manioc_thailande_data['Exportations - Qua
print(f"La quantité de Manioc exportée par la Thaïlande est de : {exporta


#Calculer la production de Manioc en Thaïlande
production_manioc_thailande = manioc_thailande_data['Production'].sum()
print(f"La production de Manioc en Thaïlande est de : {production_manioc_

# Calculer la proportion d'exportations par rapport à la production
proportion_export_production = exportations_manioc_thailande / production

print(f"Proportion d'exportations de Manioc par rapport à la production :
```

```
La quantité de Manioc exportée par la Thaïlande est de : 25214000.00 kilos
La production de Manioc en Thaïlande est de : 30228000.00 kilos
Proportion d'exportations de Manioc par rapport à la production : 83.41%
```

In [ ]:
```
#Quelle est la disponibilité alimentaire en manioc pour la Thaïlande

# Calculer la disponibilité par personne de manioc en Thaïlande (kg)
dispo_totale_manioc_thailande = manioc_thailande_data['Disponibilité alim
print(f"La disponibilité alimentaire en manioc (kg/personne/an) en Thaïla
```

```
La disponibilité alimentaire en manioc (kg/personne/an) en Thaïlande est d
e : 13.00 kg
```

Les exportations de manioc vers la nouvelle Zélande progressent fortement (400% en janvier 2020) Cette augmentation est due à la demande croissante en alimentation animale dans les fermes laitières (source https://asian-agribiz.com/2020/04/13/thailands-cassava-exports-to-new-zealand-jumps-400/)

Etape 6 - CONCLUSION

Causes principales de sous-nutrition dans le monde

Troubles politiques / corruption
Guerres (Tchad ...)
Fermeture des frontières (Corée du Nord)
Catastrophes naturelles (Haïti..)
Manque d'infrastructures
Education...

Réponses possibles

1- Intégrer des protéines végétales dans nos assiettes pour réduire la production destinée aux animaux d'élevage
2 - Utiliser plus efficacement les terres agricoles et les ressources d'eau
3 - Obtenir plus de financement pour des organismes comme le PAM
4 - Réduire les pertes alimentaires (4,6% de la dispo alimentaire mondiale)

À l'aube du XXIe siècle, la sécurité alimentaire mondiale représente un défi de taille alors que nous faisons face à une population mondiale croissante et à des ressources

limitées. Cependant, une lueur d'espoir émerge à travers les récentes recherches et les pratiques durables : en repensant nos habitudes alimentaires et en adoptant une approche plus équilibrée et durable, nous pourrions non seulement /

- améliorer notre santé

mais aussi

- contribuer à nourrir une population mondiale en pleine expansion.

L'une des pistes les plus prometteuses réside dans la transition vers une alimentation plus végétale et l'intégration de protéines végétales dans notre régime alimentaire. Ce changement pourrait avoir un impact significatif sur la quantité de nourriture nécessaire pour subvenir à nos besoins alimentaires :

- En réduisant la part des produits destinés à l'élevage animal et
- en accordant une place plus importante aux végétaux dans nos assiettes,

nous pourrions libérer des ressources précieuses et contribuer à une utilisation plus efficace des terres agricoles et des ressources en eau.

Sources :

Rapports des Nations Unies et de la FAO :
Rapport de la FAO : "The State of Food and Agriculture" - Ce rapport explore les tendances mondiales de la sécurité alimentaire et met en lumière l'importance des choix alimentaires durables pour l'avenir de la sécurité alimentaire mondiale.

Rapport du Panel international des ressources : "Creating a Sustainable Food Future" - Ce rapport examine les défis et les opportunités pour nourrir une population mondiale croissante de manière durable.

The World Resources Institute (WRI) : Le WRI publie des recherches et des analyses approfondies sur les systèmes alimentaires durables et l'impact des choix alimentaires sur l'environnement.

The EAT-Lancet Commission : Cette commission a élaboré un régime alimentaire sain pour les personnes et la planète, appelé le régime de référence de la Commission EAT-Lancet, qui met en avant une alimentation riche en végétaux.