# CLUSTERING ASSIGNMENT- HELP INTERNATIONAL

Rishabh Gupta

# Problem Statement

- HELP International, an NGO wants to help the countries fight poverty and help them at the time of disaster and calamity by providing funds and resources. They want to weigh in socio economic and health factors of all the countries and choose the top 5 countries who are in dire need of teh resources

- Aim is to cluster the data sets based on socio economic, health and overall development of the country to segment them and then analyse based on GDP per capita, Income and Child Mortality to recommend top 5 countries to the CEO.

# Approach

**Steps we'll follow with the assignment are as follows:**

- Data Inspection and EDA - Preparation and Basic Analysis
- Outlier Analysis and Scaling - Data Preparation
- Checking the tendency of the data - Hopkins Statistics
- Modelling- Both Kmeans and Heirarchical
- Visualizations and Cluster Profiling
- Results and Recommendations

# Reading Data

```
# Import Libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

import datetime as dt

import sklearn
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score

from scipy.cluster.hierarchy import linkage
from scipy.cluster.hierarchy import dendrogram
from scipy.cluster.hierarchy import cut_tree

import warnings
warnings.filterwarnings('ignore')
```

```
# read the dataset

df = pd.read_csv("Country-data.csv", header=0)
df.head()
```

| | 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 |

```
df.shape
```

```
(167, 10)
```

```
df.columns
```

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

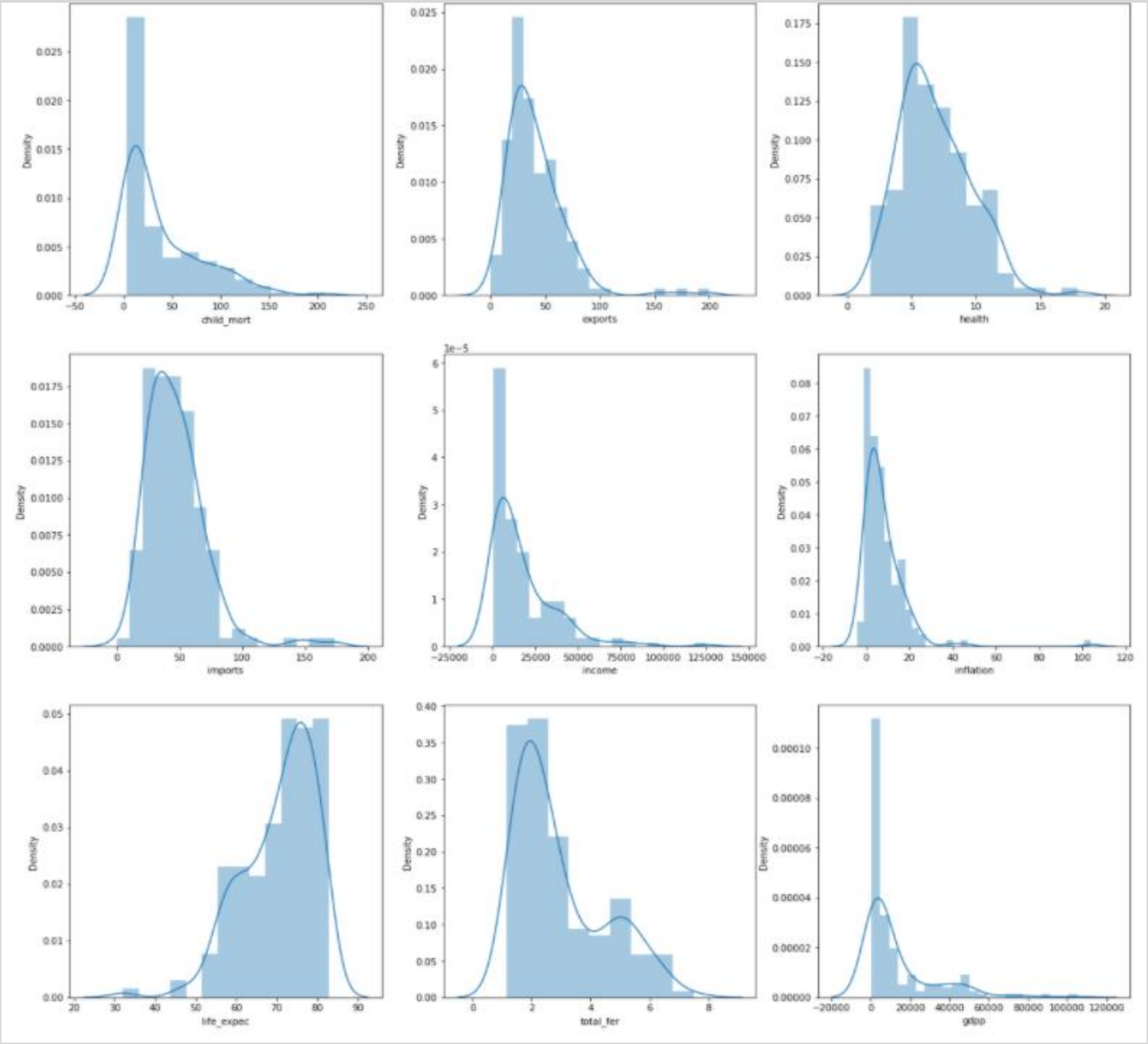
**Since most columns are already in the form we want for clustering we do not need to change/ cpnvert anything, we can start with our EDA to get us a primilary analysis, for that we will create another df which will have all numeric data which we will actually use in clustering. Also we would use all the variables for analysis and just use gdpp, income and child_mort for cluster profiling**

```
# Creating numerical df

num_df = df[['child_mort', 'exports', 'health', 'imports', 'income','inflation', 'life_expec', 'total_f
num_df.head()
```

| | child_mort | exports | health | imports | income | inflation | life_expec | total_fer | gdpp |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 90.2 | 10.0 | 7.58 | 44.9 | 1610 | 9.44 | 56.2 | 5.82 | 553 |
| 1 | 16.6 | 28.0 | 6.55 | 48.6 | 9930 | 4.49 | 76.3 | 1.65 | 4090 |
| 2 | 27.3 | 38.4 | 4.17 | 31.4 | 12900 | 16.10 | 76.5 | 2.89 | 4460 |
| 3 | 119.0 | 62.3 | 2.85 | 42.9 | 5900 | 22.40 | 60.1 | 6.16 | 3530 |
| 4 | 10.3 | 45.5 | 6.03 | 58.9 | 19100 | 1.44 | 76.8 | 2.13 | 12200 |

