0.) Import and Clean data

In [2]:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
```

In [3]:

```
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
```

In [6]:

```
df = pd.read_csv("Country-data.csv", sep = ",")
```

In [77]:

```
df
```

Out[77]:

	country	child_mort	exports	health	imports	income	inflation	life_expec	total_fe
0	Afghanistan	90.2	10.0	7.58	44.9	1610	9.44	56.2	5.8
1	Albania	16.6	28.0	6.55	48.6	9930	4.49	76.3	1.6
2	Algeria	27.3	38.4	4.17	31.4	12900	16.10	76.5	2.8
3	Angola	119.0	62.3	2.85	42.9	5900	22.40	60.1	6.1
4	Antigua and Barbuda	10.3	45.5	6.03	58.9	19100	1.44	76.8	2.1
...	...	...	...	...	...	...	...	...	.
162	Vanuatu	29.2	46.6	5.25	52.7	2950	2.62	63.0	3.5
163	Venezuela	17.1	28.5	4.91	17.6	16500	45.90	75.4	2.4
164	Vietnam	23.3	72.0	6.84	80.2	4490	12.10	73.1	1.9
165	Yemen	56.3	30.0	5.18	34.4	4480	23.60	67.5	4.6
166	Zambia	83.1	37.0	5.89	30.9	3280	14.00	52.0	5.4

167 rows × 10 columns

In [8]:

```
df.columns
```

Out[8]:

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

In [9]:

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


```

In [10]:

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

In [ ]:

```

```

In [ ]:

```

```

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

In [21]:

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

In [ ]:

```

```

2.) Pick two features to visualize across

In [22]:

```
X.columns
```

Out[22]:

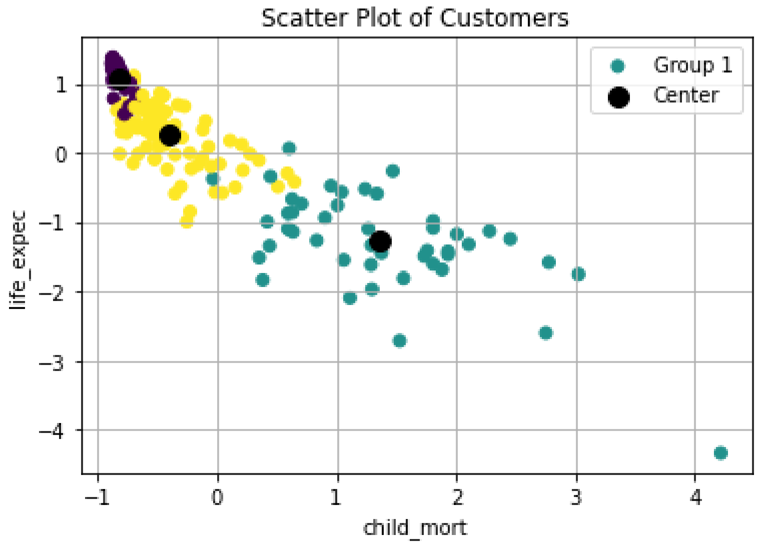
```
Index(['child_mort', 'exports', 'health', 'imports', 'income', 'inflatio
n',
       'life_expec', 'total_fer', 'gdpp'],
      dtype='object')
```

In [61]:

```
# CHANGE THESE BASED ON WHICH IS INTERESTING TO YOU
x1_index =0
x2_index = 6


plt.scatter(X_scaled[:, x1_index], X_scaled[:, x2_index], c=kmeans.lab
plt.scatter(kmeans.cluster_centers_[:, x1_index], kmeans.cluster_cente

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()
```



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

In [62]:

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

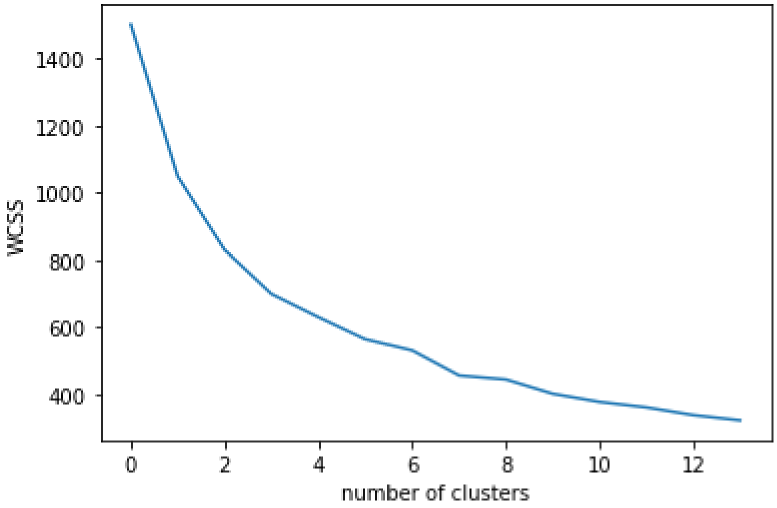
```
C:\Users\parzu\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:103
6: UserWarning: KMeans is known to have a memory leak on Windows with MK
L, when there are less chunks than available threads. You can avoid it by
setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
```

