Clustering Assignment

Prepared By:

Mohammad Aftab Alam

Abstract

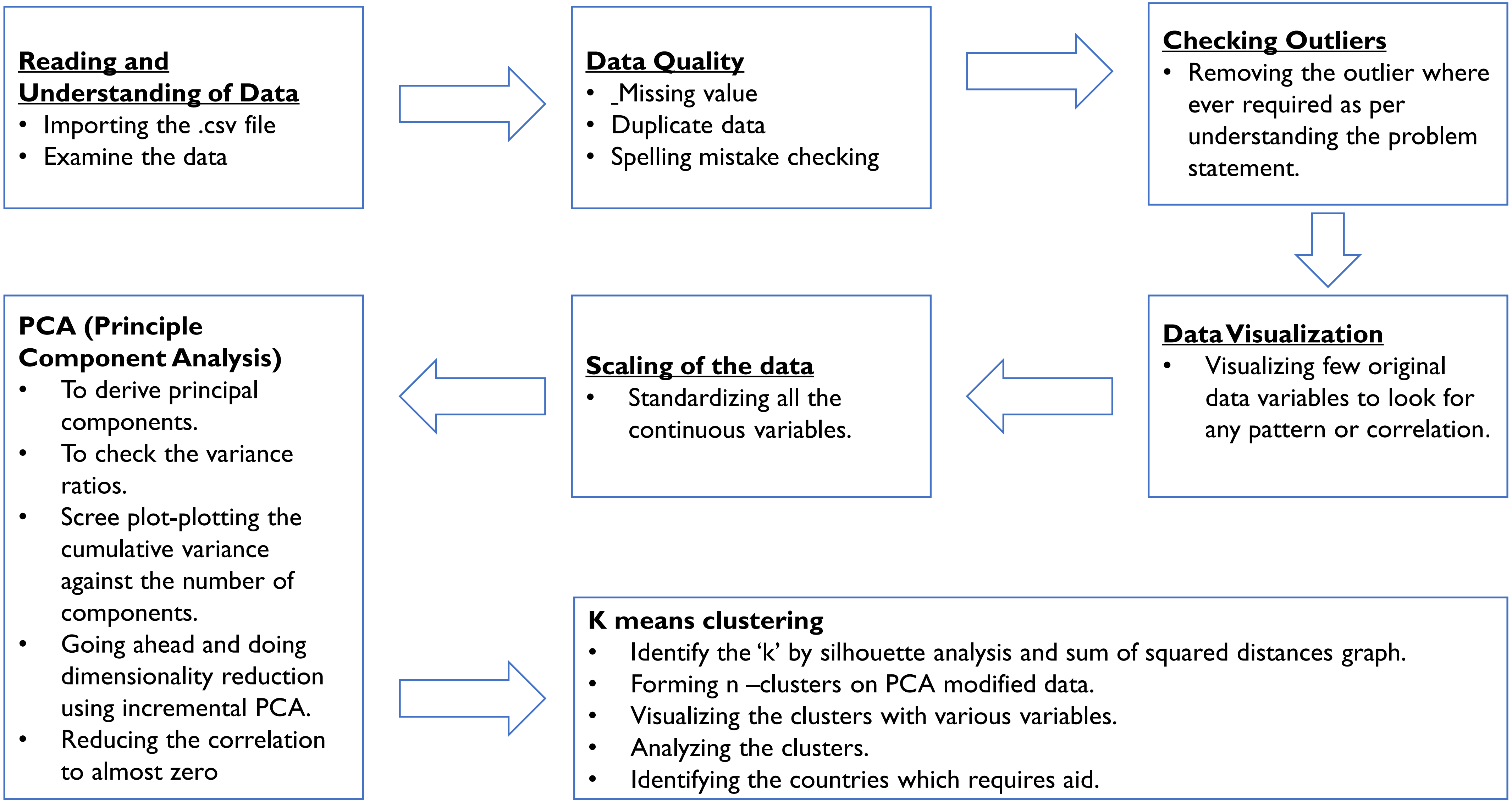
Problem statement :-

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.

And this is where you come in as a data analyst. Your job is to categorise the countries using some socio-economic and health factors that determine the overall development of the country. Then you need to suggest the countries which the CEO needs to focus on the most. The datasets containing those socio-economic factors and the corresponding data dictionary

Analysis Methodology



Reading and Understanding of Data

- Importing the .csv file
- Examine the data

Data Quality

- _Missing value
- Duplicate data
- Spelling mistake checking

Checking Outliers

- Removing the outlier where ever required as per understanding the problem statement.

Data Visualization

- Visualizing few original data variables to look for any pattern or correlation.

Scaling of the data

- Standardizing all the continuous variables.

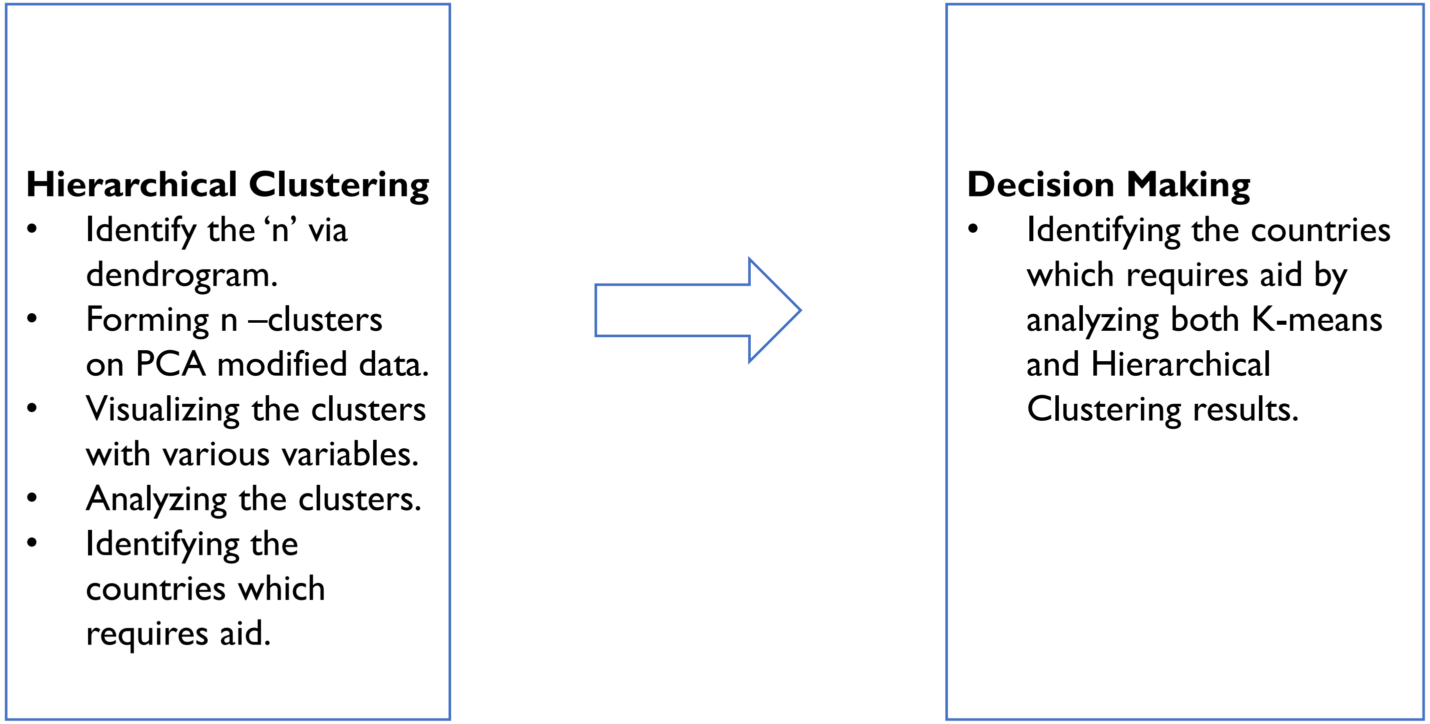
PCA (Principle Component Analysis)

- To derive principal components.
- To check the variance ratios.
- Scree plot-plotting the cumulative variance against the number of components.
- Going ahead and doing dimensionality reduction using incremental PCA.
- Reducing the correlation to almost zero

K means clustering

- Identify the ‘k’ by silhouette analysis and sum of squared distances graph.
- Forming n –clusters on PCA modified data.
- Visualizing the clusters with various variables.
- Analyzing the clusters.
- Identifying the countries which requires aid.

Analysis Methodology Cont…



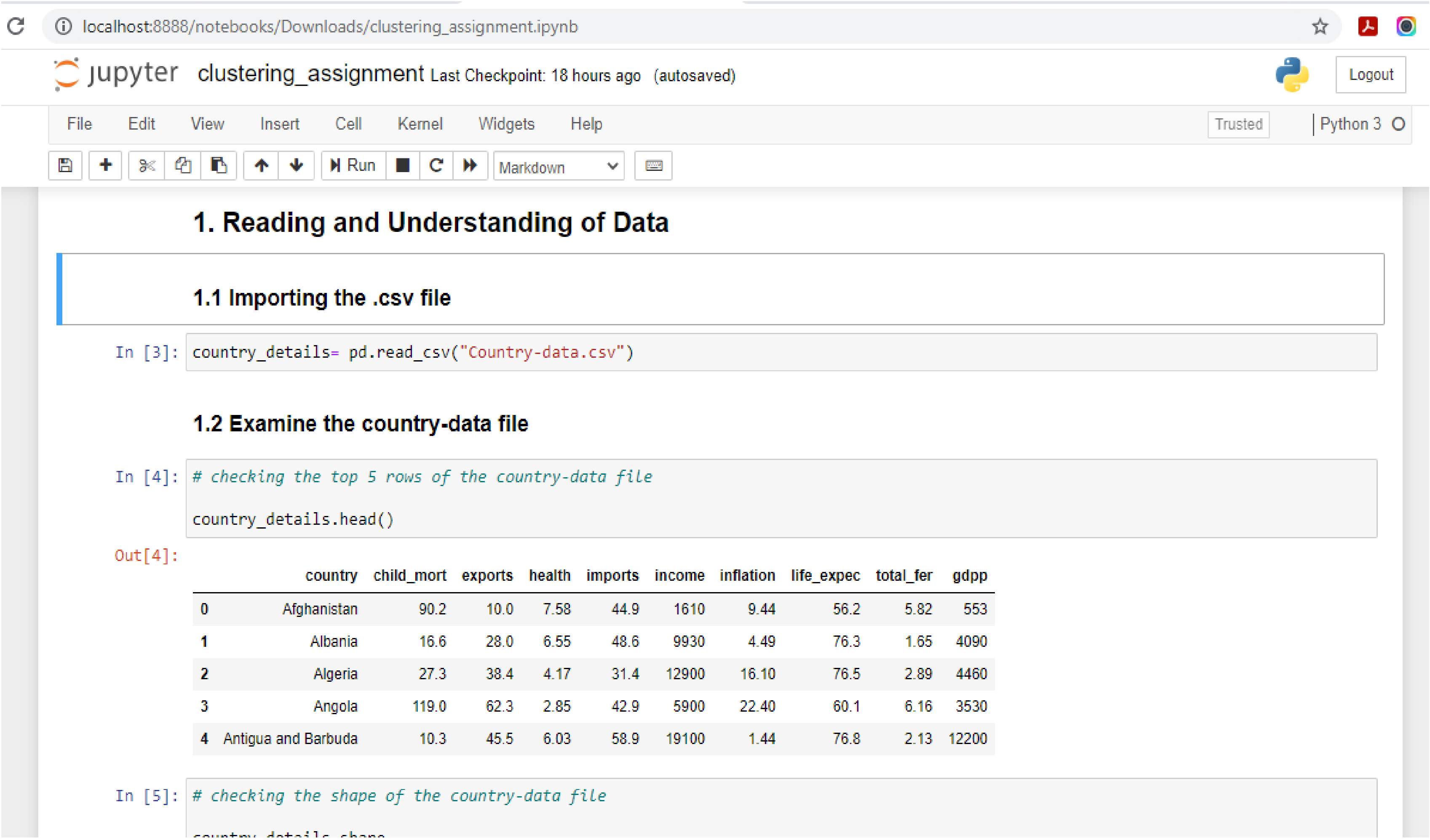
Hierarchical Clustering

- Identify the ‘n’ via dendrogram.
- Forming n –clusters on PCA modified data.
- Visualizing the clusters with various variables.
- Analyzing the clusters.
- Identifying the countries which requires aid.

Decision Making

- Identifying the countries which requires aid by analyzing both K-means and Hierarchical Clustering results.

Reading and Understanding of Data


localhost:8888/notebooks/Downloads/clustering_assignment.ipynb
jupyter
clustering_assignment Last Checkpoint: 18 hours ago (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Markdown

1. Reading and Understanding of Data

1.1 Importing the .csv file

```
In [3]: country_details= pd.read_csv("Country-data.csv")
```

1.2 Examine the country-data file

```
In [4]: # checking the top 5 rows of the country-data file

        country_details.head()
```

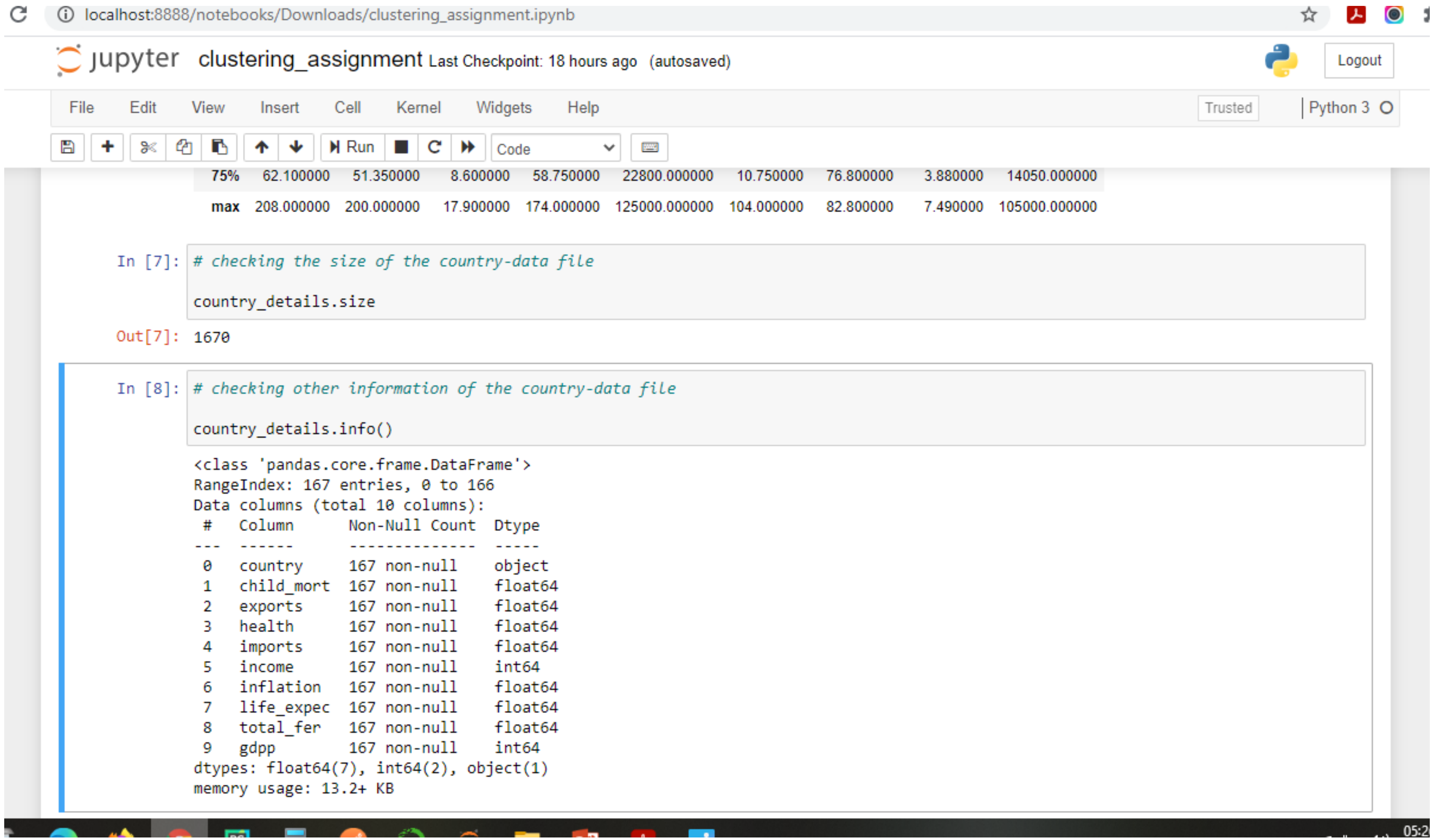
Out[4]:

	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

```
In [5]: # checking the shape of the country-data file

        country_details.shape
```

- Importing the .csv file using pandas.
- Reading the .csv file.
- Checking Describe details

Reading and Understanding of Data Cont…



localhost:8888/notebooks/Downloads/clustering_assignment.ipynb
jupyter clustering_assignment Last Checkpoint: 18 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

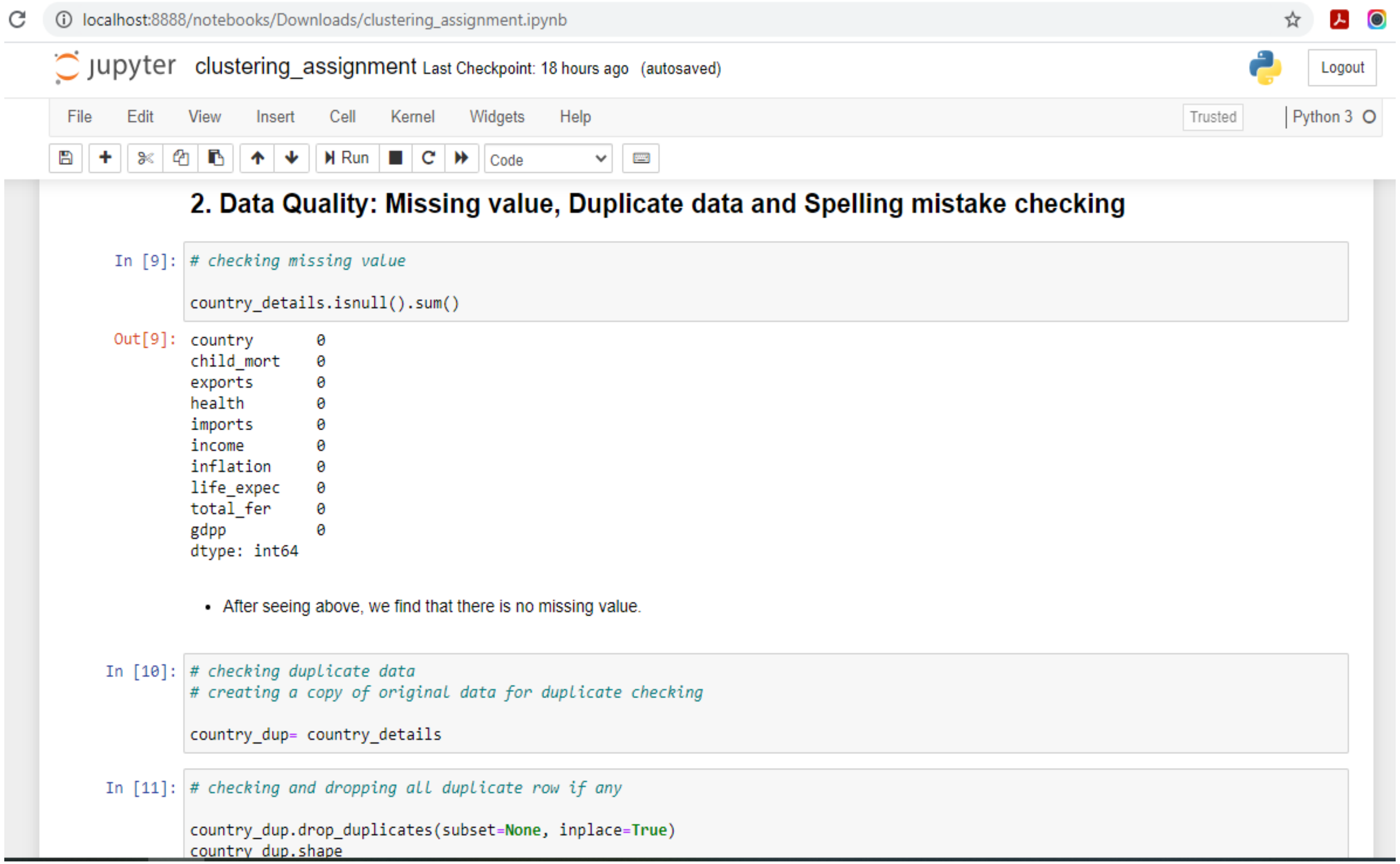
```
75%   62.100000  51.350000   8.600000   58.750000   22800.000000   10.750000  76.800000  3.880000   14050.000000
max  208.000000 200.000000  17.900000  174.000000  125000.000000  104.000000  82.800000  7.490000  105000.000000
```

```
In [7]: # checking the size of the country-data file

        country_details.size

Out[7]: 1670
```

```
In [8]: # checking other information of the country-data file

        country_details.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 167 entries, 0 to 166
Data columns (total 10 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   country     167 non-null    object
 1   child_mort  167 non-null    float64
 2   exports     167 non-null    float64
 3   health      167 non-null    float64
 4   imports     167 non-null    float64
 5   income      167 non-null    int64
 6   inflation   167 non-null    float64
 7   life_expec  167 non-null    float64
 8   total_fer   167 non-null    float64
 9   gdpp        167 non-null    int64
dtypes: float64(7), int64(2), object(1)
memory usage: 13.2+ KB
```

- Checking size of the data.
- Checking infor all other details

Data Quality



localhost:8888/notebooks/Downloads/clustering_assignment.ipynb
jupyter clustering_assignment Last Checkpoint: 18 hours ago (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code

2. Data Quality: Missing value, Duplicate data and Spelling mistake checking

In [9]:

```
# checking missing value

country_details.isnull().sum()
```

Out[9]:

```
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
```

- After seeing above, we find that there is no missing value.

In [10]:

```
# checking duplicate data
# creating a copy of original data for duplicate checking

country_dup= country_details
```

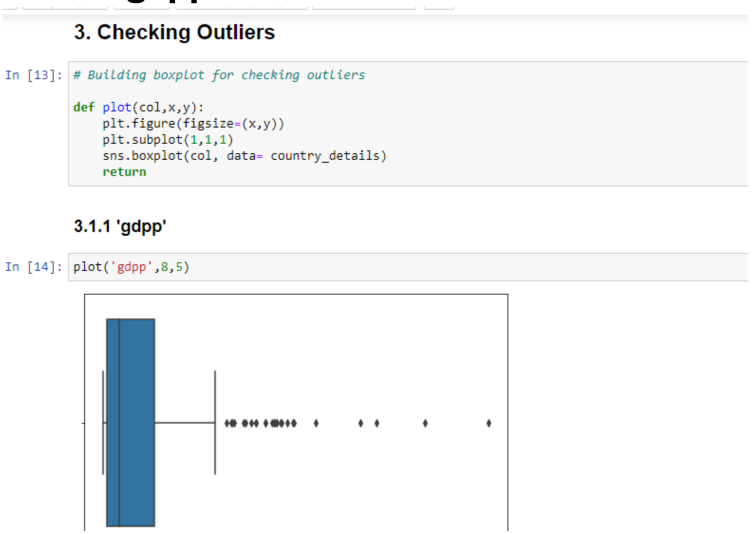
In [11]:

```
# checking and dropping all duplicate row if any

country_dup.drop_duplicates(subset=None, inplace=True)
country_dup.shape
```

- No Missing value found.
- No Duplicate data found.
- So we will move to the further step.

