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

```
from google.colab import drive
drive.mount('/content/gdrive/', force_remount = True)
```

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

```
df = pd.read_csv("/content/gdrive/MyDrive/stroke/country.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 × 10 columns

```
df.sort_values(by='life_expec', ascending = False)
```

	country	child_mort	exports	health	imports	income	inflation	life_expec	total_fer	gdpp
77	Japan	3.2	15.0	9.49	13.6	35800	-1.900	82.8	1.39	44500
133	Singapore	2.8	200.0	3.96	174.0	72100	-0.046	82.7	1.15	46600
145	Switzerland	4.5	64.0	11.50	53.3	55500	0.317	82.2	1.52	74600
68	Iceland	2.6	53.4	9.40	43.3	38800	5.470	82.0	2.20	41900
7	Australia	4.8	19.8	8.73	20.9	41400	1.160	82.0	1.93	51900
...	...	...	...	...	...	...	...	...	...	...
94	Malawi	90.5	22.8	6.59	34.9	1030	12.100	53.1	5.31	459
166	Zambia	83.1	37.0	5.89	30.9	3280	14.000	52.0	5.40	1460
31	Central African Republic	149.0	11.8	3.98	26.5	888	2.010	47.5	5.21	446
87	Lesotho	99.7	39.4	11.10	101.0	2380	4.150	46.5	3.30	1170
66	Haiti	208.0	15.3	6.91	64.7	1500	5.450	32.1	3.33	662

167 rows × 10 columns

```
names = df[["country"]]
```

```
X = df.drop(["country"], axis = 1)
```

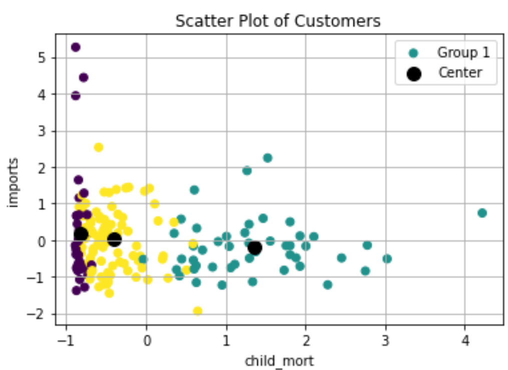
```
scaler = StandardScaler().fit(X)
X_scaled = scaler.transform(X)
```

```
kmeans = KMeans(n_clusters= 3, n_init= 20
, random_state=42).fit(X_scaled)
```

```
x1_index = 0
x2_index = 3
```

```
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', color='black', s=100)

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()
```




```
df['cluster'] = kmeans.labels_
```

```
df.groupby('cluster').mean()
```

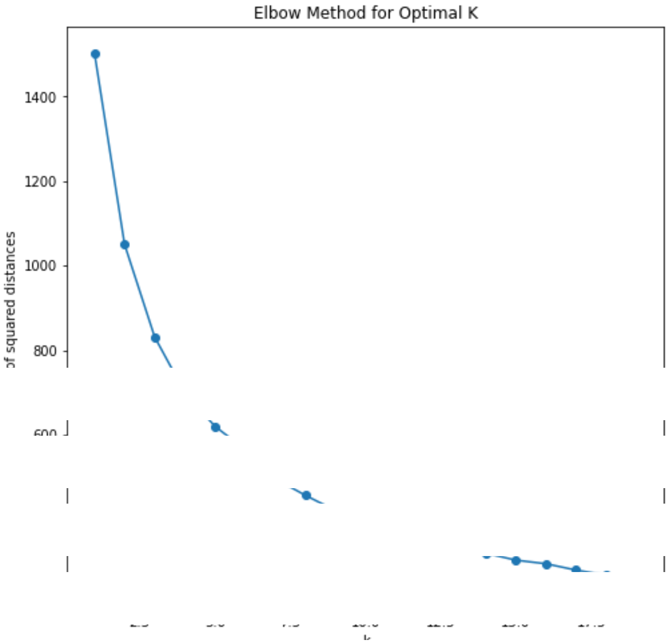
cluster	child_mort	exports	health	imports	income	inflation	life_expec	total_fer	gdpp
0	5.000000	58.738889	8.807778	51.491667	45672.222222	2.671250	80.127778	1.752778	42494.444444
1	92.961702	29.151277	6.388511	42.323404	3942.404255	12.019681	59.187234	5.008085	1922.382979
2	21.927381	40.243917	6.200952	47.473404	12305.595238	7.600905	72.814286	2.307500	6486.452381

```
sum_of_squared_distance = []

for k in range(1,20):
    kmeans = KMeans(n_clusters= k, init = 'random', n_init= 100
            , random_state=42).fit(X_scaled)
    # find SSE
    sum_of_squared_distance.append(kmeans.inertia_)
```

```
plt.figure(figsize=(8,8))

plt.plot(range(1,20), sum_of_squared_distance, marker = 'o')
plt.xlabel('k')
plt.ylabel('Sum of squared distances')
plt.title('Elbow Method for Optimal K')
plt.show()
```



I chose 10 clusters because the marginal benefit of adding clusters would make our model harder to interpret.

```
kmeans = KMeans(n_clusters= 10, init = 'random', n_init= 100
                , random_state=42).fit(X_scaled)
```

```
df['cluster'] = kmeans.labels_
```

```
df.groupby('cluster').mean()
```

cluster	child_mort	exports	health	imports	income	inflation	life_e
0	57.228571	27.971429	11.307143	77.742857	2170.000000	5.065714	61.3
1	130.000000	25.300000	5.070000	17.400000	5150.000000	104.000000	60.5
2	59.661538	37.238077	5.054615	43.265385	6578.076923	10.137692	64.5
3	4.295652	40.730435	10.513478	38.247826	40265.217391	1.334913	80.8
4	111.479167	25.096667	6.602917	39.291667	1744.291667	9.272708	56.6
5	11.050000	64.900000	3.000000	36.666667	71516.666667	13.363333	77.0
6	31.531250	26.861812	4.642500	25.154119	11003.750000	18.028750	70.6
7	16.258065	30.941935	7.298387	40.919355	12574.516129	4.751065	75.0
8	4.133333	176.000000	6.793333	156.666667	64033.333333	2.468000	81.4
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

```
df[df['cluster']==5]
```

	country	child_mort	exports	health	imports	income	inflation	life_expec	total_fer	gdpp	cluster
23	Brunei	10.5	67.4	2.84	28.0	80600	16.70	77.1	1.84	35300	5
82	Kuwait	10.8	66.7	2.63	30.4	75200	11.20	78.2	2.21	38500	5
115	Oman	11.7	65.7	2.77	41.2	45300	15.60	76.1	2.90	19300	5
123	Qatar	9.0	62.3	1.81	23.8	125000	6.98	79.5	2.07	70300	5
128	Saudi Arabia	15.7	49.6	4.29	33.0	45400	17.20	75.1	2.96	19300	5
157	United Arab Emirates	8.6	77.7	3.66	63.6	57600	12.50	76.5	1.87	35000	5

Double-click (or enter) to edit

Our clustering performed well since it grouped the gulf countries, excluding Brunei. Life expec and total_fer in all countries have realtively similar values. The gdpp in countries with low population is high. All countries except Saudi and UAE have almost the same exports.

✓ 0s completed at 5:04 PM