GREAT LAKES
EXECUTIVE LEARNING

ADVANCED STATISTICS PROJECT

Ashwin Kumar A.G

PGDSBA (Nov Batch)

1.1 State the null and the alternate hypothesis for conducting one-way ANOVA for both Education and Occupation individually.

For Education: $H0$: The means of 'Education' variable with respect to each Salary is equal.

$H1$: At least one of the means of 'Education' variable with respect to each Salary is unequal.

For Occupation: $H0$: The means of 'Occupation' variable with respect to each Salary is equal.

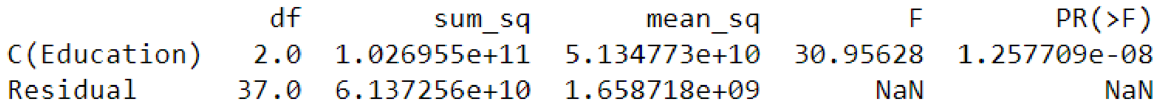
$H1$: At least one of the means of 'Occupation' variable with respect to each Salary is unequal.

1.2 Perform a one-way ANOVA on Salary with respect to Education. State whether the null hypothesis is accepted or rejected based on the ANOVA results.

One-way ANOVA of 'Education' variable with the 'Salary' variable.

$H0$: The means of 'Education' variable with respect to each Salary is equal.

$H1$: At least one of the means of 'Education' variable with respect to each Salary is unequal.



```
                 df        sum_sq       mean_sq          F        PR(>F)
C(Education)    2.0  1.026955e+11  5.134773e+10  30.95628  1.257709e-08
Residual       37.0  6.137256e+10  1.658718e+09       NaN           NaN
```

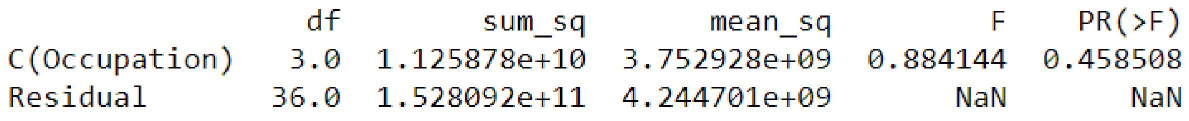
Since the p value in this scenario is less than α (0.05), we can reject the Null Hypothesis ($H0$) & conclude that there is a difference in the means of 'Education' variable w.r.t Salary is unequal.

1.3 Perform a one-way ANOVA on Salary with respect to Occupation. State whether the null hypothesis is accepted or rejected based on the ANOVA results.

One-way ANOVA of 'Occupation' variable with the 'Salary' variable.

$H0$: The means of 'Occupation' variable with respect to each Salary is equal.

$H1$: At least one of the means of 'Occupation' variable with respect to each Salary is unequal.



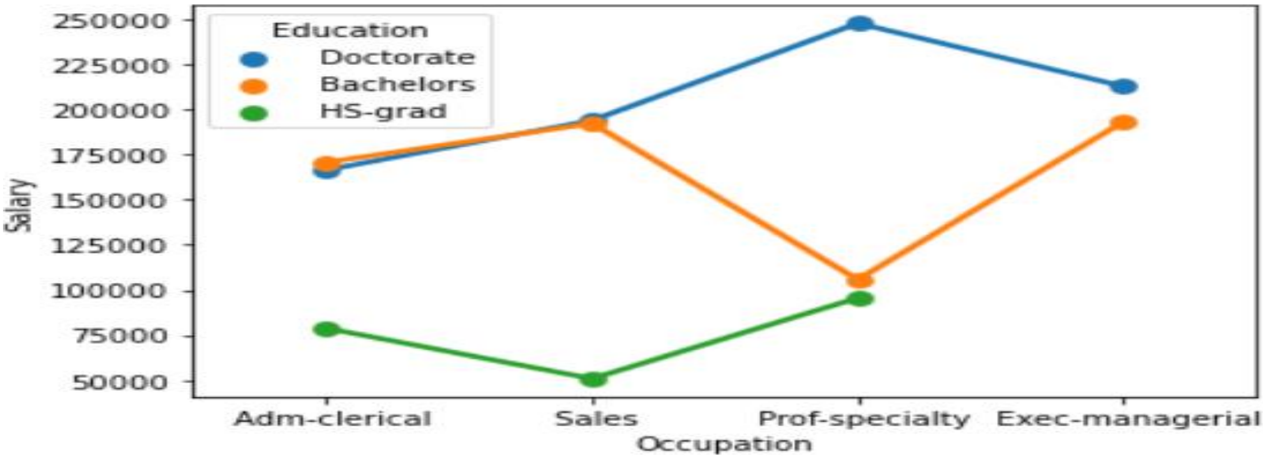
```
                   df        sum_sq       mean_sq         F    PR(>F)
C(Occupation)     3.0  1.125878e+10  3.752928e+09  0.884144  0.458508
Residual         36.0  1.528092e+11  4.244701e+09       NaN       NaN
```

Since the p value in this scenario is greater than α (0.05), we can say that we fail to reject the Null Hypothesis (H0).

1.4 If the null hypothesis is rejected in either (2) or in (3), find out which class means are significantly different. Interpret the result.

The Null Hypothesis is rejected in Scenario Education .We can also check class mean are different due to difference in the group means by using Tukeyhsd( ) function in Python.

1.5 What is the interaction between the two treatments? Analyze the effects of one variable on the other (Education and Occupation) with the help of an interaction plot.



Comments: From the above interaction plots,there seems to be an interaction between the two Education & Occupation variables I.E (Doctrate & Bachelors –Education Variables) corresponds with (Adm-Clerical(veryclose) & Sales-Occupation Variable) w.r.t to Salary. Other than these mentioned the rest of the values sees almost no interaction amongst the variables.

*1.6 Perform a two-way ANOVA based on Salary with respect to both Education and Occupation (along with their interaction Education*Occupation). State the null and alternative hypotheses and state your results. How will you interpret this result?*

H0: The means of 'Salary' variable with respect to each Education and Occupation is equal.

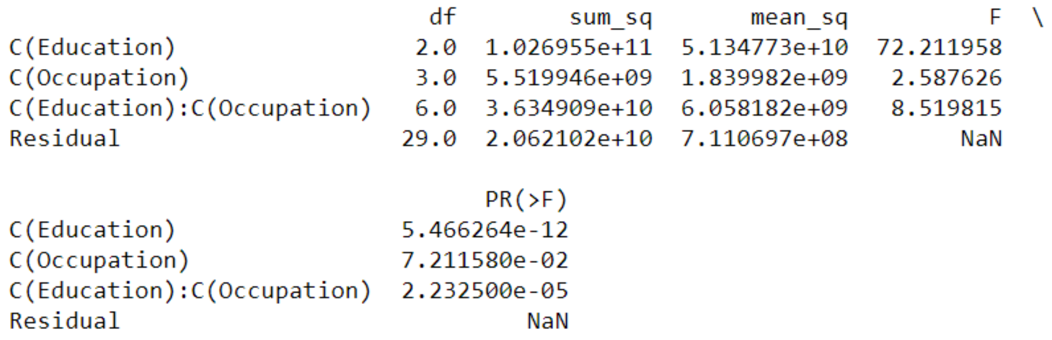
H1: At least one of the means of 'Salary' variable with respect to each Education and Occupation is unequal.

ANOVA with both 'Education' and 'Occupation' variables with respect to the variable 'Salary'

```
                  df      sum_sq        mean_sq          F              PR(>F)

C(Education)    2.0  1.026955e+11    5.134773e+10   31.257677    1.981539e-08

C(Occupation)   3.0  5.519946e+09  1.839982e+09   1.120080      3.545825e-01

Residual           34.0  5.585261e+10   1.642724e+09       NaN           NaN
```

Comments: Considering both the factors Education & Occupation, Education is a significant factor variable as the p value is <0.05 whereas Occupation is not a significant variable as p value of Occupation is >0.05.

INTERACTION EFFECT:



```
                                df        sum_sq       mean_sq          F  \
C(Education)                   2.0  1.026955e+11  5.134773e+10  72.211958   
C(Occupation)                  3.0  5.519946e+09  1.839982e+09   2.587626   
C(Education):C(Occupation)     6.0  3.634909e+10  6.058182e+09   8.519815   
Residual                      29.0  2.062102e+10  7.110697e+08        NaN   

                                  PR(>F)  
C(Education)                5.466264e-12  
C(Occupation)               7.211580e-02  
C(Education):C(Occupation)  2.232500e-05  
Residual                             NaN  
```

Comments :** **By the interaction effect, Education:Occupation has become important since we can see that there was an interaction between those two variables. Thus,interaction between the variables has now become important because interaction effect P value is <0.05

1.7 Explain the business implications of performing ANOVA for this particular case study.

By performing an ANOVA test we can see that Education is the major factor impacting the Salary across different Occupation. However ,while interaction between (Education: Occupation) there are few observations that both has a slight interaction corresponding with Salary. The interaction between (Education: Occupation) is slightly important. But, Education is the significant predictor of Salary.

2.1 Perform Exploratory Data Analysis [both univariate and multivariate analysis to be performed]. What insight do you draw from the EDA?

Inference:

EDA was performed for the dataset using Python; both Univariate & Bivariate Analysis was done.

Univariate Analysis.