Checking Outliers

‘gdpp’ variables



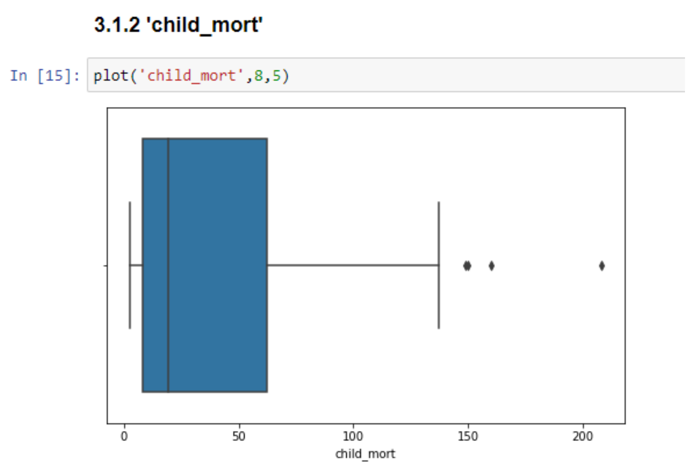
3. Checking Outliers

```
In [13]: # Building boxplot for checking outliers

         def plot(col,x,y):
             plt.figure(figsize=(x,y))
             plt.subplot(1,1,1)
             sns.boxplot(col, data= country_details)
             return
```

3.1.1 'gdpp'

```
In [14]: plot('gdpp',8,5)
```

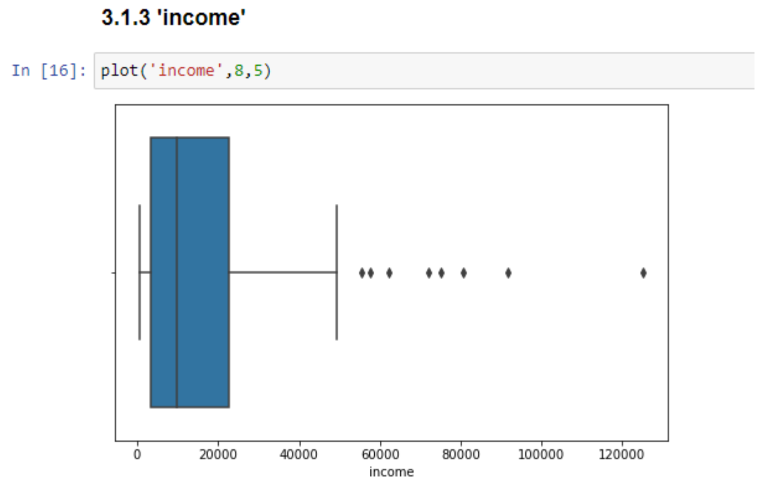
‘child_mort’ variables



3.1.2 'child_mort'

```
In [15]: plot('child_mort',8,5)
```

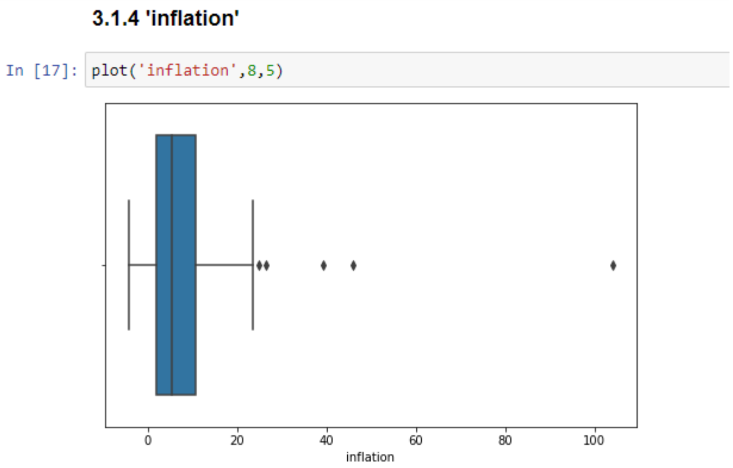
‘income’ variables



3.1.3 'income'

```
In [16]: plot('income',8,5)
```

‘inflation’ variables



3.1.4 'inflation'

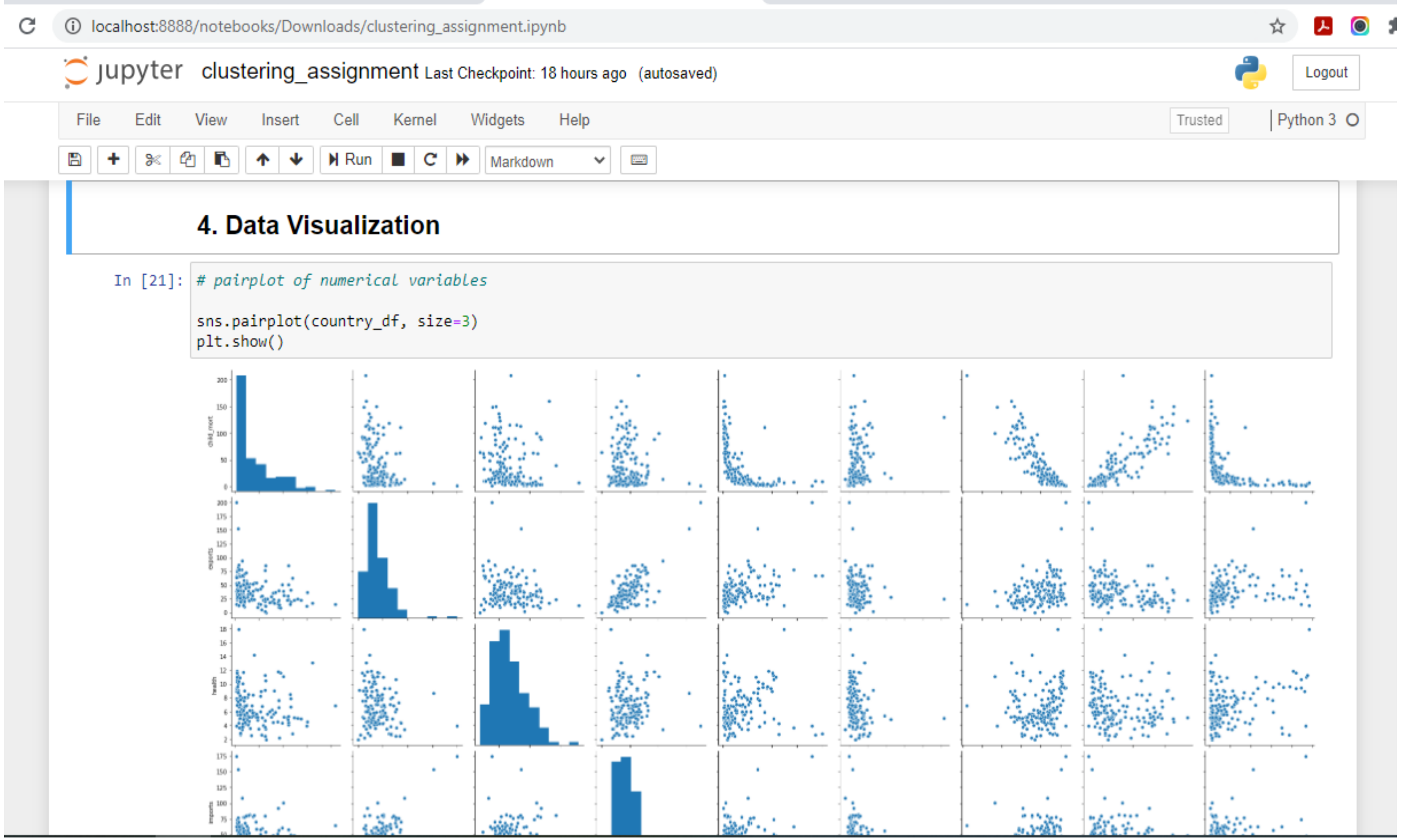
```
In [17]: plot('inflation',8,5)
```

After seeing the outliers of some important variables we will conclude that :-

- Variables ‘gdpp’, ‘child_mort’, ‘income’ and ‘inflation’ columns has high outliers.
- As of now, not remove the outliers as it may suits the business needs or a lot of countries are getting removed.

Data Visualization

Visualization using Pairplot



clustering_assignment Last Checkpoint: 18 hours ago (autosaved)

4. Data Visualization

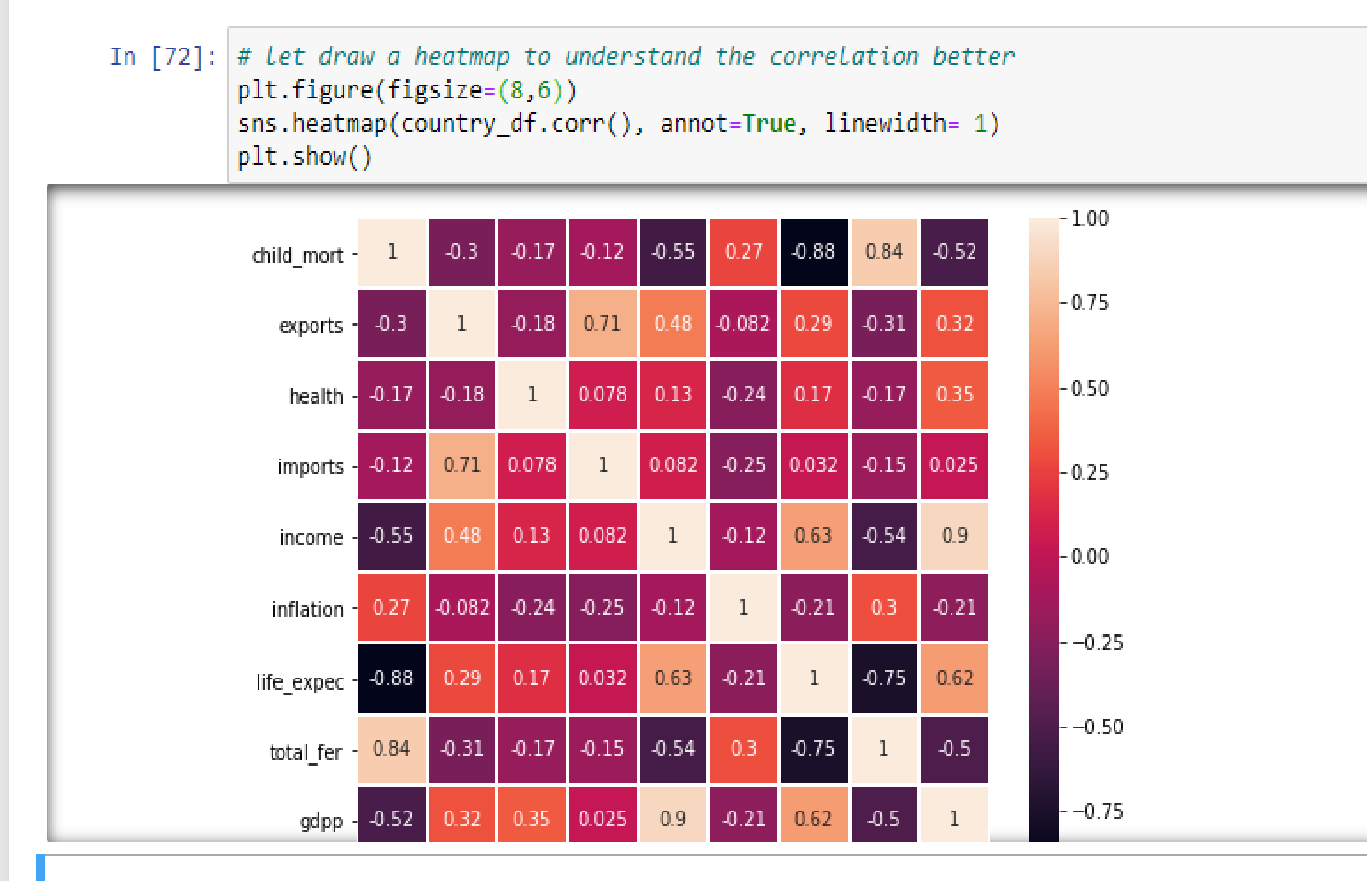
In [21]:

```
# pairplot of numerical variables

sns.pairplot(country_df, size=3)
plt.show()
```

- By seeing Pairplot, we are getting the clear result about the correlation between the variables.
- Thus we can go through for Heatmap for more clear result.

