
Étude :

la sous-nutrition dans le monde

« aider à construire un monde libéré de la faim »



**Food and Agriculture
Organization of the
United Nations**

sommaire

- ❖ La sous-nutrition dans le monde
- ❖ La disponibilité alimentaire
- ❖ L'utilisation de la nourriture et aide alimentaire
- ❖ Explication des résultats



La population mondiale en 2017 est de : **7,548 milliards d'habitants**

source : df_population2017

L'année **2017**

536,7 millions d'habitants recensées en sous-nutrition selon la FAO.

1,047 milliards d'habitants selon les estimations

(prise en compte des valeurs manquantes et des différences de zonages entre les dataframes)



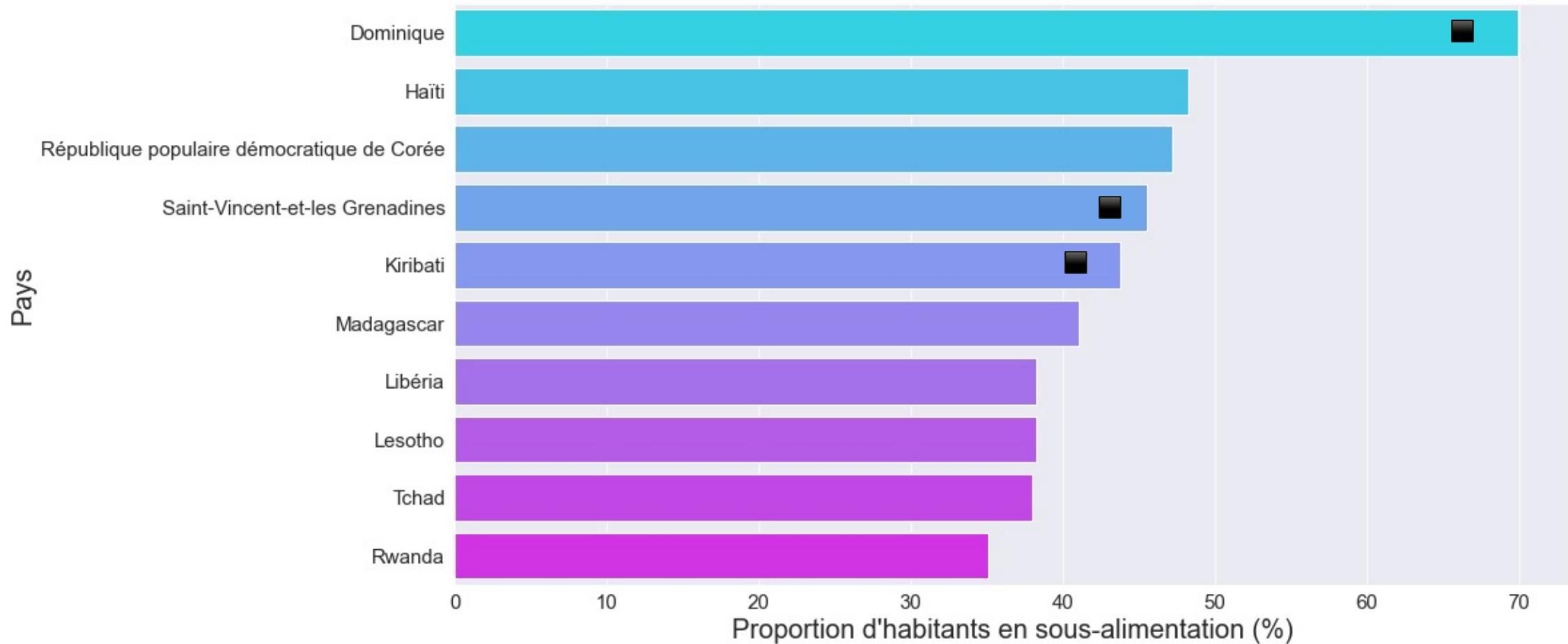
Population recensée



Population estimée

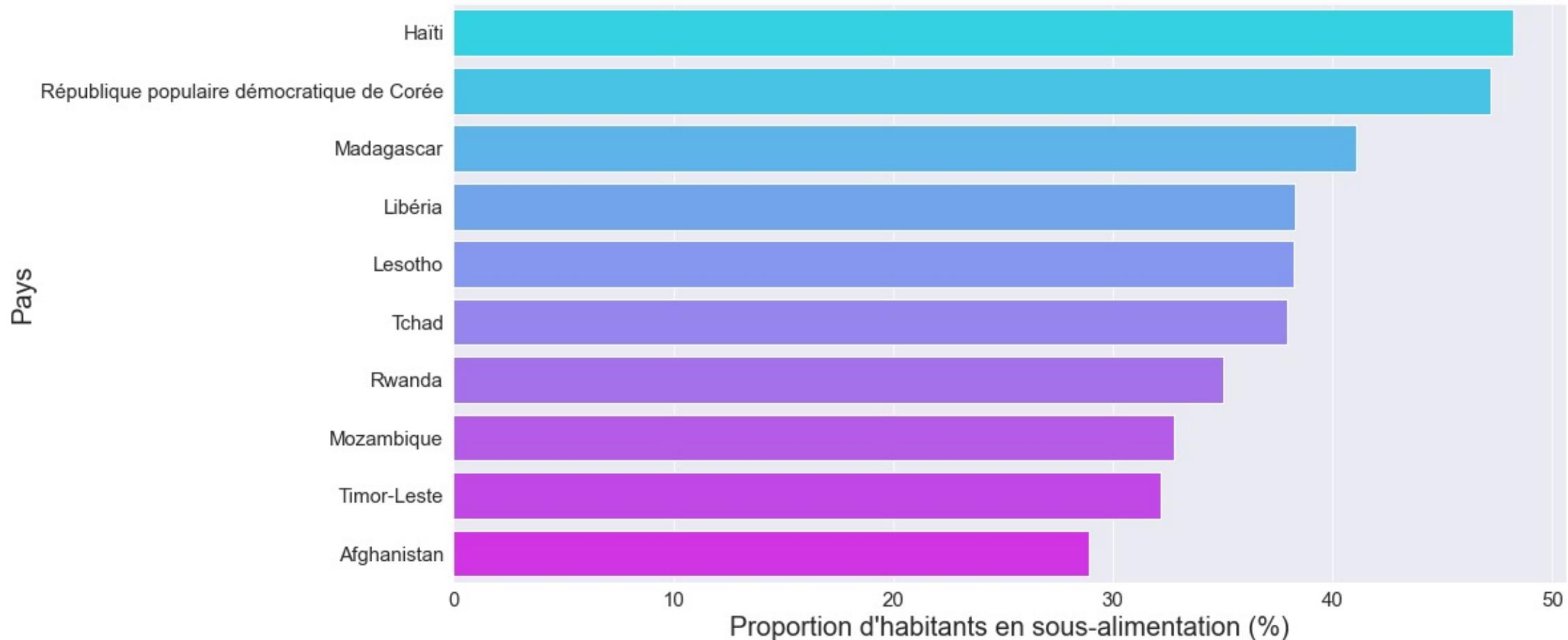
Pays ayant la **proportion** la plus importante d'habitants en **sous-alimentation**

