0.) Import and Clean data

In [78]:
```
import pandas as pd
from google.colab import drive
import matplotlib.pyplot as plt
import numpy as np
```

In [79]:
```
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
```

In [80]:
```
drive.mount('/content/gdrive/', force_remount = True)
```

```
Mounted at /content/gdrive/
```

In [81]:
```
df = pd.read_csv("/content/gdrive/MyDrive/Country-data.csv", sep = ",")
```

In [82]:
```
df.head()
```

Out[82]:

	country	child_mort	exports	health	imports	income	inflation	life_expec	total_fer	gdpp
0	Afghanistan	90.2	10.0	7.58	44.9	1610	9.44	56.2	5.82	553
1	Albania	16.6	28.0	6.55	48.6	9930	4.49	76.3	1.65	4090
2	Algeria	27.3	38.4	4.17	31.4	12900	16.10	76.5	2.89	4460
3	Angola	119.0	62.3	2.85	42.9	5900	22.40	60.1	6.16	3530
4	Antigua and Barbuda	10.3	45.5	6.03	58.9	19100	1.44	76.8	2.13	12200

In [83]:
```
df.columns
```

Out[83]:
```
Index(['country', 'child_mort', 'exports', 'health', 'imports', 'income',
       'inflation', 'life_expec', 'total_fer', 'gdpp'],
      dtype='object')
```

In [84]:
```
names = df[["country"]]
X = df.drop(["country"], axis = 1)
```

In [85]:
```
scaler = StandardScaler().fit(X)
X_scaled = scaler.transform(X)
X_scaled
```

Out[85]:
```
array([[ 1.29153238, -1.13827979,  0.27908825, ..., -1.61909203,
         1.90288227, -0.67917961],
       [-0.5389489 , -0.47965843, -0.09701618, ...,  0.64786643,
        -0.85997281, -0.48562324],
       [-0.27283273, -0.09912164, -0.96607302, ...,  0.67042323,
        -0.0384044 , -0.46537561],
       ...,
       [-0.37231541,  1.13030491,  0.0088773 , ...,  0.28695762,
        -0.66120626, -0.63775406],
       [ 0.44841668, -0.40647827, -0.59727159, ..., -0.34463279,
         1.14094382, -0.63775406],
       [ 1.11495062, -0.15034774, -0.33801514, ..., -2.09278484,
         1.6246091 , -0.62954556]])
```

In [ ]:

In [ ]:

1.) Fit a kmeans Model with any Number of Clusters

In [86]:
```
kmeans = KMeans(n_clusters= 3
                , random_state=42).fit(X_scaled)
```

```
/usr/local/lib/python3.9/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value
of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the
warning
  warnings.warn(
```

In [ ]:
```
```

2.) Pick two features to visualize across

In [87]:
```
X.columns
```

Out[87]:
```
Index(['child_mort', 'exports', 'health', 'imports', 'income', 'inflation',
       'life_expec', 'total_fer', 'gdpp'],
      dtype='object')
```

In [88]:
```
X[['child_mort', 'income']]
```

Out[88]:

