Produire une étude du marché

Réalisé par: ACHAT Hayat

Collecte des données

https://www.fao.org/faostat/fr/#data

table population:

	Domain Code	Domain	Area Code (FAO)	Area	Element Code	Element	Item Code	Item	Year Code	Year	Unit	Value	Flag	Flag Description	Note
0	OA	Annual population	2	Afghanistan	511	Total Population - Both sexes	3010	Population - Est. & Proj.	2018	2018	1000 persons	37171.921	X	International reliable sources	NaN
1	OA	Annual population	3	Albania	511	Total Population - Both sexes	3010	Population - Est. & Proj.	2018	2018	1000 persons	2882.740	X	International reliable sources	NaN
2	OA	Annual population	4	Algeria	511	Total Population - Both sexes	3010	Population - Est. & Proj.	2018	2018	1000 persons	42228.408	X	International reliable sources	NaN
3	OA	Annual population	5	American Samoa	511	Total Population - Both sexes	3010	Population - Est. & Proj.	2018	2018	1000 persons	55.465	X	International reliable sources	NaN

Nettoyage des données

vérification des lignes et colonnes:

```
population_17_df.columns
```

```
Index(['Domain Code', 'Domain', 'Area Code (FAO)', 'Area', 'Element Code',
       'Element', 'Item Code', 'Item', 'Year Code', 'Year', 'Unit', 'Value',
       'Flag', 'Flag Description', 'Note'],
      dtype='object')
```

```
population_17_df.value_counts()
```

```
Domain Code  Domain            Area Code (FAO)  Area     Element Code  Element                       Item Code  Item
Year Code  Year  Unit          Value  Flag  Flag Description              Note
OA           Annual population  218              Tokelau  511           Total Population - Both sexes  3010       Population -
Est. & Proj.  2017       2017  1000 persons  1.3    X     International reliable sources  UNDESA, Population Division ? World P
opulation Prospects, the 2017 Revision    1
dtype: int64
```

Nettoyage des données

vérifier les nans:

```
population_17_df.Value.isna().sum()
```

```
0
```

vérification des valeurs nuls:

```
: population_17_df.isnull().sum()
```

```
Domain Code           0
Domain                0
Area Code (FAO)       0
Area                  0
Element Code          0
Element               0
Item Code             0
Item                  0
Year Code             0
Year                  0
Unit                  0
Value                 0
Flag                  0
Flag Description      0
Note                232
dtype: int64
```

vérification des doublons

vérifier les doublons dans la table population:

```
population_17_df.loc[population_17_df.duplicated(keep=False),:]
```

Domain Code	Domain	Area Code (FAO)	Area	Element Code	Element	Item Code	Item	Year Code	Year	Unit	Value	Flag	Flag Description	Note

supprimer les doublons (provinces de chine)dans la table population:

```
population_17_df.drop(population_17_df[population_17_df.Area.str.startswith("China,") == True].index, inplace=True)
```

unifier les deux tables population 2017 et population 2018

```
population_mrg=pd.merge(population_18_df,population_17_df,on='Area Code (FAO)')
population_mrg.head()
```

	Domain Code_x	Domain_x	Area Code (FAO)	Area_x	Element Code_x	Element_x	Item Code_x	Item_x	Year Code_x	Year_x	...	Element_y	Item Code_y	Item_y	Year Code_y	Year_y	
0	OA	Annual population	2	Afghanistan	511	Total Population - Both sexes	3010	Population - Est. & Proj.	2018	2018	...	Total Population - Both sexes	3010	Population - Est. & Proj.	2017	2017	p
1	OA	Annual population	3	Albania	511	Total Population - Both sexes	3010	Population - Est. & Proj.	2018	2018	...	Total Population - Both sexes	3010	Population - Est. & Proj.	2017	2017	p
2	OA	Annual population	4	Algeria	511	Total Population - Both	3010	Population - Est. & Proj.	2018	2018	...	Total Population - Both	3010	Population - Est. & Proj.	2017	2017	p

Pourcentage de croissance de la population

calculer le **pourcentage de croissance** en calculant t la différence entre le nombre d'habitants en 2018 et ceux de l'année antérieure(2017)

```
population_croi=population.assign(prc_croissance=((population['Value_x']) - (population['Value_y']))/100)
population_croi.head()
```

	Area Code (FAO)	Area_x	Year_x	Value_x	Year_y	Value_y	Unit_y	prc_croissance
0	2	Afghanistan	2018	37171.921	2017	36296.113	1000 persons	8.75808
1	3	Albania	2018	2882.740	2017	2884.169	1000 persons	-0.01429
2	4	Algeria	2018	42228.408	2017	41389.189	1000 persons	8.39219
3	5	American Samoa	2018	55.465	2017	55.620	1000 persons	-0.00155
4	6	Andorra	2018	77.006	2017	77.001	1000 persons	0.00005

Collecter les données

table disponibilité alimentaire:

	Domain Code	Domain	Area Code	Area	Element Code	Element	Item Code	Item	Year Code	Year	Unit	Value	Flag	Flag Description
0	FBS	Food Balances (2010-)	2	Afghanistan	645	Food supply quantity (kg/capita/yr)	2511	Wheat and products	2018	2018	kg	160.12	Fc	Calculated data
1	FBS	Food Balances (2010-)	2	Afghanistan	664	Food supply (kcal/capita/day)	2511	Wheat and products	2018	2018	kcal/capita/day	1372.00	Fc	Calculated data
2	FBS	Food Balances (2010-)	2	Afghanistan	674	Protein supply quantity (g/capita/day)	2511	Wheat and products	2018	2018	g/capita/day	37.00	Fc	Calculated data

Nettoyage des données

vérifier les nans:

```
dispo_18_df.isna().describe()
```

	Domain Code	Domain	Area Code	Area	Element Code	Element	Item Code	Item	Year Code	Year	Unit	Value	Flag	Flag Description
count	59624	59624	59624	59624	59624	59624	59624	59624	59624	59624	59624	59624	59624	59624
unique	1	1	1	1	1	1	1	1	1	1	1	1	1	1
top	False	False	False	False	False	False	False	False	False	False	False	False	False	False
freq	59624	59624	59624	59624	59624	59624	59624	59624	59624	59624	59624	59624	59624	59624

```
dispo_18_df.value_counts()
```

```
Domain Code  Domain                Area Code  Area     Element Code  Element                          Item Code  Item
Year Code  Year  Unit            Value   Flag  Flag Description
FBS          Food Balances (2010-)  1          Armenia  645           Food supply quantity (kg/capita/yr)  2511       Wheat and
products  2018       2018  kg              125.27  Fc    Calculated data     1
                                 167        Czechia  664           Food supply (kcal/capita/day)        2765       Crustacea
ns        2018       2018  kcal/capita/day  0.00    Fc    Calculated data     1
                                                                                                        2734       Poultry M
eat       2018       2018  kcal/capita/day  83.00   Fc    Calculated data     1
                                                                                                        2735       Meat, Oth
```

Nettoyage des données

vérification des valeurs nuls:

```
dispo_18_df.isnull().sum()
```

```
Domain Code          0
Domain               0
Area Code            0
Area                 0
Element Code         0
Element              0
Item Code            0
Item                 0
Year Code            0
Year                 0
Unit                 0
Value                0
Flag                 0
Flag Description     0
dtype: int64
```

vérification des valeurs isna:

```
dispo_18_df.isna().sum()
```

```
Domain Code          0
Domain               0
Area Code            0
Area                 0
Element Code         0
Element              0
Item Code            0
Item                 0
Year Code            0
Year                 0
Unit                 0
Value                0
Flag                 0
Flag Description     0
dtype: int64
```