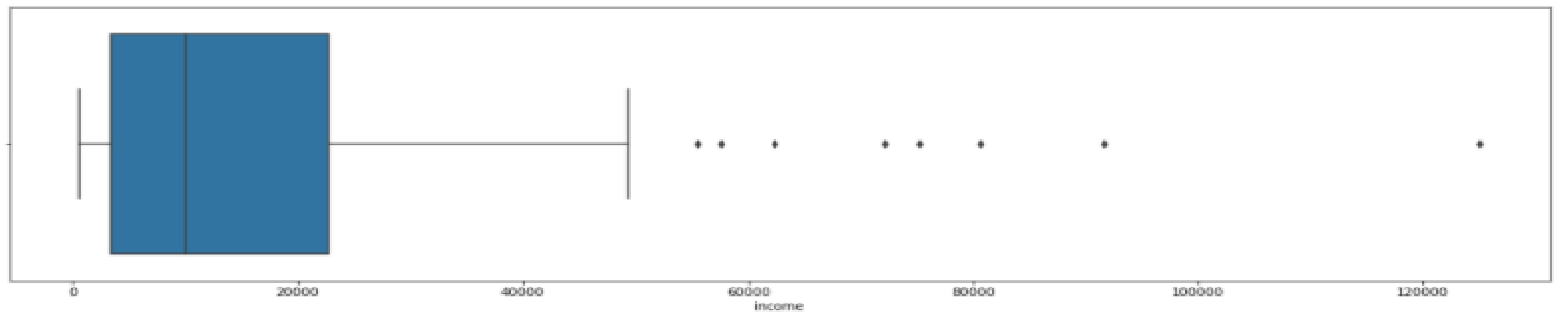
# Univariate Analysis

- For Univariate Analysis we created displots and box plots to check distribution and outliers respectively
- Displots Inferences: Income, gdpp, child_mort, total_fer do not seem to be following the normal ditribution as teh initial values are too big and they decrease as the variable values increase, anyway we were using them to profile our clusters

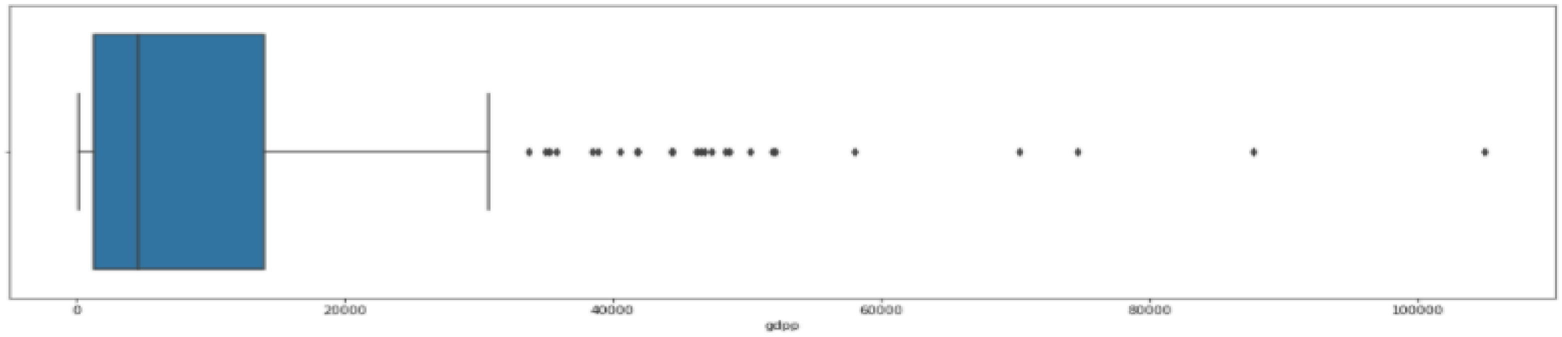


# Box Plots and Inferences

<AxesSubplot:xlabel='income'>

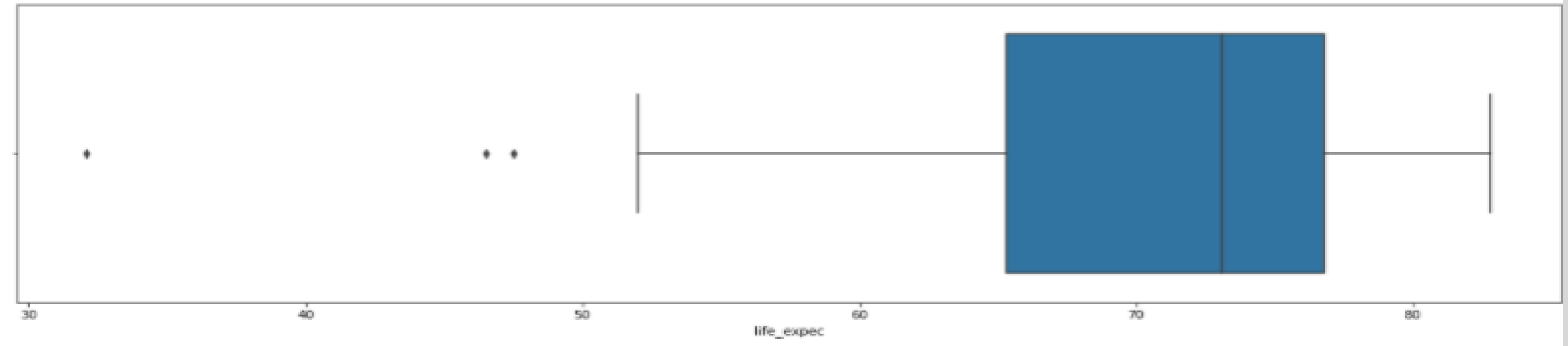



**There are a couple of outliers on the higher side but they seem to be very few and based on business knowledge we may expect some nations net income to be very large, we might want to treat them with a upper cap**

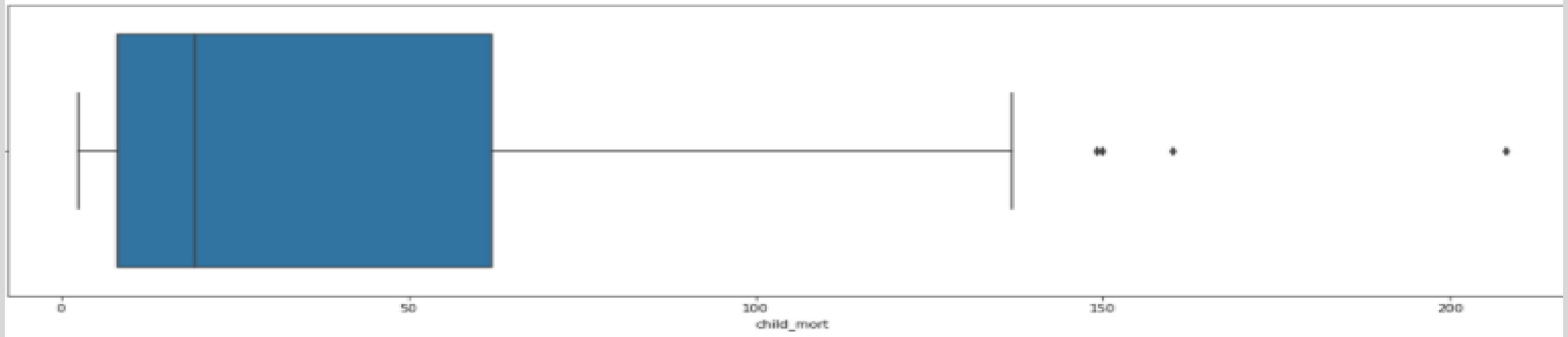



**There are a couple of outliers on the higher side but they seem to be very few and based on business knowledge we may expect some nations net gdpp to be very large, we might want to treat them with quantile upper cap**

# Box Plots and Inferences




**There are a few countries with life expectancy less that 40 years also, this doesn't seem right, although we might need expect advise on this, we will remove the lower outliers**




**Again, very few outliers here as well, but a mortality rate >200 seems a bit off,but if we see even the 75%ile is really high as well, so lets not do any treatment here. We can come back here after firts iteration of analysis**

# Outlier Treatment

**Not a lot of outliers so would not impact our analysis much, hence we do not do anything. Post this we see that we want to do our outlier treatment on gdpp and income (upper cap treatment) and life_expec (lower cap treatment)**

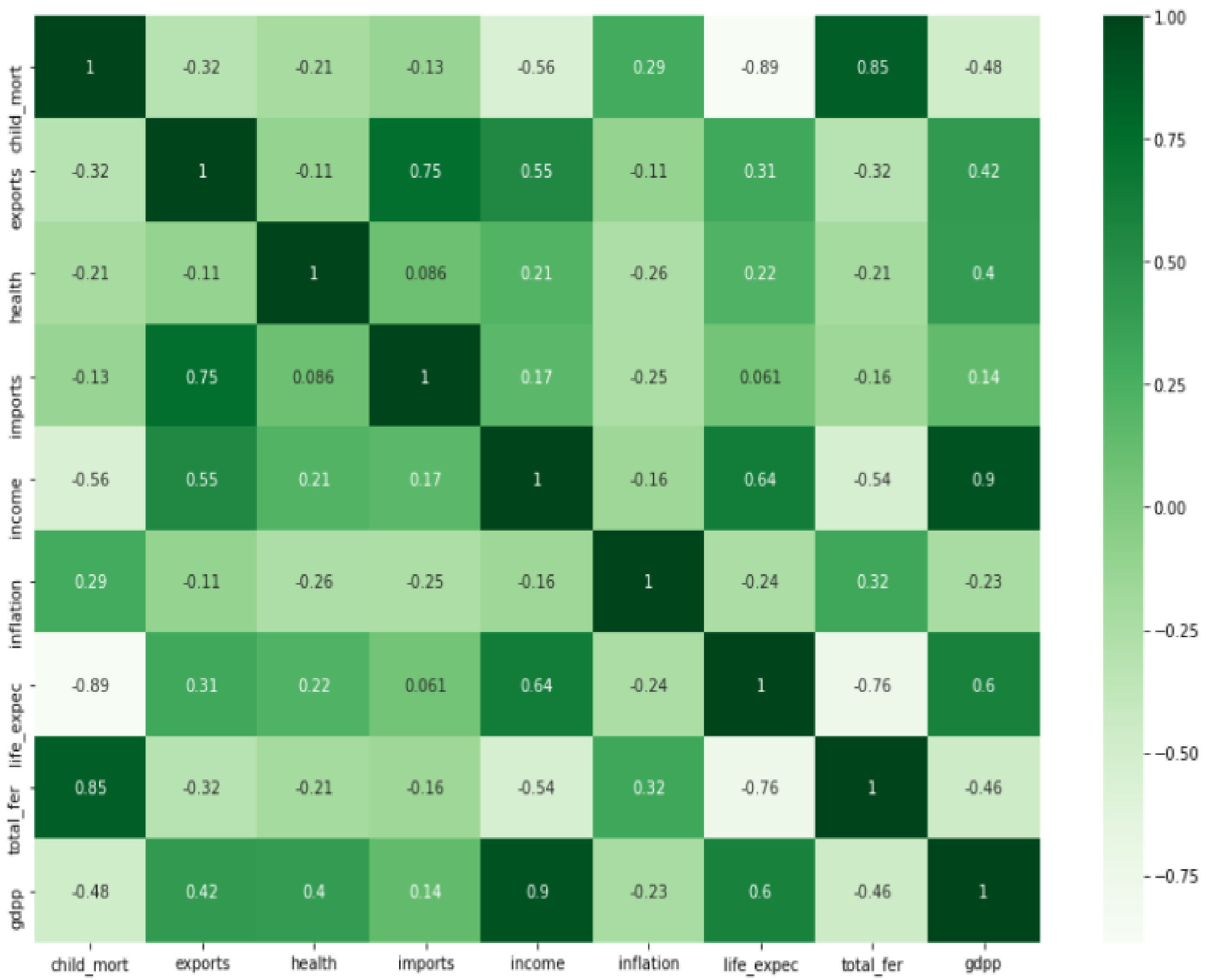
```
# removing (statistical) outliers
Q1 = df.gdpp.quantile(0.05)
Q3 = df.gdpp.quantile(0.95)
IQR = Q3 - Q1
df = df[(num_df.gdpp <= Q3 + 1.5*IQR)]


Q1 = num_df.income.quantile(0.05)
Q3 = num_df.income.quantile(0.95)
IQR = Q3 - Q1
df = df[(num_df.income <= Q3 + 1.5*IQR)]


Q1 = num_df.life_expec.quantile(0.05)
Q3 = num_df.life_expec.quantile(0.95)
IQR = Q3 - Q1
df = df[(num_df.life_expec >= Q1 - 1.5*IQR)]
```

