

PRESENTATION

Mission FAO

SOMMAIRE

MISSION

BILANS ALIMENTAIRES FAO 2013

CHIFFRES CLES

CAUSES DE LA FAIM

PRODUCTION POUR LES ANIMAUX

ECONOMIE CONTRE POPULATION

MISSION

**FAO : Organisation des nations unies
pour l'alimentation et l'agriculture**

- **Enrayer la faim dans le monde**
- **Proposer des plans d'action**

Réaliser une analyse sur la faim dans le monde

- **Cibler les pays impactés**
- **Identifier les principales causes**

Utiliser une base de données

- **Source : FAO**

BILANS ALIMENTAIRES FAO 2013

Balance commerciale

**importations / exportations
productions**

Répartition de la disponibilité intérieure

**l'alimentation pour les animaux
l'alimentation pour les populations**

Disponibilité alimentaire

Energie (kcal) et protéines

CHIFFRES CLES

25000 personnes meurent de faim chaque jour

11 % de la population mondiale souffre de la faim

FAO : Statut quo des 11 % jusqu'en 2030

2013 : 842 millions

2050 : 1 Milliard

CAUSES DE LA FAIM

- **pauvreté du pays**
- **hausse des prix des produits alimentaires**
- **utilisation de la production pour l'économie**

PRODUCTION POUR LES ANIMAUX

- **45 % des céréales pour l'alimentation animale**
- **Pour les Etats-Unis :**
 - 10 % de viande produite en moins**
 - 13 Milliards de Tonnes de céréales en plus**
- **En 2013 : 7 milliards de personnes nourries**

ECONOMIE CONTRE POPULATION

Thaïlande

- **83 % du manioc produit est exporté**
- **5,6 Millions de personnes souffrant de la faim (8%)**
- **Les pouvoirs publics ne contrôlent pas les industries de production**

CONCLUSION

Pas de cause technique pour la faim

**10 Milliards en 2050 :
Meilleur usage de la production**

PARTIE TECHNIQUE

Mission FAO

SOMMAIRE

DONNEES TELECHARGEES

DONNEES TRAITEES

REQUETES SQL

ALIMENTER LA BASE DE DONNEES

DONNEES TELECHARGEES

- **Source** : FAO (<http://www.fao.org/faostat/fr/#data>)

- **5 fichiers csv :**

 - fr_animaux.csv**

 - fr_vegetaux.csv**

 - fr_population.csv**

 - fr_sousalimentation.csv**

 - fr_céréales.csv :**

 - récupérer les codes produits des céréales**

fr_animaux.csv + fr_vegetaux.csv

Code zone	Zone	Code Élément	Élément	Code Produit	Produit	Année	Unité	Valeur
2	Afghanistan	5511	Production	2731	Viande de Bovins	2013	Milliers de tonnes	134
2	Afghanistan	5611	Importations - Quantité	2731	Viande de Bovins	2013	Milliers de tonnes	6
2	Afghanistan	5301	Disponibilité intérieure	2731	Viande de Bovins	2013	Milliers de tonnes	140
2	Afghanistan	5142	Nourriture	2731	Viande de Bovins	2013	Milliers de tonnes	140
2	Afghanistan	645	Disponibilité alimentaire en quantité (kg/personne/an)	2731	Viande de Bovins	2013	kg	4.59
2	Afghanistan	664	Disponibilité alimentaire (Kcal/personne/jour)	2731	Viande de Bovins	2013	Kcal/personne/jour	27
2	Afghanistan	674	Disponibilité de protéines en quantité (g/personne/jour)	2731	Viande de Bovins	2013	g/personne/jour	1.89
2	Afghanistan	684	Disponibilité de matière grasse en quantité (g/personne/jour)	2731	Viande de Bovins	2013	g/personne/jour	2.1
2	Afghanistan	5511	Production	2732	Viande d'Ovins/Caprins	2013	Milliers de tonnes	150
2	Afghanistan	5301	Disponibilité intérieure	2732	Viande d'Ovins/Caprins	2013	Milliers de tonnes	150
2	Afghanistan	5142	Nourriture	2732	Viande d'Ovins/Caprins	2013	Milliers de tonnes	150
2	Afghanistan	645	Disponibilité alimentaire en quantité (kg/personne/an)	2732	Viande d'Ovins/Caprins	2013	kg	4.92
2	Afghanistan	664	Disponibilité alimentaire (Kcal/personne/jour)	2732	Viande d'Ovins/Caprins	2013	Kcal/personne/jour	31
2	Afghanistan	674	Disponibilité de protéines en quantité (g/personne/jour)	2732	Viande d'Ovins/Caprins	2013	g/personne/jour	1.91
2	Afghanistan	684	Disponibilité de matière grasse en quantité (g/personne/jour)	2732	Viande d'Ovins/Caprins	2013	g/personne/jour	2.55

Code zone	Zone	Code Élément	Élément	Code Produit	Produit	Année	Unité	Valeur
2	Afghanistan	5511	Production	2511	Blé	2013	Milliers de tonnes	5169
2	Afghanistan	5611	Importations - Quantité	2511	Blé	2013	Milliers de tonnes	1173
2	Afghanistan	5072	Variation de stock	2511	Blé	2013	Milliers de tonnes	-350
2	Afghanistan	5301	Disponibilité intérieure	2511	Blé	2013	Milliers de tonnes	5992
2	Afghanistan	5527	Semences	2511	Blé	2013	Milliers de tonnes	322
2	Afghanistan	5123	Pertes	2511	Blé	2013	Milliers de tonnes	775
2	Afghanistan	5142	Nourriture	2511	Blé	2013	Milliers de tonnes	4895
2	Afghanistan	645	Disponibilité alimentaire en quantité (kg/personne/an)	2511	Blé	2013	kg	160.23
2	Afghanistan	664	Disponibilité alimentaire (Kcal/personne/jour)	2511	Blé	2013	Kcal/personne/jour	1369
2	Afghanistan	674	Disponibilité de protéines en quantité (g/personne/jour)	2511	Blé	2013	g/personne/jour	36.91
2	Afghanistan	684	Disponibilité de matière grasse en quantité (g/personne/jour)	2511	Blé	2013	g/personne/jour	4.69

fr_population.csv

Code zone	Zone	Année	Unité	Valeur
2	Afghanistan	2013	1000 personnes	30552
202	Afrique du Sud	2013	1000 personnes	52776
3	Albanie	2013	1000 personnes	3173
4	Algérie	2013	1000 personnes	39208
79	Allemagne	2013	1000 personnes	82727
7	Angola	2013	1000 personnes	21472
8	Antigua-et-Barbuda	2013	1000 personnes	90
194	Arabie saoudite	2013	1000 personnes	28829
9	Argentine	2013	1000 personnes	41446
1	Arménie	2013	1000 personnes	2977
10	Australie	2013	1000 personnes	23343
11	Autriche	2013	1000 personnes	8495
52	Azerbaïdjan	2013	1000 personnes	9413
12	Bahamas	2013	1000 personnes	377
16	Bangladesh	2013	1000 personnes	156595
14	Barbade	2013	1000 personnes	285

fr_sousalimentation.csv

Code zone	Zone	Année	Unité	Valeur
2	Afghanistan	2012-2014	millions	7.9
2	Afghanistan	2013-2015	millions	8.8
2	Afghanistan	2014-2016	millions	9.6
2	Afghanistan	2015-2017	millions	10.2
2	Afghanistan	2016-2018	millions	10.6
202	Afrique du Sud	2012-2014	millions	2.6
202	Afrique du Sud	2013-2015	millions	2.8
202	Afrique du Sud	2014-2016	millions	3.2
202	Afrique du Sud	2015-2017	millions	3.4
202	Afrique du Sud	2016-2018	millions	3.5
3	Albanie	2012-2014	millions	0.2
3	Albanie	2013-2015	millions	0.2
3	Albanie	2014-2016	millions	0.2
3	Albanie	2015-2017	millions	0.2
3	Albanie	2016-2018	millions	0.2
4	Algérie	2012-2014	millions	1.7

DONNEES TRAITEES

- 5 fichiers csv :

dispo_alim.csv
equilibre_prod.csv
balance_com.csv
population.csv
sous_nutrition.csv

- **FICHIERS CSV = DATAFRAMES = TABLES**

dispo_alim.csv

pays	code_pays	annee	produit	code_produit	origin	dispo_alim_tonnes	dispo_alim_kcal_p_j	dispo_prot	dispo_mat_gr
Arménie	1	2013	Blé	2511	végétal	389000	1024.0	30.52	3.6
Arménie	1	2013	Avoine	2516	végétal	1000	2.0	0.09	0.03
Arménie	1	2013	Céréales, Autres	2520	végétal	6000	14.0	0.39	0.04
Arménie	1	2013	Pommes de Terre	2531	végétal	134000	82.0	1.94	0.12
Arménie	1	2013	Sucre Eq Brut	2542	végétal	98000	302.0	0.0	0.0
Arménie	1	2013	Edulcorants Autres	2543	végétal	5000	18.0	0.0	0.0
Arménie	1	2013	Noix	2551	végétal	6000	18.0	0.42	1.76
Arménie	1	2013	Arachides Decortiquees	2556	végétal	1000	6.0	0.29	0.55
Arménie	1	2013	Olives	2563	végétal	3000	3.0	0.03	0.28
Arménie	1	2013	Huile de Soja	2571	végétal	1000	4.0	0.0	0.5
Arménie	1	2013	Huile de Tournesol	2573	végétal	26000	216.0	0.04	24.38
Arménie	1	2013	Huile de Coco	2578	végétal	1000	7.0	0.0	0.81
Arménie	1	2013	Tomates	2601	végétal	262000	41.0	1.94	0.48
Arménie	1	2013	Oignons	2602	végétal	52000	15.0	0.53	0.1
Arménie	1	2013	Légumes, Autres	2605	végétal	724000	143.0	6.66	1.18
Arménie	1	2013	Oranges, Mandarines	2611	végétal	14000	4.0	0.07	0.01
Arménie	1	2013	Citrons & Limes	2612	végétal	1000	0.0	0.01	0.0

CLEF PRIMAIRE = code_pays + code_produit

equilibre_prod.csv

pays	code_pays	annee	produit	code_produit	dispo_int	alim_an	semences	pertes	transfo	nourriture	autres_utilisations
Arménie	1	2013	Blé	2511	554	93	30	32	10	389	0
Arménie	1	2013	Orge	2513	198	137	14	15	7	0	25
Arménie	1	2013	Maïs	2514	102	96	0	7	0	0	-1
Arménie	1	2013	Seigle	2515	1	1	0	0	0	0	0
Arménie	1	2013	Avoine	2516	6	4	0	0	0	1	1
Arménie	1	2013	Céréales, Autres	2520	21	1	1	1	0	6	12
Arménie	1	2013	Pommes de Terre	2531	672	225	123	96	78	134	16
Arménie	1	2013	Sucre, betterave	2537	55	1	0	0	54	0	0
Arménie	1	2013	Sucre Eq Brut	2542	93	0	0	0	0	98	-5
Arménie	1	2013	Edulcorants Autres	2543	5	0	0	0	0	5	0
Arménie	1	2013	Haricots	2546	5	5	0	0	0	0	0
Arménie	1	2013	Pois	2547	2	2	0	0	0	0	0
Arménie	1	2013	Légumineuses Autres	2549	3	3	0	0	0	0	0
Arménie	1	2013	Noix	2551	6	0	0	0	0	6	0
Arménie	1	2013	Arachides Decortiquees	2556	1	0	0	0	0	1	0
Arménie	1	2013	Graines de tournesol	2557	2	0	0	0	2	0	0

