



 DATA BASE

DATA MINING



ANALYTICS



PRE-PROCESSING

MINI PROJET 2



EVALUATION

**Clustering Hiérarchique pour l'Étude de
l'Impact du COVID-19 en Europe : Analyse
Médicale, Économique et Sociologique**

Prepared for :

- Mme AOUDJIT
- Mme OUDJOUDI

Prepared by :

- Anis BENINI

Sommaire

I. Introduction.....	3
II. Structure du code.....	3
1. Importation des Bibliothèques.....	4
2. Chargement des données.....	5
3. Exploration et Nettoyage des Données.....	5
a) Exploration des données.....	5
b) Nettoyage des données.....	5
4. Analyse descriptive.....	6
5. Préparation des Données pour le Clustering.....	7
6. Clustering Hiérarchique et Dendrogramme.....	7
7. Visualisation des Résultats.....	9
III. Conclusion.....	10

I. Introduction :

Le projet vise à analyser l'impact médical, économique et sociologique de la pandémie de COVID-19 en Europe à l'aide du clustering hiérarchique. L'objectif principal est d'identifier les pays les plus touchés en termes de nouveaux cas et décès signalés quotidiennement.

Les étapes suivies pour la réalisation de ce projet :

- Charger et explorer un dataset contenant les données de **COVID-19** pour les pays européens.
 - Normaliser les données pour éliminer les biais liés aux échelles différentes.
 - Appliquer un clustering hiérarchique pour regrouper les pays en fonction de la prévalence de la pandémie.
 - Visualiser les résultats à l'aide de dendrogrammes et graphiques.
-

II. Structure du code

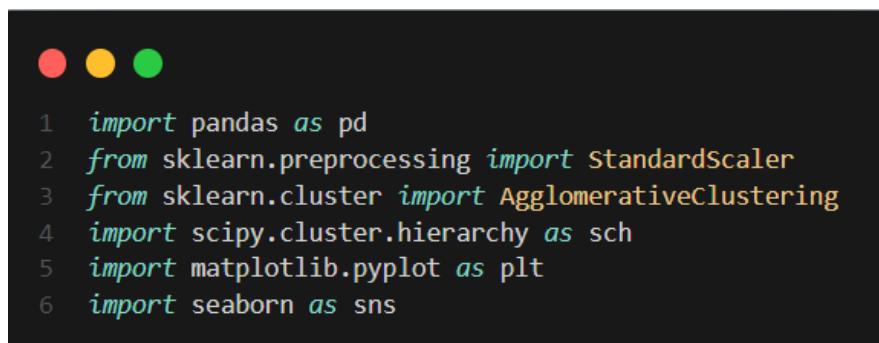
Le code est divisé en plusieurs sections principales :

1. Importation des Bibliothèques
2. Chargement des Données
3. Exploration et Nettoyage des Données
4. Analyse Descriptive
5. Préparation des Données pour le Clustering
6. Clustering Hiérarchique et Dendrogramme
7. Visualisation des Résultats

1. Importation des Bibliothèques :

Les bibliothèques suivantes sont importées pour effectuer les différentes tâches :

- **Pandas** : Pour la manipulation des données.
- **StandardScaler** : Pour la normalisation des données.
- **AgglomerativeClustering** : Pour le clustering hiérarchique.
- **Scipy.cluster.hierarchy** : Pour la création de dendrogrammes.
- **Matplotlib.pyplot & Seaborn** : Pour la visualisation des données.



```
1 import pandas as pd
2 from sklearn.preprocessing import StandardScaler
3 from sklearn.cluster import AgglomerativeClustering
4 import scipy.cluster.hierarchy as sch
5 import matplotlib.pyplot as plt
6 import seaborn as sns
```

2. Chargement des données :

Les données sont chargées à partir d'un fichier CSV <**data-Clustering Hiérarchique.csv**>. Le fichier contient des informations sur les cas de santé (nombre de cas, décès, population, etc.) par pays et par date.

	dateRep	day	month	year	cases	deaths	countriesAndTerritories	geoId	countryterritoryCode	popData2020	continentExp
0	23/10/2022	23	10	2022	3557.0	0.0	Austria	AT	AUT	8901064	Europe
1	22/10/2022	22	10	2022	5494.0	4.0	Austria	AT	AUT	8901064	Europe
2	21/10/2022	21	10	2022	7776.0	4.0	Austria	AT	AUT	8901064	Europe
3	20/10/2022	20	10	2022	8221.0	6.0	Austria	AT	AUT	8901064	Europe
4	19/10/2022	19	10	2022	10007.0	8.0	Austria	AT	AUT	8901064	Europe

3. Exploration et Nettoyage des Données :

a) Exploration des données :

→ Informations sur les données : Le code utilise `data.info()` affiche des informations sur les colonnes, les types de données, et les valeurs manquantes.

→ Statistiques descriptives : `data.describe()` est utilisé pour obtenir des statistiques de base comme la moyenne, l'écart-type, les valeurs minimales et maximales.

→ Valeurs manquantes : Le code vérifie les valeurs manquantes par colonnes avec `data.isnull().sum()`.

b) Nettoyage des données :

→ Suppression des valeurs aberrantes : On a fait ce script pour éviter d'avoir des lignes où les cas ou les décès étaient négatifs et les supprimer si c'est le cas.

→ Remplacement des valeurs manquantes : Les valeurs manquantes dans cases et deaths ont été remplacées par 0.

```
● ● ●
1 # Supprimer Les valeurs aberrantes (cas et décès négatifs)
2 df_cleaned = data[(data['cases'] >= 0) & (data['deaths'] >= 0)].copy()
3
4 # Remplacer Les valeurs manquantes par 0 pour `cases` et `deaths`
5 df_cleaned['cases'] = df_cleaned['cases'].fillna(0)
6 df_cleaned['deaths'] = df_cleaned['deaths'].fillna(0)
```

Après le nettoyage, les données sont prêtes pour l'analyse. Les valeurs aberrantes et manquantes ont été traitées avec succès .

Les valeurs manquantes :

<code>dateRep</code>	0
<code>day</code>	0
<code>month</code>	0
<code>year</code>	0
<code>cases</code>	93
<code>deaths</code>	292
<code>countriesAndTerritories</code>	0
<code>geoId</code>	0
<code>countryterritoryCode</code>	0
<code>popData2020</code>	0
<code>continentExp</code>	0
<code>dtype: int64</code>	



Valeurs manquantes après nettoyage :

<code>dateRep</code>	0
<code>day</code>	0
<code>month</code>	0
<code>year</code>	0
<code>cases</code>	0
<code>deaths</code>	0
<code>countriesAndTerritories</code>	0
<code>geoId</code>	0
<code>countryterritoryCode</code>	0
<code>popData2020</code>	0
<code>continentExp</code>	0
<code>dtype: int64</code>	

4. Analyse Descriptive :

Dans cette étape on a agrégé les données par pays pour calculer les totaux de cas et de décès, ainsi que les taux d'incidence et de mortalité.