Nettoyage des données

supprimer les doublons (provinces de chine)dans la table population:

```
dispo_18_df.drop(dispo_18_df[dispo_18_df.Area.str.startswith("China,") == True].index, inplace=True)
```

Préparer les tables pour les opérations

pivoter la colonne éléments dans la table des Disponibilités :

```
dispo_18 = dispo_18_df.pivot_table(index=['Area', 'Item','Year'], columns='Element', values='Value',
           aggfunc = sum).reset_index()
dispo_18.head()
```

Element	Area	Item	Year	Fat supply quantity (g/capita/day)	Food supply (kcal/capita/day)	Food supply quantity (kg/capita/yr)	Protein supply quantity (g/capita/day)
0	Afghanistan	Apples and products	2018	0.03	7.0	4.94	0.04
1	Afghanistan	Bananas	2018	0.03	6.0	3.85	0.07
2	Afghanistan	Barley and products	2018	0.02	2.0	0.21	0.06
3	Afghanistan	Beans	2018	NaN	NaN	0.00	NaN
4	Afghanistan	Beer	2018	0.00	0.0	0.00	0.00

proportion de protéines d'origine animale par rapport à la quantité totale de protéines:

```
dispo_18.loc[dispo_18.Item.isin(anim_item),'Type'] = 'anim'
dispo_18.loc[~dispo_18.Item.isin(anim_item),'Type'] = 'other'
```

```
dispo_prot_proportion=dispo_prot_mrg.assign(proportion_protéines=((dispo_prot_mrg['dispo_prot_anim']) / (dispo_prot_mrg['Protein
```

```
dispo_prot_proportion
```

	Area	Protein supply quantity (g/capita/day)	dispo_prot_anim	proportion_protéines
0	Afghanistan	57.61	10.73	0.186252
1	Albania	112.81	61.02	0.540909
2	Algeria	90.30	24.61	0.272536
3	Angola	53.83	16.77	0.311536
4	Antigua and Barbuda	80.68	52.00	0.644522

Disponibilité alimentaire en protéines par habitant:

```
dispo_18_ans=dispo_18.assign(dispo_protéines_ans=((dispo_18['Protein supply quantity (g/capita/day)'])*365))
```

```
dispo_prot_habit=pd.DataFrame(dispo_18_ans.groupby(["Area"])['dispo_protéines_ans'].agg('sum').reset_index())
dispo_prot_habit
```

	Area	dispo_protéines_ans
0	Afghanistan	21027.65
1	Albania	41175.65
2	Algeria	32959.50

Disponibilité alimentaire en calories par habitant:

```
dispo_18_ans=dispo_18.assign(dispo_cal_ans=((dispo_18['Food supply (kcal/capita/day)'])*1000*365))
dispo_18_ans.head()
```

	Element	Area	Item	Year	Fat supply quantity (g/capita/day)	Food supply (kcal/capita/day)	Food supply quantity (kg/capita/yr)	Protein supply quantity (g/capita/day)	Type	dispo_cal_ans
0	Afghanistan	Apples and products	2018	0.03	7.0	4.94	0.04	other	2555000.0
1	Afghanistan	Bananas	2018	0.03	6.0	3.85	0.07	other	2190000.0

```
dispo_cal_habit=pd.DataFrame(dispo_18_ans.groupby(["Area"])['dispo_cal_ans'].agg('sum').reset_index())
dispo_cal_habit.head()
```

	Area	dispo_cal_ans
0	Afghanistan	8.274550e+08
1	Albania	1.202675e+09
2	Algeria	1.233335e+09
3	Angola	8.924250e+08
4	Antigua and Barbuda	9.008200e+08

Préparer les données pour la mission deux

unifier entre les trois table pour la suite de l'exercice

```
data_anal=data1.merge(data2)
data_anal
```

	Area	dispo_protéines_ans	proportion_protéines	dispo_cal_ans	prc_croissance
0	Afghanistan	21027.65	0.186252	8.274550e+08	8.75808
1	Albania	41175.65	0.540909	1.202675e+09	-0.01429
2	Algeria	32959.50	0.272536	1.233335e+09	8.39219

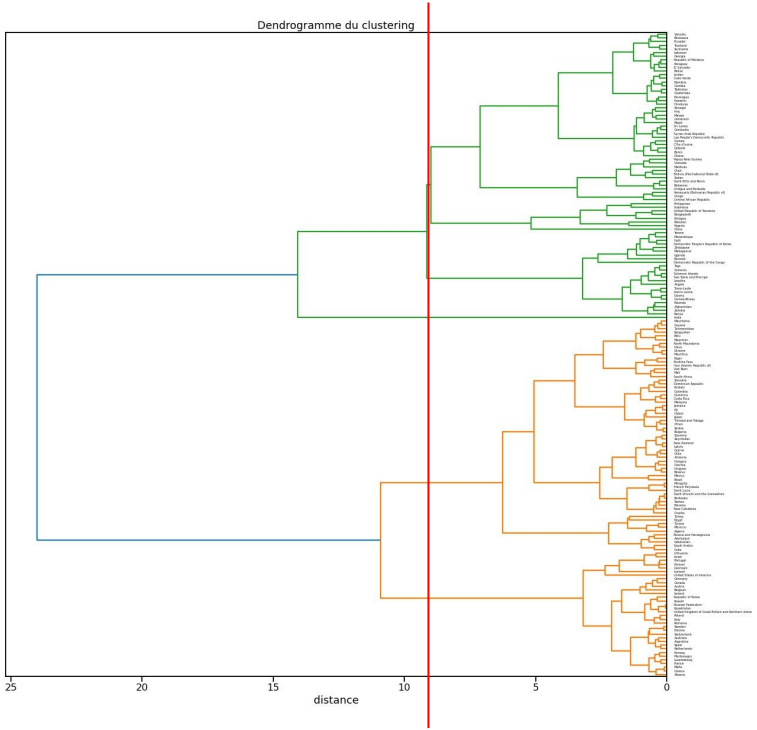
Préparer les données pour la mission deux

unifier entre les trois table pour la suite de l'exercice

```
data_anal=data1.merge(data2)
data_anal
```

	Area	dispo_protéines_ans	proportion_protéines	dispo_cal_ans	prc_croissance
0	Afghanistan	21027.65	0.186252	8.274550e+08	8.75808
1	Albania	41175.65	0.540909	1.202675e+09	-0.01429
2	Algeria	32959.50	0.272536	1.233335e+09	8.39219