Load the csv file & derive the data types of the variable & checking the number of rows & columns present. Analysing the variables whether they are Integer, Object or a Float Value.

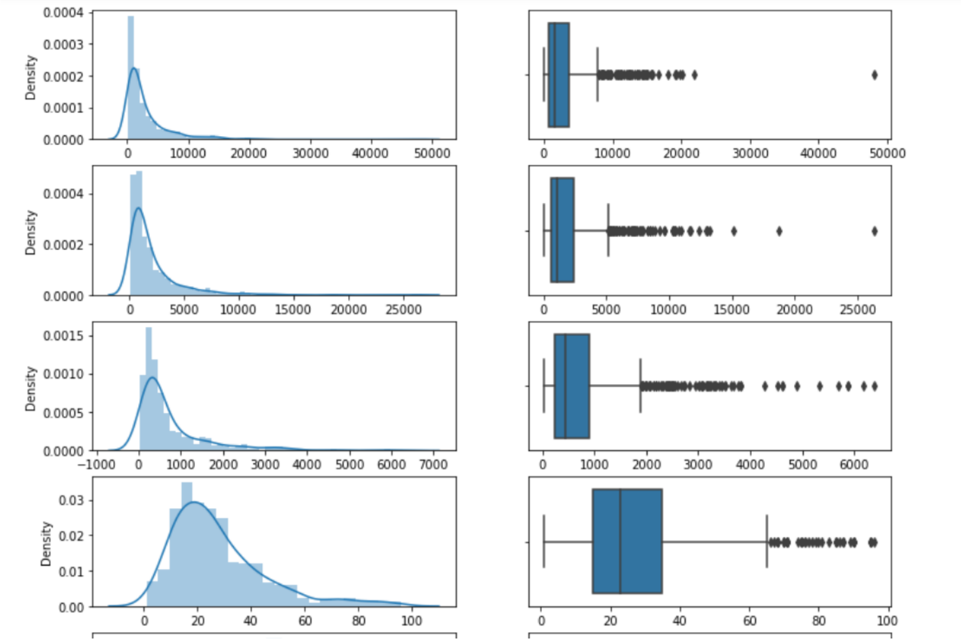
Df1.shape

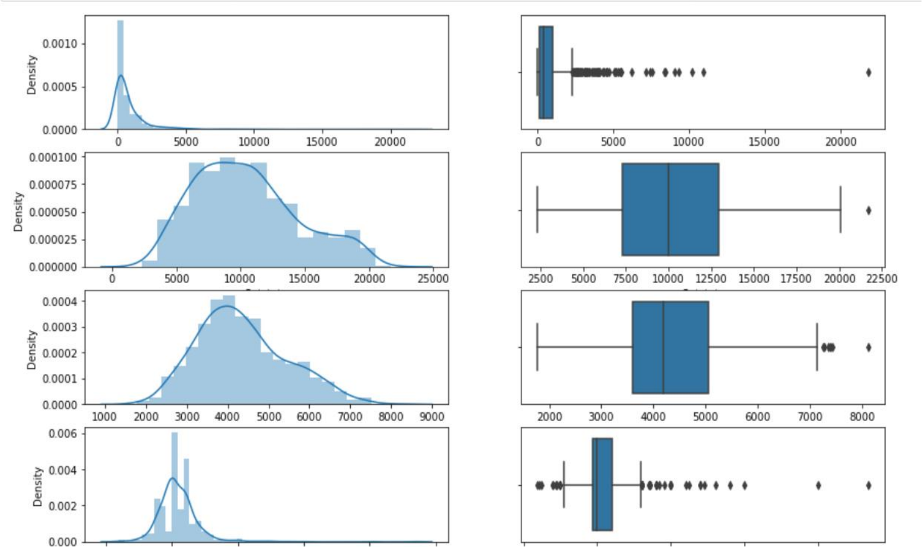
Names object Apps int64 Accept int64 Enroll int64 Top10perc int64 Top25perc int64 F.Undergrad int64 P.Undergrad int64 Outstate int64 Room.Board int64 Books int64 Personal int64 PhD int64 Terminal int64 S.F.Ratio float64 perc.alumni int64 Expend int64 Grad.Rate int64 dtype: object

Then we describe the data to check the min,max,mean,std deviation & the summary of numerical columns.

Missing Values was not found in the dataset as well as Duplicate rows were not found in the dataset.

We, check the normality of the data with the help of a distplot to see if the data is normally distributed.



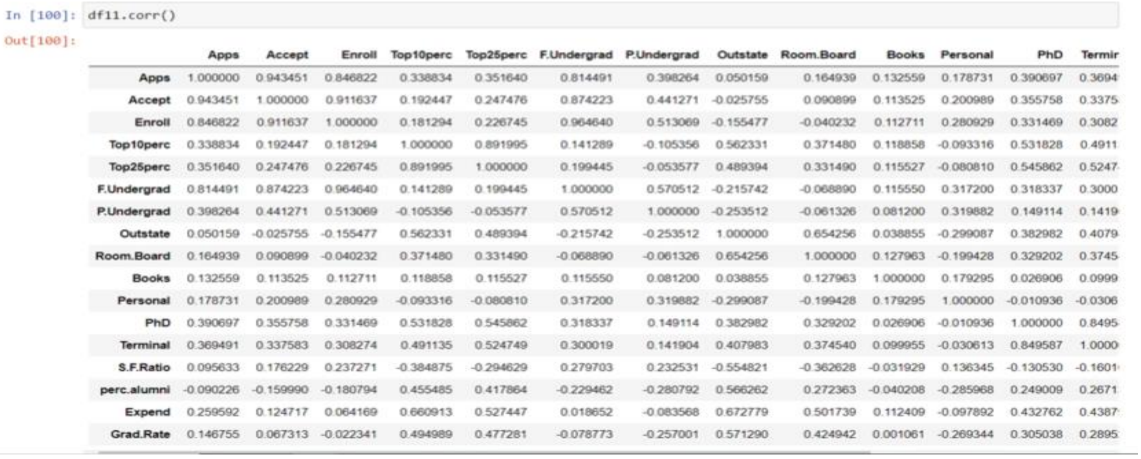


df1.skew(axis=0)

Apps 1.166334 Accept 1.151199 Enroll 1.155018 Top10perc 0.880389 Top25perc 0.259340 F.Undergrad 1.149049 P.Undergrad 1.166203 Outstate 0.507441 Room.Board 0.442530 Books 0.239716 Personal 0.792012 PhD -0.606822 Terminal -0.710578 S.F.Ratio 0.266828 perc.alumni 0.572760 Expend 0.875535 Grad.Rate -0.106733

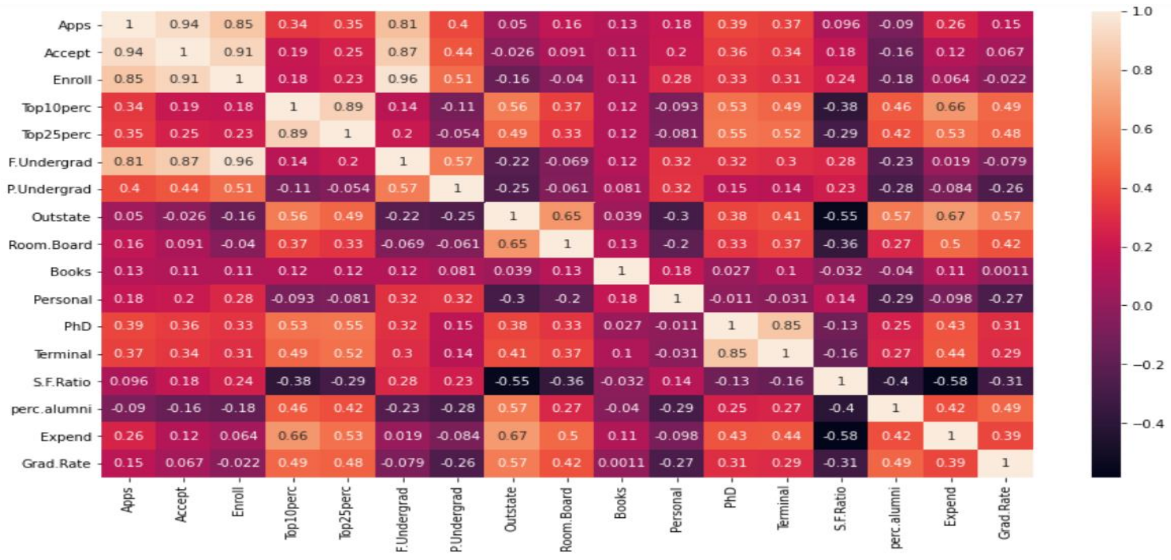
To check for Normality distribution ,skewness is formulated.If the Skewness value is 0 the data is normally distributed.If the Skewness Value is >0 the data is skewed towards left & if the value is <0 the data is skewed towards right.ormulated.If the Skewness value is 0 the data is normally distributed.If the Skewness Value is >0 the data is skewed towards left & if the value is <0 the data is skewed towards right.

