Actividad: Análisis exploratorio con técnicas de agrupamiento

In [29]:
```
#import libreries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.datasets import make_blobs
# clustering methods
from sklearn.cluster import KMeans
from sklearn.cluster import AgglomerativeClustering
from sklearn.cluster import SpectralClustering
from sklearn.cluster import OPTICS
from sklearn.cluster import DBSCAN
from scipy.cluster.hierarchy import dendrogram, linkage
# Metrics for evaluating clustering results
from sklearn.metrics import adjusted_rand_score
from sklearn.metrics import silhouette_score
from sklearn.metrics import calinski_harabasz_score
from sklearn.metrics import davies_bouldin_score
# Distance metrics
from sklearn.metrics import pairwise_distances
# SOM
from minisom import MiniSom
#read csv
df = pd.read_csv('Country-data.csv', delimiter=',')
df
```

Out[29]:

	country	child_mort	exports	health	imports	income	inflation	life_expe
0	Afghanistan	90.2	10.0	7.58	44.9	1610	9.44	56.
1	Albania	16.6	28.0	6.55	48.6	9930	4.49	76.
2	Algeria	27.3	38.4	4.17	31.4	12900	16.10	76.
3	Angola	119.0	62.3	2.85	42.9	5900	22.40	60.
4	Antigua and Barbuda	10.3	45.5	6.03	58.9	19100	1.44	76.
...	...	...	...	...	...	...	...	.
162	Vanuatu	29.2	46.6	5.25	52.7	2950	2.62	63.
163	Venezuela	17.1	28.5	4.91	17.6	16500	45.90	75.
164	Vietnam	23.3	72.0	6.84	80.2	4490	12.10	73.
165	Yemen	56.3	30.0	5.18	34.4	4480	23.60	67.
166	Zambia	83.1	37.0	5.89	30.9	3280	14.00	52.

167 rows × 10 columns

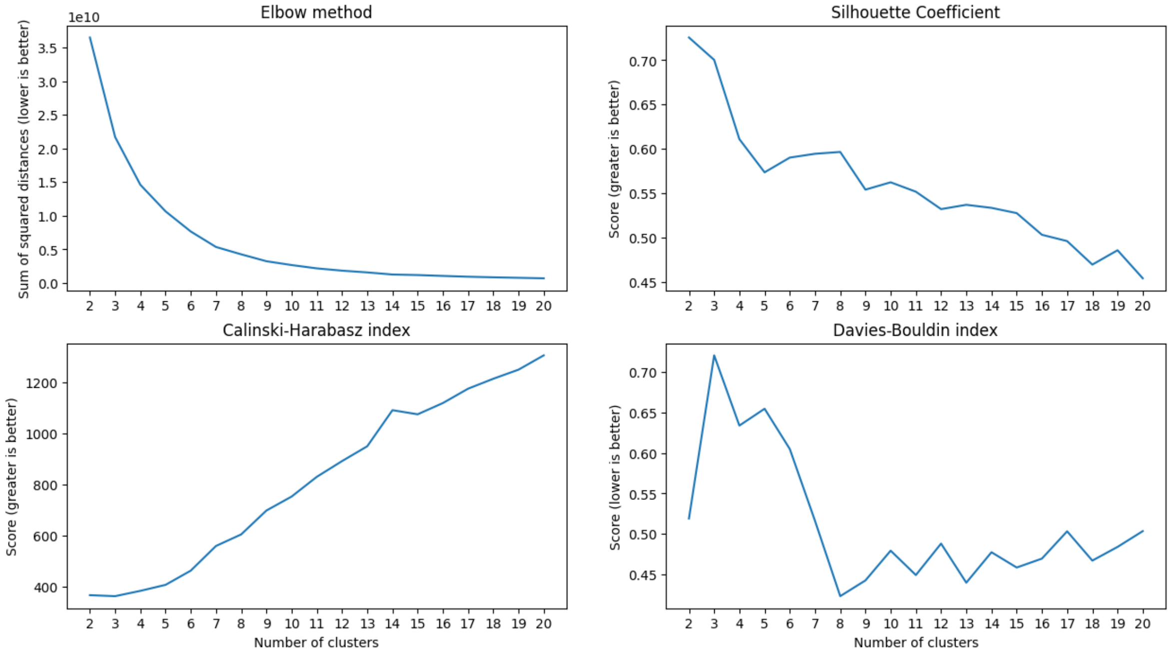
In [30]:
```
#declare x and y
x = df.iloc[:, 1:].values
y = df.iloc[:,0].values
```

1 Aplica k-medias sobre le conjunto de datos para generar un agrupamiento para los países de la base de datos. Utiliza al menos dos métodos para estimar el número óptimo de grupos.