Construire un dendrogramme des pays étudiés



Réalisation d'une classification hiérarchique

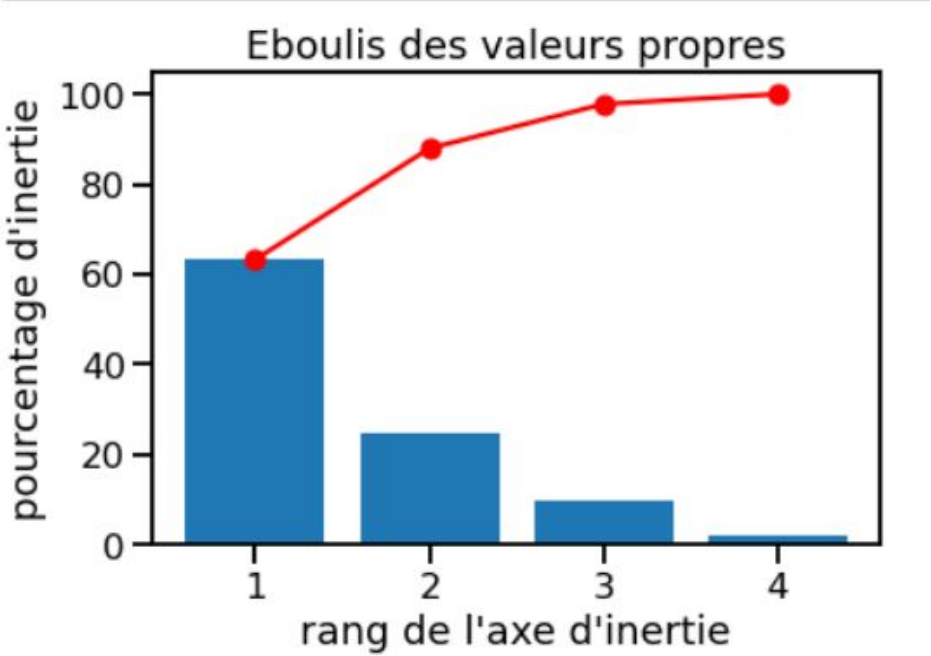
pour **obtenir un dendrogramme** des pays coupé en 5 groupes
pour former **5 clusters**
Méthode : Ward → Maximise l'inertie entre les groupes

```
: data_class.clusters.value_counts()
```

```
2    63
4    54
1    34
3    23
5     1
```

les groupes sont répartis d'une manière **inégalitaire** (un groupe solitaire)

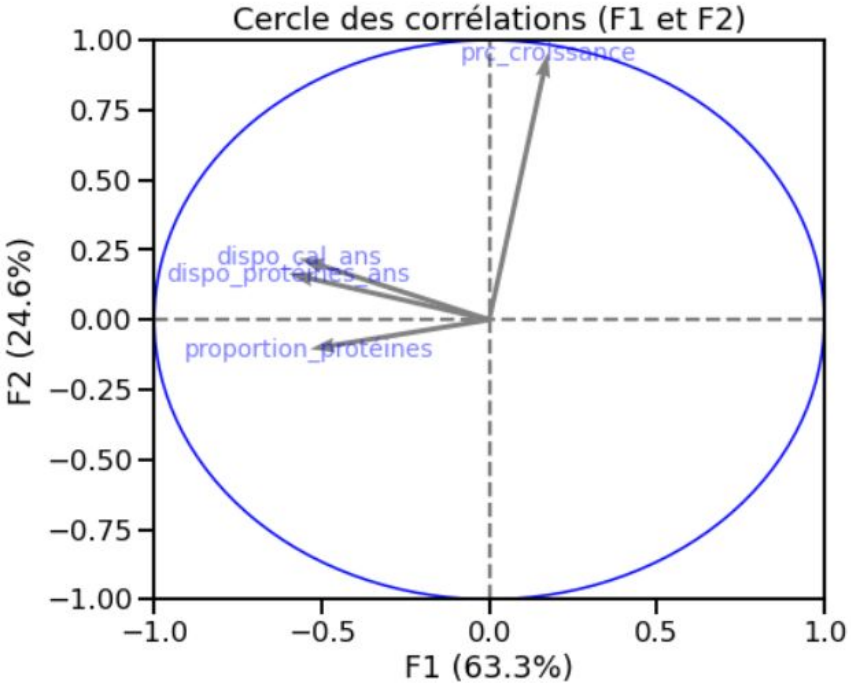
Visualisation des partitions dans le premier plan factoriel Analyse en Composante Principales(ACP)



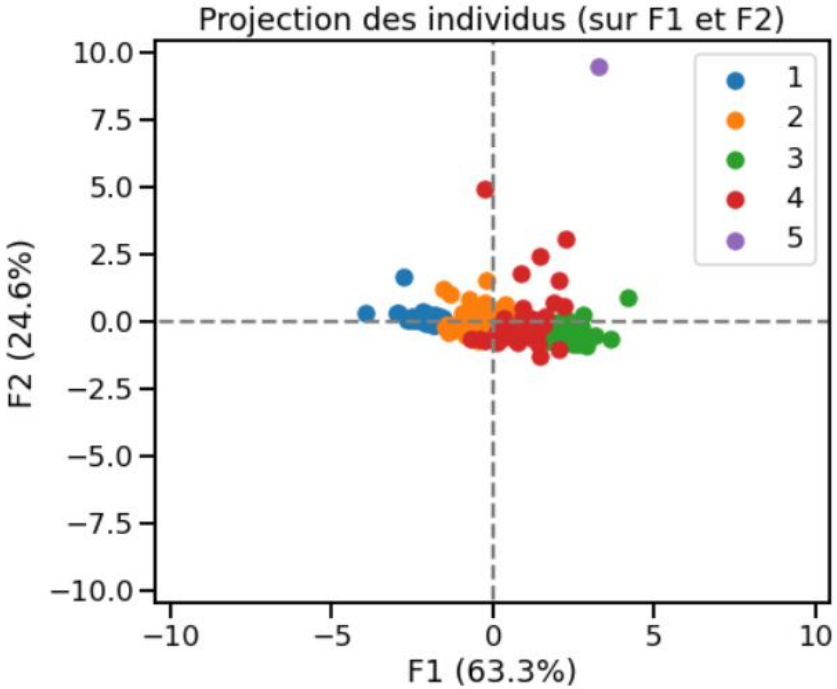
- on ne prend pas en compte les axes dont l'inertie associée est inférieure à (100/p)%
- 100/4 = 25 ce qui correspond à la première composante F1 et la deuxième F2

Visualisation des partitions dans le premier plan factoriel Analyse en Composante Principales

- l**a croissance** correspond à l'axe F2 et elle est dans **le sens positif**
- les disponibilités se projettent sur l'axe **F1** , s'orientent vers la **direction négatif** et sont presques égales
- on résume en deux axes principaux: **disponibilités** et **croissances**



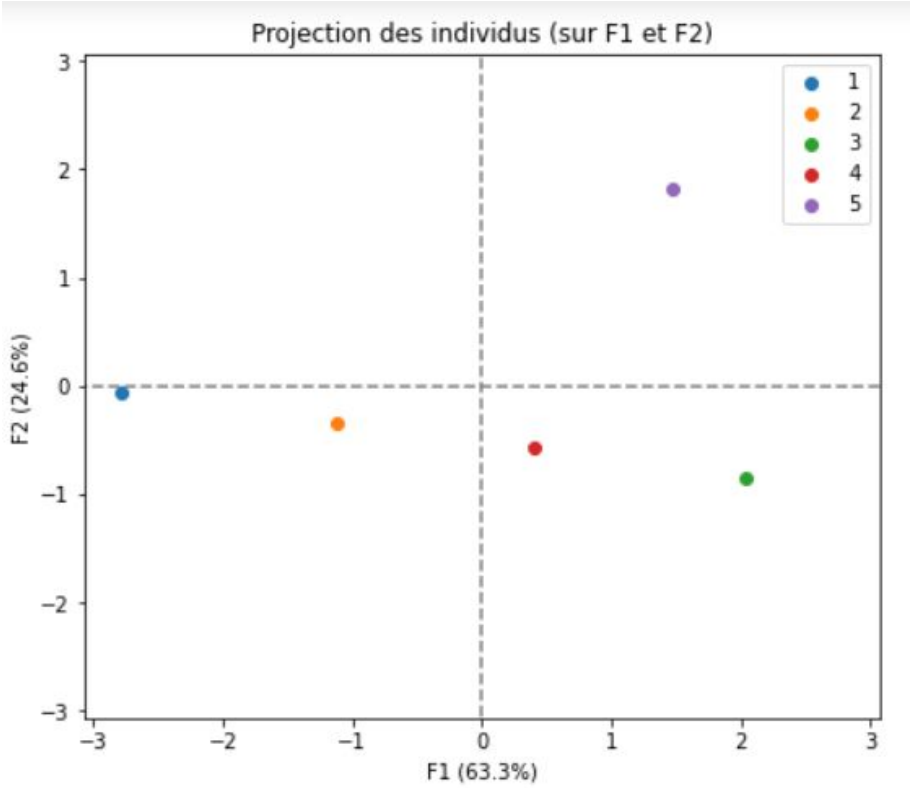
Projection des individus



- la partition n’est pas claire
- la répartition de nos groupe sur les deux plans factoriels suit l’axe de la disponibilité
- les groupe se resserre sur eux même
- le 4ème groupe est plutôt étalé selon l’axe de l croissance.
- le 5éme groupe comprend un seul pay