Data Visualization Cont…

Visualization using Heatmap

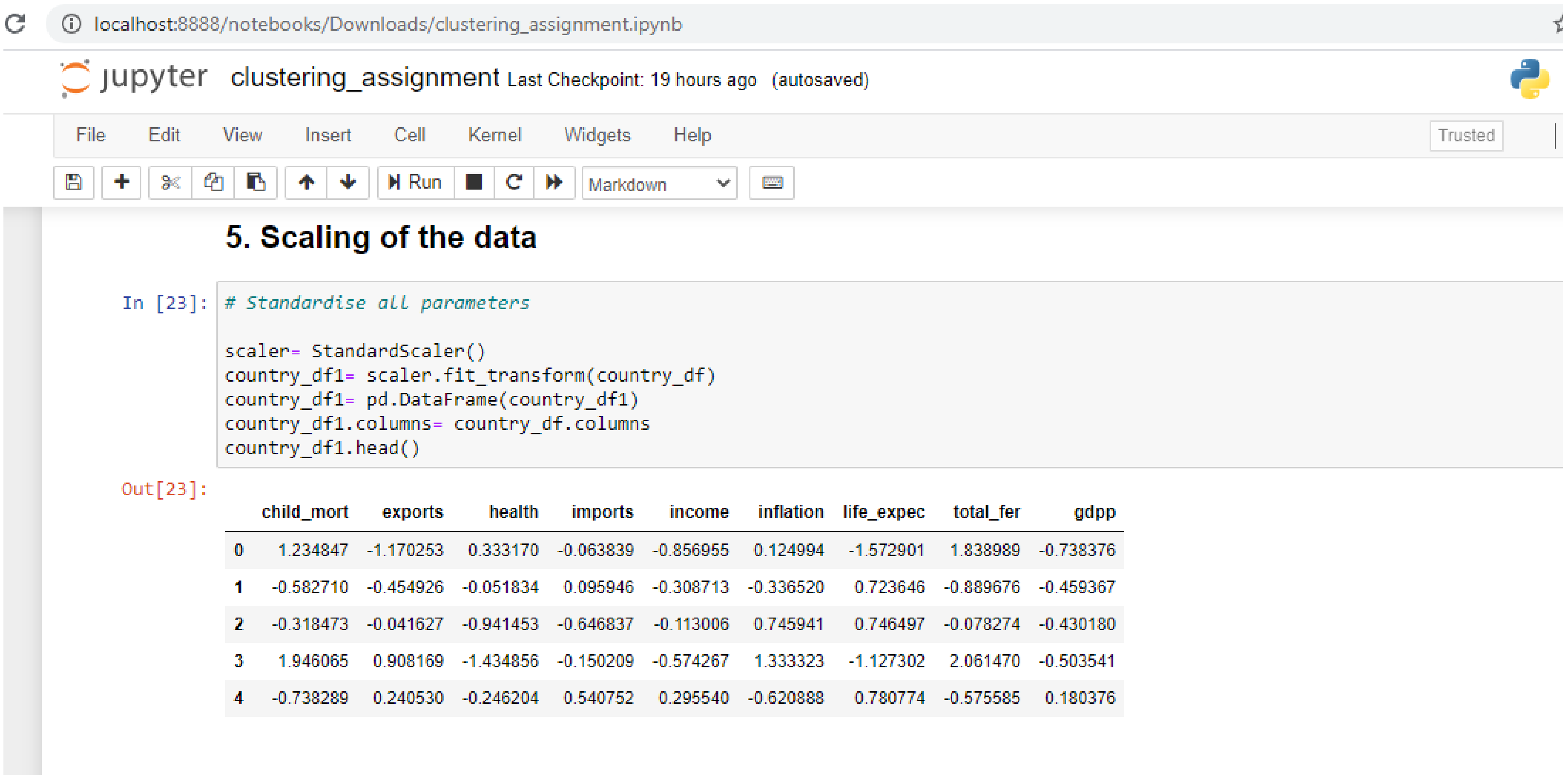


```
In [72]: # let draw a heatmap to understand the correlation better
         plt.figure(figsize=(8,6))
         sns.heatmap(country_df.corr(), annot=True, linewidth= 1)
         plt.show()
```

child_mort	1	-0.3	-0.17	-0.12	-0.55	0.27	-0.88	0.84	-0.52
exports	-0.3	1	-0.18	0.71	0.48	-0.082	0.29	-0.31	0.32
health	-0.17	-0.18	1	0.078	0.13	-0.24	0.17	-0.17	0.35
imports	-0.12	0.71	0.078	1	0.082	-0.25	0.032	-0.15	0.025
income	-0.55	0.48	0.13	0.082	1	-0.12	0.63	-0.54	0.9
inflation	0.27	-0.082	-0.24	-0.25	-0.12	1	-0.21	0.3	-0.21
life_expec	-0.88	0.29	0.17	0.032	0.63	-0.21	1	-0.75	0.62
total_fer	0.84	-0.31	-0.17	-0.15	-0.54	0.3	-0.75	1	-0.5
gdpp	-0.52	0.32	0.35	0.025	0.9	-0.21	0.62	-0.5	1

- By plotting heatmap, we can see high correlation between:
  - 'child_mort' and 'total_fer‘.
  - 'exports' and 'imports‘.
  - 'income' and 'gdpp‘
- This will cause problem for the upcoming analysis, hence need to be removed but they have valuable information which we can't afford to loose.
- So, we will use PCA to overcome this multicollinearity.
- This will not only take care of multicollinearity but also will preserve the valuable information and also demensionality reduction.

Scaling of the data



localhost:8888/notebooks/Downloads/clustering_assignment.ipynb
jupyter
clustering_assignment Last Checkpoint: 19 hours ago (autosaved)
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Run
Markdown

5. Scaling of the data

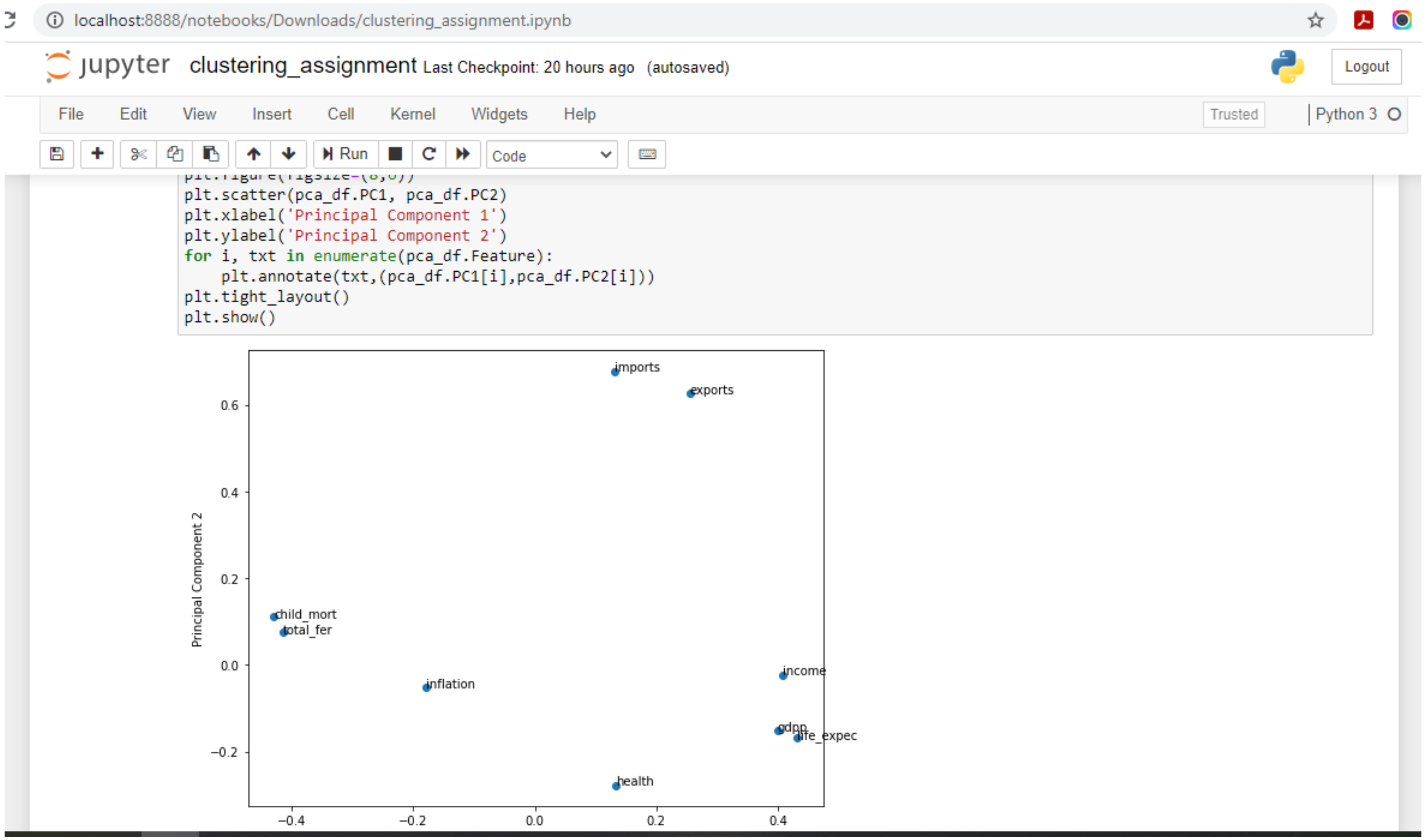
```
In [23]: # Standardise all parameters

         scaler= StandardScaler()
         country_df1= scaler.fit_transform(country_df)
         country_df1= pd.DataFrame(country_df1)
         country_df1.columns= country_df.columns
         country_df1.head()
```

Out[23]:

	child_mort	exports	health	imports	income	inflation	life_expec	total_fer	gdpp
0	1.234847	-1.170253	0.333170	-0.063839	-0.856955	0.124994	-1.572901	1.838989	-0.738376
1	-0.582710	-0.454926	-0.051834	0.095946	-0.308713	-0.336520	0.723646	-0.889676	-0.459367
2	-0.318473	-0.041627	-0.941453	-0.646837	-0.113006	0.745941	0.746497	-0.078274	-0.430180
3	1.946065	0.908169	-1.434856	-0.150209	-0.574267	1.333323	-1.127302	2.061470	-0.503541
4	-0.738289	0.240530	-0.246204	0.540752	0.295540	-0.620888	0.780774	-0.575585	0.180376

- We are using scaling of the data for standardizing all the continuous variables.

PCA (Principle Component Analysis)

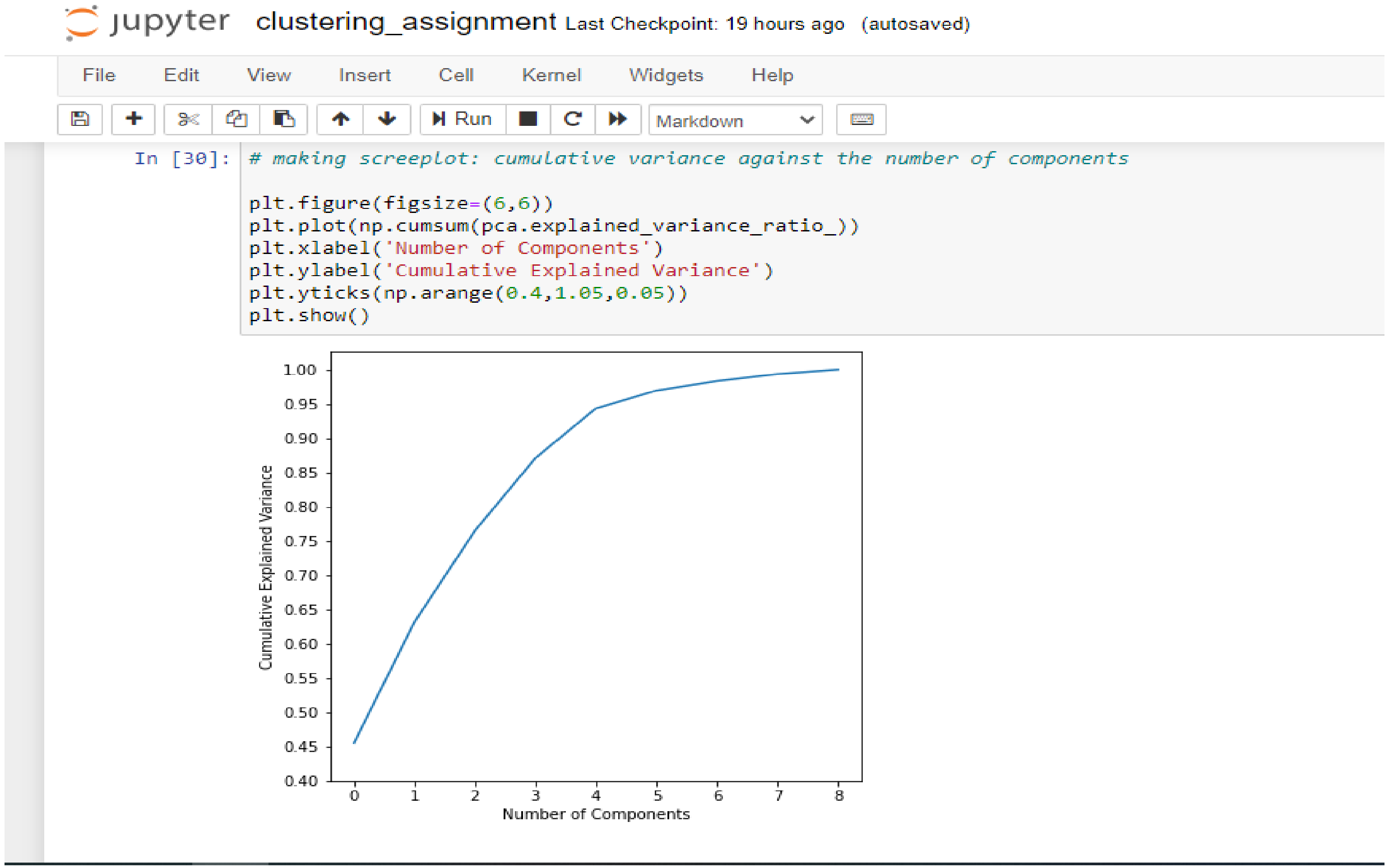


```
plt.scatter(pca_df.PC1, pca_df.PC2)
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
for i, txt in enumerate(pca_df.Feature):
    plt.annotate(txt,(pca_df.PC1[i],pca_df.PC2[i]))
plt.tight_layout()
plt.show()
```

Visualising the features by loaded along PC1 and PC2

- We see that features like gdpp, life-expectancy and income are along the direction of PC1 and other features like total-fertility and child- mortality are along PC2 direction.

PCA (Principle Component Analysis) Cont…



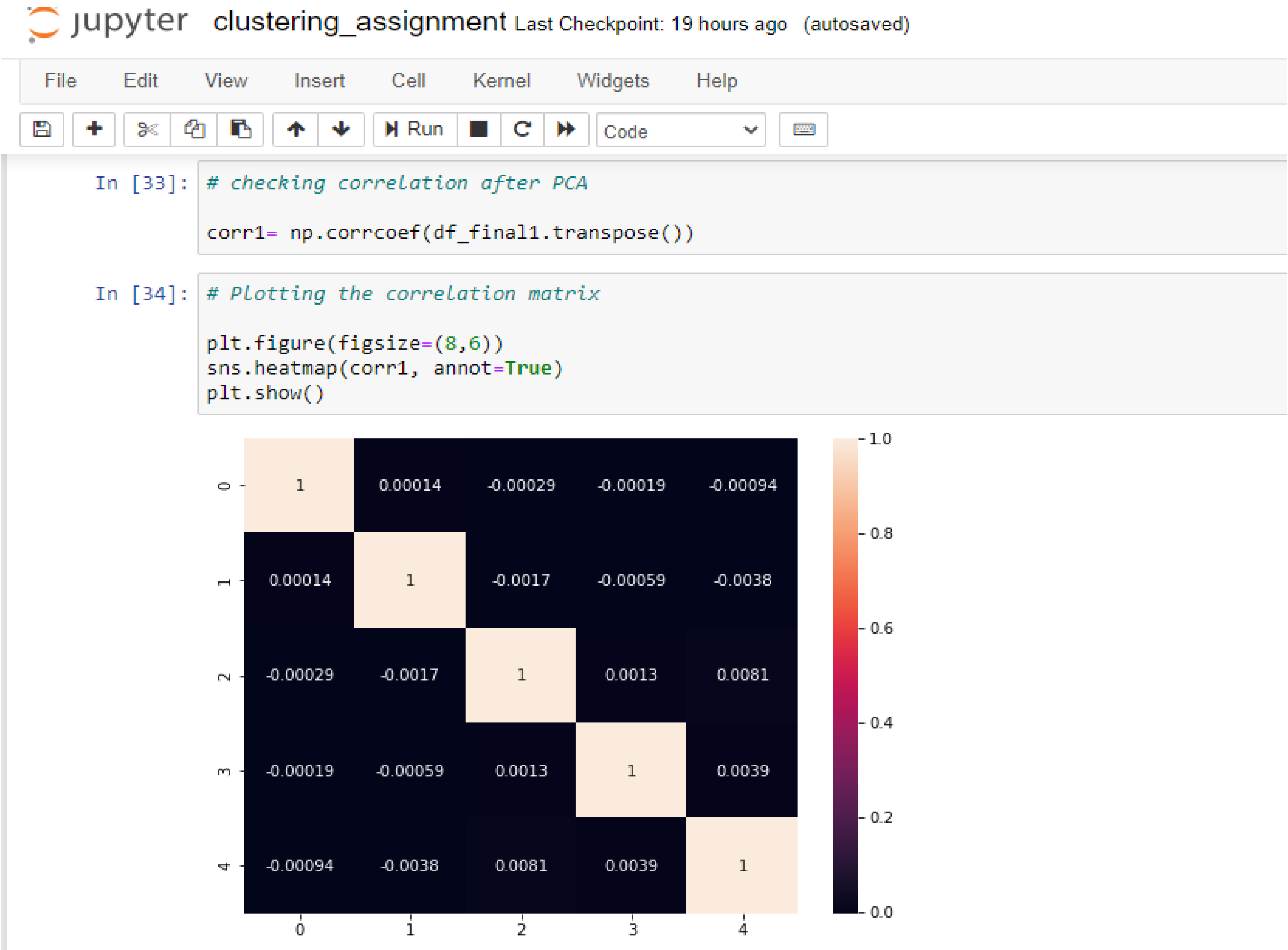
```
# making screeplot: cumulative variance against the number of components

plt.figure(figsize=(6,6))
plt.plot(np.cumsum(pca.explained_variance_ratio_))
plt.xlabel('Number of Components')
plt.ylabel('Cumulative Explained Variance')
plt.yticks(np.arange(0.4,1.05,0.05))
plt.show()
```

Scree-plot, plotting b/w the cumulative variance against the number of components.

Conclusion:

- Around 95% of the cumulative variance is being explained by 5 components

PCA (Principle Component Analysis) Cont…



jupyter clustering_assignment Last Checkpoint: 19 hours ago (autosaved)
File Edit View Insert Cell Kernel Widgets Help
Run
Code
1.0
0.8
0.6
0.4
0.2
0.0

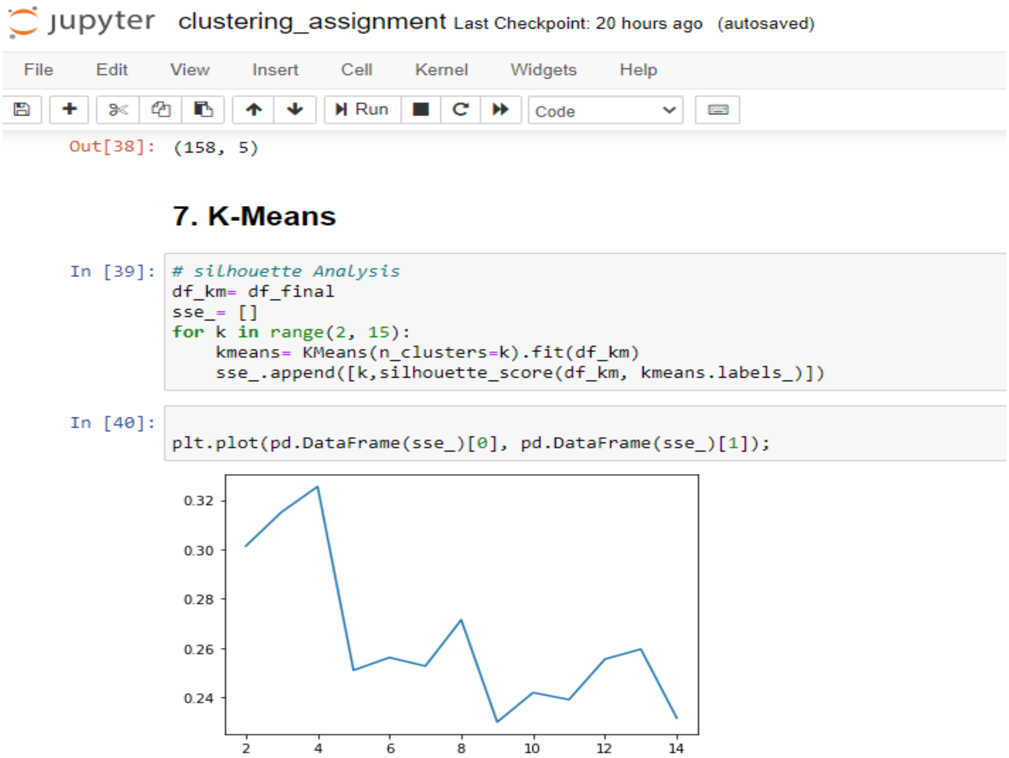
```
In [33]: # checking correlation after PCA

         corr1= np.corrcoef(df_final1.transpose())
```

```
In [34]: # Plotting the correlation matrix

         plt.figure(figsize=(8,6))
         sns.heatmap(corr1, annot=True)
         plt.show()
```

	0	1	2	3	4
0	1	0.00014	-0.00029	-0.00019	-0.00094
1	0.00014	1	-0.0017	-0.00059	-0.0038
2	-0.00029	-0.0017	1	0.0013	0.0081
3	-0.00019	-0.00059	0.0013	1	0.0039
4	-0.00094	-0.0038	0.0081	0.0039	1

- After doing dimensionality reduction via incremental PCA by taking 5 components, we see that the correlation in the data has almost reduced to zero.

K means clustering

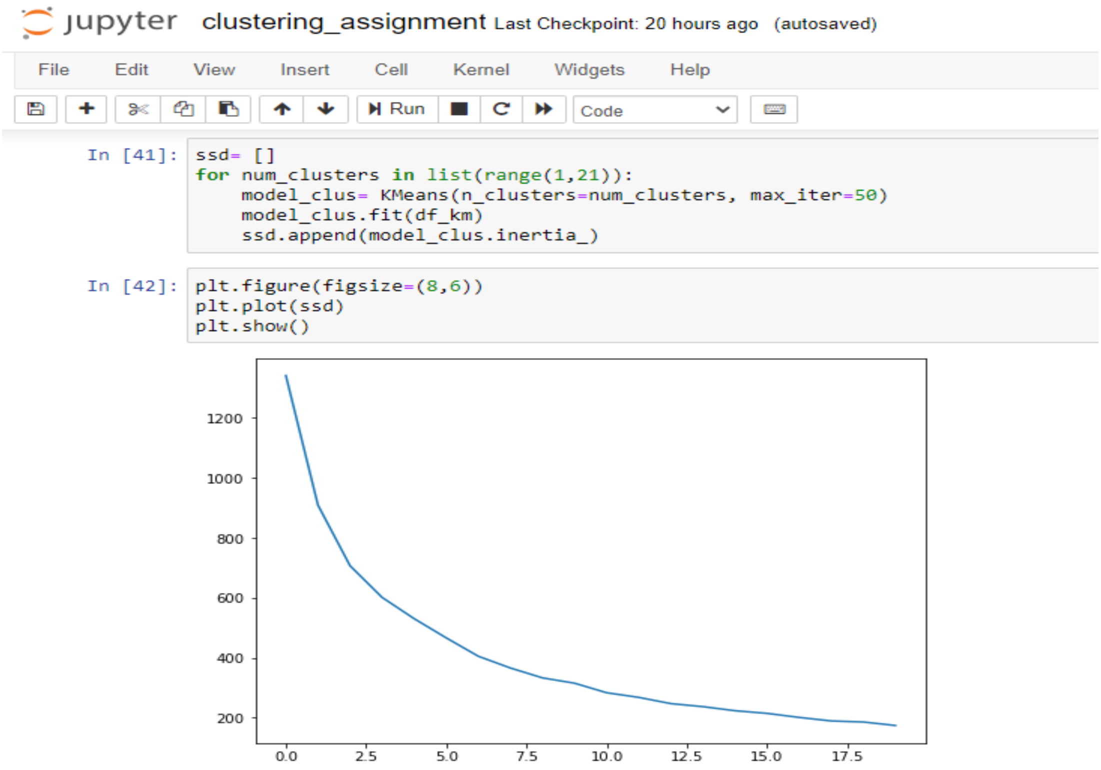

jupyter clustering_assignment Last Checkpoint: 20 hours ago (autosaved)
File Edit View Insert Cell Kernel Widgets Help
Run
Code
0.32
0.30
0.28
0.26
0.24
2
4
6
8
10
12
14

```
Out[38]: (158, 5)
```

7. K-Means

```
In [39]: # silhouette Analysis
         df_km= df_final
         sse_= []
         for k in range(2, 15):
             kmeans= KMeans(n_clusters=k).fit(df_km)
             sse_.append([k,silhouette_score(df_km, kmeans.labels_)])

In [40]: plt.plot(pd.DataFrame(sse_)[0], pd.DataFrame(sse_)[1]);
```