# Bivariate Analysis

- As part of the bivariate analysis we'll do correlation matrix, heatmaps, scatter plots in order to find teh relationships between variables and see how best we can go forward with our clustering modelling
- Inferences: From the heatmap we see that Income and Gdpp are very highly co-related and child_mort and Total_fer too are pretty much very highly correlated, also life_expec and child_mortality is highly negetively correlated and hence, we might want to see if we want to include all variables in our clustering modelling, since a couple of them might be double counted, we might need business insights here, because even though some variables are highly correlated the meaning of those are different and could be important factors in segmentation. So overall we wont be dropping anything and would do our analysis based on all the variables

```
plt.figure(figsize = (15,10))
sns.heatmap(df_corr, cmap = 'Greens', annot = True)
```

<AxesSubplot:>



# Scaling

## Scaling

```
# Scaling
scaler = StandardScaler()

# fit_transform
num_df_scaled = scaler.fit_transform(num_df)
num_df_scaled.shape
```

```
(166, 9)
```

```
# Changing scaled data to a df

num_df_scaled = pd.DataFrame(num_df_scaled)
num_df_scaled.columns = num_df.columns
num_df_scaled.head()
```

| | child_mort | exports | health | imports | income | inflation | life_expec | total_fer | gdpp |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 1.285341 | -1.132262 | 0.270004 | -0.088197 | -0.857748 | 0.156410 | -1.613131 | 1.895616 | -0.678732 |
| 1 | -0.542572 | -0.474423 | -0.108804 | 0.065058 | -0.378307 | -0.311873 | 0.653990 | -0.861767 | -0.479766 |
| 2 | -0.276829 | -0.094338 | -0.984108 | -0.647370 | -0.207160 | 0.786464 | 0.676549 | -0.041826 | -0.458953 |
| 3 | 2.000611 | 0.779125 | -1.469570 | -0.171037 | -0.610536 | 1.382462 | -1.173241 | 2.120438 | -0.511268 |
| 4 | -0.699037 | 0.165142 | -0.300047 | 0.491687 | 0.150116 | -0.600412 | 0.710386 | -0.544370 | -0.023556 |

# Hopkin Statistics and Modelling Steps

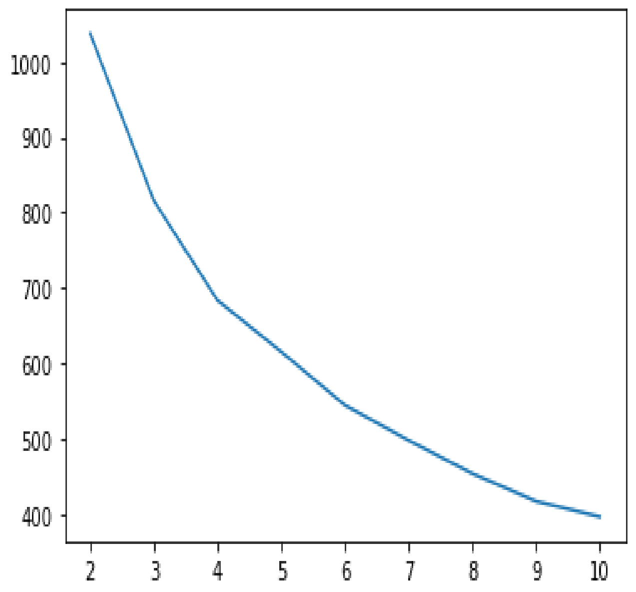
```
hopkins(num_df_scaled)
```

```
0.8687164446649124
```

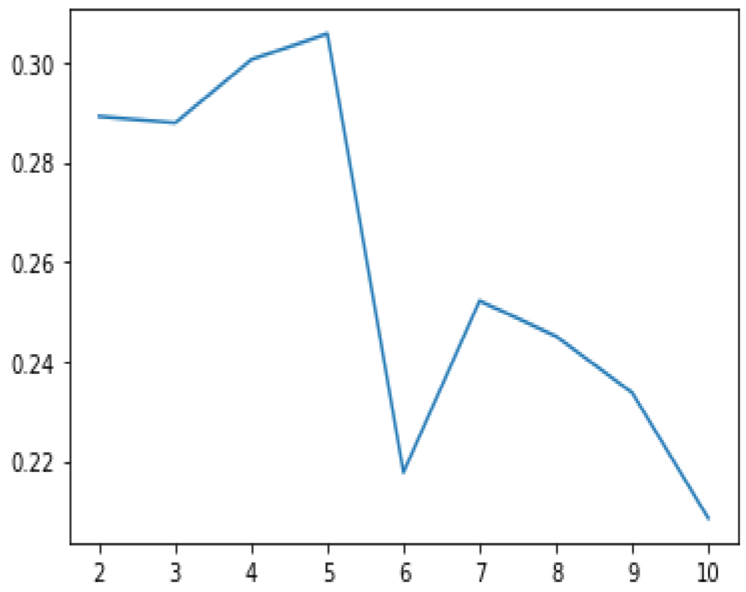
**The value of Hopkins metric is pretty high so we can go forward and do our modelling on the data set. So for modelling we would focus on K- means and Heirarchical modelling. We would do the following steps in modelling:**