MultiVariate Analysis:



```
In [100]: df11.corr()
Out[100]:
```

	Apps	Accept	Enroll	Top10perc	Top25perc	F.Undergrad	P.Undergrad	Outstate	Room.Board	Books	Personal	PhD	Termir
Apps	1.000000	0.943451	0.846822	0.338834	0.351640	0.814491	0.398264	0.050159	0.164939	0.132559	0.178731	0.390697	0.3694
Accept	0.943451	1.000000	0.911637	0.192447	0.247476	0.874223	0.441271	-0.025755	0.090899	0.113525	0.200989	0.355758	0.3375
Enroll	0.846822	0.911637	1.000000	0.181294	0.226745	0.964640	0.513069	-0.155477	-0.040232	0.112711	0.280929	0.331469	0.3082
Top10perc	0.338834	0.192447	0.181294	1.000000	0.891995	0.141289	-0.105356	0.562331	0.371480	0.118858	-0.093316	0.531828	0.4911
Top25perc	0.351640	0.247476	0.226745	0.891995	1.000000	0.199445	-0.053577	0.489394	0.331490	0.115527	-0.080810	0.545862	0.5247
F.Undergrad	0.814491	0.874223	0.964640	0.141289	0.199445	1.000000	0.570512	-0.215742	-0.068890	0.115550	0.317200	0.318337	0.3000
P.Undergrad	0.398264	0.441271	0.513069	-0.105356	-0.053577	0.570512	1.000000	-0.253512	-0.061326	0.081200	0.319882	0.149114	0.1419
Outstate	0.050159	-0.025755	-0.155477	0.562331	0.489394	-0.215742	-0.253512	1.000000	0.654256	0.038855	-0.299087	0.382982	0.4079
Room.Board	0.164939	0.090899	-0.040232	0.371480	0.331490	-0.068890	-0.061326	0.654256	1.000000	0.127963	-0.199428	0.329202	0.3745
Books	0.132559	0.113525	0.112711	0.118858	0.115527	0.115550	0.081200	0.038855	0.127963	1.000000	0.179295	0.026906	0.0999
Personal	0.178731	0.200989	0.280929	-0.093316	-0.080810	0.317200	0.319882	-0.299087	-0.199428	0.179295	1.000000	-0.010936	-0.0306
PhD	0.390697	0.355758	0.331469	0.531828	0.545862	0.318337	0.149114	0.382982	0.329202	0.026906	-0.010936	1.000000	0.8495
Terminal	0.369491	0.337583	0.308274	0.491135	0.524749	0.300019	0.141904	0.407983	0.374540	0.099955	-0.030613	0.849587	1.0000
S.F.Ratio	0.095633	0.176229	0.237271	-0.384875	-0.294629	0.279703	0.232531	-0.554821	-0.362628	-0.031929	0.136345	-0.130530	-0.1601
perc.alumni	-0.090226	-0.159990	-0.180794	0.455485	0.417864	-0.229462	-0.280792	0.566262	0.272363	-0.040208	-0.285968	0.249009	0.2671
Expend	0.259592	0.124717	0.064169	0.660913	0.527447	0.018652	-0.083568	0.672779	0.501739	0.112409	-0.097892	0.432762	0.4387
Grad.Rate	0.146755	0.067313	-0.022341	0.494989	0.477281	-0.078773	-0.257001	0.571290	0.424942	0.001061	-0.269344	0.305038	0.2895

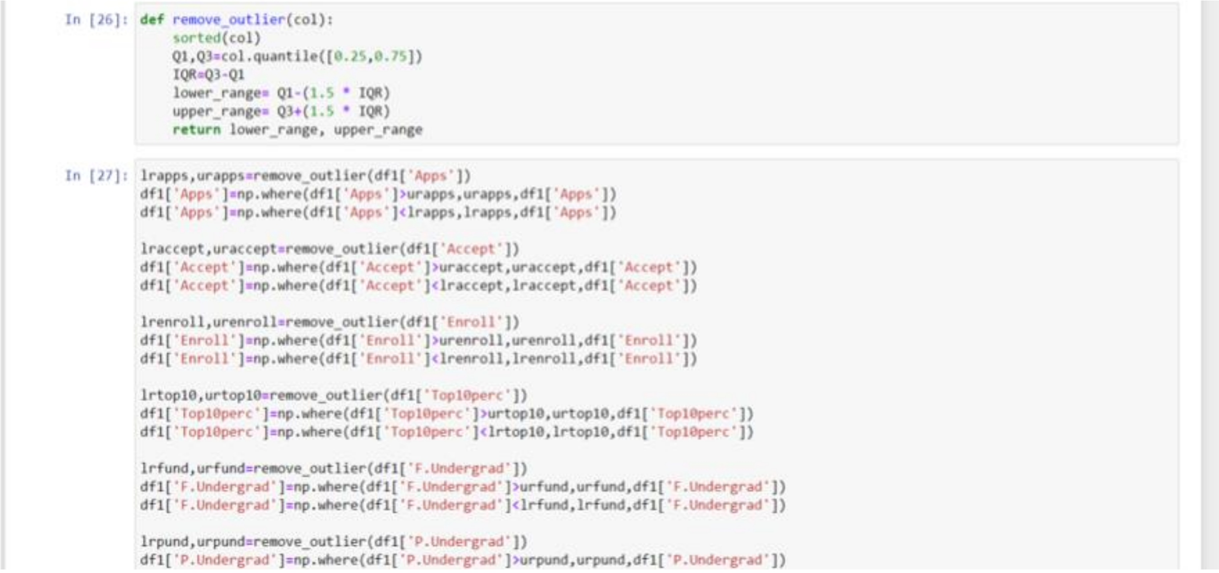


Corrleation for the data is formulated & with the help of heat map we can relate the variables based on positive & negative Correlation.

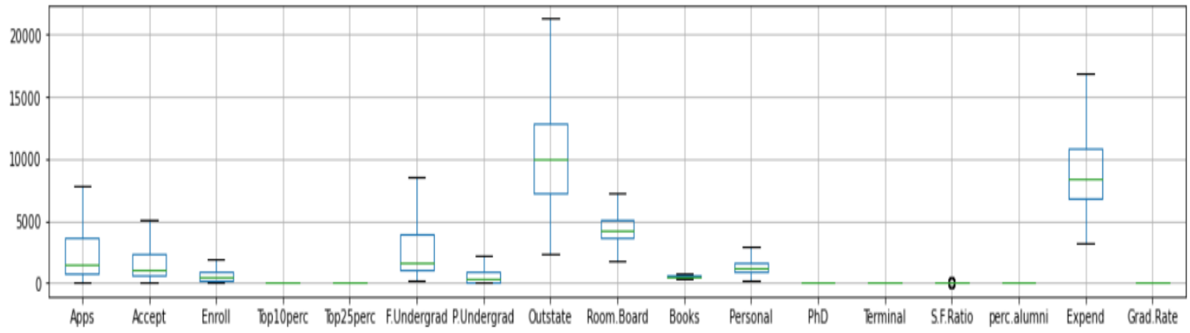
2.2 Is scaling necessary for PCA in this case? Give justification and perform scaling.

Yes,Scaling is necessary to normalise the given data.We need to standardise the given data to calculate process the data quicker for performing PCA.

Outliers were removed which were present in the data.

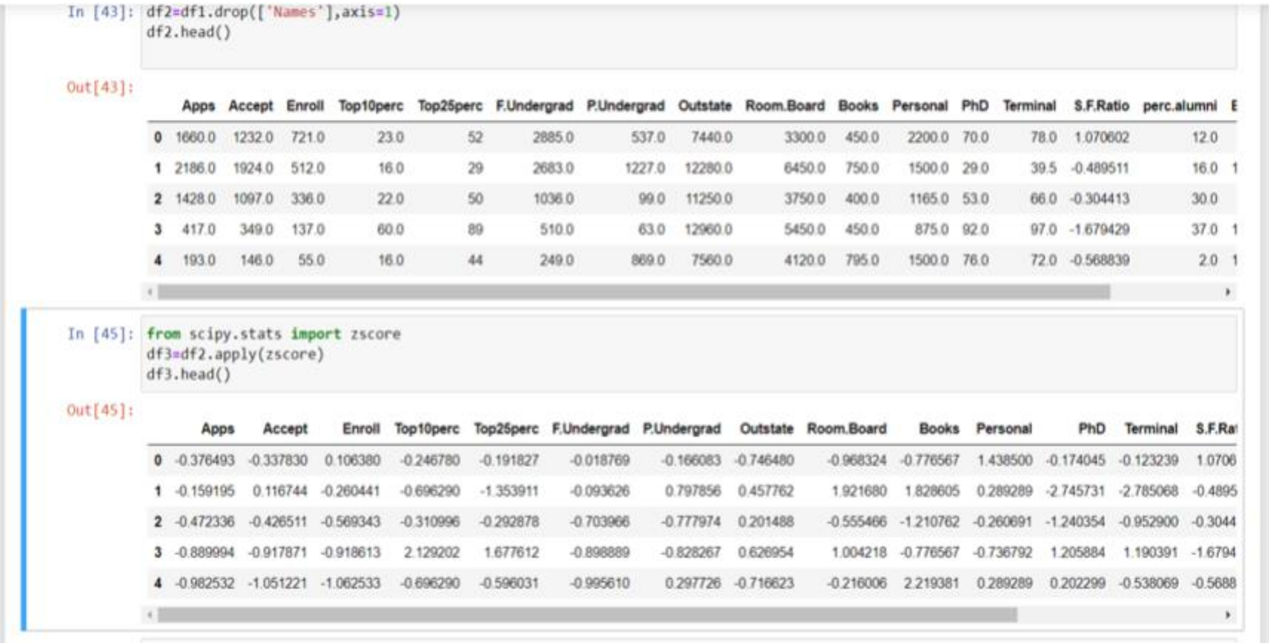


```
In [26]: def remove_outlier(col):
             sorted(col)
             Q1,Q3=col.quantile([0.25,0.75])
             IQR=Q3-Q1
             lower_range= Q1-(1.5 * IQR)
             upper_range= Q3+(1.5 * IQR)
             return lower_range, upper_range

In [27]: lrapps,urapps=remove_outlier(df1['Apps'])
         df1['Apps']=np.where(df1['Apps']>urapps,urapps,df1['Apps'])
         df1['Apps']=np.where(df1['Apps']<lrapps,lrapps,df1['Apps'])

         lraccept,uraccept=remove_outlier(df1['Accept'])
         df1['Accept']=np.where(df1['Accept']>uraccept,uraccept,df1['Accept'])
         df1['Accept']=np.where(df1['Accept']<lraccept,lraccept,df1['Accept'])

         lrenroll,urenroll=remove_outlier(df1['Enroll'])
         df1['Enroll']=np.where(df1['Enroll']>urenroll,urenroll,df1['Enroll'])
         df1['Enroll']=np.where(df1['Enroll']<lrenroll,lrenroll,df1['Enroll'])

         lrtop10,urtop10=remove_outlier(df1['Top10perc'])
         df1['Top10perc']=np.where(df1['Top10perc']>urtop10,urtop10,df1['Top10perc'])
         df1['Top10perc']=np.where(df1['Top10perc']<lrtop10,lrtop10,df1['Top10perc'])

         lrfund,urfund=remove_outlier(df1['F.Undergrad'])
         df1['F.Undergrad']=np.where(df1['F.Undergrad']>urfund,urfund,df1['F.Undergrad'])
         df1['F.Undergrad']=np.where(df1['F.Undergrad']<lrfund,lrfund,df1['F.Undergrad'])

         lrpund,urpund=remove_outlier(df1['P.Undergrad'])
         df1['P.Undergrad']=np.where(df1['P.Undergrad']>urpund,urpund,df1['P.Undergrad'])
```