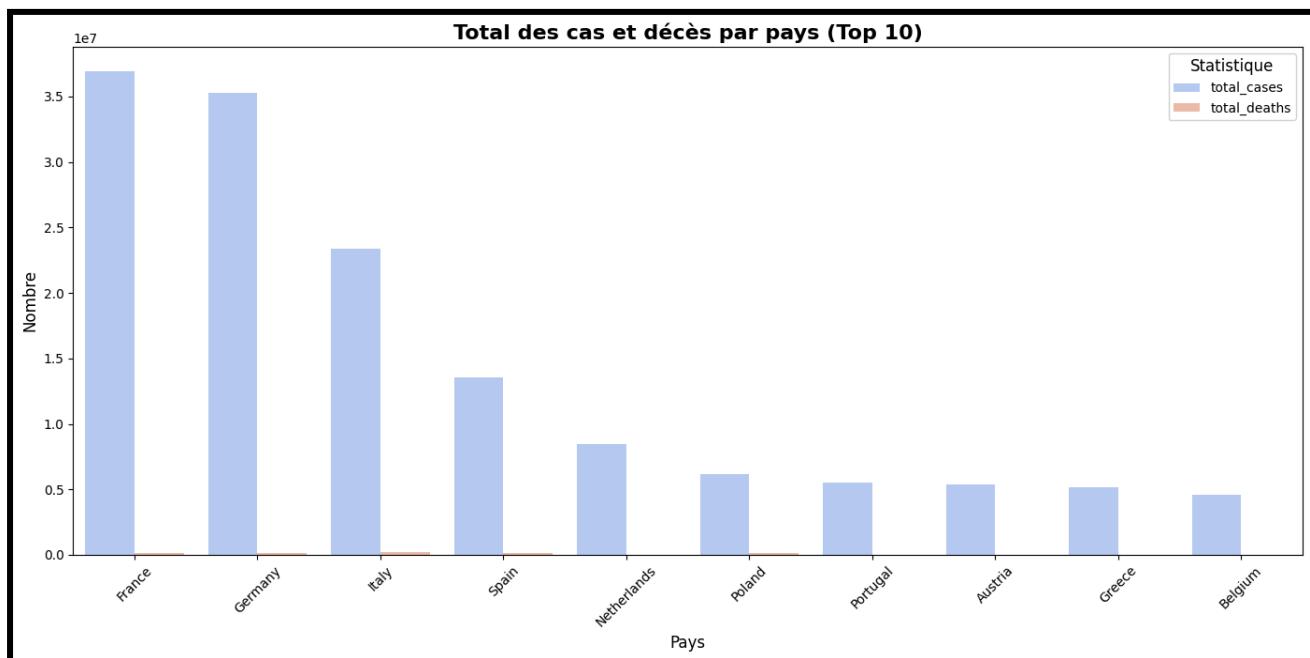
- **groupby()** : Regroupe les données par pays.
 - **agg()** : Applique des fonctions d'agrégation (somme des cas, somme des décès, etc.).
 - **Calcul des taux** : Les taux d'incidence et de mortalité sont calculés en divisant le nombre total de cas ou de décès par la population, puis en multipliant par **100 000**.
- **Résultat:** Le DataFrame **country_stats** contient maintenant des informations agrégées par pays, avec les taux d'incidence et de mortalité. Cela nous permet de comparer les pays en fonction de ces indicateurs.