1. Determine optimal K using elbow curve method and silhouette method
2. Perform K means using final value of K
3. Visualize the clusters using scatter plots and box plots
4. Perform heirarchical clustering
5. Plot single linkage Dendogram
6. Plot complete linkage Dendogram
7. Visualize the clusters

# K means Clustering – Finding K

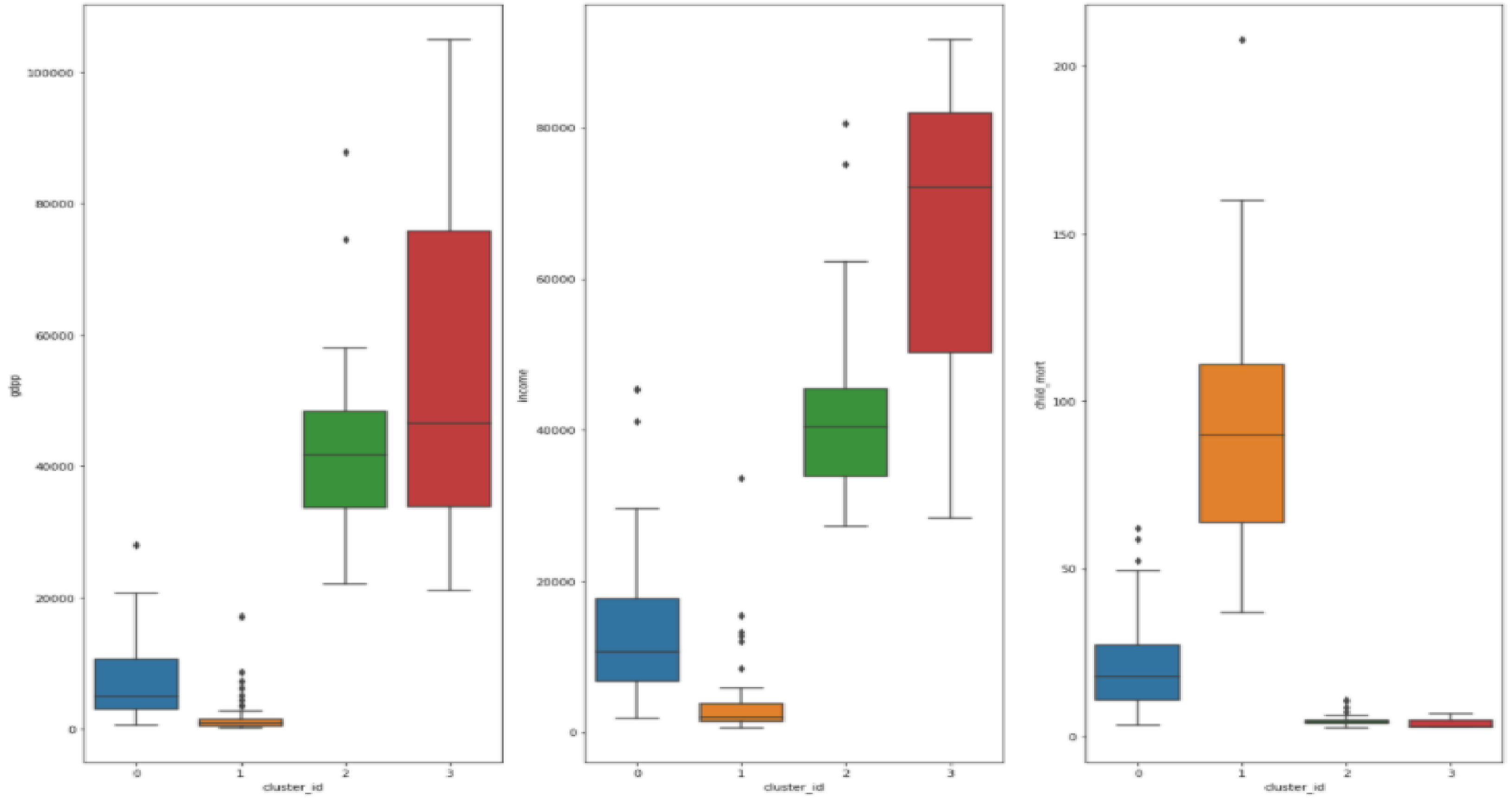



So by using elbow curve method, we get a 3 as the K value, but we could use 4 as well the change of slope is not too big from 3 to 4. Lets see what Sihouette score gives us, if we are still confused we would go ahead with both 3 and 4 and see what gives us better results




From Silhouette method we get 4 as the highest score, so lets go ahead and use 4 as the K value and then we can repeat our modelling using 3 and then compare the results

# Using k = 4 first since it had the highest SS value




| | gdpp | child_mort | income |
|---|---|---|---|
| **cluster_id** | | | |
| **0** | 6988.069767 | 20.889535 | 13076.162791 |
| **1** | 1902.916667 | 92.366667 | 3937.770833 |
| **2** | 42403.448276 | 4.813793 | 42500.000000 |
| **3** | 57566.666667 | 4.133333 | 64033.333333 |

**So we see that the clusters 0 and 2 state somewhat the same facts and the 1, 3 are quite similar, we if we chage to 3 clusters most likely cluster number 0 and 2 will merge. Also from the graph we see that cluster number 3 has teh least GDP and Income and Cluster number 2 have most GDP, Income and least Child Mortality rate, which means we can safely say cluster 2 are teh most developed countries**

