Clustering/PCA

Raviraj Kuber

Data Cleansing/Massaging

In [4]:
```
# Basic checks on data
base_data_df.country.value_counts()
base_data_df.info()
base_data_df.isnull().sum()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 167 entries, 0 to 166
Data columns (total 10 columns):
country       167 non-null object
child_mort    167 non-null float64
exports       167 non-null float64
health        167 non-null float64
imports       167 non-null float64
income        167 non-null int64
inflation     167 non-null float64
life_expec    167 non-null float64
total_fer     167 non-null float64
gdpp          167 non-null int64
dtypes: float64(7), int64(2), object(1)
memory usage: 13.1+ KB
```

Out[4]:
```
country       0
child_mort    0
exports       0
health        0
imports       0
income        0
inflation     0
life_expec    0
total_fer     0
gdpp          0
dtype: int64
```

Observations- All columns Non Null. No Need of Dropping any columns/rows

All Columns are Non-Null. Hence no dropping of columns or any Data Infusion Required.

Outlier Treatment

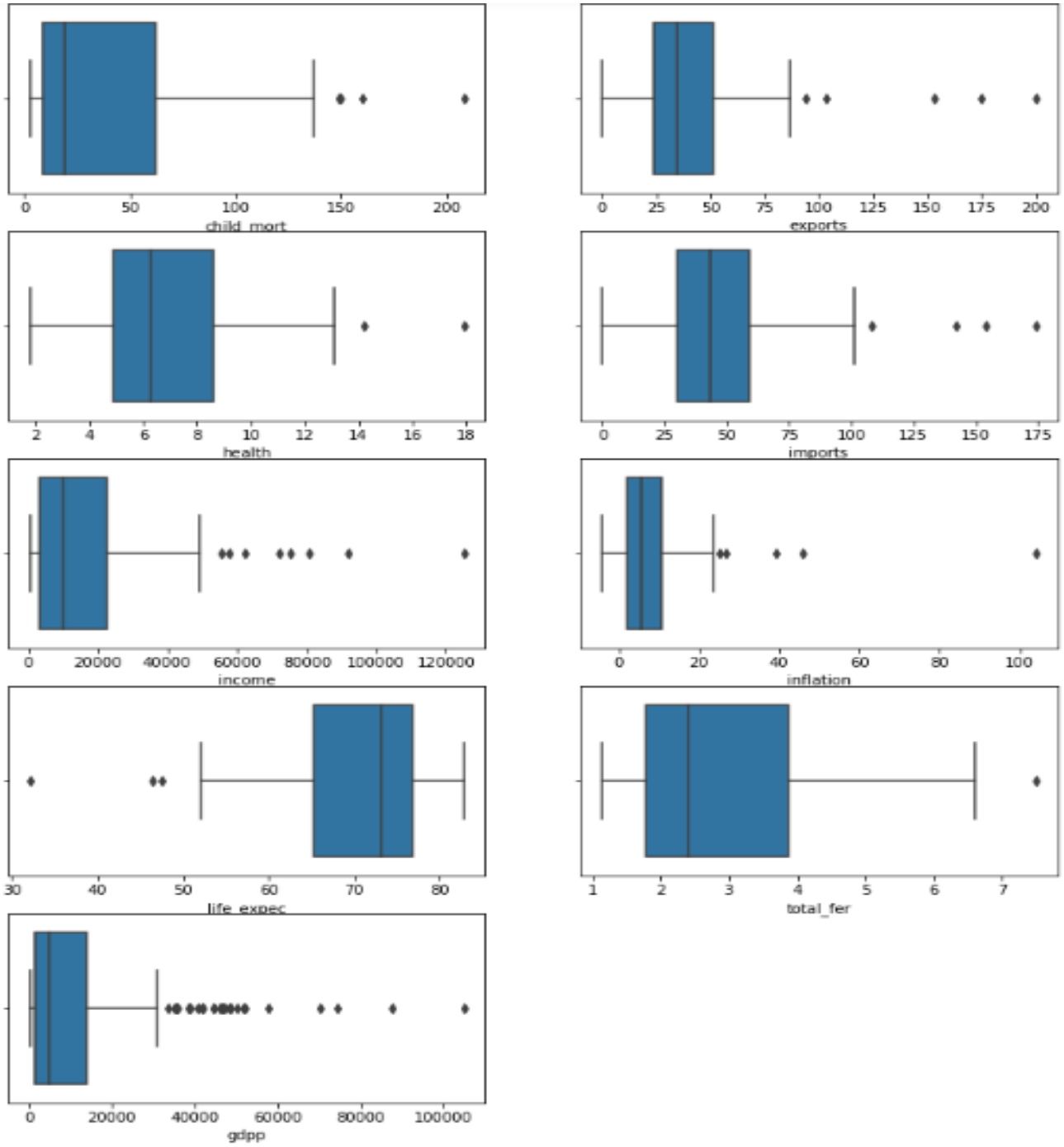
The number of Outliers are considerably huge in the dataset.
In the dataset, each country has single records against itself & dropping any outlier might result in loss of information i.e. records against a country might be eliminated.