■ : changement de la variable population sous-nutrition <0.1 par 0.05



Pays ayant la **proportion** la plus importante d'habitants en **sous-alimentation**

(sans modification de la variable population sous-nutrition)



sommaire

- ❖ La sous-nutrition dans le monde
- ❖ La disponibilité alimentaire
- ❖ L'utilisation de la nourriture et aide alimentaire
- ❖ Explication des résultats

Disponibilité alimentaire mondiale

Jusqu'à 113 % de la population mondiale
qui pourrait être nourrie (2017)

cad : 8,5 milliards d'individus

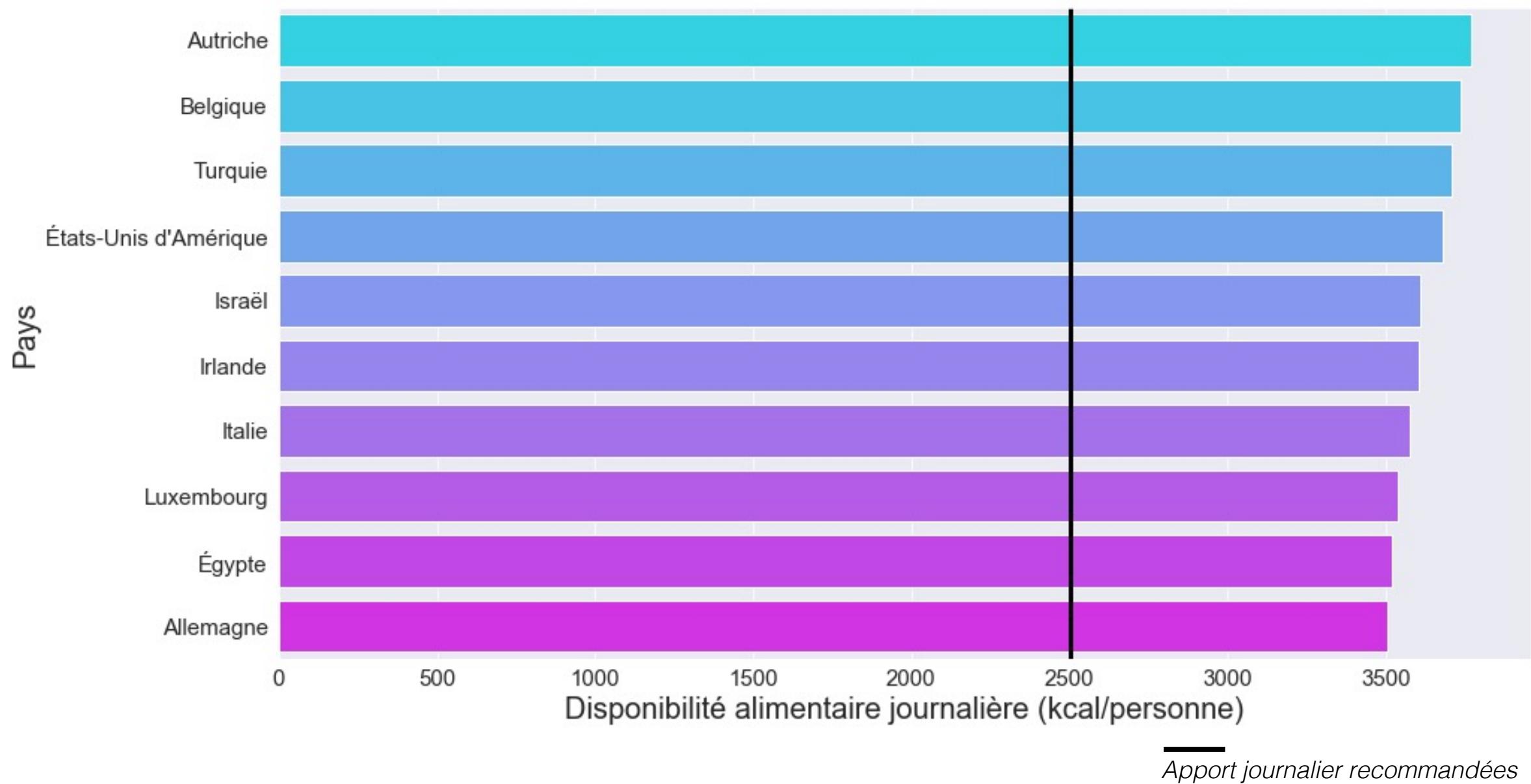
Disponibilité alimentaire mondiale