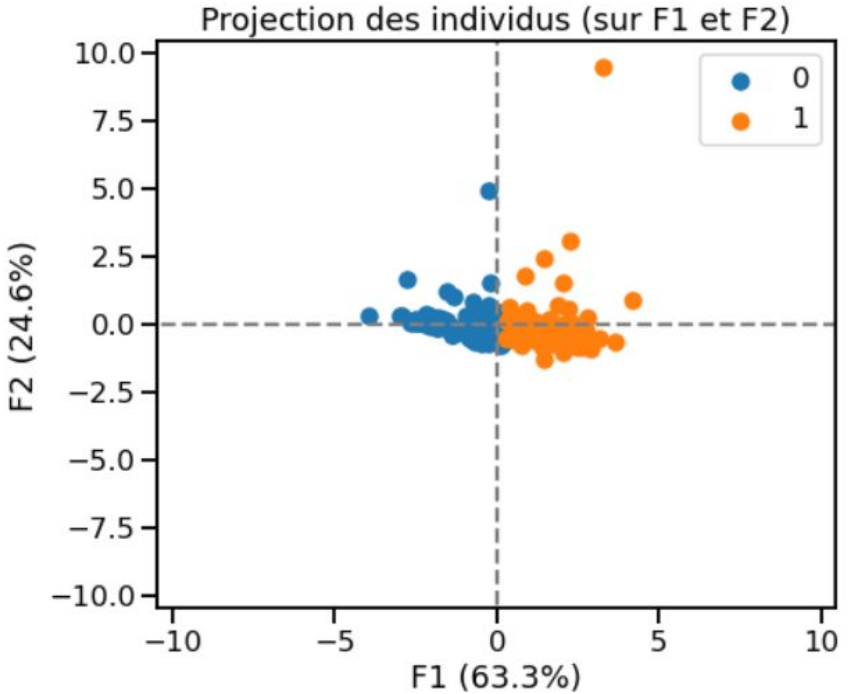
Projection des centroïdes

- les quatres clusters traçant une droite parallèle à l'axe de la disponibilité
- cluster1: une forte disponibilité en et une croissance moyenne
- le 2 et le 4:une moyenne disponibilité et croissance
- le 3:une très faible croissance et disponibilité
- quant au 5 ème une très forte croissance population et très faible disponibilité



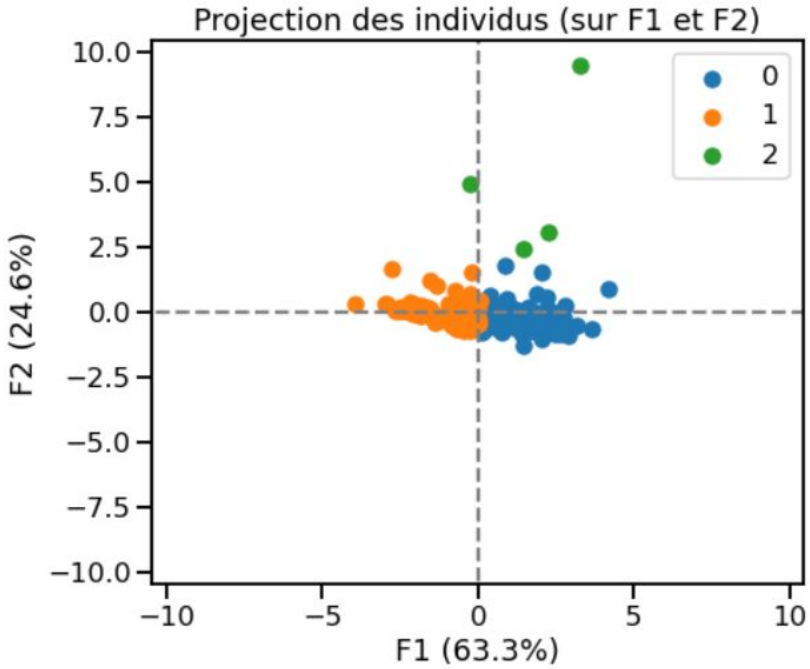
K Means (l'*algorithme des centres mobiles*)

K-means avec 2 clusters:



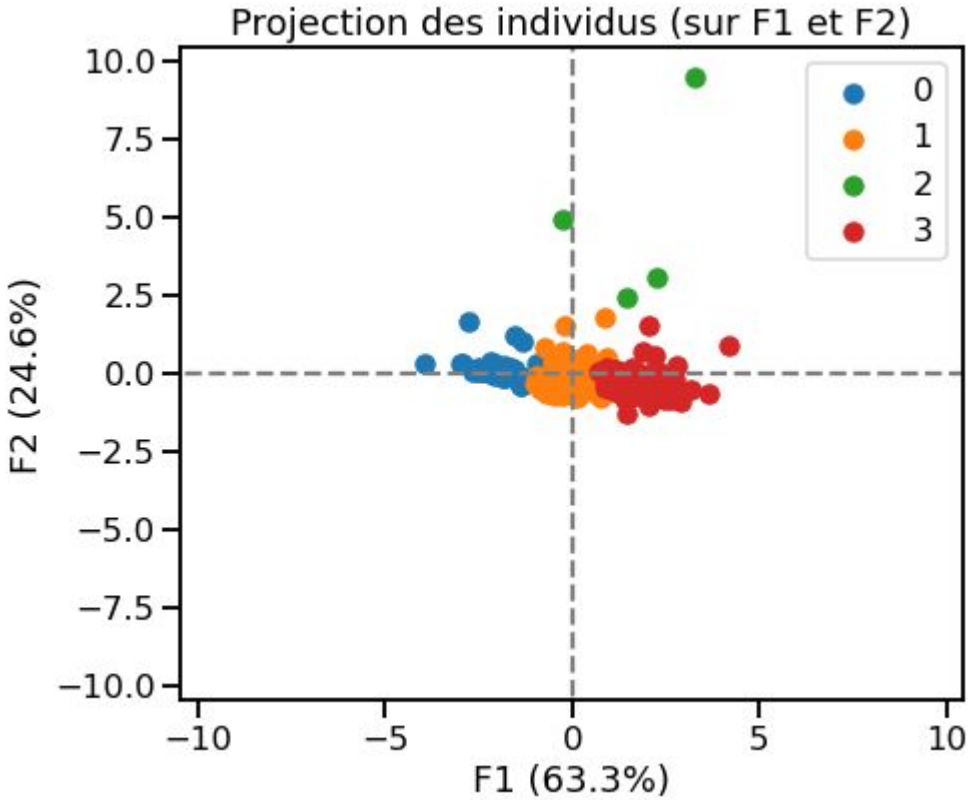
- Les deux cluster se répartissent à gauche et à droite de l'axe F1
- s'étalent suivant l'axe F2
- Le 1er cluster se répartis à gauche et représente le groupe de pays ayant les plus fortes disponibilité et croissance
- Le 2éme se situe à droite représente le groupe de pays ayant de faible disponibilités et croissance