1) This might cause discrepancies in final analysis, where all the countries are not considered.

2) Although PCA is prone to Outliers, the effect of dropping Outliers on Variance, before & after PCA is marginal.

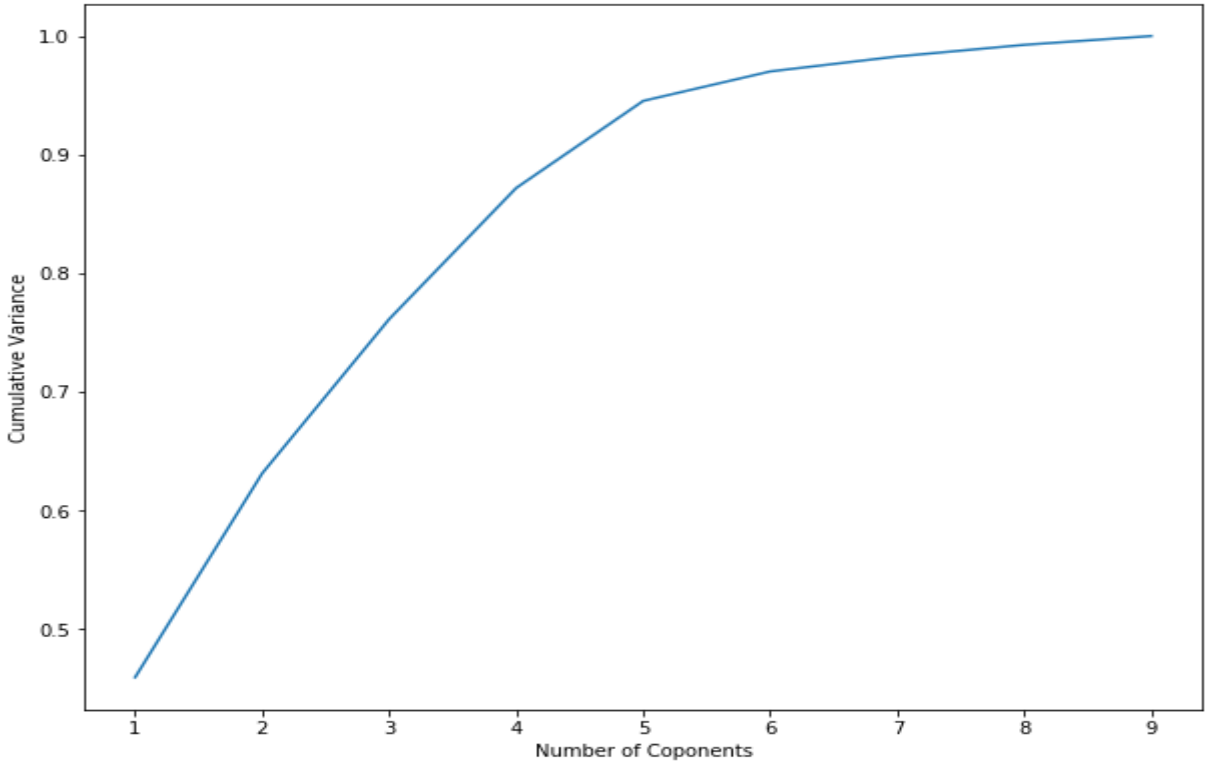
3) Since the outliers are huge, they might create a new cluster for further analysis.

Considering the above 3 points, **Outliers have not been eliminated.**



PCA- Variance & Scree Plot

```
In [12]: from sklearn.decomposition import PCA
         pca = PCA(svd_solver = 'randomized', random_state = 42)
         pca.fit(base_data_df_new2)

Out[12]: PCA(copy=True, iterated_power='auto', n_components=None, random_state=42,
             svd_solver='randomized', tol=0.0, whiten=False)

In [13]: #IDentifying Value of Vectors Post PCA
         pca.components_

Out[13]: array([[-0.41951945,  0.28389698,  0.15083782,  0.16148244,  0.39844111,
                  -0.19317293,  0.42583938, -0.40372896,  0.39264482],
                [ 0.19288394,  0.61316349, -0.24308678,  0.67182064,  0.02253553,
                  -0.00840447, -0.22270674,  0.15523311, -0.0460224 ],
                [-0.02954353,  0.14476069, -0.59663237, -0.29992674,  0.3015475 ,
                   0.64251951,  0.11391854,  0.01954925,  0.12297749],
                [ 0.37065326,  0.00309102,  0.4618975 , -0.07190746,  0.39215904,
                   0.15044176, -0.20379723,  0.37830365,  0.53199457],
                [-0.16896968,  0.05761584,  0.51800037,  0.25537642, -0.2471496 ,
                   0.7148691 ,  0.1082198 , -0.13526221, -0.18016662],
                [ 0.20062815, -0.05933283,  0.00727646, -0.03003154,  0.16034699,
                   0.06628537, -0.60112652, -0.75068875,  0.01677876],
                [-0.07948854, -0.70730269, -0.24983051,  0.59218953,  0.09556237,
                   0.10463252,  0.01848639,  0.02882643,  0.24299776],
                [-0.68274306, -0.01419742,  0.07249683, -0.02894642,  0.35262369,
                  -0.01153775, -0.50466425,  0.29335267, -0.24969636],
                [ 0.3275418 , -0.12308207,  0.11308797,  0.09903717,  0.61298247,
                  -0.02523614,  0.29403981, -0.02633585, -0.62564572]])

In [14]: # Identifying the Variance Ratio of Components Post PCA
         pca.explained_variance_ratio_

Out[14]: array([0.4595174 , 0.17181626, 0.13004259, 0.11053162, 0.07340211,
                0.02484235, 0.0126043 , 0.00981282, 0.00743056])
```