Also,it was observed a string column 'Names' which was removed in order to apply 'Z Score'.

The Data was standardised with the use of Zscore in order to process PCA



In [43]:
```
df2=df1.drop(['Names'],axis=1)
df2.head()
```

Out[43]:

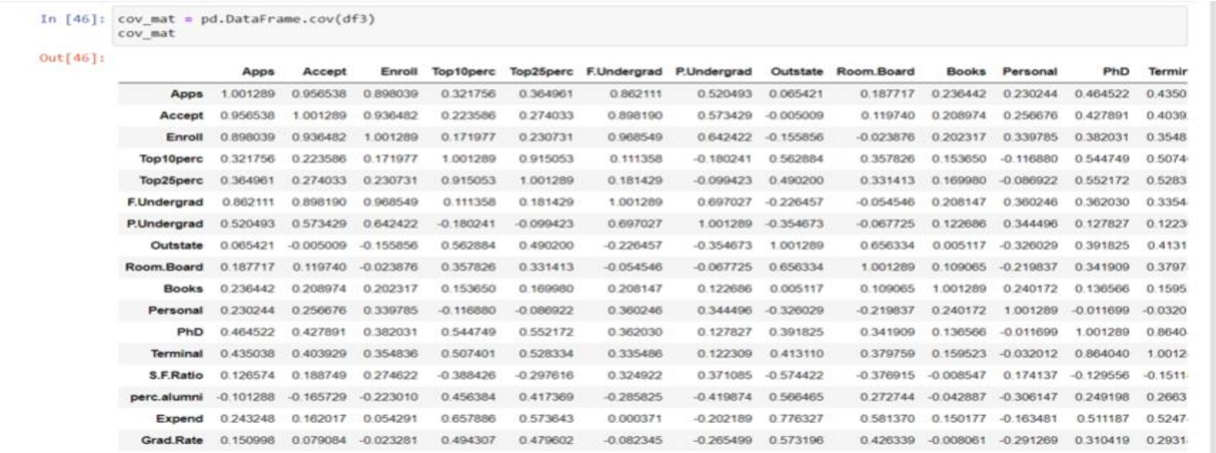
	Apps	Accept	Enroll	Top10perc	Top25perc	F.Undergrad	P.Undergrad	Outstate	Room.Board	Books	Personal	PhD	Terminal	S.F.Ratio	perc.alumni	E
0	1660.0	1232.0	721.0	23.0	52	2885.0	537.0	7440.0	3300.0	450.0	2200.0	70.0	78.0	1.070602	12.0	
1	2186.0	1924.0	512.0	16.0	29	2683.0	1227.0	12280.0	6450.0	750.0	1500.0	29.0	39.5	-0.489511	16.0	1
2	1428.0	1097.0	336.0	22.0	50	1036.0	99.0	11250.0	3750.0	400.0	1165.0	53.0	66.0	-0.304413	30.0	
3	417.0	349.0	137.0	60.0	89	510.0	63.0	12960.0	5450.0	450.0	875.0	92.0	97.0	-1.679429	37.0	1
4	193.0	146.0	55.0	16.0	44	249.0	869.0	7560.0	4120.0	795.0	1500.0	76.0	72.0	-0.568839	2.0	1

In [45]:
```
from scipy.stats import zscore
df3=df2.apply(zscore)
df3.head()
```

Out[45]:

	Apps	Accept	Enroll	Top10perc	Top25perc	F.Undergrad	P.Undergrad	Outstate	Room.Board	Books	Personal	PhD	Terminal	S.F.Ra
0	-0.376493	-0.337830	0.106380	-0.246780	-0.191827	-0.018769	-0.166083	-0.746480	-0.968324	-0.776567	1.438500	-0.174045	-0.123239	1.0706
1	-0.159195	0.116744	-0.260441	-0.696290	-1.353911	-0.093626	0.797856	0.457762	1.921680	1.828605	0.289289	-2.745731	-2.785068	-0.4895
2	-0.472336	-0.426511	-0.569343	-0.310996	-0.292878	-0.703966	-0.777974	0.201488	-0.555466	-1.210762	-0.260691	-1.240354	-0.952900	-0.3044
3	-0.889994	-0.917871	-0.918613	2.129202	1.677612	-0.898889	-0.828267	0.626954	1.004218	-0.776567	-0.736792	1.205884	1.190391	-1.6794
4	-0.982532	-1.051221	-1.062533	-0.696290	-0.596031	-0.995610	0.297726	-0.716623	-0.216006	2.219381	0.289289	0.202299	-0.538069	-0.5688

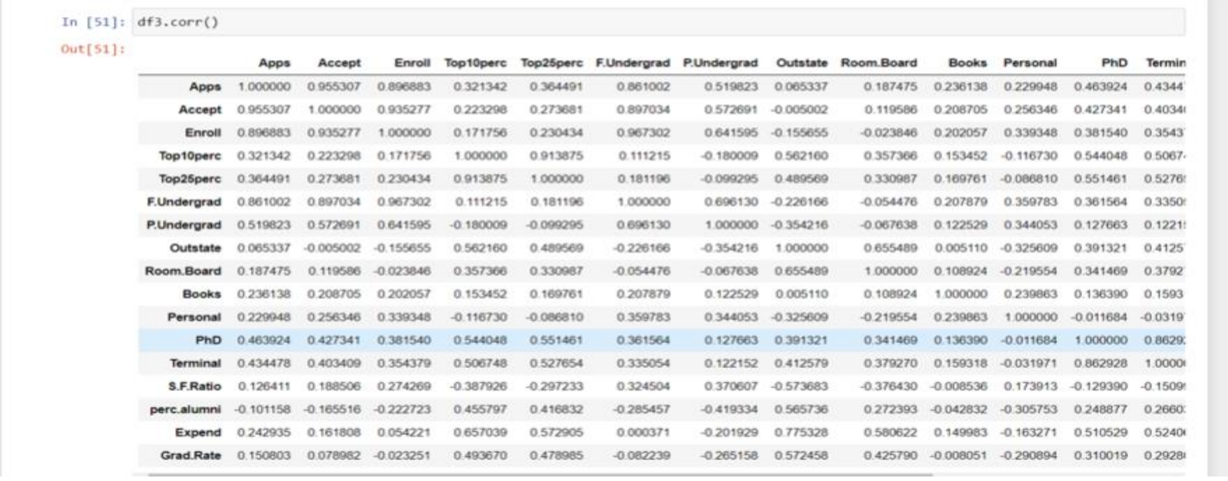
2.3 Comment on the comparison between the covariance and the correlation matrices from this data.



