Sec_1_Homework_9

March 6, 2024

1 0.) Import and Clean data

[81]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
```

[49]:
```
#drive.mount('/content/gdrive/', force_remount = True)
df = pd.read_csv("Country-data.csv", sep = ",")
df.set_index('country', inplace = True)
df
```

[49]:
```
                     child_mort  exports  health  imports  income  inflation  \
country
Afghanistan                90.2     10.0    7.58     44.9    1610       9.44
Albania                    16.6     28.0    6.55     48.6    9930       4.49
Algeria                    27.3     38.4    4.17     31.4   12900      16.10
Angola                    119.0     62.3    2.85     42.9    5900      22.40
Antigua and Barbuda        10.3     45.5    6.03     58.9   19100       1.44
…                          …       …       …     …       …       …
Vanuatu                    29.2     46.6    5.25     52.7    2950       2.62
Venezuela                  17.1     28.5    4.91     17.6   16500      45.90
Vietnam                    23.3     72.0    6.84     80.2    4490      12.10
Yemen                      56.3     30.0    5.18     34.4    4480      23.60
Zambia                     83.1     37.0    5.89     30.9    3280      14.00

                     life_expec  total_fer   gdpp
country
Afghanistan                56.2       5.82    553
Albania                    76.3       1.65   4090
Algeria                    76.5       2.89   4460
Angola                     60.1       6.16   3530
Antigua and Barbuda        76.8       2.13  12200
…                          …         …      …
Vanuatu                    63.0       3.50   2970
Venezuela                  75.4       2.47  13500
```

```
Vietnam                        73.1       1.95    1310
Yemen                          67.5       4.67    1310
Zambia                         52.0       5.40    1460

[167 rows x 9 columns]
```

[91]:
```python
X = df
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

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

[93]:
```python
kmeans = KMeans(n_clusters=5, n_init=1).fit(X_scaled)
kmeans
```

[93]:
```
KMeans(n_clusters=5, n_init=1)
```

3 2.) Pick two features to visualize across

[59]:
```python
X.columns
```

[59]:
```
Index(['child_mort', 'exports', 'health', 'imports', 'income', 'inflation',
       'life_expec', 'total_fer', 'gdpp'],
      dtype='object')
```

[90]:
```python
import matplotlib.pyplot as plt

x1_index = 0
x2_index = 4

# Apply KMeans clustering
kmeans = KMeans(n_clusters=3)  # Example: assuming 3 clusters
kmeans.fit(X_scaled)

# Obtain cluster labels
#cluster_labels = kmeans.labels_

# Plot the scatter plot of data points with KMeans clustering labels
scatter = plt.scatter(X_scaled[:, x1_index], X_scaled[:, x2_index],cmap =␣
 ↪'viridis', c=kmeans.labels_, label='Clusters')

# Plot the cluster centers
centers = plt.scatter(kmeans.cluster_centers_[:, x1_index], kmeans.
 ↪cluster_centers_[:, x2_index], marker='o', color='black', s=100,␣
 ↪label='Centers')
```