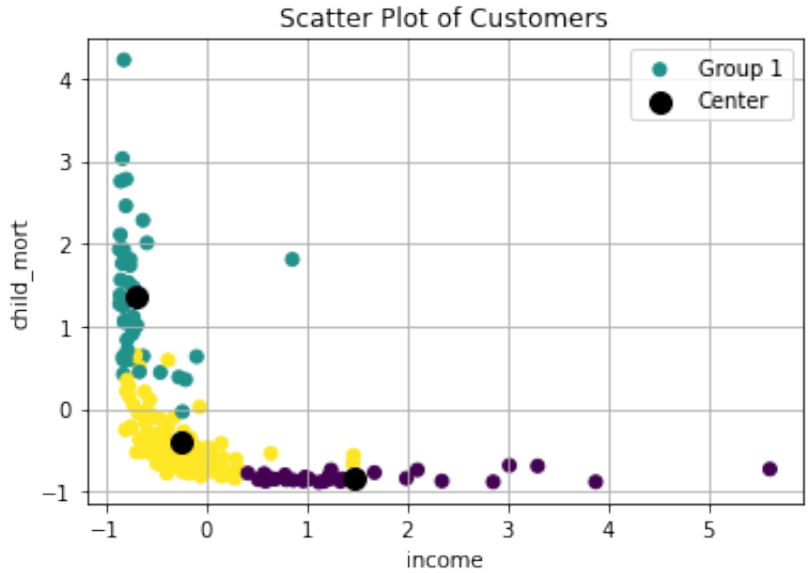
	child_mort	income
0	90.2	1610
1	16.6	9930
2	27.3	12900
3	119.0	5900
4	10.3	19100
...	...	...
162	29.2	2950
163	17.1	16500
164	23.3	4490
165	56.3	4480
166	83.1	3280

167 rows × 2 columns

In [89]:
```
# CHANGE THESE BASED ON WHICH IS INTERESTING TO YOU
x1_index = 4
x2_index = 0


plt.scatter(X_scaled[:, x1_index], X_scaled[:, x2_index], c=kmeans.labels_, cmap='viridis')
plt.scatter(kmeans.cluster_centers_[:, x1_index], kmeans.cluster_centers_[:, x2_index], marker='o', colo

plt.xlabel(X.columns[x1_index])
plt.ylabel(X.columns[x2_index])
plt.title('Scatter Plot of Customers')
plt.legend(["Group 1", "Center", "Group 2"])
plt.grid()
plt.show()
```



From the above graph, we can get the intuition that with rising income, the child mortality is going to go down. As we see, that at low level of income, the child mortality is pretty high. As the income goes up, the child mortality seems to be falling.

We will check this intuition in the upcoming codes.

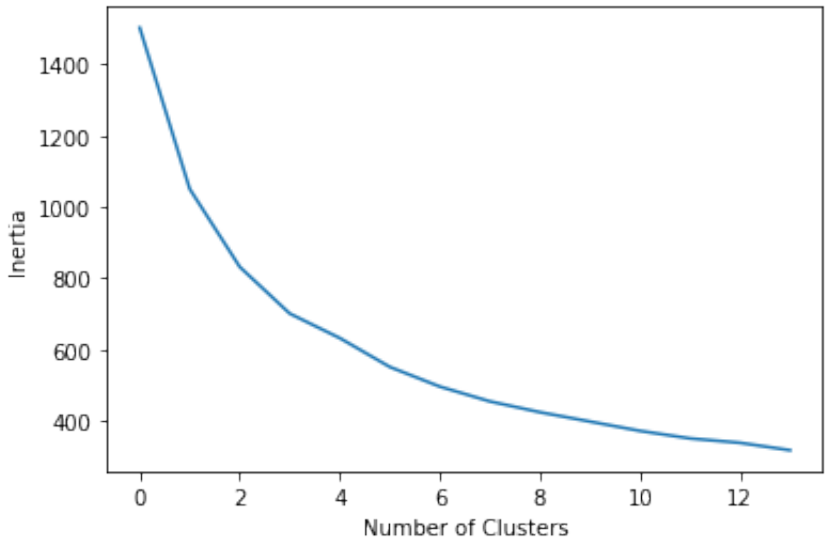
3.) Check a range of k-clusters and visualize to find the elbow. Test 30 different random starting places for the centroid means

In [90]:
```
from sklearn.cluster import KMeans
```

In [91]:
```
WCSS = []
Ks = range(1,15)
for k in Ks:
  kmeans = KMeans(n_clusters = k , n_init = 30)
  kmeans.fit(X_scaled)
  WCSS.append(kmeans.inertia_)
```

4.) Use the above work and economic critical thinking to choose a number of clusters. Explain why you chose the number of clusters and fit a model accordingly.