→ **Output :**

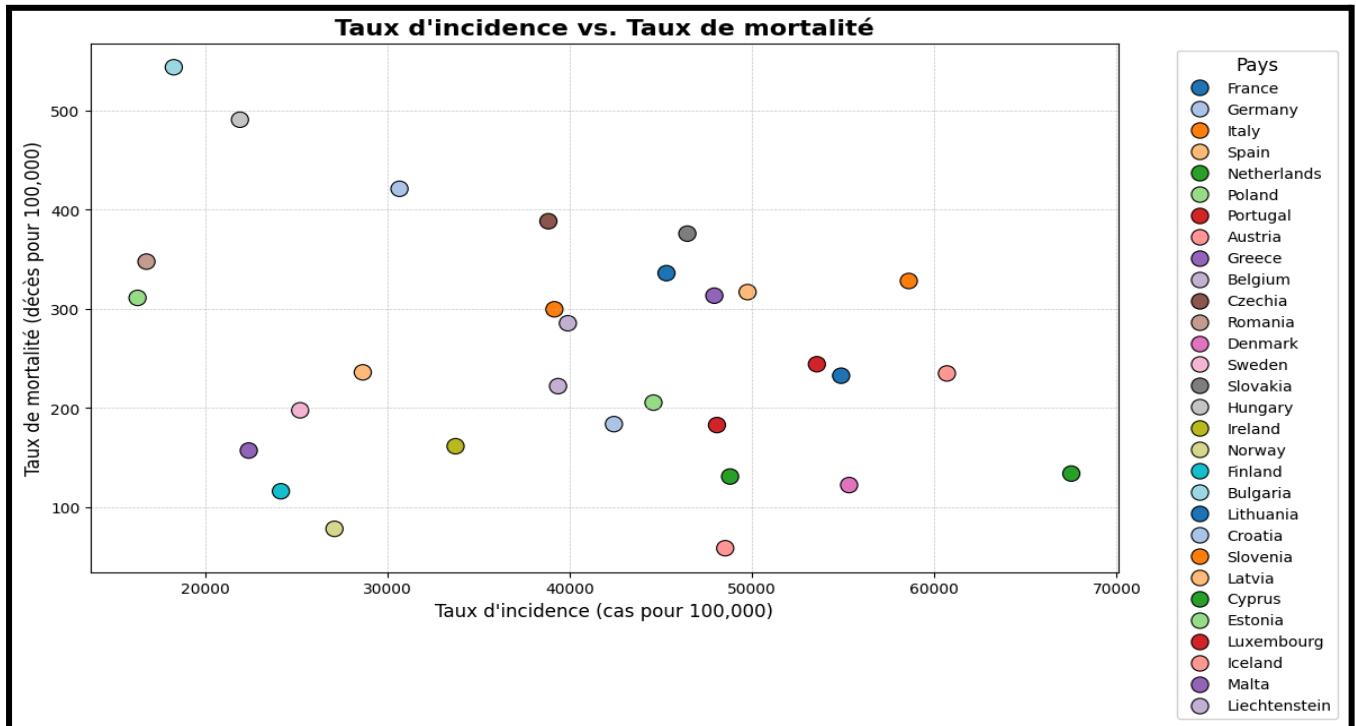
Stats de chaque pays :				
	countriesAndTerritories	total_cases	total_deaths	population
9	France	36952159.0	156518.0	67320216
10	Germany	35287690.0	152753.0	83166711
15	Italy	23359251.0	178617.0	59641488
28	Spain	13564823.0	111649.0	47332614
21	Netherlands	8494705.0	22771.0	17407585
		incidence_rate	mortality_rate	
9		54890.137310	232.497769	
10		42430.065558	183.670844	
15		39166.110343	299.484480	
28		28658.512289	235.881754	
21		48798.871297	130.810793	

Et pour visualiser tous ca, on utilisé deux visualisations :

- **Barplot des cas totaux et des décès** : il compare le nombre total de cas et de décès pour les 10 premiers pays en termes de données disponibles. Les données sont reformulées avec **melt** pour permettre une distinction entre les deux variables (cas et décès) à l'aide de couleurs contrastées.



- Scatter plot des taux d'incidence et de mortalité : il montre une corrélation possible entre le taux d'incidence (cas pour 100 000 habitants) et le taux de mortalité (décès pour 100 000 habitants) pour tous les pays. Chaque point représente un pays, avec une couleur unique pour chaque pays afin de faciliter leur identification. Des ajustements comme la grille, les bordures, et la légende permettent de rendre ce graphique plus informatif et lisible.



5. Préparation des Données pour le Clustering :

Les données doivent être normalisées avant d'appliquer le clustering hiérarchique, car les variables peuvent avoir des échelles différentes.

- StandardScaler() : Normalise les données pour que chaque variable ait une moyenne de **0** et un écart-type de **1**.
- fit_transform() : Appliquer la normalisation aux colonnes **incidence_rate** et **mortality_rate**.