In [46]:
```
cov_mat = pd.DataFrame.cov(df3)
cov_mat
```

Out[46]:

	Apps	Accept	Enroll	Top10perc	Top25perc	F.Undergrad	P.Undergrad	Outstate	Room.Board	Books	Personal	PhD	Termir
Apps	1.001289	0.956538	0.898039	0.321756	0.364961	0.862111	0.520493	0.065421	0.187717	0.236442	0.230244	0.464522	0.4350
Accept	0.956538	1.001289	0.936482	0.223586	0.274033	0.898190	0.573429	-0.005009	0.119740	0.208974	0.256676	0.427891	0.4039
Enroll	0.898039	0.936482	1.001289	0.171977	0.230731	0.968549	0.642422	-0.155856	-0.023876	0.202317	0.339785	0.382031	0.3548
Top10perc	0.321756	0.223586	0.171977	1.001289	0.915053	0.111358	-0.180241	0.562884	0.357826	0.153650	-0.116880	0.544749	0.5074
Top25perc	0.364961	0.274033	0.230731	0.915053	1.001289	0.181429	-0.099423	0.490200	0.331413	0.169980	-0.086922	0.552172	0.5283
F.Undergrad	0.862111	0.898190	0.968549	0.111358	0.181429	1.001289	0.697027	-0.226457	-0.054546	0.208147	0.360246	0.362030	0.3354
P.Undergrad	0.520493	0.573429	0.642422	-0.180241	-0.099423	0.697027	1.001289	-0.354673	-0.067725	0.122686	0.344496	0.127827	0.1223
Outstate	0.065421	-0.005009	-0.155856	0.562884	0.490200	-0.226457	-0.354673	1.001289	0.656334	0.005117	-0.326029	0.391825	0.4131
Room.Board	0.187717	0.119740	-0.023876	0.357826	0.331413	-0.054546	-0.067725	0.656334	1.001289	0.109065	-0.219837	0.341909	0.3797
Books	0.236442	0.208974	0.202317	0.153650	0.169980	0.208147	0.122686	0.005117	0.109065	1.001289	0.240172	0.136566	0.1595
Personal	0.230244	0.256676	0.339785	-0.116880	-0.086922	0.360246	0.344496	-0.326029	-0.219837	0.240172	1.001289	-0.011699	-0.0320
PhD	0.464522	0.427891	0.382031	0.544749	0.552172	0.362030	0.127827	0.391825	0.341909	0.136566	-0.011699	1.001289	0.8640
Terminal	0.435038	0.403929	0.354836	0.507401	0.528334	0.335486	0.122309	0.413110	0.379759	0.159523	-0.032012	0.864040	1.0012
S.F.Ratio	0.126574	0.188749	0.274622	-0.388426	-0.297616	0.324922	0.371085	-0.574422	-0.376915	-0.008547	0.174137	-0.129556	-0.1511
perc.alumni	-0.101288	-0.165729	-0.223010	0.456384	0.417369	-0.285825	-0.419874	0.566465	0.272744	-0.042887	-0.306147	0.249198	0.2663
Expend	0.243248	0.162017	0.054291	0.657886	0.573643	0.000371	-0.202189	0.776327	0.581370	0.150177	-0.163481	0.511187	0.5247
Grad.Rate	0.150998	0.079084	-0.023281	0.494307	0.479602	-0.082345	-0.265499	0.573196	0.426339	-0.008061	-0.291269	0.310419	0.2931



In [51]:
```
df3.corr()
```

Out[51]:

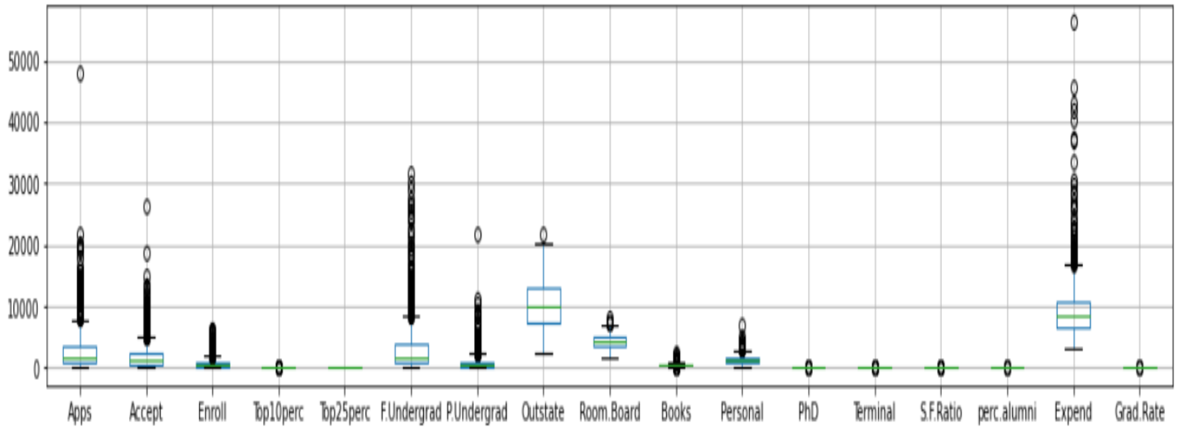
	Apps	Accept	Enroll	Top10perc	Top25perc	F.Undergrad	P.Undergrad	Outstate	Room.Board	Books	Personal	PhD	Termin
Apps	1.000000	0.955307	0.896883	0.321342	0.364491	0.861002	0.519823	0.065337	0.187475	0.236138	0.229948	0.463924	0.4344
Accept	0.955307	1.000000	0.935277	0.223298	0.273681	0.897034	0.572691	-0.005002	0.119586	0.208705	0.256346	0.427341	0.4034
Enroll	0.896883	0.935277	1.000000	0.171756	0.230434	0.967302	0.641595	-0.155655	-0.023846	0.202057	0.339348	0.381540	0.3543
Top10perc	0.321342	0.223298	0.171756	1.000000	0.913875	0.111215	-0.180009	0.562160	0.357366	0.153452	-0.116730	0.544048	0.5067
Top25perc	0.364491	0.273681	0.230434	0.913875	1.000000	0.181196	-0.099295	0.489569	0.330987	0.169761	-0.086810	0.551461	0.5276
F.Undergrad	0.861002	0.897034	0.967302	0.111215	0.181196	1.000000	0.696130	-0.226166	-0.054476	0.207879	0.359783	0.361564	0.3350
P.Undergrad	0.519823	0.572691	0.641595	-0.180009	-0.099295	0.696130	1.000000	-0.354216	-0.067638	0.122529	0.344053	0.127663	0.1221
Outstate	0.065337	-0.005002	-0.155655	0.562160	0.489569	-0.226166	-0.354216	1.000000	0.655489	0.005110	-0.325609	0.391321	0.4125
Room.Board	0.187475	0.119586	-0.023846	0.357366	0.330987	-0.054476	-0.067638	0.655489	1.000000	0.108924	-0.219554	0.341469	0.3792
Books	0.236138	0.208705	0.202057	0.153452	0.169761	0.207879	0.122529	0.005110	0.108924	1.000000	0.239863	0.136390	0.1593
Personal	0.229948	0.256346	0.339348	-0.116730	-0.086810	0.359783	0.344053	-0.325609	-0.219554	0.239863	1.000000	-0.011684	-0.0319
PhD	0.463924	0.427341	0.381540	0.544048	0.551461	0.361564	0.127663	0.391321	0.341469	0.136390	-0.011684	1.000000	0.8629
Terminal	0.434478	0.403409	0.354379	0.506748	0.527654	0.335054	0.122152	0.412579	0.379270	0.159318	-0.031971	0.862928	1.0000
S.F.Ratio	0.126411	0.188506	0.274269	-0.387926	-0.297233	0.324504	0.370607	-0.573683	-0.376430	-0.008536	0.173913	-0.129390	-0.1509
perc.alumni	-0.101158	-0.165516	-0.222723	0.455797	0.416832	-0.285457	-0.419334	0.565736	0.272393	-0.042832	-0.305753	0.248877	0.2660
Expend	0.242935	0.161808	0.054221	0.657039	0.572905	0.000371	-0.201929	0.775328	0.580622	0.149983	-0.163271	0.510529	0.5240
Grad.Rate	0.150803	0.078982	-0.023251	0.493670	0.478985	-0.082239	-0.265158	0.572458	0.425790	-0.008051	-0.290894	0.310019	0.2928

From the above data, we can observe that the Covariance & Correlation matrices values are the same for the variables after scaling. Correlation becomes scaled after deriving covariance

,when the values of the variables are positive they are positively correlated;if they are negative they are negatively correlated. When the values of the variables are zero they are uncorrelated.

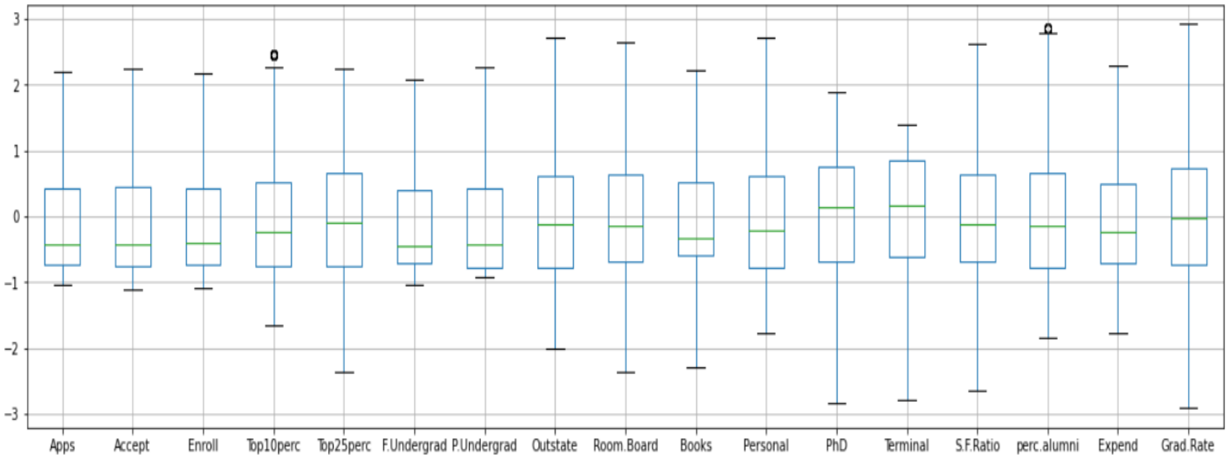
2.4 Check the dataset for outliers before and after scaling. What insight do you derive here?

