DATA MINING PROJECT REPORT

DSBA

Krishnameera K S

Table of Contents

Part 1: PCA 2

1.1 Perform Exploratory Data Analysis [both univariate and multivariate analysis to be performed]. The inferences drawn from this should be properly documented 2

1.2 Scale the variables and write the inference for using the type of scaling function for this case study..... 6

1.3 Comment on the comparison between covariance and the correlation matrix after scaling..... 8

1.4 Check the dataset for outliers before and after scaling. Draw your inferences from this exercise. 10

1.5. Build the covariance matrix, eigenvalues and eigenvector. 11

1.6 Write the explicit form of the first PC (in terms of Eigen Vectors). 13

1.7 Discuss the cumulative values of the eigenvalues. How does it help you to decide on the optimum number of principal components? What do the eigenvectors indicate? Perform PCA and export the data of the Principal Component scores into a data frame. 14

Part 2: Clustering: 17

2.1 Clustering: Read the data and do exploratory data analysis. Describe the data briefly. (Check the null values, Data types, shape, EDA, etc)..... 17

2.2. Do you think scaling is necessary for clustering in this case? Justify..... 19

2.3. Apply hierarchical clustering to scaled data. Identify the number of optimum clusters using Dendrogram and briefly describe them. 21

2.4. Apply K-Means clustering on scaled data and determine optimum clusters. Apply elbow curve and find the silhouette score. 22

2.5 Describe cluster profiles for the clusters defined. Recommend different priority based actions that need to be taken for different clusters on the bases of their vulnerability situations according to their Economic and Health Conditions. 24

Part 1: PCA

Problem Statement: The 'Hair Salon.csv' dataset contains various variables used for the context of Market Segmentation. This particular case study is based on various parameters of a salon chain of hair products. You are expected to do Principal Component Analysis for this case study according to the instructions given in the rubric. Kindly refer to the PCA_Data_Dictionary.jpg file for the Data Dictionary of the Dataset. Note: This particular dataset contains the target variable satisfaction as well. Please do drop this variable before doing Principal Component Analysis.

1.1 Perform Exploratory Data Analysis [both univariate and multivariate analysis to be performed]. The inferences drawn from this should be properly documented.

The first 10 rows of the data

	ID	ProdQual	Ecom	TechSup	CompRes	Advertising	ProdLine	SalesFImage	ComPricing	WartyClaim	OrdBilling	DelSpeed	Satisfaction
0	1	8.5	3.9	2.5	5.9	4.8	4.9	6.0	6.8	4.7	5.0	3.7	8.2
1	2	8.2	2.7	5.1	7.2	3.4	7.9	3.1	5.3	5.5	3.9	4.9	5.7
2	3	9.2	3.4	5.6	5.6	5.4	7.4	5.8	4.5	6.2	5.4	4.5	8.9
3	4	6.4	3.3	7.0	3.7	4.7	4.7	4.5	8.8	7.0	4.3	3.0	4.8
4	5	9.0	3.4	5.2	4.6	2.2	6.0	4.5	6.8	6.1	4.5	3.5	7.1
5	6	6.5	2.8	3.1	4.1	4.0	4.3	3.7	8.5	5.1	3.6	3.3	4.7
6	7	6.9	3.7	5.0	2.6	2.1	2.3	5.4	8.9	4.8	2.1	2.0	5.7
7	8	6.2	3.3	3.9	4.8	4.6	3.6	5.1	6.9	5.4	4.3	3.7	6.3
8	9	5.8	3.6	5.1	6.7	3.7	5.9	5.8	9.3	5.9	4.4	4.6	7.0
9	10	6.4	4.5	5.1	6.1	4.7	5.7	5.7	8.4	5.4	4.1	4.4	5.5

The last 10 rows of the data

	ID	ProdQual	Ecom	TechSup	CompRes	Advertising	ProdLine	SalesFImage	ComPricing	WartyClaim	OrdBilling	DelSpeed	Satisfaction
90	91	9.1	3.7	7.0	4.1	4.4	6.3	5.4	7.3	7.5	4.4	3.3	7.4
91	92	7.1	4.2	4.1	2.6	2.1	3.3	4.5	9.9	5.5	2.0	2.4	4.8
92	93	9.2	3.9	4.6	5.3	4.2	8.4	4.8	7.1	6.2	4.4	4.2	7.6
93	94	9.3	3.5	5.4	7.8	4.6	7.5	5.9	4.6	6.4	4.8	4.6	8.9
94	95	9.3	3.8	4.0	4.6	4.7	6.4	5.5	7.4	5.3	3.6	3.4	7.7
95	96	8.6	4.8	5.6	5.3	2.3	6.0	5.7	6.7	5.8	4.9	3.6	7.3
96	97	7.4	3.4	2.6	5.0	4.1	4.4	4.8	7.2	4.5	4.2	3.7	6.3
97	98	8.7	3.2	3.3	3.2	3.1	6.1	2.9	5.6	5.0	3.1	2.5	5.4
98	99	7.8	4.9	5.8	5.3	5.2	5.3	7.1	7.9	6.0	4.3	3.9	6.4
99	100	7.9	3.0	4.4	5.1	5.9	4.2	4.8	9.7	5.7	3.4	3.5	6.4

Information of Data

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 13 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   ID            100 non-null    int64  
 1   ProdQual      100 non-null    float64
 2   Ecom          100 non-null    float64
 3   TechSup       100 non-null    float64
 4   CompRes       100 non-null    float64
 5   Advertising   100 non-null    float64
 6   ProdLine      100 non-null    float64
 7   SalesFImage   100 non-null    float64
 8   ComPricing    100 non-null    float64
 9   WartyClaim    100 non-null    float64
 10  OrdBilling    100 non-null    float64
 11  DelSpeed      100 non-null    float64
 12  Satisfaction  100 non-null    float64
dtypes: float64(12), int64(1)
memory usage: 10.3 KB
```

There are 100 rows and 13 columns in the data. Under 13 variables, “Satisfaction” is dependent variable and following 11 are independent variables.

```
(100, 13)
```

Expansion of variables:

ProdQual = Product Quality

Ecom = E-Commerce

TechSup = Technical Support

CompRes = Complaint Resolution

Advertising = Advertising

ProdLine = Product Line

SalesFImage = Salseforce Image

ComPricing = Competitive Pricing

WartyClaim = Warranty & Claims

OrdBilling = Order Billing

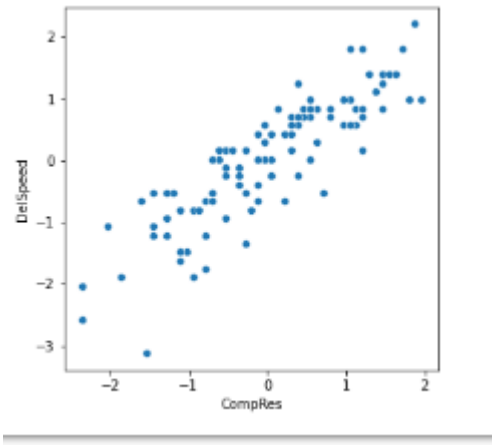
DelSpeed = Delivery Speed

Satisfaction = Customer Satisfaction

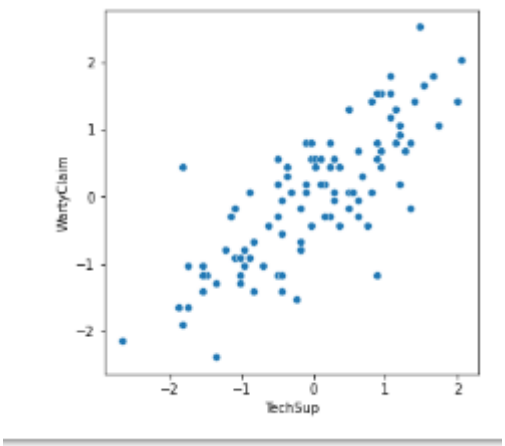
There are no duplicates and there is no missing values

After dropping categorical variables: (ID and Satisfaction columns)

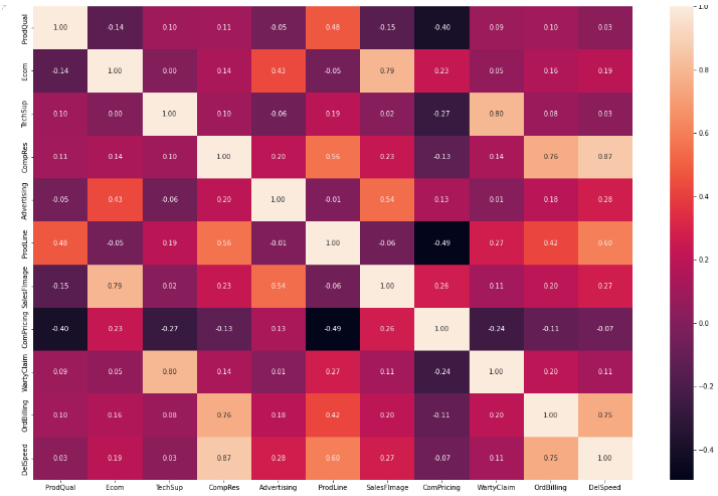
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 11 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   ProdQual     100 non-null    float64
 1   Ecom         100 non-null    float64
 2   TechSup      100 non-null    float64
 3   CompRes      100 non-null    float64
 4   Advertising  100 non-null    float64
 5   ProdLine     100 non-null    float64
 6   SalesFImage  100 non-null    float64
 7   ComPricing   100 non-null    float64
 8   WartyClaim   100 non-null    float64
 9   OrdBilling   100 non-null    float64
 10  DelSpeed     100 non-null    float64
dtypes: float64(11)
memory usage: 8.7 KB
```