K Means avec 3 clusters



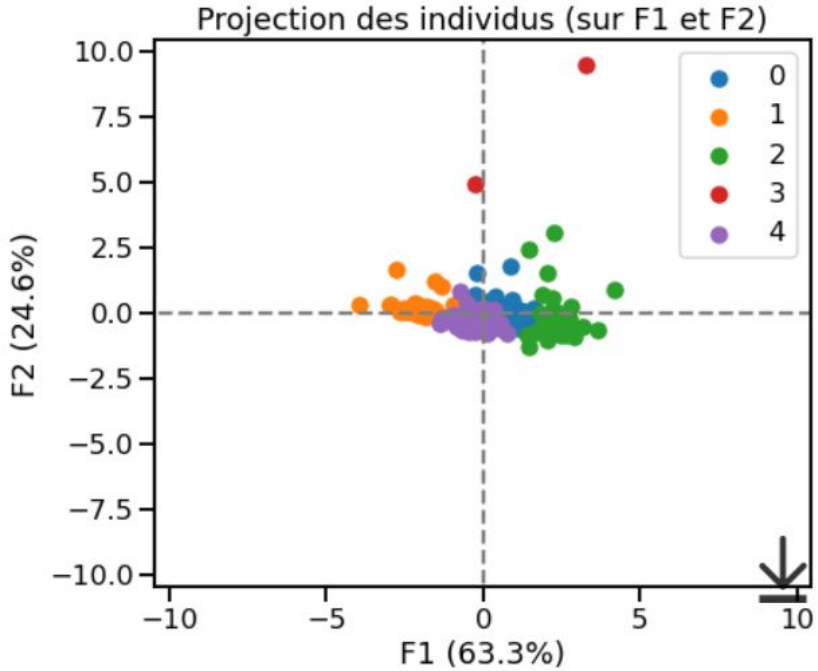
- la partie étalée et qui s’ éloignent des deux groupes forment le troisième cluster et se distinguent par une très forte croissance et une faible disponibilité

K Means avec 4 clusters



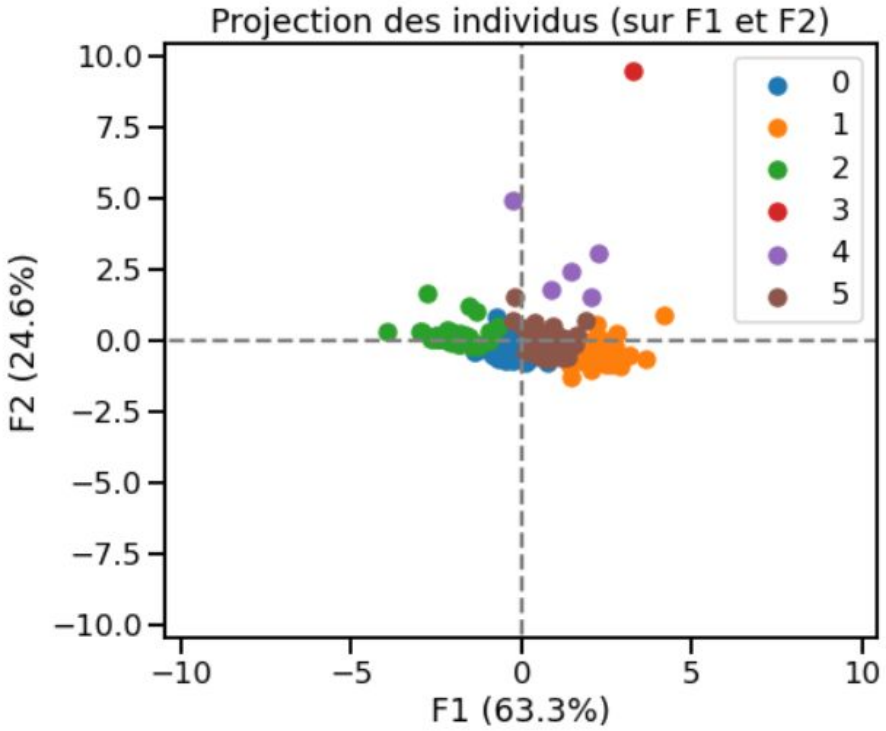
- un quatrième groupe intermédiaire c’ est formé au centre des axes, représente une disponibilité et croissance moyenne

K Means avec 5 clusters



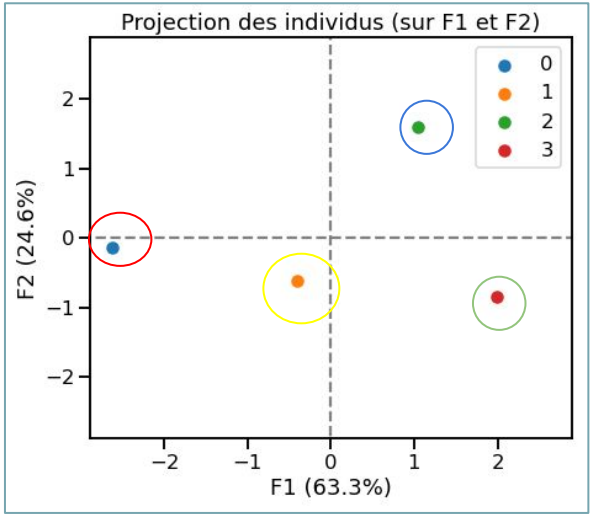
- de nouveaux le groupe situé au centre se divise en deux parties inégales parallèlement à l'axe F1 pour former deux groupes
- le G0 représente les pays ayant une croissance plus forte que le G4

K Means avec 6 clusters

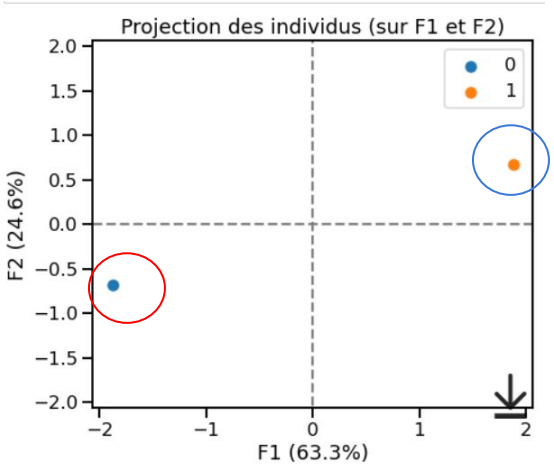


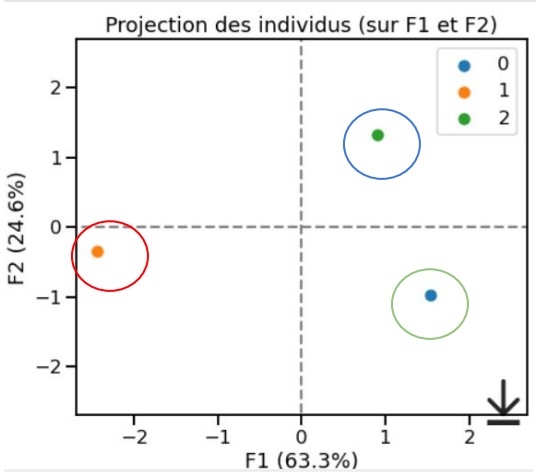
- à 6 clusters les groupes continuent de se diviser
- des class ayant un effectif très faible
- pas de grande différence entre les class