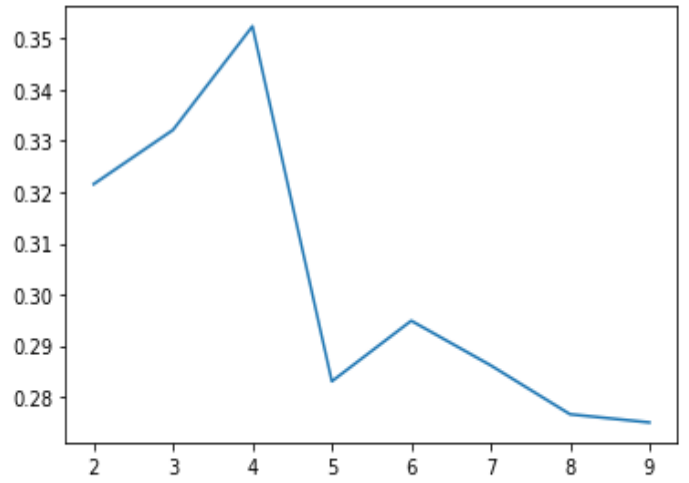
From the Above Variance Ratio values & Scree Plot, The first 4 Components describe the Maximum Variance (about 87%).Hence Considering the 4 Components for PCA.

K-Means Clustering- Number of Clusters

```
In [67]: #Calculating Hopkins Value
         hopkins(pca_df2.drop('ID', axis=1))
         #Hopkins Value is Fluctuating between range of 79 & 87. Hence Clustering can be performed on the mentioned dataset.

Out[67]: 0.8392613641113088
```

Hopkins value is Fluctuating between 79% & 87 % .Hence Clustering can be applied on the above dataset.

```
In [23]: #Calculating Silhouette Score
         from sklearn.metrics import silhouette_score
         ss = []
         for k in range(2,10):
             kmeans = KMeans(n_clusters = k).fit(cluster_df)
             ss.append([k, silhouette_score(cluster_df, kmeans.labels_)])
         plt.plot(pd.DataFrame(ss)[0], pd.DataFrame(ss)[1])

Out[23]: [<matplotlib.lines.Line2D at 0x1bbdfc2b898>]
```



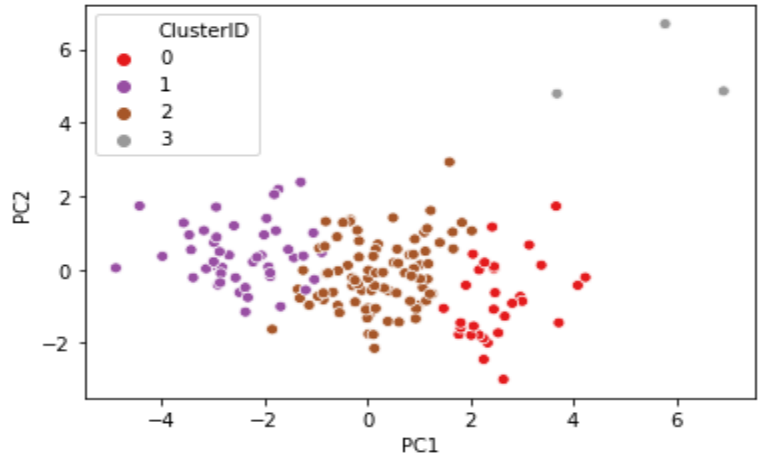
Silhouette Score Drastically Drops Post n=4, hence clustering with n=4

The Silhouette Score increase from n=2 till n=4 & then drastically drops Post n=4.Hence Considering number of clusters=4.

Scatter Plot of Principal Components- K Means Clustering

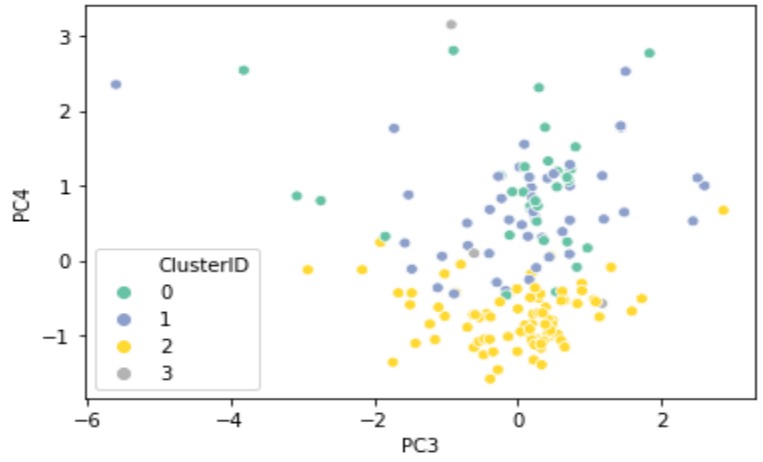
In [27]:
```
# Scatter Plots - PC1 & PC2
sns.scatterplot(x = 'PC1', y = 'PC2', hue = 'ClusterID', data = dat_km, palette='Set1')
```

Out[27]: <matplotlib.axes._subplots.AxesSubplot at 0x1bbe57239e8>

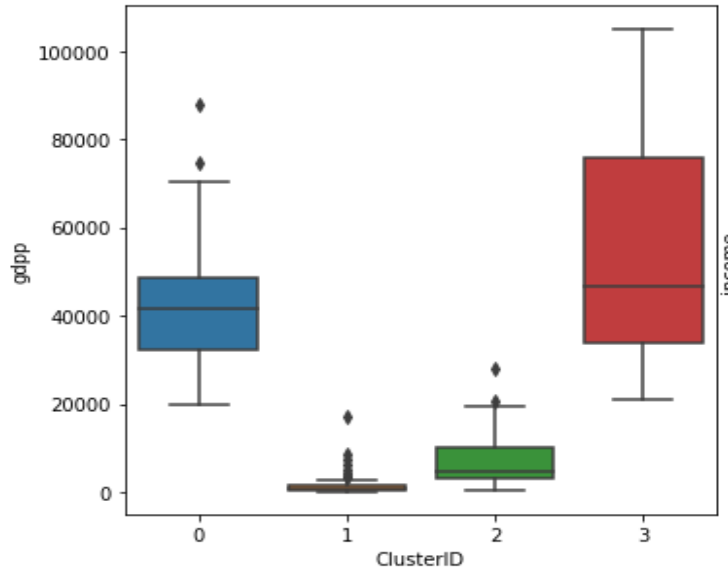


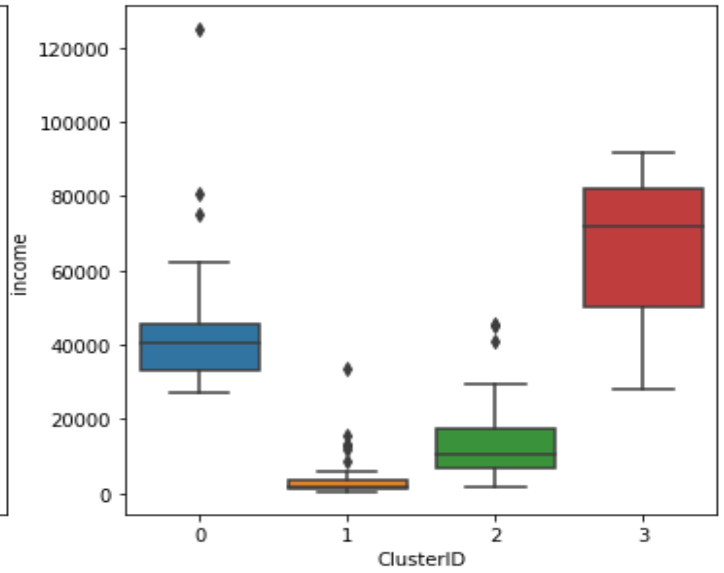
In [28]:
```
# Scatter Plots - PC3 & PC4
sns.scatterplot(x = 'PC3', y = 'PC4', hue = 'ClusterID', data = dat_km, palette='Set2')
```