In [5]:
```
# Optimal number of clusters
sum_of_squared_distances = []
sscore = []
chscore = []
dbscore = []
ks = np.arange(2, 21)
for k in ks: figsize=(15, 8)
    # Find clustering model
    kmeans = KMeans(n_clusters=k).fit(x)
    # Evaluate sum of squared distances
    sum_of_squared_distances.append(kmeans.inertia_)
    # Evaluate Silhouette score
    sscore.append(silhouette_score(x, kmeans.labels_))
    # Evaluate Calinski-Harabasz index
    chscore.append(calinski_harabasz_score(x, kmeans.labels_))
    # Evaluate Davies-Bouldin index
    dbscore.append(davies_bouldin_score(x, kmeans.labels_))
fig, axs = plt.subplots(2, 2, figsize=(15, 8))
axs[0][0].plot(ks, sum_of_squared_distances)
axs[0][0].set_ylabel('Sum of squared distances (lower is better)')
axs[0][0].set_title('Elbow method')
axs[0][0].set_xticks(ks)
axs[0][1].plot(ks, sscore)
axs[0][1].set_ylabel('Score (greater is better)')
axs[0][1].set_title('Silhouette Coefficient')
axs[0][1].set_xticks(ks)
axs[1][0].plot(ks, chscore)
axs[1][0].set_xlabel('Number of clusters')
axs[1][0].set_ylabel('Score (greater is better)')
axs[1][0].set_title('Calinski-Harabasz index')
axs[1][0].set_xticks(ks)
axs[1][1].plot(ks, dbscore)
axs[1][1].set_xlabel('Number of clusters')
axs[1][1].set_ylabel('Score (lower is better)')
axs[1][1].set_title('Davies-Bouldin index')
axs[1][1].set_xticks(ks)
plt.show()
```

```
/home/alanv/Documents/7/mate/venv/lib/python3.11/site-packages/sklearn/clu
ster/_kmeans.py:1412: FutureWarning: The default value of `n_init` will ch
ange from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to sup
press the warning
  super()._check_params_vs_input(X, default_n_init=10)
/home/alanv/Documents/7/mate/venv/lib/python3.11/site-packages/sklearn/clu
ster/_kmeans.py:1412: FutureWarning: The default value of `n_init` will ch
ange from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to sup
press the warning
  super()._check_params_vs_input(X, default_n_init=10)
/home/alanv/Documents/7/mate/venv/lib/python3.11/site-packages/sklearn/clu
ster/_kmeans.py:1412: FutureWarning: The default value of `n_init` will ch
ange from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to sup
press the warning
  super()._check_params_vs_input(X, default_n_init=10)
/home/alanv/Documents/7/mate/venv/lib/python3.11/site-packages/sklearn/clu
ster/_kmeans.py:1412: FutureWarning: The default value of `n_init` will ch
ange from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to sup
press the warning
  super()._check_params_vs_input(X, default_n_init=10)
/home/alanv/Documents/7/mate/venv/lib/python3.11/site-packages/sklearn/clu
ster/_kmeans.py:1412: FutureWarning: The default value of `n_init` will ch
ange from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to sup
press the warning
  super()._check_params_vs_input(X, default_n_init=10)
/home/alanv/Documents/7/mate/venv/lib/python3.11/site-packages/sklearn/clu
ster/_kmeans.py:1412: FutureWarning: The default value of `n_init` will ch
ange from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to sup
press the warning
  super()._check_params_vs_input(X, default_n_init=10)
/home/alanv/Documents/7/mate/venv/lib/python3.11/site-packages/sklearn/clu
ster/_kmeans.py:1412: FutureWarning: The default value of `n_init` will ch
ange from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to sup
press the warning
  super()._check_params_vs_input(X, default_n_init=10)
/home/alanv/Documents/7/mate/venv/lib/python3.11/site-packages/sklearn/clu