6. Clustering Hiérarchique et Dendrogramme :

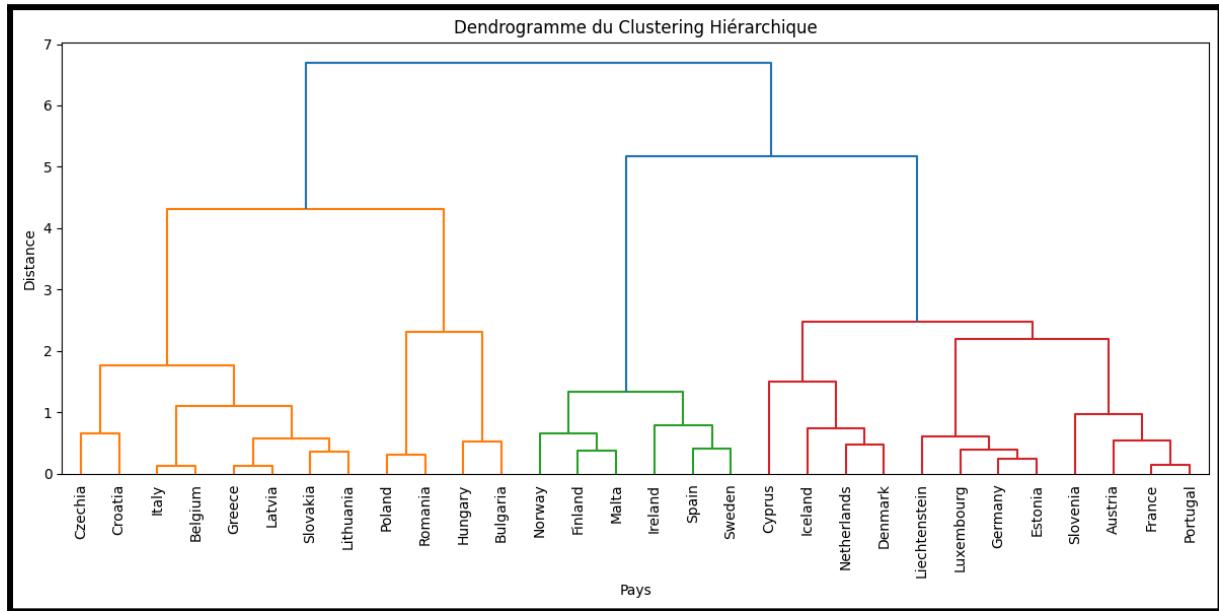
Le **clustering hiérarchique** est une méthode d'apprentissage non supervisé utilisée pour regrouper des données en fonction de leur similarité en clusters basée sur la distance entre les points dans notre cas, les taux d'incidence et de mortalité des pays sont utilisés pour regrouper les pays en clusters. Les données sont d'abord normalisées pour s'assurer que toutes les variables ont la même échelle, ce qui est essentiel pour éviter que certaines variables dominent le clustering, ensuite le clustering hiérarchique est appliqué en utilisant la méthode de liaison "**WARD**", qui minimise la variance intra-clusters.

Contrairement à d'autres algorithmes comme le **k-means**, il n'exige pas de spécifier à l'avance le nombre de clusters. Il fonctionne en construisant une structure hiérarchique, souvent

représentée sous forme d'un **arbre** (ou **dendrogramme**) , où chaque observation commence comme un cluster individuel, et ces clusters sont fusionnés progressivement en fonction de leur proximité jusqu'à former un seul cluster global.

Enfin, l'arbre hiérarchique est découpé en un nombre fixé de clusters (par exemple k=4) , et les résultats sont ajoutés au DataFrame pour une analyse ultérieure.

- Notre dendrogramme est créée avec **sch.dendrogram()**
et **sch.linkage()** : Calcule la matrice de liaison pour le clustering hiérarchique.