Delivery speed and Complaint resolution is positively correlated



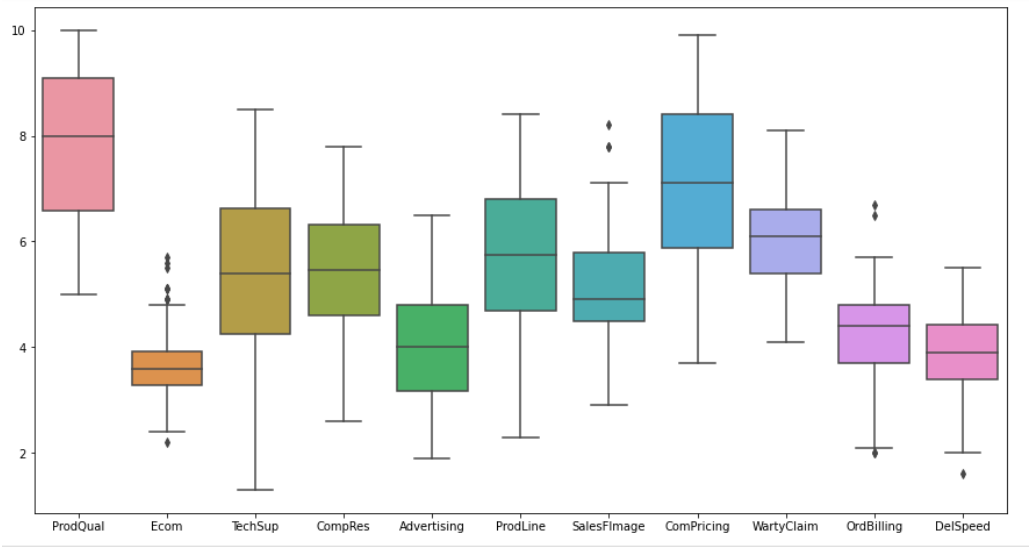
Technical support and warrant claimed is positively correlated



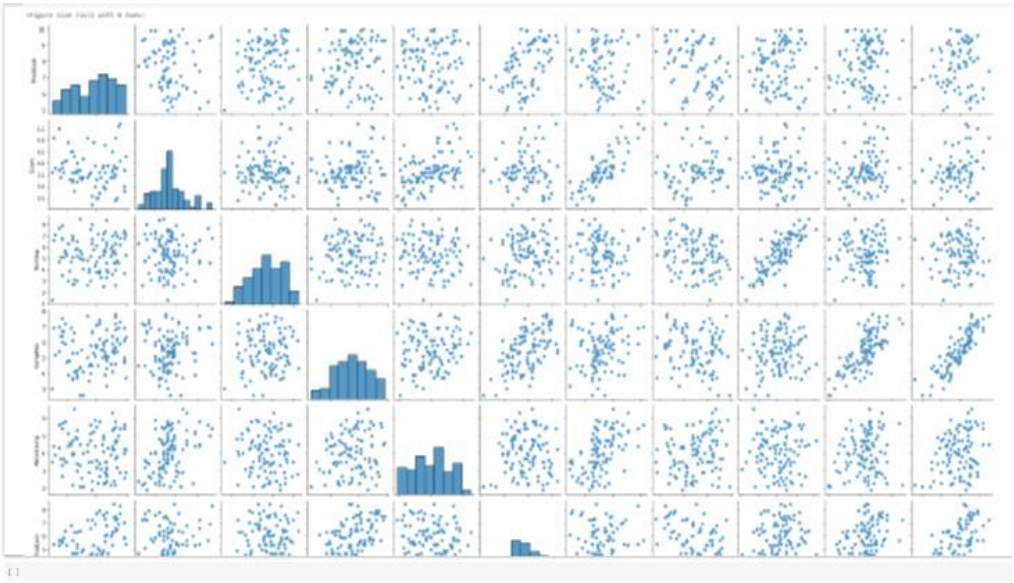
From Heatmap we are able to analyse that Product Line and Competitive pricing is inversely proportional.

Using the formulas provided, treated the missing values and there are no null values currently in the data.

Box Plot of Hair salon Data before scaling:



Pair plot of Hair salon Data:



1.2 Scale the variables and write the inference for using the type of scaling function for this case study.

Scaling is very important in clustering, and we can observe the data mean, min, max std to understand the difference before the scaling and after the scaling is done. It is easier and faster to perform clustering after scaling the data. Hence, used z score scaling to scale the data.

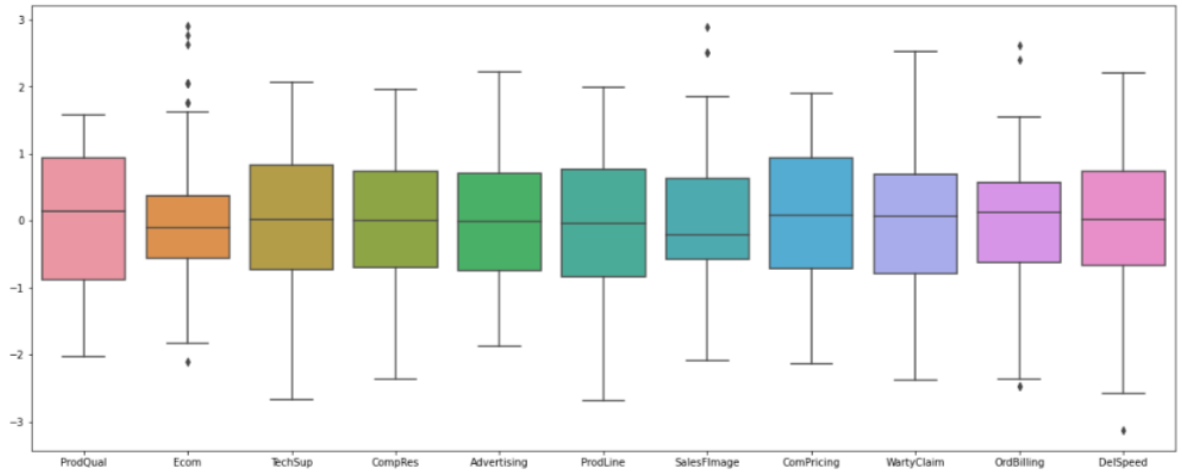
Scaling is important because larger variances will not dominate the analysis over variables with smaller variances.

Scale the Data using z-score method.

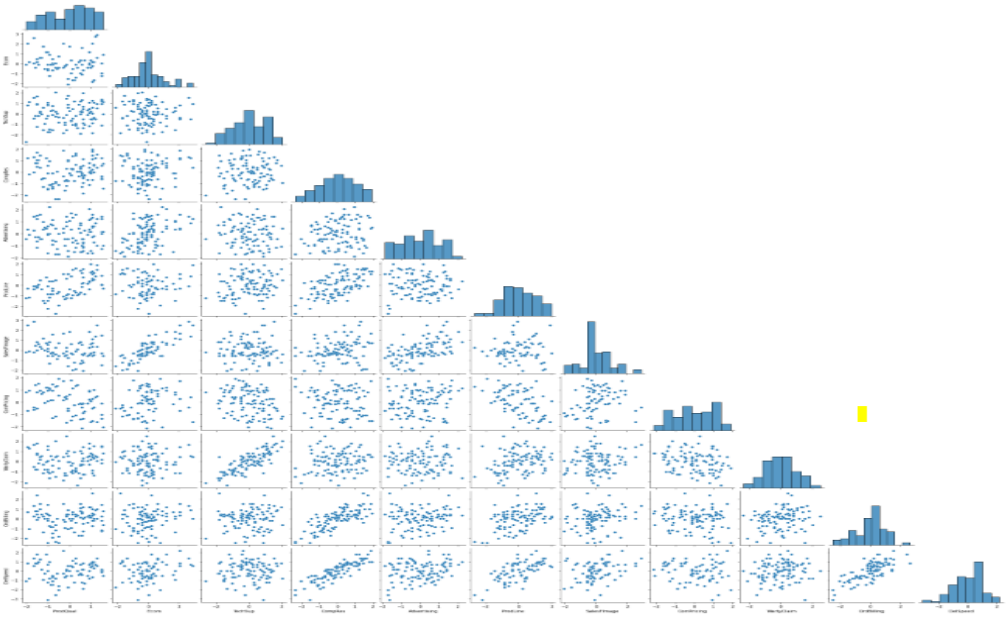
After scaling the Variables

	ProdQual	Ecom	TechSup	CompRes	Advertising	ProdLine	SalesFImage	ComPricing	WartyClaim	OrdBilling	DelSpeed
0	0.496660	0.327114	-1.881421	0.380922	0.704543	-0.691530	0.821973	-0.113185	-1.646582	0.781230	-0.254531
1	0.280721	-1.394538	-0.174023	1.462141	-0.544014	1.600835	-1.896068	-1.088915	-0.665744	-0.409009	1.387605
2	1.000518	-0.390241	0.154322	0.131410	1.239639	1.218774	0.634522	-1.609304	0.192489	1.214044	0.840226
3	-1.014914	-0.533712	1.073690	-1.448834	0.615361	-0.844354	-0.583910	1.187789	1.173327	0.023805	-1.212443
4	0.856559	-0.390241	-0.108354	-0.700298	-1.614207	0.149004	-0.583910	-0.113185	0.069885	0.240212	-0.528220