ster/_kmeans.py:1412: FutureWarning: The default value of `n_init` will ch
ange from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to sup
press the warning
  super()._check_params_vs_input(X, default_n_init=10)
/home/alanv/Documents/7/mate/venv/lib/python3.11/site-packages/sklearn/clu
ster/_kmeans.py:1412: FutureWarning: The default value of `n_init` will ch
ange from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to sup
press the warning
  super()._check_params_vs_input(X, default_n_init=10)
/home/alanv/Documents/7/mate/venv/lib/python3.11/site-packages/sklearn/clu
ster/_kmeans.py:1412: FutureWarning: The default value of `n_init` will ch
ange from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to sup
press the warning
  super()._check_params_vs_input(X, default_n_init=10)
/home/alanv/Documents/7/mate/venv/lib/python3.11/site-packages/sklearn/clu
ster/_kmeans.py:1412: FutureWarning: The default value of `n_init` will ch
ange from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to sup
press the warning
  super()._check_params_vs_input(X, default_n_init=10)
/home/alanv/Documents/7/mate/venv/lib/python3.11/site-packages/sklearn/clu
ster/_kmeans.py:1412: FutureWarning: The default value of `n_init` will ch
ange from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to sup
press the warning
  super()._check_params_vs_input(X, default_n_init=10)
```

```
/home/alanv/Documents/7/mate/venv/lib/python3.11/site-packages/sklearn/clu
ster/_kmeans.py:1412: FutureWarning: The default value of `n_init` will ch
ange from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to sup
press the warning
  super()._check_params_vs_input(X, default_n_init=10)
/home/alanv/Documents/7/mate/venv/lib/python3.11/site-packages/sklearn/clu
ster/_kmeans.py:1412: FutureWarning: The default value of `n_init` will ch
ange from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to sup
press the warning
  super()._check_params_vs_input(X, default_n_init=10)
/home/alanv/Documents/7/mate/venv/lib/python3.11/site-packages/sklearn/clu
ster/_kmeans.py:1412: FutureWarning: The default value of `n_init` will ch
ange from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to sup
press the warning
  super()._check_params_vs_input(X, default_n_init=10)
/home/alanv/Documents/7/mate/venv/lib/python3.11/site-packages/sklearn/clu
ster/_kmeans.py:1412: FutureWarning: The default value of `n_init` will ch
ange from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to sup
press the warning
  super()._check_params_vs_input(X, default_n_init=10)
/home/alanv/Documents/7/mate/venv/lib/python3.11/site-packages/sklearn/clu
ster/_kmeans.py:1412: FutureWarning: The default value of `n_init` will ch
ange from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to sup
press the warning
  super()._check_params_vs_input(X, default_n_init=10)
/home/alanv/Documents/7/mate/venv/lib/python3.11/site-packages/sklearn/clu
ster/_kmeans.py:1412: FutureWarning: The default value of `n_init` will ch
ange from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to sup
press the warning
  super()._check_params_vs_input(X, default_n_init=10)
/home/alanv/Documents/7/mate/venv/lib/python3.11/site-packages/sklearn/clu
ster/_kmeans.py:1412: FutureWarning: The default value of `n_init` will ch
ange from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to sup
press the warning
  super()._check_params_vs_input(X, default_n_init=10)
```