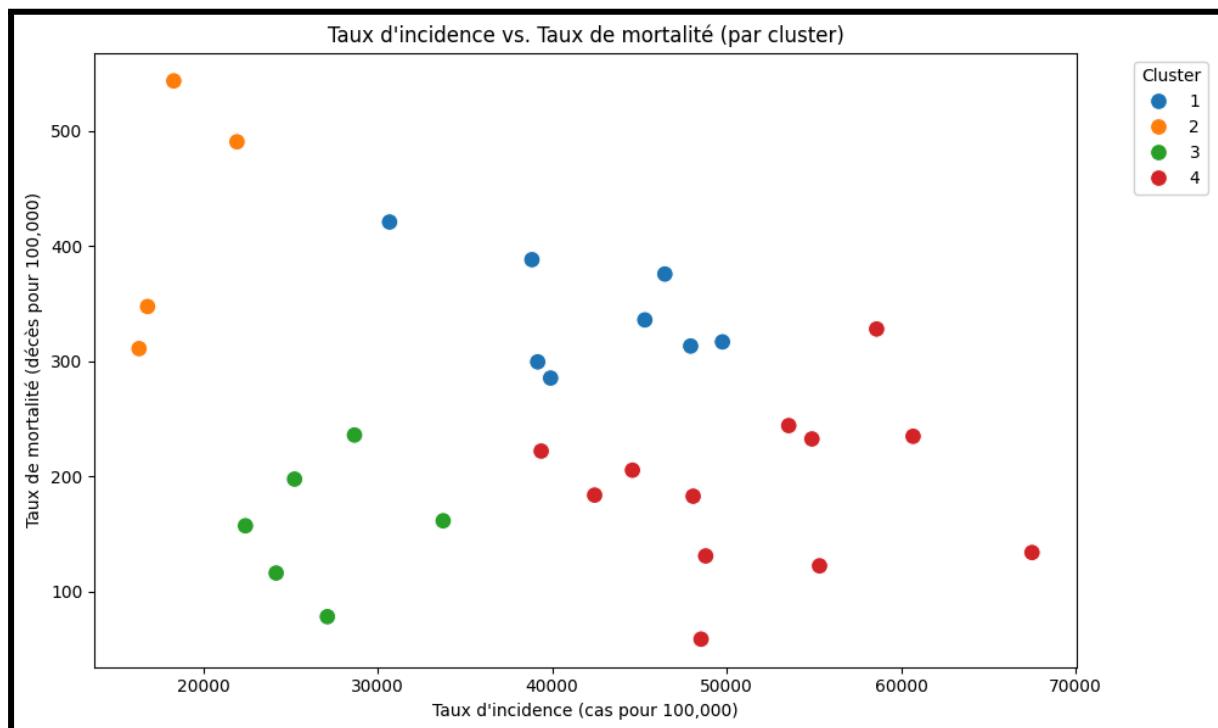
- Un tableau montre chaque pays et son cluster associé, trié par cluster.

Pays et leurs clusters :		
	countriesAndTerritories	cluster
26	Slovakia	1
16	Latvia	1
18	Lithuania	1
1	Belgium	1
11	Greece	1
5	Czechia	1
15	Italy	1
3	Croatia	1
25	Romania	2
12	Hungary	2
2	Bulgaria	2
23	Poland	2
28	Spain	3
29	Sweden	3
20	Malta	3
14	Ireland	3
22	Norway	3
8	Finland	3
7	Estonia	4
4	Cyprus	4
19	Luxembourg	4
13	Iceland	4
...		
21	Netherlands	4
10	Germany	4