In [63]:

```
plt.plot(WCSSs)
plt.xlabel("number of clusters")
plt.ylabel ("WCSS")
plt.show
```

Out[63]:

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



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 [78]:

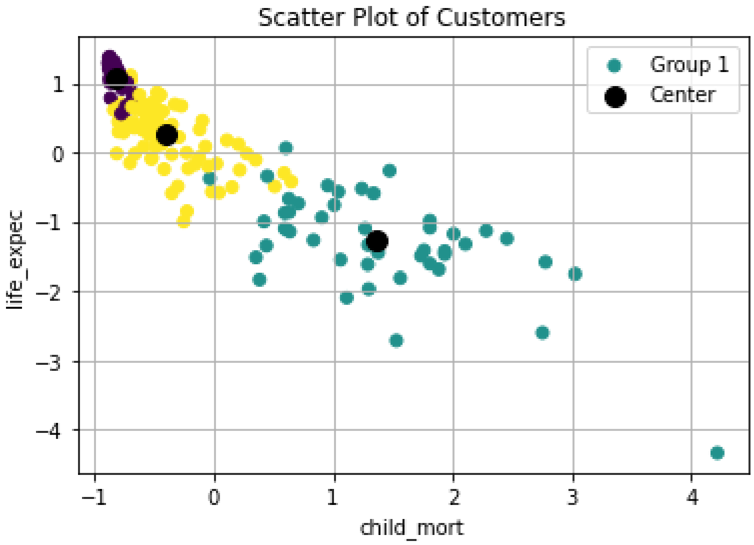
```
kmeans = KMeans(n_clusters=3, random_state=42).fit(X_scaled)
```

In [79]:

```
x1_index = 0
x2_index = 6

plt.scatter(X_scaled[:, x1_index], X_scaled[:, x2_index], c=kmeans.lab
plt.scatter(kmeans.cluster_centers_[:, x1_index], kmeans.cluster_cente

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", "Group 3", "Group 4"])
plt.grid()
plt.show()
```



I have chosen a cluster number of 3 because from above, there seems to be a convergence/elbow point around 2 to 3

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

In [80]:

```
cluster_labels = kmeans.predict(X_scaled)

# Loop through each cluster and print out the countries in that cluste
for i in range(3):
    print(f"Cluster {i+1} countries:")
    countries = list(X.index[cluster_labels == i])
    print(countries)
```

```
Cluster 1 countries:
[7, 8, 11, 15, 23, 29, 42, 43, 44, 53, 54, 58, 60, 68, 73, 74, 75, 77, 8
2, 91, 98, 110, 111, 114, 122, 123, 133, 134, 135, 138, 139, 144, 145, 15
7, 158, 159]
Cluster 2 countries:
[0, 3, 17, 21, 25, 26, 28, 31, 32, 36, 37, 38, 40, 49, 50, 55, 56, 59, 6
3, 64, 66, 72, 80, 81, 84, 87, 88, 93, 94, 97, 99, 106, 108, 112, 113, 11
6, 126, 129, 132, 137, 142, 147, 149, 150, 155, 165, 166]
Cluster 3 countries:
[1, 2, 4, 5, 6, 9, 10, 12, 13, 14, 16, 18, 19, 20, 22, 24, 27, 30, 33, 3
4, 35, 39, 41, 45, 46, 47, 48, 51, 52, 57, 61, 62, 65, 67, 69, 70, 71, 7
6, 78, 79, 83, 85, 86, 89, 90, 92, 95, 96, 100, 101, 102, 103, 104, 105,
107, 109, 115, 117, 118, 119, 120, 121, 124, 125, 127, 128, 130, 131, 13
6, 140, 141, 143, 146, 148, 151, 152, 153, 154, 156, 160, 161, 162, 163,
164]
```

Both clusters contain similar countries. However, cluster 3 contains almost all countries.

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 [81]:

```
# Get cluster centroids
centroids = kmeans.cluster_centers_

# Create a DataFrame to store the statistics
stats_df = pd.DataFrame(centroids, columns=X.columns)
stats_df.index.name = 'Cluster'
stats_df.reset_index(inplace=True)

print(stats_df.to_string(index=False))
```

```
 Cluster  child_mort   exports    health   imports    income  inflation
life_expec  total_fer      gdpp
       0   -0.827449  0.645080  0.727411  0.190639  1.484243  -0.484921
1.079579  -0.791877  1.615995
       1    1.360218 -0.437533 -0.155984 -0.189204 -0.686894   0.402111
-1.282180   1.364944 -0.604242
       2   -0.406453 -0.031653 -0.224471  0.024162 -0.251770  -0.017167
0.254734  -0.424343 -0.354481
```

Q7.) Write an observation about the descriptive statistics.

Compared to the mean values in the other clusters, child mort's mean value is -0.827. This shows that nations in cluster 0 have a lower child mortality rate than those in the other groups. Similar to exports, health, income, life expec, and gdpp, cluster 0 has greater mean values than the other clusters, showing that the nations in this cluster have higher levels of these variables than the clusters. On the other side, cluster 1 has the lowest mean values for income, life expectancy, and GDP per capita and the highest mean value for child mort, indicating that nations in this cluster have higher child mortality rates.

In [ ]: