Sec_1_Homework_9

March 7, 2024

1 0.) Import and Clean data

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

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

[ ]:

	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
..	…	…	…	…	…	…	…	…	…	…
162	Vanuatu	29.2	46.6	5.25	52.7	2950	2.62	63.0	3.50	2970
163	Venezuela	17.1	28.5	4.91	17.6	16500	45.90	75.4	2.47	13500
164	Vietnam	23.3	72.0	6.84	80.2	4490	12.10	73.1	1.95	1310
165	Yemen	56.3	30.0	5.18	34.4	4480	23.60	67.5	4.67	1310
166	Zambia	83.1	37.0	5.89	30.9	3280	14.00	52.0	5.40	1460

```
[167 rows x 10 columns]
```

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

[ ]:
```
names = df[['country']].copy()
X = df.drop('country', axis = 1)
```

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

[ ]:
```
kmeans = KMeans(n_clusters=5).fit(X_scaled)
```

```
c:\Users\nikpa\anacondafinal\Lib\site-packages\sklearn\cluster\_kmeans.py:1412:
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)
c:\Users\nikpa\anacondafinal\Lib\site-packages\sklearn\cluster\_kmeans.py:1436:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=1.
  warnings.warn(
```

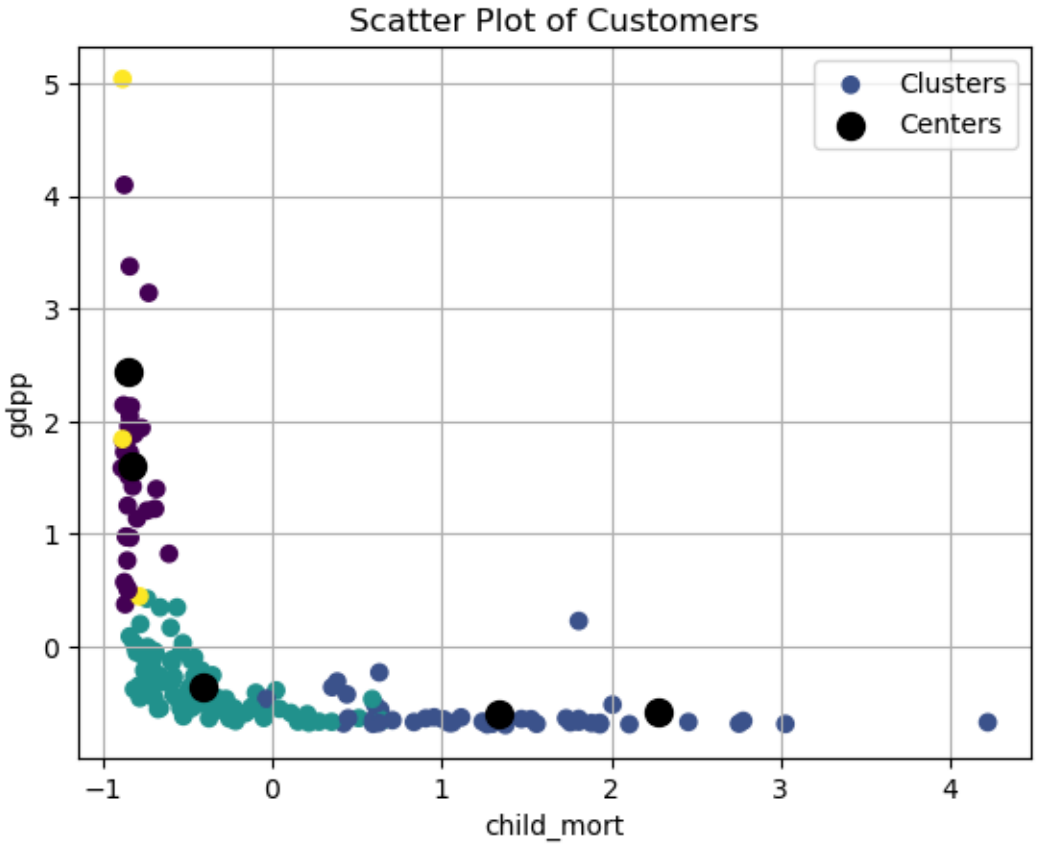
3 2.) Pick two features to visualize across

[ ]:
```
X.columns
```

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

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

x1_index = 0
x2_index = -1


scatter = plt.scatter(X_scaled[:, x1_index], X_scaled[:, x2_index], c=kmeans.
 ↪labels_, cmap='viridis', label='Clusters')


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

plt.xlabel(X.columns[x1_index])
```

```
plt.ylabel(X.columns[x2_index])
plt.title('Scatter Plot of Customers')

# Generate legend
plt.legend()

plt.grid()
plt.show()
```



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

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

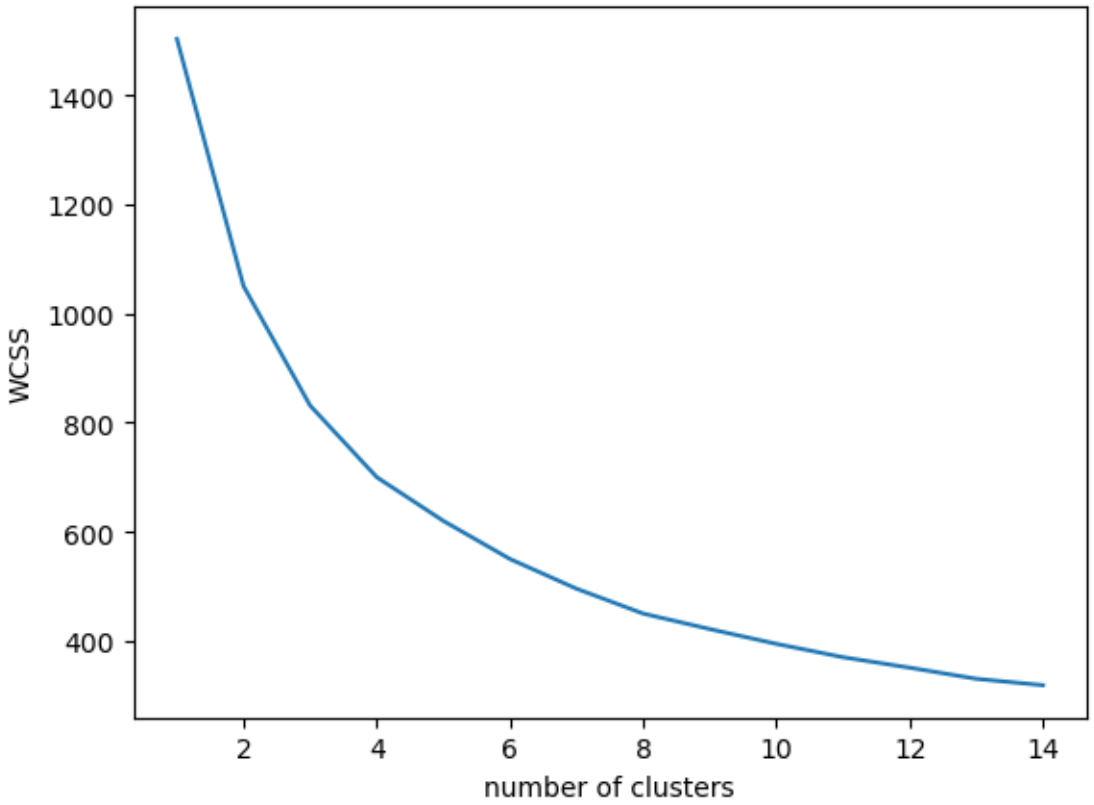
```
c:\Users\nikpa\anacondafinal\Lib\site-packages\sklearn\cluster\_kmeans.py:1436:
```

```
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=1.
  warnings.warn(
c:\Users\nikpa\anacondafinal\Lib\site-packages\sklearn\cluster\_kmeans.py:1436:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=1.
  warnings.warn(
c:\Users\nikpa\anacondafinal\Lib\site-packages\sklearn\cluster\_kmeans.py:1436:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=1.
  warnings.warn(
c:\Users\nikpa\anacondafinal\Lib\site-packages\sklearn\cluster\_kmeans.py:1436:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=1.
  warnings.warn(
c:\Users\nikpa\anacondafinal\Lib\site-packages\sklearn\cluster\_kmeans.py:1436:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=1.
  warnings.warn(
c:\Users\nikpa\anacondafinal\Lib\site-packages\sklearn\cluster\_kmeans.py:1436:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=1.
  warnings.warn(
c:\Users\nikpa\anacondafinal\Lib\site-packages\sklearn\cluster\_kmeans.py:1436:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=1.
  warnings.warn(
c:\Users\nikpa\anacondafinal\Lib\site-packages\sklearn\cluster\_kmeans.py:1436:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=1.
  warnings.warn(
c:\Users\nikpa\anacondafinal\Lib\site-packages\sklearn\cluster\_kmeans.py:1436:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=1.
  warnings.warn(
c:\Users\nikpa\anacondafinal\Lib\site-packages\sklearn\cluster\_kmeans.py:1436:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=1.
```

```
  warnings.warn(
c:\Users\nikpa\anacondafinal\Lib\site-packages\sklearn\cluster\_kmeans.py:1436:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=1.
  warnings.warn(
c:\Users\nikpa\anacondafinal\Lib\site-packages\sklearn\cluster\_kmeans.py:1436:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=1.
  warnings.warn(
c:\Users\nikpa\anacondafinal\Lib\site-packages\sklearn\cluster\_kmeans.py:1436:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=1.
  warnings.warn(
c:\Users\nikpa\anacondafinal\Lib\site-packages\sklearn\cluster\_kmeans.py:1436:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=1.
  warnings.warn(
```

[ ]:

```
plt.plot(Ks, WCSSs)
plt.xlabel('number of clusters')
plt.ylabel('WCSS')
plt.show()
```



For interpretability we would use 2 clusters, developed and underdeveloped economies

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.

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

```
c:\Users\nikpa\anacondafinal\Lib\site-packages\sklearn\cluster\_kmeans.py:1436:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=1.
  warnings.warn(
```

[ ]:
```
preds = pd.DataFrame(kmeams.labels_)
```

[ ]:
```
output = pd.concat([preds, df], axis = 1)
```

6 6.) Do the same for a silhoutte plot

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

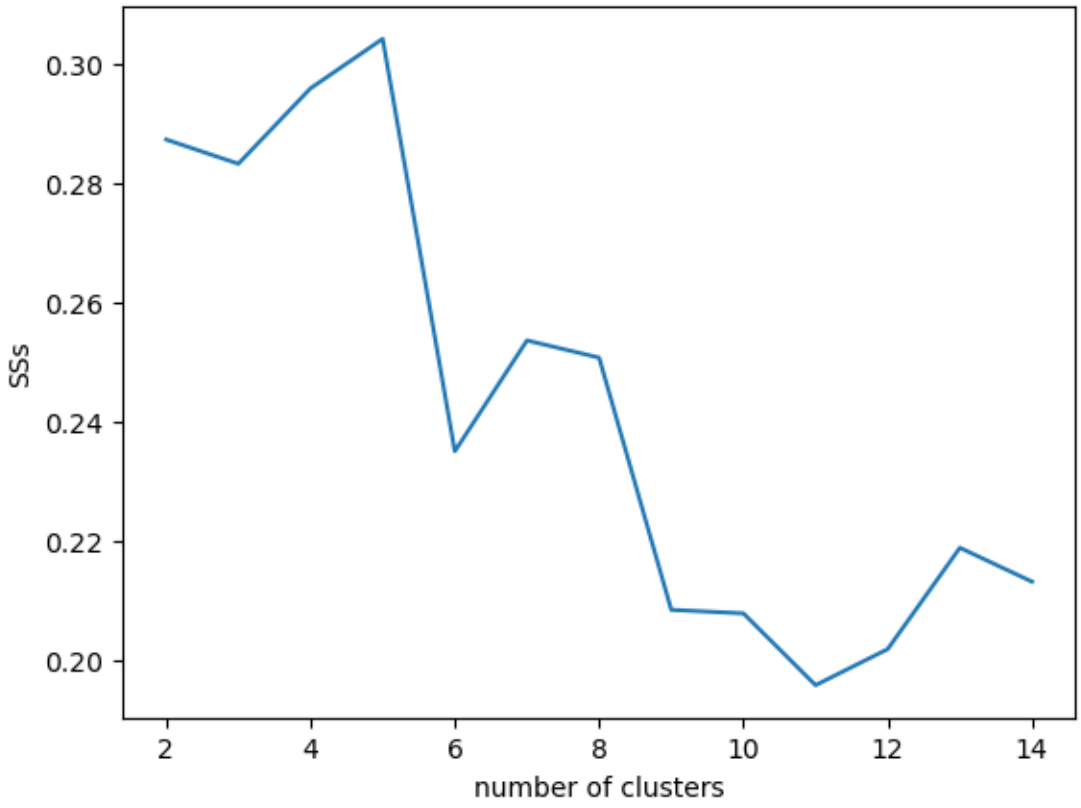
[ ]:
```python
SSs = []
Ks = range(2,15)
for  k in Ks:
    kmeans = KMeans(n_clusters=k, n_init=30).fit(X_scaled)
    sil = silhouette_score(X_scaled, kmeans.labels_)
    SSs.append(sil)
```