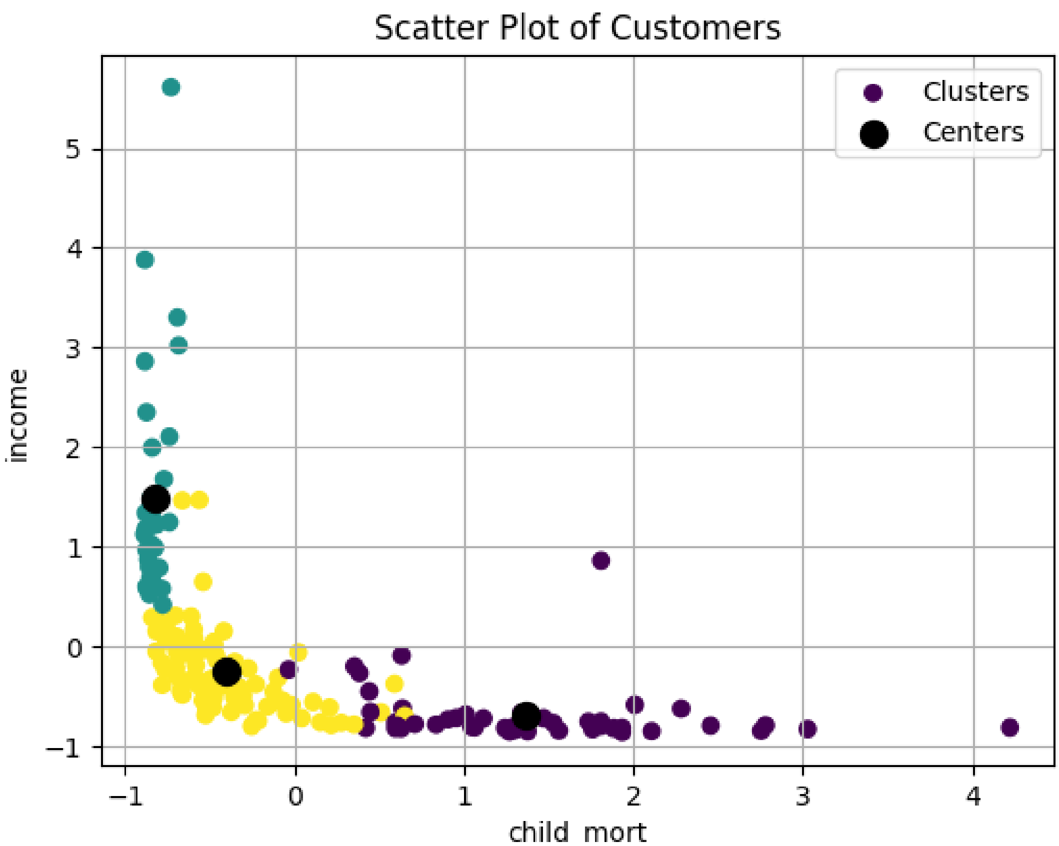
```
# Set labels and title
plt.xlabel(X.columns[x1_index])
plt.ylabel(X.columns[x2_index])
plt.title('Scatter Plot of Customers')

# Generate legend
plt.legend()

# Display grid
plt.grid()

# Show the plot
plt.show()
```

```
/Users/trujillo/anaconda3/lib/python3.11/site-
packages/sklearn/cluster/_kmeans.py:1416: 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
  super()._check_params_vs_input(X, default_n_init=10)
```



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

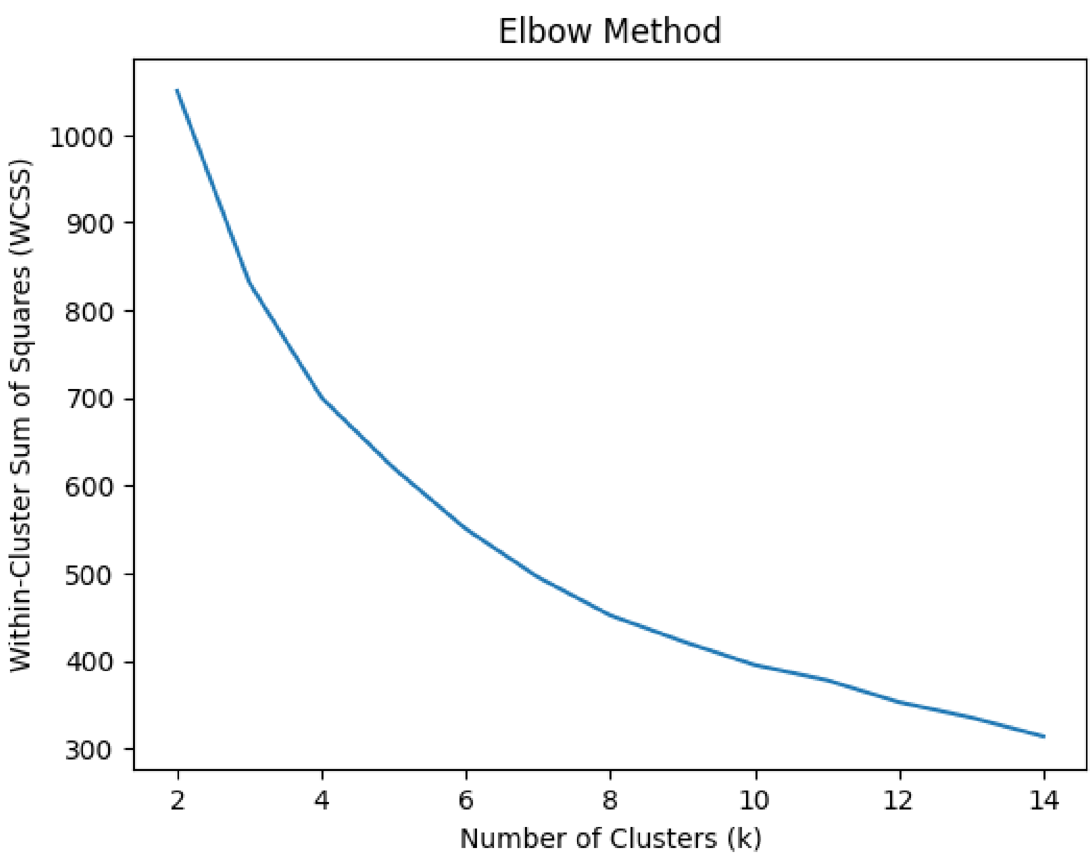
[87]:
```
WCSSs = []
Ks = range(2, 15)
for k in Ks:
    kmeans = KMeans(n_clusters=k, n_init=30).fit(X_scaled)
    WCSSs.append(kmeans.inertia_)
```

[ ]:
```

```

[88]:
```
plt.plot(Ks,WCSSs)
plt.xlabel('Number of Clusters (k)')
plt.ylabel('Within-Cluster Sum of Squares (WCSS)')
plt.title('Elbow Method')
```

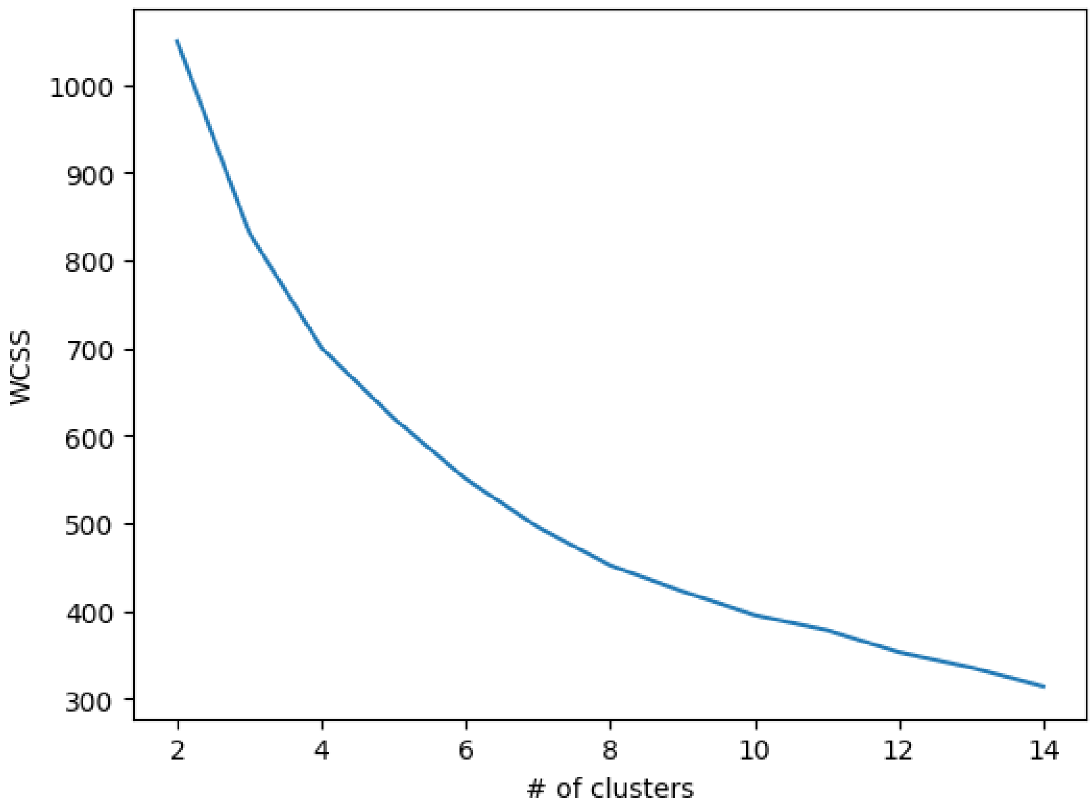
[88]: Text(0.5, 1.0, 'Elbow Method')



5 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.

[89]:
```
plt.plot(Ks,WCSSs)
plt.xlabel('# of clusters')
plt.ylabel('WCSS')
```

[89]:
```
Text(0, 0.5, 'WCSS')
```



Even if the optimal number of clusters cannot be seen in this plot (there is no kink), economic rationale would tell us to ask for two clusters. That way we only divide the sample into advanced and developed economies. There might be many more options, but to keep it simple we will select only 2.