produits végétaux

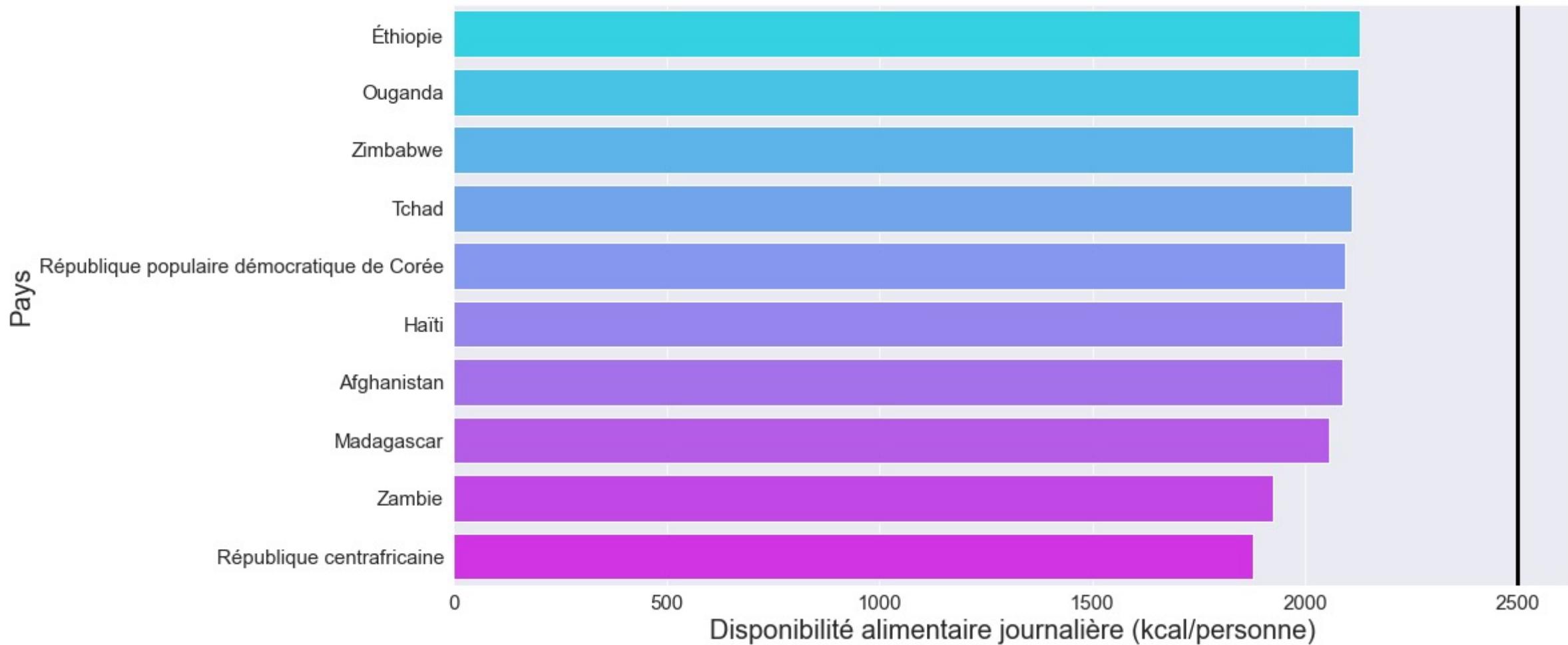
Jusqu'à 91 % de la population mondiale en 2017
pourrait être nourrie avec des produits végétaux

cad : 6,9 milliards d'individus

Disponibilité alimentaire journalière (top 10)



Disponibilité alimentaire journalière (bottom 10)



Apport journalier recommandées

Utilisation **disponibilité intérieure**

Rappel du calcul :

Production + Importations + Variations de stock - Exportations

=

Disponibilité intérieure

=

Nourriture + Aliments pour Animaux + Semences + Traitement + Autres utilisations + Pertes