BEFORE SCALING:



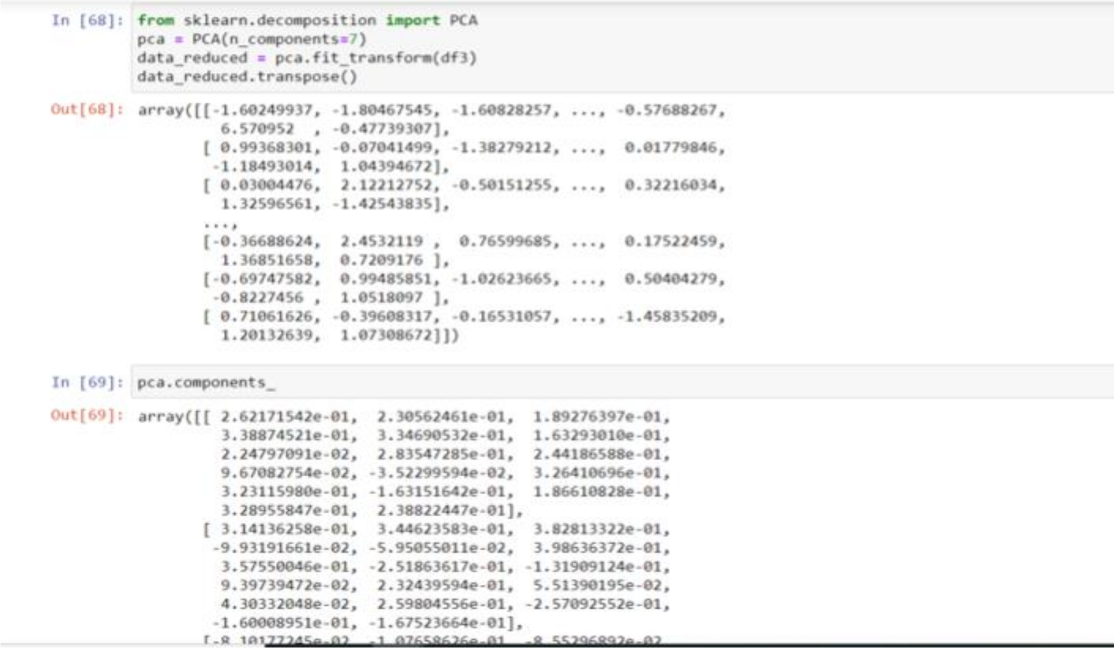
We can a presence of Outliers almost in all the columns of the given dataset before Scaling.

AFTER SCALING:

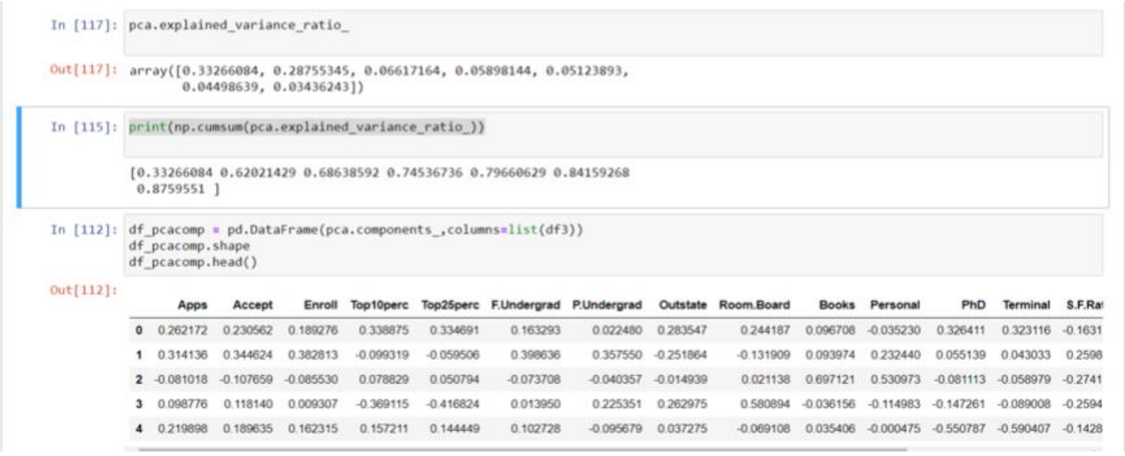


By performing a Univariate Analysis, we can observe the presence of one or two outliers with the help of a boxplot even after standardization. By removing the outliers & through performing Scaling we were able to preprocess and normalize the data to perform PCA.

2.5 Perform PCA and export the data of the Principal Component scores into a data frame.

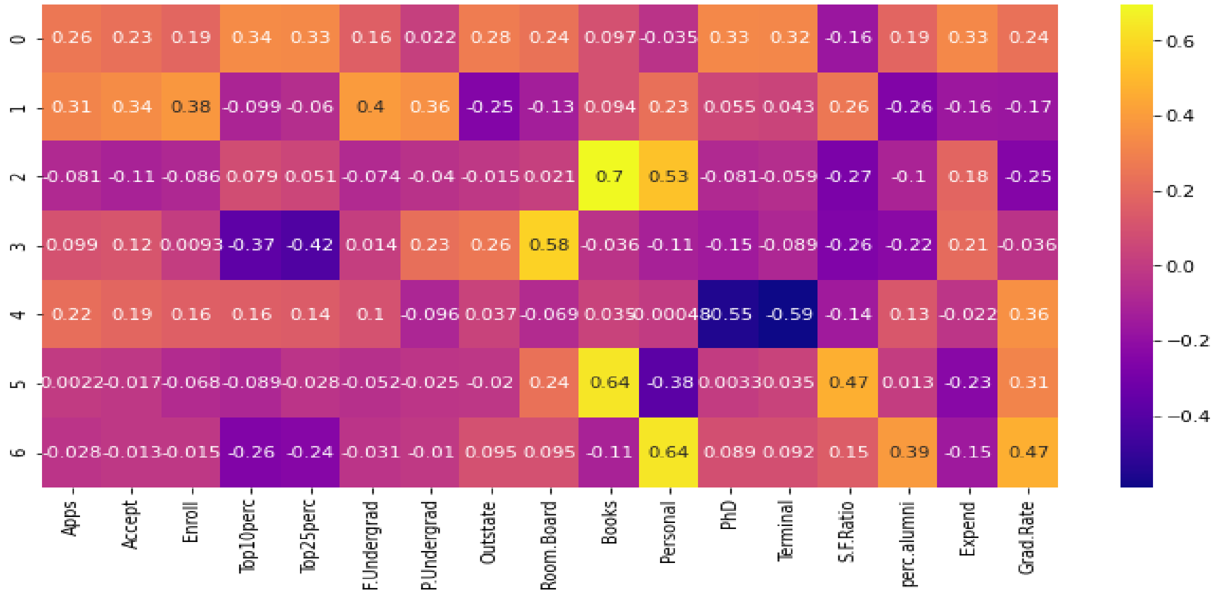


```
In [68]: from sklearn.decomposition import PCA
         pca = PCA(n_components=7)
         data_reduced = pca.fit_transform(df3)
         data_reduced.transpose()

Out[68]: array([[-1.60249937, -1.80467545, -1.60828257, ..., -0.57688267,
                  6.570952  , -0.47739307],
                [ 0.99368301, -0.07041499, -1.38279212, ...,  0.01779846,
                 -1.18493014,  1.04394672],
                [ 0.03004476,  2.12212752, -0.50151255, ...,  0.32216034,
                  1.32596561, -1.42543835],
                ...,
                [-0.36688624,  2.4532119 ,  0.76599685, ...,  0.17522459,
                  1.36851658,  0.7209176 ],
                [-0.69747582,  0.99485851, -1.02623665, ...,  0.50404279,
                 -0.8227456 ,  1.0518097 ],
                [ 0.71061626, -0.39608317, -0.16531057, ..., -1.45835209,
                  1.20132639,  1.07308672]])

In [69]: pca.components_

Out[69]: array([[ 2.62171542e-01,  2.30562461e-01,  1.89276397e-01,
                  3.38874521e-01,  3.34690532e-01,  1.63293010e-01,
                  2.24797091e-02,  2.83547285e-01,  2.44186588e-01,
                  9.67082754e-02, -3.52299594e-02,  3.26410696e-01,
                  3.23115980e-01, -1.63151642e-01,  1.86610828e-01,
                  3.28955847e-01,  2.38822447e-01],
                [ 3.14136258e-01,  3.44623583e-01,  3.82813322e-01,
                 -9.93191661e-02, -5.95055011e-02,  3.98636372e-01,
                  3.57550046e-01, -2.51863617e-01, -1.31909124e-01,
                  9.39739472e-02,  2.32439594e-01,  5.51390195e-02,
                  4.30332048e-02,  2.59804556e-01, -2.57092552e-01,
                 -1.60008951e-01, -1.67523664e-01],
                [-8.10177245e-02, -1.07658626e-01, -8.55296892e-02,
```



```
In [117]: pca.explained_variance_ratio_

Out[117]: array([0.33266084, 0.28755345, 0.06617164, 0.05898144, 0.05123893,
                 0.04498639, 0.03436243])

In [115]: print(np.cumsum(pca.explained_variance_ratio_))

          [0.33266084 0.62021429 0.68638592 0.74536736 0.79660629 0.84159268
           0.8759551 ]

In [112]: df_pcacomp = pd.DataFrame(pca.components_,columns=list(df3))
          df_pcacomp.shape
          df_pcacomp.head()

Out[112]:
```