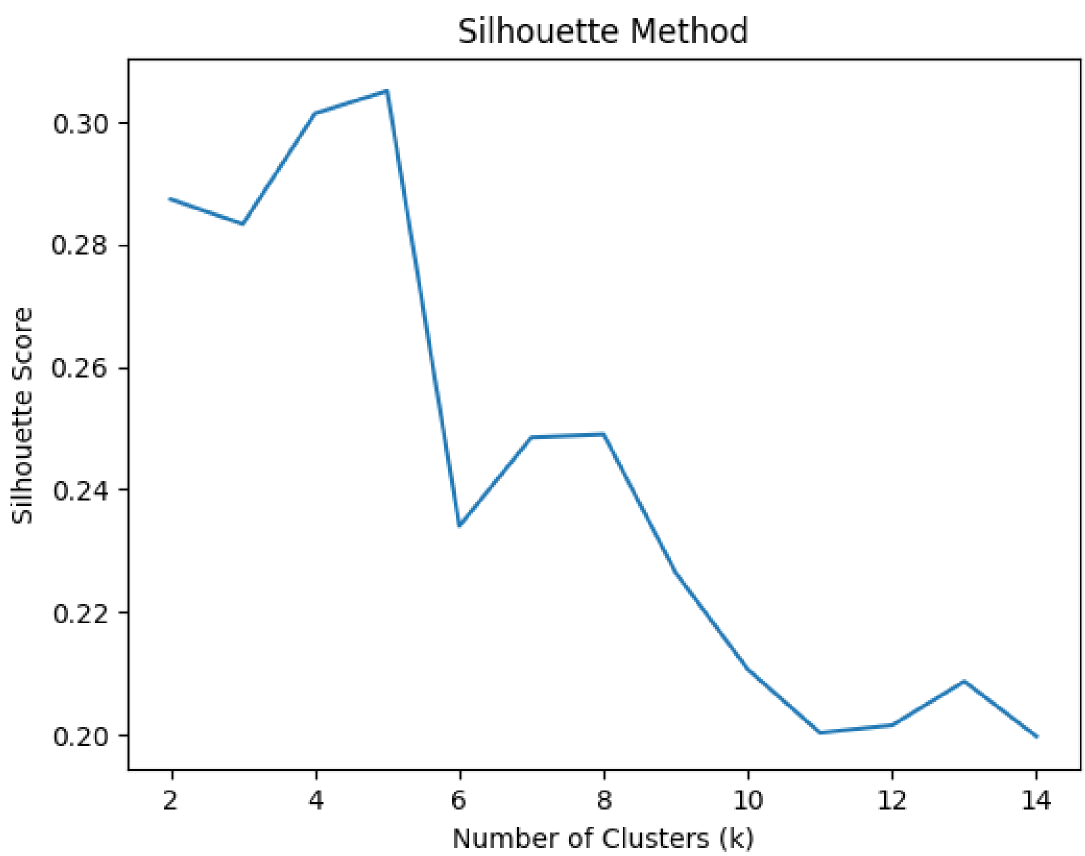
6 6.) Do the same for a silhoutte plot

[94]:
```
from sklearn.metrics import silhouette_score
```

[95]:
```python
SSs = []
Ks = range(2, 15)
for k in Ks:
    kmeans = KMeans(n_clusters=k, n_init=30).fit(X_scaled)
    sil = silhouette_score(X_scaled, kmeans.labels_)
    SSs.append(sil)
```

[41]:
```python
plt.plot(Ks,SSs)
plt.xlabel('Number of Clusters (k)')
plt.ylabel('Silhouette Score')
plt.title('Silhouette Method')
```

[41]: Text(0.5, 1.0, 'Silhouette Method')



7 7.) Create a list of the countries that are in each cluster. Write interesting things you notice.