Perte alimentaire

Alimentation animale

Alimentation humaine

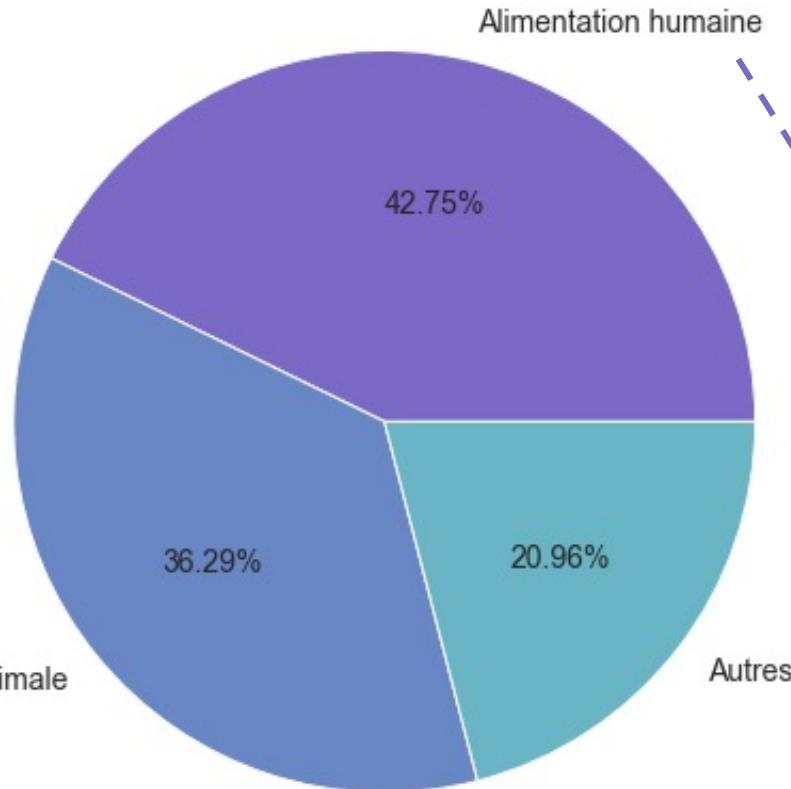
source : df_dispoAlimentaire

sommaire

- ❖ La sous-nutrition dans le monde
- ❖ La disponibilité alimentaire
- ❖ L'utilisation de la nourriture et aide alimentaire
- ❖ Explication des résultats

Le cas des céréales

Répartition de l'utilisation des céréales selon la disponibilité intérieure mondiale



Les céréales représentent **67%**
de l'alimentation animale

Les céréales représentent **21%**
de l'alimentation humaine

La Thaïlande et le manioc

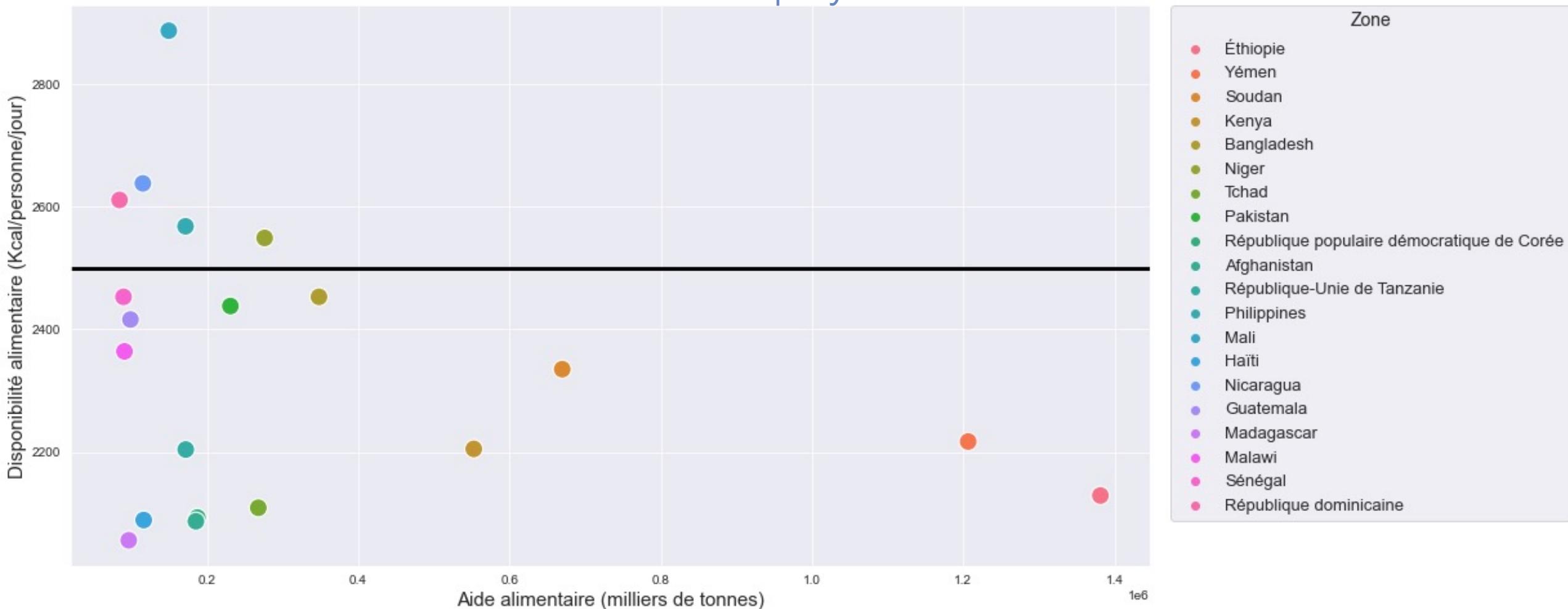
9% de la population en sous-alimentation

83% d'exportation du manioc à l'internationale



70% de l'exportation du manioc mondiale

Bonus : Aide alimentaire en fonction de la disponibilité alimentaire du pays



Apport journalier recommandées

L'aide alimentaire

Comment est alloué l'aide alimentaire dans le monde?

- Pas de relation selon la disponibilité alimentaire du pays;

Piste d'amélioration:

Avoir une redistribution de l'aide alimentaire selon la disponibilité alimentaire du pays.

