Projet 5

Produisez une étude de marché

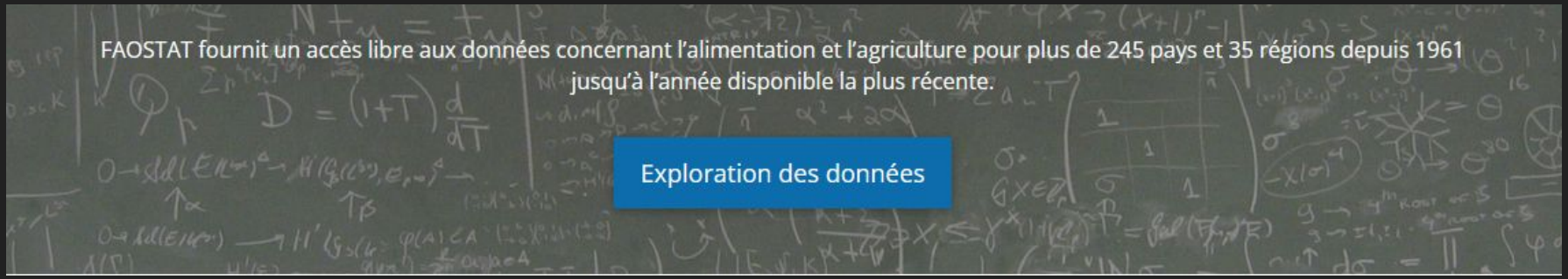
l'année 2020

Market study for a year 2020

Notre objectif

Notre entreprise alimentaire vise à se développer sur les marchés internationaux. La viande de poulet étant notre spécialité, nous voulons trouver les meilleurs pays pour importer ce produit.

Réflexion et approche



Nous commençons par télécharger les données de la FAO, qui fait partie des Nations Unies. Nous nous concentrons sur l'alimentation, la population, la richesse et la stabilité générale du marché. Pour cette raison, nous avons défini quelques variables principales pour notre sélection de données. Il s'agit de la dynamique de changement de la population, des valeurs et habitudes alimentaires, du PIB par habitant, de la stabilité politique, de la quantité de viande de poulet produite et importée.

Nettoyage des données

```python
# population df

# nettoyage des données
# data cleaning

exclusion_list = [512, 513, 551, 561]
population = population[~population['Element Code'].isin(exclusion_list)]

exlusion_list2 = ['World', 'Africa', 'Eastern Africa', 'Middle Africa',
                  'Northern Africa', 'Southern Africa', 'Western Africa',
                  'Americas', 'Northern America', 'Central America', 'Caribbean',
                  'South America', 'Asia', 'Central Asia', 'Eastern Asia',
                  'Southern Asia', 'South-Eastern Asia', 'Western Asia', 'Europe',
                  'Eastern Europe', 'Northern Europe', 'Southern Europe', 'Western Europe',
                  'Oceania', 'Australia and New Zealand', 'Melanesia', 'Micronesia', 'Polynesi
                  'European Union', 'Least Developed Countries', 'Land Locked Developing Count
                  'Small Island Developing States', 'Low Income Food Deficit Countries',
                  'Net Food Importing Developing Countries']

population = population[~population['Area'].isin(exlusion_list2)]

# exclude data for years we dont need
# exclure les données pour les années dont nous n'avons pas besoin

exlusion_list3 = [2015, 2020]

population = population[population['Year'].isin(exlusion_list3)]
```

```python
# enregistrer la variable comme un df séparé
# save variable as separate df
fb_food = food_balance.loc[(food_balance['Item'] == 'Grand Total') &
                           (food_balance['Element'] == 'Food supply (kcal/capita/day)')]
fb_protein = food_balance.loc[(food_balance['Item'] == 'Grand Total') &
                              (food_balance['Element'] == 'Protein supply quantity (g/capita/day)')]
fb_animal_protein = food_balance.loc[(food_balance['Item'] == 'Animal Products') &
                                     (food_balance['Element'] == 'Protein supply quantity (g/capita/day)')]
```

```python
# enregistrer la variable comme un df séparé
# save variable as separate df
fb_anim_prot_prop = fb_protein[['Area Code', 'Area', 'Item', 'Unit', 'Value']].merge(
    fb_animal_protein[['Area Code', 'Item', 'Value']], how='right', on='Area Code')

fb_anim_prot_prop['anim_prot_prop'] = fb_anim_prot_prop['Value_y'] / fb_anim_prot_prop['Value_x']

fb_anim_prot_prop.loc[fb_anim_prot_prop['Item_x'] == 'Grand Total'].head(3)
```