Boxplot after Scaling



Pairplot after Scaling

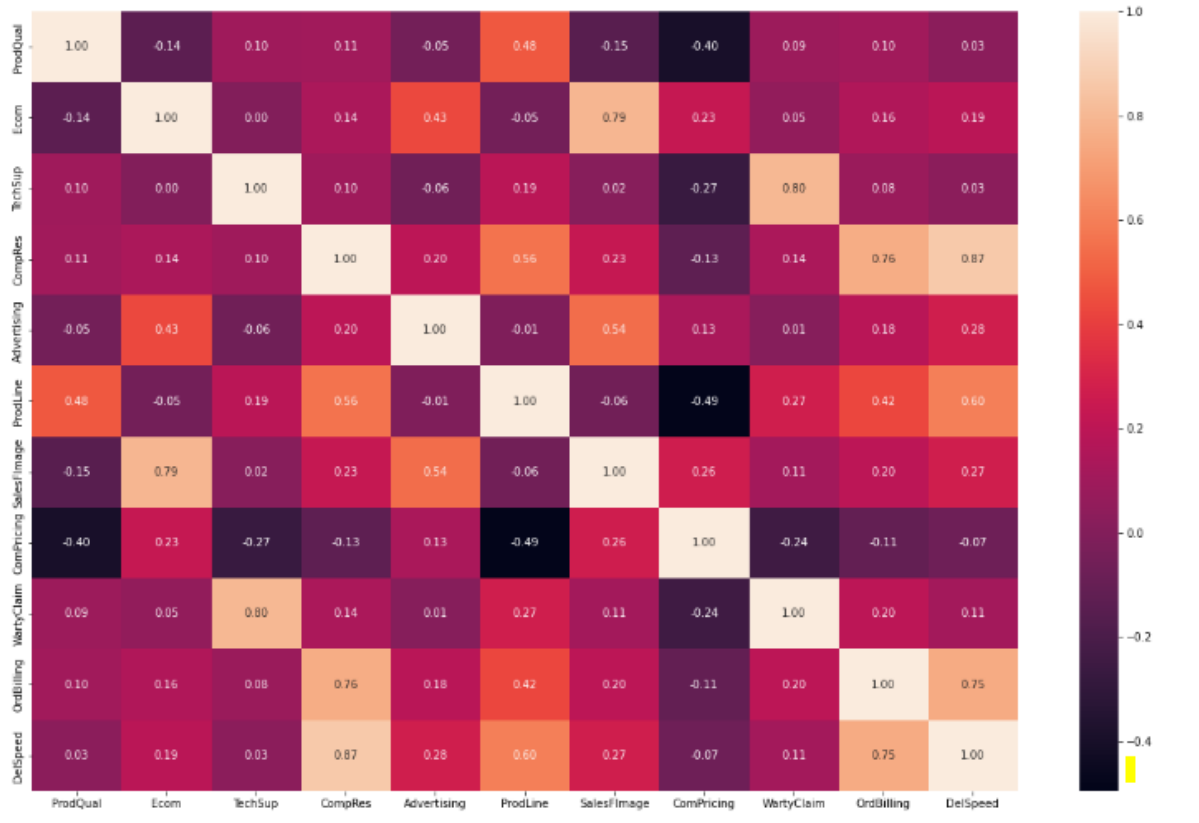


1.3 Comment on the comparison between covariance and the correlation matrix after scaling.

Correlation matrix measures the degree to which two variables are linearly related to each other.

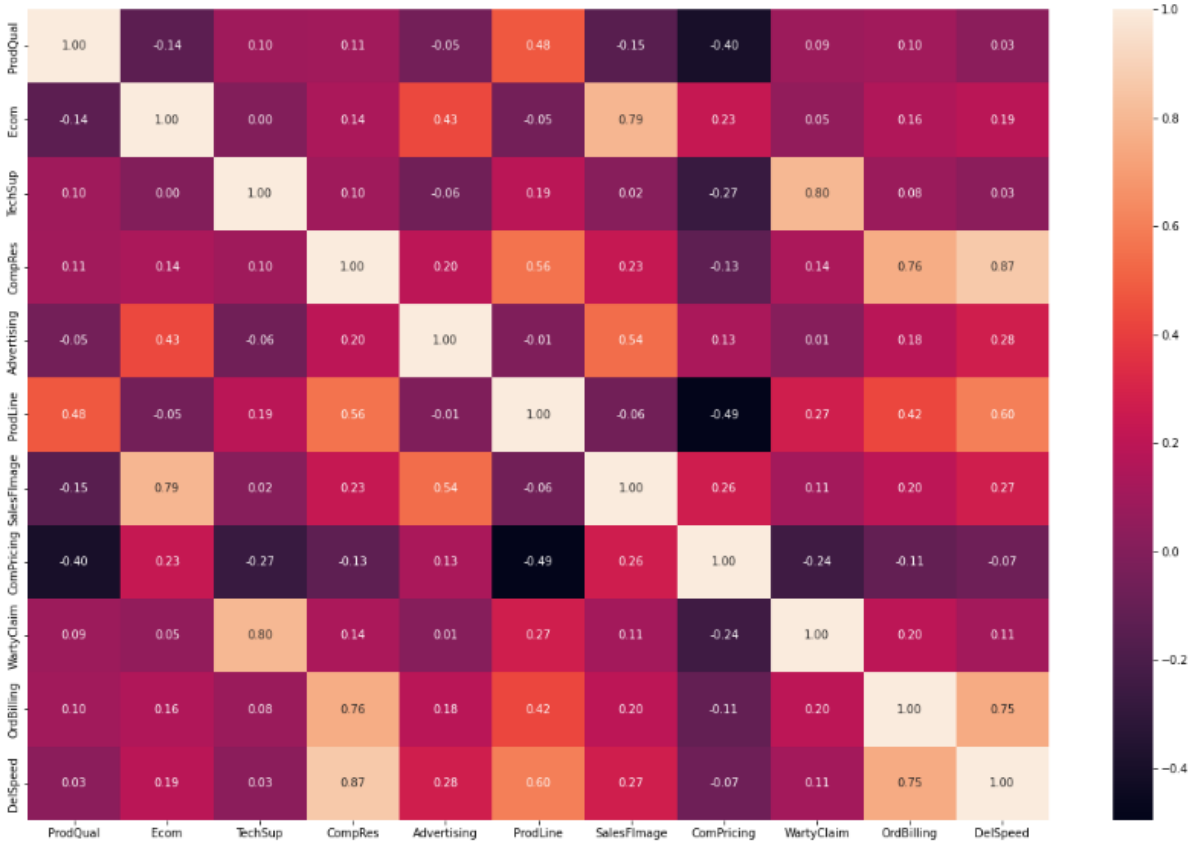
<u>Correlation between the variables before the scaling</u>

	ProdQual	Ecom	TechSup	CompRes	Advertising	ProdLine	SalesFImage	ComPricing	WartyClaim	OrdBilling	DelSpeed
ProdQual	1.000000	-0.161588	0.095600	0.106370	-0.053473	0.477493	-0.146498	-0.401282	0.088312	0.102495	0.024332
Ecom	-0.161588	1.000000	-0.018786	0.109386	0.425123	-0.096342	0.779244	0.268064	0.027380	0.146505	0.168147
TechSup	0.095600	-0.018786	1.000000	0.096657	-0.062870	0.192625	0.009836	-0.270787	0.797168	0.085443	0.028898
CompRes	0.106370	0.109386	0.096657	1.000000	0.196917	0.561417	0.226647	-0.127954	0.140408	0.757995	0.868846
Advertising	-0.053473	0.425123	-0.062870	0.196917	1.000000	-0.011551	0.542923	0.134217	0.010792	0.188005	0.272973
ProdLine	0.477493	-0.096342	0.192625	0.561417	-0.011551	1.000000	-0.062584	-0.494948	0.273078	0.423870	0.600272
SalesFImage	-0.146498	0.779244	0.009836	0.226647	0.542923	-0.062584	1.000000	0.271246	0.100953	0.194695	0.271213
ComPricing	-0.401282	0.268064	-0.270787	-0.127954	0.134217	-0.494948	0.271246	1.000000	-0.244986	-0.113318	-0.070289
WartyClaim	0.088312	0.027380	0.797168	0.140408	0.010792	0.273078	0.100953	-0.244986	1.000000	0.198106	0.116168
OrdBilling	0.102495	0.146505	0.085443	0.757995	0.188005	0.423870	0.194695	-0.113318	0.198106	1.000000	0.752298
DelSpeed	0.024332	0.168147	0.028898	0.868846	0.272973	0.600272	0.271213	-0.070289	0.116168	0.752298	1.000000



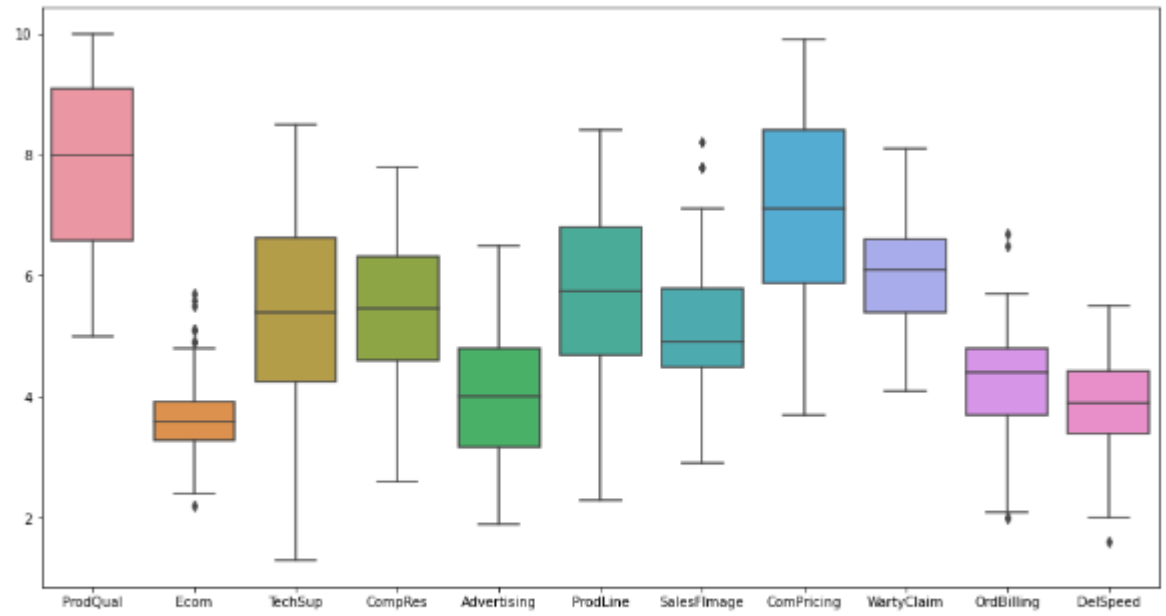
Correlation between the variables after the scaling

	ProdQual	Ecom	TechSup	CompRes	Advertising	ProdLine	SalesFImage	ComPricing	WartyClaim	OrdBilling	DelSpeed
ProdQual	1.000000	-0.137163	0.095600	0.106370	-0.053473	0.477493	-0.151813	-0.401282	0.088312	0.104303	0.027718
Ecom	-0.137163	1.000000	0.000867	0.140179	0.429891	-0.052688	0.791544	0.229462	0.051898	0.156147	0.191636
TechSup	0.095600	0.000867	1.000000	0.096657	-0.062870	0.192625	0.016991	-0.270787	0.797168	0.080102	0.025441
CompRes	0.106370	0.140179	0.096657	1.000000	0.196917	0.561417	0.229752	-0.127954	0.140408	0.756869	0.865092
Advertising	-0.053473	0.429891	-0.062870	0.196917	1.000000	-0.011551	0.542204	0.134217	0.010792	0.184236	0.275863
ProdLine	0.477493	-0.052688	0.192625	0.561417	-0.011551	1.000000	-0.061316	-0.494948	0.273078	0.424408	0.601850
SalesFImage	-0.151813	0.791544	0.016991	0.229752	0.542204	-0.061316	1.000000	0.264597	0.107455	0.195127	0.271551
ComPricing	-0.401282	0.229462	-0.270787	-0.127954	0.134217	-0.494948	0.264597	1.000000	-0.244986	-0.114567	-0.072872
WartyClaim	0.088312	0.051898	0.797168	0.140408	0.010792	0.273078	0.107455	-0.244986	1.000000	0.197065	0.109395
OrdBilling	0.104303	0.156147	0.080102	0.756869	0.184236	0.424408	0.195127	-0.114567	0.197065	1.000000	0.751003
DelSpeed	0.027718	0.191636	0.025441	0.865092	0.275863	0.601850	0.271551	-0.072872	0.109395	0.751003	1.000000

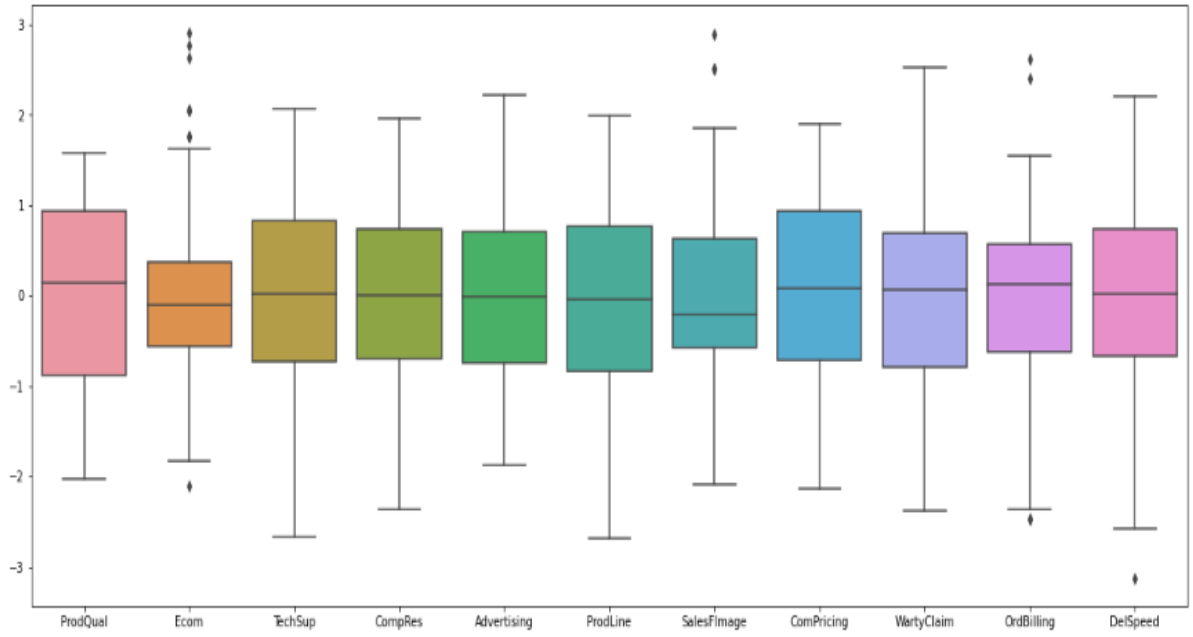


1.4 Check the dataset for outliers before and after scaling. Draw your inferences from this exercise.

Before scaling dataset for outliers, Output is given below



After scaling dataset for outliers, below is the output.



1.5. Build the covariance matrix, eigenvalues and eigenvector.

Bartletts Test of Sphericity Bartlett's test of sphericity tests the hypothesis that the variables are uncorrelated in the population. If the null hypothesis cannot be rejected, then PCA is not advisable.

$\boldsymbol{H0}$: All variables in the data are uncorrelated

$\boldsymbol{H1}$: At least one pair of variables in the data are correlated Inference:

```
P-Value is 1.793370009363654e-96
```

Since p-value: 0.000000, we reject the null hypothesis is rejected.

KMO Test

The Kaiser-Meyer-Olkin (KMO) - measure of sampling adequacy (MSA) is an index used to examine how appropriate PCA is. Generally, if MSA is less than 0.5, PCA is not recommended, since no reduction is expected. On the other hand, MSA > 0.7 is expected to provide a considerable reduction is the dimension and extraction of meaningful components.

```
MSA is 0.6531422230688922
```

Step 1- Create the covariance Matrix Covariance Matrix

Below is the covariance matrix

```
array([[ 0.53581067, -0.03206695,  0.33199836, ..., -0.10256236,
         0.15565905,  0.24606466],
       [-0.03206695,  1.52499975,  0.30034491, ...,  0.43658831,
        -0.83920064, -0.34955129],
       [ 0.33199836,  0.30034491,  0.97112912, ..., -0.30144808,
         0.0986921 , -0.41334004],
       ...,
       [-0.10256236,  0.43658831, -0.30144808, ...,  1.9800132 ,
        -0.80825161,  0.23444317],
       [ 0.15565905, -0.83920064,  0.0986921 , ..., -0.80825161,
         0.8837573 ,  0.09352861],
       [ 0.24606466, -0.34955129, -0.41334004, ...,  0.23444317,
         0.09352861,  1.14891955]])
```

Step 2- Get eigen values and eigen vector

Eigen Vector

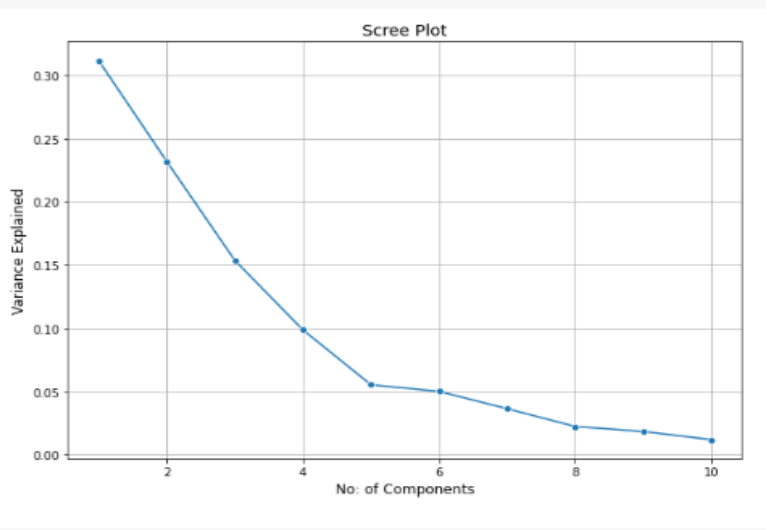
```
array([[-0.13378962, -0.16595278, -0.15769263, -0.47068359, -0.18373495,
        -0.38676517, -0.2036696 ,  0.15168864, -0.21293363, -0.43721774,
        -0.47308914],
       [-0.31349802,  0.44650918, -0.23096734,  0.01944394,  0.36366471,
        -0.28478056,  0.47069599,  0.4134565 , -0.19167191,  0.02639905,
         0.07305172],
       [ 0.06227164, -0.23524791, -0.61095105,  0.21035078, -0.08809705,
         0.11627864, -0.2413421 ,  0.05304529, -0.59856398,  0.16892981,
         0.23262477],
       [ 0.6431362 ,  0.27238033, -0.19339314, -0.20632037,  0.31789448,
         0.20290226,  0.22217722, -0.33354348, -0.18530205, -0.23685365,
        -0.1973299 ],
       [ 0.2316662 ,  0.42228844, -0.02395667,  0.02865743, -0.80387024,
         0.11667416,  0.20437283,  0.24892601, -0.03292706,  0.02675377,
        -0.03543294],
       [-0.56456996,  0.26325703, -0.10876896, -0.02815231, -0.20056937,
         0.09819533,  0.10497225, -0.70973595, -0.13983966, -0.11947974,
         0.02979992],
       [ 0.19164132,  0.05962621, -0.01719992, -0.0084996 , -0.06306962,
        -0.60814755,  0.00143735, -0.30824887, -0.03064024,  0.65931989,
        -0.23423927],
       [ 0.13547311, -0.12202642,  0.46470964,  0.51339754, -0.05347713,
        -0.3332071 ,  0.16910665, -0.09883227, -0.4435404 , -0.36601754,
         0.06539059],
       [ 0.0313281 , -0.54251104, -0.35929961,  0.09324751, -0.15468169,
        -0.08415534,  0.64489911, -0.09414389,  0.31756604, -0.09907265,
        -0.02188514],
       [ 0.06659717,  0.28155772, -0.3881709 ,  0.53467243,  0.03715799,
        -0.23479794, -0.35341191, -0.04518224,  0.43534752, -0.30386545,
        -0.12010386]])
```

Eigen Values

```
array([3.4615872 , 2.57666335, 1.70805705, 1.09753137, 0.61557989,
       0.55745836, 0.40557389, 0.249446  , 0.20560936, 0.13418341])
```

1.6 Write the explicit form of the first PC (in terms of Eigen Vectors).

Optimum number of PCs and scree plot has been shown below.



From the above graph and cumulative explained variance, 6 PCs are chosen

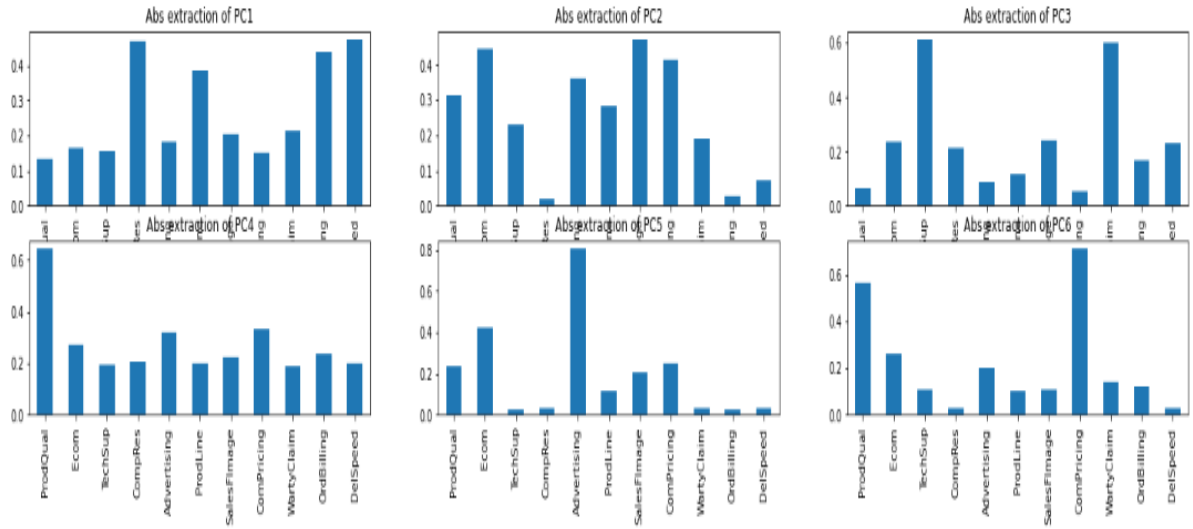
```
array([0.31154285, 0.54344255, 0.69716768, 0.79594551, 0.8513477 ,
       0.90151895, 0.9380206 , 0.96047074, 0.97897558, 0.99105209])
```

	PC1	PC2	PC3	PC4	PC5	PC6
ProdQual	-0.133790	-0.313498	0.062272	0.643136	0.231666	-0.564570
Ecom	-0.165953	0.446509	-0.235248	0.272380	0.422288	0.263257
TechSup	-0.157693	-0.230967	-0.610951	-0.193393	-0.023957	-0.108769
CompRes	-0.470684	0.019444	0.210351	-0.206320	0.028657	-0.028152
Advertising	-0.183735	0.363665	-0.088097	0.317894	-0.803870	-0.200569
ProdLine	-0.386765	-0.284781	0.116279	0.202902	0.116674	0.098195
SalesFImage	-0.203670	0.470696	-0.241342	0.222177	0.204373	0.104972
ComPricing	0.151689	0.413457	0.053045	-0.333543	0.248926	-0.709736
WartyClaim	-0.212934	-0.191672	-0.598564	-0.185302	-0.032927	-0.139840
OrdBilling	-0.437218	0.026399	0.168930	-0.236854	0.026754	-0.119480
DelSpeed	-0.473089	0.073052	0.232625	-0.197330	-0.035433	0.029800

1.7 Discuss the cumulative values of the eigenvalues. How does it help you to decide on the optimum number of principal components? What do the eigenvectors indicate? Perform PCA and export the data of the Principal Component scores into a data frame.

Correlation between PCs and original Varible

	PC1	PC2	PC3	PC4	PC5	PC6
ProdQual	-0.133790	-0.313498	0.062272	0.643136	0.231666	-0.564570
Ecom	-0.165953	0.446509	-0.235248	0.272380	0.422288	0.263257
TechSup	-0.157693	-0.230967	-0.610951	-0.193393	-0.023957	-0.108769
CompRes	-0.470684	0.019444	0.210351	-0.206320	0.028657	-0.028152
Advertising	-0.183735	0.363665	-0.088097	0.317894	-0.803870	-0.200569
ProdLine	-0.386765	-0.284781	0.116279	0.202902	0.116674	0.098195
SalesFImage	-0.203670	0.470696	-0.241342	0.222177	0.204373	0.104972
ComPricing	0.151689	0.413457	0.053045	-0.333543	0.248926	-0.709736
WartyClaim	-0.212934	-0.191672	-0.598564	-0.185302	-0.032927	-0.139840
OrdBilling	-0.437218	0.026399	0.168930	-0.236854	0.026754	-0.119480
DelSpeed	-0.473089	0.073052	0.232625	-0.197330	-0.035433	0.029800



	PC1	PC2	PC3	PC4	PC5	PC6
0	0.079551	1.543198	1.895046	1.168119	-0.113909	0.086364
1	-1.100966	-2.420298	2.045521	-0.427083	-0.550453	0.475588
2	-2.197067	-0.727440	0.166800	1.310312	-1.061797	0.244819
3	1.562933	0.171366	-1.827179	-1.192240	-0.939629	-0.957207
4	0.767570	-1.428111	0.234356	0.069525	1.206498	-0.251606
5	2.908622	0.309387	1.532706	-0.746605	-0.848741	-0.334014
6	5.293191	1.057481	-0.644861	0.028470	1.303547	0.107495
7	1.476591	1.111083	0.705905	-0.567127	-1.086442	0.494166
8	-0.613948	1.379473	0.575067	-1.769264	0.367067	-0.127711
9	-0.423660	1.981541	0.336888	-0.365410	0.119662	0.310581

Linear equation of first PC

	PC1	PC2	PC3	PC4	PC5	PC6
ProdQual	-0.133790	-0.313498	0.062272	0.643136	0.231666	-0.564570
Ecom	-0.165953	0.446509	-0.235248	0.272380	0.422288	0.263257
TechSup	-0.157693	-0.230967	-0.610951	-0.193393	-0.023957	-0.108769
CompRes	-0.470684	0.019444	0.210351	-0.206320	0.028657	-0.028152
Advertising	-0.183735	0.363665	-0.088097	0.317894	-0.803870	-0.200569
ProdLine	-0.386765	-0.284781	0.116279	0.202902	0.116674	0.098195
SalesFImage	-0.203670	0.470696	-0.241342	0.222177	0.204373	0.104972
ComPricing	0.151689	0.413457	0.053045	-0.333543	0.248926	-0.709736
WartyClaim	-0.212934	-0.191672	-0.598564	-0.185302	-0.032927	-0.139840
OrdBilling	-0.437218	0.026399	0.168930	-0.236854	0.026754	-0.119480
DelSpeed	-0.473089	0.073052	0.232625	-0.197330	-0.035433	0.029800

(-0.13) * ProdQual + (-0.17) * Ecom + (-0.16) * TechSup + (-0.47) * CompRes + (-0.18) * Advertising + (-0.39) * ProdLine

(-0.2) * SalesFImage + (0.15) * ComPricing + (-0.21) * WartyClaim + (-0.44) * OrdBilling + (-0.47) * DelSpeed +

Part 2: Clustering:

The State_wise_Health_income.csv dataset given is about the Health and economic conditions in different States of a country. The Group States based on how similar their situation is, so as to provide these groups to the government so that appropriate measures can be taken to escalate their Health and Economic conditions.

2.1 Clustering: Read the data and do exploratory data analysis. Describe the data briefly. (Check the null values, Data types, shape, EDA, etc)

Head of the Data:

	Unnamed: 0	States	Health_indeces1	Health_indices2	Per_capita_income	GDP
0	0	Bachevo	417	66	564	1823
1	1	Balgarchevo	1485	646	2710	73662
2	2	Belasitsa	654	299	1104	27318
3	3	Belo_Pole	192	25	573	250
4	4	Beslen	43	8	528	22
5	5	Bogolin	69	14	527	73
6	6	Bogoroditsa	307	69	707	1724
7	7	Buchino	10219	1508	7049	449003
8	8	Budiltsi	744	115	809	7497
9	9	Cherniche	2975	857	1600	153299

Information of data:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 297 entries, 0 to 296
Data columns (total 6 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Unnamed: 0         297 non-null    int64
 1   States             297 non-null    object
 2   Health_indeces1    297 non-null    int64
 3   Health_indices2    297 non-null    int64
 4   Per_capita_income  297 non-null    int64
 5   GDP                297 non-null    int64
dtypes: int64(5), object(1)
memory usage: 14.0+ KB
```

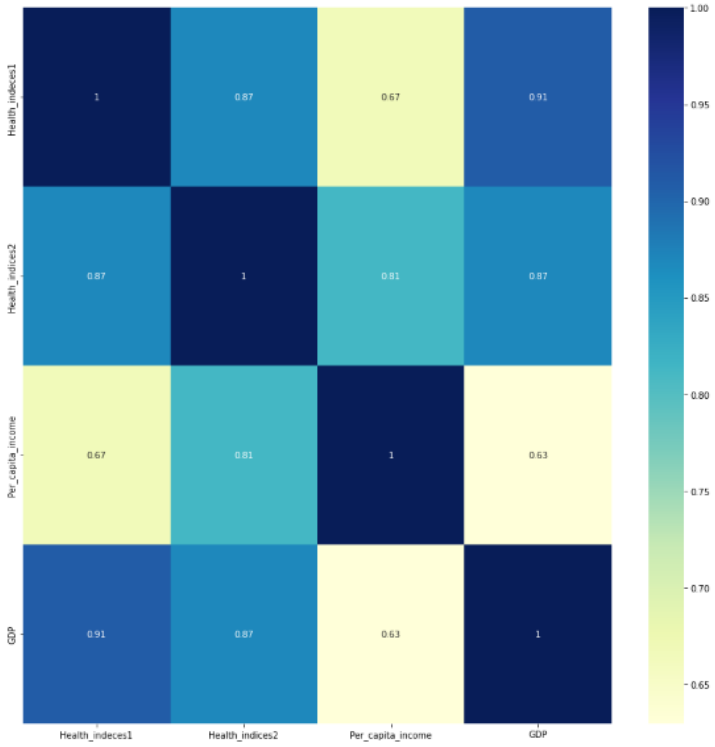
There is no missing values and duplicates.

```
Health_indeces1      0
Health_indices2      0
Per_capita_income    0
GDP                  0
dtype: int64
```

Shape of Data

```
(297, 4)
```

Heatmap of data:

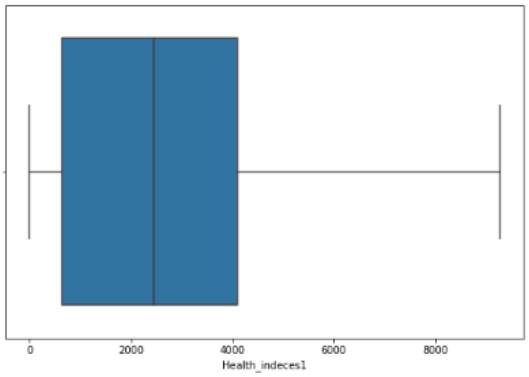


2.2. Do you think scaling is necessary for clustering in this case? Justify

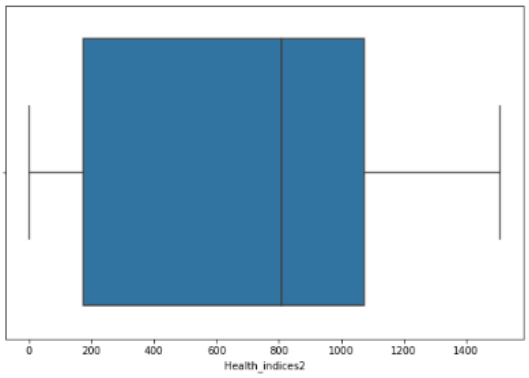
First we have to treat the outliers before K-clustering.

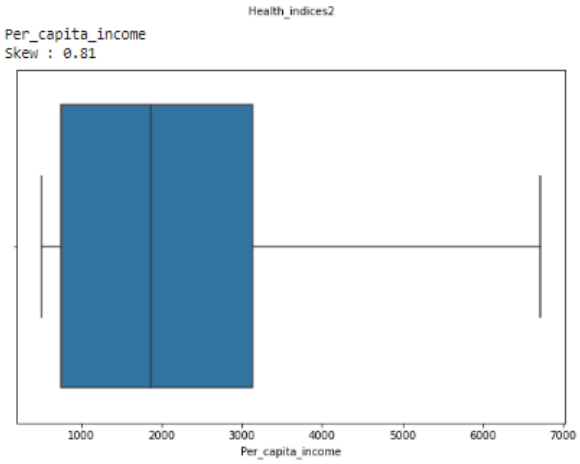
After removing the Outliers

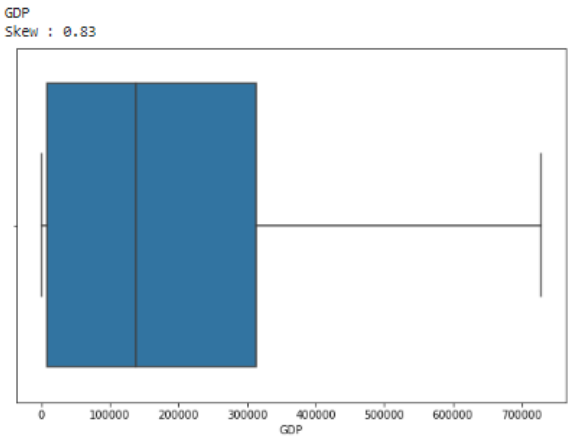
Health_indeces1
Skew : 0.67



Health_indices2
Skew : -0.17







We can treat outliers using IQR method and also z-score method.

After scaling using Z-score method

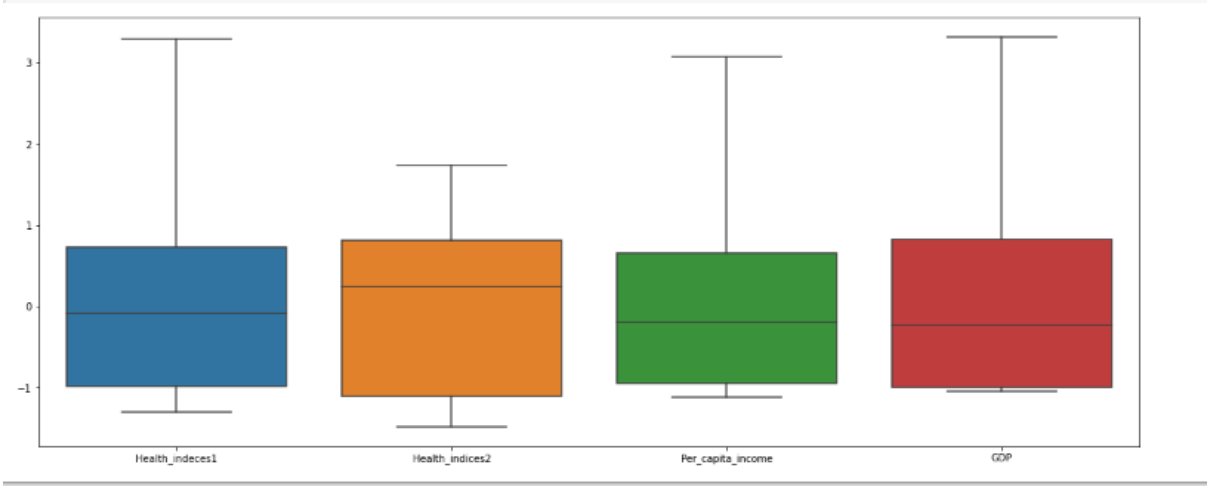
Head of the data

	Health_indeces1	Health_indices2	Per_capita_income	GDP
0	-1.092498	-1.340654	-1.071354	-1.035304
1	-0.564428	-0.101746	0.373007	-0.604838
2	-0.975314	-0.842955	-0.707908	-0.882536
3	-1.203748	-1.428232	-1.065297	-1.044730
4	-1.277421	-1.464545	-1.095584	-1.046096

Summary of data

	Health_indeces1	Health_indices2	Per_capita_income	GDP
count	2.970000e+02	297.000000	2.970000e+02	2.970000e+02
mean	-4.784800e-17	0.000000	-9.569599e-17	4.784800e-17
std	1.001688e+00	1.001688	1.001688e+00	1.001688e+00
min	-1.303627e+00	-1.481634	-1.114429e+00	-1.046096e+00
25%	-9.817414e-01	-1.107825	-9.454942e-01	-9.939707e-01
50%	-8.679136e-02	0.248566	-1.957187e-01	-2.242731e-01
75%	7.255859e-01	0.810346	6.603983e-01	8.298516e-01
max	3.286577e+00	1.739527	3.069237e+00	3.319468e+00

Boxplot after scaling

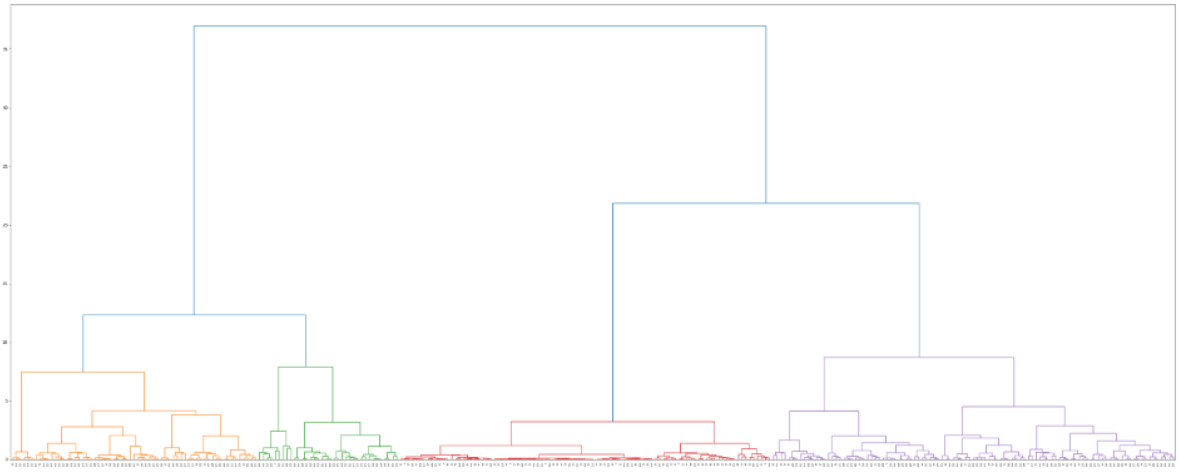


2.3. Apply hierarchical clustering to scaled data. Identify the number of optimum clusters using Dendrogram and briefly describe them.

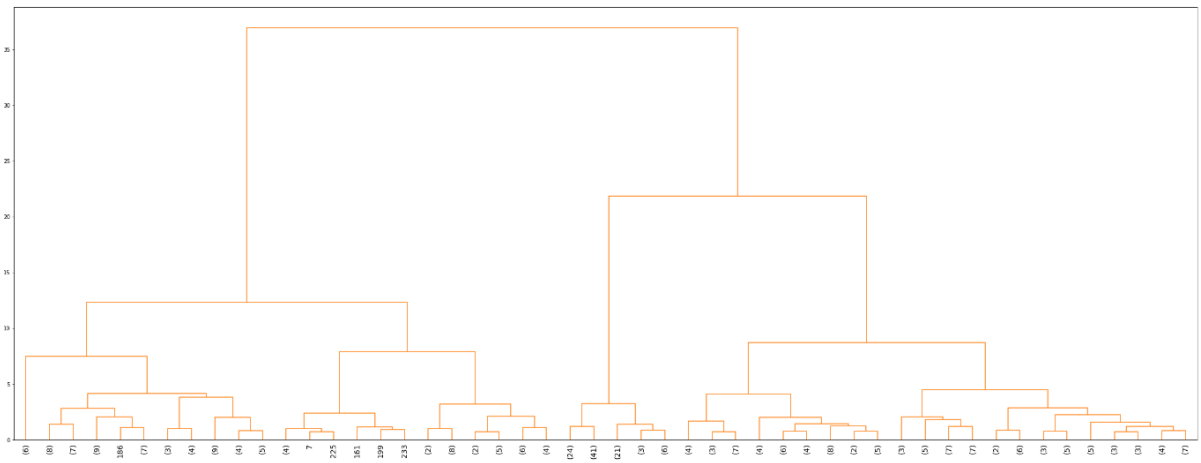
Dendrogram without truncating

Each branch in Dendogram is called clade. The terminal end of each clade is called a leaf. The arrangement of the clades tells us which leaves are most similar to each other. The height of the branching points indicates how similar or different they are from each other: the greater the height, the greater the difference.

If we draw a line through blue vertical line we get 3 clusters.



This data is not readable for the customers in the business, hence we used truncate to get a better view of the dendrogram.



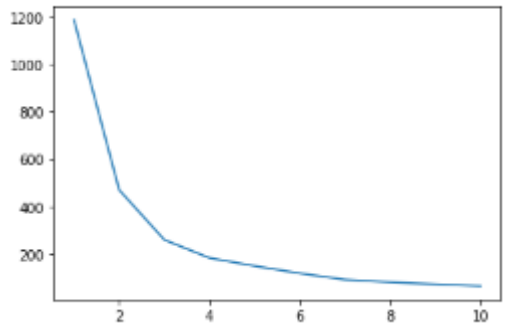
2.4. Apply K-Means clustering on scaled data and determine optimum clusters. Apply elbow curve and find the silhouette score.

K-Mean

```
[1188.0,
 469.3294729619331,
 258.4467041090695,
 181.7379852621384,
 147.72959194996764,
 116.68393632715788,
 90.00364346654787,
 79.06119171118948,
 70.64176638319206,
 62.99133600587566]
```

Elbow Plot:

```
[<matplotlib.lines.Line2D at 0x7f8ff6d017c0>]
```



Above is the elbow plot created for clusters upto 10. 3 is the optimum number of clusters for kmeans algorithm. We could see 2 to 3 drop is high , 3 to 4 is low.

Silhouette scores for up to 10 clusters and identify optimum number of clusters.

	Health_indeces1	Health_indices2	Per_capita_income	GDP	kmeans	sil_width	Clus_kmeans_	KMEANS_LABELS
0	417.0	66.0	564.0	1823.0	1	0.823627	0	1
1	1485.0	646.0	2710.0	73662.0	2	0.225539	4	3
2	654.0	299.0	1104.0	27318.0	1	0.306273	0	1
3	192.0	25.0	573.0	250.0	1	0.830682	0	1
4	43.0	8.0	528.0	22.0	1	0.810350	0	1

Silhouette score is given below

```
0.5167477722669528
```

```
For clusters = 2 The average silhouette_score is: 0.5313945367693249
For clusters = 3 The average silhouette_score is: 0.5340151343712788
For clusters = 4 The average silhouette_score is: 0.5524561729411546
For clusters = 5 The average silhouette_score is: 0.5208181010553294
For clusters = 6 The average silhouette_score is: 0.5320893142414745
For clusters = 7 The average silhouette_score is: 0.5550906360809267
For clusters = 8 The average silhouette_score is: 0.5342932176693953
For clusters = 9 The average silhouette_score is: 0.5118712298769437
For clusters = 10 The average silhouette_score is: 0.49094688987818824
```

```
0    97
1    41
2    47
3    27
4    48
5     9
6    22
7     6
Name: Clus_kmeans_, dtype: int64
```

In Hierarchical method, we got 5 clusters while in KMeans, we got 5 (using elbow plot) and 6 clusters (using silhouette score)

2.5 Describe cluster profiles for the clusters defined. Recommend different priority based actions that need to be taken for different clusters on the bases of their vulnerability situations according to their Economic and Health Conditions.

KMEANS_LABELS	0	1	2	3	4	5
Health_indeces1	4816.07	444.40	4116.97	2362.12	8208.22	2815.98
Health_indices2	1140.82	108.02	1293.00	848.38	1369.67	675.96
Per_capita_income	2319.30	686.81	4728.33	3160.00	5555.44	1530.96
GDP	399053.82	7241.66	342126.67	143591.27	426759.11	133085.91

Observation:

Health_indeces1: A single score that provides a summary of how the health system is performing in the State.

Cluster 4 has the highest score for health system that performing in the State.

Health_indeces2: A single score that provides a summary of how the health system is performing in certain areas of the States.

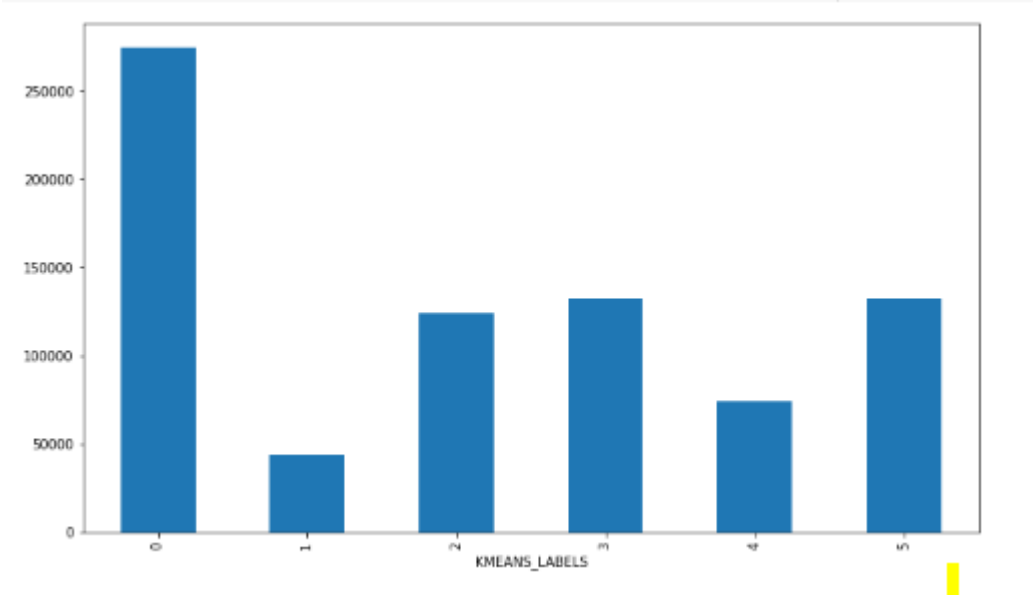
Cluster 2 and 4 has the highest score for the health system is performing in certain areas of the States.

Per_capita_income-Per capita income (PCI) measures the average income earned per person in a given area (city, region, country, etc.) in a specified year. It is calculated by dividing the area's total income by its total population.

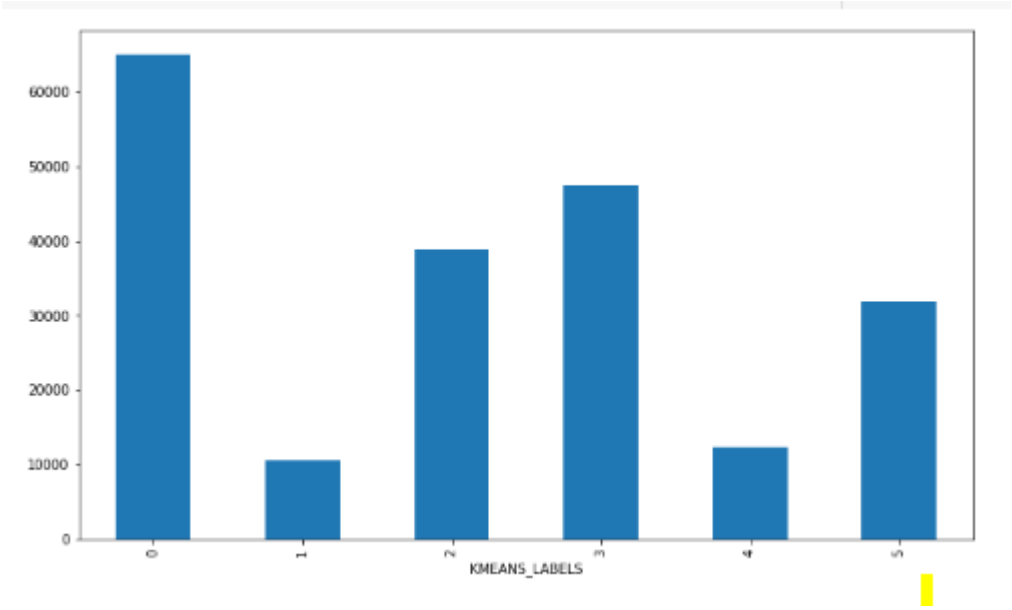
Cluster 2 and 4 has the highest Per capita income that is average income per person in a given area.

GDP: GDP provides an economic snapshot of a country/state, used to estimate the size of an economy and growth rate.

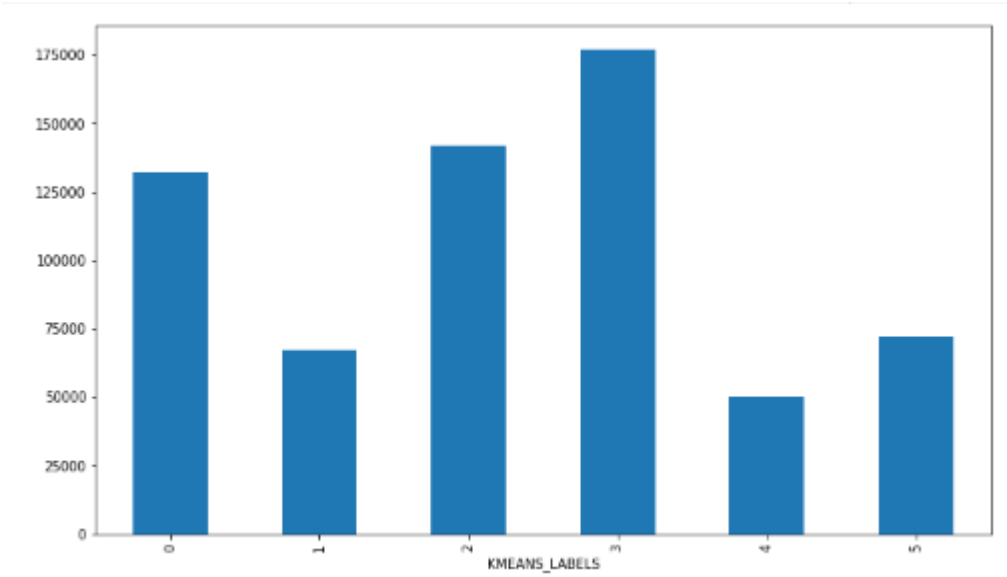
Cluster 4 has highest GDP means GDP is used to estimate the size of an economy and growth rate



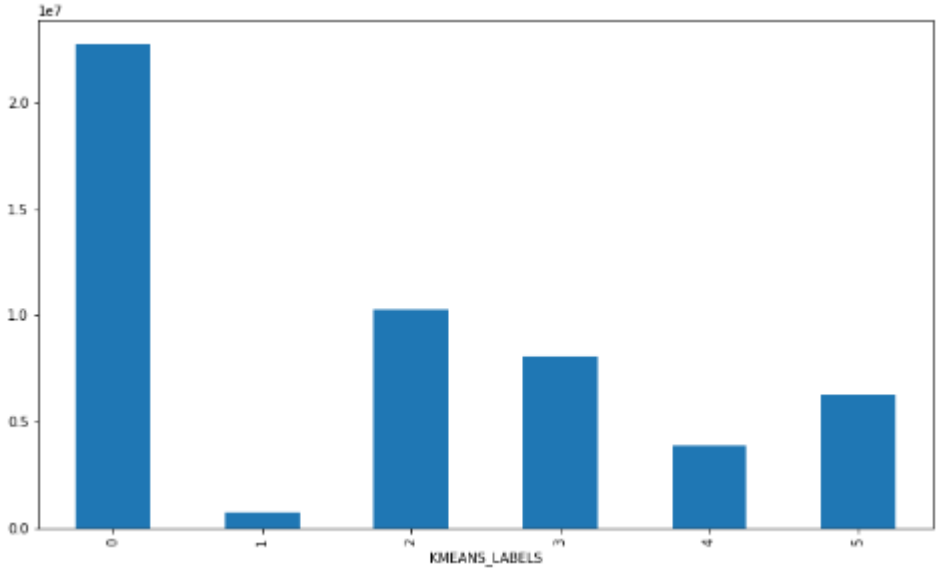
Health_indeces1



Health_indices2



Per_capita_income



GDP