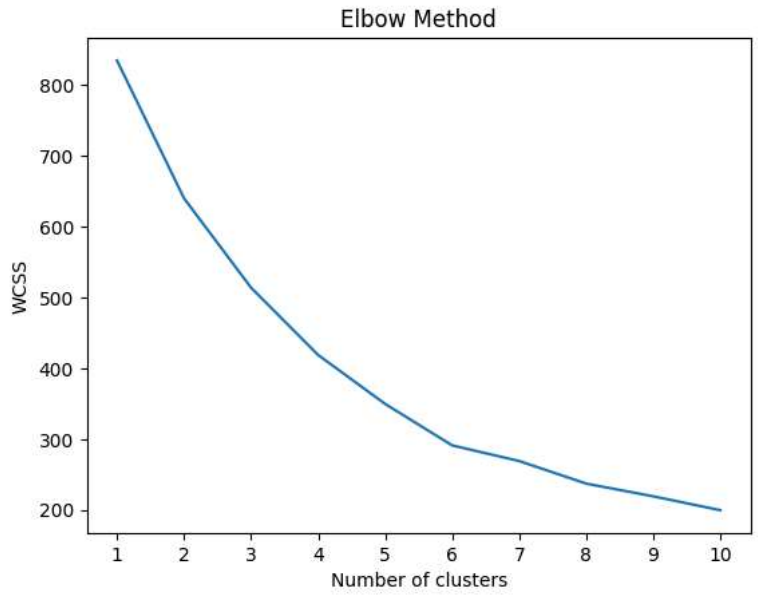
```
import pandas as pd
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt


# Load the dataset
data = pd.read_csv('Country_data.csv')

# Select relevant features for clustering
X = data[['child_mort', 'exports', 'health', 'imports', 'income']]


# Standardize the features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)


# Determine the optimal number of clusters using the Elbow method
wcss = []

for i in range(1, 11):
    kmeans = KMeans(n_clusters=i, random_state=42)
    kmeans.fit(X_scaled)
    wcss.append(kmeans.inertia_)


# Plot the Elbow method graph with x-axis ticks in intervals of 1
plt.plot(range(1, 11), wcss)
plt.title('Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.xticks(range(1, 11))
plt.show()
```




```
n_clusters = 3  # Choose the number of clusters based on the Elbow method
kmeans = KMeans(n_clusters=n_clusters, init='k-means++', random_state=42)
kmeans.fit(X_scaled)

# Add the cluster labels to the original dataset
data['Cluster'] = kmeans.labels_


# Plot clusters with centroids
plt.figure(figsize=(10, 6))

for cluster_id in range(n_clusters):
    cluster_data = X_scaled[data['Cluster'] == cluster_id]
    plt.scatter(cluster_data[:, 0], cluster_data[:, 4], label=f'Cluster {cluster_id}')
```

```
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 4], s=300, c='red', marker='x', label='Centroids')
plt.title('Clusters with Centroids')
plt.xlabel('Income')
plt.ylabel('Child Mortality Rate')
plt.legend()
plt.grid(True)
plt.show()
```