	Area Code	Area	Item_x	Unit	Value_x	Item_y	Value_y	anim_prot_prop
0	2	Afghanistan	Grand Total	g/capita/day	55.52	Animal Products	10.79	0.19
1	3	Albania	Grand Total	g/capita/day	115.74	Animal Products	61.75	0.53
2	4	Algeria	Grand Total	g/capita/day	91.83	Animal Products	24.73	0.27

```python
# calculer le changement de population en %.
# calculate population change as %

pop2015 = population.loc[population['Year'] == 2015].copy()
pop2020 = population.loc[population['Year'] == 2020].copy()

pop2020['pop%change'] = -((pop2015.Value.values - pop2020.Value.values) /
                          ((pop2015.Value.values + pop2020.Value.values) / 2)) * 100

pop2020.head(3)
```

	Area Code	Area	Item Code	Item	Element Code	Element	Year Code	Year	Unit	Value	Flag	Note	pop%change
70	2	Afghanistan	3010	Population - Est. & Proj.	511	Total Population - Both sexes	2020	2020	1000 persons	38,928.35	X	NaN	12.31
725	3	Albania	3010	Population - Est. & Proj.	511	Total Population - Both sexes	2020	2020	1000 persons	2,877.80	X	NaN	-0.44
1380	4	Algeria	3010	Population - Est. & Proj.	511	Total Population - Both sexes	2020	2020	1000 persons	43,851.04	X	NaN	9.87

Nettoyage des données

Chaque fichier csv de FAOSTAT comporte plusieurs éléments pour chaque pays, ventilés soit par une année spécifique (par exemple 2015), soit par une série d'années (par exemple 2016-2018). De plus, les données sont présentées par région et par continent. Nous ne sélectionnons que les éléments dont nous avons besoin pour notre objectif spécifique - données sur la viande de poulet, données diététiques, données sur la population, etc. Nous supprimons également tous les doublons possibles afin de ne disposer que de points de données spécifiques à chaque pays.

Notre échantillon final de données

Area	Pop%change	Food_supply_kcalcapyear	Protein_supply_gcapyear	Animal_protein_proportion	Chkn_production_tonnes	Chkn_import_tonnes	Import_price_USD_tonne	GDP_percapita_USD	Polit_stab
Afghanistan	12.31	744,600.00	20,264.80	0.19	28,493.00	23,913.00	1,344.03	2,065.00	-2.65
Albania	-0.44	1,226,400.00	42,245.10	0.54	11,633.00	11,588.00	1,350.48	13,671.50	0.12
Angola	16.40	870,525.00	19,363.25	0.30	49,034.00	322,678.00	1,047.10	6,670.30	-0.31
Antigua and Barbuda	4.56	892,425.00	29,269.35	0.64	32.00	5,944.00	1,938.10	21,548.70	0.96
Argentina	4.80	1,207,055.00	38,971.05	0.64	2,202,707.00	6,624.00	1,311.06	22,063.90	-0.12

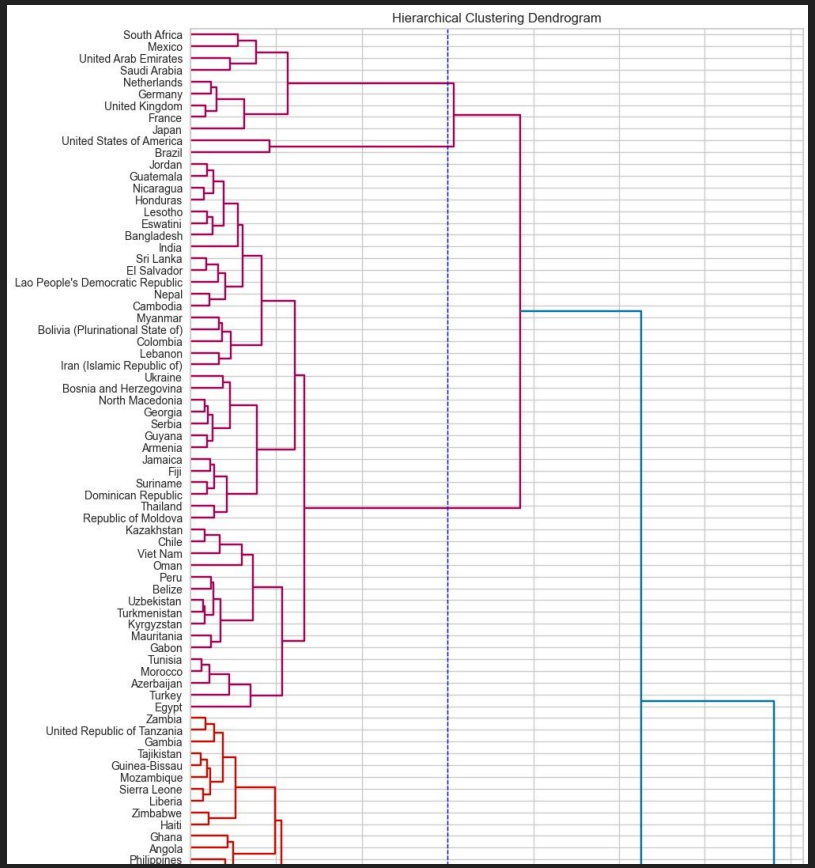
```
# Quantité de valeurs manquantes, par colonne, en %.
# amount of missing values, per column, in %
(df.isna().sum()/df.shape[0]*100).round(2)
```

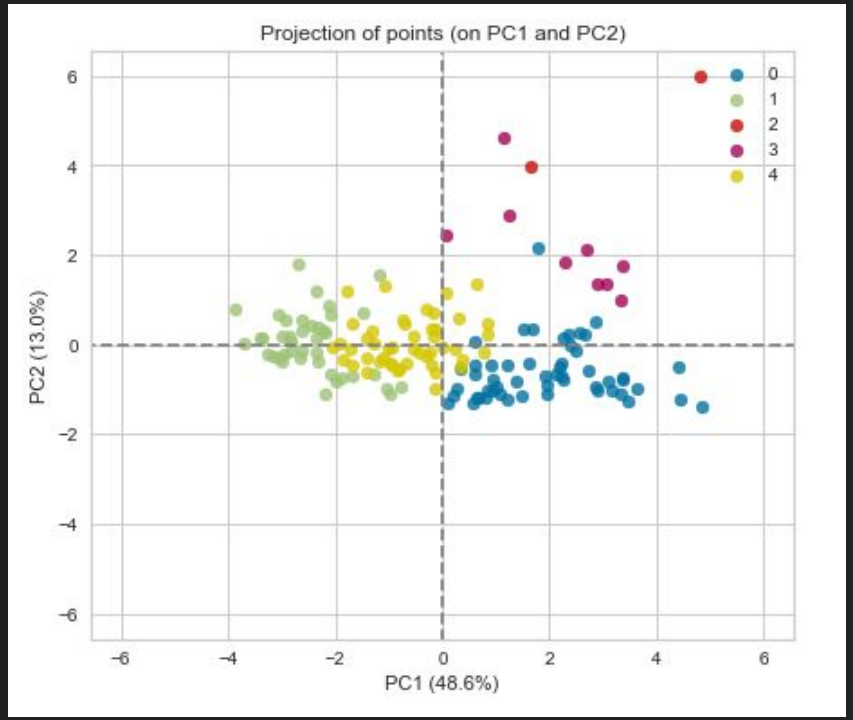
```
Area Code                    0.00
Area                         0.00
Pop%change                   0.00
Food_supply_kcalcapyear     27.47
Protein_supply_gcapyear     27.47
Animal_protein_proportion   27.47
Chkn_production_tonnes      17.60
Chkn_import_tonnes          19.74
Chkn_export_tonnes          44.64
ChknMeat_price_USD_tonne    83.26
Import_price_USD_tonne      20.17
GDP_percapita_USD           20.60
Polit_stab                  16.74
dtype: float64
```

```
# vérifier qu'il n'y a pas de données manquantes après le nettoyage
# verify there is no missing data after cleaning
(df.isna().sum()/df.shape[0]*100).round(2)
```

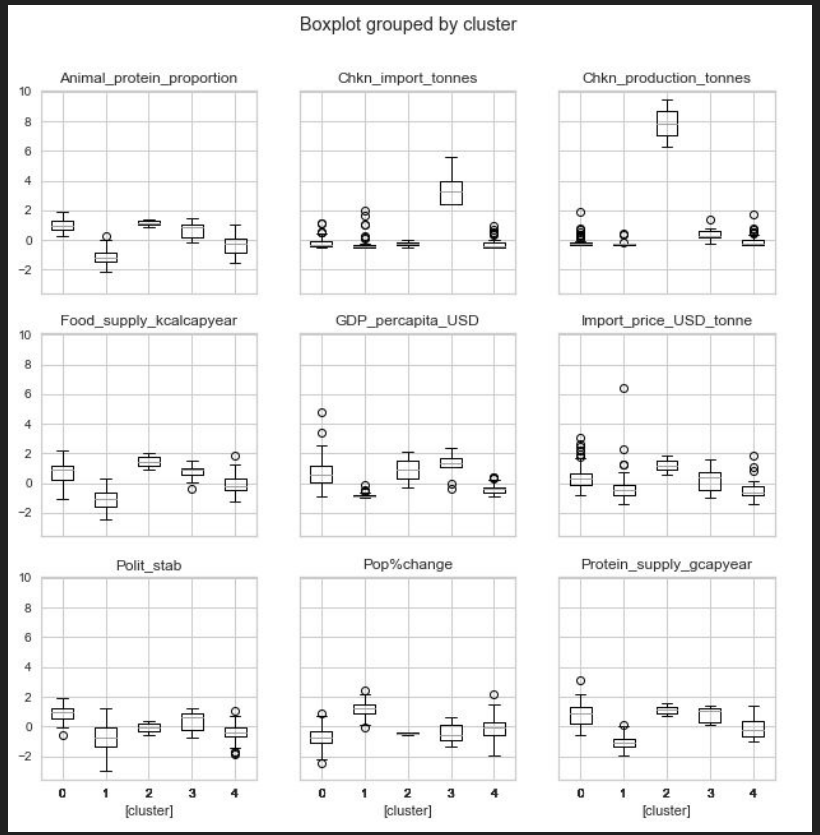
```
Pop%change                   0.00
Food_supply_kcalcapyear      0.00
Protein_supply_gcapyear      0.00
Animal_protein_proportion    0.00
Chkn_production_tonnes       0.00
Chkn_import_tonnes           0.00
Import_price_USD_tonne       0.00
GDP_percapita_USD            0.00
Polit_stab                   0.00
dtype: float64
```

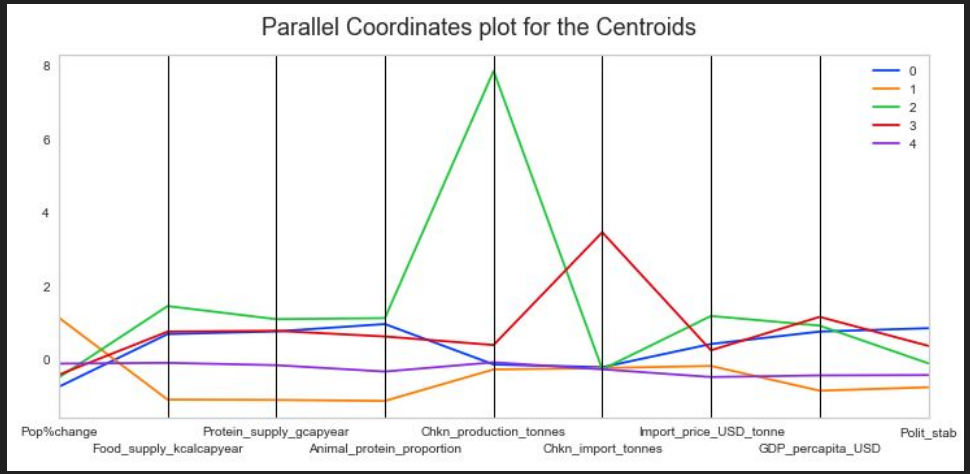
Clustering


Hierarchical Clustering Dendrogram
South Africa
Mexico
United Arab Emirates
Saudi Arabia
Netherlands
Germany
United Kingdom
France
Japan
United States of America
Brazil
Jordan
Guatemala
Nicaragua
Honduras
Lesotho
Eswatini
Bangladesh
India
Sri Lanka
El Salvador
Lao People's Democratic Republic
Nepal
Cambodia
Myanmar
Bolivia (Plurinational State of)
Colombia
Lebanon
Iran (Islamic Republic of)
Ukraine
Bosnia and Herzegovina
North Macedonia
Georgia
Serbia
Guyana
Armenia
Jamaica
Fiji
Suriname
Dominican Republic
Thailand
Republic of Moldova
Kazakhstan
Chile
Viet Nam
Oman
Peru
Belize
Uzbekistan
Turkmenistan
Kyrgyzstan
Mauritania
Gabon
Tunisia
Morocco
Azerbaijan
Turkey
Egypt
Zambia
United Republic of Tanzania
Gambia
Tajikistan
Guinea-Bissau
Mozambique
Sierra Leone
Liberia
Zimbabwe
Haiti
Ghana
Angola
Philippines


Projection of points (on PC1 and PC2)
PC2 (13.0%)
PC1 (48.6%)
6
4
2
0
−2
−4
−6
−6
−4
−2
0
2
4
6
0
1
2
3
4

Clustering


Boxplot grouped by cluster
Animal_protein_proportion
Chkn_import_tonnes
Chkn_production_tonnes
Food_supply_kcalcapyear
GDP_percapita_USD
Import_price_USD_tonne
Polit_stab
Pop%change
Protein_supply_gcapyear
[cluster]
[cluster]
[cluster]


Parallel Coordinates plot for the Centroids
0
1
2
3
4
Pop%change
Food_supply_kcalcapyear
Protein_supply_gcapyear
Animal_protein_proportion
Chkn_production_tonnes
Chkn_import_tonnes
Import_price_USD_tonne
GDP_percapita_USD
Polit_stab

Clustering

Les résultats obtenus par l'algorithme Dendrogramme nous montrent que parmi les 5 clusters, il y a deux bons groupes de pays pour notre objectif. Le cluster 0 et le cluster 3. Les pays de ces groupes ont une grande stabilité politique, un PIB élevé, une proportion élevée de protéines animales dans le régime alimentaire de la population et une grande quantité de viande de poulet importée.

Nous pouvons sélectionner des pays individuels dans chaque groupe en fonction de nos souhaits, ou nous pouvons appliquer une approche plus axée sur les données pour obtenir une liste exacte de pays.

```python
# réduire la liste des pays en utilisant nos données
# narrow down list of countries using our data

# better way to see which countries heavily rely on imports
country_selection = df_clustered.copy()
country_selection['prod_imp_proportion'] = country_selection['Chkn_import_tonnes'] / country_selection['Chkn_production_t
country_selection = country_selection.replace([np.inf, -np.inf], 0)
country_selection = country_selection.sort_values(by='prod_imp_proportion', ascending=False)

# remove countries with bad political stability, low animal protein proportion, and with dominant production and\or low i
country_selection = country_selection.loc[(country_selection['Polit_stab'] >= 0) &
                                          (country_selection['Animal_protein_proportion'] >= 0.50) &
                                          (country_selection['prod_imp_proportion'] >= 1.50)]
# sort by richest countries among those
country_selection = country_selection.sort_values(by='GDP_percapita_USD', ascending=False)
# save result into csv
country_selection.index.name = 'Country'
country_selection.to_csv('target_countries.csv')
country_selection
```