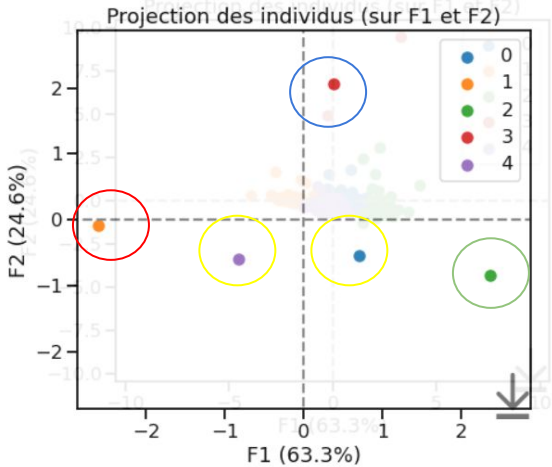
K Means (les centroïdes)

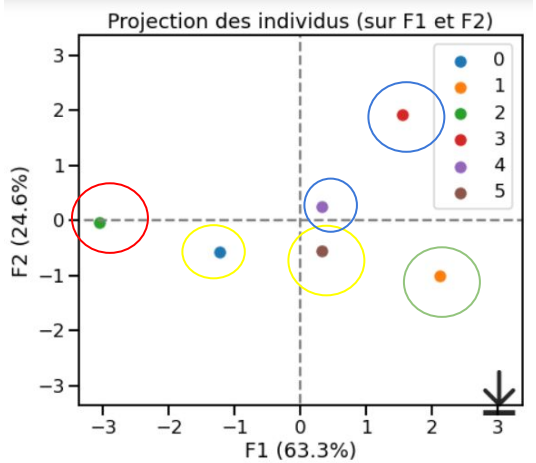


j’ai choisis 4 clusters car ils semblent être le plus stable minimise l’inertie interclass

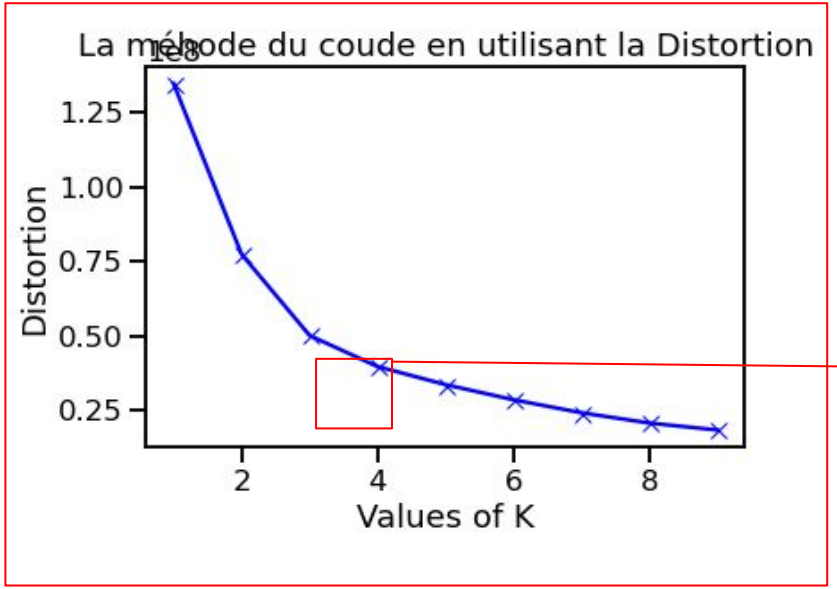


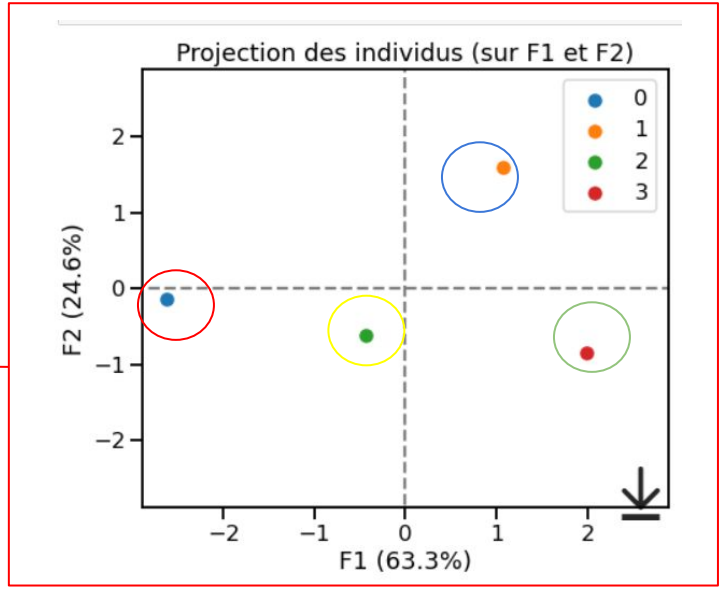






Pays ciblés



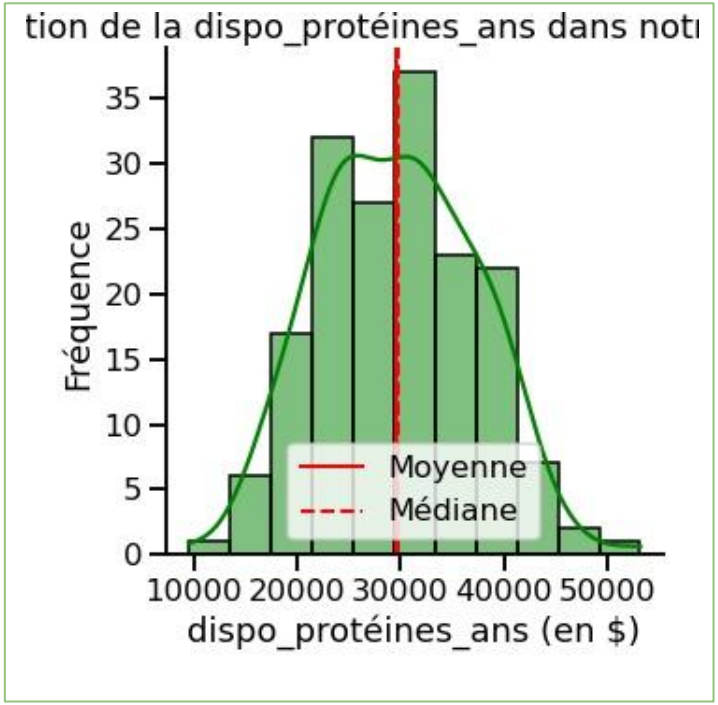


on remarque que l'inertie commence à se stabiliser à partir de K=4 donc je prend K=4 pour mes tests et pour choisir les pays

Test d'adéquation :la variables disponibilité en protéine/an

on effectue un **Test de Kolmogorov-Smirnov** pour la variable **disponibilité en protéines par ans**
on confirme avec le test **shapiro wilk**