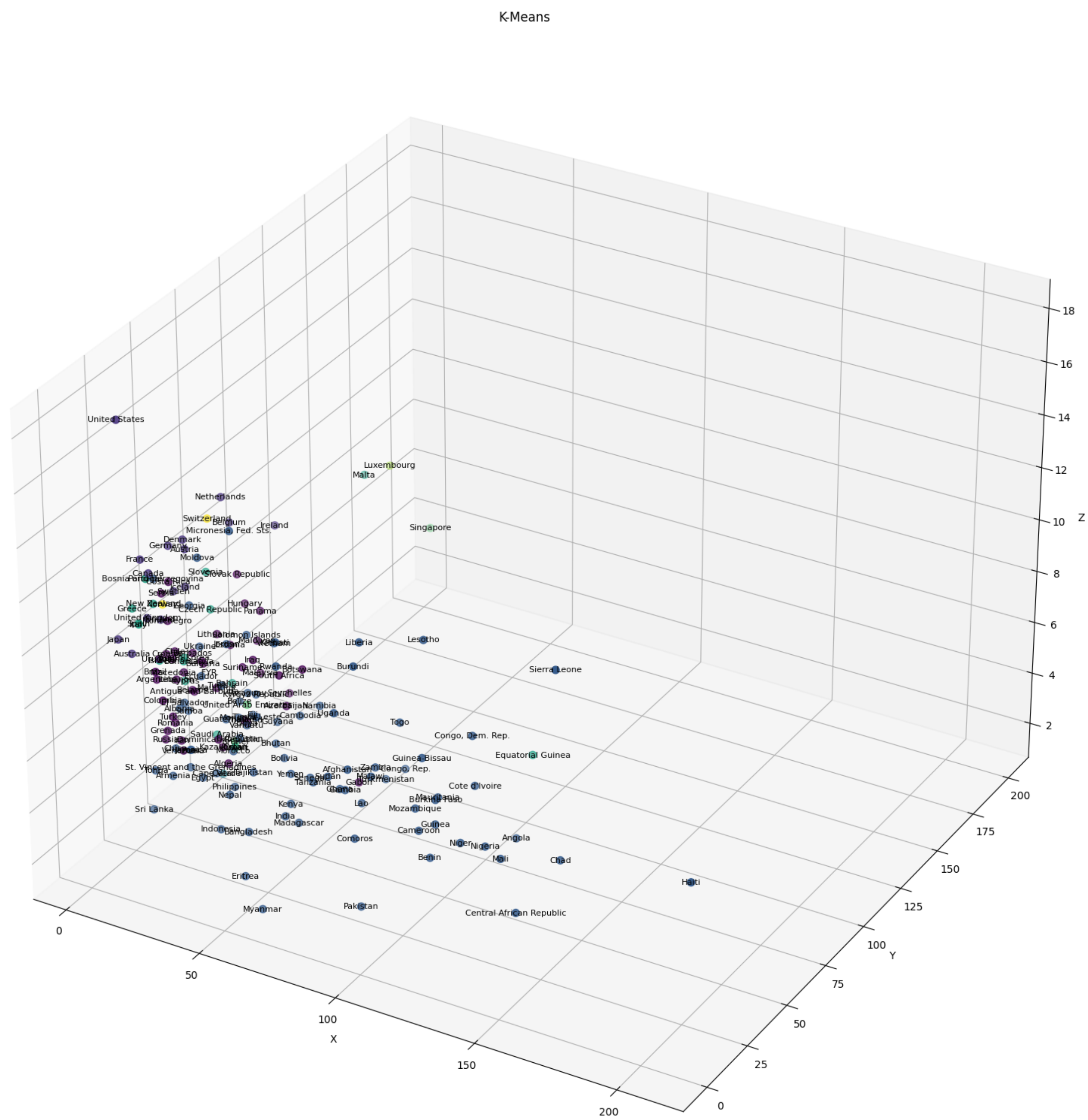


In [42]:

```
####### Helper funtion for plotting #######
def plot_data(points, labels, title):
    fig = plt.figure()
    if points.shape[1] > 2:
```

```
        ax = fig.add_subplot(projection='3d')
        ax.scatter(points[:,0], points[:,1], points[:,2], c=labels, cmap=
        ax.set_xlabel('X')
        ax.set_ylabel('Y')
        ax.set_zlabel('Z')
        ax.set_title(title)
    else:
        plt.scatter(points[:,0], points[:,1], c=labels, cmap='viridis')
        plt.xlabel('X')
        plt.ylabel('Y')
        plt.title(title)
    plt.show()

####### K-means #######
print('----- K-means -----')
kmeans = KMeans(n_clusters=8).fit(x)
clustering_labels = kmeans.labels_
centers = kmeans.cluster_centers_
print('Labels: ', clustering_labels)
#print('Centers: ', centers)
label_names = y
plot_data_with_labels(x, clustering_labels, label_names, 'K-Means')
```