	Apps	Accept	Enroll	Top10perc	Top25perc	F.Undergrad	P.Undergrad	Outstate	Room.Board	Books	Personal	PhD	Terminal	S.F.Ra
0	0.262172	0.230562	0.189276	0.338875	0.334691	0.163293	0.022480	0.283547	0.244187	0.096708	-0.035230	0.326411	0.323116	-0.1631
1	0.314136	0.344624	0.382813	-0.099319	-0.059506	0.398636	0.357550	-0.251864	-0.131909	0.093974	0.232440	0.055139	0.043033	0.2598
2	-0.081018	-0.107659	-0.085530	0.078829	0.050794	-0.073708	-0.040357	-0.014939	0.021138	0.697121	0.530973	-0.081113	-0.058979	-0.2741
3	0.098776	0.118140	0.009307	-0.369115	-0.416824	0.013950	0.225351	0.262975	0.580894	-0.036156	-0.114983	-0.147261	-0.089008	-0.2594
4	0.219898	0.189635	0.162315	0.157211	0.144449	0.102728	-0.095679	0.037275	-0.069108	0.035406	-0.000475	-0.550787	-0.590407	-0.1428



PCA was performed.The data of PC scores exported into a dataframe.

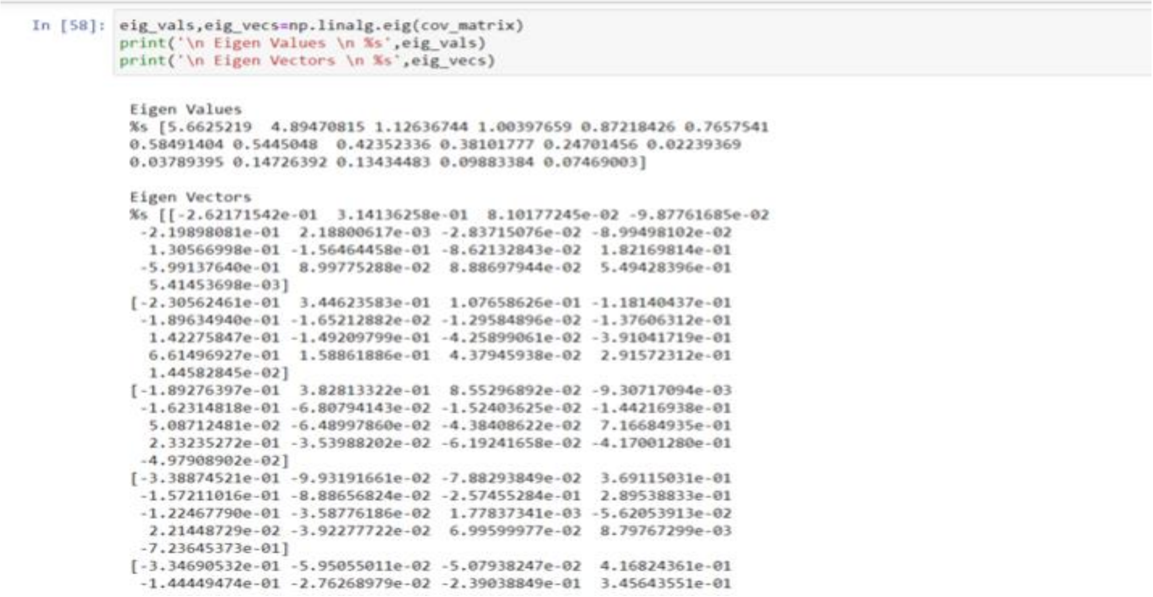
pca.explained_variance_ratio_

array([0.33266084, 0.28755345, 0.06617164, 0.05898144, 0.05123893, 0.04498639, 0.03436243])

This constitutes the variance explained by each component. The 1st Component signifies 33.2% of variance in the model.The 2nd component explains 28.7% of variance in the model and so on..

This explains how each principle component is marked from 0-7 ; is associated with the dataset.

2.6 Extract the eigenvalues, and eigenvectors.

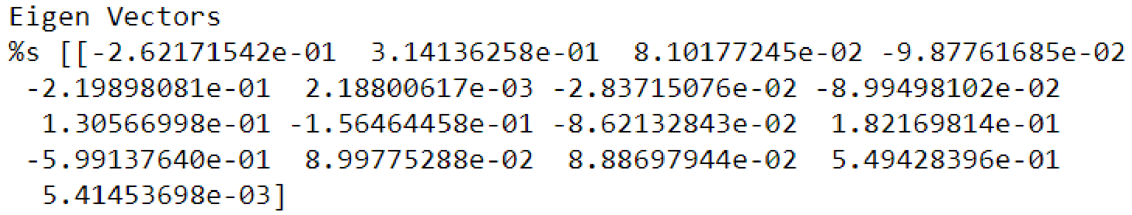


```
In [58]: eig_vals,eig_vecs=np.linalg.eig(cov_matrix)
         print('\n Eigen Values \n %s',eig_vals)
         print('\n Eigen Vectors \n %s',eig_vecs)

          Eigen Values
          %s [5.6625219  4.89470815 1.12636744 1.00397659 0.87218426 0.7657541
          0.58491404 0.5445048  0.42352336 0.38101777 0.24701456 0.02239369
          0.03789395 0.14726392 0.13434483 0.09883384 0.07469003]

          Eigen Vectors
          %s [[-2.62171542e-01  3.14136258e-01  8.10177245e-02 -9.87761685e-02
            -2.19898081e-01  2.18800617e-03 -2.83715076e-02 -8.99498102e-02
             1.30566998e-01 -1.56464458e-01 -8.62132843e-02  1.82169814e-01
            -5.99137640e-01  8.99775288e-02  8.88697944e-02  5.49428396e-01
             5.41453698e-03]
           [-2.30562461e-01  3.44623583e-01  1.07658626e-01 -1.18140437e-01
            -1.89634940e-01 -1.65212882e-02 -1.29584896e-02 -1.37606312e-01
             1.42275847e-01 -1.49209799e-01 -4.25899061e-02 -3.91041719e-01
             6.61496927e-01  1.58861886e-01  4.37945938e-02  2.91572312e-01
             1.44582845e-02]
           [-1.89276397e-01  3.82813322e-01  8.55296892e-02 -9.30717094e-03
            -1.62314818e-01 -6.80794143e-02 -1.52403625e-02 -1.44216938e-01
             5.08712481e-02 -6.48997860e-02 -4.38408622e-02  7.16684935e-01
             2.33235272e-01 -3.53988202e-02 -6.19241658e-02 -4.17001280e-01
            -4.97908902e-02]
           [-3.38874521e-01 -9.93191661e-02 -7.88293849e-02  3.69115031e-01
            -1.57211016e-01 -8.88656824e-02 -2.57455284e-01  2.89538833e-01
            -1.22467790e-01 -3.58776186e-02  1.77837341e-03 -5.62053913e-02
             2.21448729e-02 -3.92277722e-02  6.99599977e-02  8.79767299e-03
            -7.23645373e-01]
           [-3.34690532e-01 -5.95055011e-02 -5.07938247e-02  4.16824361e-01
            -1.44449474e-01 -2.76268979e-02 -2.39038849e-01  3.45643551e-01
```

The Eigen Values & Eigen Vectors were extracted using np.linalg.eig(cov_matrix) function. By extracting Eigen Values & Vectors the efficiency of performing a PCA is improved where it reduces the dimensions of the data.