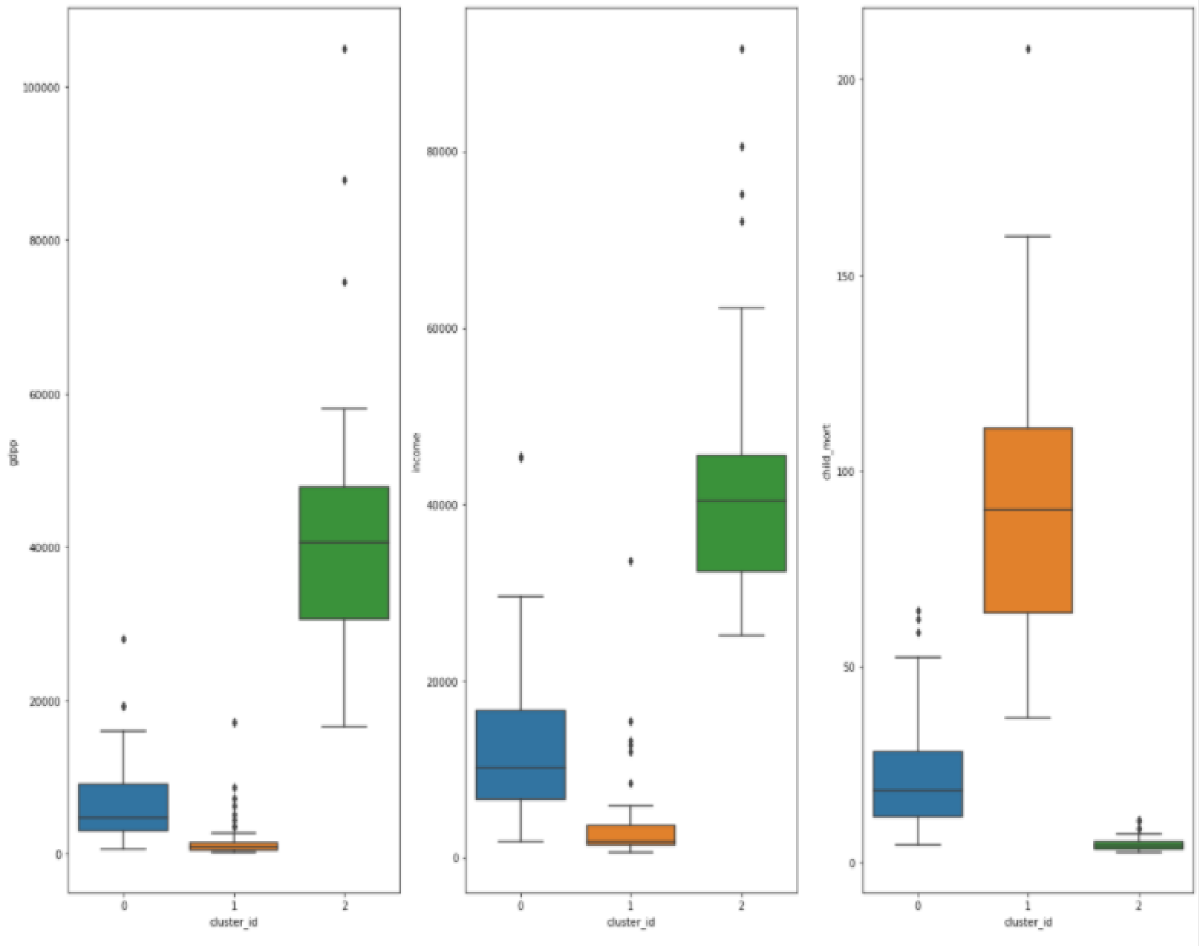
# Using k = 3 now

```
mean_df = df[['gdpp', 'child_mort', 'income', 'cluster_id']].groupby('cluster_id').mean()
mean_df
```

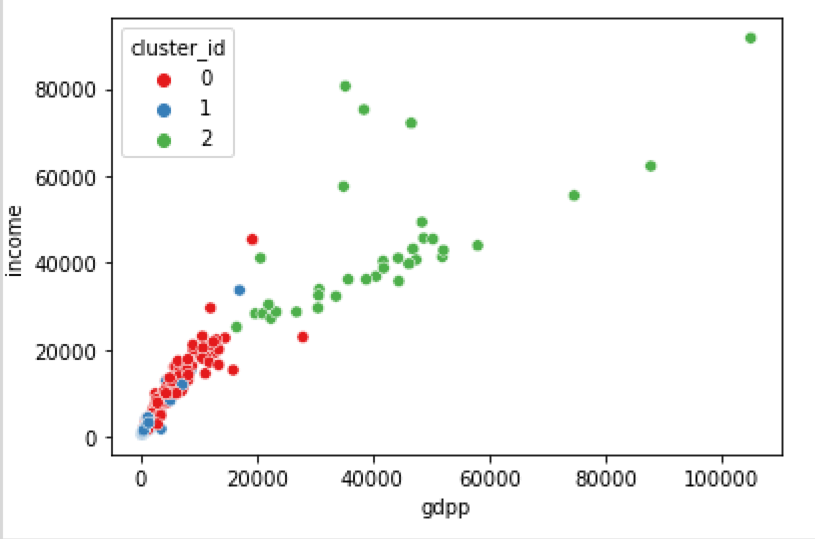
| cluster_id | gdpp | child_mort | income |
|---|---|---|---|
| 0 | 6486.452381 | 21.927381 | 12305.595238 |
| 1 | 1922.382979 | 92.961702 | 3942.404255 |
| 2 | 41700.000000 | 4.885714 | 43405.714286 |

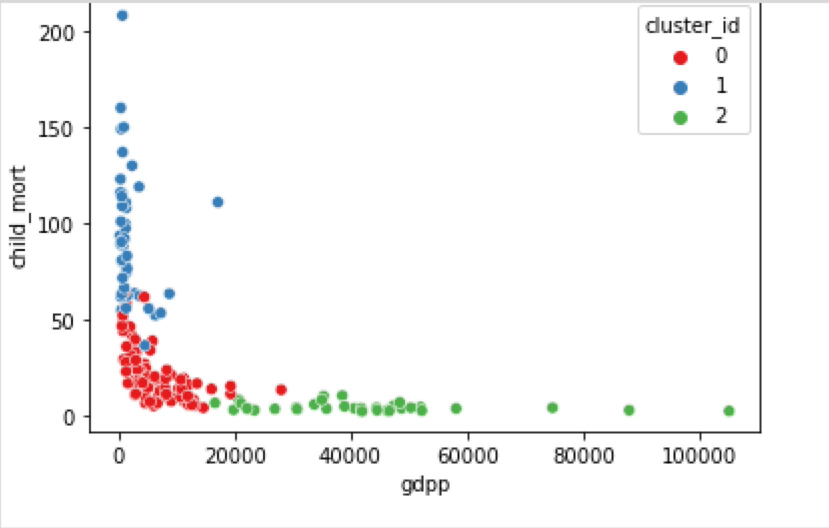
***From this we see that cluster ID 1 are the most under-developed countries and cluster number 2 are the most developed countries because the per capita GDP and Income is soo much higher than other clusters and child mortality rate is the lowest***