Certains pays ont des disponibilité alimentaire suffisante et reçoivent de l'aide alimentaire

Merci pour votre attention !

sommaire

- ❖ La sous-nutrition dans le monde
- ❖ La disponibilité alimentaire
- ❖ L'utilisation de la nourriture et aide alimentaire
- ❖ Explication des résultats

La population mondiale

```
# Calcul de la population mondiale en 2017
df_population2017 = df_population[df_population['Année'] == 2017]
population2017 = df_population2017['Valeur'].sum()

print("La population mondiale en 2017 est de {:.2f} millions habitants environ".format(population2017/1000).replace('
```

```
La population mondiale en 2017 est de 7 548.13 millions habitants environ
```

	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
...
1411	Zimbabwe	2014	13586.707
1412	Zimbabwe	2015	13814.629
1413	Zimbabwe	2016	14030.331
1414	Zimbabwe	2017	14236.595
1415	Zimbabwe	2018	14438.802

La sous-nutrition dans le monde

df_sousNutrition

	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

df_sousNutrition.describe

	Zone	Année	Valeur
count	1218	1218	624
unique	203	6	139
top	Iraq	2016-2018	<0.1
freq	6	203	120



Modification 'valeur' en float

```
df_sousNutrition.loc[df_sousNutrition['Valeur'] == '<0.1', 'Valeur'] = 0.05
# Nous n'avons pas mis 0 car <0.1 ne veut pas dire qu'il n'y a aucune personne en état de sous-nutrition.
```

```
: # Conversion des valeurs en numérique (float)
df_sousNutrition['Valeur'] = pd.to_numeric(df_sousNutrition['Valeur'])
```

La sous-nutrition dans le monde

```
: # Sélection de l'année 2017
sousNutrition2017 = df_sousNutrition.loc[df_sousNutrition['Année'] == '2016-2018']

# Calcul de pour savoir s'il nous manque des données
print(sousNutrition2017.isna().mean()*100) # % de données manquantes en moyenne
print()
print("Il y a", '{:1.2%}'.format(sousNutrition2017.iloc[:,2].isna().mean()), 'de données \'NaN\' pour notre variable \\'Valeur\'')
```

```
Zone      0.000000
Année     0.000000
Valeur    48.768473
dtype: float64
```

Il y a 48.77% de données 'NaN' pour notre variable 'Valeur' dans l'année 2017.

Attention : Les données NaN ne veulent pas dire qu'il manque des données. Cela peut aussi signifier que le pays ne recense aucune personne en état de sous-nutrition.

Nous allons calculer le nombre de personnes en sous-nutrition durant l'année 2017 pour les pays dont **cette valeur est connue**.

Par la suite, nous pourrions faire une estimation de la population globale en effectuant ce calcul :

```
= totalSousNutrition / (1 - [pourcentage de données NaN dans notre variable 'Valeur'])

: # Population en sous-nutrition (recensée)
totalSousNutrition = sousNutrition2017['Valeur'].sum()
print('La population recensée selon nos données en sous-nutrition en 2017 serait de', '{:,}'.format(totalSousNutrition))

La population recensée selon nos données en sous-nutrition en 2017 serait de 536.7 millions d'habitants.

: # Estimation de la population en sous-nutrition
totalEstimSousNutrition = totalSousNutrition/(1-(sousNutrition2017['Valeur'].isna().mean()))
print('La population estimée en sous-nutrition en 2017 serait de', '{:.2f}'.format(totalEstimSousNutrition).replace(',', ','))

La population estimée en sous-nutrition en 2017 serait de 1047.60 millions d'habitants.
```

Sélection de l'année (rappel : analyse de l'année 2017)

La disponibilité alimentaire mondiale

df_dispoAlimentaire

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
0	Afghanistan	Abats Comestible	animale	NaN	NaN	5.0	1.72	0.20	0.77	53.0
1	Afghanistan	Agrumes, Autres	vegetale	NaN	NaN	1.0	1.29	0.01	0.02	41.0
2	Afghanistan	Aliments pour enfants	vegetale	NaN	NaN	1.0	0.06	0.01	0.03	2.0

Calcul de la disponibilité alimentaire journalière disponible en kcal/personne

```
[4]: # Calcul de la disponibilité alimentaire journalière disponible en kcal/personne
dispoAlimentaireJour = df_dispoAlimentaire['Disponibilité alimentaire (Kcal/personne/jour)'].sum()
dispoAlim_nbZone = dispoAlimentaireJour/nb_zone
print('La disponibilité alimentaire journalière disponible est de {:.0f} kcal/personne dans le monde.'.format(dispoA
```

La disponibilité alimentaire journalière disponible est de 2 847 kcal/personne dans le monde.

Nombre de personnes pouvant être nourri dans le monde selon la disponibilité alimentaire journalière disponible en kcal/personne

```
]: # Calcul du nombres de personnes pouvant être nourri dans le monde
ajr = 2500

nbPersonneNourri = (dispoAlim_nbZone/ajr)*(population2017/1000)
print('D\'après nos calculs, environ', '{:.2f}'.format(nbPersonneNourri).replace(',', '.'), 'millions de la population')
print()
print("Cela représente environ {:.2%} de la population mondiale".format((nbPersonneNourri/(population2017/1000))))
```

D'après nos calculs, environ 8 596.93 millions de la population mondiale pourraient être nourri selon la disponibilité alimentaire journalière (Kcal/personne/jour).

Cela représente environ 113.89% de la population mondiale

La disponibilité alimentaire mondiale des produits végétaux

df_vegetaux

```
# Crédit au dataframe comportant que les végétaux en origine  
df_vegetaux = df_dispoAlimentaire[df_dispoAlimentaire['Origine'] == 'végétale']  
df_vegetaux.head()
```

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	
1	Afghanistan	Agrumes, Autres	végétale	NaN	NaN	1.0	1.29	0.01	0.02	41.0	2.0
2	Afghanistan	Aliments pour	végétale	NaN	NaN	1.0	0.06	0.01	0.03	2.0	NaN

```
: # Calcul de la disponibilité alimentaire journalière pour une personne disponible grâce aux produits végétaux  
dispoVegetaux = (df_vegetaux['Disponibilité alimentaire (Kcal/personne/jour)'].sum()) / nb_zone  
print('La disponibilité alimentaire journalière disponible grâce aux produits végétaux est de {:.0f} kcal/personne dans le monde')
```