Silhouette Analysis

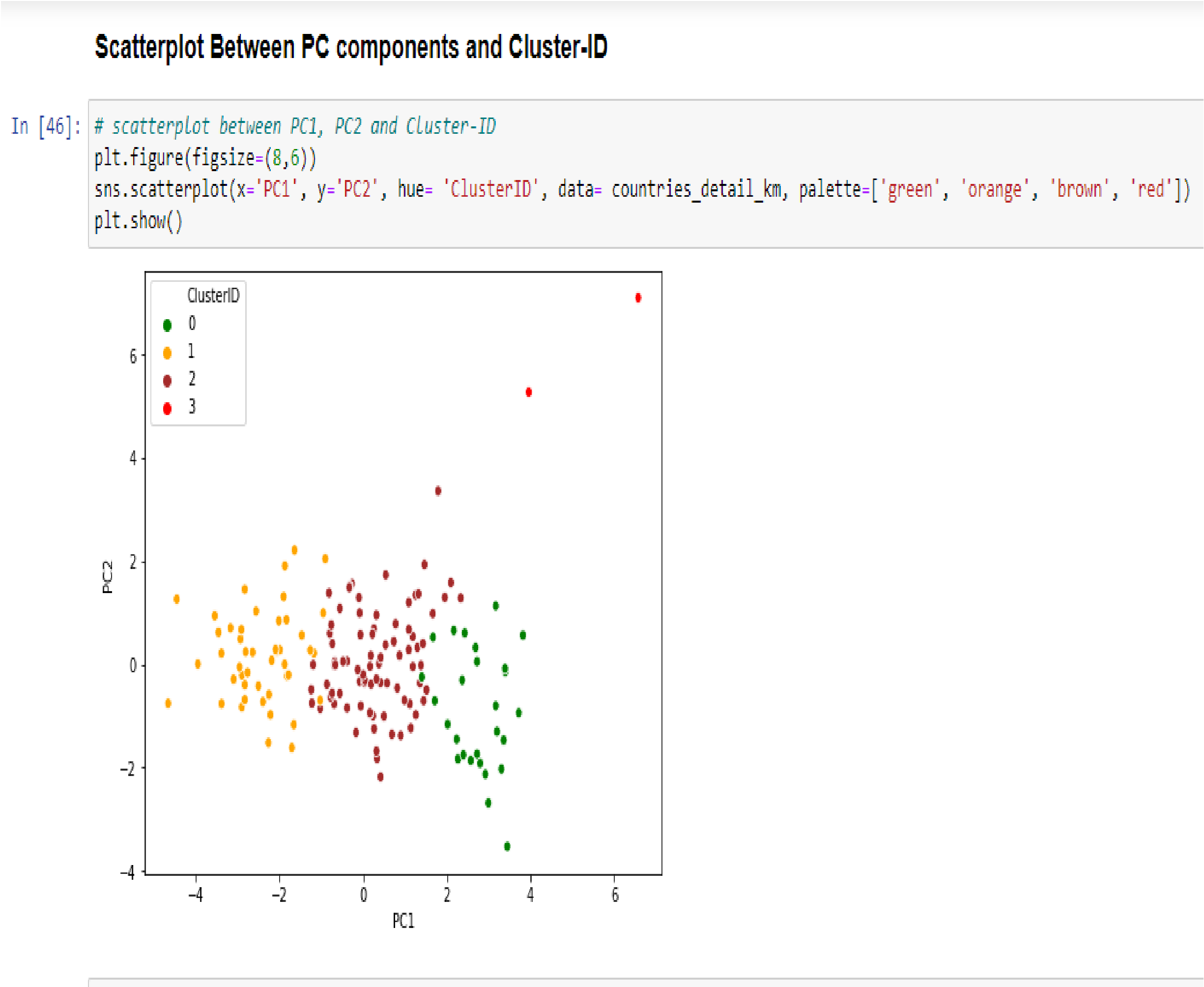

jupyter clustering_assignment Last Checkpoint: 20 hours ago (autosaved)
File Edit View Insert Cell Kernel Widgets Help
Run
Code
1200
1000
800
600
400
200
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5

```
In [41]: ssd= []
         for num_clusters in list(range(1,21)):
             model_clus= KMeans(n_clusters=num_clusters, max_iter=50)
             model_clus.fit(df_km)
             ssd.append(model_clus.inertia_)

In [42]: plt.figure(figsize=(8,6))
         plt.plot(ssd)
         plt.show()
```

Sum of squared distance

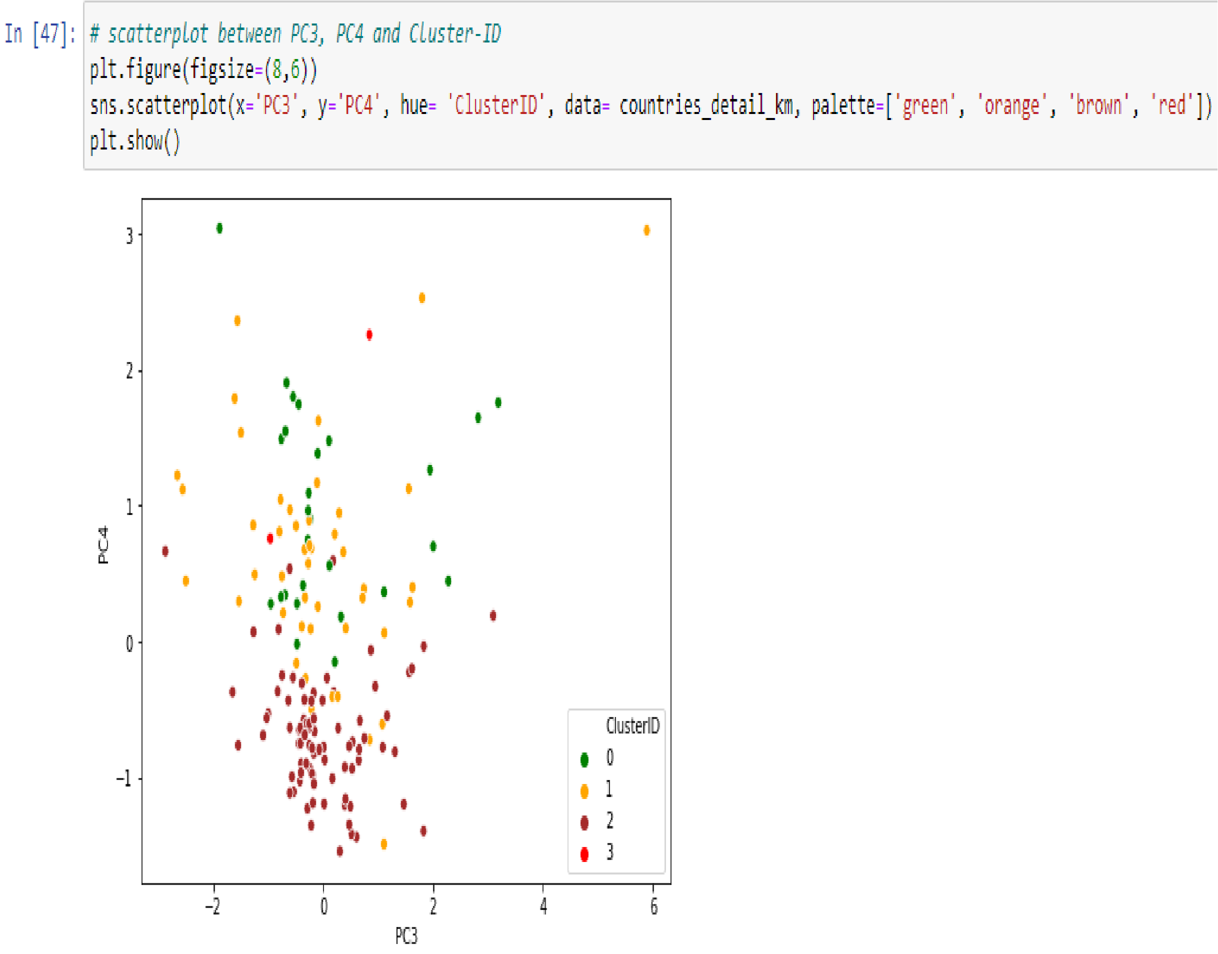
- By looking silhouette analysis, we see the highest peak is at k=4 and in sum of squared distances graph, we see that the elbow is in the range of 3 to 5, so we are going ahead with k as 4.

K means clustering Cont…



Scatterplot Between PC components and Cluster-ID

```
In [46]: # scatterplot between PC1, PC2 and Cluster-ID
         plt.figure(figsize=(8,6))
         sns.scatterplot(x='PC1', y='PC2', hue= 'ClusterID', data= countries_detail_km, palette=['green', 'orange', 'brown', 'red'])
         plt.show()
```

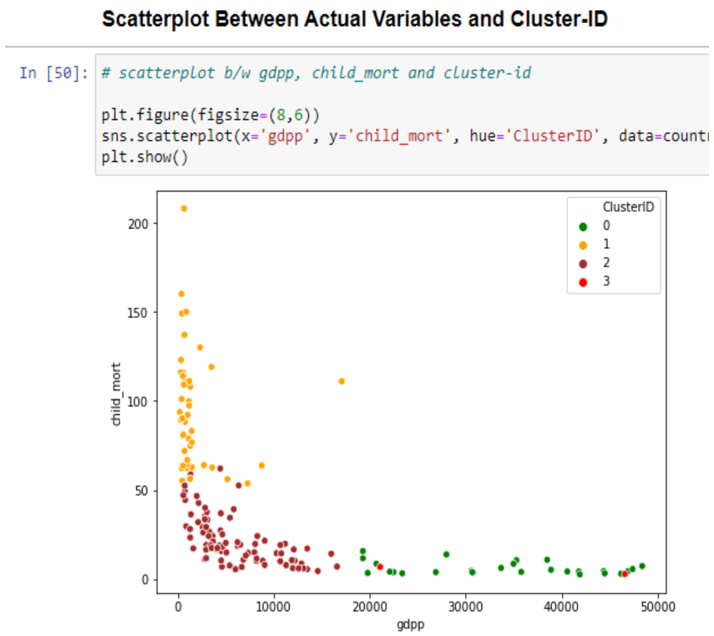


```
In [47]: # scatterplot between PC3, PC4 and Cluster-ID
         plt.figure(figsize=(8,6))
         sns.scatterplot(x='PC3', y='PC4', hue= 'ClusterID', data= countries_detail_km, palette=['green', 'orange', 'brown', 'red'])
         plt.show()
```

Scatterplot Between PC1 and PC2 and Cluster-ID

Scatterplot Between PC3 and PC4 and Cluster-ID

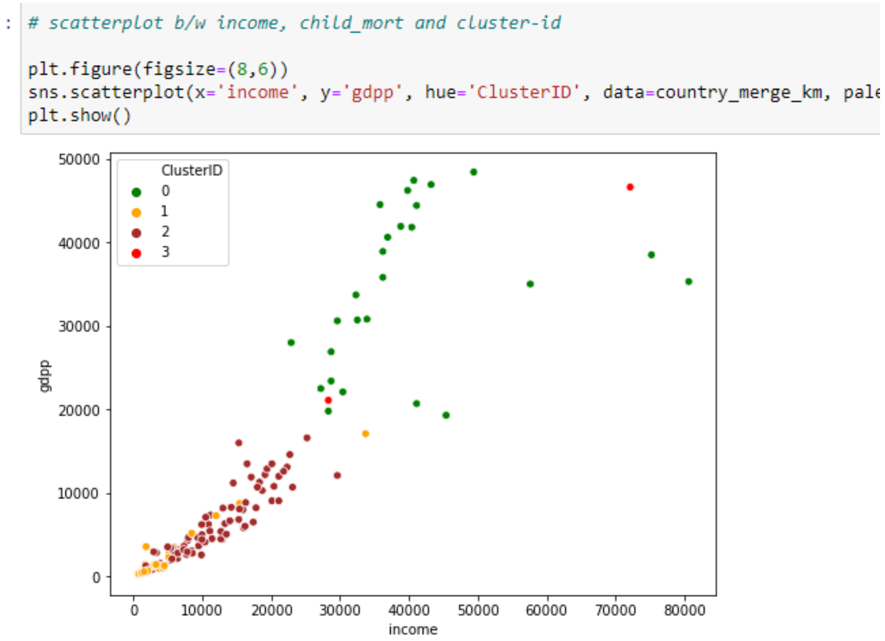
K means clustering Cont…



Scatterplot Between Actual Variables and Cluster-ID

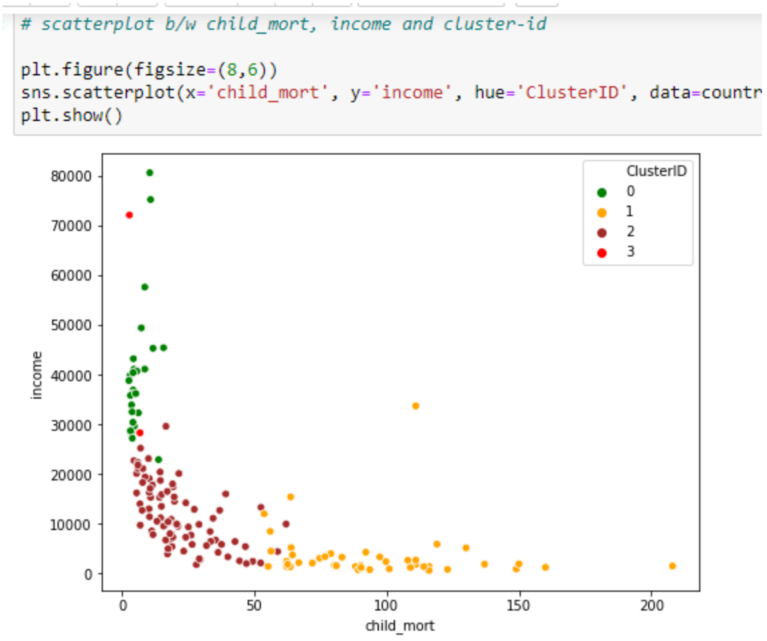
```
In [50]: # scatterplot b/w gdpp, child_mort and cluster-id

         plt.figure(figsize=(8,6))
         sns.scatterplot(x='gdpp', y='child_mort', hue='ClusterID', data=countr
         plt.show()
```

Scatterplot b/w gdpp, child_mort and ClusterID



```
: # scatterplot b/w income, child_mort and cluster-id

  plt.figure(figsize=(8,6))
  sns.scatterplot(x='income', y='gdpp', hue='ClusterID', data=country_merge_km, pale
  plt.show()
```

Scatterplot b/w income, gdpp and ClusterID

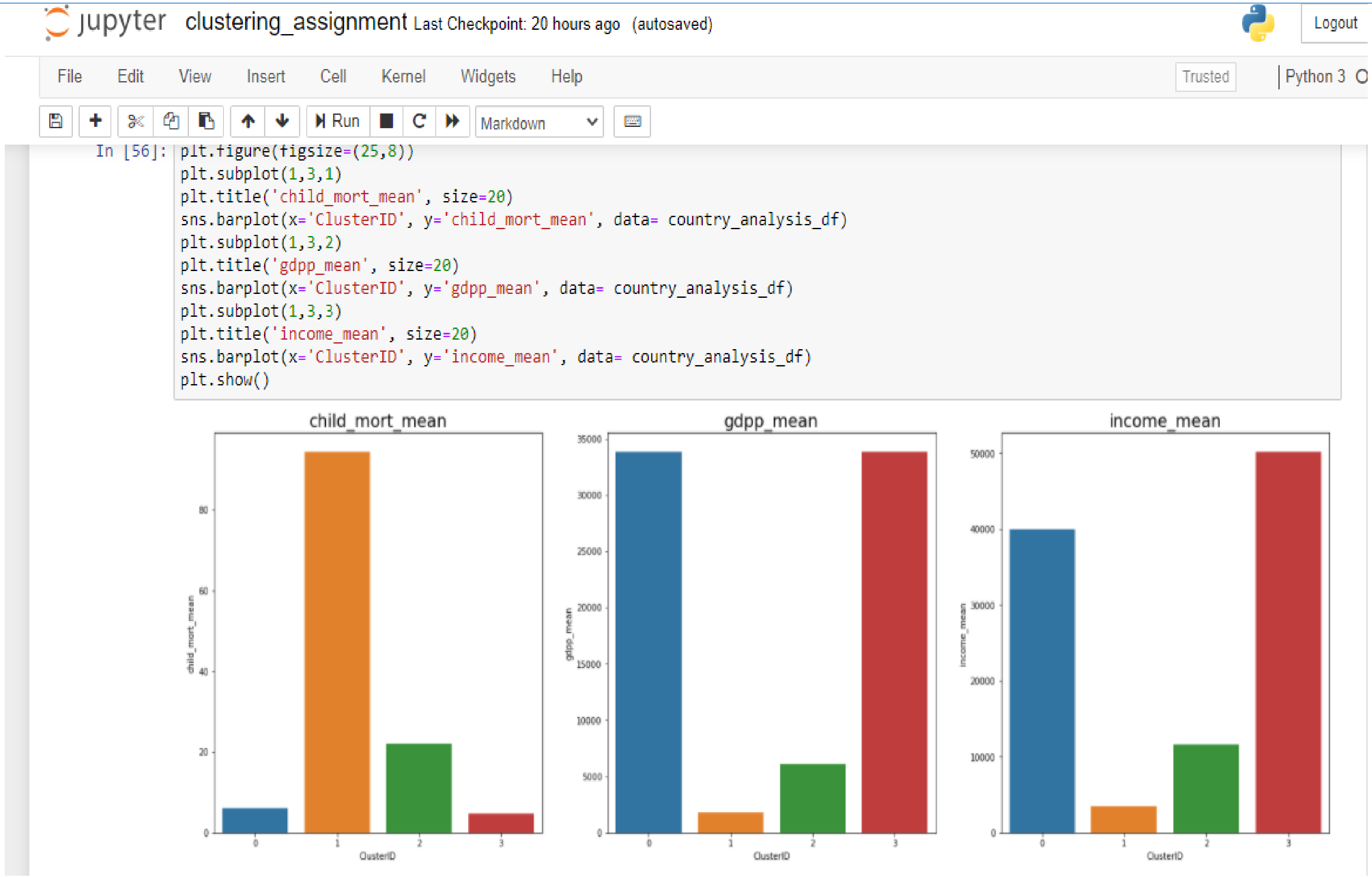


```
# scatterplot b/w child_mort, income and cluster-id

plt.figure(figsize=(8,6))
sns.scatterplot(x='child_mort', y='income', hue='ClusterID', data=countr
plt.show()
```

Scatterplot b/w child_mort, incomeand ClusterID

Scatterplot Between Actual Variables and Cluster-ID

K means clustering Cont…



```
plt.figure(figsize=(25,8))
plt.subplot(1,3,1)
plt.title('child_mort_mean', size=20)
sns.barplot(x='ClusterID', y='child_mort_mean', data= country_analysis_df)
plt.subplot(1,3,2)
plt.title('gdpp_mean', size=20)
sns.barplot(x='ClusterID', y='gdpp_mean', data= country_analysis_df)
plt.subplot(1,3,3)
plt.title('income_mean', size=20)
sns.barplot(x='ClusterID', y='income_mean', data= country_analysis_df)
plt.show()
```

Analysing the Clusters by using these three variables- 'gdpp', 'child_mort', 'income'

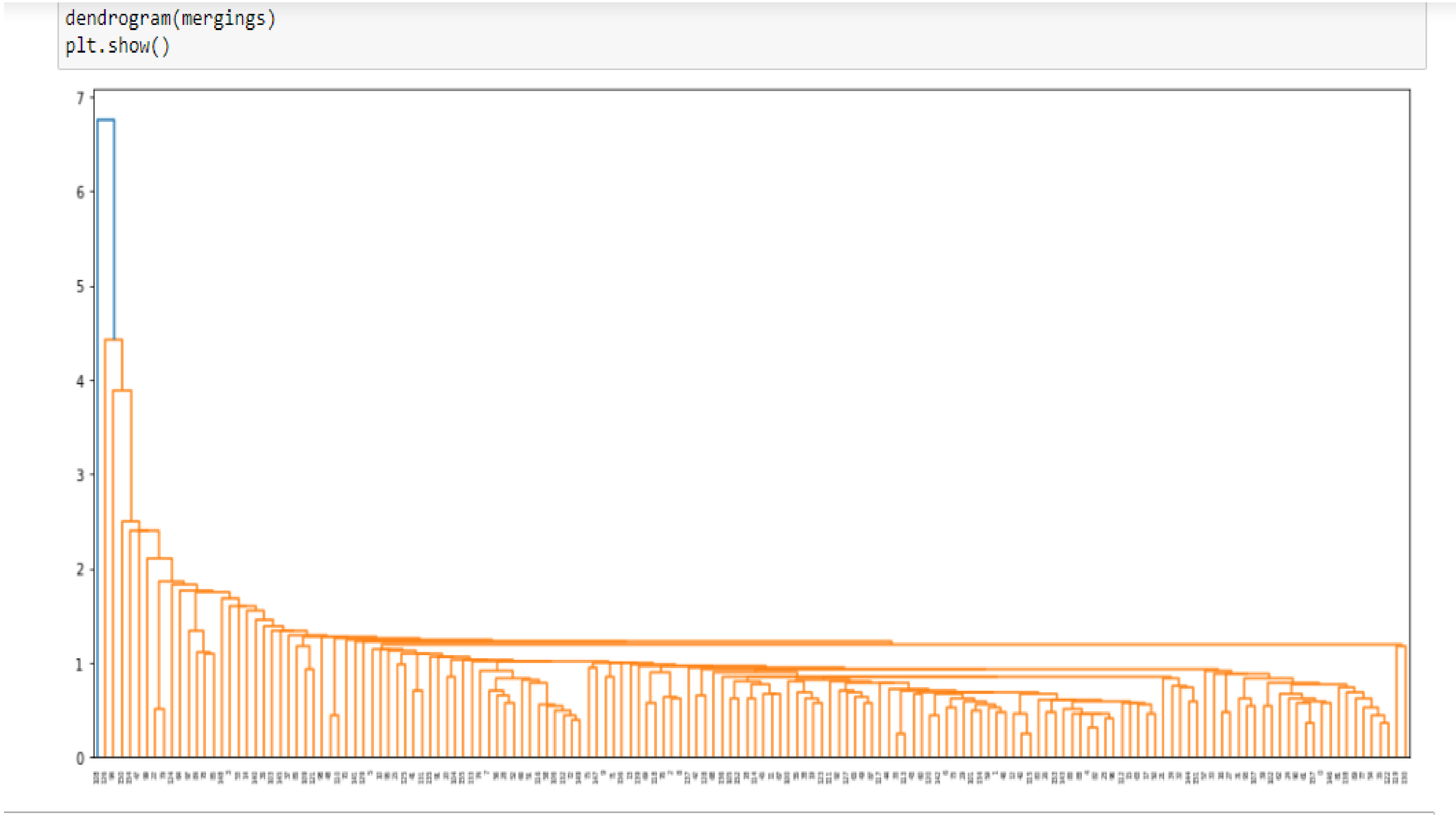
- By seeing the graph, we can conclude that 'cluster-1' is our cluster of concern because :-
  - It has highest child_mort.
  - Lowest gdpp.
  - And lowest income.

K means clustering Cont…

As per K-Means clustering, the country which are direst need of aid are :

- Burundi
- Liberia
- Congo, Dem, Rep.
- Niger
- Sierra Leone
- Madagascar
- Mozambique
- Central African Republic
- Malawi
- Eritrea

Hierarchical Clustering

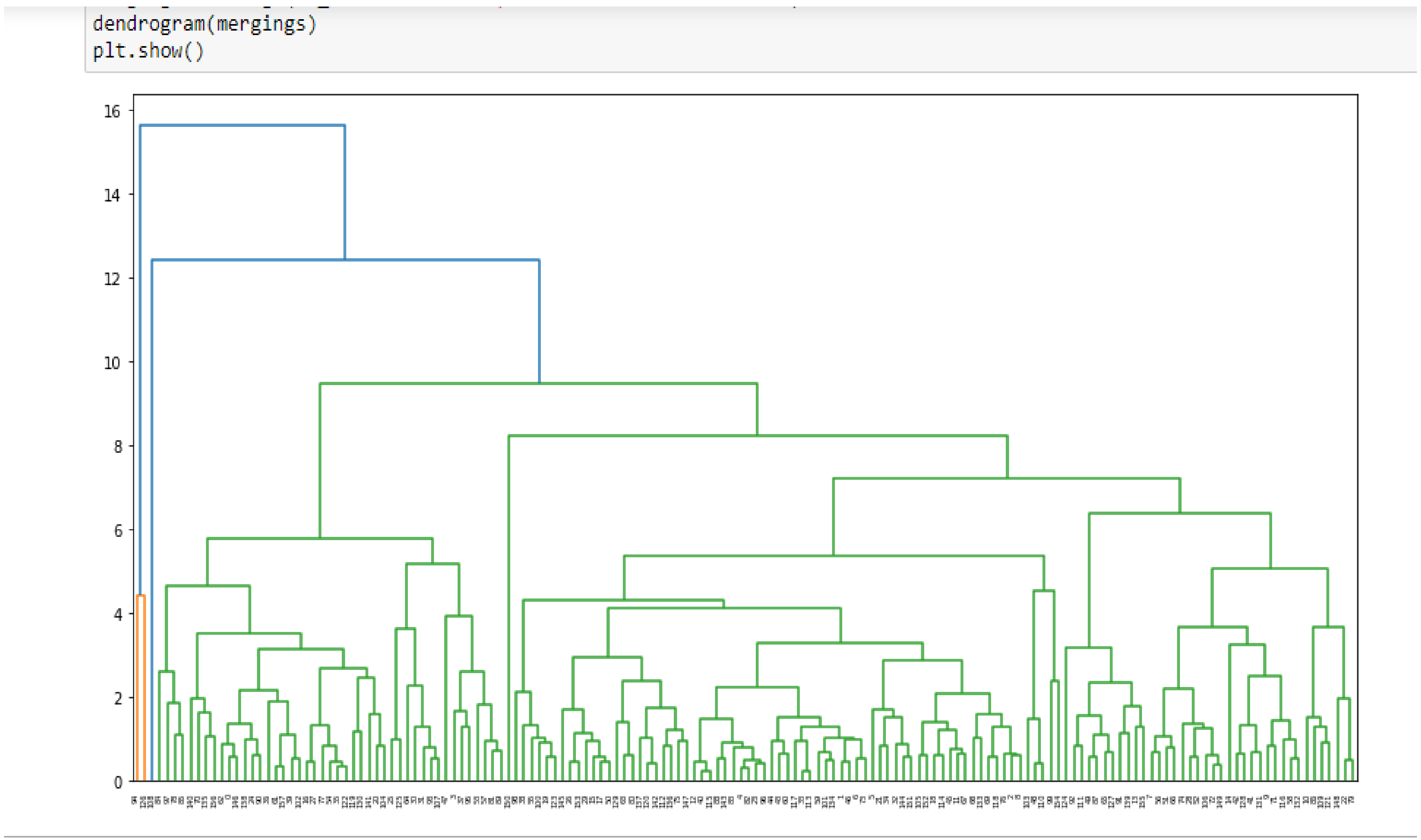


```
dendrogram(mergings)
plt.show()
```

Single Method Hierarchical Clustering

- By single method Hierarchical clustering the things are not clear.
- so now we go for complete method hierarchical clustering.

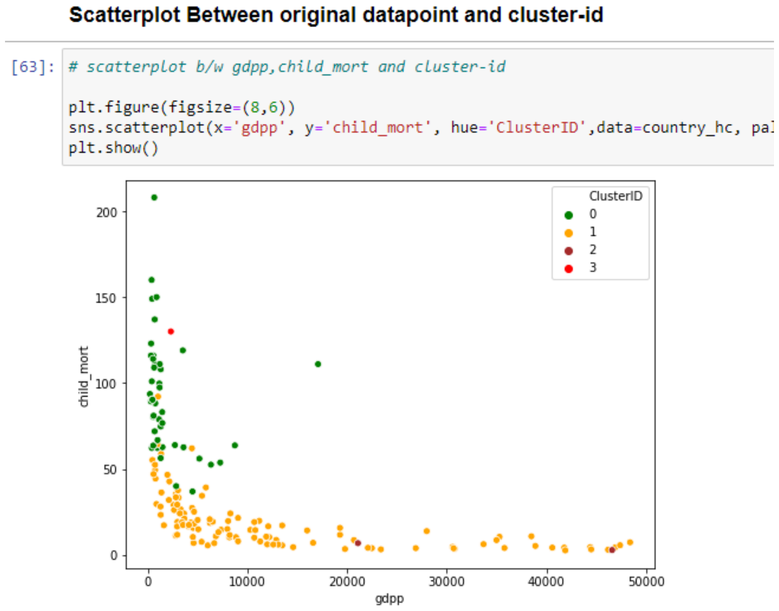
Hierarchical Clustering Cont…



```
dendrogram(mergings)
plt.show()
```

Complete Method Hierarchical Clustering

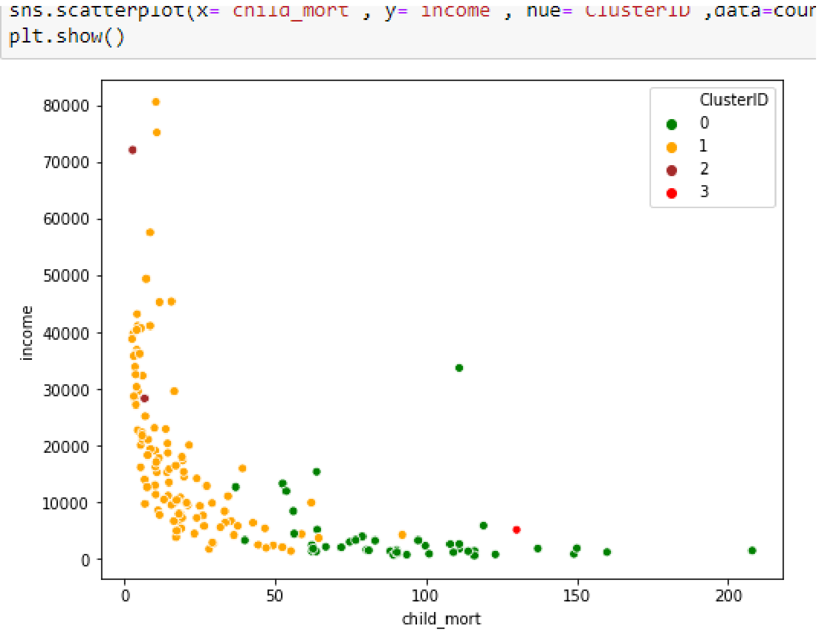
Hierarchical Clustering Cont…



Scatterplot Between original datapoint and cluster-id

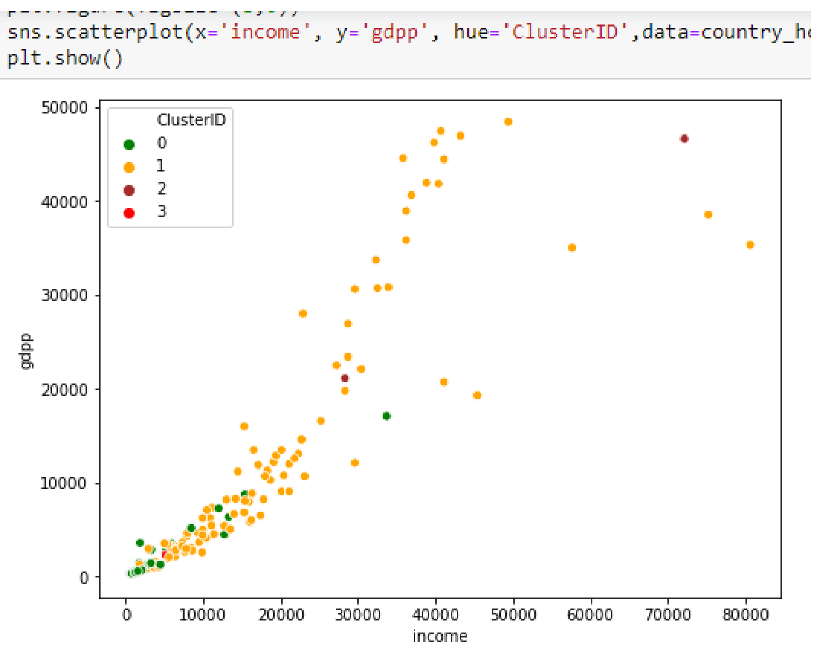
```
[63]: # scatterplot b/w gdpp,child_mort and cluster-id

      plt.figure(figsize=(8,6))
      sns.scatterplot(x='gdpp', y='child_mort', hue='ClusterID',data=country_hc, pa
      plt.show()
```

Scatterplot b/w gdpp, child_mort and ClusterID



```
sns.scatterplot(x='child_mort', y='income', hue='ClusterID',data=cour
plt.show()
```

Scatterplot b/w child_mort , income and ClusterID

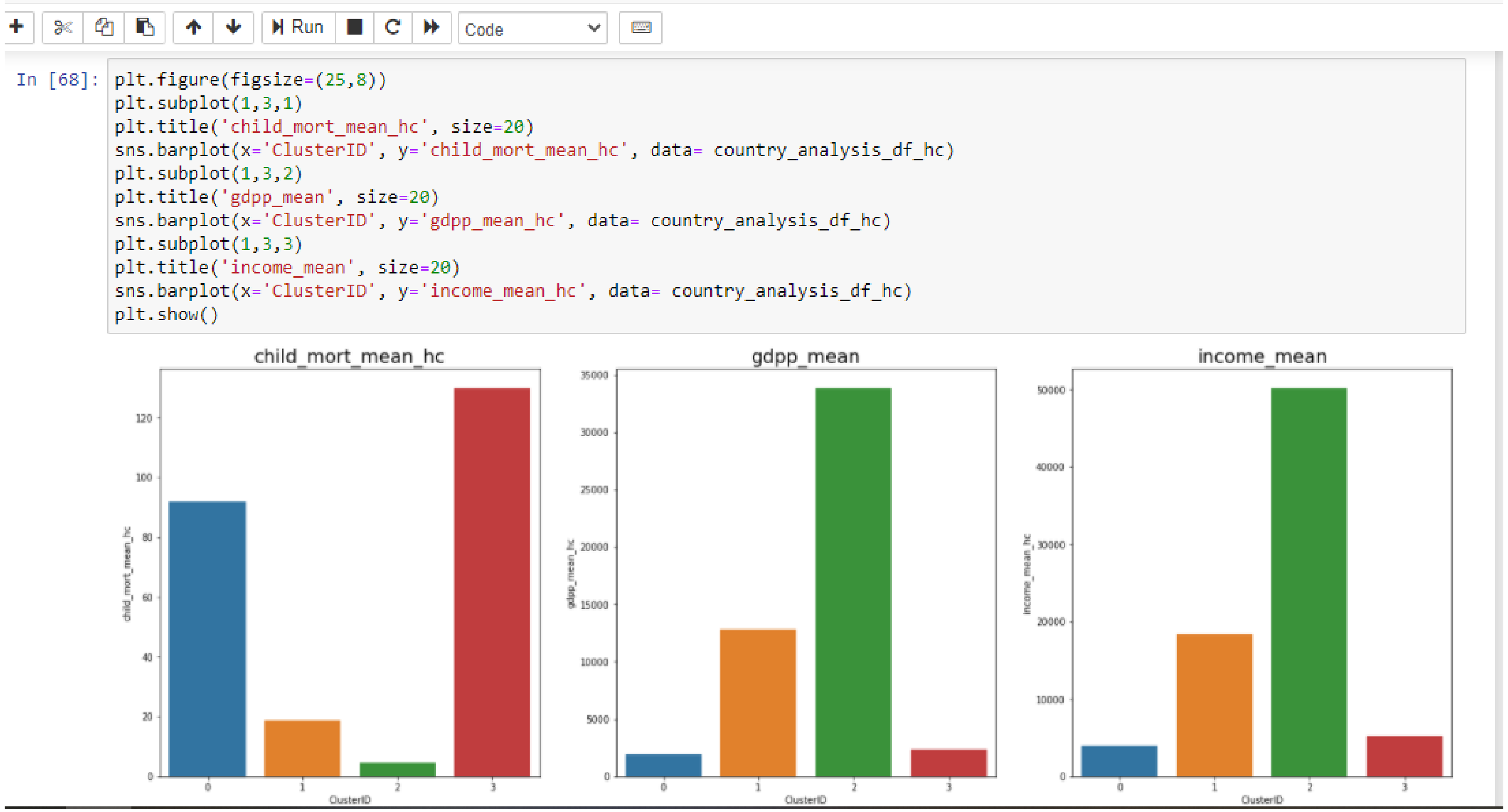


```
sns.scatterplot(x='income', y='gdpp', hue='ClusterID',data=country_h
plt.show()
```

Scatterplot b/w child_mort , income and ClusterID

Scatterplot Between original datapoint and cluster-id

Hierarchical Clustering Cont…



```
In [68]: plt.figure(figsize=(25,8))
         plt.subplot(1,3,1)
         plt.title('child_mort_mean_hc', size=20)
         sns.barplot(x='ClusterID', y='child_mort_mean_hc', data= country_analysis_df_hc)
         plt.subplot(1,3,2)
         plt.title('gdpp_mean', size=20)
         sns.barplot(x='ClusterID', y='gdpp_mean_hc', data= country_analysis_df_hc)
         plt.subplot(1,3,3)
         plt.title('income_mean', size=20)
         sns.barplot(x='ClusterID', y='income_mean_hc', data= country_analysis_df_hc)
         plt.show()
```

- By seeing above graph, we can conclude that here are two clusters that is 'cluster- 3' and 'cluster- 0' for our cluster of concern but we leave 'cluster- 3' as it has only 1 country and we use 'cluster- 0' because:-
  - ➢ Cluster-0 has highest child_mort.
  - ➢ Lowest gdpp.
  - ➢ And lowest income.

Analysing the Hierarchical-Clusters by using these three variables- 'gdpp', 'child_mort', 'income'

Hierarchical Clustering Cont…

As per Hierarchical clustering, the country which are direst need of aid are :

- Burundi
- Liberia
- Congo, Dem, Rep.
- Niger
- Sierra Leone
- Madagascar
- Mozambique
- Central African Republic
- Malawi
- Togo

Decision Making

We got same countries by both K-Means and Hierarchical clustering. Therefore following are the countries ehic are direst need of aid by considering socio-economic factor into consideration:-

- Burundi
- Liberia
- Congo, Dem, Rep.
- Niger
- Sierra Leone
- Madagascar
- Mozambique
- Central African Republic
- Malawi