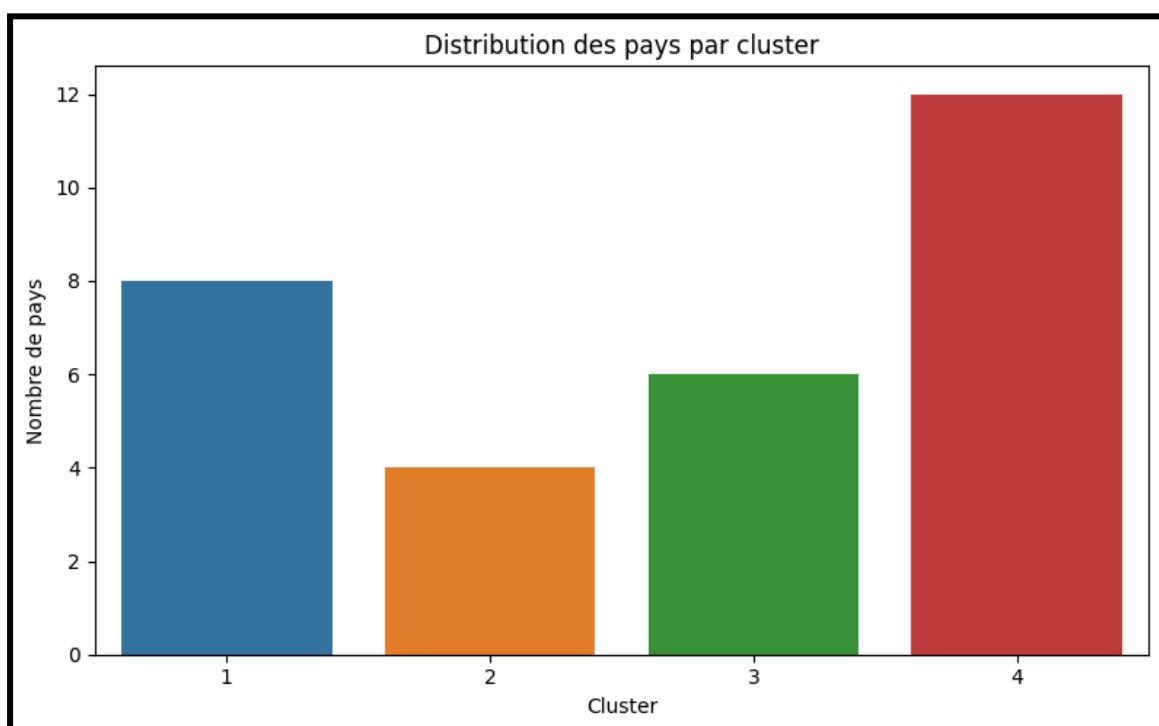
7. Visualisation des Résultats :

Pour mieux comprendre les résultats du clustering, deux visualisations sont utilisées : un scatter plot et un barplot.

→ **Le scatter plot** montre les clusters de pays en fonction de leurs taux d'incidence et de mortalité. Chaque point représente un pays, et les couleurs représentent les clusters. Cela permet de voir clairement les regroupements et les relations entre les clusters.



→ **Un barplot** est utilisé pour visualiser la distribution des pays dans chaque cluster. Ce graphique montre combien de pays se trouvent dans chaque cluster, ce qui donne une idée de la taille et de la répartition des clusters.



III. Conclusion

Le travail réalisé dans ce projet a permis d'explorer et d'appliquer des techniques de clustering hiérarchique sur un ensemble de données liées aux cas et aux décès de COVID-19 dans différents pays. À travers l'analyse des données, nous avons pu identifier des tendances et des regroupements naturels parmi les pays en fonction de leur incidence et de leur taux de mortalité. Cette approche a mis en lumière des similarités et des différences significatives entre les pays, offrant ainsi une perspective utile pour comprendre l'impact de la pandémie à l'échelle mondiale.

L'utilisation de méthodes de prétraitement des données, telles que le nettoyage des valeurs aberrantes et le remplacement des valeurs manquantes, a été essentielle pour garantir la qualité des résultats. De plus, l'application de l'algorithme de clustering hiérarchique a permis de visualiser les relations entre les pays sous forme de dendrogrammes, facilitant ainsi l'interprétation des regroupements.

Enfin, ce projet a également souligné l'importance de l'analyse exploratoire des données et de la visualisation dans le processus de découverte de connaissances. Les résultats obtenus peuvent servir de base pour des études plus approfondies, notamment pour identifier les facteurs sous-jacents qui influencent les différences observées entre les pays. Ce travail démontre l'utilité des techniques de clustering dans l'analyse de données épidémiologiques et ouvre la voie à des applications futures dans le domaine de la santé publique.