[96]:
```
kmeans = KMeans(n_clusters=2, n_init=30).fit(X_scaled)
```

[98]:
```
preds = kmeans.labels_

df['preds'] = preds
```

[99]:
```
print('Cluster 1: ', df.index[preds==0] )
```

```
Cluster 1:  Index(['Albania', 'Algeria', 'Antigua and Barbuda', 'Argentina',
'Armenia',
       'Australia', 'Austria', 'Azerbaijan', 'Bahamas', 'Bahrain', 'Barbados',
       'Belarus', 'Belgium', 'Belize', 'Bhutan', 'Bosnia and Herzegovina',
       'Brazil', 'Brunei', 'Bulgaria', 'Canada', 'Cape Verde', 'Chile',
       'China', 'Colombia', 'Costa Rica', 'Croatia', 'Cyprus',
       'Czech Republic', 'Denmark', 'Dominican Republic', 'Ecuador',
       'El Salvador', 'Estonia', 'Fiji', 'Finland', 'France', 'Georgia',
       'Germany', 'Greece', 'Grenada', 'Hungary', 'Iceland', 'Iran', 'Ireland',
       'Israel', 'Italy', 'Jamaica', 'Japan', 'Jordan', 'Kazakhstan', 'Kuwait',
       'Latvia', 'Lebanon', 'Libya', 'Lithuania', 'Luxembourg',
       'Macedonia, FYR', 'Malaysia', 'Maldives', 'Malta', 'Mauritius',
       'Moldova', 'Montenegro', 'Morocco', 'Netherlands', 'New Zealand',
       'Norway', 'Oman', 'Panama', 'Paraguay', 'Peru', 'Poland', 'Portugal',
       'Qatar', 'Romania', 'Russia', 'Saudi Arabia', 'Serbia', 'Seychelles',
       'Singapore', 'Slovak Republic', 'Slovenia', 'South Korea', 'Spain',
       'Sri Lanka', 'St. Vincent and the Grenadines', 'Suriname', 'Sweden',
       'Switzerland', 'Thailand', 'Tunisia', 'Turkey', 'Ukraine',
       'United Arab Emirates', 'United Kingdom', 'United States', 'Uruguay',
       'Venezuela', 'Vietnam'],
      dtype='object', name='country')
```

[100]:
```
print('Cluster 2: ', df.index[preds==1] )
```

```
Cluster 2:  Index(['Afghanistan', 'Angola', 'Bangladesh', 'Benin', 'Bolivia',
'Botswana',
       'Burkina Faso', 'Burundi', 'Cambodia', 'Cameroon',
       'Central African Republic', 'Chad', 'Comoros', 'Congo, Dem. Rep.',
       'Congo, Rep.', 'Cote d'Ivoire', 'Egypt', 'Equatorial Guinea', 'Eritrea',
       'Gabon', 'Gambia', 'Ghana', 'Guatemala', 'Guinea', 'Guinea-Bissau',
       'Guyana', 'Haiti', 'India', 'Indonesia', 'Iraq', 'Kenya', 'Kiribati',
       'Kyrgyz Republic', 'Lao', 'Lesotho', 'Liberia', 'Madagascar', 'Malawi',
       'Mali', 'Mauritania', 'Micronesia, Fed. Sts.', 'Mongolia', 'Mozambique',
       'Myanmar', 'Namibia', 'Nepal', 'Niger', 'Nigeria', 'Pakistan',
       'Philippines', 'Rwanda', 'Samoa', 'Senegal', 'Sierra Leone',
       'Solomon Islands', 'South Africa', 'Sudan', 'Tajikistan', 'Tanzania',
       'Timor-Leste', 'Togo', 'Tonga', 'Turkmenistan', 'Uganda', 'Uzbekistan',
```

```
 'Vanuatu', 'Yemen', 'Zambia'],
dtype='object', name='country')
```

[ ]:
```
#### Write an observation
```

It seems that the clustering was roughly fine. For example, in this case it is clear that cluster 1 refers to the advanced economies because we have countries as United Kingdom, United States, Netherlands, Germany, among others. However, maybe the threshold was a little bit fuzzy because we also get El Salvador, Venezuela, and Vietnam in this cluster. For cluster 2 (developing countries) all selections seem right by inspection.

#8.) 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 interpretation

[77]:
```
df.groupby('preds').mean()
```

[77]:

preds	child_mort	exports	health	imports	income	inflation
0	12.161616	48.603030	7.314040	49.121212	26017.171717	5.503545
1	76.280882	30.198515	6.090147	43.642146	4227.397059	11.098750

preds	life_expec	total_fer	gdpp
0	76.493939	1.941111	20507.979798
1	61.910294	4.413824	1981.235294

[78]:
```
df.groupby('preds').std()
```

[78]:

preds	child_mort	exports	health	imports	income	inflation
0	8.523122	30.116032	2.716652	26.928785	20441.749847	6.957187
1	38.076068	18.201742	2.645319	19.323451	4890.581414	13.682630

preds	life_expec	total_fer	gdpp
0	3.735757	0.486744	20578.727127
1	6.897418	1.285590	2528.509189

8 9.) Write an observation about the descriptive statistics.

As thought before, the first cluster refers to the advanced economies since we can see lower child mortality, higher exports, much higher income and life expectancy. Also, its standard error is much lower across metrics (except exports and imports) so we can argue that the developing economies are still very far away from each other, while the advanced might be reaching some stationary state.

[ ]: