Assignment – Clustering and PCA Part 1

KAVITHA M

Problem Statement

- HELP International is an international humanitarian NGO that is committed to fighting poverty and providing the people of backward countries with basic amenities and relief during the time of disasters and natural calamities. It runs a lot of operational projects from time to time along with advocacy drives to raise awareness as well as for funding purposes.

- After the recent funding programmes, they have been able to raise around $ 10 million. Now the CEO of the NGO needs to decide how to use this money strategically and effectively. The significant issues that come while making this decision are mostly related to choosing the countries that are in the direst need of aid.

Understanding the data

	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

Let's convert imports, exports and health spending from percentage values to actual values of their GDP per capita .Because the percentage values don't give a clear picture of that country. For ex. Afghanistan and Albania have similar imports percentage but their gdpp has a huge gap which doesn't give an accurate idea of which country is more developed than the other.

Post conversion, data would look like,

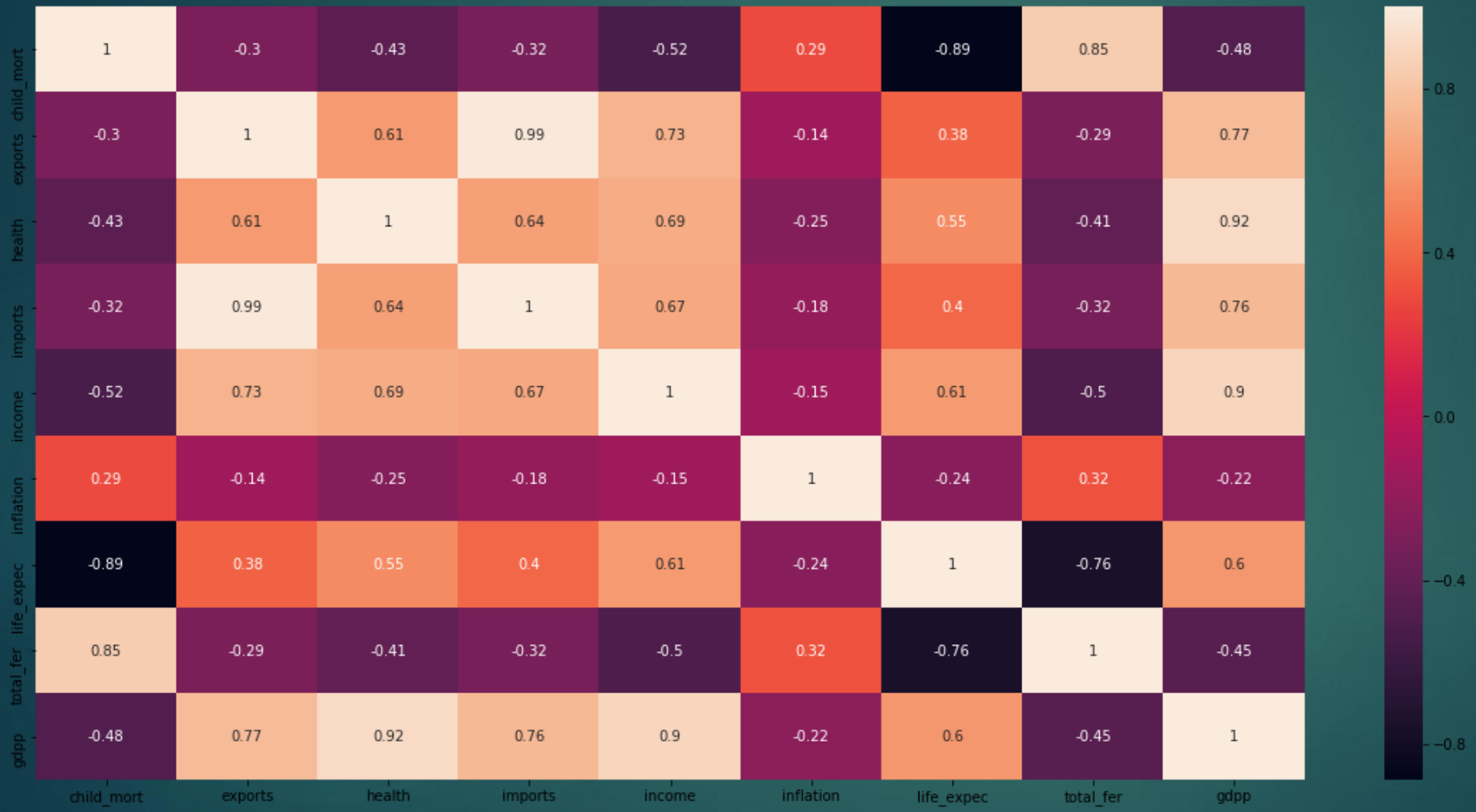
	country	child_mort	exports	health	imports	income	inflation	life_expec	total_fer	gdpp
0	Afghanistan	90.2	55.30	41.9174	248.297	1610	9.44	56.2	5.82	553
1	Albania	16.6	1145.20	267.8950	1987.740	9930	4.49	76.3	1.65	4090
2	Algeria	27.3	1712.64	185.9820	1400.440	12900	16.10	76.5	2.89	4460
3	Angola	119.0	2199.19	100.6050	1514.370	5900	22.40	60.1	6.16	3530
4	Antigua and Barbuda	10.3	5551.00	735.6600	7185.800	19100	1.44	76.8	2.13	12200

```
<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
```

Basic data cleaning checks

```
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
```

Checking for correlation..



A lot of highly correlated variables exist, hence the usage of PCA is justified. Now let's proceed to doing it on the dataset

The final matrix would only contain the data columns. Hence let's drop the country column

	child_mort	exports	health	imports	income	inflation	life_expec	total_fer	gdpp
0	90.2	55.30	41.9174	248.297	1610	9.44	56.2	5.82	553
1	16.6	1145.20	267.8950	1987.740	9930	4.49	76.3	1.65	4090
2	27.3	1712.64	185.9820	1400.440	12900	16.10	76.5	2.89	4460
3	119.0	2199.19	100.6050	1514.370	5900	22.40	60.1	6.16	3530
4	10.3	5551.00	735.6600	7185.800	19100	1.44	76.8	2.13	12200

Rescaling data in order to perform PCA

```
array([[ 1.29153238, -0.4110113 , -0.56503989, ..., -1.61909203,
         1.90288227, -0.67917961],
       [-0.5389489 , -0.35019096, -0.43921769, ...,  0.64786643,
        -0.85997281, -0.48562324],
       [-0.27283273, -0.31852577, -0.48482608, ...,  0.67042323,
        -0.0384044 , -0.46537561],
       ...,
       [-0.37231541, -0.36146329, -0.53848844, ...,  0.28695762,
        -0.66120626, -0.63775406],
       [ 0.44841668, -0.39216643, -0.55059641, ..., -0.34463279,
         1.14094382, -0.63775406],
       [ 1.11495062, -0.38395214, -0.54049845, ..., -2.09278484,
         1.6246091 , -0.62954556]])
```

Applying PCA..

```
#let's apply PCA
pca.fit(dat2)
```

```
PCA(copy=True, iterated_power='auto', n_components=None, random_state=42,
  svd_solver='randomized', tol=0.0, whiten=False)
```

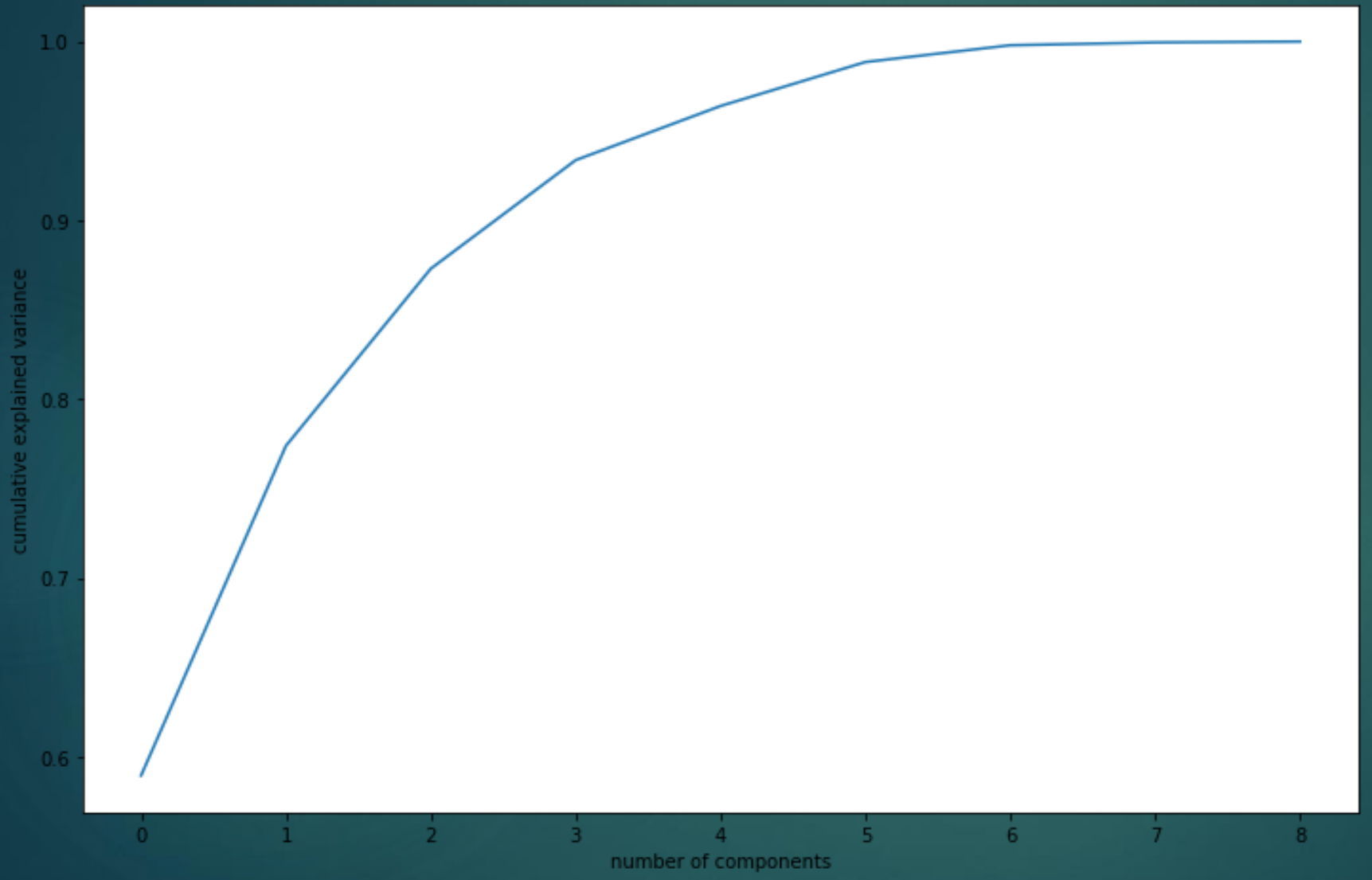
List of PCA components. It would be the same as the number of variables

```
array([[-0.31639186,  0.34288671,  0.358535  ,  0.34486492,  0.38004113,
        -0.14308531,  0.34385651, -0.30284224,  0.39998795],
       [ 0.47626735,  0.39731091,  0.1550529 ,  0.37078075,  0.12838448,
         0.22126089, -0.36981973,  0.4597152 ,  0.2006241 ],
       [-0.15001225, -0.03057367, -0.07570322, -0.07217386,  0.14576421,
         0.94841868,  0.19675173, -0.07783431,  0.01033941],
       [-0.14805195,  0.44942527, -0.59971228,  0.46179779, -0.15480592,
        -0.00762798, -0.01839465, -0.21392805, -0.36477239],
       [ 0.1019948 , -0.03853829, -0.49319984, -0.2527867 ,  0.79407469,
        -0.13642345, -0.15404105, -0.02033568,  0.08750149],
       [ 0.19658519, -0.03891112,  0.18069888, -0.01217988, -0.03814681,
         0.10840284, -0.58600986, -0.75390075,  0.04538167],
       [ 0.76126725, -0.01366973, -0.06461567,  0.02718244, -0.02311312,
        -0.02207663,  0.58120846, -0.27314534, -0.04402264],
       [ 0.00644411, -0.05526371,  0.43007213,  0.1311355 ,  0.3938113 ,
        -0.00607016,  0.002966  ,  0.03429334, -0.79902242],
       [-0.00495137, -0.71792388, -0.13034593,  0.66568664,  0.07901102,
         0.01128137, -0.03159406,  0.02368185,  0.12846398]])
```

checking the variance ratios

```
array([5.89372984e-01, 1.84451685e-01, 9.91147170e-02, 6.07227801e-02,
       3.02917253e-02, 2.45982702e-02, 9.39743701e-03, 1.55641971e-03,
       4.93981394e-04])
```

Cumulative Variance VS Number of components

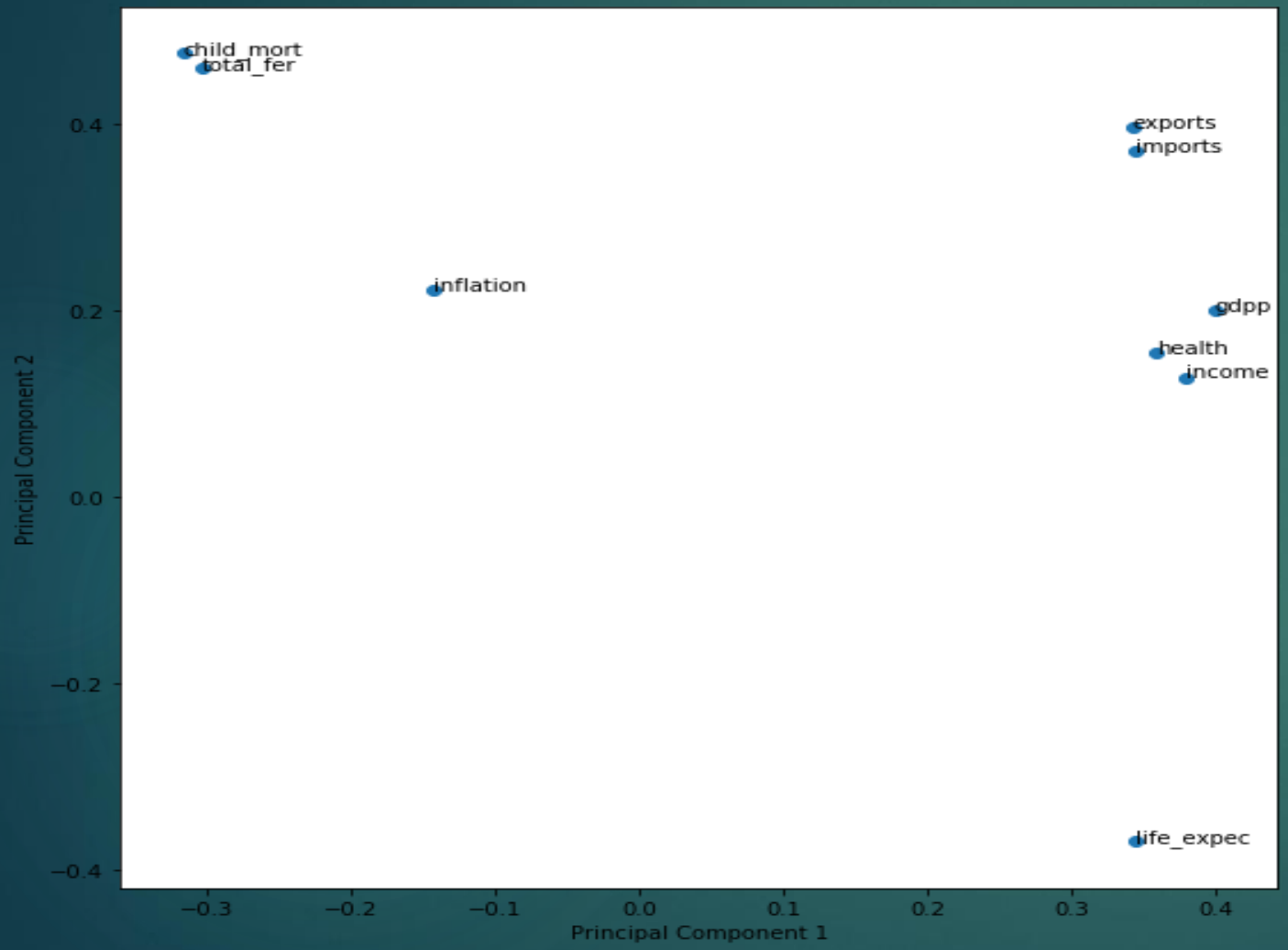


Clearly over 90% of the data is properly explained by the first 3 principal components. Let's use them only for our clustering process

Checking the 1st 3 components..

	Feature	PC1	PC2	PC3
0	child_mort	-0.316392	0.476267	-0.150012
1	exports	0.342887	0.397311	-0.030574
2	health	0.358535	0.155053	-0.075703
3	imports	0.344865	0.370781	-0.072174
4	income	0.380041	0.128384	0.145764
5	inflation	-0.143085	0.221261	0.948419
6	life_expec	0.343857	-0.369820	0.196752
7	total_fer	-0.302842	0.459715	-0.077834
8	gdpp	0.399988	0.200624	0.010339

Plotting the 1st 3 components..

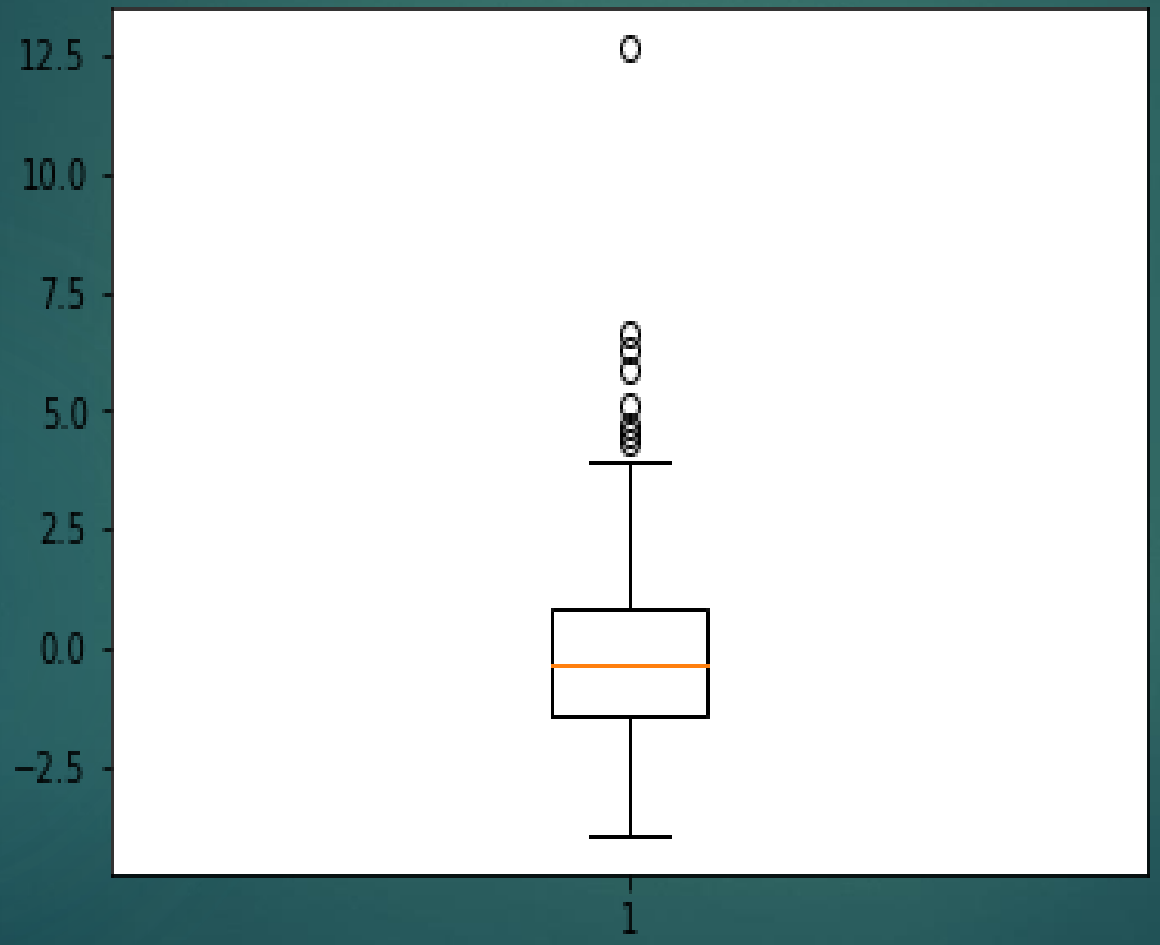


A lot of variables have a good loading score on the first principal component. Similarly Child mortality and total fertility is well explained by the 2nd principal component.

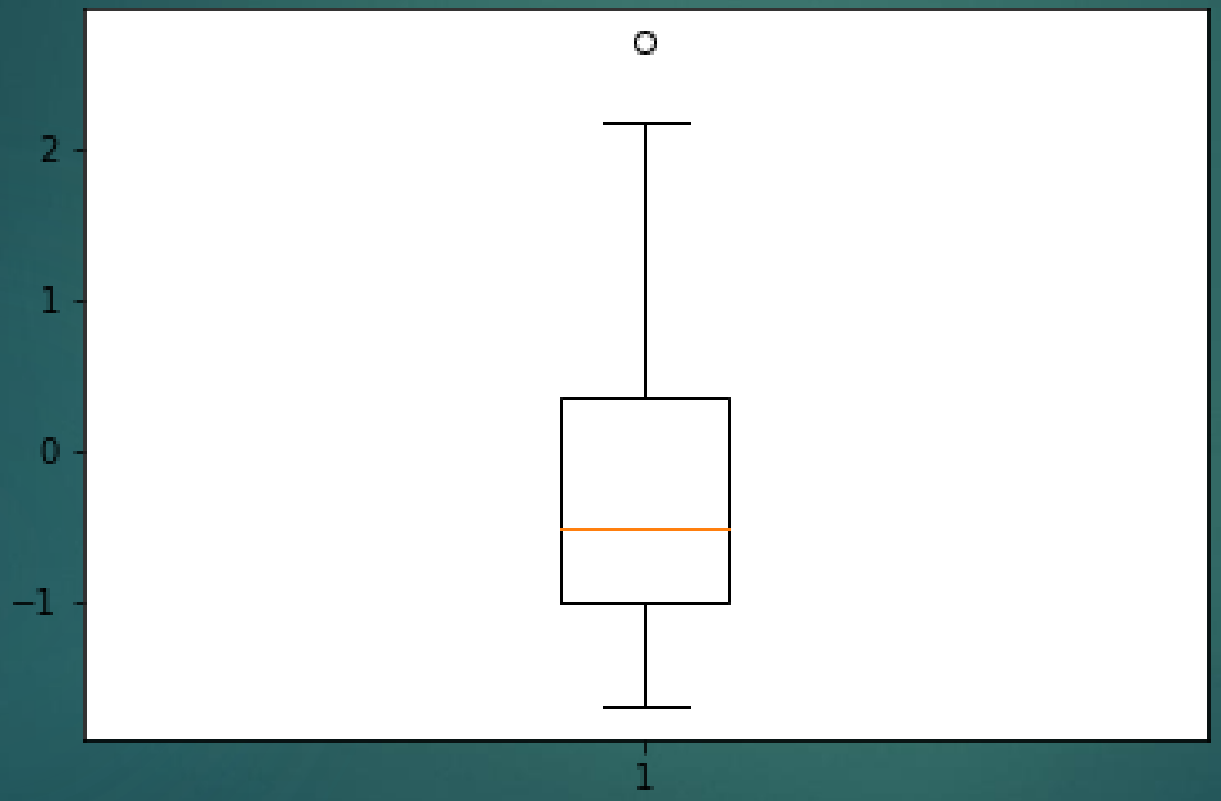
Creating the newer matrix according to the given principal components

	country	PC1	PC2	PC3
0	Afghanistan	-2.637442	1.469038	-0.541359
1	Albania	-0.022277	-1.431896	-0.020701
2	Algeria	-0.457626	-0.673301	0.961867
3	Angola	-2.724520	2.174583	0.606687
4	Antigua and Barbuda	0.649849	-1.024374	-0.250103

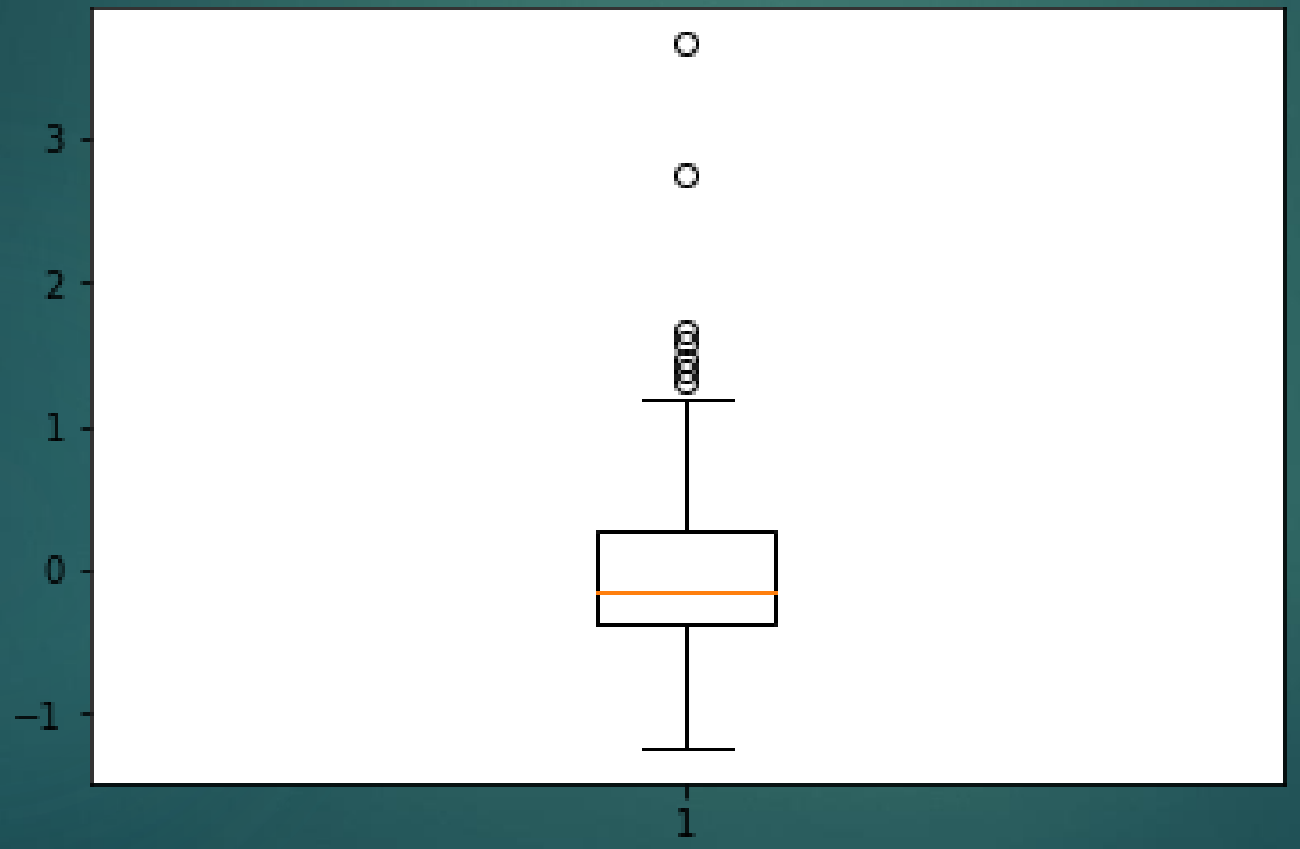
Checking for outliers.. – Principal Component 1



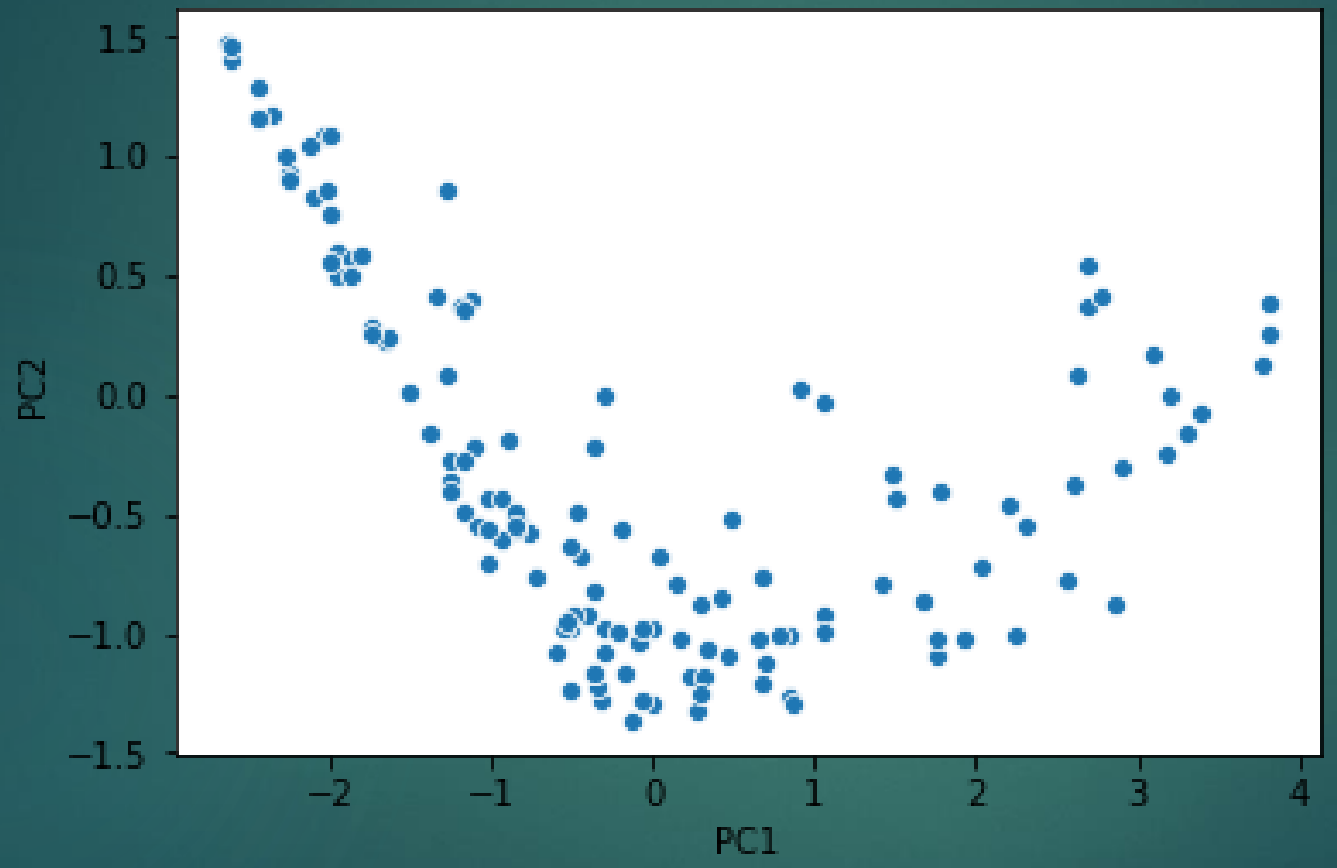
Checking for outliers.. – Principal Component 2



Checking for outliers.. – Principal Component 3



Checking the spread of dataset



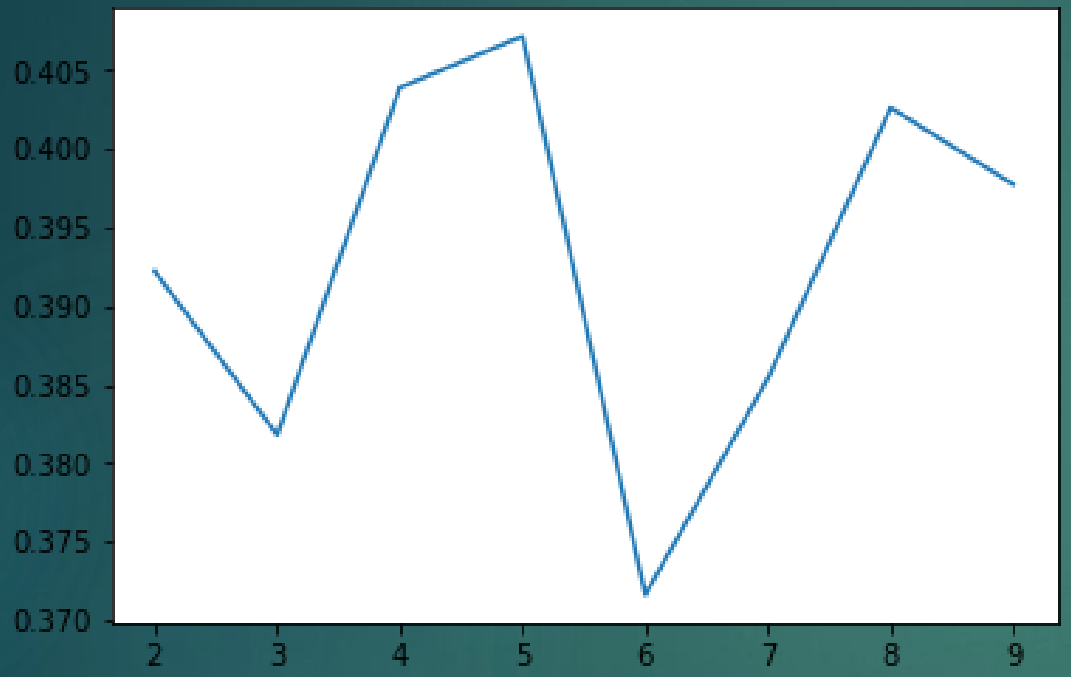
Stepping into clustering..

- As we checked previously the dataset looks of similar magnitude. Hence no further standardization is necessary. Let's proceed to calculating the Hopkins statistic to ensure that the data is good for clustering.

```
#Let's check the Hopkins measure
hopkins(pcs_df2.drop(['country'],axis=1))
```

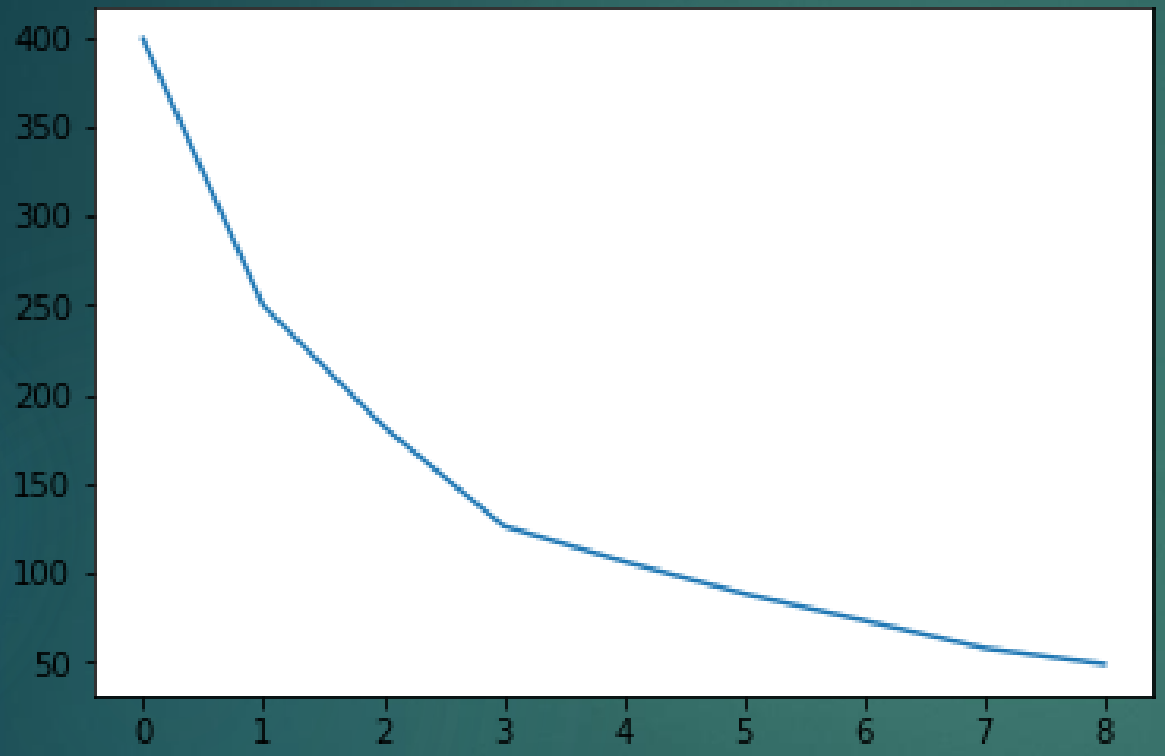
```
0.8286149182127605
```

Implementing K-means Clustering



The silhouette score reaches a peak at around 5 clusters indicating that it might be the ideal number of clusters.

Using elbow curve to identify cluster number...



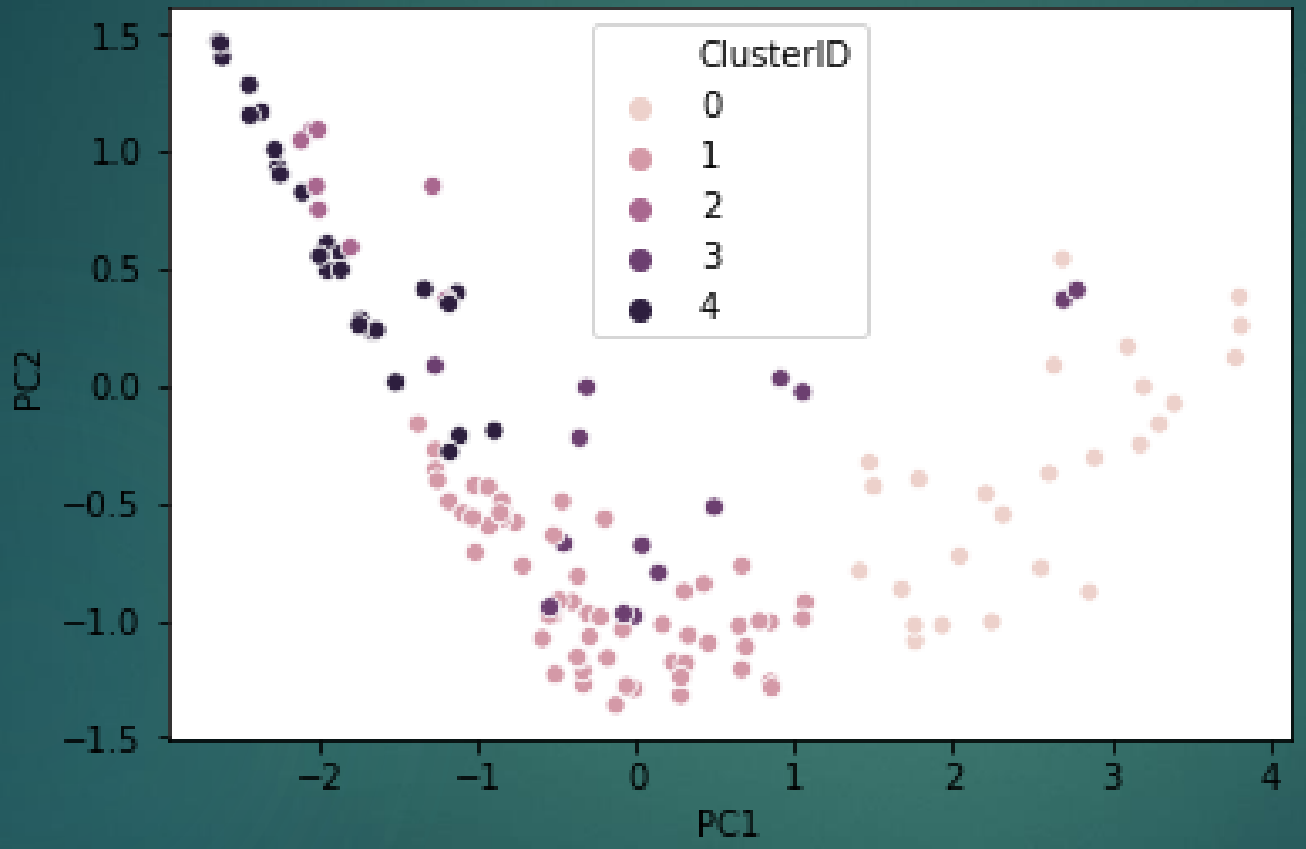
A distinct elbow is formed at around 3-7 clusters. Let's finally create the clusters and see for ourselves which ones fare better

Implementing K-means with k=5 clusters

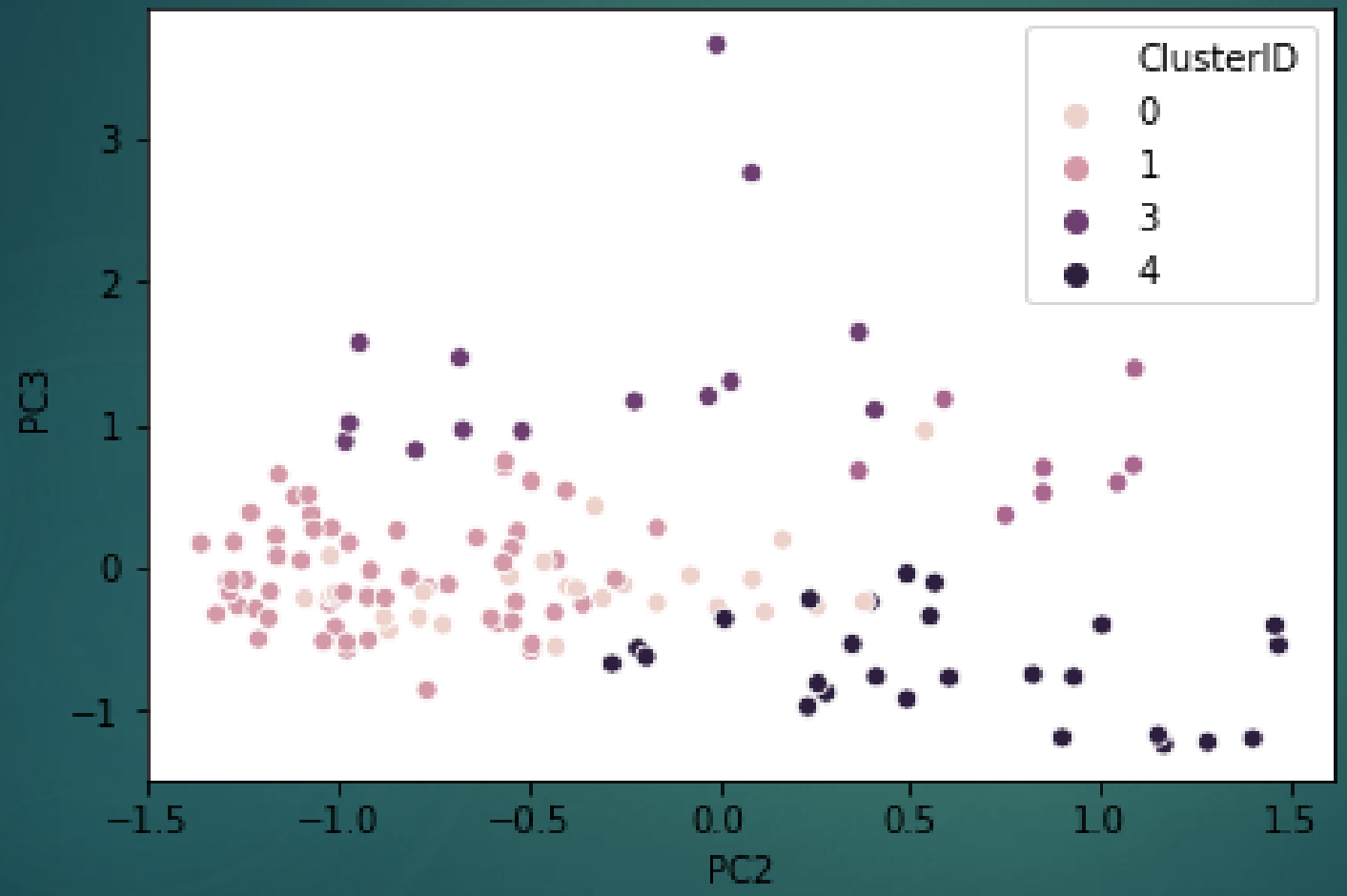
Implementing k-means..

	country	PC1	PC2	PC3	ClusterID
0	Afghanistan	-2.637442	1.469038	-0.541359	4
1	Algeria	-0.457626	-0.673301	0.961867	3
2	Antigua and Barbuda	0.649849	-1.024374	-0.250103	1
3	Argentina	0.037197	-0.680889	1.466963	3
4	Armenia	-0.332692	-1.274517	0.176636	1

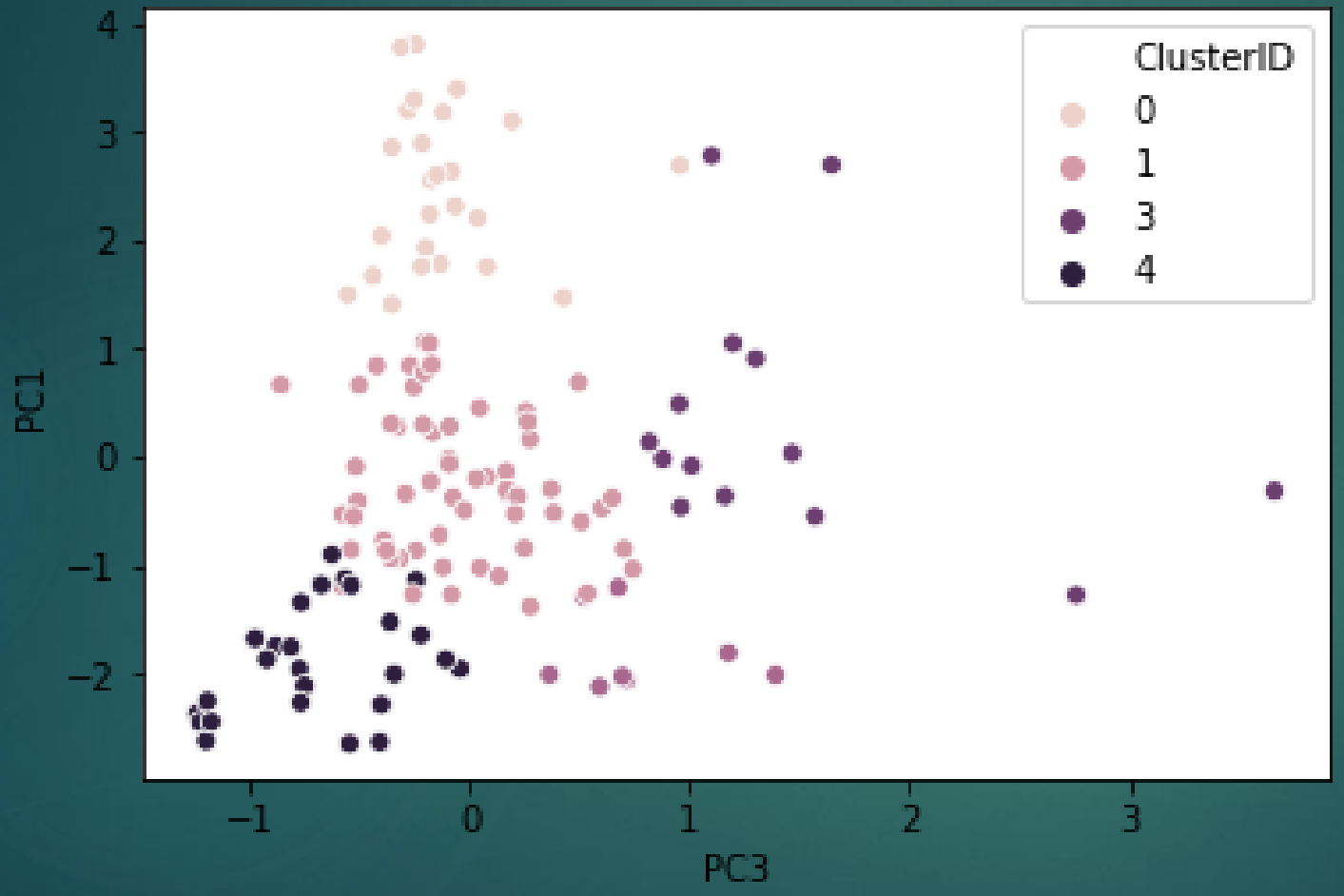
Visualizing clusters on PC-1



Visualizing clusters on PC-2



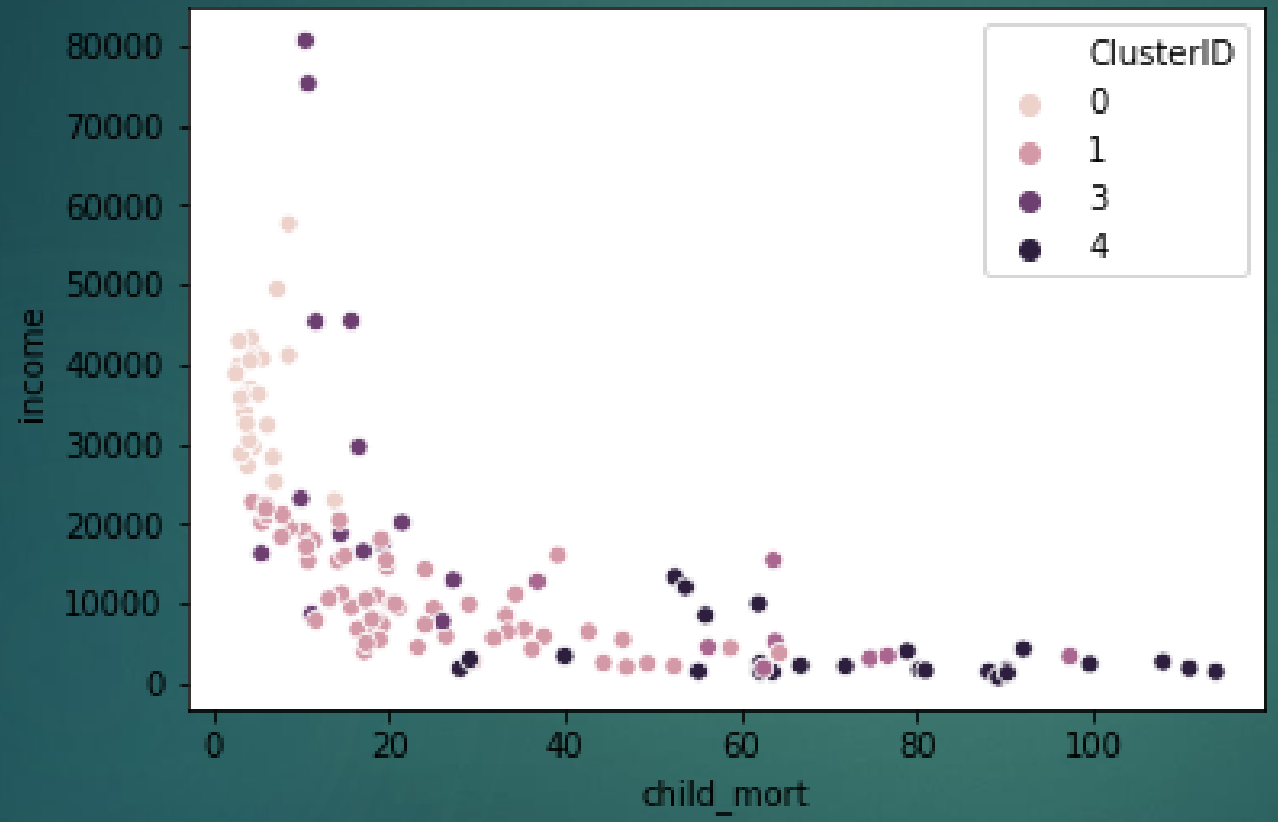
Visualizing clusters on PC-3



Even though some distinct clusters are being formed, some are not so good. Creating the cluster means wrt to the various variables mentioned in the question and plot and see how they are related

ClusterID	ClusterID	Child_Mortality	Exports	Imports	Health_Spending	Income	Inflation	Life_Expectancy	Total_Fertility	GDPpcapita
0	0.0	5.111538	14074.200000	13725.319231	3335.156154	35707.692308	1.858577	79.838462	1.731538	34550.000000
1	1.0	23.535593	2612.041839	2876.256629	376.153092	10690.508475	5.770017	72.737288	2.322034	5781.050847
2	2.0	66.525000	1364.012500	966.090000	170.390000	6171.250000	19.887500	65.725000	4.827500	3111.250000
3	3.0	15.557143	7198.816429	4200.612143	528.118714	29805.000000	20.321429	73.814286	2.307857	13610.714286
4	NaN	72.657692	639.146415	857.742692	126.934292	3425.000000	5.267500	60.619231	4.371538	1740.615385

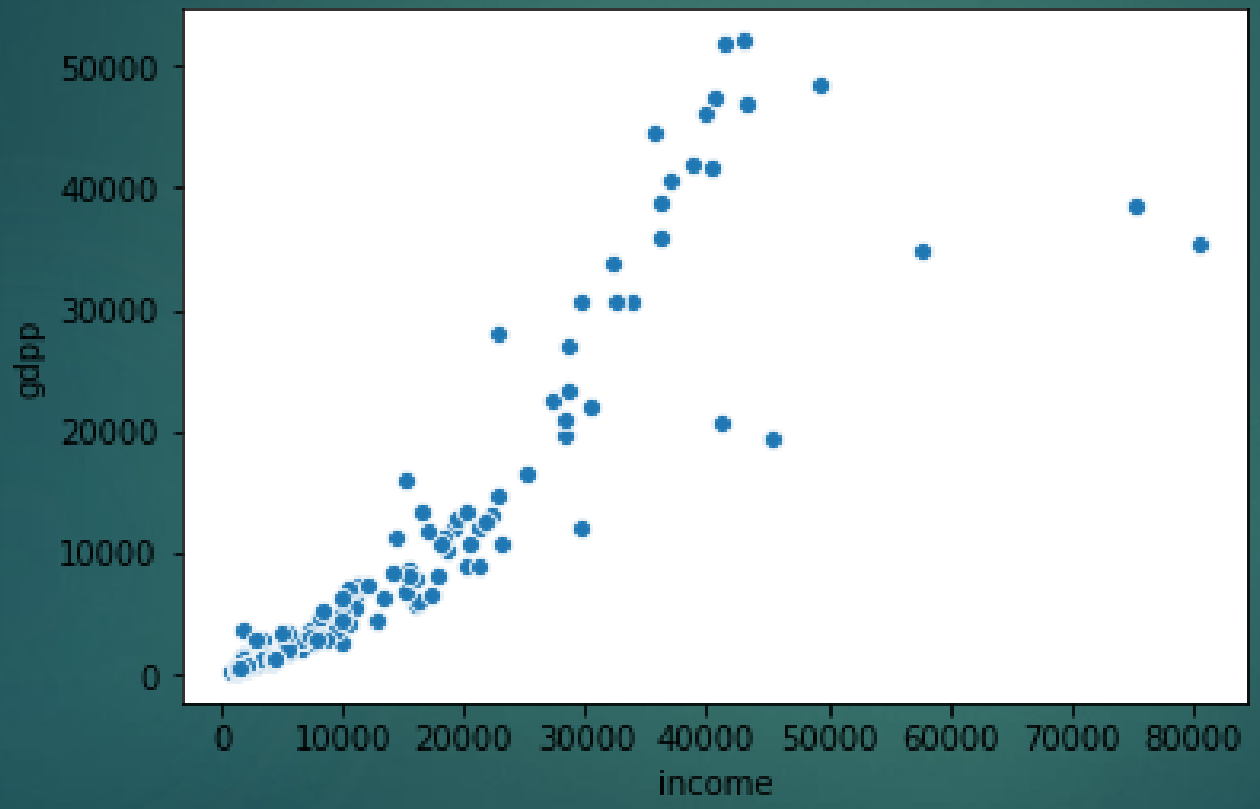
Along child-mortality and income



Let's take a look at those countries clusters and try to make sense if the clustering process worked well

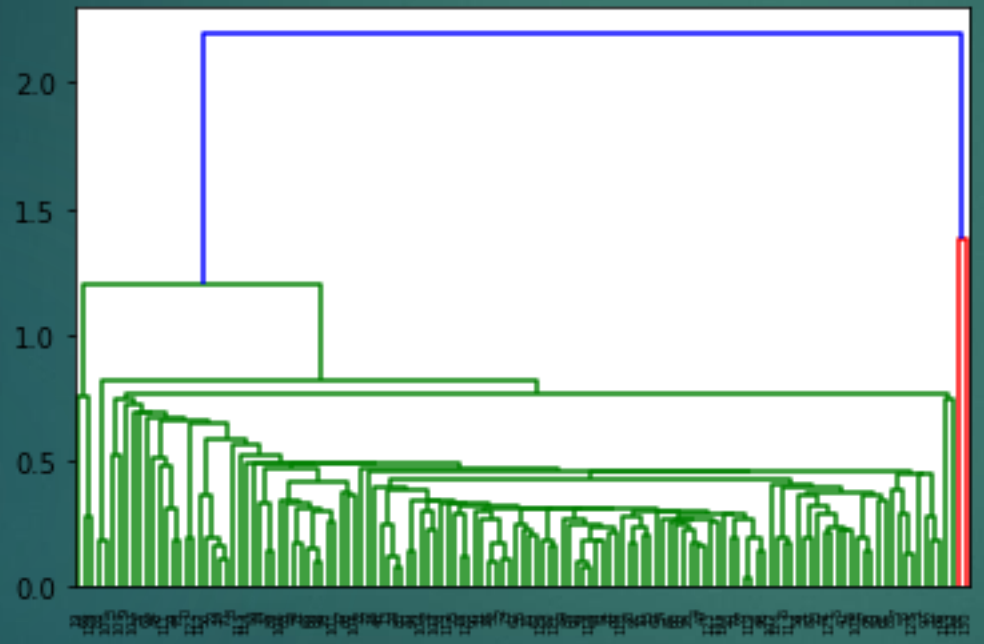
	country	child_mort	exports	imports	health	income	inflation	life_expec	total_fer	gdpp	ClusterID
5	Australia	4.8	10276.2	10847.1	4530.87	41400	1.160	82.0	1.93	51900	0
6	Austria	4.3	24059.7	22418.2	5159.00	43200	0.873	80.5	1.44	46900	0
8	Bahamas	13.8	9800.0	12236.0	2209.20	22900	-0.393	73.8	1.86	28000	0
9	Bahrain	8.6	14386.5	10536.3	1028.79	41100	7.440	76.0	2.16	20700	0
23	Canada	5.6	13793.4	14694.0	5356.20	40700	2.870	81.3	1.63	47400	0
31	Cyprus	3.6	15461.6	17710.0	1838.76	33900	2.010	79.9	1.42	30800	0
32	Czech Republic	3.4	13068.0	12454.2	1560.24	28300	-1.430	77.5	1.51	19800	0
40	Finland	3.0	17879.4	17278.8	4134.90	39800	0.351	80.0	1.87	46200	0
41	France	4.2	10880.8	11408.6	4831.40	36900	1.050	81.4	2.03	40600	0
45	Germany	4.2	17681.4	15507.8	4848.80	40400	0.758	80.1	1.39	41800	0
47	Greece	3.9	5944.9	8258.3	2770.70	28700	0.673	80.4	1.48	26900	0
53	Iceland	2.6	22374.6	18142.7	3938.60	38800	5.470	82.0	2.20	41900	0
58	Israel	4.6	10710.0	10067.4	2334.78	29600	1.770	81.4	3.03	30600	0
59	Italy	4.0	9021.6	9737.6	3411.74	36200	0.319	81.7	1.46	35800	0
61	Japan	3.2	6675.0	6052.0	4223.05	35800	-1.900	82.8	1.39	44500	0
77	Malta	6.8	32283.0	32494.0	1825.15	28300	3.830	80.3	1.36	21100	0
87	New Zealand	6.2	10211.1	9436.0	3403.70	32300	3.730	80.9	2.17	33700	0
95	Portugal	3.9	6727.5	8415.0	2475.00	27200	0.643	79.8	1.39	22500	0
103	Slovak Republic	7.0	12665.8	12914.8	1459.14	25200	0.485	75.5	1.43	16600	0
104	Slovenia	3.2	15046.2	14718.6	2201.94	28700	-0.987	79.5	1.57	23400	0
107	South Korea	4.1	10917.4	10210.2	1531.53	30400	3.160	80.1	1.23	22100	0

From the clusters it is observed that cluster 2 and 4 have pretty low values of the 4 indicators that we chose. Hence these are the countries that we need to focus



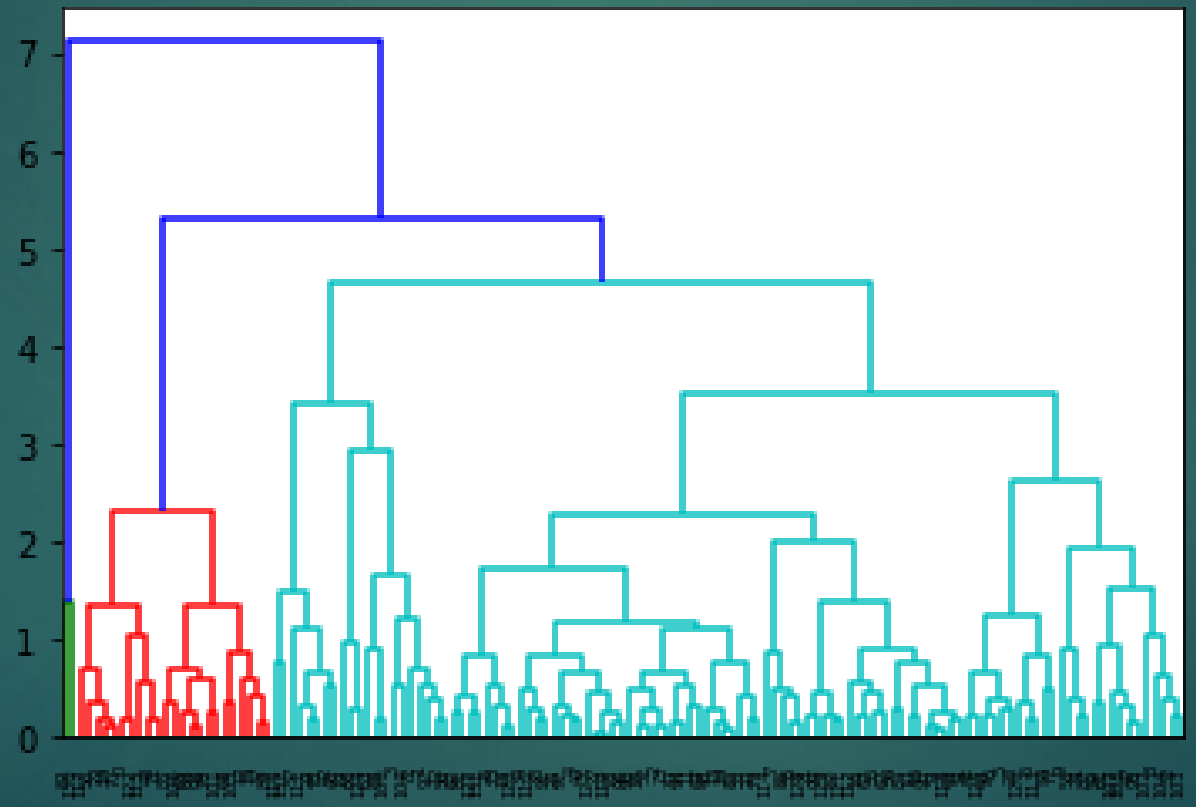
Trying Hierarchical clustering

- Trying single linkage procedure



Trying Hierarchical clustering

- Trying complete linkage procedure instead..



Now we are seeing some good clusters here. Let's see if they make sense if we eliminate the barriers

	country	PC1	PC2	PC3	ClusterID
0	Afghanistan	-2.637442	1.469038	-0.541359	0
1	Algeria	-0.457626	-0.673301	0.961867	1
2	Antigua and Barbuda	0.649849	-1.024374	-0.250103	2
3	Argentina	0.037197	-0.680889	1.466963	1
4	Armenia	-0.332692	-1.274517	0.176636	2

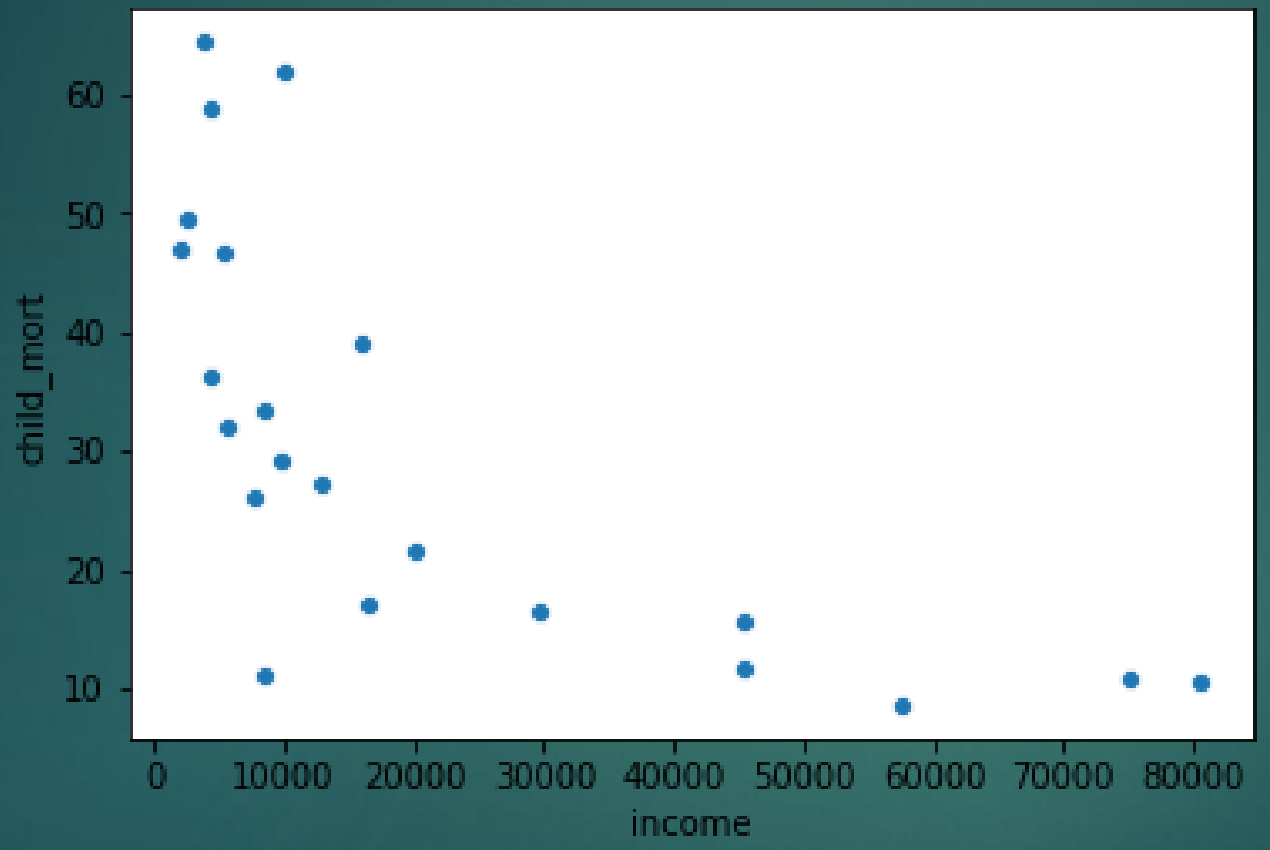
	country	child_mort	exports	imports	health	income	inflation	life_expec	total_fer	gdpp	ClusterID
0	Afghanistan	90.2	55.30	248.297	41.9174	1610	9.44	56.2	5.82	553	0
1	Algeria	27.3	1712.64	1400.440	185.9820	12900	16.10	76.5	2.89	4460	1
2	Antigua and Barbuda	10.3	5551.00	7185.800	735.6600	19100	1.44	76.8	2.13	12200	2
3	Argentina	14.5	1946.70	1648.000	834.3000	18700	20.90	75.8	2.37	10300	1
4	Armenia	18.1	669.76	1458.660	141.6800	6700	7.77	73.3	1.69	3220	2

Checking for data point count in each cluster formed..

```
2     61
3     26
0     23
1     21
4      2
Name: ClusterID, dtype: int64
```

Cluster 4 doesn't have enough amount of clusters.

Visualizing the clusters formed..



Final result

We use the clusters formed during K-means clustering to find the countries that we require since Hierarchical clustering is not showing proper clusters here. For K-means part, we got Cluster 2 and 4 might be the ones which has a proper need of aid

Converting exports, imports and health spending percentages to absolute values.

	country	child_mort	exports	health	imports	income	inflation	life_expec	total_fer	gdpp
0	Afghanistan	90.2	55.30	41.9174	248.297	1610	9.44	56.2	5.82	553
1	Albania	16.6	1145.20	267.8950	1987.740	9930	4.49	76.3	1.65	4090
2	Algeria	27.3	1712.64	185.9820	1400.440	12900	16.10	76.5	2.89	4460
3	Angola	119.0	2199.19	100.6050	1514.370	5900	22.40	60.1	6.16	3530
4	Antigua and Barbuda	10.3	5551.00	735.6600	7185.800	19100	1.44	76.8	2.13	12200

Let's use the binning with gdpp first to see the list of countries which might be important.
The upper limit that we got from the clustering process was 1700.
Let's filter the complete dataset with 1700 as the cut-off limit for gdpp.

After Binning..

	country	child_mort	exports	health	imports	income	inflation	life_expec	total_fer	gdpp
0	Afghanistan	90.2	55.3000	41.9174	248.297	1610	9.440	56.2	5.82	553
12	Bangladesh	49.4	121.2800	26.6816	165.244	2440	7.140	70.4	2.33	758
17	Benin	111.0	180.4040	31.0780	281.976	1820	0.885	61.8	5.36	758
25	Burkina Faso	116.0	110.4000	38.7550	170.200	1430	6.810	57.9	5.87	575
26	Burundi	93.6	20.6052	26.7960	90.552	764	12.300	57.7	6.26	231

```
len(fin2)
```

```
48
```

#So we got 48 countries here. We can create further sub categories by taking another good clustering indicator.

	child_mort	exports	health	imports	income	inflation	life_expec	total_fer	gdpp
count	48.000000	48.000000	48.000000	48.000000	48.000000	48.000000	48.000000	48.000000	48.000000
mean	84.808333	242.988282	53.166544	389.688794	2209.229167	8.849688	60.789583	4.552500	847.583333
std	37.864382	208.411190	36.338142	306.718665	1134.428833	5.849055	7.282776	1.382764	384.444824
min	17.200000	1.076920	12.821200	0.651092	609.000000	0.885000	32.100000	1.270000	231.000000
25%	61.350000	101.630250	31.079500	175.909500	1390.000000	4.080000	57.175000	3.465000	551.500000
50%	82.050000	150.912000	44.388600	280.956000	1900.000000	8.215000	61.250000	4.875000	758.000000
75%	108.250000	388.087500	60.501250	450.765000	2857.500000	12.150000	66.125000	5.370000	1205.000000
max	208.000000	943.200000	190.710000	1279.550000	4490.000000	23.600000	73.100000	7.490000	1630.000000

From the clustering process we got child_mortality to be at least 76 for the most downtrodden cluster.

Let's see how many countries lie within that range

```
len(fin2[fin2['child_mort']>=76])
```

```
28
```

Ok so we got 28 countries now. We can stop here or take one more indicator and find the final list.
#Here we are taking income as the next one, where around 3200 was the income mean of the downtrodden cluster.

```
fin3=fin2[fin2['child_mort']>=76]
fin4=fin3[fin3['income']<3200]
len(fin4)
```

```
23
```

We've got 23 countries now, let's use the describe function to see how they're aligned again.

	child_mort	exports	health	imports	income	inflation	life_expec	total_fer	gdpp
count	23.000000	23.000000	23.000000	23.000000	23.000000	23.000000	23.000000	23.000000	23.000000
mean	113.743478	165.144704	41.870378	292.629174	1444.913043	7.125435	56.065217	5.463478	627.173913
std	30.255681	140.898936	23.154120	222.532254	587.515796	5.428622	6.853042	0.953232	288.989407
min	80.300000	20.605200	17.750800	90.552000	609.000000	0.885000	32.100000	3.300000	231.000000
25%	90.400000	79.379500	30.663050	170.185000	974.000000	3.420000	55.300000	5.080000	432.500000
50%	109.000000	126.885000	37.332000	248.297000	1410.000000	5.450000	57.300000	5.340000	562.000000
75%	119.500000	188.290000	46.119600	328.251000	1740.000000	10.020000	58.750000	6.010000	733.000000
max	208.000000	617.320000	129.870000	1181.700000	2690.000000	20.800000	65.900000	7.490000	1310.000000

The final list of countries

	country	child_mort	exports	health	imports	income	inflation	life_expec	total_fer	gdpp
0	Afghanistan	90.2	55.3000	41.9174	248.297	1610	9.440	56.2	5.82	553
17	Benin	111.0	180.4040	31.0780	281.976	1820	0.885	61.8	5.36	758
25	Burkina Faso	116.0	110.4000	38.7550	170.200	1430	6.810	57.9	5.87	575
26	Burundi	93.6	20.6052	26.7960	90.552	764	12.300	57.7	6.26	231
28	Cameroon	108.0	290.8200	67.2030	353.700	2660	1.910	57.3	5.11	1310
31	Central African Republic	149.0	52.6280	17.7508	118.190	888	2.010	47.5	5.21	446
32	Chad	150.0	330.0960	40.6341	390.195	1930	6.390	56.5	6.59	897
36	Comoros	88.2	126.8850	34.6819	397.573	1410	3.870	65.9	4.75	769
37	Congo, Dem. Rep.	116.0	137.2740	26.4194	165.664	609	20.800	57.5	6.54	334
40	Cote d'Ivoire	111.0	617.3200	64.6600	528.260	2690	5.390	56.3	5.27	1220
56	Gambia	80.3	133.7560	31.9778	239.974	1660	4.300	65.5	5.71	562
63	Guinea	109.0	196.3440	31.9464	279.936	1190	16.100	58.0	5.34	648
64	Guinea-Bissau	114.0	81.5030	46.4950	192.544	1390	2.970	55.6	5.05	547
66	Haiti	208.0	101.2860	45.7442	428.314	1500	5.450	32.1	3.33	662
87	Lesotho	99.7	460.9800	129.8700	1181.700	2380	4.150	46.5	3.30	1170
88	Liberia	89.3	62.4570	38.5860	302.802	700	5.470	60.8	5.02	327
94	Malawi	90.5	104.6520	30.2481	160.191	1030	12.100	53.1	5.31	459
97	Mali	137.0	161.4240	35.2584	248.508	1870	4.370	59.5	6.55	708
106	Mozambique	101.0	131.9850	21.8299	193.578	918	7.640	54.5	5.56	419
112	Niger	123.0	77.2560	17.9568	170.868	814	2.550	58.8	7.49	348
132	Sierra Leone	160.0	67.0320	52.2690	137.655	1220	17.200	55.0	5.20	399
150	Togo	90.3	196.1760	37.3320	279.624	1210	1.180	58.7	4.87	488
155	Uganda	81.0	101.7450	53.6095	170.170	1540	10.600	56.8	6.15	595