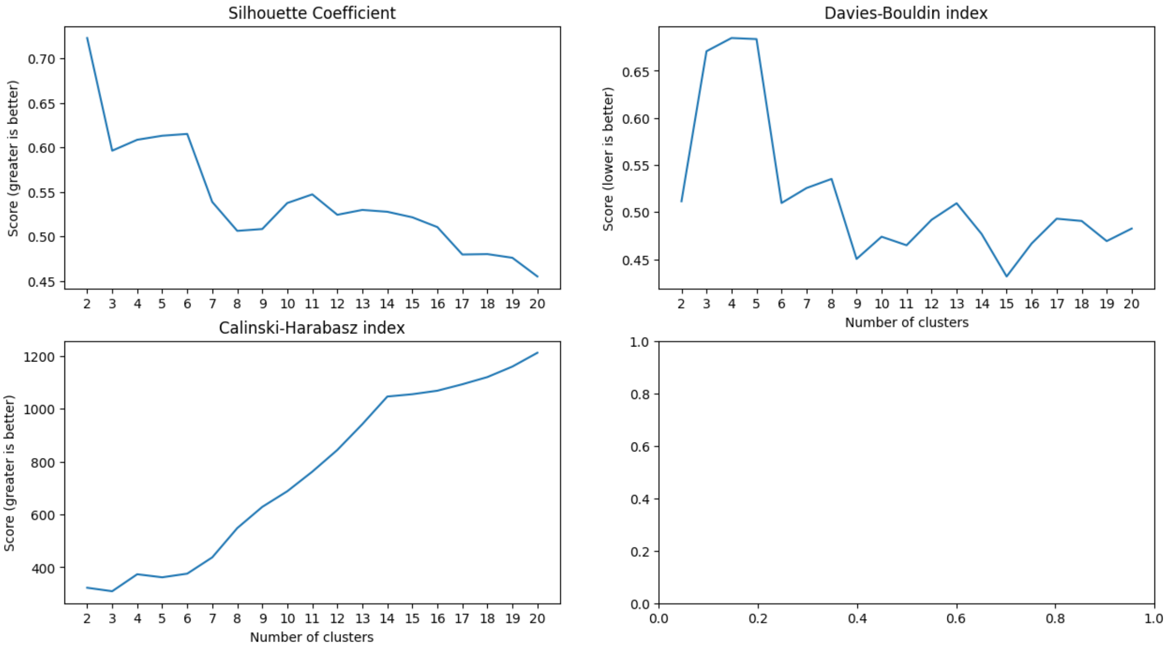
```
----- K-means -----
Labels:  [2 2 0 2 0 0 2 1 1 0 4 4 2 0 0 1 2 2 2 2 2 0 0 5 0 2 2 2 2 1 2 2
2 0 2 0 2
 2 2 0 2 0 4 4 1 0 2 2 2 4 2 0 2 1 1 0 2 2 1 2 4 0 2 2 2 2 2 0 1 2 2 0 0 1
 4 4 2 1 2 0 2 2 5 2 2 0 0 2 2 0 0 6 0 2 2 0 0 2 4 2 0 2 2 2 0 2 2 2 2 2 1
 4 2 2 7 4 2 0 2 2 2 0 4 3 0 0 2 2 4 2 0 0 2 5 0 4 2 0 4 4 2 2 2 0 1 7 2 2
 0 2 2 2 2 0 2 2 2 5 1 1 0 2 2 0 2 2 2]
/home/alanv/Documents/7/mate/venv/lib/python3.11/site-packages/sklearn/clu
ster/_kmeans.py:1412: FutureWarning: The default value of `n_init` will ch
ange from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to sup
press the warning
  super()._check_params_vs_input(X, default_n_init=10)
<Figure size 640x480 with 0 Axes>
```



2 Repita lo anterior, pero con otro método de agrupamiento que elijas.

In [43]:
```
# Optimal number of clusters
sum_of_squared_distances = []
sscore = []
chscore = []
dbscore = []
ks = np.arange(2, 21)
for k in ks:
    # Find clustering model
    agl = AgglomerativeClustering(n_clusters=k).fit(x)
    # Evaluate Silhouette score
    sscore.append(silhouette_score(x, agl.labels_))
    # Evaluate Calinski-Harabasz index
    chscore.append(calinski_harabasz_score(x,agl.labels_))
    # Evaluate Davies-Bouldin index
    dbscore.append(davies_bouldin_score(x, agl.labels_))
fig, axs = plt.subplots(2, 2, figsize=(15, 8))

axs[0][0].plot(ks, sscore)
```