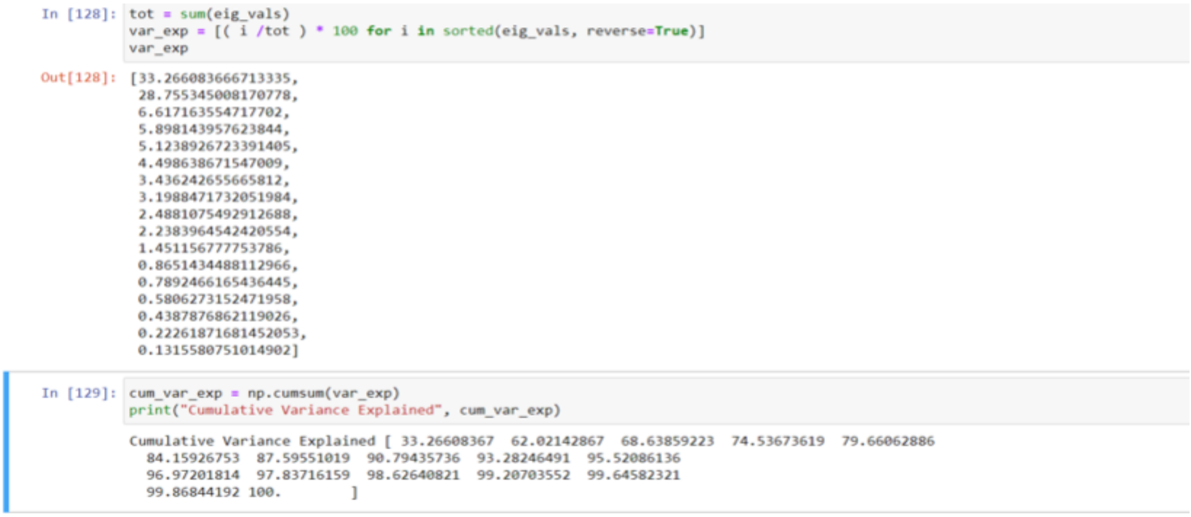
2.7 Write down the explicit form of the first PC (in terms of the eigenvectors. Use values with two places of decimals only).

EigenVector of 1st PC:



```
Eigen Vectors
%s [[-2.62171542e-01  3.14136258e-01  8.10177245e-02 -9.87761685e-02
  -2.19898081e-01  2.18800617e-03 -2.83715076e-02 -8.99498102e-02
   1.30566998e-01 -1.56464458e-01 -8.62132843e-02  1.82169814e-01
  -5.99137640e-01  8.99775288e-02  8.88697944e-02  5.49428396e-01
   5.41453698e-03]
```

2.8 Consider 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?



```
In [128]: tot = sum(eig_vals)
          var_exp = [( i /tot ) * 100 for i in sorted(eig_vals, reverse=True)]
          var_exp

Out[128]: [33.266083666713335,
           28.755345008170778,
           6.617163554717702,
           5.898143957623844,
           5.1238926723391405,
           4.498638671547009,
           3.436242655665812,
           3.1988471732051984,
           2.4881075492912688,
           2.2383964542420554,
           1.451156777753786,
           0.8651434488112966,
           0.7892466165436445,
           0.5806273152471958,
           0.4387876862119026,
           0.22261871681452053,
           0.1315580751014902]
```

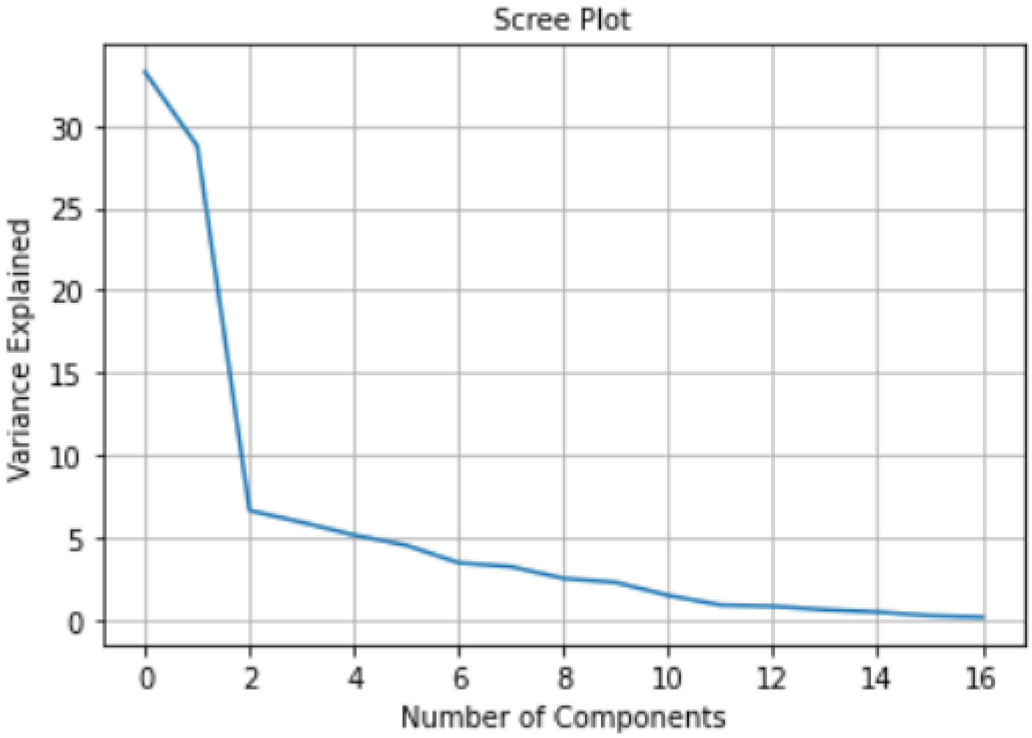
```
In [129]: cum_var_exp = np.cumsum(var_exp)
          print("Cumulative Variance Explained", cum_var_exp)

          Cumulative Variance Explained [ 33.26608367  62.02142867  68.63859223  74.53673619  79.66062886
            84.15926753  87.59551019  90.79435736  93.28246491  95.52086136
            96.97201814  97.83716159  98.62640821  99.20703552  99.64582321
            99.86844192 100.        ]
```

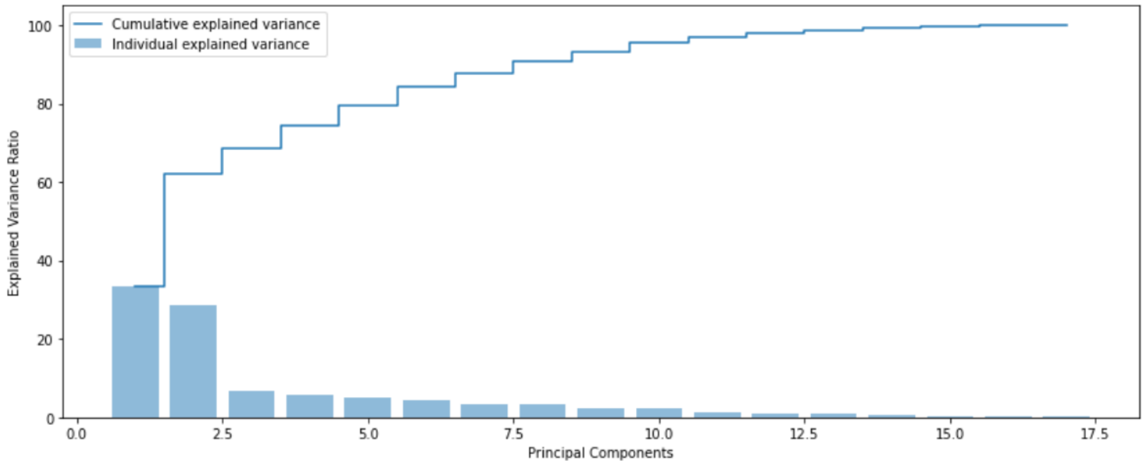
First, after determining the objective for conducting the PCA we decide to keep number of Eigen Values & Vectors as formulated through Python. We can proceed with the PCA components as 7 ;But depending on requirement 90% variation or 7 PCA dimensions will do good . Thus, we have now reduced the dimensionality for processing & arriving insights from the data..

We should then plot the cumulative sum of eigen values .If we divide the each value by the total of sum of eigenvalues before plotting the variables then the plot displays the fraction of total variance retained vs Number of eigen values.

SCREE PLOT:

The Scree Plot plotted below denotes the importance of factors, a sharp drop in the plot indicates that the subsequent factors are ignored.We can obserrve that the steep drop in variance is explained with increase in number of PC's

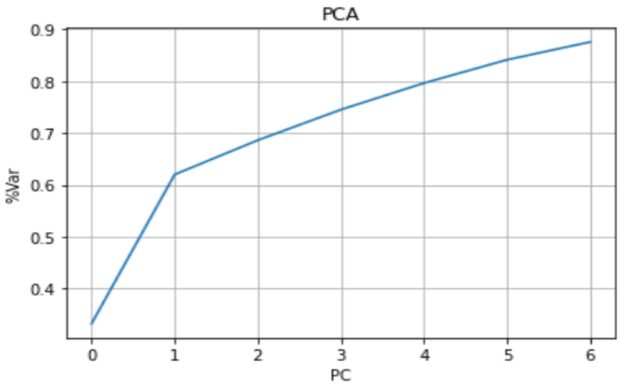




Based on the Cumulative explained variance & Individual explained variance we can plot a graph against Explained Variance Ratio which indicates the cumulative explained variance as well as Individual explained variance for each PC.

2.9 Explain the business implication of using the Principal Component Analysis for this case study. How may PCs help in the further analysis? [Hint: Write Interpretations of the Principal Components Obtained]

- ***PCA is a statistical technique which transforms & converts a set of observations of correlated variables into a set of values of uncorrelated variables helps in reduction of multidimensional data to lower dimensions while keeping most of the required information. By reducing, the dimensionality of the data ,variation is kept as much as possible.***
- ***By transforming the data into a continuous variable PCA was performed.***
- ***From the given Dataset the first feature captures about 33.3% of the variance ,while the first two captures 62.1% variance; goin on until 7 features which captures about 87.6% of variance within the dataset.***



- ***From 18 variables in the dataset , we have reduced to 7 components with the help of PCA which captures about 87.6% variance in the given dataset.***