La disponibilité alimentaire journalière disponible grâce aux produits végétaux est de 2 292 kcal/personne dans le monde

```
: # Calcul du nombre de personnes pouvant être nourries dans le monde grâce aux produits végétaux  
nbPerNourriVegetaux = (dispoVegetaux / ajr) * (population2017 / 1000) # pour rappel, la population est exprimé en milliers  
print('D\'après nos calculs, environ {:.2f}'.format(nbPerNourriVegetaux).replace(',', '.'), 'millions de la population mondiale')  
print()  
print("Cela représente environ {:.2%} de la population mondiale".format((nbPerNourriVegetaux / (population2017 / 1000))))
```

D'après nos calculs, environ 6 919.68 millions de la population mondiale pourraient être nourri selon la disponibilité alimentaire journalière des produits végétaux (Kcal/personne/jour).

Cela représente environ 91.67% de la population mondiale

La disponibilité alimentaire mondiale des produits végétaux

Rappel du calcul :

Production + Importations + Variations de stock - Exportations

=

Disponibilité intérieure

=

Nourriture + Aliments pour Animaux + Semences + Traitement + Autres utilisations + Pertes

```
: # Calcul de la disponibilité intérieure mondiale
dispoIntérieure = int(df_dispoAlimentaire['Disponibilité intérieure'].sum())
print('La disponibilité intérieure mondiale est de {:,} milliers de tonnes.'.format(dispoIntérieure).replace(',', ','))
```

La disponibilité intérieure mondiale est de 9 848 994 milliers de tonnes.



```
: # Calcul de la part attribué à l'alimentation animale, alimentation humaine & la perte alimentaire
alimentationAnimale = int(df_dispoAlimentaire['Aliments pour animaux'].sum())
alimentationHumaine = int(df_dispoAlimentaire['Nourriture'].sum())
pertesAlimentaire = int(df_dispoAlimentaire['Pertes'].sum())
```



L'alimentation animale mondiale représente 1 304 245 de milliers de tonnes, cela équivaut à 13.24% de la disponibilité intérieure mondiale.

L'alimentation humaine mondiale représente 4 876 258 de milliers de tonnes, cela équivaut à 49.51% de la disponibilité intérieure mondiale.

La perte alimentaire mondiale représente 453 698 de milliers de tonnes, cela équivaut à 4.61% de la disponibilité intérieure mondiale.

Utilisation des céréales dans le monde

Organisation des Nations Unies pour l'alimentation et l'agriculture

À propos | En action | Pays | Thèmes | Média | العربية | 中文

FAOSTAT

Bilans Alimentaires (2014-)

DOWNLOAD DATA | VISUALIZE DATA | METADATA | REPORT

PAYS RÉGIONS GROUPES SPÉCIAUX FAO

Filtrer les résultats par ex. afghanistan

- Afghanistan
- Afrique du Sud
- Albanie
- Algérie
- Allemagne
- Angola

Tout sélectionner Tout effacer

ÉLÉMENTS

Filtrer les résultats par ex. population totale

- Population totale
- Production - Quantité
- Importations - Quantité
- Variation de stock
- Exportations - Quantité
- Disponibilité intérieure

Tout sélectionner Tout effacer

PRODUITS GROUPES DE PRODUITS

Filtrer les résultats par ex. population

- Produits Végétaux + (Total)
- Produits Végétaux > (Liste)
- Produits Animaux + (Total)
- Produits Animaux > (Liste)

Tout sélectionner Tout effacer

ANNÉES

Filtrer les résultats par ex. 2018

- 2018
- 2017
- 2016
- 2015

Tout sélectionner Tout effacer

df_cereales

	Code Domaine	Domaine	Code zone (FAO)	Zone	Code Élément	Code Produit	Code année	Année	Unité	Valeur	Symbole	Description du Symbole	
0	FBS	Bilans Alimentaires (2014-)	2	Afghanistan	5511	Production	2511	Blé et produits	2017	2017	Milliers de tonnes	4281.0	Im Données de la FAO basées sur une méthodologie d'imputation
1	FBS	Bilans Alimentaires (2014-)	2	Afghanistan	5611	Importations - Quantité	2511	Blé et produits	2017	2017	Milliers de tonnes	3080.0	Im Données de la FAO basées sur une méthodologie d'imputation
2	FBS	Bilans Alimentaires (2014-)	2	Afghanistan	5072	Variation de stock	2511	Blé et produits	2017	2017	Milliers de tonnes	32.0	Im Données de la FAO basées sur une méthodologie d'imputation