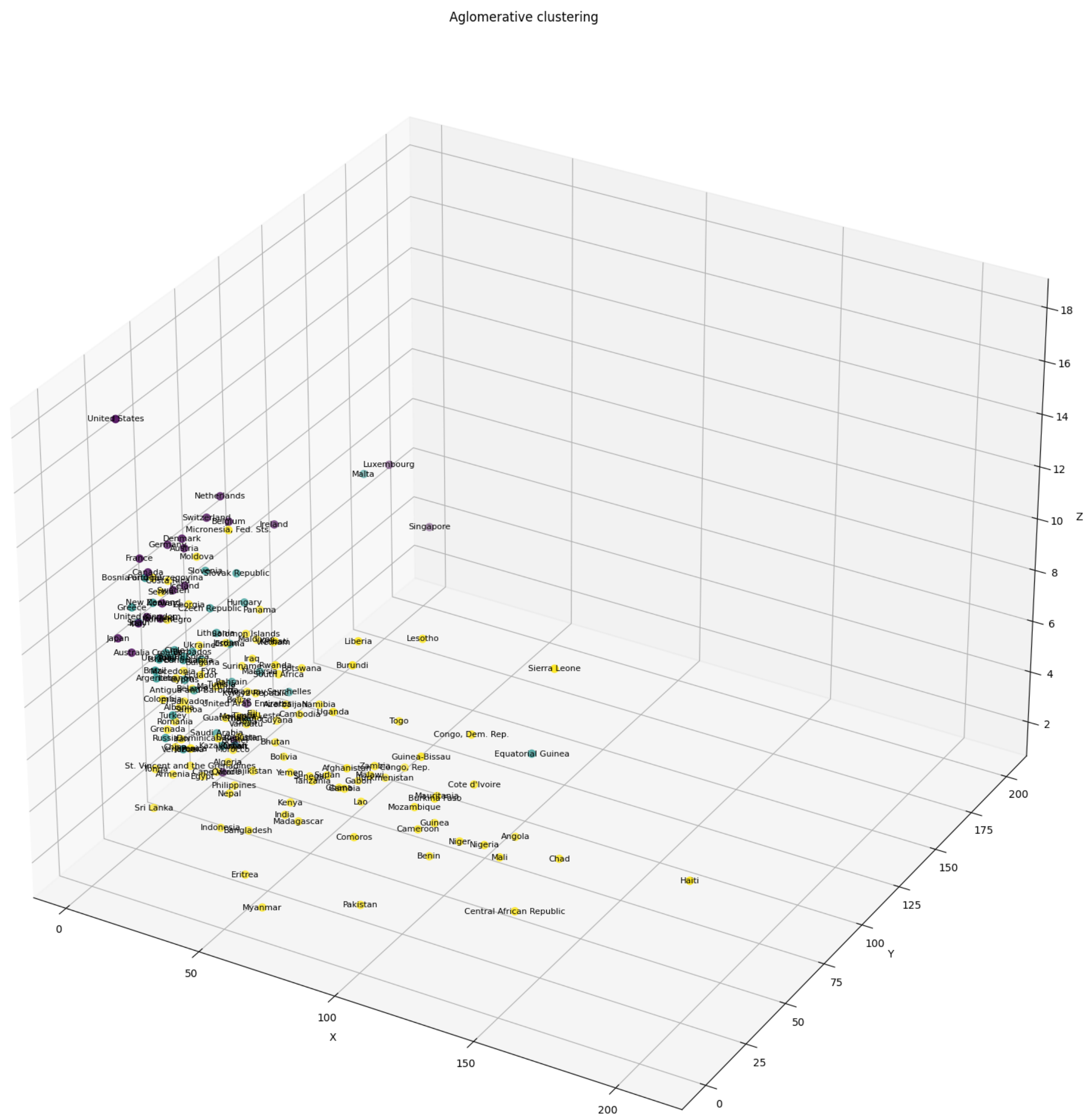
```
axs[0][0].set_ylabel('Score (greater is better)')
axs[0][0].set_title('Silhouette Coefficient')
axs[0][0].set_xticks(ks)
axs[1][0].plot(ks, chscore)
axs[1][0].set_xlabel('Number of clusters')
axs[1][0].set_ylabel('Score (greater is better)')
axs[1][0].set_title('Calinski-Harabasz index')
axs[1][0].set_xticks(ks)
axs[0][1].plot(ks, dbscore)
axs[0][1].set_xlabel('Number of clusters')
axs[0][1].set_ylabel('Score (lower is better)')
axs[0][1].set_title('Davies-Bouldin index')
axs[0][1].set_xticks(ks)
plt.show()
```