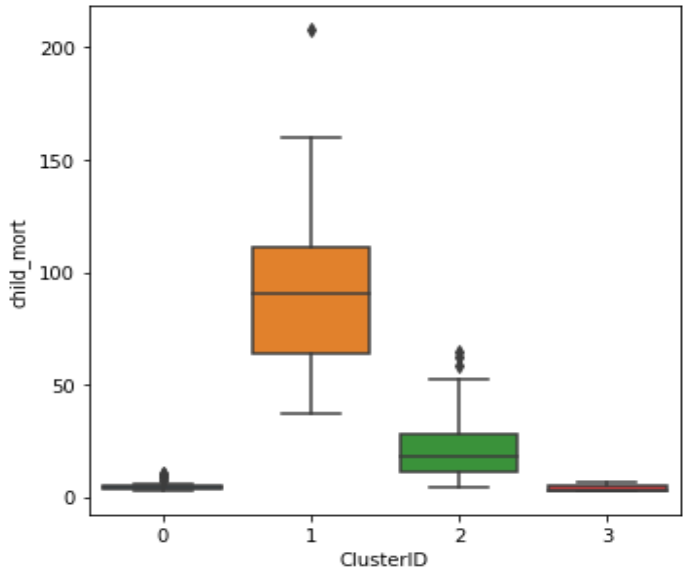
Out[28]: <matplotlib.axes._subplots.AxesSubplot at 0x1bbe57cc400>



K Means Clustering- Outliers Post PCA







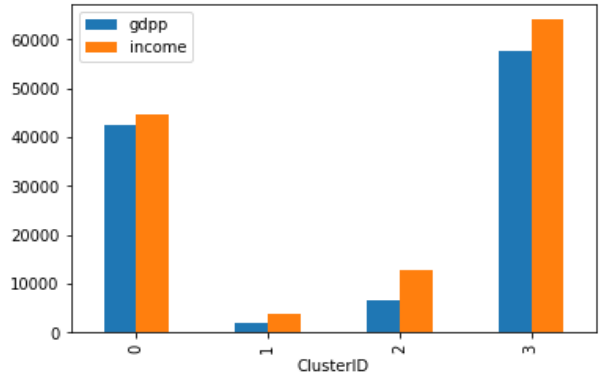
From the Box plots, it is seen that for Cluster 1 Countries Income & GDP is low but Child Mortality is high. Hence these countries require Maximum AID.

It was observed that the Outlier Behaviour Pre & Post PCA are almost identical. Since the Number of Outliers are large, we are considering not to Drop Any Outlier, as they contain Valuable information.

K-Means Clustering- Cluster Analysis

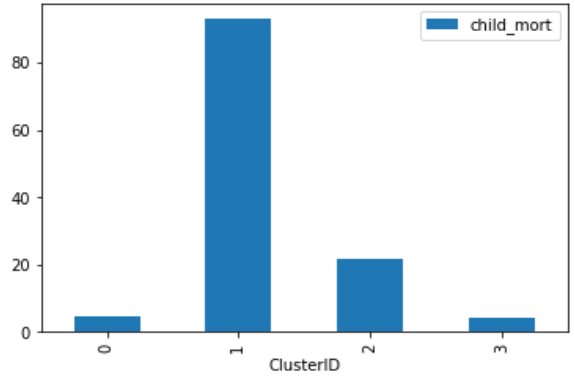
In [34]:
```
# Cluster Analysis for gdpp,income
km_clustered_df[['gdpp', 'income', 'ClusterID']].groupby('ClusterID').mean().plot(kind = 'bar')
```

Out[34]: <matplotlib.axes._subplots.AxesSubplot at 0x1bbe6b6f358>



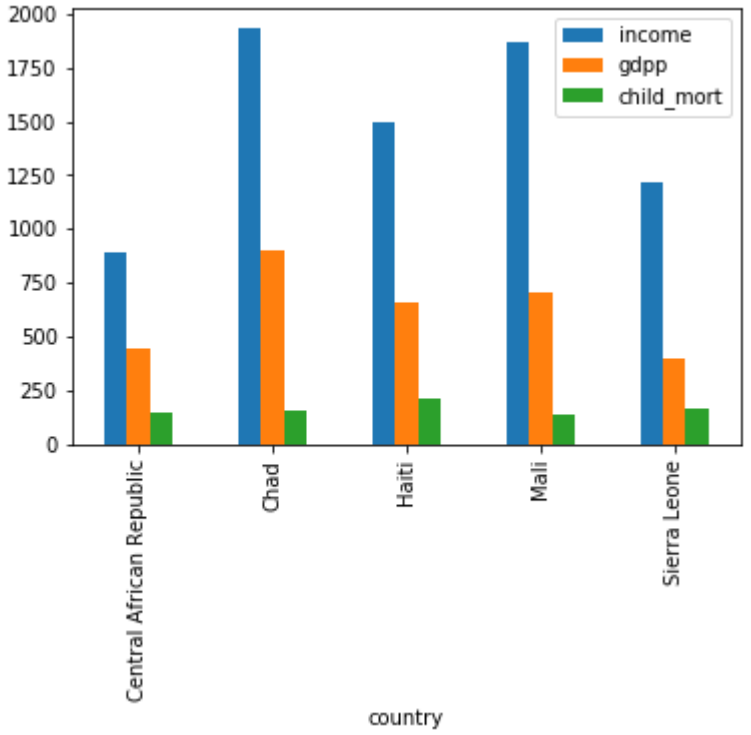
In [35]:
```
# Cluster Analysis for child_mort
km_clustered_df[['child_mort','ClusterID']].groupby('ClusterID').mean().plot(kind = 'bar')
```

Out[35]: <matplotlib.axes._subplots.AxesSubplot at 0x1bbe6d76358>



From the Above Charts, for GDP,Income & Child Mortality, it is seen that Countries in Cluster 1 have Low GDP & Income Rate & High rate of Child Mortality. Hence These are the countries that need to be taken in to consideration for HELP.

K Means Clustering- Identifying Countries



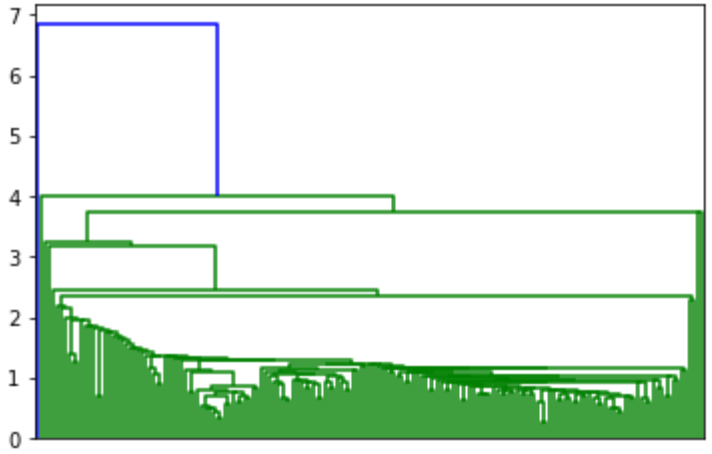
ClusterID	country
0	31
1	47
2	86
3	3

country	child_mort
Haiti	208.0
Sierra Leone	160.0
Chad	150.0
Central African Republic	149.0
Mali	137.0

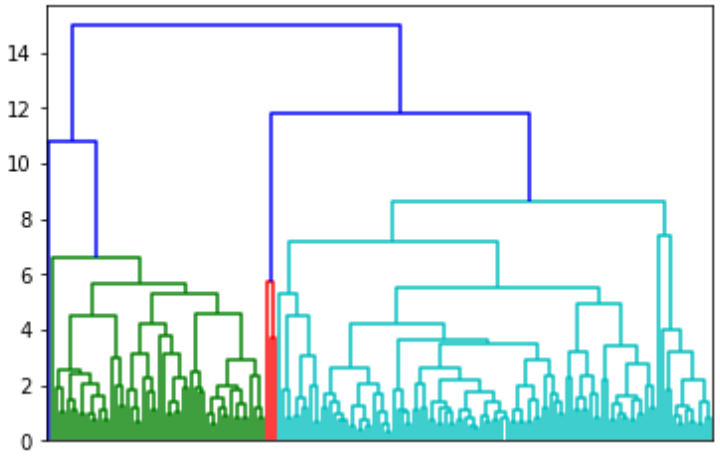
From the Above Analysis, using PCA & Kmeans Clustering, there are 47 countries in Cluster 1 that required AID. By Sorting the countries based on Child Mortality, High to Low, above are the Top 5 Countries that require AID at the earliest

Since Countries in Cluster 1 require Aid on priority, Identifying the countries in Cluster 1.

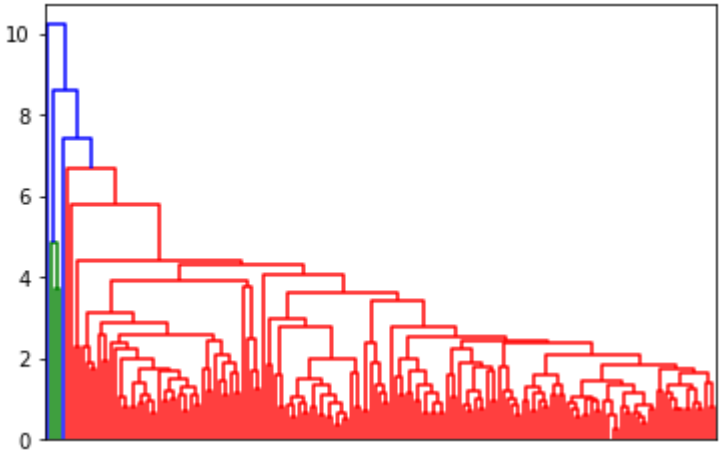
Hierarchical Clustering



Single Linkage



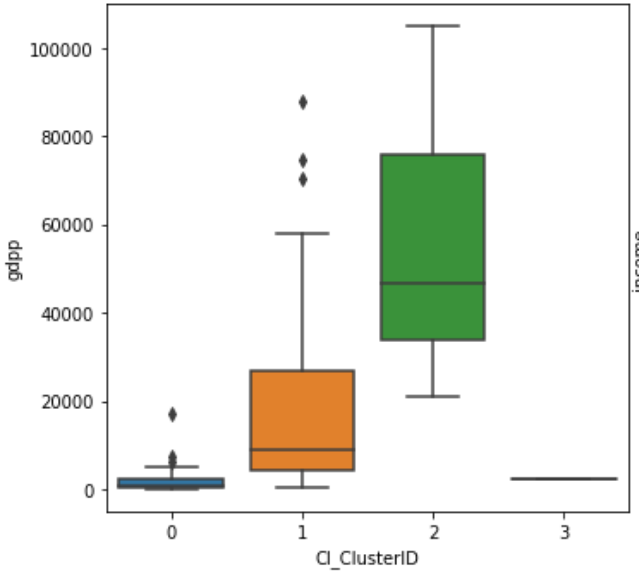
Complete Linkage

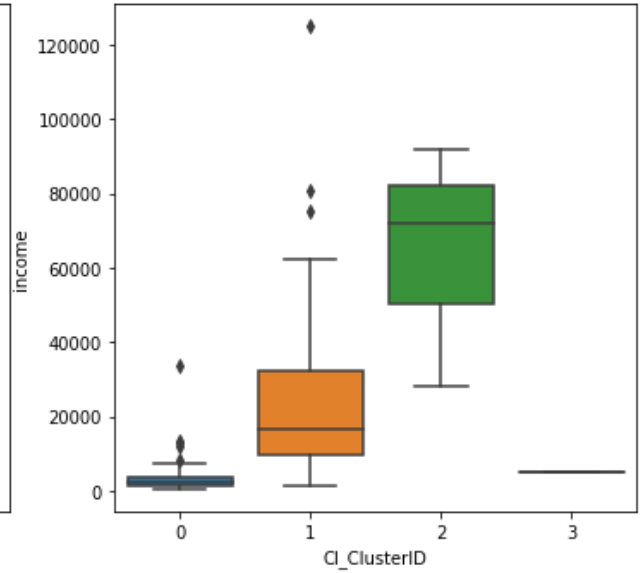


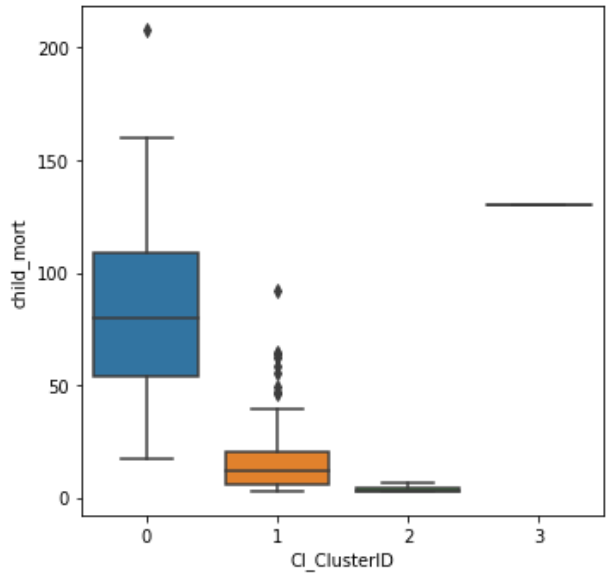
Average Linkage

From the Above Types of Hierarchical Clustering, Considering the Complete Linkage type, we can draw a line @ height=8 and derive 4 clusters. Hence n=4,for hierarchical Clustering.

Hierarchical Clustering- Box Plot Analysis





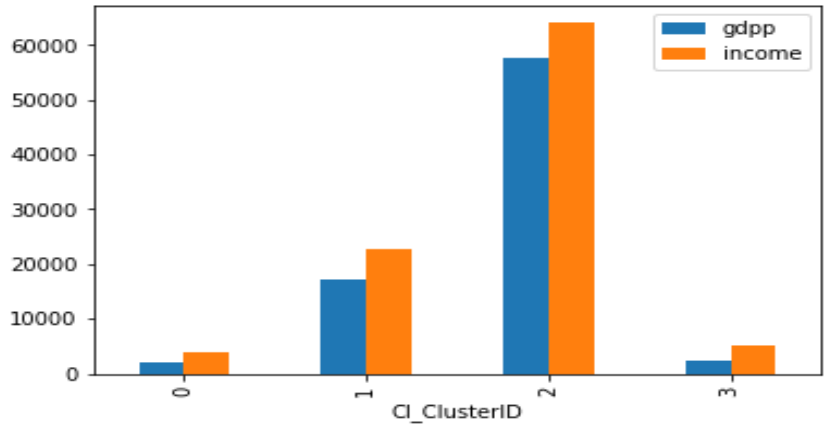


From the Box plots, it is seen that for Cluster 0 Countries Income & GDP is low but Child Mortality is high. Hence these countries require Maximum AID.