```
c:\Users\nikpa\anacondafinal\Lib\site-packages\sklearn\cluster\_kmeans.py:1436:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=1.
  warnings.warn(
c:\Users\nikpa\anacondafinal\Lib\site-packages\sklearn\cluster\_kmeans.py:1436:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=1.
  warnings.warn(
c:\Users\nikpa\anacondafinal\Lib\site-packages\sklearn\cluster\_kmeans.py:1436:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=1.
  warnings.warn(
c:\Users\nikpa\anacondafinal\Lib\site-packages\sklearn\cluster\_kmeans.py:1436:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=1.
  warnings.warn(
c:\Users\nikpa\anacondafinal\Lib\site-packages\sklearn\cluster\_kmeans.py:1436:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=1.
  warnings.warn(
c:\Users\nikpa\anacondafinal\Lib\site-packages\sklearn\cluster\_kmeans.py:1436:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=1.
  warnings.warn(
c:\Users\nikpa\anacondafinal\Lib\site-packages\sklearn\cluster\_kmeans.py:1436:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=1.
  warnings.warn(
c:\Users\nikpa\anacondafinal\Lib\site-packages\sklearn\cluster\_kmeans.py:1436:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
```

```
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=1.
  warnings.warn(
c:\Users\nikpa\anacondafinal\Lib\site-packages\sklearn\cluster\_kmeans.py:1436:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=1.
  warnings.warn(
c:\Users\nikpa\anacondafinal\Lib\site-packages\sklearn\cluster\_kmeans.py:1436:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=1.
  warnings.warn(
c:\Users\nikpa\anacondafinal\Lib\site-packages\sklearn\cluster\_kmeans.py:1436:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=1.
  warnings.warn(
c:\Users\nikpa\anacondafinal\Lib\site-packages\sklearn\cluster\_kmeans.py:1436:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=1.
  warnings.warn(
c:\Users\nikpa\anacondafinal\Lib\site-packages\sklearn\cluster\_kmeans.py:1436:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=1.
  warnings.warn(
```

[ ]:

```
plt.plot(Ks, SSs)
plt.xlabel('number of clusters')
plt.ylabel('SSs')
plt.show()
```



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

[ ]:
```
print("Cluster 1: ")
list(output.loc[output[0] == 0, "country"])
```

```
Cluster 1:
```

[ ]:
```
['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']
```

[ ]:
```
print("Cluster 2: ")
list(output.loc[output[0] == 1, "country"])
```

```
Cluster 2:
```

```
[ ]: ['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']
```

#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 interprotation

[ ]:
```
new_df = output.drop('country', axis = 1)
```

[ ]:
```
new_df.groupby(0).mean()
```

[ ]:

0	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

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

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

On of the most apparent observation we can make is about the child mortality rRate. The child mortality rate appears to vary significantly between the two groups. The developed countries group indicates a significantly lower child mortality rate compared to the least developed countries, suggesting huge differences in healthcare and socio-economic conditions between the countries represented. Moreover, least developed countries second appear to have lower exports in relation to imports, which suggests trade deficits or a less export-oriented economy.