CLEF PRIMAIRE = code_pays + code_produit

balance_com.csv

pays	code_pays	annee	produit	code_produit	productions	importations	exportations	variations
Arménie	1	2013	Blé	2511	312	361	1	-118
Arménie	1	2013	Orge	2513	189	9	0	0
Arménie	1	2013	Maïs	2514	21	82	0	-1
Arménie	1	2013	Seigle	2515	1	0	0	0
Arménie	1	2013	Avoine	2516	5	1	0	0
Arménie	1	2013	Céréales, Autres	2520	17	4	0	0
Arménie	1	2013	Pommes de Terre	2531	661	5	9	15
Arménie	1	2013	Sucre, betterave	2537	55	0	0	0
Arménie	1	2013	Sucre Eq Brut	2542	8	85	2	2
Arménie	1	2013	Edulcorants Autres	2543	2	2	0	1
Arménie	1	2013	Haricots	2546	5	0	0	0
Arménie	1	2013	Pois	2547	0	2	0	0
Arménie	1	2013	Légumineuses Autres	2549	0	3	0	0
Arménie	1	2013	Noix	2551	5	2	1	0
Arménie	1	2013	Arachides Decortiquees	2556	0	1	0	0
Arménie	1	2013	Graines de tournesol	2557	0	2	0	0

CLEF PRIMAIRE = code_pays + code_produit

population.csv

pays	code_pays	annee	population
Afghanistan	2	2013	30552000
Afrique du Sud	202	2013	52776000
Albanie	3	2013	3173000
Algérie	4	2013	39208000
Allemagne	79	2013	82727000
Angola	7	2013	21472000
Antigua-et-Barbuda	8	2013	90000
Arabie saoudite	194	2013	28829000
Argentine	9	2013	41446000
Arménie	1	2013	2977000
Australie	10	2013	23343000
Autriche	11	2013	8495000
Azerbaïdjan	52	2013	9413000
Bahamas	12	2013	377000
Bangladesh	16	2013	156595000
Barbade	14	2013	285000

CLEF PRIMAIRE = code_pays

sous_nutrition.csv

pays	code_pays	annee	nb_personnes
Afghanistan	2	2013	7900000
Afrique du Sud	202	2013	2600000
Albanie	3	2013	200000
Algérie	4	2013	1700000
Angola	7	2013	8100000
Arabie saoudite	194	2013	1600000
Argentine	9	2013	1500000
Arménie	1	2013	100000
Bangladesh	16	2013	26100000
Bénin	53	2013	1000000
Bolivie (État plurinational de)	19	2013	2200000
Botswana	20	2013	500000
Bulgarie	27	2013	400000
Burkina Faso	233	2013	3400000
Cambodge	115	2013	2800000
Cameroun	32	2013	1700000

CLEF PRIMAIRE = code_pays

ALGEBRE RELATIONNEL

AGGREGATION

JOINTURE

Interne
Externe gauche

RESTRICTION

AGGREGATION

	code_pays	pays	produit	ratio_energie_poids	ratio_proteines
0	1	Arménie	Blé	2860.368946	0.085252
1	1	Arménie	Avoine	2173.210000	0.097794
2	1	Arménie	Céréales, Autres	2535.411667	0.070629
3	1	Arménie	Pommes de Terre	664.937388	0.015731
4	1	Arménie	Sucre Eq Brut	3348.517449	0.000000
...
9196	276	Soudan	Oeufs	1458.616842	0.105750
9197	276	Soudan	Poissons Eau Douce	893.990968	0.107279
9198	276	Soudan	Poissons Marins, Autres	0.000000	0.110855
9199	276	Soudan	Riz (Eq Blanchi)	4041.584167	0.080832
9200	276	Soudan	Lait - Excl Beurre	740.819908	0.037041

9201 rows × 5 columns

Colonne : produit
Oopération : médiane
Valeurs : ratio énergétiques

AGGREGATION

```
# median is better than mean against outliers  
table_ratio = table_ratio.groupby('produit').median()
```

	ratio_energie_poids	ratio_proteines
produit		
Abats Comestible	1121.479091	0.177801
Agrumes, Autres	232.894333	0.004562
Aliments pour enfants	3610.417778	0.151012
Ananas	397.850000	0.002816
Animaux Aquatiques Autre	0.000000	0.040675
...
Viande de Suides	2554.527647	0.101140
Viande de Volailles	1261.103077	0.124061
Viande, Autre	1168.234111	0.174804
Vin	704.632500	0.000000
Épices, Autres	3425.206105	0.105868