# K= 3 Interpretation




We realize that cluster 2 has the highest GDP and Income and lowest Child Mortality Rate, also we realize that cluster 1 has highest child mortality rate and lowest GDP and Income, this is the population which the CEO should look at



# K- means recommendations

```
c1.sort_values(by = ['gdpp', 'child_mort', 'income'], ascending = [True, False, True]).head(5)
```

| | country | child_mort | exports | health | imports | income | inflation | life_expec | total_fer | gdpp | cluster_id |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **26** | Burundi | 93.6 | 8.92 | 11.60 | 39.2 | 764 | 12.30 | 57.7 | 6.26 | 231 | 1 |
| **88** | Liberia | 89.3 | 19.10 | 11.80 | 92.6 | 700 | 5.47 | 60.8 | 5.02 | 327 | 1 |
| **37** | Congo, Dem. Rep. | 116.0 | 41.10 | 7.91 | 49.6 | 609 | 20.80 | 57.5 | 6.54 | 334 | 1 |
| **112** | Niger | 123.0 | 22.20 | 5.16 | 49.1 | 814 | 2.55 | 58.8 | 7.49 | 348 | 1 |
| **132** | Sierra Leone | 160.0 | 16.80 | 13.10 | 34.5 | 1220 | 17.20 | 55.0 | 5.20 | 399 | 1 |

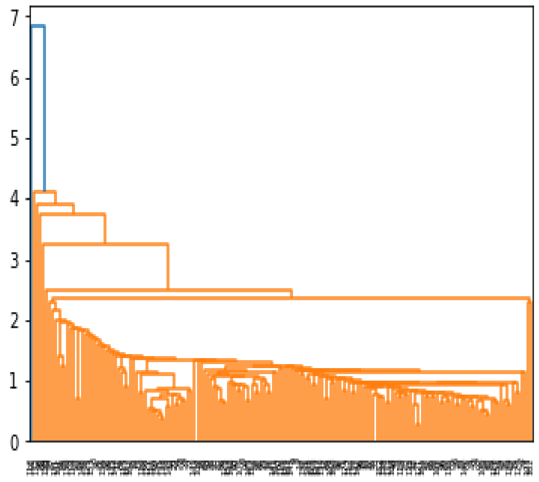
**When we choose cluster 0 and sort it by the per capita GDP, Child Mortality Rate, Income we find that the top 5 countries in dire need to relief funds are Burundi, Liberia, Congo Republic, Niger and Sierra Leone**

**Lets now move forward to the heirarchical clustering to see whether our results match and if not what differences we observe**

# Hierarchical clustering

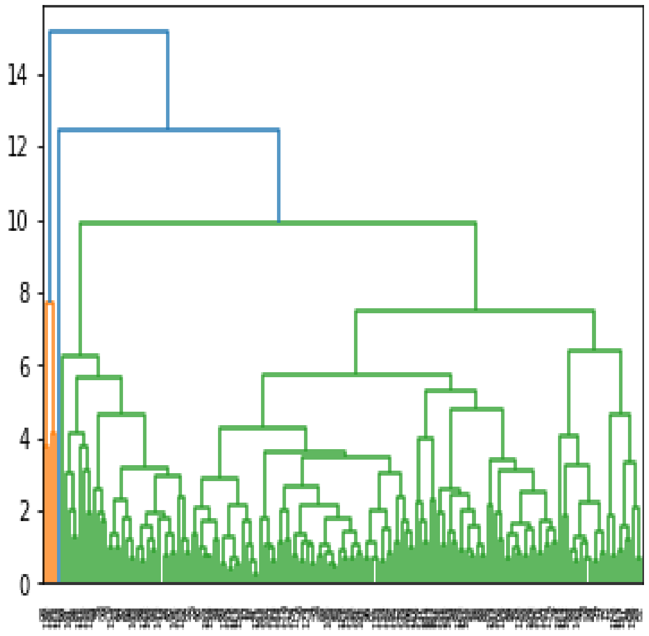
**using Single Linkage first**

```
# single linkage
mergings = linkage(num_df_scaled, method="single", metric='euclidean')
dendrogram(mergings)
plt.show()
```




**using Complete Linkage now as the densogram of the single linkage is not very intuitive and we can't draw much insights from it**

```
# complete linkage
mergings = linkage(num_df_scaled, method="complete", metric='euclidean')
dendrogram(mergings)
plt.show()
```



# K = 3 results when length = 10

```
df.cluster_labels_3.value_counts()
```

```
0    161
1      4
2      1
Name: cluster_labels_3, dtype: int64
```

**When we cluster by choosing 3 we do not have a very great segmentation as most of the records get segmented in the cluster 0 . To change this, lets choose length of dendrogram to be 9 and hence we would use 4 clusters this time to segment our data**