- la p value étant supérieur à 5% pour ce niveau de test
- on ne peut pas rejeter l'hypothèse nul,
- donc **l'échantillon suit une lois normale**




```
#pour etre plus sur on effectue un test de shapiro
stats.shapiro(data_mrg_K4['dispo_protéines_ans'])
```

```
ShapiroResult(statistic=0.9916048049926758, pvalue=0.4014256000518799)
```

```
]: stats.kstest(data_mrg_K4["dispo_protéines_ans"],list(np.random.normal(np.mean(data_mrg_K4["dispo_protéines_ans"]), np.std(data_r
```

```
: KstestResult(statistic=0.076, pvalue=0.3375935653719374)
```

Test de comparaison entre deux populations (dans le cas gaussien) cluster 1 et 3

Avant que le test t puisse être effectué, je tester les hypothèses.D'abord pour tester l'homogénéité des variances. Pour ce faire, j'utilise le test d'homogénéité de la variance de Levene.

```
Le test d'égalité des variances :

{H0:  σ21=σ22
 H1:  σ21≠σ22
```

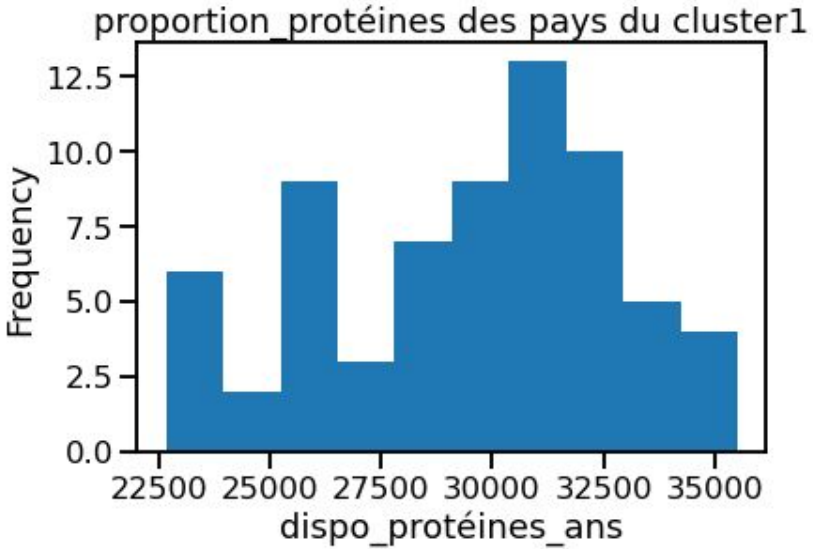
- la p value étant supérieur à 5% pour ce niveau de test on ne rejete pas l'hypothèse nulle c'est-à-dire il y a une homogénéité des variances,
- on continue les étapes suivantes pour tester l'hypothèse de normalité.

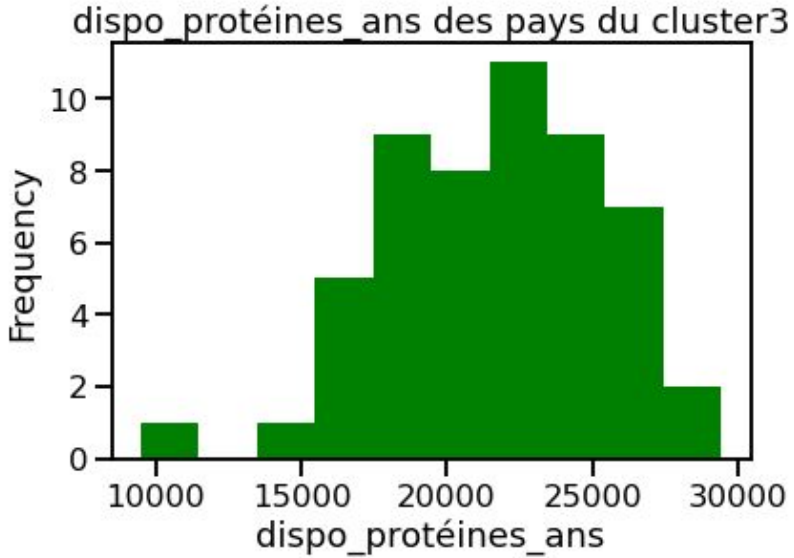
```
stats.levene(data_var1,data_var2)
```

```
LeveneResult(statistic=0.9543626855314353, pvalue=0.33059308348238403)
```

Test de comparaison de deux populations (dans le cas gaussien) cluster 3 et 4

égalité des moyennes





Test de comparaison entre deux populations (dans le cas gaussien) cluster 3 et 4

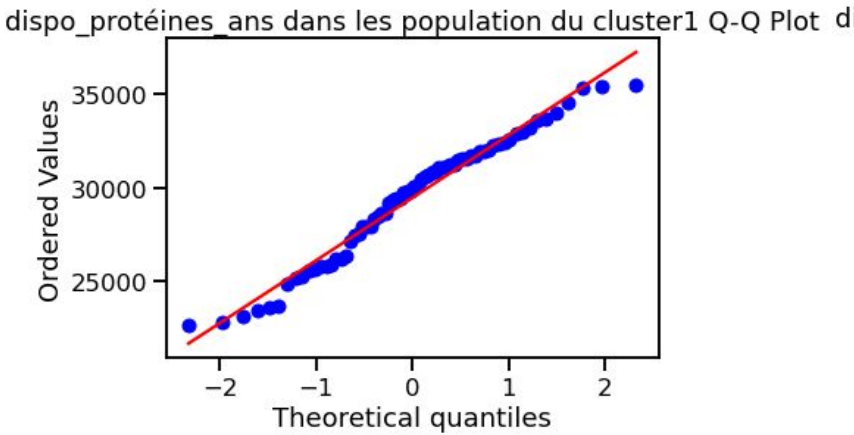
égalité des moyennes

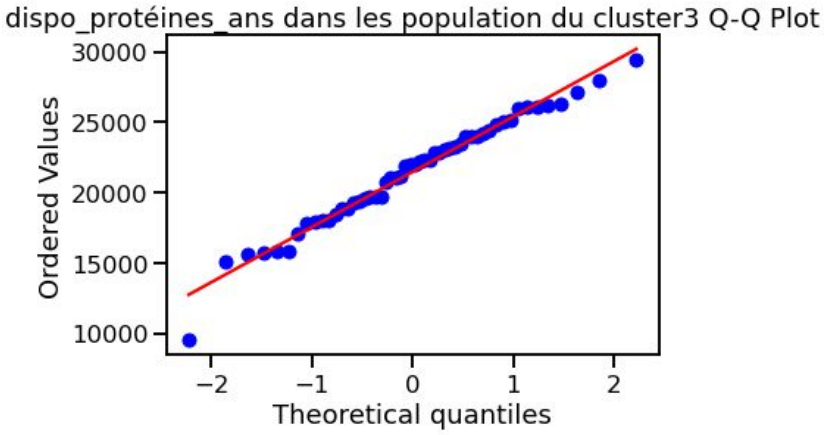
```
data_gb_means=data_mrg_K4.groupby("clusters_K4").mean()
data_gb_means
```

clusters_K4	dispo_protéines_ans	proportion_protéines	dispo_cal_ans	prc_croissance
0	38473.807692	0.571365	1.234065e+09	1.735096
1	29281.226119	0.477086	1.035276e+09	2.814539
2	26974.412500	0.332657	9.954462e+08	75.103448
3	21462.280769	0.267391	8.819312e+08	4.684985

Test de comparaison entre deux populations (dans le cas gaussien) cluster 1 et 3

pour avoir une idée de la normalité en visualisant les données sous forme de graphique q-q



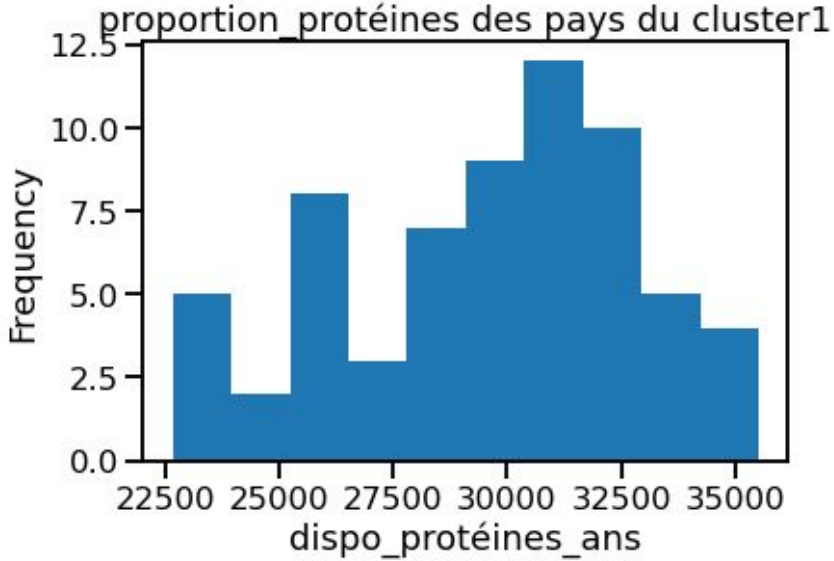


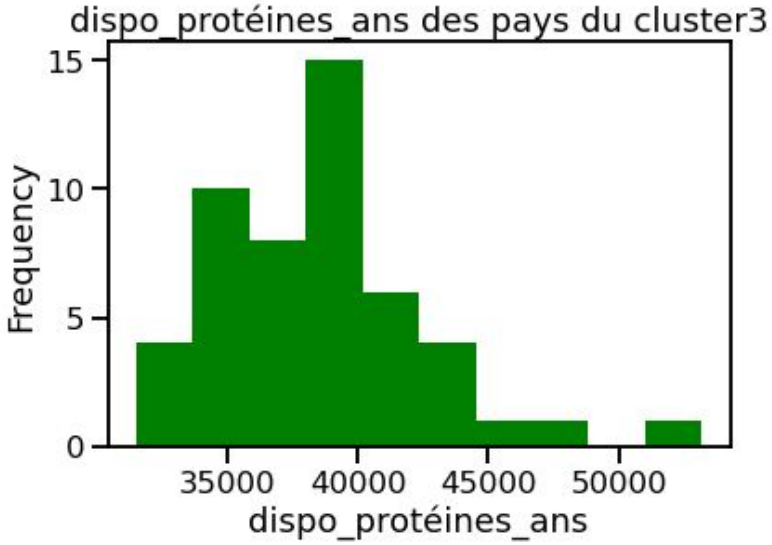
Il y a un certain écart par rapport à la normalité dans le diagramme q-q, mais cela ne semble pas être une grande violation. Dans l'ensemble, les données semblent avoir une normalité

Test de comparaison de deux populations (dans le cas gaussien) cluster 1 et 3

tester l'égalité des moyennes

Tout d'abord, je vais les vérifier visuellement.





Test de comparaison de deux populations (dans le cas gaussien) cluster 1 et 3

```
Le test d'égalité des moyennes :

{H0:μ1=μ2
 H1:μ1<>μ2
```

test statistique en utilisant le test de normalité de Shapiro-Wilk.

```
stats.shapiro(data_eff1['dispo_protéines_ans'])
```

```
ShapiroResult(statistic=0.9690171480178833, pvalue=0.09285737574100494)
```

```
stats.shapiro(data_eff2['dispo_protéines_ans'])
```

```
ShapiroResult(statistic=0.9788655638694763, pvalue=0.47880813479423523)
```

aucun des tests de normalité n'est significatif (p value>5%), ce qui signifie qu'on ne peut pas rejeter l'hypothèse de normalité.
on continue avec le test t-indépendant

Test de comparaison de deux populations (dans le cas gaussien) cluster 1 et 3

test t indépendant

```
stats.ttest_ind(data_eff1['dispo_protéines_ans'], data_eff2['dispo_protéines_ans'], equal_var=True)
```

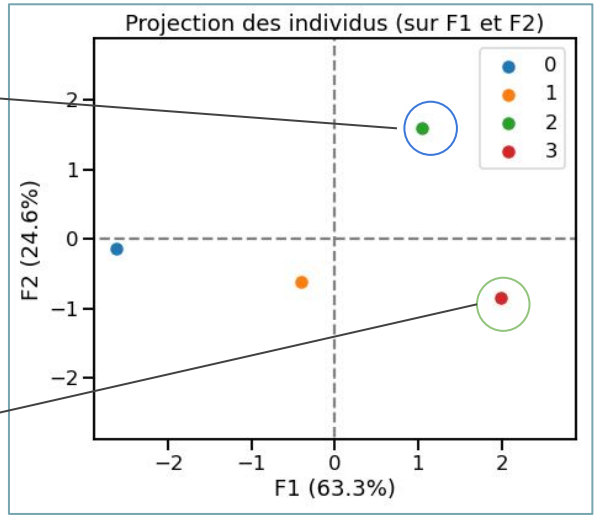
```
Ttest_indResult(statistic=11.904277661676698, pvalue=6.819949158307224e-22)
```

- on a la p value largement <5% un résultat signifiant ce qui nous mène à rejeter l'hypothèse H0 en faveur de l'hypothèse alternative,
- les deux cluster sont indépendants.
- on conclut a une différence différence des moyens entre les populations des deux clusters 1 et 3 .

pays choisis

Area
China
India
Nigeria

```
array(['Afghanistan', 'Angola', 'Bangladesh', 'Benin', 'Burundi',
       "Côte d'Ivoire", 'Cambodia', 'Cameroon',
       'Central African Republic', 'Comoros', 'Congo',
       "Democratic People's Republic of Korea",
       'Democratic Republic of the Congo', 'Djibouti', 'Eswatini',
       'Ethiopia', 'Gambia', 'Ghana', 'Guatemala', 'Guinea',
       'Guinea-Bissau', 'Haiti', 'Honduras', 'Iraq', 'Jordan', 'Kenya',
       "Lao People's Democratic Republic", 'Lesotho', 'Liberia',
       'Madagascar', 'Malawi', 'Mozambique', 'Namibia', 'Nepal',
       'Nicaragua', 'Pakistan', 'Rwanda', 'Sao Tome and Principe',
       'Senegal', 'Sierra Leone', 'Solomon Islands', 'Sri Lanka',
       'Syrian Arab Republic', 'Tajikistan', 'Timor-Leste', 'Togo',
       'Uganda', 'United Republic of Tanzania',
       'Venezuela (Bolivarian Republic of)', 'Yemen', 'Zambia',
       'Zimbabwe'], dtype=object)
```



pays choisis

le choix c’est porté sur 25 Pays d’afriques ainsi que l’inde et la chine

Conclusion

- deux axes à prendre en considération F1 et F2
- deux variable variables principales peuvent résumer toutes les variables :la disponibilité (sur l'axe F1 et la croissance (sur l'axe F2).
- la classification la plus adapté est le clustering en en 4 groupe par le k means
- les pays qu'on doit prendre en compte pour notre étude sont ceux faisant partie du 3eme et 4eme cluster représentant une faible disponibilité en protéines et une forte croissance.
- existe une différence significative entre les populations des deux clusters 1 et 3.