94 rows × 2 columns

Source : P3_01_coderéponses.ipynb

JOINTURE INTERNE

A

	pays	produit	dispo_int
0	Arménie	Blé	554
1	Arménie	Orge	198
2	Arménie	Maïs	102
3	Arménie	Seigle	1
4	Arménie	Avoine	6
...
11167	Soudan	Miel	1
11168	Soudan	Poissons Eau Douce	31
11169	Soudan	Poissons Marins, Autres	5
11170	Soudan	Riz (Eq Blanchi)	49
11171	Soudan	Lait - Excl Beurre	5892

11172 rows × 3 columns

B

	pays	produit	origin
0	Arménie	Blé	végétal
1	Arménie	Orge	végétal
2	Arménie	Maïs	végétal
3	Arménie	Seigle	végétal
4	Arménie	Avoine	végétal
...
11167	Soudan	Miel	végétal
11168	Soudan	Poissons Eau Douce	animal
11169	Soudan	Poissons Marins, Autres	animal
11170	Soudan	Riz (Eq Blanchi)	végétal
11171	Soudan	Lait - Excl Beurre	animal

11172 rows × 3 columns

```
A = p3.equilibre_prod[['pays', 'produit', 'dispo_int']].copy()
B = p3.dispo_alim[['pays', 'produit', 'origin']].copy()
```

Source : P3_04_module_fao.py

JOINTURE INTERNE

C RELATION 1 = 1

	pays	produit	dispo_int	origin
0	Arménie	Blé	554	végétal
1	Arménie	Orge	198	végétal
2	Arménie	Maïs	102	végétal
3	Arménie	Seigle	1	végétal
4	Arménie	Avoine	6	végétal
...
11167	Soudan	Miel	1	végétal
11168	Soudan	Poissons Eau Douce	31	animal
11169	Soudan	Poissons Marins, Autres	5	animal
11170	Soudan	Riz (Eq Blanchi)	49	végétal
11171	Soudan	Lait - Excl Beurre	5892	animal

11172 rows × 4 columns

```
C = A.merge(B,how = 'inner',on = ['pays','produit'])
```

Source : P3_04_module_fao.py

JOINTURE EXTERNE GAUCHE

A

	pays	produit	dispo_int
0	Arménie	Blé	554
1	Arménie	Orge	198
2	Arménie	Maïs	102
3	Arménie	Seigle	1
4	Arménie	Avoine	6
...
11167	Soudan	Miel	1
11168	Soudan	Poissons Eau Douce	31
11169	Soudan	Poissons Marins, Autres	5
11170	Soudan	Riz (Eq Blanchi)	49
11171	Soudan	Lait - Excl Beurre	5892

11172 rows × 3 columns

B

	produit	ratio_energie_poids
0	Abats Comestible	1121.479091
1	Agrumes, Autres	232.894333
2	Aliments pour enfants	3610.417778
3	Ananas	397.850000
4	Animaux Aquatiques Autre	0.000000
...
89	Viande de Suides	2554.527647
90	Viande de Volailles	1261.103077
91	Viande, Autre	1168.234111
92	Vin	704.632500
93	Épices, Autres	3425.206105

94 rows × 2 columns

```
A = p3.equilibre_prod[['pays', 'produit', 'dispo_int']].copy()
B = table_ratio[['produit', 'ratio_energie_poids']].copy()
```

Source : P3_04_module_fao.py

JOINTURE EXTERNE GAUCHE

C

RELATION plusieurs = 1

	pays	produit	dispo_int	ratio_energie_poids
0	Arménie	Blé	554	2827.954285
1	Arménie	Orge	198	2460.608788
2	Arménie	Maïs	102	3081.622000
3	Arménie	Seigle	1	2688.988953
4	Arménie	Avoine	6	2027.528132
...
11167	Soudan	Miel	1	3071.582120
11168	Soudan	Poissons Eau Douce	31	653.750640
11169	Soudan	Poissons Marins, Autres	5	660.193750
11170	Soudan	Riz (Eq Blanchi)	49	3618.228409
11171	Soudan	Lait - Excl Beurre	5892	587.231693

11172 rows × 4 columns

```
C = A.merge(B,how = 'left' ,on = ['produit'])
```

Source : P3_04_module_fao.py

RESTRICTION

e

	alim_an	code_produit	pays
0	93	2511	Arménie
1	137	2513	Arménie
2	96	2514	Arménie
3	1	2515	Arménie
4	4	2516	Arménie
...
11167	0	2745	Soudan
11168	0	2761	Soudan
11169	0	2764	Soudan
11170	0	2805	Soudan
11171	0	2848	Soudan

11172 rows × 3 columns

```
e = p3.equilibre_prod[['alim_an', 'code_produit', 'pays']].copy()
```

Source : P3_01_coderéponses.ipynb

RESTRICTION

```
# WHERE : pays USA AND code_produit IN liste_cereales  
e = e[ (e['pays'] == "États-Unis d'Amérique") & (e['code_produit'].isin(p3.Liste.cereales()))]
```

	alim_an	code_produit	pays
10155	6196	2511	États-Unis d'Amérique
10156	1440	2513	États-Unis d'Amérique
10157	128024	2514	États-Unis d'Amérique
10158	184	2515	États-Unis d'Amérique
10159	1430	2516	États-Unis d'Amérique
10160	394	2517	États-Unis d'Amérique
10161	2351	2518	États-Unis d'Amérique
10162	77	2520	États-Unis d'Amérique
10246	0	2805	États-Unis d'Amérique

REQUÊTES SQL

REQUÊTES SQL + Résultats

Description des "autres utilisations"

REQUÊTES SQL

Les 10 pays ayant le plus haut ratio disponibilité alimentaire/habitant en termes de protéines (en kg) par habitant, puis en termes de kcal par habitant.

```
SELECT      d.pays,  
             sum(d.dispo_prot*365/1000)  
             AS dispo_prot_kg_habitant  
  
FROM        fao.dispo_alim d  
  
GROUP BY    d.pays  
  
ORDER BY    dispo_prot_kg_habitant DESC  
  
LIMIT 10 ;
```

#	pays	dispo_prot_kg_habitant
1	Islande	48.5669000000000004
2	Israël	46.719999999999999
3	Lituanie	45.3914000000000004
4	Maldives	44.6468000000000006
5	Finlande	42.9094000000000005
6	Luxembourg	41.4786
7	Monténégro	40.8435
8	Pays-Bas	40.682900000000002
9	Albanie	40.65005
10	Portugal	40.467549999999996

REQUÊTES SQL

Les 10 pays ayant le plus haut ratio disponibilité alimentaire/habitant en termes de protéines (en kg) par habitant, puis en termes de kcal par habitant.

```
SELECT d.pays,  
       sum(d.dispo_alim_kcal_p_j*365)  
       AS dispo_kcal_habitant  
  
FROM   fao.dispo_alim d  
  
GROUP BY d.pays  
  
ORDER BY dispo_kcal_habitant DESC  
  
LIMIT 10 ;
```

#	pays	dispo_kcal_habitant
1	Autriche	1376050
2	Belgique	1364005
3	Turquie	1353420
4	États-Unis d'Amérique	1343930
5	Israël	1317650
6	Irlande	1314730
7	Italie	1305970
8	Luxembourg	1292100
9	Égypte	1284070
10	Allemagne	1278595

REQUÊTES SQL

Pour l'année 2013, les 10 pays ayant le plus faible ratio disponibilité alimentaire/habitant en termes de protéines (en kg) par habitant.

```
SELECT      d.pays,  
            sum(d.dispo_prot*365/1000)  
            AS dispo_prot_kg_habitant  
  
FROM        fao.dispo_alim d  
  
GROUP BY    d.pays  
  
ORDER BY    dispo_prot_kg_habitant ASC  
  
LIMIT 10 ;
```

#	pays	dispo_prot_kg_habitant
1	Libéria	13.745899999999999
2	Guinée-Bissau	16.078250000000004
3	Mozambique	16.673200000000005
4	République centrafricaine	16.8046
5	Madagascar	17.041850000000004
6	Haïti	17.410500000000006
7	Zimbabwe	17.636799999999997
8	Congo	18.764650000000003
9	Ouganda	19.213600000000001
10	Sao Tomé-et-Principe	19.381500000000003

REQUÊTES SQL

La quantité totale (en kg) de produits perdus par pays en 2013.

```
SELECT      e.pays,  
            sum(e.pertes*1000000)  
            AS pertes_kg  
  
FROM        fao.equilibre_prod e  
  
GROUP BY   e.pays ;
```

#	pays	pertes_kg
1	Arménie	228000000
2	Afghanistan	1135000000
3	Albanie	276000000
4	Algérie	3753000000
5	Angola	4799000000
6	Antigua-et-Barbuda	0
7	Argentine	3522000000
8	Australie	520000000
9	Autriche	459000000
10	Bahamas	2000000

REQUÊTES SQL

Les 10 pays pour lesquels la proportion de personnes sous-alimentées est la plus forte.

```
SELECT  p.pays,  
          (s.nb_personnes*100/p.population)  
          AS pourcentage  
  
FROM    fao.population p,  
          fao.sous_nutrition s  
  
WHERE    p.code_pays=s.code_pays  
  
ORDER BY pourcentage_ss_nutrition DESC  
  
LIMIT 10 ;
```

#	pays	pourcentage
1	Haïti	50.4022
2	Zambie	48.1464
3	Zimbabwe	46.6431
4	République centrafric...	43.3276
5	République populaire ...	42.5788
6	Congo	40.4676
7	Tchad	38.2066
8	Angola	37.7235
9	Libéria	37.2613
10	Madagascar	35.7688

REQUÊTES SQL

Les 10 produits pour lesquels le ratio **Autres utilisations/Disponibilité** intérieure est le plus élevé.

```
SELECT  e.produit,  
avg(e.autres_utilisations*100/e.dispo_int)  
AS  pourcentage_au_di  
  
FROM      fao.equilibre_prod e  
  
GROUP BY  e.produit  
  
ORDER BY  pourcentage_au_di DESC  
  
LIMIT 10 ;
```

#	produit	pourcentage
1	Alcool, non Comestible	100.0000...
2	Plantes Aquatiques	94.07359...
3	Huile de Palme	61.21989...
4	Huil Plantes Oleif Autr	56.93121...
5	Huile de Palmistes	52.94912...
6	Huile de Colza&Moutarde	45.89388...
7	Huiles de Poissons	40.49814...
8	Huile de Coco	36.34081...
9	Graisses Animales Crue	30.12386...
10	Manioc	26.622611...

REQUÊTES SQL

"autres utilisations" de la dernière requête

fabrication de savons

**fabrication de parfums
et produits cosmétiques**

fabrication d' agro-carburant

utilisation comme lubrifiant

#	produit	pourcentage
1	Alcool, non Comestible	100.0000...
2	Plantes Aquatiques	94.07359...
3	Huile de Palme	61.21989...
4	Huil Plantes Oleif Autr	56.93121...
5	Huile de Palmistes	52.94912...
6	Huile de Colza&Moutarde	45.89388...
7	Huiles de Poissons	40.49814...
8	Huile de Coco	36.34081...
9	Graisses Animales Crue	30.12386...
10	Manioc	26.622611...

ALIMENTER LA BASE DE DONNEES

```
#!/bin/bash

sudo mysql -uroot <<MYSQL_SCRIPT
CREATE DATABASE fao;
CREATE USER 'django_admin'@'localhost' IDENTIFIED BY 'Tkf821!jc';
GRANT ALL PRIVILEGES ON fao.* TO 'django_admin'@'localhost';
FLUSH PRIVILEGES;
MYSQL_SCRIPT
```

Créer la base de données MySQL

```
# Database
# https://docs.djangoproject.com/en/3.0/ref/settings/#databases

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'OPTIONS': {
            'read_default_file': '/etc/mysql/my.cnf',
        },
    },
}
```

Connecter DJANGO avec la base de données

ALIMENTER LA BASE DE DONNEES

```
1 from django.db import models
2
3 class Population(models.Model):
4     id_population = models.AutoField(primary_key = True)
5     pays          = models.CharField(max_length=100)
6     code_pays     = models.IntegerField()
7     annee         = models.IntegerField()
8     population    = models.BigIntegerField()
9
10    class Meta:
11        unique_together = (('code_pays', 'annee'),)
12        db_table = 'population'
13
14    class Sous_nutrition(models.Model):
15        id_sous_nutrition = models.AutoField(primary_key = True)
16        pays              = models.CharField(max_length = 100)
17        code_pays         = models.IntegerField()
18        annee             = models.IntegerField()
19        nb_personnes      = models.BigIntegerField()
20
21        class Meta:
22            unique_together = (('code_pays', 'annee'),)
23            db_table = 'sous_nutrition'
24
```

**Créer les tables dans le
fichier models.py**

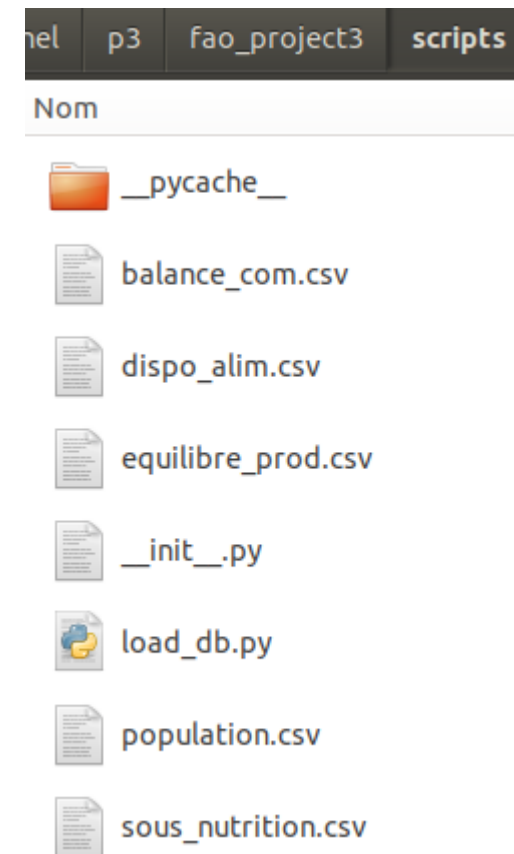
**./manage.py
makemigrations**

./manage.py migrate

ALIMENTER LA BASE DE DONNEES

```
load_db.py
1 import pandas as pd
2
3 from fao_database.models import *
4 from django.db import transaction
5
6 script_path = './scripts/'
7
8 dataframe_name = {0 : 'population',
9                  1 : 'sous_nutrition',
10                 2 : 'dispo_alim',
11                 3 : 'equilibre_prod',
12                 4 : 'balance_com'
13                }
14
15
16 population = pd.read_csv(script_path + dataframe_name[0] + '.csv', header = 0)
17 sous_nutrition = pd.read_csv(script_path + dataframe_name[1] + '.csv', header = 0)
18 dispo_alim = pd.read_csv(script_path + dataframe_name[2] + '.csv', header = 0)
19 equilibre_prod = pd.read_csv(script_path + dataframe_name[3] + '.csv', header = 0)
20 balance_com = pd.read_csv(script_path + dataframe_name[4] + '.csv', header = 0)
21
22 def run():
23
24     with transaction.atomic():
25
26         print(dataframe_name[0])
27         for row in population.itertuples():
28
29             Population.objects.create(pays = row[1],
30                                     code_pays = row[2],
31                                     annee = row[3],
32                                     population = row[4]
33                                     )
34
35
36
```

Charger les données traitées



./manage.py runscript load_db

QUESTIONS