# K = 4 when length = 9

```
df.cluster_labels.value_counts()
```

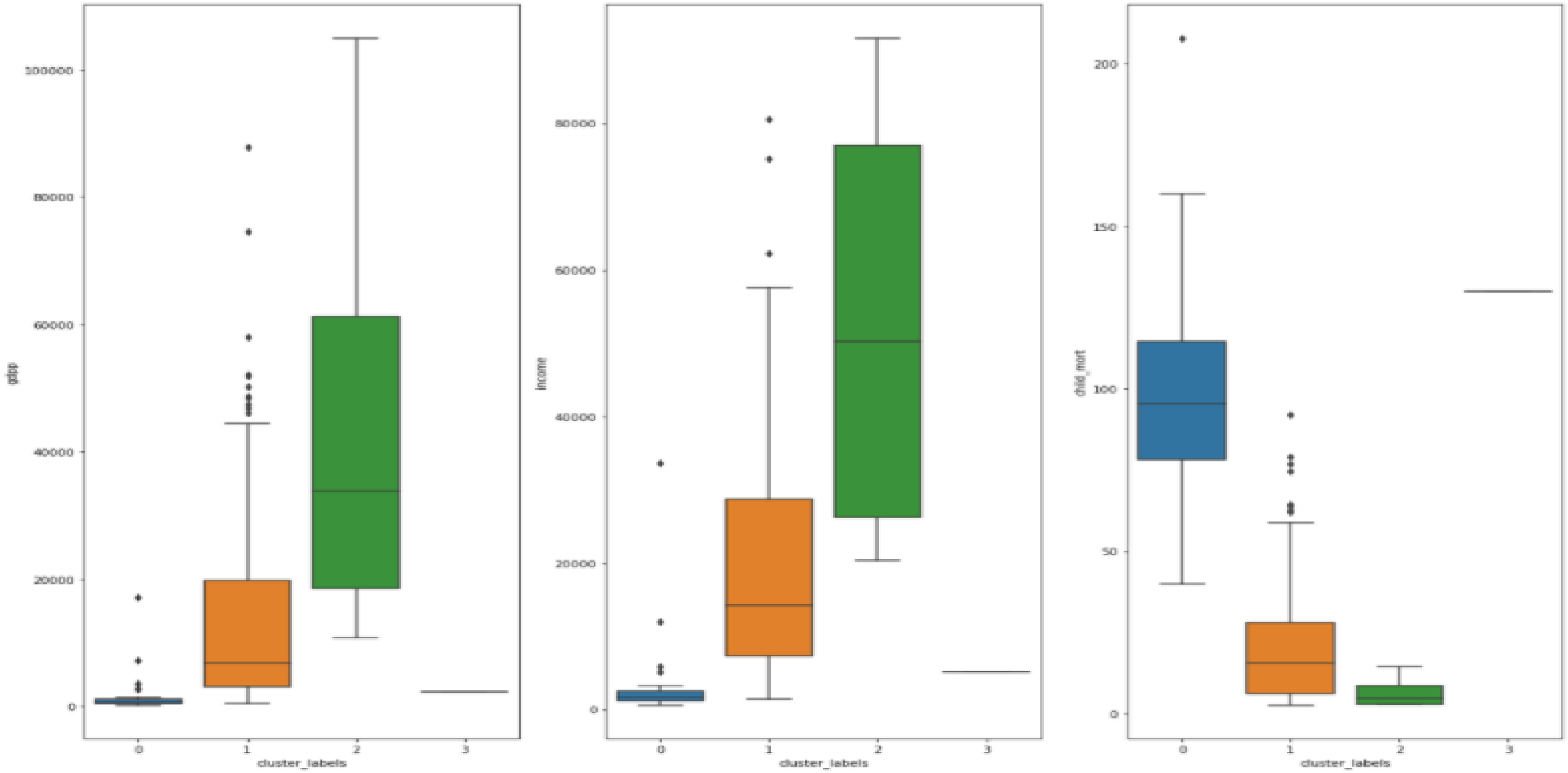
```
1    125
0     36
2      4
3      1
Name: cluster_labels, dtype: int64
```

**Clustering into 4 clusters is so much btter than the 3 ones but still cluster 2 and cluster 3 have very less records, we can try and cluster into 5 segments but because we do not want a lot of clusters lets just go forward and analyse what we see from these 4 clusters**

```
mean_df2 = df[['gdpp', 'child_mort', 'income', 'cluster_labels']].groupby('cluster_labels').mean()
mean_df2
```

| | gdpp | child_mort | income |
|---|---|---|---|
| **cluster_labels** | | | |
| **0** | 1548.055556 | 99.266667 | 3088.416667 |
| **1** | 14825.232000 | 21.213600 | 19274.640000 |
| **2** | 45875.000000 | 6.700000 | 53125.000000 |
| **3** | 2330.000000 | 130.000000 | 5150.000000 |

# Interpretation of clusters




**We see that 0 is the cluster where chirl mortality rate is quite high and income and gdp are pretty less, also 3 is somewhat near to 0 but 3 has only 1 record so we should not take that into account, 1 and 2 have high income and gdp and very less child mortality rate so these countires seem to be developed**

# Hierarchical clustering – Recommendations

```
c2.sort_values(by = ['gdpp', 'child_mort', 'income'], ascending = [True, False, True]).head(5)
```

| | country | child_mort | exports | health | imports | income | inflation | life_expec | total_fer | gdpp | cluster_id | cluster_labels_3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 26 | Burundi | 93.6 | 8.92 | 11.60 | 39.2 | 764 | 12.30 | 57.7 | 6.26 | 231 | 1 | 0 |
| 88 | Liberia | 89.3 | 19.10 | 11.80 | 92.6 | 700 | 5.47 | 60.8 | 5.02 | 327 | 1 | 0 |
| 37 | Congo, Dem. Rep. | 116.0 | 41.10 | 7.91 | 49.6 | 609 | 20.80 | 57.5 | 6.54 | 334 | 1 | 0 |
| 112 | Niger | 123.0 | 22.20 | 5.16 | 49.1 | 814 | 2.55 | 58.8 | 7.49 | 348 | 1 | 0 |
| 132 | Sierra Leone | 160.0 | 16.80 | 13.10 | 34.5 | 1220 | 17.20 | 55.0 | 5.20 | 399 | 1 | 0 |

**After choosing 0 as our desired cluster and sorting the values based on GDP, Child Mortality and Income we find the top 5 countries in dire need to relief fiunds are Burundi, Liberia, Congo Republic, Niger and Sierra Leone**

# Final Conclusion

- We see that the results and the recommendations to the CEO are the same in the cases of both K means and Hierarchical Clustering which are Burundi, Liberia, Congo Republic, Niger and Sierra Leone, even though the number of clusters observed in both the methods were different.
- K means should be used when the problem statement is simpler and we have less data, hierarchical is more difficult to scale once the data increases.