Hierarchical Clustering – Cluster Analysis

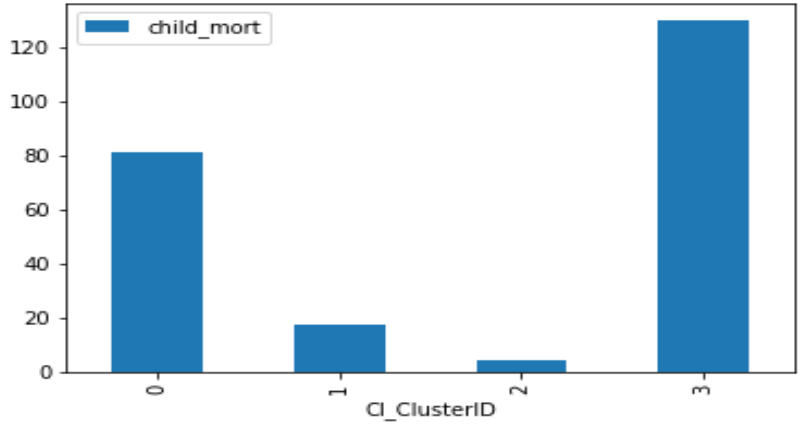
In [44]:
```
# Cluster Analysis for gdpp,income for Hierarchical Clustering
km_clustered_df[['gdpp', 'income', 'Cl_ClusterID']].groupby('Cl_ClusterID').mean().plot(kind = 'bar')
```

Out[44]: <matplotlib.axes._subplots.AxesSubplot at 0x1bbe4a6a278>



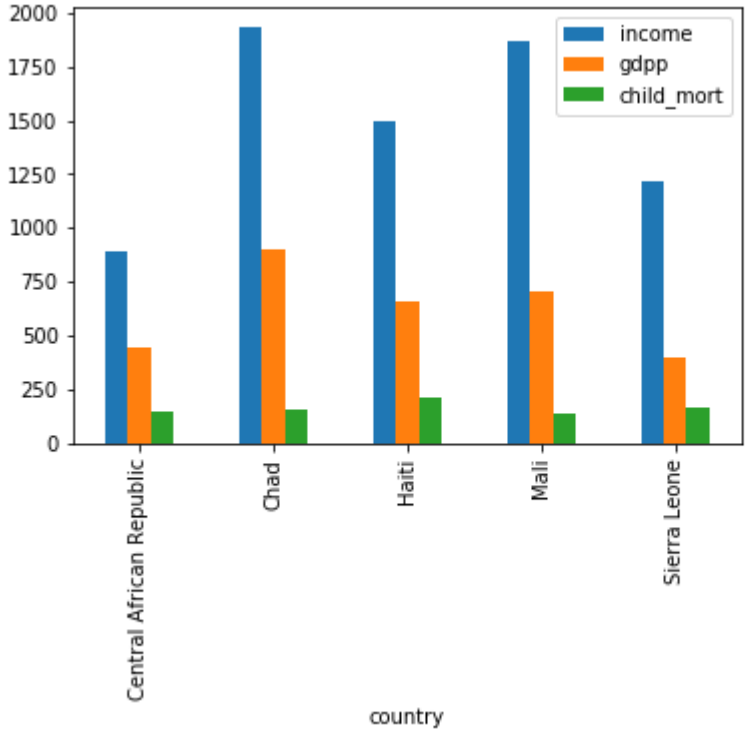
In [45]:
```
# Cluster Analysis for gdpp,income for Hierarchical Clustering
km_clustered_df[['child_mort', 'Cl_ClusterID']].groupby('Cl_ClusterID').mean().plot(kind = 'bar')
```

Out[45]: <matplotlib.axes._subplots.AxesSubplot at 0x1bbe5a918d0>



From the Above Charts, for GDP,Income & Child Mortality, it is seen that Countries in Cluster 0 & Cluster 3 have Low GDP & Income Rate & High rate of Child Mortality. Hence These are the countries that need to be taken in to consideration for HELP.

Hierarchical Clustering- Identifying Countries



country	child_mort
Haiti	208.0
Sierra Leone	160.0
Chad	150.0
Central African Republic	149.0
Mali	137.0

Cl_ClusterID	country
0	54
1	109
2	3
3	1

From the Above Analysis, using PCA & Hierarchical Clustering, there are 54 countries in Cluster 0 that required AID. By Sorting the countries based on Child Mortality, High to Low, above are the Top 5 Countries that require AID at the earliest

Final List of Countries

- Considering Both K-Means Clustering & Hierarchical Clustering, following are the Top 5 Countries that Require urgent Aid.
  - Haiti
  - Sierra Leone
  - Chad
  - Central African Republic
  - Mali