Country	Pop%change	Food_supply_kcalcapyear	Protein_supply_gcapyear	Animal_protein_proportion	Chkn_production_tonnes	Chkn_import_tonnes	Import_p
Luxembourg	9.93	1,264,725.00	39,657.25	0.62	294.00	7,816.00	
Bahamas	4.96	969,075.00	29,451.85	0.64	6,206.00	19,536.00	
Antigua and Barbuda	4.56	892,425.00	29,269.35	0.64	32.00	5,944.00	
Montenegro	0.18	1,277,500.00	41,803.45	0.61	3,769.00	7,268.00	
Grenada	2.63	876,730.00	26,031.80	0.56	415.00	6,101.00	
Saint Lucia	2.48	955,570.00	31,324.30	0.61	1,429.00	3,544.00	
Saint Vincent and the Grenadines	1.63	1,083,320.00	32,915.70	0.56	452.00	7,219.00	
Mongolia	8.92	941,335.00	31,868.15	0.65	192.00	11,642.00	
Dominica	1.12	1,077,480.00	28,864.20	0.57	362.00	3,916.00	
Samoa	2.50	1,105,950.00	32,065.25	0.60	482.00	16,640.00	

Sélectionnez les pays avec:

- une stabilité positive
- proportion élevée de protéines animales
- proportion élevée d'importations par rapport à la production

Trier par PIB, en commençant par les plus riches.

Tests statistiques

Test d'adequation

Goodness-of-fit test to find a variable with normal distribution. All our variables are continuous, we use Kolmogorov-Smirnov test.

```
# Kolmogorov-Smirnov (for continuous variable)
print(ks_2samp(df['Pop%change'],
        list(np.random.normal(np.mean(df['Pop%change']), np.std(df['Pop%change']), 1000))))

# Shapiro-Wilk
print(stats.shapiro(df['Pop%change']))
```

```
KstestResult(statistic=0.08394736842105263, pvalue=0.2920721727279667)
ShapiroResult(statistic=0.9866617321968079, pvalue=0.15317919850349426)
```

KS pvalue = 0,3, nous ne rejetons pas N0 (la distribution est normale) au niveau de confiance de 5%;
KS pvalue = 0.3, we do not reject N0 (distribution is normal) at 5% confidence level;

Shapiro pvalue = 0.15, nous ne rejetons pas N0 (la distribution est normale) au niveau de confiance de 5%;
Shapiro pvalue = 0.15, we do not reject N0 (distribution is normal) at 5% confidence level;

```
print(ks_2samp(df['GDP_percapita_USD'],
        list(np.random.normal(np.mean(df['GDP_percapita_USD']), np.std(df['GDP_percapita_USD']), 1000))))

print(stats.shapiro(df['GDP_percapita_USD']))
```

```
KstestResult(statistic=0.17894736842105263, pvalue=0.00036030577401113817)
ShapiroResult(statistic=0.837209939956665, pvalue=1.052164171760954e-11)
```

KS pvalue est très faible, nous rejetons donc N0 et acceptons l'hypothèse alternative selon laquelle la distribution n'est PAS normale, avec un niveau de confiance de 5 %;
KS pvalue is very small hence we reject N0 and accept Alternative Hypothesis that distribution is NOT normal, at 5% confidence level;

Shapiro pvalue est très faible, nous rejetons donc N0 et acceptons l'hypothèse alternative selon laquelle la distribution n'est PAS normale, avec un niveau de confiance de 5 %;
Shapiro pvalue is very small, we reject N0 and accept Alternative Hypothesis that distribution is NOT normal, at 5% confidence level;

Pour vérifier que nos algorithmes de clustering ont produit des groupes de pays réellement différents, nous avons effectué des tests statistiques pour le vérifier.

Test de comparaison de deux populations (dans le cas gaussien)

equality of variances

https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.bartlett.html
Bartlett's test tests the null hypothesis that all input samples are from populations with equal variances.

```
1 scipy.stats.bartlett(cl0_pop, cl1_pop)
```

```
BartlettResult(statistic=2.2949589049547128, pvalue=0.12979464612085712)
```

pvalue = 0.12, nous ne rejetons pas N0 (égalité des variances) à un niveau de confiance de 5% ;
pvalue = 0.12, we do not reject N0 (equality of variances) at 5% test confidence;

```
1 scipy.stats.bartlett(cl0_gdp, cl1_gdp)
```

```
BartlettResult(statistic=96.33709430753933, pvalue=9.689745963772276e-23)
```

La valeur p est très faible, donc nous rejetons N0 et acceptons l'hypothèse alternative avec un niveau de confiance de 5% (les variances ne sont PAS égales)
;
pvalue is very small, hence we reject N0 and accept Alternative hypothesis at 5% test confidence (variances are NOT equal);

Test de comparaison de deux populations (dans le cas gaussien)

equality of means

https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.ttest_ind.html
two-sided test for the null hypothesis that 2 independent samples have identical average (expected) values. This test assumes that the populations have identical variances by default.

```
1 scipy.stats.ttest_ind(cl0_pop, cl1_pop, equal_var=True)
```

```
Ttest_indResult(statistic=-14.523115790467816, pvalue=1.5437377159055225e-25)
```

La valeur p est très faible, donc nous rejetons N0 et acceptons l'hypothèse alternative avec un niveau de confiance de 5% (les moyennes ne sont PAS égales) ;
pvalue is very small, hence we reject N0 and accept Alternative hypothesis at 5% test confidence (means are NOT equal);

```
1 scipy.stats.ttest_ind(cl0_gdp, cl1_gdp, equal_var=True)
```

```
Ttest_indResult(statistic=9.92253790864671, pvalue=3.3230010104369944e-16)
```

La valeur p est très faible, donc nous rejetons N0 et acceptons l'hypothèse alternative avec un niveau de confiance de 5% (les moyennes ne sont PAS égales) ;
pvalue is very small, hence we reject N0 and accept Alternative hypothesis at 5% test confidence (means are NOT equal);

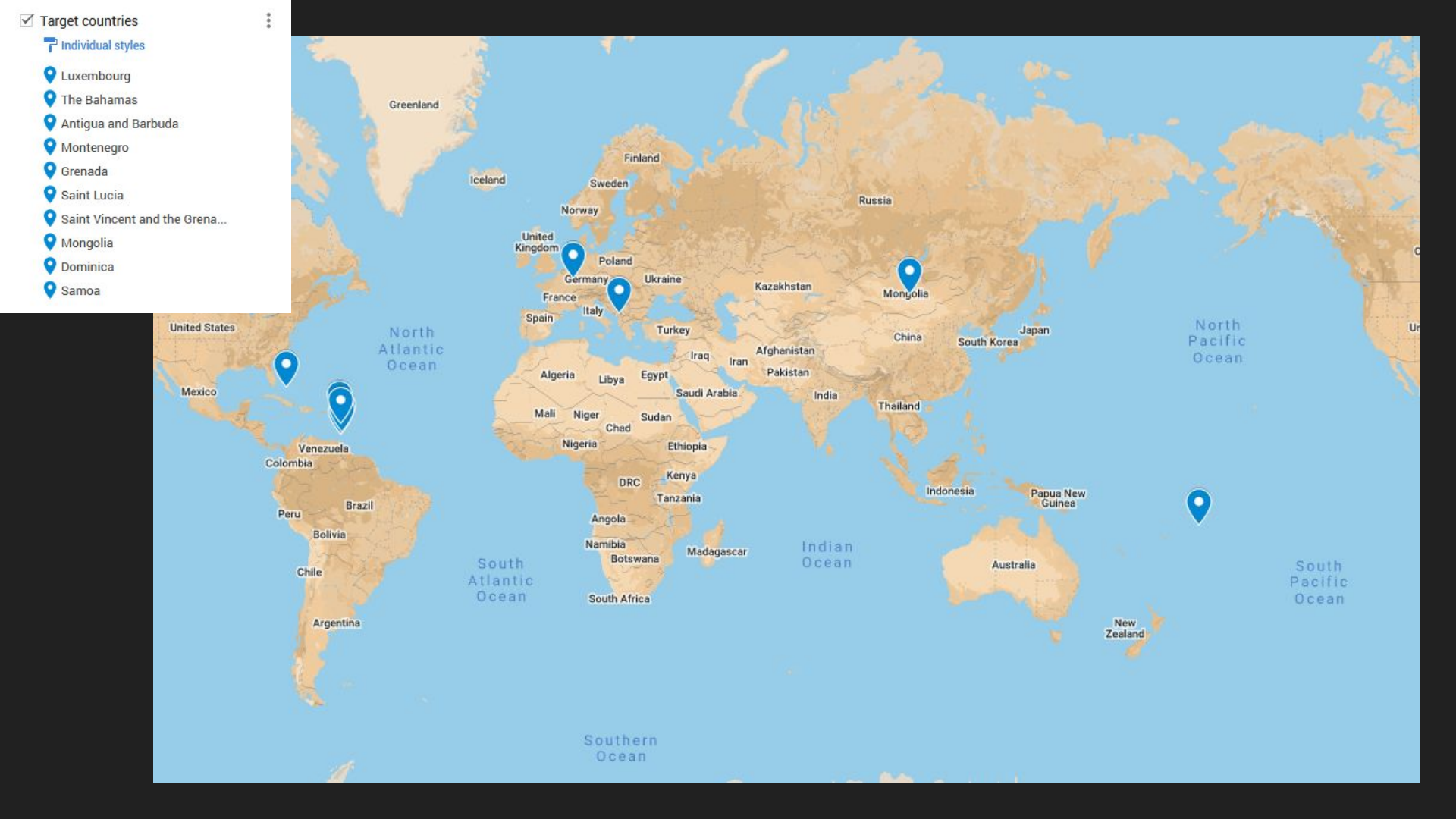
Conclusions générales : Grâce aux tests effectués, nous pouvons affirmer avec confiance que nos échantillons (clusters) sont différents.
Overall conclusions: Thanks to the carried out tests, we can confidently say that our samples (clusters) are different.

Conclusions générales après l'analyse effectuée

Globalement, nous voyons 2 clusters viables pour les importations de viande de poulet - le cluster 0 et le cluster 3.

Caractéristiques du cluster 0 - proportion élevée de protéines animales, production moyenne de poulet, importations élevées, prix d'importation élevé, PIB élevé, stabilité politique maximale. Groupe 3 - importations de poulet les plus élevées, PIB le plus élevé, stabilité politique positive.

La majorité d'entre eux sont des pays insulaires en développement qui n'ont pas de grandes possibilités de production et dont le régime alimentaire est fortement axé sur les protéines d'origine animale. La liste complète se trouve dans target_countries.csv.


Target countries
Individual styles
Luxembourg
The Bahamas
Antigua and Barbuda
Montenegro
Grenada
Saint Lucia
Saint Vincent and the Grena...
Mongolia
Dominica
Samoa
Greenland
Iceland
Finland
Sweden
Norway
Russia
United Kingdom
Poland
Germany
Ukraine
Kazakhstan
Mongolia
France
Italy
Spain
Turkey
China
South Korea
Japan
United States
North Atlantic Ocean
North Pacific Ocean
Iraq
Iran
Afghanistan
Pakistan
India
Thailand
Algeria
Libya
Egypt
Saudi Arabia
Mexico
Mali
Niger
Sudan
Chad
Nigeria
Ethiopia
Venezuela
Colombia
DRC
Kenya
Tanzania
Indonesia
Papua New Guinea
Brazil
Peru
Angola
Bolivia
Namibia
Botswana
Madagascar
Indian Ocean
Australia
Chile
South Atlantic Ocean
South Africa
South Pacific Ocean
Argentina
New Zealand
Southern Ocean