In [92]:
```
plt.plot(WCSS)
plt.xlabel("Number of Clusters")
plt.ylabel("Inertia")
plt.show()
```



In [93]:
```
k = 4 #maybe low, middle-low, middle and high

kmeans = KMeans(n_clusters = k).fit(X_scaled)
```

```
/usr/local/lib/python3.9/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value
of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the
warning
  warnings.warn(
```

5.) Create a list of the countries that are in each cluster. Write interesting things you notice. Hint : Use .predict(method)

In [94]:
```
preds = pd.DataFrame(kmeans.predict(X_scaled))

output = pd.concat([preds, names], axis = 1)

output[output[0] == 0]["country"]
print("Cluster 1: ", output) #can change clusters here
```

```
Cluster 1:       0              country
0    0          Afghanistan
1    2              Albania
2    2              Algeria
3    0               Angola
4    2  Antigua and Barbuda
..  ..                  ...
162  2              Vanuatu
163  2            Venezuela
164  2              Vietnam
165  0                Yemen
166  0               Zambia

[167 rows x 2 columns]
```

In [95]:

```
# Make predictions on the scaled data
preds = kmeans.predict(X_scaled)

# Combine the cluster labels with the original data
output = pd.concat([pd.DataFrame(preds, columns=['cluster']), names], axis=1)

# Loop over the four clusters and print the country names for each
for i in range(4):
    cluster_i = output[output['cluster'] == i]['country']
    print(f"Cluster {i + 1}: {list(cluster_i)}")
```

```
Cluster 1: ['Afghanistan', 'Angola', 'Benin', 'Botswana', 'Burkina Faso', 'Burundi', 'Cameroon', 'Centr
al African Republic', 'Chad', 'Comoros', 'Congo, Dem. Rep.', 'Congo, Rep.', "Cote d'Ivoire", 'Equatoria
l Guinea', 'Eritrea', 'Gabon', 'Gambia', 'Ghana', 'Guinea', 'Guinea-Bissau', 'Haiti', 'Iraq', 'Kenya',
'Kiribati', 'Lao', 'Lesotho', 'Liberia', 'Madagascar', 'Malawi', 'Mali', 'Mauritania', 'Mozambique', 'N
amibia', 'Niger', 'Nigeria', 'Pakistan', 'Rwanda', 'Senegal', 'Sierra Leone', 'South Africa', 'Sudan',
'Tanzania', 'Timor-Leste', 'Togo', 'Uganda', 'Yemen', 'Zambia']
Cluster 2: ['Luxembourg', 'Malta', 'Singapore']
Cluster 3: ['Albania', 'Algeria', 'Antigua and Barbuda', 'Argentina', 'Armenia', 'Azerbaijan', 'Bahamas
', 'Bahrain', 'Bangladesh', 'Barbados', 'Belarus', 'Belize', 'Bhutan', 'Bolivia', 'Bosnia and Herzegovi
na', 'Brazil', 'Bulgaria', 'Cambodia', 'Cape Verde', 'Chile', 'China', 'Colombia', 'Costa Rica', 'Croat
ia', 'Czech Republic', 'Dominican Republic', 'Ecuador', 'Egypt', 'El Salvador', 'Estonia', 'Fiji', 'Geo
rgia', 'Grenada', 'Guatemala', 'Guyana', 'Hungary', 'India', 'Indonesia', 'Iran', 'Jamaica', 'Jordan',
'Kazakhstan', 'Kyrgyz Republic', 'Latvia', 'Lebanon', 'Libya', 'Lithuania', 'Macedonia, FYR', 'Malaysia
', 'Maldives', 'Mauritius', 'Micronesia, Fed. Sts.', 'Moldova', 'Mongolia', 'Montenegro', 'Morocco', 'M
yanmar', 'Nepal', 'Oman', 'Panama', 'Paraguay', 'Peru', 'Philippines', 'Poland', 'Romania', 'Russia', '
Samoa', 'Saudi Arabia', 'Serbia', 'Seychelles', 'Slovak Republic', 'Solomon Islands', 'Sri Lanka', 'St.
Vincent and the Grenadines', 'Suriname', 'Tajikistan', 'Thailand', 'Tonga', 'Tunisia', 'Turkey', 'Turkm
enistan', 'Ukraine', 'Uruguay', 'Uzbekistan', 'Vanuatu', 'Venezuela', 'Vietnam']
Cluster 4: ['Australia', 'Austria', 'Belgium', 'Brunei', 'Canada', 'Cyprus', 'Denmark', 'Finland', 'Fra
nce', 'Germany', 'Greece', 'Iceland', 'Ireland', 'Israel', 'Italy', 'Japan', 'Kuwait', 'Netherlands', '
New Zealand', 'Norway', 'Portugal', 'Qatar', 'Slovenia', 'South Korea', 'Spain', 'Sweden', 'Switzerland
', 'United Arab Emirates', 'United Kingdom', 'United States']
```

From the cluster groups above, we see that African countries like Benin, Burundi and others are categorised in the low income country. While developed countries such as United States, Singapore are cateorised in the middle-income to high income category.

This also shows that developing countries have been categorised in the low-income category while the developed countries are categorised in the high-income category.

#6.) Create a table of Descriptive Statistics. Rows being the Cluster number and columns being all the features. Values being the mean of the centroid. Use the nonscaled X values for interprotation

In [96]:
```
Q6DF = pd.concat([pd.DataFrame(preds), X], axis=1)
Q6DF
```

Out[96]:

	0	child_mort	exports	health	imports	income	inflation	life_expec	total_fer	gdpp
0	0	90.2	10.0	7.58	44.9	1610	9.44	56.2	5.82	553
1	2	16.6	28.0	6.55	48.6	9930	4.49	76.3	1.65	4090
2	2	27.3	38.4	4.17	31.4	12900	16.10	76.5	2.89	4460
3	0	119.0	62.3	2.85	42.9	5900	22.40	60.1	6.16	3530
4	2	10.3	45.5	6.03	58.9	19100	1.44	76.8	2.13	12200
...	...	...	...	...	...	...	...	...	...	...
162	2	29.2	46.6	5.25	52.7	2950	2.62	63.0	3.50	2970
163	2	17.1	28.5	4.91	17.6	16500	45.90	75.4	2.47	13500
164	2	23.3	72.0	6.84	80.2	4490	12.10	73.1	1.95	1310
165	0	56.3	30.0	5.18	34.4	4480	23.60	67.5	4.67	1310
166	0	83.1	37.0	5.89	30.9	3280	14.00	52.0	5.40	1460

167 rows × 10 columns

In [97]:
```
Q6DF.groupby(0).mean()
```

Out[97]:

0	child_mort	exports	health	imports	income	inflation	life_expec	total_fer	gdpp
0	92.961702	29.151277	6.388511	42.323404	3942.404255	12.019681	59.187234	5.008085	1922.382979
1	4.133333	176.000000	6.793333	156.666667	64033.333333	2.468000	81.433333	1.380000	57566.666667
2	21.389655	41.290678	6.235862	48.038689	12968.620690	7.413460	72.935632	2.286552	6919.103448
3	4.953333	45.826667	9.168667	39.736667	45250.000000	2.742200	80.376667	1.795333	43333.333333

Q7.) Write an observation about the descriptive statistics.

From the table above, my dataset is divided into 4 groups which can be categorised as:

1. Low income
2. Lower-middle income
3. Middle income
4. High income

Analysing the 'income' column in the table, we see that the first row are the countries which are categorised as 'low income' with a mean income of 3539.844444 and the third row having the higher mean income of 64033.333333.

Comparing it with child mortality column, we see that as income increases, the child mortality falls. Our intuition from before is, therefore, justified. We can imagine this to happen because as the income increases, the families can spend more towards the well being of the child. Further, a higher income signifies that poeple can spend more on factors like hygeine, medical care and education.