In [45]:

```
####### Aglomerative clustering #######
print('----- Aglomerative clustering -----')
agl = AgglomerativeClustering(n_clusters=3).fit(x)
clustering_labels = agl.labels_
print('Labels: ', clustering_labels)
#print('Centers: ', centers)
label_names = y
plot_data_with_labels(x, clustering_labels, label_names, 'Aglomerative cl
```

```
----- Aglomerative clustering -----
Labels:  [2 2 2 2 1 1 2 0 0 2 1 1 2 1 2 0 2 2 2 2 2 2 1 0 2 2 2 2 2 0 2 2
2 1 2 2 2
 2 2 2 2 1 1 1 0 2 2 2 2 1 2 1 2 0 0 2 2 2 0 2 1 2 2 2 2 2 2 1 0 2 2 2 2 0
 1 0 2 0 2 1 2 2 0 2 2 1 2 2 2 1 1 0 2 2 2 1 2 2 1 2 2 2 2 2 2 2 2 2 2 2 0
 1 2 2 0 1 2 2 2 2 2 1 1 0 2 1 2 2 1 2 2 1 2 0 1 1 2 2 1 1 2 2 2 2 0 0 2 2
 2 2 2 2 2 1 2 2 2 0 0 0 1 2 2 1 2 2 2]
<Figure size 640x480 with 0 Axes>
```



3. Investiga qué librerías hay en Python para la implementación de mapas autoorganizados, y selecciona alguna para el agrupamiento de los datos de este ejercicio. Algunos ejemplos de librerías son: Minosom, sklearn-som

In [52]:
```
# data normalization
data = x
data = (data - np.mean(data, axis=0)) / np.std(data, axis=0)

# Initialization and training
som_shape = (1, 3)
som = MiniSom(som_shape[0], som_shape[1], data.shape[1], sigma=.5, learni
              neighborhood_function='gaussian', random_seed=10)

som.train_batch(data, 500, verbose=True)


# each neuron represents a cluster
winner_coordinates = np.array([som.winner(x) for x in data]).T
# with np.ravel_multi_index we convert the bidimensional
# coordinates to a monodimensional index
cluster_index = np.ravel_multi_index(winner_coordinates, som_shape)
# plotting the clusters using the first 2 dimentions of the data
```