Utilisation des **céréales** dans le monde

```
: # Création de la liste contenant les différents types de céréales :
liste_cereales = pd.Series(df_cereales['Produit'].unique())
liste_cereales

0      Blé et produits
1      Riz et produits
2      Orge et produits
3      Mais et produits
4      Seigle et produits
5          Avoine
6      Millet et produits
7      Sorgho et produits
8      Céréales, Autres
dtype: object

: # Comparaison avec la liste actuelle du dataframe
liste_cereales_df = df_dispoAlimentaire['Produit'].unique()
liste_cereales_df

array(['Abats Comestible', 'Agrumes, Autres', 'Aliments pour enfants',
       'Ananas', 'Bananes', 'Beurre, Ghee', 'Bière', 'Blé',
       'Boissons Alcooliques', 'Café', 'Coco (Incl Coprah)', 'Crème',
       'Céréales, Autres', 'Dattes', 'Edulcorants Autres',
       'Fève de Cacao', 'Fruits, Autres', 'Graines de coton',
       'Graines de tournesol', 'Graisses Animales Crue',
       'Huile Plantes Oleif Autr', 'Huile Graines de Coton',
       'Huile d'Arachide", "Huile d'Olive", 'Huile de Colza&Moutarde',
       'Huile de Palme', 'Huile de Soja', 'Huile de Sésame',
       'Huile de Tournesol', 'Lait - Excl Beurre', 'Légumes, Autres',
       'Légumineuses Autres', 'Maïs', 'Miel', 'Millet', 'Miscellanees',
       'Noix', 'Oeufs', 'Olives', 'Oranges, Mandarines', 'Orge',
       'Plantes Oleiferes, Autre', 'Poissons Eau Douce', 'Poivre',
       'Pommes', 'Pommes de Terre', 'Raisin', 'Riz (Eq Blanchi)',
       'Sucre Eq Brut', 'Sucre, betterave', 'Sucre, canne', 'Sésame',
       'Thé', 'Tomates', "Viande d'Ovins/Caprins", 'Viande de Bovins',
       'Viande de Volailles', 'Viande, Autre', 'Vin', 'Épices, Autres',
       'Alcool, non Comestible', 'Animaux Aquatiques Autre',
       'Arachides Decortiquees', 'Avoine', 'Bananes plantains',
       'Boissons Fermentés', 'Cephalopodes', 'Citrons & Limes',
       'Crustacés', 'Girofles', 'Graines Colza/Moutarde', 'Haricots',
       'Huile de Coco', 'Huile de Germe de Maïs', 'Huile de Palmistes',
       'Huiles de Foie de Poisso', 'Huiles de Poissons', 'Ignames',
       'Manioc', 'Mollusques, Autres', 'Oignons', 'Palmistes',
       'Pamplemousse', 'Patates douces', 'Perciform', 'Piments',
       'Plantes Aquatiques', 'Pois', 'Poissons Marins, Autres',
       'Poissons Pelagiques', 'Racines nda', 'Seigle', 'Soja', 'Sorgho',
       'Viande de Suides', 'Huile de Son de Riz', 'Sucre non centrifugé',
       'Viande de Anim Aquatiq'], dtype=object)
```

Utilisation des **céréales** dans le monde

```
: # Modification de notre liste pour qu'elle correspond à notre df
liste_cereales2 = liste_cereales.replace(' et produits','',regex=True)
liste_cereales2 = liste_cereales2.replace('Riz','Riz (Eq Blanchi)',regex=True)
liste_cereales2
```

```
0          Blé
1    Riz (Eq Blanchi)
2          Orge
3          Maïs
4        Seigle
5      Avoine
6      Millet
7      Sorgho
8  Céréales, Autres
dtype: object
```

```
: # Crédit d'un nouveau dataframe + ajout colonne is_cereales
df_dispoAlimentaire2 = df_dispoAlimentaire.copy() #.copy() permet de ne pas intégrer avec le premier dataframe
df_dispoAlimentaire2['is_cereales'] = df_dispoAlimentaire2['Produit'].isin(liste_cereales2) # test booléen pour savoir si le produit est une céréale
df_dispoAlimentaire2[df_dispoAlimentaire2['is_cereales']==True]['Produit'].unique() # vérification de la liste du dataframe
```



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

```
: # Calcul du tonnes des aliments dans la liste céréales pour l'alimentation humaine et la nourriture animale
cereales_humaines = df_dispoAlimentaire2[df_dispoAlimentaire2['is_cereales']== True]['Nourriture'].sum()
cereales_animales = df_dispoAlimentaire2[df_dispoAlimentaire2['is_cereales']== True]['Aliments pour animaux'].sum()
```

Autre méthode pour calculer la quantité de céréales sans ajouter une colonne au dataframe:

```
cereales_humaines2 = df_dispoAlimentaire2[df_dispoAlimentaire2['Produit'].isin(liste_cereales2)]['Nourriture'].sum()
```

Utilisation des **céréales** dans le monde

```
: # Modification de notre liste pour qu'elle correspond à notre df
liste_cereales2 = liste_cereales.replace(' et produits','',regex=True)
liste_cereales2 = liste_cereales2.replace('Riz','Riz (Eq Blanchi)',regex=True)
liste_cereales2
```

```
0          Blé
1    Riz (Eq Blanchi)
2          Orge
3          Maïs
4        Seigle
5      Avoine
6      Millet
7      Sorgho
8  Céréales, Autres
dtype: object
```

```
: # Crédit d'un nouveau dataframe + ajout colonne is_cereales
df_dispoAlimentaire2 = df_dispoAlimentaire.copy() #.copy() permet de ne pas intégrer avec le premier dataframe
df_dispoAlimentaire2['is_cereales'] = df_dispoAlimentaire2['Produit'].isin(liste_cereales2) # test booléen pour savoir si le produit est une céréale
df_dispoAlimentaire2[df_dispoAlimentaire2['is_cereales']==True]['Produit'].unique() # vérification de la liste du dataframe
```

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

```
: # Calcul du tonnes des aliments dans la liste céréales pour l'alimentation humaine et la nourriture animale
cereales_humaines = df_dispoAlimentaire2[df_dispoAlimentaire2['is_cereales']== True]['Nourriture'].sum()
cereales_animales = df_dispoAlimentaire2[df_dispoAlimentaire2['is_cereales']== True]['Aliments pour animaux'].sum()
```

Autre méthode pour calculer la quantité de céréales sans ajouter une colonne au dataframe:

```
cereales_humaines2 = df_dispoAlimentaire2[df_dispoAlimentaire2['Produit'].isin(liste_cereales2)]['Nourriture'].sum()
```