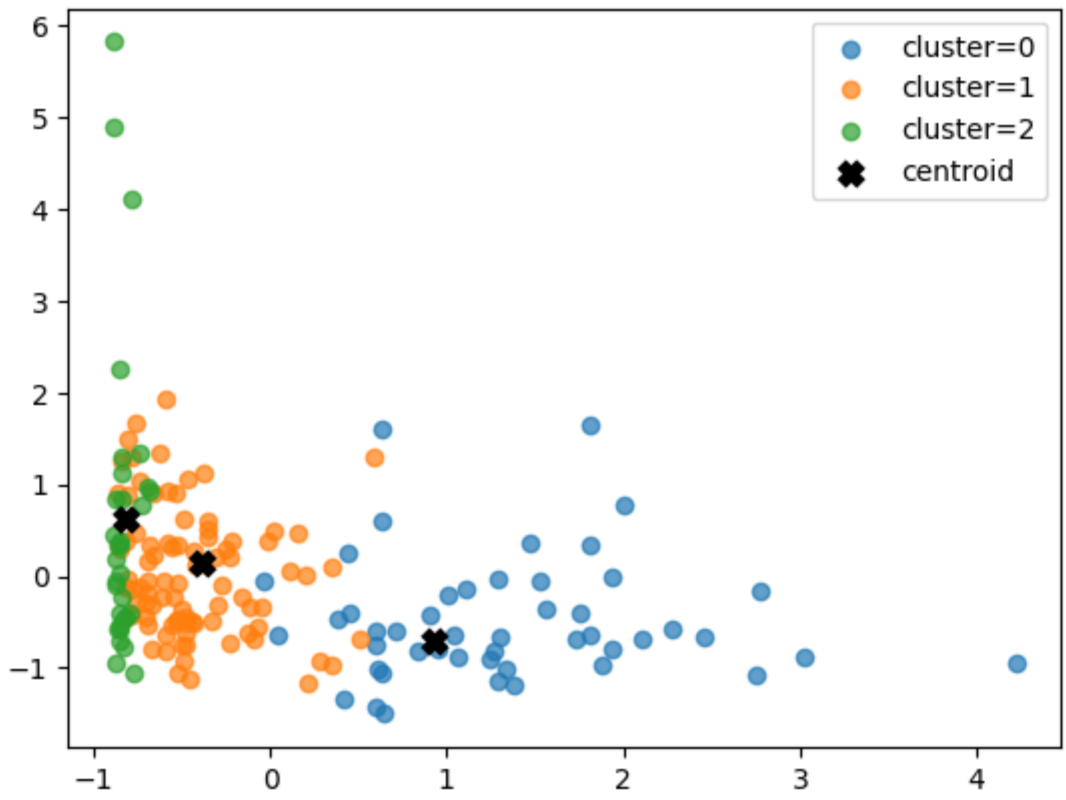
```
for c in np.unique(cluster_index):
    plt.scatter(data[cluster_index == c, 0],
                data[cluster_index == c, 1], label='cluster='+str(c), alp

# plotting centroids
for centroid in som.get_weights():
    plt.scatter(centroid[:, 0], centroid[:, 1], marker='x',
                s=10, linewidths=10, color='k', label='centroid')
plt.legend()

df['labels'] = winner_coordinates[1]
df.sort_values(by=['labels'], ascending=True)
```

```
[ 500 / 500 ] 100% - 0:00:00 left
quantization error: 2.0575072770943352
```

Out[52]:

	country	child_mort	exports	health	imports	income	inflation	life_expe
0	Afghanistan	90.2	10.0	7.58	44.9	1610	9.440	56.
88	Liberia	89.3	19.1	11.80	92.6	700	5.470	60.
87	Lesotho	99.7	39.4	11.10	101.0	2380	4.150	46.
84	Lao	78.9	35.4	4.47	49.3	3980	9.200	63.
165	Yemen	56.3	30.0	5.18	34.4	4480	23.600	67.
...	...	...	...	...	...	...	...	.
135	Slovenia	3.2	64.3	9.41	62.9	28700	-0.987	79.
29	Canada	5.6	29.1	11.30	31.0	40700	2.870	81.
139	Spain	3.8	25.5	9.54	26.8	32500	0.160	81.
58	Germany	4.2	42.3	11.60	37.1	40400	0.758	80.
7	Australia	4.8	19.8	8.73	20.9	41400	1.160	82.

167 rows × 11 columns



4. De los resultados que se obtienen del agrupamiento, indica si los grupos formados siguen algun patrón que esperabas, o tiene información nueva que no hayas considerado anteriormente.

Si tienen un patrón esperado, sin importar el número de clusters, la división de países depende de su ingreso per cápita. Por ejemplo, cuando se eligieron 8 clusters, dejó solito a Luxemburgo, el país con el mayor ingreso per cápita del mundo, y luego está otro cluster donde están Noruega y Suiza, que curiosamente son los países con el segundo y tercer lugar con mayor ingreso per cápita. Por otra parte, cuando se dividen en 3 clusters, esperaría que se dividieran en países de primer, segundo y tercer mundo, que es lo que hace. Esto es fácil de predecir debido a las columnas del dataset, que están orientadas principalmente a lo económico, y justo una de las variables predictoras es el ingreso.