Thaïlande et manioc

```
# Création du dataframe Thaïlande - Manioc
df_ThailandeManioc = df_dispoAlimentaire[(df_dispoAlimentaire['Zone']=='Thaïlande')\
                                         &(df_dispoAlimentaire['Produit']=='Manioc')]
exportThailandeManioc = df_ThailandeManioc['Exportations - Quantité'].sum()
df_ThailandeManioc
```

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 (g/personne/jour)	Exportations - Quantité	
13809	Thaïlande	Manioc	vegetale	1800.0	2081.0	40.0	13.0	0.05	0.14	6264.0	25214.0

Petit rappel sur la disponibilité intérieure :

Disponibilité intérieure = Production + Importations + Variations de stock (?) - Exportations

Exportation de manioc Population en sous-nutrition Part de l'exportation du manioc dans le monde

83.41 %

8.96 %

69.94%

Sous-alimentation : proportion de personnes sous-alimentées

Jointure entre les 2 dataframes

```
: # Population en sous-nutrition en 2017 pour chaque pays
df_sousNutrition2017=df_sousNutrition[df_sousNutrition['Année']=='2016-2018']

: # Jointure entre les deux dataframes :
j_sousNutritionPop2017 = pd.merge(
    df_population2017,df_sousNutrition2017, on='Zone')
j_sousNutritionPop2017 = j_sousNutritionPop2017.drop('Année_y', 1)
j_sousNutritionPop2017 = j_sousNutritionPop2017.rename(columns = {'Année_x':'Année',
                                                               "Valeur_x":"Population",
                                                               "Valeur_y":"Population sous-nutrition"
                                                               })
j_sousNutritionPop2017.head()
```

	Zone	Année	Population	Population sous-nutrition
0	Afghanistan	2017	36296.113	10.5
1	Afrique du Sud	2017	57009.756	3.1
2	Albanie	2017	2884.169	0.1
3	Algérie	2017	41389.189	1.3
4	Allemagne	2017	82658.409	NaN

Sous-alimentation : proportion de personnes sous-alimentées

Création d'une colonne proportion avant nous permettant de faire le classement

$$\text{Proportion} = \text{population sous-nutrition}/\text{population totale}$$

```
: # Top 10 : pays ayant la proportion de personnes sous-alimentées la plus importante en 2017
j_sousNutritionPop2017.nlargest(10,columns="proportion") # 1ère méthode

j_sousNutritionPop2017.sort_values('proportion',ascending=False).head(10) # 2ème méthode
```

	Zone	Année	Population	Population sous-nutrition	proportion
51	Dominique	2017	71.458	0.05	69.971172
78	Haiti	2017	10982.366	5.30	48.259182
157	République populaire démocratique de Corée	2017	25429.825	12.00	47.188685
164	Saint-Vincent-et-les Grenadines	2017	109.827	0.05	45.526146
98	Kiribati	2017	114.158	0.05	43.798945
108	Madagascar	2017	25570.512	10.50	41.062924
103	Libéria	2017	4702.226	1.80	38.279742
100	Lesotho	2017	2091.534	0.80	38.249438
183	Tchad	2017	15016.753	5.70	37.957606
161	Rwanda	2017	11980.961	4.20	35.055619

Sous-alimentation : proportion de personnes sous-alimentées

Si on ne veut pas de la modification '<0.1' en 0.05

```
: j_sousNutritionPop2017[j_sousNutritionPop2017['Population sous-nutrition']!=0.05].sort_values('proportion', ascending=
```

	Zone	Année	Population	Population sous-nutrition	proportion
78	Haiti	2017	10982.366	5.3	48.259182
157	République populaire démocratique de Corée	2017	25429.825	12.0	47.188685
108	Madagascar	2017	25570.512	10.5	41.062924
103	Libéria	2017	4702.226	1.8	38.279742
100	Lesotho	2017	2091.534	0.8	38.249438
183	Tchad	2017	15016.753	5.7	37.957606
161	Rwanda	2017	11980.961	4.2	35.055619
121	Mozambique	2017	28649.018	9.4	32.810898
186	Timor-Leste	2017	1243.258	0.4	32.173531
0	Afghanistan	2017	36296.113	10.5	28.928718

Aide alimentaire : depuis 2013

df_aideAlimentaire

	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

```
# Calcul du top 10 des pays les plus bénéficiaires de l'aide alimentaire :  
df_aideAlimentaire.groupby('Pays bénéficiaire').agg(  
    {'Valeur':np.sum}).sort_values('Valeur',ascending=False).head(10)
```

Pays bénéficiaire	Valeur
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

Fonction d'agrégation nous permettant d'avoir la somme
des valeurs pour chacun des pays.



Disponibilité alimentaire par habitant

Jointure avec fonction d'agrégation pour la somme de la disponibilité alimentaire pour chaque pays/jour

```
# Jointure entre le dataframe population2017 et celui de la disponibilité alimentaire pour chaque pays.  
df_dispoHabitant = pd.merge(df_population2017,  
                           df_dispoAlimentaire.groupby('Zone').agg({'Disponibilité alimentaire (Kcal/personne/jour)':  
df_dispoHabitant = df_dispoHabitant.rename(columns = {'Valeur':'Population'})  
df_dispoHabitant.head()
```

	Zone	Année	Population	Disponibilité alimentaire (Kcal/personne/jour)
0	Afghanistan	2017	36296.113	2087.0
1	Afrique du Sud	2017	57009.756	3020.0
2	Albanie	2017	2884.169	3188.0
3	Algérie	2017	41389.189	3293.0
4	Allemagne	2017	82658.409	3503.0

Pearson Correlation of Features : df_dispoAlimentaire

Graphique : coefficient de corrélation linéaire
(person correlation)

